Motion Blur in Digital Images

Analys, Detection and Correction of Motion Blur in Photogrammetry

by
Till Sieberth

A Doctoral Thesis
submitted in partial fulfillment of the requirements
for the award of
Doctor of Philosophy of Loughborough University

Abstract

Key words: Bundle Adjustment, Blur, Camera Calibration, Correction, Digital, Geometric, Image Processing, Photrogrammetry, Spatial Resolution, UAV

Unmanned aerial vehicles (UAV) have become an interesting and active research topic for photogrammetry. Current research is based on images acquired by an UAV, which have a high ground resolution and good spectral and radiometrical resolution, due to the low flight altitudes combined with a high resolution camera. UAV image flights are also cost effective and have become attractive for many applications including, change detection in small scale areas.

One of the main problems preventing full automation of data processing of UAV imagery is the degradation effect of blur caused by camera movement during image acquisition. This can be caused by the normal flight movement of the UAV as well as strong winds, turbulence or sudden operator inputs. This blur disturbs the visual analysis and interpretation of the data, causes errors and can degrade the accuracy in automatic photogrammetric processing algorithms. The detection and removal of these images is currently achieved manually, which is both time consuming and prone to error, particularly for large image-sets. To increase the quality of data processing an automated process is necessary, which must be both reliable and quick.

This thesis proves the negative affect that blurred images have on photogrammetric processing. It shows that small amounts of blur do have serious impacts on target detection and that it slows down processing speed due to the requirement of human intervention. Larger blur can make an image completely unusable and needs to be excluded from processing. To exclude images out of large image datasets an algorithm was developed. The newly developed method makes it possible to detect blur caused by linear camera displacement. The method is based on human detection of blur. Humans detect blurred images best by comparing it to other images in order to establish whether an image is blurred or not. The developed algorithm simulates this procedure by creating an image for comparison using image processing. Creating internally a comparable image

makes the method independent of additional images. However, the calculated blur value named SIEDS (saturation image edge difference standard-deviation) on its own does not provide an absolute number to judge if an image is blurred or not. To achieve a reliable judgement of image sharpness the SIEDS value has to be compared to other SIEDS values of the same dataset.

This algorithm enables the exclusion of blurred images and subsequently allows photogrammetric processing without them. However, it is also possible to use deblurring techniques to restor blurred images. Deblurring of images is a widely researched topic and often based on the Wiener or Richardson-Lucy deconvolution, which require precise knowledge of both the blur path and extent. Even with knowledge about the blur kernel, the correction causes errors such as ringing, and the deblurred image appears “muddy” and not completely sharp. In the study reported in this paper, overlapping images are used to support the deblurring process. An algorithm based on the Fourier transformation is presented. This works well in flat areas, but the need for geometrically correct sharp images for deblurring may limit the application. Another method to enhance the image is the ‘unsharp mask’ method, which improves images significantly and makes photogrammetric processing more successful. However, deblurring of images needs to focus on geometric correct deblurring to assure geometric correct measurements. Furthermore, a novel edge shifting approach was developed which aims to do geometrically correct deblurring. The idea of edge shifting appears to be promising but requires more advanced programming.

Contents

List of Figures **V**

List of Tables **VIII**

List of Equations **IX**

List of Program Code Fragments **X**

Nomenclature **XI**

1 Introduction **1**

1.1 Motivation for the thesis 3

1.2 Aims and objectives of the thesis 4

1.3 Contribution to knowledge 5

1.4 Structure 5

1.5 Software 6

2 Detailed overview and supporting concepts **9**

2.1 General literature review 9

2.1.1 Images and image processing 10

2.1.1.1 Image compression and colour space 10

2.1.1.2 The resolutions 14

2.1.1.3 Image processing 17

2.1.2 Photogrammetry 27

2.1.2.1 Aerial photogrammetry 28

2.1.2.2 UAVs 30

2.1.2.3 Blur prevention and preventive hardware modifications . 32

2.2 Generating motion blurred images 38
2.2.1 Problem of current blur methods 39
2.2.2 Shake table 40
2.2.3 Camera and camera settings 44
2.2.4 Determine image blur 48
2.2.5 Results 52
2.3 Chapter summary 52

3 Blur disturbs **54**
3.1 Literature review 55
3.1.1 Photogrammetric targets 55
3.1.1.1 Examples of photogrammetric targets 56
3.1.1.2 Coded target detection, identification and measurement . 57
3.1.2 Camera calibration and adjustment theory 58
3.1.2.1 Camera calibration 58
3.1.2.2 Adjustment theory 60
3.2 Dataset I - Influence of blur on camera calibration 61
3.2.1 Setup and composition of dataset 61
3.2.2 Camera calibration with blurred images 64
3.2.3 Result of camera calibration with blurred images 66
3.3 Dataset II - Influence of blur on 3D object coordinates 71
3.3.1 Setup and composition of the dataset 71
3.3.2 Calculation of 3D coordinates with blurred images 73
3.3.3 Result of 3D coordinate calculation with blurred images 76
3.4 Dataset III - Influence of blur on target detection 81
3.4.1 Setup and composition of the dataset 81
3.4.2 Target detection in blurred images 82
3.4.3 Results of target detection in blurred images 85
3.5 Dataset IV - Influence of blur on feature matching 90
3.5.1 Setup and composition of dataset 91
3.5.2 Feature detection in blurred images 91
3.5.3 Results of feature detection in blurred images 93
3.6 Discussion 94

3.7 Conclusion - Does blur disturb? . . . 97

4 Blur detection 99

4.1 Literature review . . . 99
4.1.1 Blur detection based on edge detection . . . 100
4.1.2 Blur detection based on frequency analysis . . . 102
4.1.3 Blur detection based on various other methods . . . 103
4.1.4 Summary . . . 106
4.2 Detecting Blurred Images . . . 106
4.2.1 The idea of the algorithm . . . 107
4.2.2 The algorithm . . . 108
4.2.2.1 Input image scaling . . . 111
4.2.2.2 RGB to HSV . . . 113
4.2.2.3 Re-blurring SVB image . . . 116
4.2.2.4 Edge detection and discrepancy calculation . . . 118
4.2.2.5 SIEDS calculation . . . 119
4.3 Results and limits of detection . . . 122
4.3.1 Critical parameters for the algorithm . . . 122
4.3.2 Calculated SIEDS . . . 124
4.3.3 Beyond SIEDS . . . 130
4.3.4 SIEDS for mathematical blurred images . . . 131
4.3.5 Other methods to calculate SIEDS . . . 132
4.4 Discussion and future work . . . 132
4.5 Conclusion . . . 134

5 Blur correction 136

5.1 Literature review . . . 136
5.1.1 Blur correction . . . 137
5.1.1.1 Point spread function and blur kernel . . . 137
5.1.1.2 Blind deconvolution . . . 138
5.1.1.3 Non-blind deconvolution . . . 140
5.1.1.4 Unsharp mask . . . 141
5.1.1.5 Summary . . . 143
5.2 Processing blurred images . . . 144

5.2.1 Frequency transfer method using image overlap 144
5.2.2 Edge shift - an overlap independent method 152
5.3 Results and limits of correction 157
5.3.1 Fourier transformation approach 157
5.3.2 Edge correction approach 159
5.4 Discussion 163
5.5 Conclusion 164

6 Application to UAV imagery **165**
6.1 Processing 165
6.1.1 Blur detection 166
6.1.1.1 UAV datasets 166
6.1.1.2 Close range 172
6.1.1.3 Results of blur detection in real images 175
6.1.2 Blur correction 175
6.2 Conclusion 178

7 Final conclusion **180**
7.1 Future work 182

References **185**

A Additional material **196**
A.1 Blur disturbs 196

B Publications **200**
B.1 Double-blind peer reviewed 200
B.2 Peer reviewed 221

C Patent **241**
C.1 Coarse patent search 241
C.2 Detailed patent search 243

D Program documentation **248**

List of Figures

Figure 1.1 Image acquired by an UAV. 2
Figure 1.2 Examples two UAVs used to collect datasets. 2

Figure 2.1 Comparison of vector and raster graphic. 11
Figure 2.2 Different compression level of JPEG. 13
Figure 2.3 Visualisation of different colour spaces. 14
Figure 2.4 Histogram operations on an image. 18
Figure 2.5 Blur filter operations on an image. 20
Figure 2.6 Edge detection on an image. 23
Figure 2.7 Morphological filter operations. 24
Figure 2.8 2D Fourier domain transformation. 24
Figure 2.9 Perspective transformation. 26
Figure 2.10 Perspective transformed image. 26
Figure 2.11 Aerial photogrammetry flight path. 28
Figure 2.12 Ground control points. 30
Figure 2.13 Sharp target vs. blurred targets. 31
Figure 2.14 Modulation transfer function for the used optical system. 34
Figure 2.15 Different kind of motion blur. 38
Figure 2.16 APS 400 ELECTRO-SEIS. 41
Figure 2.17 Shake table setup. 41
Figure 2.18 Shake table control signal. 44
Figure 2.19 Comparison of different film speeds. 46
Figure 2.20 Images acquired from shake table. 47
Figure 2.21 shake table measurements. 49
Figure 2.22 Strongest blurred images. 50

Figure 3.1 Various targets. 56
Figure 3.2 Circular shaped targets. 57
Figure 3.3 First dataset setup. 62
Figure 3.4 Detection of blurred targets. 65
Figure 3.5 Detection of targets for camera calibration. 66
Figure 3.6 Semi-automatic target detection and measurement. 67
Figure 3.7 Results of camera calibration with blurred images. 69
Figure 3.8 Second dataset set up. 72
Figure 3.9 Detection of blurred targets. 75
Figure 3.10 Comparison of similar camera displacement for Nikon D80 and D7000. 77
Figure 3.11 Discrepancies between blurred and sharp set for Nikon D80. 78
Figure 3.12 Discrepancies between blurred and sharp set for Nikon D7000. . . . 79
Figure 3.13 Blurred target of D7000. 79
Figure 3.14 Third dataset setup. 82
Figure 3.15 Chromatic aberration on black target. 85
Figure 3.16 Target size referenced between sharp and blurred image. 86
Figure 3.17 Size of blurred targets related to camera displacement. 87
Figure 3.18 Influence of blur on small targets. 89
Figure 3.19 Matching feature points. 93
Figure 3.20 Filtered matches of feature points. 95

Figure 4.1 Example dataset with sharp and blurred edge. 100
Figure 4.2 Example of fast Fourier transformation on dataset with sharp and blurred edge. 102
Figure 4.3 Example showing re-blurred images. 104
Figure 4.4 Test results of Crete et al. (2007). 105
Figure 4.5 Comparing of insets of UAV images. 107
Figure 4.6 Flow chart of developed algorithm. 109
Figure 4.7 Comparing of insets of UAV images. 110
Figure 4.8 Effect of reducing image resolution. 113
Figure 4.9 Influence of image scaling on calculated SIEDS. 123
Figure 4.10 Influence of additional blur on calculated SIEDS. 124
Figure 4.11 Different image channels used for SIEDS calculation. 125

Figure 4.12 Calculation of SIEDS for homogeneous image. 127
Figure 4.13 Calculation of SIEDS for homogeneous pattern. 128
Figure 4.14 Calculation of SIEDS for steep edge. 129
Figure 4.15 SIEDS calculated for images blurred using Matlab. 131

Figure 5.1 PSF and blur kernel as representation of blur. 138
Figure 5.2 Lucy-Richardson deconvolution result. 140
Figure 5.3 Unsharp mask image enhancement principle. 142
Figure 5.4 Unsharp mask image enhancement example. 143
Figure 5.5 Flow of blur correction using Fourier transformation. 145
Figure 5.6 Deblurring using Fourier approach. 152
Figure 5.7 Flow of blur correction using edge shift. 153
Figure 5.8 Displacement of edges in blurred images. 153
Figure 5.9 Blur kernel. 154
Figure 5.10 Direction of displacement of blurred edges. 155
Figure 5.11 Holes cause by 'ripping' apart edges. 156
Figure 5.12 Detected and identified targets in deblurred image. 158
Figure 5.13 Detected and matched features in deblurred image. 158
Figure 5.14 Edge shifting result. 160
Figure 5.15 Edge shifting in blurred image. 161

Figure 6.1 Examples of real application UAV images. 167
Figure 6.2 SIEDS for rotary wing UAV images. 168
Figure 6.3 Example images with calculated SIEDS for rotary wing UAV images. 169
Figure 6.4 SIEDS for fixed wing UAV images. 170
Figure 6.5 Example images with calculated SIEDS for fixed wing UAV images. 171
Figure 6.6 Inter-comparability of UAV blur detection results. 172
Figure 6.7 SIEDS for terrestrial images. 173
Figure 6.8 Example images with calculated SIEDS for terrestrial images. . . . 174
Figure 6.9 Two overlapping images for deblurring. 176
Figure 6.10 Feature matching in UAV images. 176
Figure 6.11 Feature detection and matching in enhanced images. 177
Figure 6.12 Enhanced UAV images. 177

List of Tables

Table 2.1 Comparison of graphic file format storage use. 12
Table 2.2 Classification of UAVs 31
Table 2.3 Accelerometer properties. 48
Table 2.4 Overview over generated test images. 51

Table 3.1 Camera calibration dataset 63
Table 3.2 Object coordinate dataset 73
Table 3.3 Detection of blurred targets with Nikon D80. 74
Table 3.4 Detection of blurred targets with Nikon D7000. 76
Table 3.5 Target detection dataset 83
Table 3.6 Comparison between counted and theoretical target size. 84
Table 3.7 Feature detection and matching dataset 91
Table 3.8 Camera displacement vs. detected and correctly referenced feature points. 93

Table 4.1 Comparison of Crete (2007) to developed method. 133

Table 6.1 UAV datasets used for case application. 166
Table 6.2 Comparison of SIEDS for complete image and image patches. 170
Table 6.3 Terrestrial datasets used for case application. 172
Table 6.4 Feature detection in deblurred images. 176

Table A.1 Number of detected targets (out of 3) in Nikon D80 dataset 196
Table A.2 Number of detected targets (out of 3) in Nikon D7000 dataset . . . 198

Table C.1 Search details for ESPACNET 245
Table C.2 Search details for USPTO 247

List of Equations

Equation 2.1 Image file size calculation. 12

Equation 2.2 Ground sampling distance. 16

Equation 2.3 Mathematical convolution. 21

Equation 2.4 2D mathematical convolution. 21

Equation 2.6 Collinearity equation. 29

Equation 2.7 Harmonic oscillation. 42

Equation 3.1 Shorten observations. 60

Equation 3.2 Adjust unknowns. 60

Equation 3.3 Adjust observation. 61

Equation 3.4 Theoretical and practical target size relation. 87

Equation 3.5 Blurred target size. 88

Equation 3.6 Camera displacement of UAVs. 89

Equation 3.7 Target size for UAV flights. 90

Equation 4.1 Damping function. 130

Equation 5.1 Bayes's theorem. 139

List of Program Code Fragments

3.1 Feature detection and matching 91

4.1 Scaling an image 111
4.2 Colour space conversion 113
4.3 Creating copy of image matrix 116
4.4 Low-pass filtering of an image matrix 117
4.5 High-pass filtering of an image matrix 118
4.6 Calculation of the discrepancy image. 119
4.7 Calculation of SIEDS 120

5.1 Unsharp mask method 142
5.2 Perspective transformation parameter from matched features 144
5.3 Calculating overlap area from perspective transformation parameters . . . 146
5.4 Warping the image using perspective transformation 148
5.5 Fourier transformation 148
5.6 Adding frequencies 149
5.7 Inverse Fourier transformation 150

Nomenclature

1D	One dimension(al).
2D	Two dimension(s/al).
3D	Three dimension(s/al).
AMC	Angular motion compensation.
CCD	Charge-coupled device.
CMOS	Complementary metal-oxide-semiconductor.
CMYK	cyan, magenta, yellow, key.
DFT	Discreet Fourier transformation.
DN	Digital number.
DSLR	Digital single lens reflex.
DSM	Digital surface model.
EMS	Electromagnetic spectrum.
EPO	European Patent Office.
FFT	Fast Fourier transformation.
FMC	Forward motion compensation.
fps	Frames per second.
GCP	Ground control point.
GNSS	Global navigation satellite systems.
GSD	Ground sampling distance.
HCP	Height control point.
HSI	hue, saturation, intensity.
HSL	hue, saturation, luminance.

HSV ..... hue, saturation, value.

IMU ..... Inertial measurement unit.

JNB ..... Just noticeable blur.

JND ..... Just noticeable difference.

JPEG ..... Joint Photographic Experts Group.

LIDAR ..... Light detection and ranging.

LSM ..... Least squares matching.

MOS ..... Mean opinion score.

MTF ..... Modulation transfer function.

OpenCV ..... Open computer vision.

pixel ..... Picture element.

PNG ..... Portable Network Graphics.

POI ..... Points of interest.

POV ..... Point of view.

PSF ..... Point spread function.

RAD ..... Ringed automatically detected.

RGB ..... red, green, blue.

ROI ..... Region of interest.

SIEDS ..... Saturation image edge difference standard-deviation.

SIFT ..... Scale-Invariant Feature Transform.

SURF ..... Speed-Up Robust Features.

SVB ..... Saturation, value and blue.

TDI ..... Time delay integration.

TP ..... Tie point.

UAS ..... Unmanned aerial system.

UAV ..... Unmanned aerial vehicle.

USPTO ..... United States Patent and Trademark Office.

WIPO ..... World Intellectual Property Organisation.

Mathematical Symbols

$(\Delta x', \Delta y')$ ..... Optional additional parameters.

(X, Y, Z) ..... 3D coordinate.

(X_0, Y_0, Z_0) ..... Camera perspective centre.

$\Delta_{x,y}$ ..... Difference image.

δ ..... Damping ratio.

$\hat{L}$ ..... Adjusted observations.

$\hat{X}$ ..... Adjusted unknowns.

$\hat{x}$ ..... Parameter vector.

ω ..... Angular frequency.

$\otimes$ ..... Convolution operation.

bitdepth ..... Bit depth of the image.

A ..... Design matrix.

a ..... Acceleration.

A_{sensor} ..... Sensor area.

Amp ..... Amplitude.

b ..... Blur.

c_k ..... Principal distance, focal length.

d ..... Camera displacement.

$F(x, y)$ ..... Input image for matching.

f ..... Function to be convolved.

fqu ..... Frequency.

$G(x, y)$ ..... Convolution filter.

g ..... Convolution function.

$G^a_{x,y}$ ..... Position of the anchor.

$G_{rows,cols}$ ..... Size of filter.

h ..... Distance to object, flight altitude.

$I(x, y)$ ..... Input image to be convolved.

I_{bit} ..... Image file size in bit.

I_{byte} ..... Image file size in byte.

$K(x, y)$ ..... Image to be convolved.

k ..... Convolved function.

L ..... Observations.

l ..... Shortened observations.

L_0 ..... Initial observations.

n ..... y-offset from zero.

$n_{channels}$ ..... Number of channels in the image.

n_{pixels} ..... Image size in pixel.

$P(A)$ ..... Probability of A.

$P(B)$ ..... Probability of B.

P ..... Weight matrix.

$r_{i,j}$ ..... Orthogonal rotation parameter.

s ..... Distance.

s_{PX} ..... Pixel size on ground, GSD.

s_{px} ..... Pixel Size on sensor.

$T(x, y)$ ..... Template to be matched.

t ..... Time.

t_0 ..... Minimal target size.

t_b ..... Blurred target size in pixel.

t_I ..... Target size in image space.

t_O ..... Target size in object space.

t_s ..... Theoretical target size in pixel.

v ..... Improvement vector.

vel ..... Velocity.

w ..... Damping ratio.

x', y' ..... Photo coordinates.

x'_0, y'_0 ..... Principal point offset.

X_0 ..... Initial value of unknowns.

x_0 ..... Start position.

1 Introduction

Unmanned aerial vehicles (UAV) have recently become an interesting and extensively researched topic. Remote controlled drones are used for military applications (Associated Press, 2014; BBC, 2014; Wilson, 2014), for security and surveillance (Agence France-Presse, 2014; Merrill, 2014; Skahill, 2014), search and rescue operations and for monitoring of natural disasters (Bidwell, 2014; Jefferies, 2014; Killalea, 2014). UAVs are particularly useful due to their small size, low cost and the removal of an on-board pilot. The on-board pilot was made redundant using advanced technology, faster and more powerful computers, increasing knowledge about aerodynamics and development of newer, lighter and more robust materials. This enables full automatic flight without the risk of endangering human life and at low costs.

Most scientific and civil applications focus on small scale UAVs of only a few decimetre in size, which are remote controlled or have an autopilot for fully automated flight from take-off to landing. Due to the small size they are exempt from airport fees or registration costs like large aircraft. However, many countries have introduced new laws and regulations that restrict the use of small scale UAVs. Some regulations limit the weight and size of small scale Unmanned aerial systems (UAS), which subsequently limits the payload. Due to this limitation, UAVs have to be equipped with small and light weight sensors. One type of sensor often used are cameras. Cameras used in UAVs are often consumer-grade cameras, which are both light weight and small in size. They are also much cheaper than aerial imaging systems, but also less accurate and are particularly influenced by external effects.

Typically, consumer grade cameras are used in particular with interchangeable lenses or digital single lens reflex (DSLR) cameras. These cameras provide acceptable images and

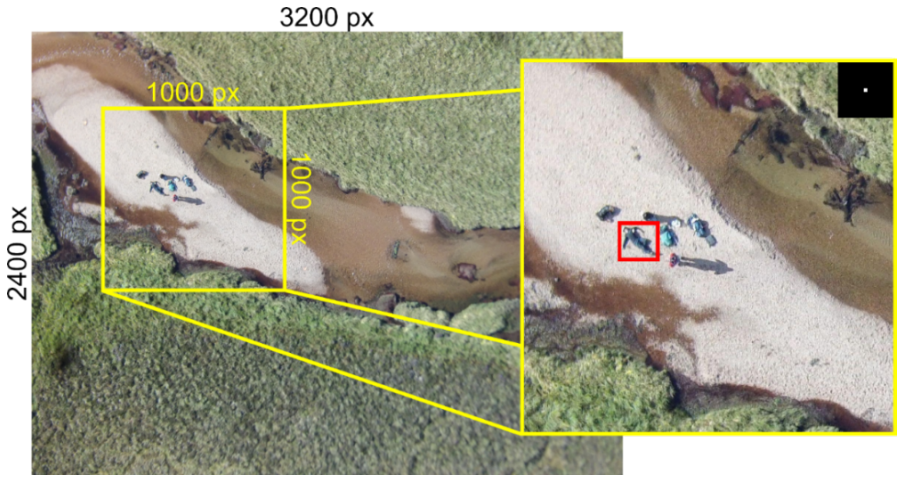
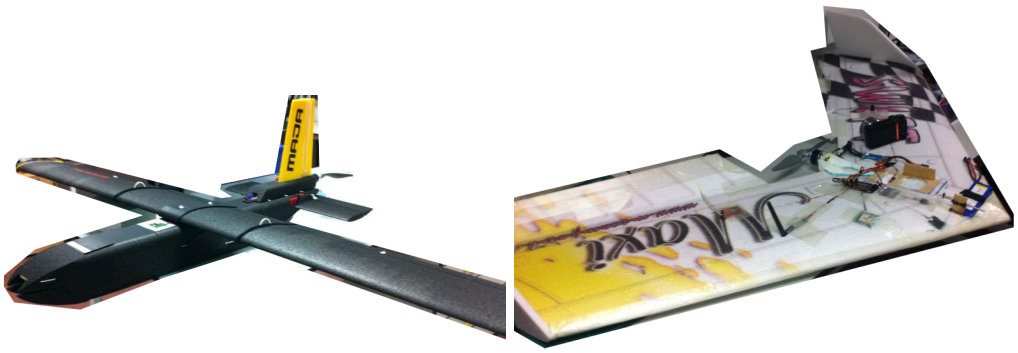


Figure 1.1: Image acquired by an UAV.



(a) Example of a fixed wing UAV.

(b) Example of a wing design UAV.

Figure 1.2: Examples two UAVs used to collect datasets.

are widely used for mapping and monitoring of sites, in scientific research projects and in commercial projects (Figure 1.1). Images acquired are of value because they provide a good ground sampling distance (GSD) with a good spectral and radiometric resolution, due to the low flight altitudes and high resolution sensors. The high geometrical resolution makes UAV images especially interesting for aerial photogrammetry, which focuses on the reconstruction of '*the position, orientation shape and size of objects from pictures*' (Kraus, 2007). To reconstruct an object from an image, two dimensional (2D) image coordinates have to be measured in the image space and have to be transformed to three dimensional (3D) object coordinates in the object space. To get accurate 3D coordinates the 2D image coordinates have to be measured accurately.

1.1 Motivation for the thesis

The motivation for this thesis is based on the photogrammetric use of images. To get accurate photogrammetric results it is necessary to measure accurately 2D image coordinates. To achieve this sharp images are required. If it is not possible to accurately measure coordinates in the image space, calculations based on these image coordinates are subsequently incorrect. In most photogrammetric applications sharp images are achieved by using a camera tripod and an external shutter release. This prevents the camera from exhibiting any movement during image exposure and creates a sharp image.

UAVs can be seen as unstable imaging platforms during flight conditions and sudden motions are commonly experienced during image acquisition. These sudden movements can be caused by gusts, turbulence, vibrations of the engine or even operator inputs and can cause significant image blur. Even the forward motion of a fixed wing UAV has a degrading effect on images due to the constant displacement of the camera body whilst the shutter is open. Due to motion blur, measuring 2D coordinates in the image space becomes difficult and inaccurate. Furthermore, other automatic image processing techniques could be expected to be influenced. As a first task the impact of this negative influence has to be assessed, examined and quantified.

If blur has a degrading effect on subsequent image processing, an obvious strategy is to exclude blurred images from further processing. This requires the detection of blurred images. Detection of blurred images is traditionally carried out manually by an operator assessing images presented visually on a computer screen. The operator assesses whether an image is sharp or blurred and acceptable for further processing. This has to be repeated for many, perhaps hundreds of images in a sequence and is exhausting for the eyes, prone to errors and time consuming. An automatic filtering algorithm would help to speed-up the process and to prevent errors and tired eyes.

The filtering of blurred images reduces the number of images available for post processing. This can significantly reduce the accuracy and precision of post processing results or even make post processing impossible. To avoid any problems during post processing

there is a danger that too many images maybe excluded. By excluding many images the number of overlapping images is reduced and degrades the accuracy of subsequent image coordinate measurements. However, redundant coordinate measurements are important to ensure accurate adjustment of errors. To provide a sufficient number of images it might be required to deblur some of the blurred images to maintain a strong image configuration. However, deblurring is an ill-posed problem, which cannot be solved. To solve this problem deblurring requires an iterative process. However, the process may not converge and it may return incorrect results. Due to the iterative process deblurring is time extensive and it is unstable when carried out for each image in a large UAV image dataset. A deblurring algorithm capable of enabling photogrammetric operations could help ensure accurate post processing.

1.2 Aims and objectives of the thesis

The initial project name was '*Efficient and Reliable Algorithms for Image Deblurring*' with the focus on UAV images. However, it was established early that the project had to be extended and needed a more extensive remit than simply blur correction. Surprisingly, the influence of motion blur on the results of photogrammetric processing has not yet been analysed, which identifies the aim of this project to analyse, detect and correct blur in digital images used for photogrammetric procedures. Specific objectives include:

- Reviewing currently available and used techniques for image blur, blur detection and deblurring. Also review important photogrammetric techniques used for processing UAV imagery.
- Generating motion blurred images with precisely known characteristics and comparing these to other methods used for image blurring.
- Analysing if and whether photogrammetry and image processing procedures are influenced by blurred images. If so, identify an appropriate threshold necessary to isolate and exclude blurred images.
- Develop an efficient and reliable algorithm for image blur detection.
- Develop an algorithm for blur correction acceptable for UAV image sets and photogrammetric purpose.

- Confirm results of laboratory tests by applying and testing UAV imagery acquired with a small scale, home-made fixed wing UAV.

1.3 Contribution to knowledge

This thesis contributes to current knowledge in the field of computer vision and image processing. Blur is a widely researched topic in computer vision. However, most of the research concentrates on **correcting** blur but rarely on determining the **influence** of blur on image processing or even photogrammetric processing. The main contribution achieved in this research is the determination of the influence of motion blur images on photogrammetry. This general contribution has been achieved through:

- Producing blurred images with known characteristics using a new approach involving a shake table.
- Analysing if and how much blur influences photogrammetry and image processing using typical processing techniques.
- Developing a blur detection algorithm for large image datasets. Although this has been evaluated for UAV imagery it has wider applicability.
- Development of a range of blur correction methods applicable for UAV images.

1.4 Structure

After careful consideration it was decided, that this thesis should be structured and presented in seven parts plus an appendix.

- The first chapter is an introduction, providing the initial context and basic justification for the research conducted.
- The second chapter provides a detailed review of the research domain. It includes a literature review to understand basic principles associated with image processing and photogrammetry, which will be helpful in the following chapters. Furthermore, the basic setup of the shake table will be explained.

- The third chapter analyses how blurred images influence photogrammetry and image processing. It starts with an explanation about photogrammetry and photogrammetric targets, and how subsequent measurements and calculations affect the quality of results. It is followed by the experimental research that was carried out, to analyse the influence of blur on these processes.
- The fourth chapter examines the detection of blurred images in an image dataset. The literature review gives an insight on techniques adopted previously before the newly developed method is explained and discussed.
- The fifth chapter concentrates on blur correction and explains the methods previously developed in the field of computer vision. A method aiming to correct photogrammetric targets is applied to the shake table images and results examined.
- The sixth chapter applies the techniques developed in the previous three chapters to a new set of UAV imagery and discusses the applicability of the methods.
- This thesis ends with a review of the work conducted. It will discuss the collected results and suggest future improvements. Furthermore, it includes concluding remarks to connect the findings.

An appendix consists of three parts.

- The first part contains additional material supporting the research presented in this thesis.
- The second part presents publications produced during this research project.
- The third part explains the patent progress regarding the blur detection algorithm.

The attached DVD also contains the used program codes and documentation for these codes. However, the program code provides only code fragments as a proof of concept. They do not represent an independently working program.

1.5 Software

The processing of digital imagery is clearly an important part of this research. Processing images requires special software, which can return different results to other software even when providing the same functions. The software used will be presented to ensure the

same outcome. Also the visualisation of images in this research depends strongly on the way they are displayed, so many images have been modified before presentation.

Image processing was carried out using *C++*. To make the results better reproducable and to comply with the university regulations to include all program codes, *C++* code sniplets were included in the text. The *C++* program code includes functions of the open computer vision (OpenCV) library (OpenCV Dev Team, 2014). OpenCV is an open source programming library providing functions and methods to process images. To use these functions the OpenCV library has to be included in a *C++* program code. OpenCV provides a range of functions and settings to customise the results of image processing. However, as an open source project OpenCV relies on volunteer developers for further developments and requires a community of users to verify published functions. OpenCV v2.4.0 was used on large image sets. The software *ImageJ* was used for examples presented in this thesis. The benefits of using *ImageJ* are that long program codes were avoided which have to be developed and debugged.

Photogrammetric processes were carried out using *PhotoModeler* (Eos Systems Inc., 2014). It is a software package for processing close range imagery for spatial measurements. It provides the necessary functionality for automatic, semi-automatic and manual photogrammetric processing. *PhotoModeler* also has the advantage of being customisable using *Visual Basic*. It also supports the export of data and the import of additional data using either, the internal interface or external plug-ins. *PhotoModeler* provides algorithms for target detection, camera calibration and which are typical of a range of photogrammetric software packages.

Matlab was used to analyse the results and numbers returned by *C++* programs, *ImageJ* and *PhotoModeler* (The MathWorks, Inc., 2015). *Matlab* provides easy to access functions to process and visualise data and is often used in research. In this research it was used to control the shake table, analyse *PhotoModeler* and *ImageJ* output and to generate diagrams. The produced diagrams are saved as vector graphics to ensure precise and clear visibility.

Inkscape (2015) was used to modify vector graphics. *Inkscape* is an open source program for creating and manipulating line art graphics. For raster graphics *GNU image ma-*

nipulation program (GIMP) was used (The GIMP Team, 2015). It provides tools to manipulate images and was used to edit images for the figures in this document.

2 Detailed overview and supporting concepts

This chapter will introduce key background information which are required for this thesis. There are three distinct contributions regarding blur in this thesis each within three related but distinct topics. Consequently the thesis is structured into three dedicated chapters, each with their own separate literature review and discussion. However, this chapter aims to clarify common concepts used in each of the subsequent chapters. At first a few basic principles concerning images and image processing are presented. This includes how images are used in photogrammetry, how they are acquired and introduces UAVs as platforms for aerial photogrammetry. Furthermore, it will explain how the blurred images used in this thesis were generated, before concluding with a chapter summary.

2.1 General literature review

This first literature review will provide general information and methods, which are required to understand the subsequent chapters of the thesis. This literature review explains how digital images are represented and how digital images are processed. Basic principles of image processing are presented before some initial photogrammetric algorithms are explained. The literature review concludes with an examination of UAV platforms and which hardware modifications are actually able to prevent motion blur in images.

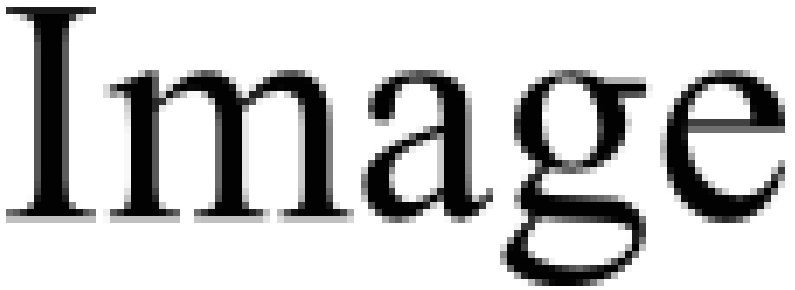
2.1.1 Images and image processing

Images are '*a representation of the external form of a person or thing ...*' more precisely '*A visible impression obtained by a camera ... or other device, or displayed on a computer or video screen ...*' (Oxford University Press, 2014). A digital image recorded by an electronic sensor or visualised by an electronic device is composed of a 2D matrix of picture elements (pixel). The pixels on a sensor are made of sub-pixels, which record one of the three additive colours at a range of wavelength. These include: red (500-700 nm), green (450-630 nm) and blue (400-500 nm) (RGB) (Ernstoff et al., 1977). The sub-pixels can be co-located in a range of ways, either side by side (Ernstoff et al., 1977), arranged in a Bayer filter mosaic (Bayer, 1976) or other arrangements. To record just one specific wavelength representing blue, green or red, each of the sub-pixels is equipped with a colour filter, which lets only light of a specific colour or wavelength pass through (Ernstoff et al., 1977). The light density illuminating the sub-pixels is recorded. Combining the different recordings of the sub-pixels results in different colours. The recording for each sub-pixel is stored as a numerical value called the digital number (DN), which are each written in separate 2D matrices and stored together in an image file (Lillesand et al., 2008).

2.1.1.1 Image compression and colour space

DNs can be stored in image files in two main ways. The area filling raster graphics are the better known version of storing DNs. Highly accurate vector graphics are the second method used to store DNs. The latter stores the DN for a specific point together with an accurate coordinate, enabling visualisation with an infinite spatial resolution (Figure 2.1(a)). Vector graphics are mostly used for line-based graphics (Murray and van Ryper, 1996). Raster graphics instead, store the DN in a 2D matrix (raster), which limits the visualisation to one cell of the raster. However, this can never be more accurate than the area of each pixel (Figure 2.1(b)). In contrast the accuracy of a vector representation is higher because each edge and point can be represented with high resolution coordinates. The disadvantage of vector representations for graphical imagery is however, that they are not "area filling" but just contain points and edges,

(a) Vector graphic. Infinite spatial resolution.



(b) Raster graphic. Limited spatial resolution.

Figure 2.1: Comparison of vector and raster graphic.

hence raster graphics are '*particularly well-suited for . . . real-world images*' (Murray and van Ryper, 1996) because they cover a complete area with DNs. Hence, this format is widely used and known. Vector graphics are normally generated by software while raster graphics are recorded using image sensors. For UAVs raster graphics are used because they are acquired with image sensors and the graphics format used and processed in this presented research. Typical file formats for raster graphics are *Windows Bitmap* (.bmp), *Graphics Interchange Format* (.gif) and *Joint Photographic Experts Group* (JPEG) files (.jpg/.jpeg) but many more are known.

Image compression The reasons for so many different file formats lies within the design of the image and file formats. A typical consumer camera will record millions of pixels, which is a vast amount of data that has to be stored in a file. Storing them by 'writing' every DN for each cell of the raster to the file would result in huge image files. As example a 16 mega pixels image recording RGB with 24 bit per channel would result in

137 MB large file (Equation 2.1).

$$
\begin{aligned}
I_{bit} &= n_{pixels} \cdot \text{bitdepth} \cdot n_{\text{channels}} \\
I_{byte} &= I_{bit}/8\text{bit}
\end{aligned}
$$

$$
\begin{aligned}
I_{bit} &\ldots \text{Image file size in bit} \\
I_{byte} &\ldots \text{Image file size in byte} \\
n_{pixels} &\ldots \text{Image size in pixel} \\
\text{bitdepth} &\ldots \text{Bit depth of the image} \\
n_{channels} &\ldots \text{Number of channels in the image}
\end{aligned}
\tag{2.1}
$$

Hence, it is normal to compress the information to prevent files becoming too large. There are broadly two compression approaches, distinguished by "lossless" and "lossy" image compressions. A lossy compressed image cannot be restored completely as during compression some information of the image is not stored. Lossless compressed images however, can always be restored to their original values without loss of information. JPEG is a lossy image compression technique, which can result in incorrect DNs and an imprecise visualisation of images (Pennebaker and Mitchell, 2004). In contrast, *Portable Network Graphics* (PNG) (.png) is a lossless image file format (Roelofs, 2014). However, there are also file formats which support lossy and lossless compressions methods for example the *Tagged Image File Format* (.tif/.tiff) (Adobe Developers Association, 1992). Table 2.1 shows that different file formats, which reduce the same input file to different sizes. Dependent on the input image the best compression method changes. Also the

File type	File Size
.pdf (vector input image)	9 KB
.gif (lossless)	4 KB
.jpg (lossy)	10 KB
.tiff (lossless)	1407 KB
.png (lossless)	1410 KB
.bmp (lossless)	1876 KB

Table 2.1: Comparison of storage used by various raster graphic file formats using the same vector graphic as input.

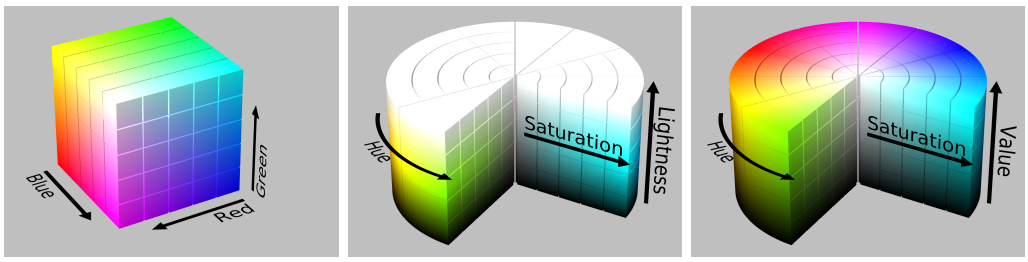
(a) Cropped JPEG image with low compression. File size 3346 KB.

(b) Cropped JPEG image with high compression. File size 300 KB.

Figure 2.2: Different compression level of JPEG.

compression strength influences the size of a file. Figure 2.2 shows an enlarged patch of a JPEG image with both, a low (Figure 2.2(a)) and high compression level (Figure 2.2(b)). It is possible to identify that the highly compressed image exhibits a loss of detail and introduces noise in the colours, compared to the low compressed image does not.

Colour space Another difference between the file formats is the colour space they use. TIFF supports the cyan, magenta, yellow and key (CMYK) 'colours'. Cyan, magenta and yellow are the subtractive colours, which can be generated by mixing RGB (Figure 2.3(a)). CMYK is an important colour space because CMYK represent the colours that are used by printers due to 'key', which is used to align the channels, which makes TIFF popular for printing applications. This popularity for printing is also because of the key channel which is used to align cyan, yellow and magenta during printing. Beside the CMYK colour space the *YCbCr* is used for digital images by JPEG. *YCbCr* is the digital version of the analogue *YPbPr* television signal. *Y* carries the luminance and *Pb* and *Pr* the difference between blue and red respectively and luma (Pennebaker and Mitchell, 2004). Luminance is also used in the hue, saturation and luminance (HSL) colour space (Figure 2.3(b)). This colour space is different from the previous colour



(a) RGB, CMYK (b) HSL (c) HSV

Figure 2.3: Visualisation of different colour spaces by (SharkD, 2010).

spaces because it reduces colour information to the hue of one channel only. Luminance and saturation provide the information necessary to define how the colour appears, if it is dark, bright, pale or saturated. The hue, saturation and value (HSV) (Figure 2.3(c)) and hue, saturation and intensity (HSI) colour space are similar to HSL (Ford and Roberts, 1996). The latter three colour spaces containing hue are often used for image editing. This is because image editing often aims to adjustment the contrast, exposure and colours, which are easily accessible by changing saturation, luminance/value/intensity and hue. These colour spaces are closely related to the RGB colour space by linear equations, which makes transformations between RGB and 'hue' colour spaces easy. There are various other colour spaces but a particularly important space to mention is the RGB colour space. This colour space is of high significance because the cones in the human eye are sensitive to light of the wavelength red (500-700 nm), green (450-630 nm) and blue (400-500 nm) (Hunt, 2004). Also computer displays use the RGB colour space to display content.

2.1.1.2 The resolutions

Raster images have multiple resolutions. Commonly known is the geometrical resolution, which is represented by the number of pixels. However, there are altogether four different resolutions, which are important for images.

Spectral resolution The wavelength recorded is dependent on the recording device. A normal consumer grade digital camera records blue, green and red light in three different

channels. However, they often have sensitivity to near-infrared light, so wavelengths above 700 nm are recorded in the red channel. This is different to the human eye, which is not sensitive to near-infrared light (www.eye.de, 2014). This should be prevented by infrared filters. However, this shows already the capability of image sensors to record more of the electromagnetic spectrum (EMS) than can be perceived by the human eye. Hyper-spectral cameras are available also, which can record many narrow bands of the EMS to generate a nearly continuous spectral curve for every pixel (Kurz et al., 2011). A spectral curve represents the amount of reflected EMS at a range of wavelength of an object (Lillesand et al., 2008). The higher the spectral resolution, the more accurate the spectral curve for each pixel. Every reflection value in a small wavelength range is recorded in a separate 2D matrix. Visualising one matrix would return a grey scale image. Hence the recorded values are also called grey values.

Radiometric resolution Grey values represent the reflected energy recorded for a range of channels (Lillesand et al., 2008). The denser the reflection, the higher the grey value. The range of grey values is defined by the radiometric resolution. The radiometric resolution is often defined as 8 bit per channel, which is equal to 256 grey values ranging from 0 to 255. There are also more precise radiometric resolutions, with much higher resolutions for example 32 bit per channel or more (Adobe Developers Association, 1992). However, this high precision cannot be differentiated by the human eye (Steinhardt, 1936).

Temporal resolution A less important resolution in photography is temporal resolution. The temporal resolution defines how many images can be taken in a certain time period. This is especially important for videography, to ensure that a film appears as a moving picture and not as many still images in a fast sequence. However, temporal resolution can be also defined as how often image acquisition of the same area is repeated. For example, the *Landsat*-satellite has a repeated coverage of 16 days (NASA, 2015). So every 16 days the satellite passes over the same area capturing a new image. This temporal resolution is important to detect changes in the test area, occurring over a certain time period.

Spatial resolution The resolution most often referred to and considered important for most consumers is spatial resolution. The spatial resolution defines how well a sensor can record spatial differences (Lillesand et al., 2008). The spatial resolution is directly related to the GSD by the distance to the object and the lens used with the camera. Knowing the sensor size, number of pixels, the camera-to-object distance and the focal length of the lens enables the calculation of the GSD (Equation 2.2 assuming square pixels).

$$
\begin{aligned}
s_{px} &= \sqrt{\frac{A_{sensor}}{n_{pixels}}} \\
\frac{s_{px}}{c_k} &= \frac{s_{PX}}{h} \\
s_{PX} &= \frac{s_{px}}{c_k} \cdot h
\end{aligned}
$$

$$
\begin{aligned}
s_{px} &\ldots \text{Pixel size on sensor} \\
s_{PX} &\ldots \text{Pixel size on ground, GSD} \\
A_{sensor} &\ldots \text{Sensor area} \\
n_{pixels} &\ldots \text{Number of pixels on sensor} \\
c_k &\ldots \text{Principal distance, focal length} \\
h &\ldots \text{Distance to object, flight altitude}
\end{aligned}
\tag{2.2}
$$

The smaller the ground area represented by a pixel, the more accurate the geometric representation of an object and it is possible to visualise textures more accurate. Even with smaller pixels, an area may consist of more than one material but might be recorded as one value within the pixel. This representation of multiple materials in one pixel is called spectral mixing (Keshava and Mustard, 2002). This effect often happens at boundaries between different materials and can cause a blurry boundary between objects, materials and colours. In Figure 2.1(b) the boundaries of the letters are blurred because the black of the letter is mixed with the white of the background which results in a grey

colour. Furthermore, the geometric resolution is also influenced by the quality of the lens, refraction, diffraction, focusing of the lens and other physical effects influencing electromagnetic radiation.

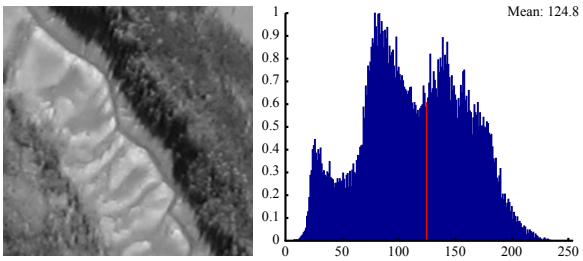
2.1.1.3 Image processing

Photographers often use software like *Photoshop*, *GIMP* or others to modify their images and to make them more appealing. Techniques used for this are collected under the term *image processing*. An image can be seen as a 2D recording of a signal (Burger and Burge, 2008). Hence, many techniques used in image processing are also used in other signal processing applications and often extended from one dimensional (1D) to 2D signals. The techniques can be classified as histogram, filter and transformation operations or other advanced algorithms (Burger and Burge, 2008). To apply these techniques, images are often separated into separate channels so that the algorithms are applied sequentially to each channel. After such operations the channels are normally combined together to recreate a multi-channel output image.

Histogram operations Histogram operations involve improving visualisation of digital images, particularly by increasing/decreasing contrast, enhancing the illumination, changing the saturation or even changing the colour (Burger and Burge, 2008). A change of colour requires a multi-channel image as an input. Typically, the hue channel is manipulated after conversion of an image to either HSL, HSV or HSI. Assuming that all colours are represented in a range from 0° - 360° (Figure 2.3(c)), a shift by 60° results in a shift of one colour. That means: green becomes yellow, yellow → red, red → magenta, magenta → blue, blue → cyan, cyan → green (Figure 2.4(b)). Change in illumination of an image involves an increase or decrease of each grey value. Instead of modifying each pixel of a multi-channel image, it is possible to modify luminance, value or intensity of an HSL, HSV or HSI converted image. The modified image looks brighter or darker than the original image (Figure 2.4(c) and 2.4(e)). In a histogram this change can be recognised by an increase of all grey values and subsequently the grey value average (Figure 2.4(d) and 2.4(f)) (Burger and Burge, 2008). Another widely used technique is contrast enhancement. This distributes the recorded intensity values over

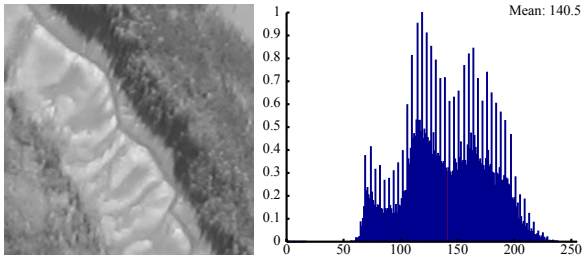
(a) Original unmodified image.

(b) Colour shifted image by 60°.



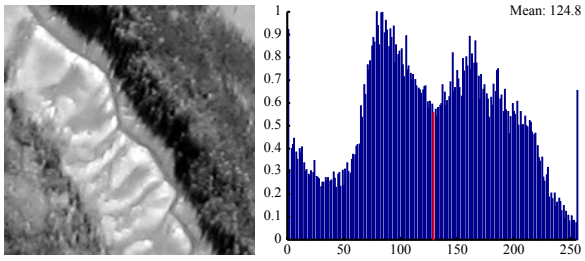
(c) Image in one channel grey scale.

(d) Histogram of grey scale image.



(e) Image with increased brightness.

(f) Histogram of increased brightness.



(g) Image with increased contrast.

(h) Histogram of increased contrast.

Figure 2.4: Histogram operations on an image.

the available range of grey values (Figure 2.4(g) and 2.4(h)) (Burger and Burge, 2008). This does not influence the average grey value, but returns an image which includes black and white pixels rather than dark or light grey.

There is a range of other techniques for modifying the histogram such as inverting, thresholding, noising or others, which involve a range of mathematical modifications to the histogram.

Filter operations The second category of image processing techniques involves the use of image filters, which are often used to accentuate or eliminate image defects (Burger and Burge, 2008). These operate on individual pixels and influence only the pixels in the filter region. There are several filters that concentrate on different kind of operations. A filter is defined by their size, operation, strength, and anchor pixel (OpenCV Dev Team, 2014). A 2D filter can be represented by a matrix. The filter matrix is moved over each pixel of the image matrix. Calculations are then performed at each position the filter matrix takes. The calculations are performed according to the operation and the returned value is written at the position occupied by the anchor.

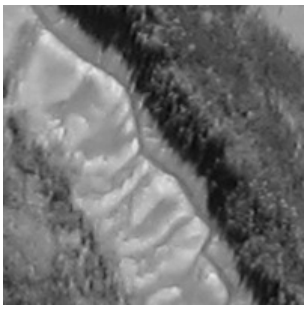
A class of popular filters are blur filters, which aim to 'smooth' an image. These filters are also called convolution filters because they convolve the signal frequency. Examples are the median filter (Figure 2.5(a)), or the Gaussian Filter (Figure 2.5(b)). The latter is based on a Gaussian normal distribution. These filters can not only be applied on 1D signals, but also on 2D signals like images. Each pixel within the area of the filter is multiplied with the value given by the filter matrix and the result written in the position of the anchor (OpenCV Dev Team, 2014). The result is a blurred image (Figure 2.5(d) and 2.5(e)). An application for blur filters is to reduce noise (Figure 2.5(f) and 2.5(g)) or to smooth areas in an image, which can be used to smooth human skin in an image (Adobe Systems Incorporated, 2015).

$$\frac{1}{13}\begin{bmatrix} & & 1 & & \\ & 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & \\ & & 1 & & \end{bmatrix}$$

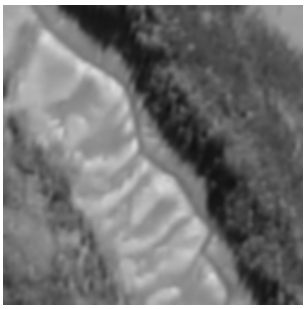
(a) Median filter. Anchor in grey.

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

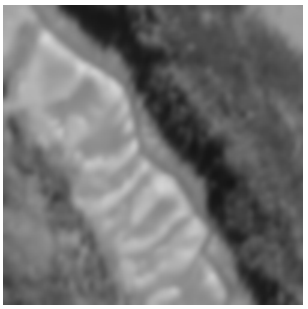
(b) Gaussian filter with $\sigma = 1$. Anchor in grey.



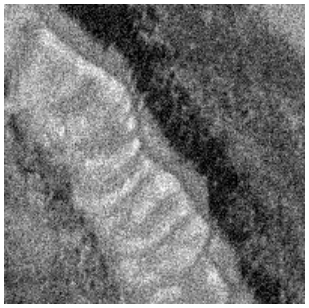
(c) Original unmodified image.



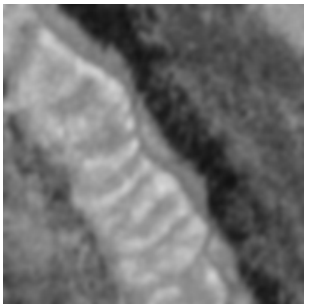
(d) Median filtered image with a 5 x 5 filter matrix and anchor in the centre.



(e) Gaussian filtered image with a 5 x 5 filter matrix and anchor in the centre.



(f) Original image after adding noise (20 dB).



(g) Applying 5 x 5 Gaussian blur filter on noisy image.

Figure 2.5: Blur filter operations on an image.

Due to the mathematical operation involved, these filters are called convolution kernel or convolution filter (OpenCV Dev Team, 2014). A convolution is a mathematical operation ($\otimes$) using a convolution function (g) on an initial function (f) returning the convolved function (k) (Equation 2.3).

$$f \otimes g = k$$

$$\begin{aligned} f &\ldots \text{Function to be convolved.} \\ g &\ldots \text{Convolution function.} \\ k &\ldots \text{Convolved function.} \\ \otimes &\ldots \text{Convolution operation.} \end{aligned} \tag{2.3}$$

For the 2D case the convolution can be written as in Equation 2.4 after Bradski and Kaehler (2008).

$$K_{x,y} = \sum_{i=0}^{G_{rows}-1} \sum_{j=0}^{G_{cols}-1} I_{x+i-G^a_{x,y},y+j-G^a_{x,y}} \otimes G_{i,j}$$

$$\begin{aligned} K &\ldots \text{Convolved image.} \\ I &\ldots \text{Input image to be convolved.} \\ G &\ldots \text{Convolution filter.} \\ G_{rows,cols} &\ldots \text{Size of convolution filter in rows and columns.} \\ G^a_{x,y} &\ldots \text{Position of anchor in convolution filter.} \\ \otimes &\ldots \text{Convolution operation.} \end{aligned} \tag{2.4}$$

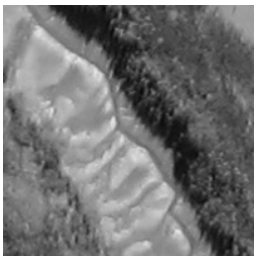
In the case of blurring, the image (F) is convolved ($\otimes$) by motion (G). The result function is the blurred (convolved) image (K).

The process of deconvolution is the reverse reconstruction of the initial non convolved image F, which is an ill posed problem due to various factors. At first, the convolution function and convolution filter can be unknown. However, the main problem is that the equation has more unknown variables u than knowns k ($k < u$) (Niemeier, 2007).

Therefore, deconvolution has to be carried out iteratively to achieve a deconvolved result (Section 5.1).

Beside blur filters, edge filters are also classified as convolution filters. They are used to find gradients in the image, which typically represent edges in the object creating a line of different grey values. In contrast to blur filters, which are low-pass filters, edge detection is a high-pass filtering process. Popular edge detection operators include the Sobel operator (OpenCV Dev Team, 2014), which can separate between horizontal (Figure 2.6(b) and 2.6(c)) and vertical gradients (Figure 2.6(d) and 2.6(e)), or the Laplace operator (OpenCV Dev Team, 2014), which combines horizontal and vertical gradient detection (Figure 2.6(f) and 2.6(g)). Edge detection is often a first step used to identify objects, structures or differences in images (OpenCV Dev Team, 2014).

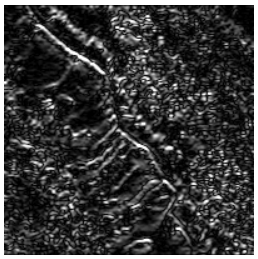
Another kind of filter includes morphological operations. These are developed for binary images, which just contain black and white values (2 bit). The other difference to the previously mentioned filters is that they do not conduct a calculation, but effectively 'scan' an area of neighbouring pixels. An erosion filter examines or scans pixels to assess whether the area covered by the filter matrix contains a zero (=black) value. If this is true, then the anchor position is defined as zero. Otherwise, a one (=white) is assigned (OpenCV Dev Team, 2014). In this way, structures are eroded or if they are small, completely eliminated (Figure 2.7(b)). The inverse process is known as dilate filter, which scans for ones. If there is a white pixel detected in the filter area, then the anchor is defined as one, otherwise it is set to zero. Structures in the image become larger using this method (Figure 2.7(c)). It is also possible to combine both operations. These combinations are known as 'opening' and 'closing' (OpenCV Dev Team, 2014). An opening operation is achieved by applying an erosion filter and then a dilation filter. The result is that small structures and noise is eliminated by the erosion filter. Larger structures which became smaller due to the erosion are restored to their original size by applying the dilation (Figure 2.7(d)). This combination is often used to delete small structures and to eliminate noise in binary images. A closing operation applies the dilation filter first and then the erosion filter. The result is that objects close to one another are connected and small structures are not deleted (Figure 2.7(e)). However,



(a) Original image.

-1	-2	-1
1	2	1

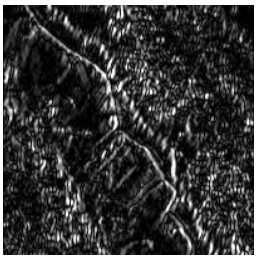
(b) Horizontal Sobel operator. Anchor in grey.



(c) Normalised image after applying horizontal Sobel operation.

-1		1
-2		2
-1		1

(d) Vertical Sobel operator. Anchor in grey.



(e) Normalised image after applying vertical Sobel operation.

	1	
1	-4	1
	1	

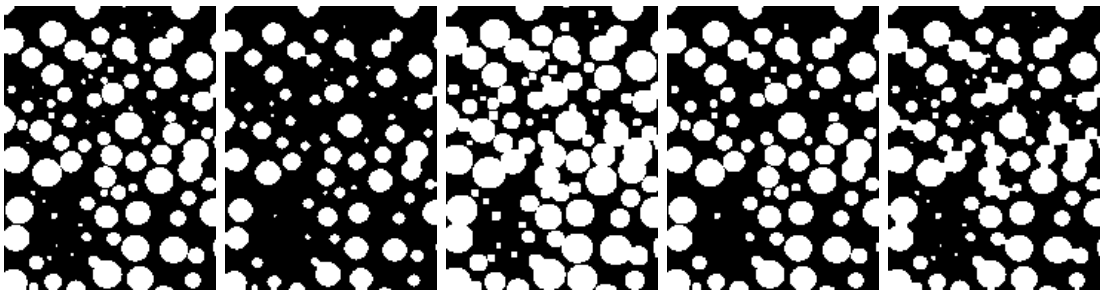
(f) Laplace operator. Anchor in grey.

(g) Normalised image after applying Laplace operation.

Figure 2.6: Different edge detection methods applied on an image.

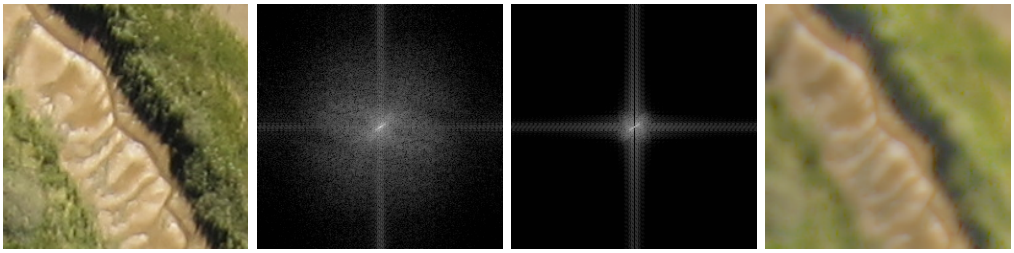
these filters have been developed further allowing them to be applied on grey scale and multi-channel images (The GIMP Team, 2015).

Transformation operations It is not only possible to modify the content of images, but also to transform images to a different kind of visualisation. This provides an effective way to identify or modify the content, or to change the image shape. The simplest transformations involve the colour spaces, or the bit depth of an image. For example, it is possible to change an 8 bit image with 256 grey values to a 2 bit image equivalent to reduce the information to black and white.



(a) Original binary image. (b) Erosion applied on binary image. (c) Dilation applied on binary image. (d) Opening applied on binary image. (e) Closing applied on binary image.

Figure 2.7: Morphological filter operations.



(a) Original image in spatial domain. (b) Fourier domain of original image. Fourier domains are mirror symetric. (c) Low pass filtered Fourier domain. (d) Filtered Fourier domain image transformed to spatial domain.

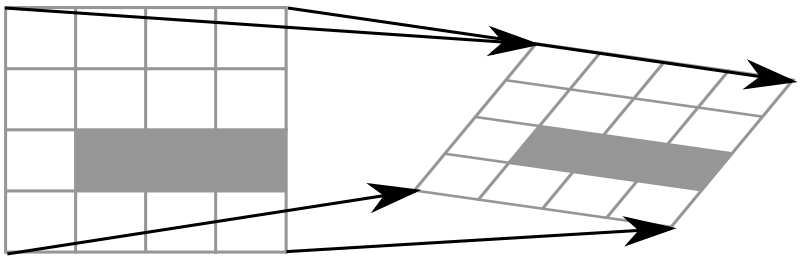
Figure 2.8: 2D Fourier domain transformation.

A very different approach is the Fourier Transformation. An image is considered a 2D, digital, electrical signal that contains frequencies, amplitudes and phase this allows Fourier transformations to be applied. By using the Fourier transformation it is possible to analyse which frequencies appear in the image. Therefore, the image is transferred from a spatial (Figure 2.8(a)) to the frequency domain (Figure 2.8(b)). Visualising the frequency domain results in a 'noisy' looking grey scale representation with low frequencies in the centre and high frequencies towards the outside. Now it is possible to filter and remove any specific frequency (Figure 2.8(c) and 2.8(d)) and when the frequency domain is reversed to the spatial domain the image is smoothed or repetitive patterns are eliminated (OpenCV Dev Team, 2014). There are several other

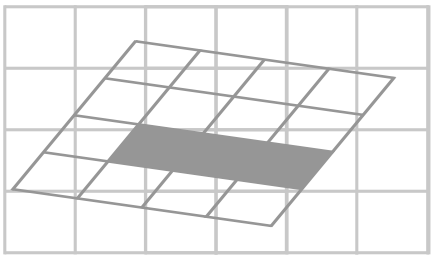
transformations with different purposes such as Hough transformation, which aims to identify lines in images (OpenCV Dev Team, 2014). More importantly for photogrammetry there are a range of geometric transformations that (un)distort the image (Section 2.1.1.3).

Simple image transformation and interpolation Image transformation is used to change the position of pixels in an image to fit them to a reference object. For example image transformation can be used to match two images taken from different positions. In photogrammetry the aim is to compensate for scale and rotation differences between images, as well as shear and barrel/pincushion distortions (OpenCV Dev Team, 2014; Singh, 2010). To achieve an image transformation it is necessary to know the function and the transformation parameters. This can be done by measuring identical points across both images to solve for the transformation parameters. With the transformation parameters determined the transformation can be conducted by repositioning each of the pixels of the frame (Figure 2.9(a)). The repositioning can cause the problem that the repositioned pixel does not exactly match a pixel of the new frame (Figure 2.9(b)). This problem has to be solved using interpolation. The nearest-neighbour interpolation takes the grey value from the nearest pixel, which can cause aliasing errors (steps in the image) (Figure 2.10(a)) (Sun Microsystems Incorporated, 1999). Another approach is the bilinear interpolation, which uses the values of the four closest pixels and interpolates the grey value for the pixel in the new frame (Sun Microsystems Incorporated, 1999). The result is a much smoother image with less aliasing (Figure 2.10(b)) (Sun Microsystems Incorporated, 1999). The bi-cubic interpolation is an approach using the 16 nearest pixels. Interpolation over 16 pixels introduces minimal aliasing. However, with increased numerical computations involving many pixels it is also the slowest method. Another problem of this method is the smoothing effect, which appears as optical blur (Figure 2.10(c)) (Sun Microsystems Incorporated, 1999).

Another method used to determine transformation parameters is feature matching. This method identifies similar features across both frames instead of manually measuring points in both images. This is nowadays based on Scale-Invariant Feature Transform (SIFT), Speed-Up Robust Features (SURF), or other similar algorithms. These algorithms detect suitable candidate features in the images, which are invariant to scaling, translation



(a) On the right the input image on the left the image after transformation.



(b) The transformed image (dark grey) does not exactly match the new image matrix (light grey).

Figure 2.9: Perspective transformation.

(a) Nearest neighbour method. (b) Bilinear interpolation. (c) Bicubic interpolation.

Figure 2.10: Perspective transformed image.

and rotation (Bay et al., 2006; Lowe, 2004). Similar features detected in two overlapping images can be referenced between them using least squares matching (LSM) or other similar methods (Brown, 1992).

If enough common features are identified between both frames, it is possible to calculate the transformation parameters and carry out image transformations.

Template matching A method often used in image processing and in photogrammetric image processing is template matching. It is the process of finding a defined shape with a certain pattern of grey values in an image. The template is a small image patch, which contains the shape defined by its grey values. This template is then moved across the sequentially whilst the algorithm detects the best match (Equation 2.5).

$$\Delta_{x,y} = \sum_{i=0}^{T_{rows}-1} \sum_{j=0}^{T_{cols}-1} \sqrt{F_{x+i,y+j} - T_{i,j}{}^2}$$

$$\text{best match} = min(\Delta)$$

$$\begin{aligned} \Delta_{x,y} &\ldots \text{Difference image.} \\ F &\ldots \text{Input image for matching.} \\ T &\ldots \text{Template to be matched.} \\ T_{rows,cols} &\ldots \text{Size of template in rows and columns.} \end{aligned} \tag{2.5}$$

There are also more advanced template matching techniques, which include not only parameters to represent shift but also scale, rotation and distortion of the template.

2.1.2 Photogrammetry

Photogrammetry is the science of reconstructing '*the position, orientation, shape and size of objects from pictures*', which is based on precise measurements of image coordinates (Kraus, 2007). Potentially, any image can be used for measurements in the object space, if they fulfil special requirements. It is possible to use images from a hand-held camera in a close range application, as well as images taken from an aircraft. To use these images for photogrammetry it is important to know the geometry of the camera to produce precise results.

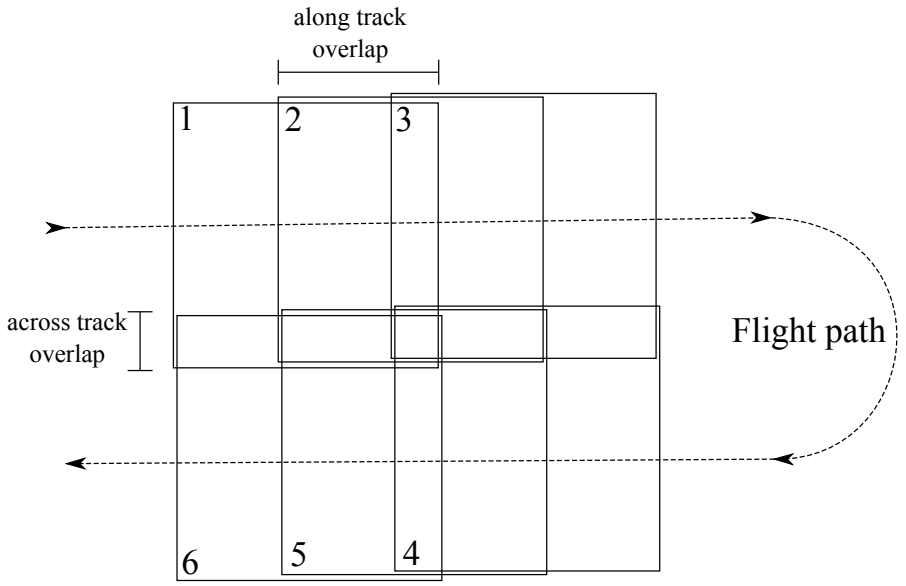


Figure 2.11: Aerial photogrammetry flight path.

2.1.2.1 Aerial photogrammetry

The possibility to measure 3D coordinates using aerial images has become the basis for aerial photogrammetry and mapping. Aerial photogrammetry based upon the stereo pair requires image overlap to have the necessary geometry for photogrammetric processing (Kraus, 2004). Flights are typically planned so that the images overlap 60 % along track and 25 % across track (Figure 2.11) (Kraus, 2004; Luhmann et al., 2014).

If the topography of the flight area is varied, or the operator is interested in measurements of heights, it is advised to choose larger overlaps (Kraus, 2004). The overlap provides the ability to detect identical points conjugate in both images and to reference them to each other. Conjugate points can be then measured in photo coordinates $(x'\, y')$ in both images. Photo coordinates are measured relative to the principal point (x'_0, y'_0) in the centre of the image. Image coordinates have their origin in the upper left image corner. The measurement of the photo coordinates allows 3D coordinates for a point in 3D object space (X, Y, Z) to be calculated. This requires application of the collinearity

equations (Equation 2.6) (Luhmann et al., 2014).

$$x' = x'_0 - c_k \frac{r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} + \Delta x'$$
$$y' = y'_0 - c_k \frac{r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} + \Delta y'$$

(x', y') ... Photo coordinates.
(x'_0, y'_0) ... Principal point offset.
$r_{i,j}$ i, j=1...3 ... Orthogonal rotation parameter.
$(\Delta x', \Delta y')$ ... Optional additional parameters
(X_0, Y_0, Z_0) ... Camera perspective centre
(X, Y, Z) ... 3D coordinate (2.6)

To fulfil the above equations, the exterior orientation of the camera needs to be known. The exterior orientation is defined by three rotations (ω, ϕ, κ) to contribute to the rotation matrix $(r_{i,j})$ and the camera perspective centre (X_0, Y_0, Z_0). The exterior orientation can be solved by direct georeferencing or the usage of ground control points (GCP) (Figure 2.12) (Bäumker et al., 1999). Direct georeferencing requires global navigation satellite systems (GNSS) and inertial measurement unit (IMU) measurements. Measurements of these, taken during image acquisition, are fed directly in the equation. If direct georeferencing is not possible, the use of GCP in marginal images and Height control point (HCP) chains through the pictured area provids the possibility to fulfil the collinearity equation (Kraus, 2007). This calculation requires tie points (TP) to be measured in overlapping image areas. With the GCP, HCP and TP the unknown exterior orientation parameters of the equation can be determined for each image using the bundle adjustment (Kraus, 2004). The Bundle adjustment is a least squares optimised procedure that distributes errors homogeneously across measurements acquired in all images, even in large datasets (Triggs et al., 2000) (Section 3.1.2.2).

Besides the exterior orientation, information about the camera itself can be determined, including parameters capable of modelling the internal geometry of the camera. The focal

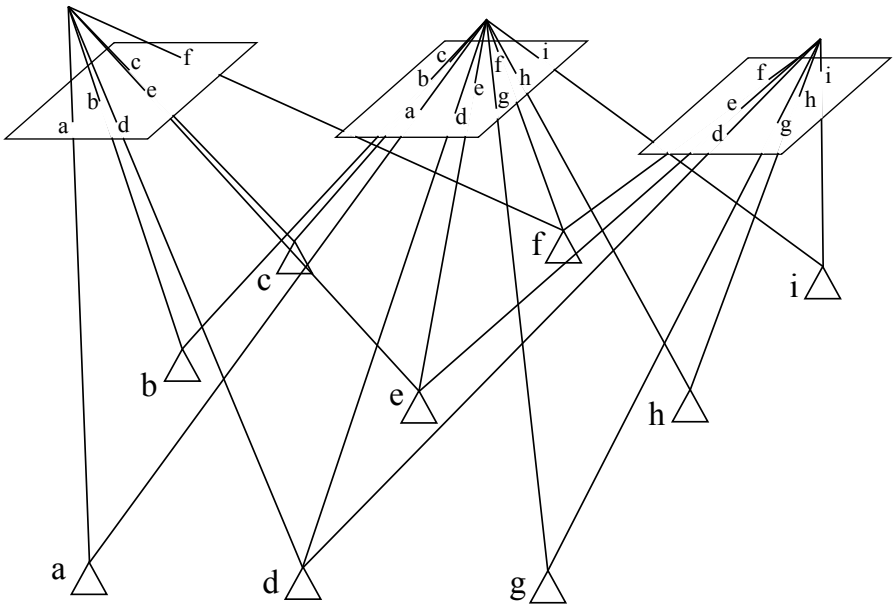


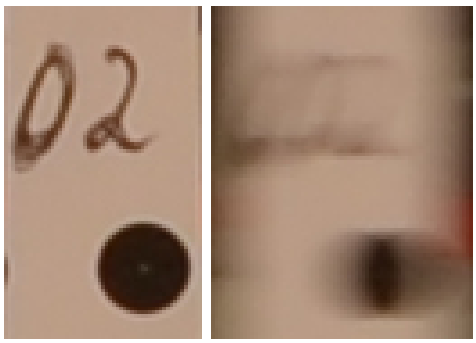
Figure 2.12: Ground control points.

length (c_k), position of the principle point (x'_0, y'_0) and optional additional parameter ($\Delta x', \Delta y'$) can be used to define the geometry of the camera. These parameters can be calculated in a camera calibration, using a photogrammetric test field, or if image geometry is strong, with an on-the-job calibration during post processing. The purpose is to determine the image distortion caused by the lens and the camera, to achieve accurate measurements and precise 3D object coordinates.

To calculate 3D coordinates, the exterior orientation and interior camera parameters are combined with 2D coordinate measurements made in the image. Measurements are more difficult, or even impossible if the image is blurred (Figure 2.13). However, blur is a well-known problem to aerial photogrammetry, traditionally solved by fast shutter speeds and motion compensation technology (Section 2.1.2.3).

2.1.2.2 UAVs

Unfortunately, UAVs rarely provide a stable camera position for accurate 2D image coordinate measurements. UAVs are affected by wind, turbulence, sudden inputs by the operator and also by the flight movement of the aircraft itself. A major advantage of



(a) Sharp target. Measurements possible without problems.

(b) Blurred target. Measurements difficult due to blur.

Figure 2.13: Sharp target vs. blurred targets.

UAVs is their use in high risk situations without endangering human life. These can also be used at low altitude or in flight profiles where manned systems cannot be flown (Eisenbeiß, 2009). Low cost sensors are practicable because of the limited payload, regulatory restrictions and vulnerability of the UAV platform (Eisenbeiß, 2011). The UAV system applicability for different situation is based on the flight range, endurance and manoeuvrability and flight path control (Eisenbeiß, 2009). Table 2.2 represents a range of different UAVs. A Kite can be considered to be a flexible wing UAV with a very limited range but can have a very high endurance, manoeuvrability and some degree of flight path control within a small area. In contrast, a Balloon has a long range and endurance but the manoeuvrability is limited to changes in altitude. Following a flight path is impossible because a balloon is completely dependent on the wind direction. The

Table 2.2: Classification of UAVs based on Eisenbeiß (2011).

	Lighter than air	Heavier than air			
		Flexible wing	Fixed Wing	Rotary Wing	Various
Unpowered	Balloon	Paraglider Kite	Hang glider Glider	Gyrocopter	Animals/Pets
Powered	Airship	Paraglider	Hang glider Propeller Jet engines	Single rotors Coaxial Quadrotors Multi-rotors	

range of potential applications for UAVs is almost unlimited. It is possible to conduct a range of typical land surveying tasks (Peterman and Mesarič, 2012), carry out SAR operations (Molina et al., 2012) or aerial photogrammetry applications including 3D modelling, DSM generation (Remondino et al., 2012) and many more (Colomina and Molina, 2014). It is also possible to carry out light detection and ranging (LIDAR) (Zhou et al., 2012).

UAV image flights are different to typical aerial photogrammetric surveys (PIEneering, 2012). An important difference is the much lower flight altitude, which is cause by regulatory restrictions. However, the lowflight altitude enables much higher ground resolution and allowes the imagin of onlique facades and structures. Another difference is the much larger image overlap used during UAV flights, typically 80 % along track and 60 % across track (PIEneering, 2012). The large overlap enables a creation of more accurate orthophotos because the 'fall over' effect of high stuructures such as buildings can be compensated by cosing a different image. The larger overlap also allowes up to 15 image coordinate meassurements for each object point. This vast amount of measurements makes it possible that coordinates can be determined more accurate and that height point chains are not required (PIEneering, 2012).

Due to its lighter weight and more unstable flight characteristics, images acquired by a UAV are often blurred, when compared to a conventional aircraft. Light weight UAVs are particularly affected by wind and turbulence. Automatic detection of image points (GCP, HCP, TP) for the bundle adjustment is difficult and prone to errors in blurred images. The manual detection of blurred images by human operators is time intensive and exhausting for the eyes. Consequently, blur should be prevented and blurred images should be identified and isolated using an automatic algorithm.

2.1.2.3 Blur prevention and preventive hardware modifications

A constraint enforced on early photographs was a stable camera position and a stationary object, which forms the basic requirement for sharp images. The first images obtained required exposure times of many days (Maison Nicéphore Niépce, 2013). Today professional photographers use a tripod and an external shutter release to prevent

the camera from any movement during image acquisition, e.g. camera calibration in photogrammetry. With the development of compact cameras it has become very popular to acquire images 'on the go', which introduced blur in images due to the hand-held use without any stabilisation. This is why there are many different approaches in hardware development and computer vision to prevent degradation of image quality and detect and to deblur blurred images. Blur and sharpness are contrary and together with image brightness, saturation and colour an important factor for image quality.

Blur and convolution filtering The overall quality of the optical system can be described with the modulation transfer function (MTF). The MTF describes in detail the ratio of the contrast at an object corner to the contrast of this corner in an image (Nasse, 2008). The larger the ratio of contrast, the better the quality of the optical system. It is well known that lenses often exhibit a better sharpness in the centre of the lens and get less sharp towards the boundaries (Nasse, 2008). This effect is commonly known as "lens fall of" (Nasse, 2008). For this thesis the MTF was calculated for the used lens (Figure 2.14) to established the quality. It was found that the lens used has in average 0.79 pixel chromatic aberration and 1300 lines per image height in the centre of the image ($n = 10$). Towards the boundary of the lens the chromatic aberration increases to 3 pixels and lines per image height decreased to 1100 ($n = 10$). This shows that the used lens is of acceptable quality and that the lens fall of is within the expected boundaries for a standard DSRL lens. However, using MTF to measure motion blur would not be possible, because the MTF is just measured along one edge. If this edge is in direction of motion blur the MTF would be the same as for a sharp image. Only edges perpendicular to the motion would exhibit a lower quality MTF. The MTF is also dependent on aperture settings of the lens and the filmspeed (ISO) controlled by the camera. The MTF describes only the 'micro contrast', which represents how accurate small structures are represented (Nasse, 2008). However, the image quality is also influenced by the 'macro contrast', which depends on the veiling glare of the lens. This veiling glare are light rays, which are reflected on the glass surfaces inside the lens and deviate from where they should originally illuminate the image sensor. The result is that dark areas are 'brightened up' by additional light rays and do not appear as dark as they are in reality (Nasse, 2008).

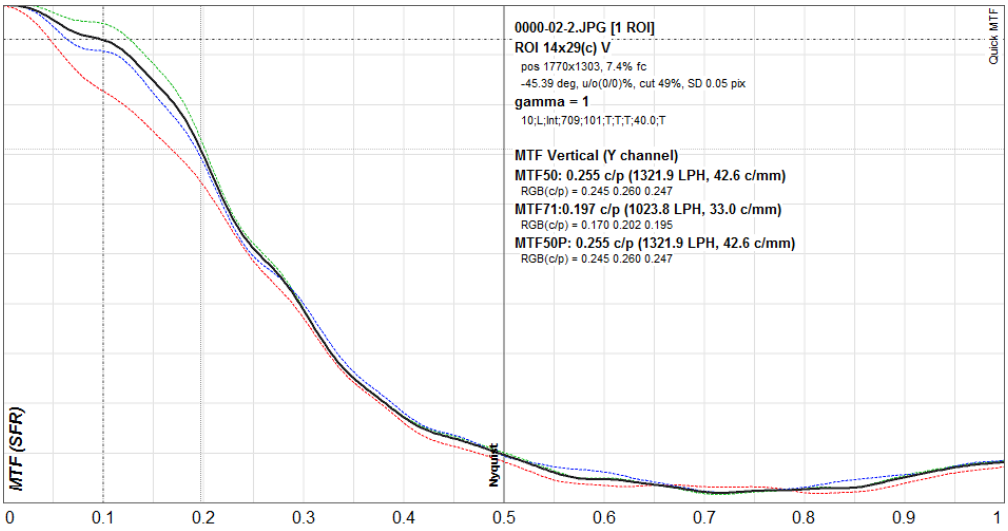


Figure 2.14: Modulation transfer function for the used optical system. MTF was measured using QuickMTF (2015)

Additinally, to the degrading effect introduces by the optical system, image quality is influenced by '*blur*'. There are two categories of blur. The first category of blur can be a result of optical distortion caused by the focusing of the lens and setting of the aperture, which is called optical blur. Another type of blur is caused by motion of the camera with respect to the target or the object with respect to the camera, which is called motion blur, even with the correct focal setting and aperture. The latter is the focus of this research, for the remainder of this thesis it will be referred to simply as 'blur' (Section 2.2).

The apperance of motion blur depends also on the shutter used in the camera. There are two major concepts the global shutter and the rolling shutter (www.red.com, 2015). A global shutter illuminates all pixels of the image sensor at the same time. If the camera is moved during exposure of the sensor, all pixels on the sensor experience the same motion blur. With a rolling shutter however, the pixel lines are exposed one by one. If the camera is moved between the exposure of each line, objects will be displaced compared to the previous pixel line and movement will affect each line differently (www.red.com, 2015).

Blur prevention Image quality is affected by a range of factors. Both the camera, with its sensor and settings and the lens are important. However, this has to be decided

prior to an image flight and should be considered in an appropriate way. In stable flight a fixed wing UAV has a constant velocity. This causes a quantifiable motion blur in every image. Unfortunately, UAV flights are rarely stable and influenced by winds and turbulence or sudden inputs by the operator. To decrease the amount of post processing on the images it is possible to prevent image blur in advance. A technique used in aerial photogrammetry is 'angular motion compensation' (AMC) and 'forward motion compensation' (FMC) (Pacey and Fricker, 2005). AMC is a system, which compensates the undesired angular motion of an aeroplane. This is achieved by mounting the camera body on a gyroscopically controlled platform. FMC is a system that moves the film, sensor, lens or camera in the direction of travel during image acquisition to compensate for blur effects. To achieve this, the altitude and the ground speed needs to be precisely known, to compensate exactly for platform movement. This information can be supplied by precise GNSS and IMU measurements. It is also possible to use the GNSS and IMU data to adjust the exposure time of the images to ensure it is shorter or equal to the GSD. Physical movement of the sensor is not ideal because it is technically difficult to realise and will be prone to errors at low altitudes. Changing the exposure times will also cause problems because images will have different brightness, which influence post processing of datasets. Another method used in digital cameras is 'time delay integration' (TDI). TDI is a digital movement of the pixel content to the following row of the sensor according to the ground speed (Pacey and Fricker, 2005). It is the flow of the recorded reflectance during exposure from one pixel to the following pixel in the direction of travel, which is the digital equivalent to the physical movement of a camera film in FMC.

Another approach is vibration reduction or image stabilisation, which is available in many low cost, compact and DSLR cameras because such high resolution sensors are increasingly prone to blur from shaky hands (Sachs et al., 2006). There are two approaches for image stabilisation. One is optical image stabilisation, which mechanically moves the image sensor or the lens. This method requires gyroscopes, which measure the rotation of the camera and controls the mechanical adjustment of the sensor or lens. The second method is electronic image stabilisation within the sensor itself. This method is preferred for video cameras because optical image stabilisation causes frame-to-frame jitter (Sachs et al., 2006). It is also possible to use a gyroscope to detect the movement of the camera

body and to adjust the image itself. No mechanical modifications are performed, but the image content is simply shifted by a certain number of pixels using software, which is also used in the case of time delayed integration (Sachs et al., 2006). An alternative approach involves tracking a small set of feature points. These can be tracked over several frames and interframe motion can be estimated. By connecting all single interframe motion estimations, it is possible to compute the total motion of the camera over time and warp the images back to the first frame (Morimoto and Chellappa, 1996).

Hardware modification Other approaches involve modifying the hardware in different ways to simplify blur detection or correction after image acquisition. The most common way involves a second image sensor. The first sensor takes images with a high GSD, but includes blur caused by movement. The second sensor is attached to the same platform as the first, hence experiences the same movement. However, it does not take high resolution images but images with a high temporal resolution, so allowing the motion of the platform to be calculated (Ben-Ezra and Nayar, 2004). With the calculated motion path it is possible to use a Richardson-Lucy deconvolution to deblur the image (Section 5.1) (Ben-Ezra and Nayar, 2004; Burger and Burge, 2008). The disadvantage is the need for a second camera with a very high frame rate. A typical exposure time for image flights is less than $\frac{1}{100}$ s which requires a frame rate of larger than 200 frames per second (fps). This is not achievable with conventional video cameras, and special video cameras are required. Another problem is the displacement of the optical path between high frame rate and high resolution camera, which causes a different or tilted point of view (POV).

Two other modifications affect the camera itself. The method used by Lelégard et al. (2012) modifies the exposure time for each of the RGB channels. The result is that neither of the channels is blurred or that all channels are blurred differently. The channel with the shortest exposure is least blurred. The blur difference between the channels can be used for the deblurring process (Lelégard et al., 2012). How much the channels are blurred can be examined by converting the channels from the spatial domain to the frequency domain with a discreet Fourier transformation (DFT) and comparing the phase differences.Dependent on the amount of blur of the least blurred channel, it is

possible to use the DFT filtered image of the least blurred channel to improve the other more blurred channels (Lelégard et al., 2012).

An alternative approach involves coding the image exposure using a fluttering shutter (Raskar et al., 2006). The fluttering can be chosen in a way that the resultant point spread function (PSF) has the maximum coverage in the Fourier domain. A PSF is a convolution function (Section 5.1.1.1). The deblurring can be achieved using a linear approach. The problem is that the camera movement, has to be known in advance to choose the fluttering of the shutter (Raskar et al., 2006).

Another approach involving hardware modification, is the attachment of an IMU to the camera body (Joshi et al., 2010; Shah and Schickler, 2012). This is an attractive approach because many UAV systems already contain an IMU in their autopilot system to support the pilot. However, based on the Nyquist-Shannon sampling theorem the IMU needs to have a high measuring frequency of larger $2 \cdot \frac{1}{\text{exposure time}}$ (Jerri, 1977). This equates to more than 200 Hz and allows the IMU to record the angular movement of the camera during image exposure. These measurements can be used for image deblurring by deconvolving the pixels and recovering a sharp image (Section 5.1). There are problems related to the sensor accuracy and the noise of the IMU. Flight altitudes with a larger distance between camera and object pose additional challenges (Joshi et al., 2010).

Most of these modifications are only applicable for minor blur (e.g. TDI). Other methods would return different spectral signatures of materials and hinder further automatic processing (Schowengerdt, 2007). Finally, the modification of low cost cameras involves additional equipment, which exceeds the payload and size of small scale UAVs used for most scientific applications. Most methods described are too expensive for small scale applications, because they require very precise GNSS receivers and/or IMUs and/or additional or more expensive cameras with special modifications. This would therefore erode the key benefit of a UAV system, which is low cost. This makes blur prevention using hardware modification more difficult for UAVs and puts the focus on blur detection (Section 4) and correction (Section 5), after image acquisition.

(a) Motion blur do to moving object in the image. (b) Motion blur due to moving camera.

Figure 2.15: Different kind of motion blur.

2.2 Generating motion blurred images

As the literature review revealed there are two different kind of blur, optical blur and motion blur. Motion blur can be subdivided in to two groups. The first kind of motion blur is caused by the object moving during image acquisition. This creates a blurred feature whilst the background of the image remains sharp (Figure 2.15(a)). The extent and shape of blur depends on how fast and on which path the object moved. Also the camera to object distance influences the shape of the blur. A larger camera to object distance lets the object appear less blurred, even when motion speed is the same as for a shorter camera to object distance.

The second cause of motion blur is camera motion during image acquisition, in this case the entire image will be blurred. However the extent of blur still depends on the camera to object distance. An area with a large camera to object distance will be less blurred than areas closer to the camera. One special case occurs, if the camera moves with an object. Then the object is sharp, even when it is moving, while the background will be blurred (Figure 2.15(b)). This second kind of motion blur is often used in sports to show that the object of interest is fast moving. The first kind of motion blur is often used when there is no special object of interest but the photo is supposed to express liveliness and motion. Both cases are used as style elements in photography (Loughborough University, 2011).

2.2.1 Problem of current blur methods

UAV images can be affected by the second type of (motion) blur due to camera movement during image acquisition. As the camera is attached to the moving platform the complete image is blurred, not only an object in the image. Simulating blur has been done before by physically shaking the camera during exposure but mainly digitally, by using mathematical convolution filters like a Gaussian or media filter (Joshi et al., 2010; Sheikh et al., 2006) (Section 2.1.1.3).

Blur in UAV imagery is caused by a moving camera and represents a problem which is currently unpreventable. To simulate motion blur in a controlled environment requires a full understanding of UAV motion blur. The acquisition of a UAV image sequence often contains one directional motion blur, due to the forward motion of the aircraft. The amount of motion blur is dependent on UAV ground speed, which is not necessarily UAV velocity through the air, due to localised wind effects caused by turbulence, gusts and operator inputs which rotate the aircraft. In addition to the velocity of the UAV, the rotations along the roll or pitch axis of a UAV are manifest as linear motion blur also. Rotations along the pitch axis occur when the aircraft begins to descend or climb. This motion is in the same direction as forward motion and would add to the forward motion blur. Rotations along the roll axis occur when the aircraft begins to turn. This motion causes across track blur which is orthogonal to the forward motion blur. The combined effect is a diagonal motion blur. Rotations along the yaw axis are more complicated as they create a circular blur in the image. Added together with forward motion, it would be described as a curve. However, due to short exposure times, most blur paths are short and can be approximated by a line. The combined effect is that all typical movements of a UAV can be simplified into linear motion blur and not in complicated motion paths. Hence, only linear motion blur needs to be simulated in test images.

Generating motion blur in images can be done using various methods. One method could involve mathematical blur filters, which use mathematical convolution to simulate motion blur. However, it was found that mathematical blur is not the same as geometric blur caused by camera shake (Section 4.3). The second way of inducing motion blur in images is by hand-held shaking of the camera. This approach is problematic because

the extent and the direction of the induced blur is unknown. Additionally, the motion speed might vary during image acquisition. Fortunately, because of the high sensitivity of modern sensors, exposure times are short and motion blur can be assumed almost constant. A third way to generate images with known motion blur is to connect a precise IMU to the camera (Joshi et al., 2010). This IMU could measure the camera displacement, direction and velocity. However, using an IMU still requires a platform that can be shaken in a way similar to the motion described by a UAV. It also requires a high precision IMU with high measuring frequency to measure the displacement for images acquired with fast shutter speeds. For an image with an 1/200 s exposure time, the measuring frequency has to be at least 400 Hz. An alternative approach is to fix the camera on a platform which exhibits known motion. A platform capable of creating such physical displacements is called a shake table and was available for this study.

2.2.2 Shake table

Instead of using hand-held or mathematical image blur a 'long stroke shaker with linear bearings' (shake table) was used to acquire motion blurred images (APS Dynamics Incorporated, 2015). Images created using a shake table contain motion blur, which has a known motion path and amplitude. As the camera is physically moved, such images also contain typical degrading effects introduced by the camera hardware, such as noise and chromatic aberration.

The '*APS 400 ELECTRO-SEIS*' of the company *APS* can be shaken in one direction (1D) up to 158 mm. A table platform of approximately 35 cm by 35 cm was mounted on top of the shake generator. The shaking can be monitored with acceleration sensors and a laser displacement sensor (Figure 2.16).

The device is classically used by structural and mechanical engineers who place models of constructions on the table platform. The force generator which moves the platform can then be used to shake the models. By observing the model the dynamic response characteristics can be studied, which is particularly important for construction in areas susceptible to earthquake, for railway sleepers materials or other materials influenced by any kind of vibration. Therefore, the table is normally jiggled with low deflection and

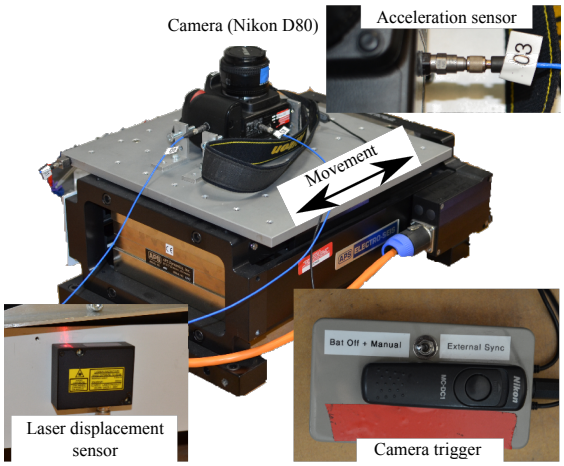
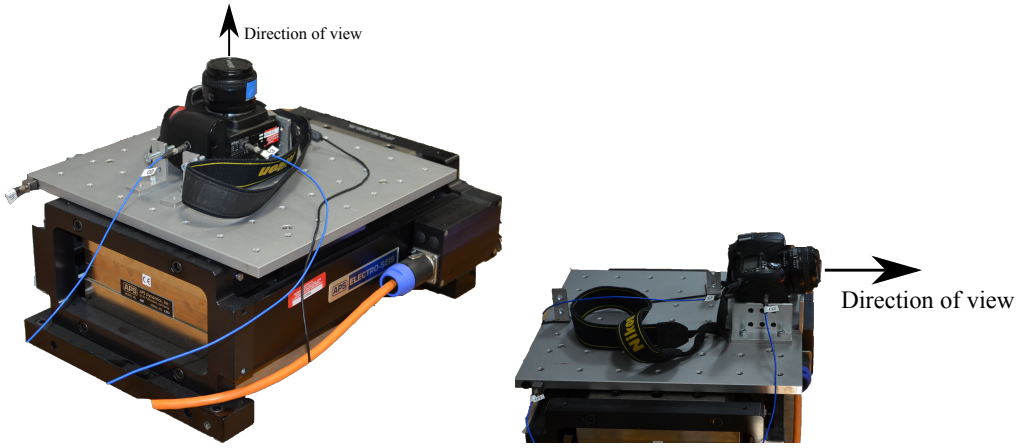


Figure 2.16: APS 400 ELECTRO-SEIS with sensors.



(a) Camera facing the ceiling. (b) Camera facing the wall.

Figure 2.17: Shake table setup.

high frequency. To simulate motion blur similar to UAVs the table needs to be shaken continuously in one direction and not 'jiggled' with high frequencies.

This project used the shake table extensively to generate images with known blur characteristics. To simulate motion blur, a camera was mounted on the table and fixed so that it moved with the table (Figure 2.17). To conduct different tests the camera could be mounted to face the ceiling (Figure 2.17(a)), where there was a test field composed of photogrammetric targets, or towards the wall where 3D objects were placed (Figure 2.17(b)). The camera has to be fixed very tightly, so that the camera

moves coincidentally with the table. The orientation of the camera was set up in a way so that the blur appears along either image columns or rows, but not diagonally. To measure the actual position of the table platform a laser displacement sensor is attached to the table (Figure 2.16). Additionally, an acceleration sensor was used to measure the acceleration of the table and two other acceleration sensors were fixed on the camera. One sensor measured the acceleration in the direction of the movement, a second sensor measured perpendicular to the table movement. Two seperate accelerometers were used to increase redundency. The sensors attached to the camera were used to demonstrate that the camera only moved in one direction (Figure 2.16).

The movement of the table was controlled by *Matlab* using an input function created to describe the velocity of the table. This function was linearly dependent on the time ($vel(t)$). Logically the first derivation with respect to t represents acceleration; the first integration describes displacement. The velocity was described by a harmonic oscillation, which ensures a smooth transition between different phases of acceleration (Equation 2.7).

$$\begin{aligned} vel(t) &= Amp \cdot \sin(2\pi fqu \cdot t) \\ a(t) &= \frac{\mathrm{d} \cdot vel(t)}{\mathrm{d}t} \\ d(t) &= \int vel(t)\,\mathrm{d}t \end{aligned}$$

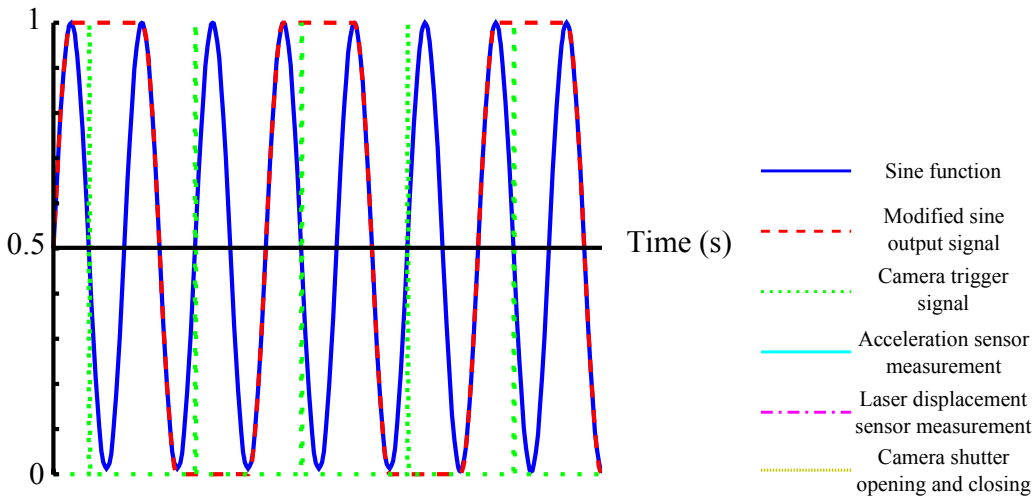
$$\begin{aligned} t &\ldots \text{Time} \\ Amp &\ldots \text{Amplitude} \\ fqu &\ldots \text{Frequency} \\ vel &\ldots \text{Velocity} \\ a &\ldots \text{Acceleration} \\ d &\ldots \text{Displacement} \end{aligned} \tag{2.7}$$

A harmonic oscillation is not constant at any point which caused the problem that the velocity described did not contain a period of constant velocity. However, a period of

constant velocity is demanded to simulate motion blur similar to blur created by UAVs. This required a modification of the function to create periods of constant velocity. The function needed to be modified in a way that the table was kept at a constant velocity for the period of one oscillation. This ensured that the table was kept at a constant velocity and neither decelerated nor accelerated (Figure 2.21). Synchronous with the period of constant velocity the camera needed to be triggered with a remote shutter, allowing an image with constant motion blur to be generated. The triggering of the camera was delayed until constant velocity was reached to make sure the table was no longer accelerating. After image acquisition the period of constant velocity was kept for a short period of time to ensure that the table was not decelerating during image acquisition. After the period of constant velocity the table was decelerated and then accelerated in the opposite direction until the table reached maximum speed again, which was kept for the period of one oscillation.

The shake table signal was modified in various ways. The number of periods of constant velocity defined how many images were taken in one shake table run. For one run the velocity at which the camera was moved during image acquisitions was similar so that images with similar motion blur were created. The periods of constant velocity were set to 10, to ensure enough test images with similar motion blur. Another modification of the signal was the frequency. A high frequency of oscillations had the consequence that one run of 10 oscillations was accomplished faster, but also that the periods of constant velocity were shorter. With shorter periods of constant velocity the exposure time had to be adjusted to guarantee that images were only acquired in the time of constant velocity. Long exposure times therefore required lower frequencies to create larger periods of constant velocity. If the frequency was chosen too low or the table moved too far in one direction, then the table platform can reach the maximum displacement possible, which results in an abrupt halt of the platform. This should be avoided to prevent damage to the table and because it would result in unknown camera displacements, making subsequent calculation of motion blur impossible.

Another element that was modified at the shaking function was the amplitude of the signal, as this defines the table velocity. The amplitude could be first defined in the *Matlab* script, but as the signal was sent via an external amplifier, the amplitude had



(a) Output signal to shake table and camera.

Figure 2.18: Shake table control signal (normalised).

to be adjusted. In a first run the amplitude should be set to 1 in the script and then the amplifier gradually increased in a way that the table can be displaced by the full extent of 158 mm. However, the amplitude must not exceed the maximum displacement of the table as this would again result in an abrupt halt of the table platform. For the following runs the amplitude was then be chosen in the *Matlab* script as $A \in [0, 1)$.

Additionally to the table output signal the camera had to be triggered at periods of constant velocity. The trigger signal was generated as a binary signal which only contained an impulse to trigger the camera shutter (Figure 2.21).

2.2.3 Camera and camera settings

The camera type, camera settings and lens had a significant influence on the outcome of the shake table experiments. The camera should be equipped with a remote shutter, necessary to release the shutter at the correct point of time during the shake table cycle. A remote shutter also prevented movement of the camera caused by pressing the shutter button manually. It had to be possible to manually adjust standard camera settings like exposure time, aperture and film speed and also more advanced settings like shake reduction. Also the number of pixels acquired by the camera was important as it directly

influenced the GSD. The ground resolution (Section 2.1.1.2) was also controlled by the sensor size and the lens. The lens had to be of good quality to prevent blur effects caused by spheric aberration, chromatic aberration, coma astigmatism and other influences. The focal length of the lens was chosen based on the desire of covered area and the desired accuracy of objects in the image. Furthermore, the GSD was also influenced by the distance between camera and object. The image resolution was important because the larger the zoom and the more accurate an object was represented in an image, the larger was the influence of camera movement. It was decided that a 24 mm and a 28 mm lens were appropriate for the planned experiment. This decision was based upon availability, adequate camera specifications, appropriate control over camera settings and possibility of connecting an external shutter to either a Nikon D80 with 3872 x2592 pixels and a Nikon D7000 with 4928 x3264 pixels.

The shake table was placed inside a laboratory, hence the camera to object distance was clearly much shorter than during a typical UAV image flight. This required focusing of the lens to eliminate optical blur and acquire a sharp image. The focal length of a lens is defined for its focus to infinity and changes when the lens is focused on a distances shorter than infinity. The focusing was carried out using the autofocus functions of the cameras to ensure that no visual impairment influences the sharpness of the images. After focusing, the lens was fixed to prevent any changes to the focal length. This was done by switching off the autofocus option and taping the focus ring so that it could not be moved any longer.

The next influencing parameter was the aperture, which controls the depth of field (Emling, 2008). The depth of field defines how wide the range of the focused area is (Emling, 2008). A small aperture opening ensures a large depth of field. However, a small aperture opening does also reduce the amount of light that gets to the image sensor. As the laboratory was inside and also no large depth of field was required, the aperture opening was large to ensure enough light reaching the sensor. However, a larger aperture opening also causes more chromatic aberration, which needed to be prevented because the effect has similarities to optical blur (Section 3.4.2).

To prevent underexposed images the film speed (ISO) and exposure time had to be adjusted. The film speed defines the sensitivity of the sensor (Emling, 2008). A high

(a) ISO 125, exposure Time 1/500.

(b) ISO 1600, exposure Time 1/1600.

(c) ISO 3200, exposure Time 1/2000.

Figure 2.19: Comparison of different film speeds.

sensitivity requires fewer protons to trigger a signal on the pixel. However, a high sensitivity is more inaccurate and results in noise in the final image (Figure 2.19) (Emling, 2008). Figure 2.19(a) shows an image acquired with a low ISO value. The writing in the image is easy to read and is sharp except a few problems around the letters. Figure 2.19(b) and Figure 2.19(c) are decreasing in readability and sharpness. The aim of UAV image flights is to acquire high quality images. Therefore the film speed should be as small as possible to reduce noise. A low ISO prevents the image from appearing blurry even when there is no optical and motion blur. An image acquired with a low ISO will be a higher quality but may have a tendency to produce a darker image (Emling, 2008).

A last setting critical to achieve acceptable images was the exposure time (Emling, 2008). Changing the exposure time directly influences the image brightness. A long exposure time creates a bright image whilst a short exposure time darkens the image. Figure 2.19(a) has the longest exposure time but because of the film speed the image appears as bright as 2.19(b) and 2.19(c) which both have shorter exposure times. Long exposure times are normally avoided as they often result in motion blurred images due to hand shake (Emling, 2008). The exposure time was important because it also controlled the size of motion blur (Section 2.2.1). By choosing long exposure times, images with larger camera displacements were generated. However, long exposure times also caused the disadvantage that the table had to move very slowly to generate small

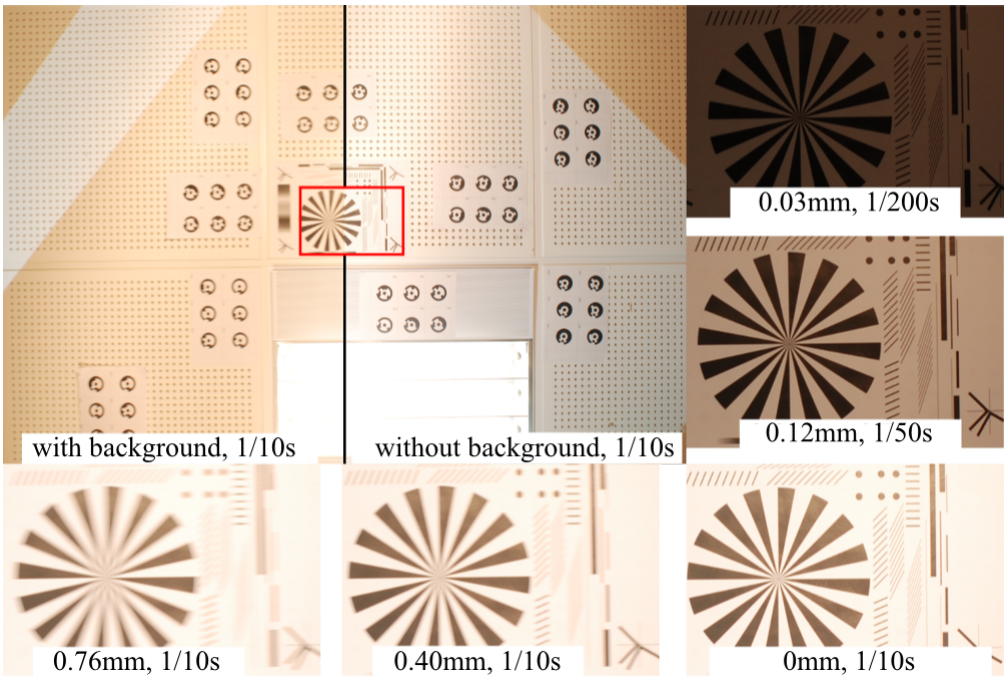


Figure 2.20: Various images taken of the test field with the camera facing towards the ceiling.

camera displacements. In the worst case, the signal was too low and the table could not overcome frictional resistance and did not move at all. Very short exposure times were also problematic because the table needed to move very fast to acquire images with large displacements. The speed, acceleration and deceleration periods could cause the table to reach its limits, causing damage or introducing additional unwanted blur. As a result the exposure time had to be varied to create images with small and large motion blur, as well as images with acceptable values of image brightness.

To support the image acquisition process inside the laboratory additional light sources were required to illuminate the scene. A projector was used both as light source, but also to perhaps project different textures on the ceiling (Figure 2.20). However, the projections proved to be too weak and were excluded from further experiments.

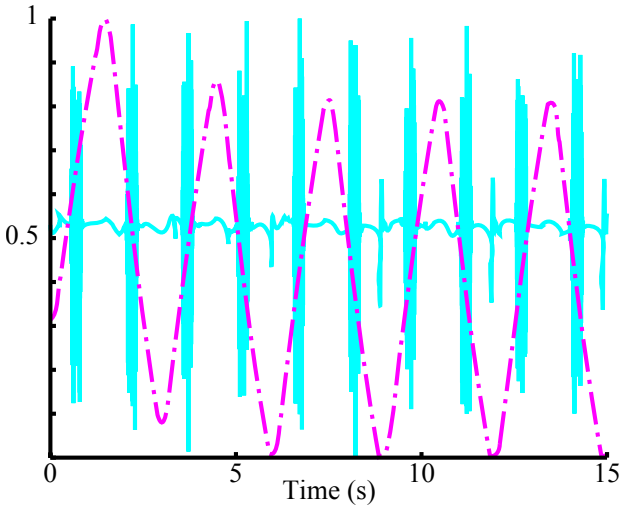
2.2.4 Determine image blur

Several shake table experiments were carried out using a range of camera settings and set-ups. During the experiments the output of the sensors attached to the camera and table were recorded and used for analysis. Laser displacement sensor observations were used to determine camera displacement during image acquisition. Figure 2.13 shows the laser displacement sensor measurements together with observations of the acceleration sensor attached alongside the camera body across direction of movement. It is possible to see that the acceleration sensor measures extreme vibrations during the period of constant velocity. If the time interval is reduced and camera trigger signal is included in the representation (Figure 2.21(b)) it is possible to see that a vibration appeared just after the trigger signal, followed by a short flat period and then followed by vibrations again. It can be assumed that these additional vibrations observed by the acceleration sensor are caused by the opening and closing of the shutter inside the camera body. The vibrations caused by the opening are so extreme that they can be measured by the acceleration sensor. The delay between shutter trigger and shutter opening can be explained by the delay caused by the hardware. For calculating the precise motion of the camera the displacement between shutter opening and shutter closing should be analysed. The surprisingly strong vibration of the moving shutter inside the camera was evaluated during post processing to assess whether this vibration influences the captured blur. Therefore, the value measured by the acceleration sensor dependent on its sensitivity has to be integrated twice over the time, to convert acceleration to displacement (Table 2.3). The calculated displacement based on the

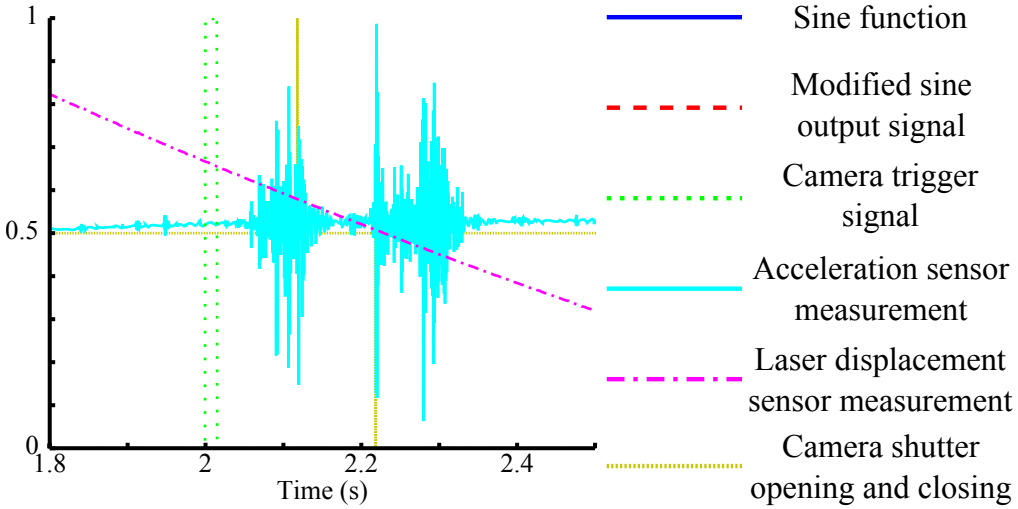
Table 2.3: Accelerometer properties of shear accelerometers manufactured by PCB Piezotronics (2015).

Sensor ID	Model ID	Sensitivity (mV/g)
38543	333B45	492
99363	352C68	99.3014
99357	352C68	102.0174

measurements of the acceleration sensor was at maximum $0.7 \cdot 10^{-5}\,\mu$m in comparison to a pixel size of $6\,\mu$m. It was therefore concluded that the movement caused by the mirror is insignificant.



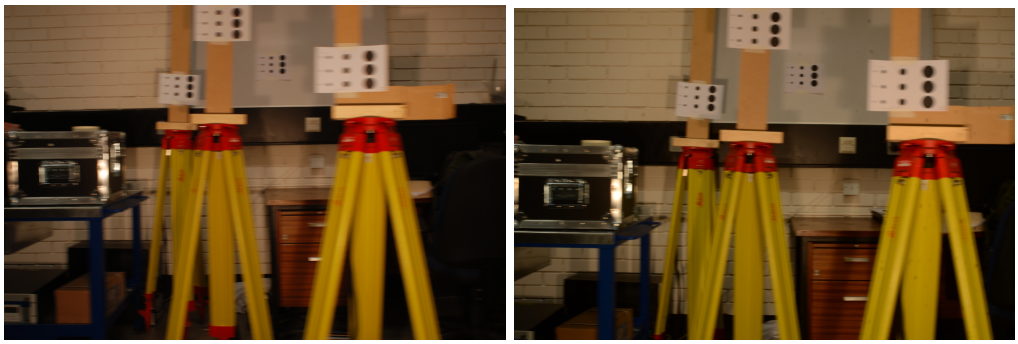
(a) Measurements of sensors.



(b) Measurements of sensors, camera trigger and camera shutting.

Figure 2.21: shake table measurements (normalised).

The vabrations measured by the acceleration sensor were used to determine when the camera shutter was opend and closed. It can be assumed that after the highest peak of the vibration the shutter was opened and closed before the highest peak of the second vibration. The highest peak is caused by the shutter arriving at its end position. The closing of the shutter occurs immediately after exposure time. After establishing the opening and closing of the shutter using the acceleration sensor, the displacement was measured with the laser displacement sensor. During the opening of the shutter the



(a) Maximal blur with Nikon D80 (b) Maximal blur with Nikon D7000

Figure 2.22: Strongest blurred images acquired from shake table.

movement was considered linear. Small deviations in the laser sensor observations can be identified but remain small when compared to the overall displacement. These are insignificant and can be considered as noise. The linear motion enables an accurate determination of camera displacement in the object space.

After determining the camera displacement in meters the displacement can be converted into a movement in terms of sensor pixels by dividing the displacement with the sensor pixel size. This is necessary to understand how far the sensor was moved and visualise how the influence of this movement appears in a digital image. The maximum displacement achieved with the shake table was approximately 396 pixels (= 2.42 mm) with the Nikon D80 and 486 pixels (= 2.33 mm) with the Nikon D7000 (Figure 2.22). In total, more than 2000 images were generated with exposure times between 1/400 s to 1/10 s using ISO 100 and apertures $f/8$ for the camera facing the ceiling and $f/13$ or $f/14$ for the camera facing the wall (Table 2.4).

Table 2.4: Overview over generated test images.

Date	Exposure times	Exposure time range (in s)		Blur range (in px)		Number images	Camera used	Camera direction	Comments
		Min	Max	Min	Max				
12/12/2013	-	-	-	-	-	-	-	-	First initialisation of the shake table and basic understanding of how it works.
07/01/2013	3	1/10	1/400	0	140	170	D80	ceiling	Not processed dataset. Test to modify the scripts. Sine function needs to be modified and more light sources are required. Changing Matlab functions and input parameters. Writing scripts for automatic analysis.
06/02/2013	3	1/10	1/200	0	185	1195	D80	ceiling	Dataset is concentrated on generating many different blurs.
26/02/2013	5	1/10	1/50	0	130	395	D80	ceiling	Dataset is concentrated on different image brightness.
24/10/2013	1	1/5	1/5	0	225	140	D80	wall	Images made of a 3D model.
14/11/2013	1	1/10	1/10	0	179	178	D7000	wall	Images made of a 3D model.
28/11/2013	1	1/5	1/5	0	396	135	D80	wall	Images made of dots at different depth.
28/11/2013	1	1/5	1/5	0	486	75	D7000	wall	Images made of dots at different depth.
12/02/2013	3	1/10	1/200	12	0	50	779	-	Mathematically blurred with Matlab motion blur filter based on 06/02/2013 dataset images without a-priori blur.

2.2.5 Results

The shake table approach provides a novel and accurate method suitable for generating blurred images with known motion blur characteristics. The main advantage is the automatic generation of a vast number of images, with a wide range of linear motion blur. It provides a method to accurately calculate the displacement subject to each image. However, the shake table is limited to a one directional movement making the generation of multi directional blurred images impossible. As this research focuses on UAVs this limitation is not considered relevant as UAV images normally only contain one directional forward motion blur, or blur caused by angular movements of the UAV (Section 2.2.1). Complex motion blur paths occur rarely as the exposure time of UAV images are too short.

The laboratory only provides a short camera to object distance, which challenges the applicability of the shake table approach for larger camera to object distances. UAVs have a flight altitude of around 100 m for a typical UAV image flights. To confirm the laboratory tests are relevant, camera displacement for typical UAV image flight should be calculated. Normal UAVs have a flight speed of around 54 km/h and an image exposure time of 1/400 s (Grenzdörffer et al., 2012). This implies that the camera should experience a displacement of 37 mm during exposure. This is 15 times greater than in the laboratory test, with a displacement of just 2.4 mm. The camera to object distances in the lab test was 1.6 m, 63 times shorter than normal UAV flight altitude of 100 m. The ratios show that the lab tests are comparable with a typical UAV image flight. However, this calculation does not consider angular movements of UAVs, which are much faster and cause more extensive image blur (Grenzdörffer et al., 2012).

2.3 Chapter summary

This chapter has provided basic background concerning images and image processing used in this thesis. It represents only a small amount of information and algorithms available for image processing, but should give a basic understanding for the following algorithms. Section 2.2 shows the basic theory for setting up a shake table experiment

and presents how blurred images used in this project were generated. The production of blurred images was necessary to prove the degrading effect of motion blur and its influence on photogrammetric processes.

3 Blur disturbs

This chapter uses the blurred images generated with a shake table to establish whether blur disturbs photogrammetric image processing procedures. Therefore, various photogrammetric operations have been conducted using imagery with varying amounts of blur on these procedures. Firstly, processes like camera calibration and bundle adjustments were examined. Next the influence of blur on the detection of photogrammetric targets was analysed. This includes assessing the impact upon both types of targets, coded and uncoded. Lastly, the influence of blur on the detection of random and unmarked feature points and the effectiveness of referencing between blurred and sharp images was investigated. Initially, a general overview of camera calibration and bundle adjustment is presented, before the various experiments are described and analysed. The chapter finishes by concluding whether blur disturbs, and establishes its significance on photogrammetric procedures.

Terminology In analysing the various procedures it is important to distinguish carefully between the meaning of the terms 'detection', 'measurement', 'identification' and 'referencing'. 'Detection' is the process of finding a target in an image. As targets used in photogrammetry are often of a circular shape, an algorithm is used to automatically detect round objects in the image. It finds the boundaries of a circle and then uses various techniques to calculate the centre of the circle (Luhmann, 2014). The detection process can be carried out in two ways: fully automatic, where the algorithm processes the complete image trying to find targets; or semi-automatic, where the operator defines a region of interest (ROI) and adjusts additional parameters. The latter approach was adopted to detect targets in most of this study. The process of detection is important as subsequent procedures are used to calculate the centre of the detected target and

derive a 'measurement', which is hopefully of sub-pixel precision and represents the actual measured location of the target in the plane of the image. However, if the target is not detected, or incorrectly detected, then automatic measurement is not possible or the measured coordinates will be incorrect. If automatic 'measurements' are unsuccessful it is still possible to manually measure coordinates in the image. However, manual measurements are rarely practicable and therefore it is important to successfully detect a target in order to ensure automation and measurements which are of sub-pixel precision.

With an automatically detected and measured target it might also be possible to carry out identification. The term 'identification' is used in this chapter to refer to the process of assigning an identifying integer to a target. This is normally achieved automatically using a coded bit pattern surrounding the target. The code needs to be clearly readable to prevent incorrect identification (Shortis and Seager, 2014). Finally, an identified target can then be 'referenced' which refers to the process of connecting identical targets across multiple images. This can be achieved by using the identifying integer of targets which have been successfully identified.

3.1 Literature review

This literature review will give an overview of techniques which are used in normal photogrammetric operations, typically applied during industrial measurements (Luhmann, 2010a), where special photogrammetric targets are used. They are required for precise camera calibrations and bundle adjustments.

3.1.1 Photogrammetric targets

Photogrammetric targets may be defined as special points of interest (POI), with the potential to improve photogrammetric procedures. They provide identifiable points and potential for fully automatic point identification (Luhmann et al., 2014). In many cases they consist of simply black shapes on white paper and are placed on walls (Luhmann,

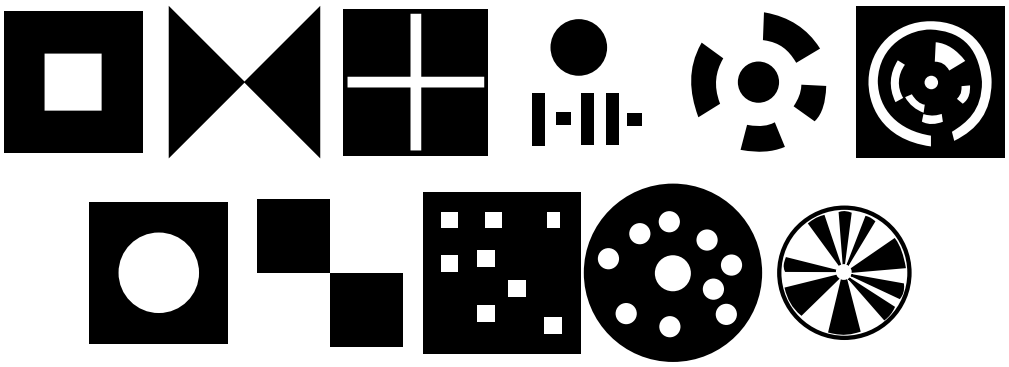


Figure 3.1: Various targets uncoded and coded.

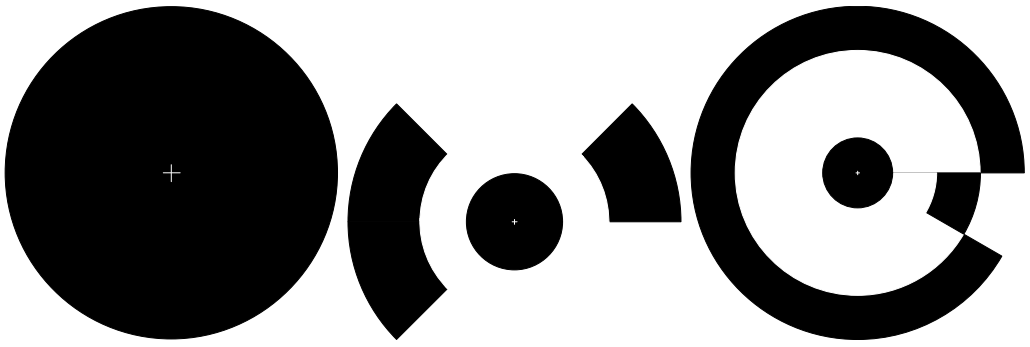
2014). However, they can also consist of more advanced structures containing a code which uniquely identifies individual points (Galantucci et al., 2008). The material, colour and especially the size of the target depends on the purpose of use. Targets for image flights used outside with a large camera to object distances should be constructed from weather resistant material and need to be physically larger; targets used in an indoor laboratory test can be only a few millimetre in size (Dold, 1996) and made of less robust materials. Targets should be:

- well defined, so they are easily identified;
- consistent, as to be visible from various POVs;
- unique, to be differentiated from other natural shapes in the image (Atkinson, 2001).

However, even if the targets meet these requirements, automatic detection methods can confuse valid targets with similar appearing objects in the image. Hence, a human operator is then required to distinguish between valid and invalid features.

3.1.1.1 Examples of photogrammetric targets

There is a range of different shaped photogrammetric targets which have been adopted (Figure 3.1). Many are uncoded but provide the possibility to be measured accurately. Many utilize a circular dot in the centre and include a small cross to define the exact central location (Figure 3.2(a)). Coded targets are often based on these simple designs,



(a) Dot target with cross in the centre.

(b) 8 bit coded target. Only the centre dot is used for measurement.

(c) RAD coded target. Only the centre dot is used for measurement.

Figure 3.2: Circular shaped targets.

but include 8 bit coded (Figure 3.2(b)) or ringed automatically detected (RAD) code that can be used by specific software such as PhotoModeler (Figure 3.2(c)).

Coded targets are rarely used in aerial applications, because they would need to be too large, expensive and hence difficult to produce. For example PhotoModelers estimates that a RAD coded target would have a diameter of 1.7 m, for a flight altitude of 100 m, using a $c_k = 24$ mm lens on a Nikon D80. These practical limitations necessitate the use of simple targets, involving just a dot, cross or chess pattern target, without the possibility of automatic identification.

3.1.1.2 Coded target detection, identification and measurement

One method used for target detection and measurement involves template matching (Bethmann and Luhmann, 2011) (Section 2.1.1.3). Once a best match has been found, the shift parameter are measured as the target centre (Luhmann, 2014). Often target detection is a separate step, before target measurement. To identify and measure a target, it is first necessary to find the target in the image. There are various approaches (Hattori et al., 2000; van den Heuvel et al., 1992) most begin by detecting the edges of an image and filtering out every edge that does not match the requirements defined by the target (Xia et al., 2012). In the case of round targets, all lines would be excluded as well as edges that are too long or short (Xia et al., 2012). All structures that pass these

tests are assumed to represent targets.

In the next step, target measurement takes place. Measuring a target accurately can be achieved using a range of different methods. One classical approach uses the target moment or gravity centre (Bradski and Kaehler, 2008). For this method the intensity values of the target are used and the Green theorem applied, which describes the integral of a plane region with a line integral (Green, 1828; OpenCV Dev Team, 2014). The gravity centre is returned and can be taken as a coordinate measurement of the target. Another method is based on the contours of the target. A best fitting ellipse is determined using the edge pixels, then the centre of this ellipse can be calculated and taken as a measurement for the target (Luhmann, 2014). All these methods return a measurement which represents the centre point of a target. The surrounding code is ignored due to its irregular shape.

In cases where it is desirable to read the target code and identify a target, a third step has to be applied. The code which is defined by the target type is relative to the target centre. As the target code often consists of a bit pattern, the identification algorithms needs to detect the bit segments and decode them (Xia et al., 2012). When the bit pattern is decoded an identifying integer can be assigned to the target (Wijenayake et al., 2014).

3.1.2 Camera calibration and adjustment theory

The concept of a discrete targeted point and its location in both the image and the object space is an important element in photogrammetry specifically for bundle adjustments and camera calibration. They provide accurate coordinate measurements as well as the possibility for repeated detection during motion.

3.1.2.1 Camera calibration

Targets are required for accurate image coordinate measurements, which can be repeated across frames, but only coded targets support easy re-identification across multiple

frames. The measurement of one target in two different images can be used to calculate 3D coordinates for the measured target (Kraus, 2007), but camera calibration is required to achieve more accurate coordinate estimates (Luhmann et al., 2014). A camera calibration procedure determines the inaccuracies interior camera parameter and can also be used to convert pixel to photo coordinates, which are measured in millimetres on the sensor and can be used for the collinearity equations to relate image to object (OpenCV Dev Team, 2014).

There are different ways of calibrating a camera, including laboratory calibration, field calibration, on-the-job calibration and others (Luhmann, 2010b). Laboratory calibration has been applied traditionally to high accuracy photogrammetric cameras that provide a permanent stable interior orientation (Hofmann, 2005; Meier, 1976). This is not applicable for DSLRs, because they are unstable and they are calibrated either using a test field, or on-the-job because these methods are easy and more practicable (Luhmann, 2010b).

Test field calibration requires a field with signalised targets of known coordinates or distances between targets. The targets should be distributed over the complete image area (Luhmann, 2010b). Subsequently targets are detected and measured in the image. Lastly the measurements are used in a bundle adjustment to determine the unknown camera parameters (Luhmann, 2010b) (Section 3.1.2.2). No changes to the camera set up should occur between the field calibration and acquisition of the project images.

Self-calibration, or on-the-job calibration also uses a bundle adjustment with unknown camera parameter. However, instead of images of a dedicated test field the images of the project are used. However, unknown 3D coordinates of object points are introduced and solved in the same process. Additional sensors such as IMU and GNSS can be included in the collinearity equation (Equation 2.6) to calculate parameters necessary to model the camera geometry.

3.1.2.2 Adjustment theory

In general, it is comparably easy to design a photogrammetric project to allow 'on-the-job' calibration. In fact there are often more measurements than required to solve the collinearity equation using a bundle adjustment. Bundle adjustment is a special case of least squares estimate. Least squares estimate is the process of adjusting observations with least residuals to a model and is based on a Gauss-Markow-Model (Triggs et al., 2000).

Adjustment theory uses least squares to minimise residuals from all acquired measurements and to determine unknowns. The design matrix (A) is a linearised functional model, which allows the relationship between observations and result to be calculated (Niemeier, 2007). In the case of a camera calibration this contains the linearised version of the camera calibration equation used to extend basic collinearity (Equation 2.6). It is also possible to combine different functional models to integrate different measurements (Schneider, 2010). In a first iteration, approximate values (X_0) for the unknowns are required, which can be calculated due to the redundancy (Niemeier, 2007). Using the design matrix (A) and the approximate values it is possible to calculate the initial observations L_0 and the shortened observations (l)(Niemeier, 2007):

$$l = L - L_0 \tag{3.1}$$

L is the vector with the values of the blurred image. The weighting matrix P can be used to give certain observations a higher or lower weight, dependent on the precision of the observations. It is defined that all values on the main diagonal of the identity matrix (I) are equal. With these values it is possible to calculate the parameter vector ($\hat{x}$) and improvement vector (v), which can be used to calculate an adjusted unknowns ($\hat{X}$)(Niemeier, 2007):

$$\hat{X} = X_0 + \hat{x} \tag{3.2}$$

and adjusted observations:

$$\hat{L} = L + v \tag{3.3}$$

In the following iteration $\hat{X}$ and $\hat{L}$ replace X_0 and L and the process is repeated until a certain number of iterations is exceeded or the parameter vector is so small that changes become insignificant (Niemeier, 2007).

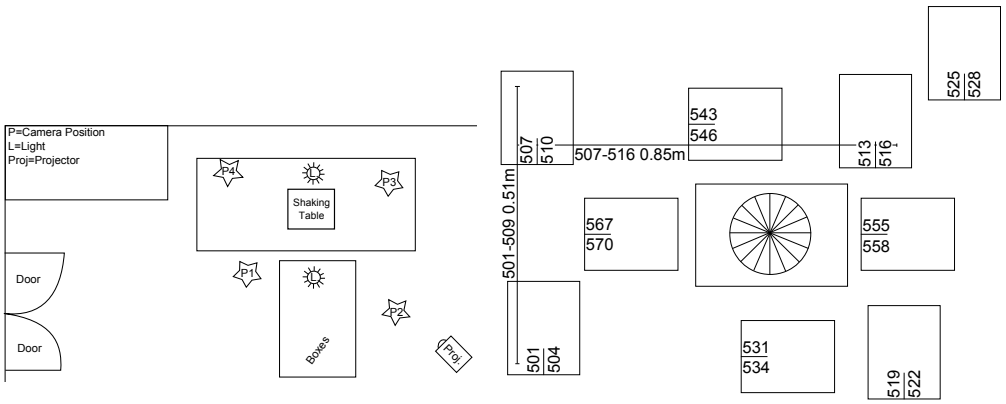
Bundle adjustment is a special case of least squares estimate. The functional model is comprised by the collinearity equation using photo coordinates as observation and object coordinates as unknowns. After each iteration the unkowns are updated ($\hat{X}$) but updating the observations (L) is not required (Cooper and Cross, 1988). However, least squares method provides the mathematical background for both, camera calibration and calculation of 3D object coordinates. It is the aim that the algorithm identifies appropriate parameters to minimise errors and inaccuracies, therefore reduce the influence of blurred images in photogrammetric processing. The experiments presented here demonstrate that the effect of blur is apparent, even if measurement redundancy is small.

3.2 Dataset I - Influence of blur on camera calibration

Three datasets were used to quantify the impact of image blur on automatic image processing. The first dataset was used to assess the influence of blur on camera calibration.

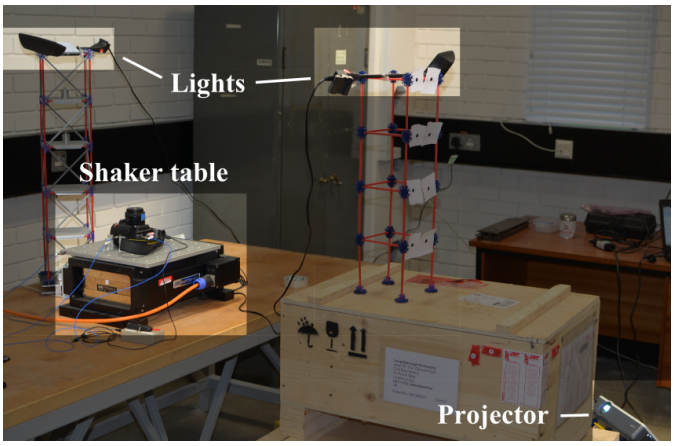
3.2.1 Setup and composition of dataset

The first dataset consisted of convergent images of a calibration field comprised of 54 coded targets and a Siemens Star located in the image centre (Figure 3.3). The Siemens Star provides a direct visual way to evaluate both the amount and the direction of blur. Coded targets were used to allow fully automated camera calibration. The calibration field was fixed to the ceiling with the shake table located beneath. The white ceiling



(a) Sketch of experimental setup.

(b) Sketch of target point composition.



(c) Photo of laboratory setup.

Figure 3.3: Setup of first dataset.

panels contained small ventilation holes which appear black and provide natural targets of good contrast and allowed a full visual analysis of blur. Furthermore, the ceiling was permanently accessible and visible to the shake table. Hence, the calibration field was fixed permanently and could be used for various experiments, without any changes to the field.

To acquire blurred pictures of the field, the camera was fixed on the shake table, facing towards the ceiling. The camera-to-object distance was approximately 1.6 m, which required focusing of the 24 mm lens. After focusing, the lens was fixed using tape and not changed during the subsequent shaking process. Fixing the lens ensures that optical errors caused by the lens are the same for all images and that the light's internal optical

path is the same for all images. The light path should be the same to ensure that any changes in the calculated calibration can be attributed to blur and not due to changes in the optical system.

With this experimental configuration approximately 2000 blurred images of the calibration field were acquired. Of these 2000 images, 13 images with various extents of blur were chosen to be analysed. The 13 images were equaly distributed over the complete available range of camera displacement created by the shake table, between 0 mm to 1.03 mm. The gap between 0.5 mm and 1.03 mm was caused by a gap in the dataset cause by the shaker table. To complete an appropriate camera calibration with a strong geometry, stable camera positions around the shake table were also established. A tripod was positioned at four locations around the shake table (Figure 3.3(a), P1-P4). The camera was fixed on the tripod, facing towards the calibration field and triggered using a remote shutter to prevented the camera from experiencing any significant shake and helped ensure sharp images. Images in four orientations were acquired including landscape, $\pm 90^{\circ}$ and 180°. In total 16 images were acquired under such stable conditions.

PhotoModeler requires six images, to achieve an acceptable camera calibration. The 16 images of the stable positions were reduced to a subset of five images, with an additional image chosen from the 13 images acquired using the shake table (Table 3.1). The camera calibration procedure was then implemented 13 times and it was clear that the degree of blur of the single image influenced greatly the calibration results.

Table 3.1: A list of parameters for the camera calibration dataset.

	Dataset 1
Camera to object distance	1.60 m
Size of target	8 mm
Number of coded targets	54
Focal length	25.35 mm
Aperture	$f/8$
Frames per process	1 blurred + 5 sharp image
Number of camera displacements	13

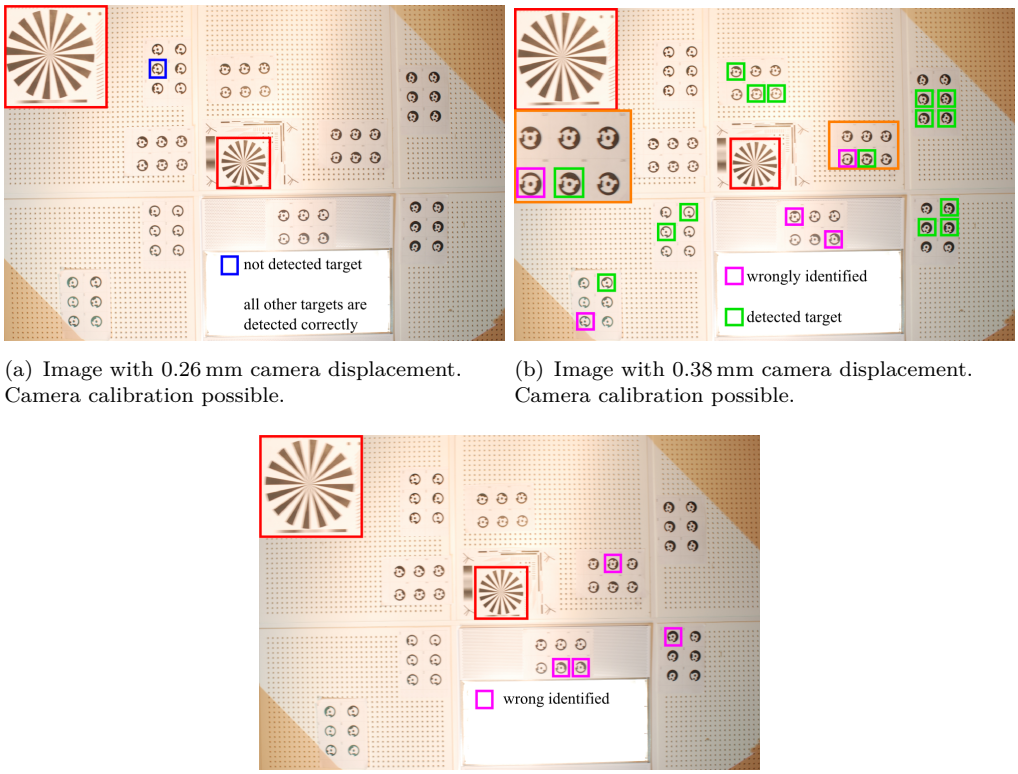
3.2.2 Camera calibration with blurred images

The first test included an image from a stable shake table, free from any camera displacement. This sharp, unblurred image was used to calculate an accurate camera calibration without any influence of blur. The result of this calibration, using sharp images only, was compared to the results of the calibration with blurred images. To carry out the calibration, *PhotoModeler* detects, identifies and measures targets automatically and carries out a self-calibrating bundle adjustment to determine best estimates to describe internal camera geometry. The 13 tests demonstrated that increasing blur causes target detection to fail, which makes subsequent identification and measurement impossible. Initial problems occurred for a camera displacement of 0.263 mm, when only 53 of 54 targets were detected (Figure 3.4(a)). With a camera displacement of 0.3 mm, eight targets remained undetected. With a camera displacement of 0.32 mm, only 38 of 54 targets were detected and identified. However, one of the detected targets was incorrectly identified. Incorrect identifications caused the problem that targets are incorrectly referenced between images, which causes a gross error in the calculation. At a camera displacement of 0.38 mm, 18 of 54 targets were detected only, with four being incorrectly identified (Figure 3.4(b)).

The orange inset in Figure 3.4(b) shows a magnified target sheet with one correctly detected target and one incorrectly identified target, at a camera displacement of 0.38 mm. The other four targets presented in the inset were not detected at all. In manual identification the targets are clear and the code is still readable, however, automatic detection methods are not able to detect all targets or identify them correctly. Figure 3.4(c) shows an image with a camera displacement of 0.53 mm. Most of the targets remain undetected and the few that are detected are identified incorrectly.

With a displacement of 0.38 mm the automatic camera calibration procedure failed. This was because the targets were identified incorrectly, the wrong integer was assigned and referencing across multiple images was wrong. Figure 3.5(a) shows the number of detected targets and how many of them were referenced using fully automated procedures.

To ensure that the camera calibration results were not just influenced by the decreased



(a) Image with 0.26 mm camera displacement. Camera calibration possible.

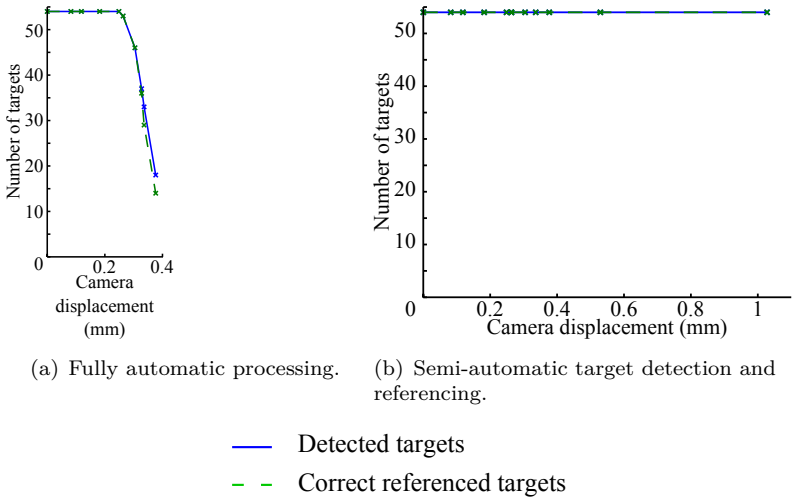
(b) Image with 0.38 mm camera displacement. Camera calibration possible.

(c) Image with 0.53 mm camera displacement. No camera calibration possible.

Figure 3.4: Success of detection of targets in blurred images. The Siemens Star clearly indicates how blurred the image appears to humans.

number of detected targets, semi-automatic detection and measurement of targets was performed using the 'sub-pixel target mode' provided by PhotoModeler (Figure 3.6). The 'sub-pixel target mode' is a tool which allows the operator to create a ROI which is used to detect targets. Due to the reduction to a small area the detection algorithm is more likely to find a target. However, the usage of the 'sub-pixel target mode' might also allow higher acceptance of other structure that are similar to targets, which can cause false detection. However, if the 'sub-pixel target mode' tool is not applied precisely on the centre dot of a coded target (Figure 3.6(a)) then accurate detection and measurement is not possible (Figure 3.6(b) and 3.6(c)). The 'sub-pixel target mode' tool has to be applied precisely on the centre dot of a target (Figure 3.6(d)) to achieve precise detections and measurements (Figure 3.6(e) and 3.6(f)).

Manual creation of the detection box, at the correct position with the right size is



(a) Fully automatic processing.

(b) Semi-automatic target detection and referencing.

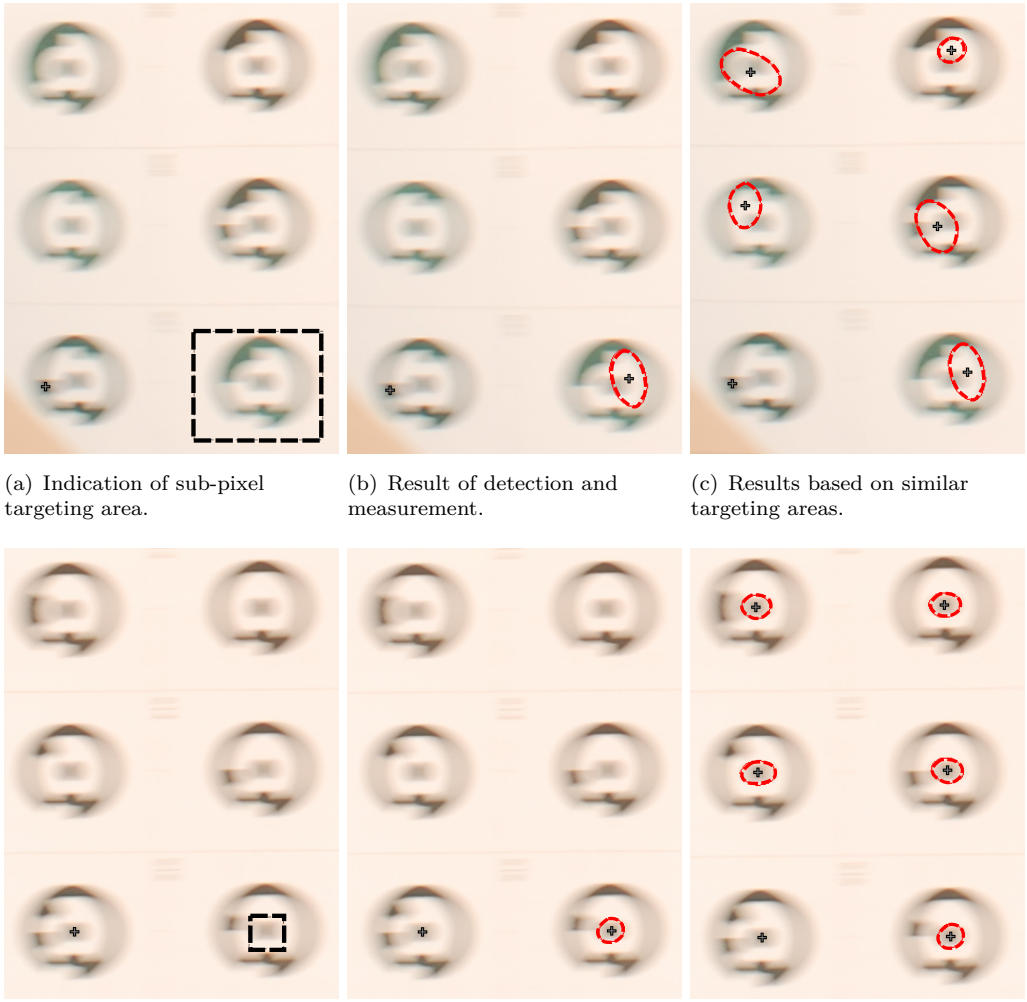
Figure 3.5: Difference between automatic and semi-automatic detected and referenced targets for camera calibration.

therefore prone to error and is also time consuming. Thus referencing these targets also needs to be carried out manually because identification of the targets is not automated. After this computer assisted intervention, all targets were detected and it was possible to assess the impact of motion blur on camera calibration. This assessment can be conducted under conditions similar to a camera calibration using sharp images (Figure 3.5(b)). Corrections calculated during the process can be distributed over all targets and the variations from calculated results can be attributed to blur.

3.2.3 Result of camera calibration with blurred images

The problems that occurred in detecting targets demonstrates that blur has three ways of influencing the level of automation achievable during camera calibration. First, misidentification of targets, which can result in incorrect referencing of targets between the images. Second, misdetections and subsequently mismeasurement which can influence coordinate calculation. Third, zero or limited detection of targets, which reduces the number of measurements used to calculate calibration parameters.

The third problem is connected to the target detection process. A detection algorithm attempts to detect the edges of a target, these edges are then analysed if they fulfil



(a) Indication of sub-pixel targeting area.

(b) Result of detection and measurement.

(c) Results based on similar targeting areas.

(d) Indication of sub-pixel targeting area.

(e) Result of detection and measurement.

(f) Results based on similar targeting areas.

Figure 3.6: Semi-automatic target detection and measurement on image with 1.03 mm camera displacement.

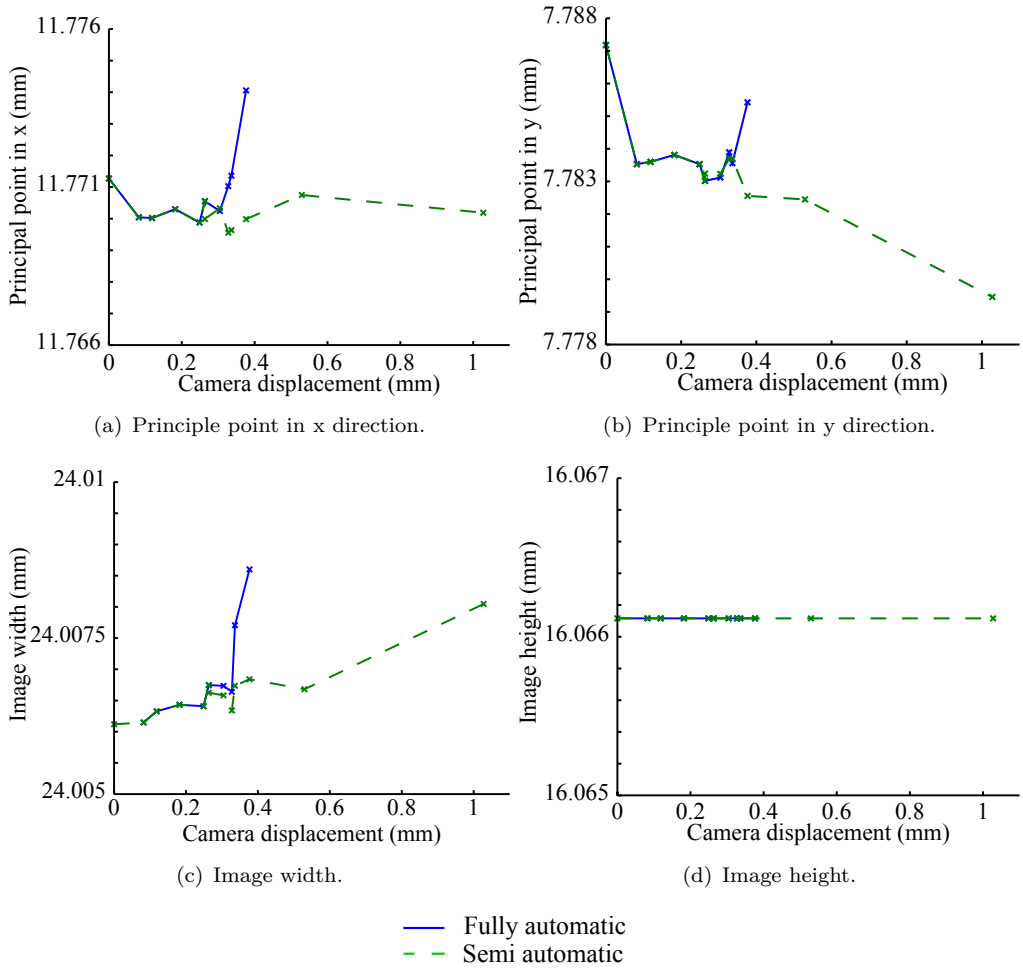
certain requirements (Section 3.1.1.1). One requirement is the target shape, which is in this case a circle. However, because of perspective distortions circles can appear as an ellipse, or can be distorted in other ways. To ensure that the algorithm does not ignore these distorted targets, a 'roundness' parameter is defined. This parameter specifies how much a target may be distorted before it is excluded. A second parameter then analyses the size of the detected shape. If the detected shape is too small or too large it is not considered as a potential target. In a blurred image the targets are distorted due to motion blur. There are three thresholds that can result in the rejection of a target if

exceeded. The edge detection might fail because of smear, which can be so strong that the transition between target and background prevents an edge from being detected. If there is no edge then no target can be found. A second cause for rejection can occur if an edge is detected and connected to a distorted shape. The target is not accepted as round and the edge rejected from further processing. A third exclusion cause is that the target could become too large due to blur. If the target is larger than expected, the detection algorithm will assume it is another structure and is rejected from further processing. These thresholds determine if a target is rejected from subsequent processing.

A more problematic error is the misidentification of targets and their subsequent inclusion in processing. The measurement of misidentified targets can affect the calculations and cause incorrect results and misidentifications need to be excluded. Alternatively, the identifying integer has to be corrected, which is preferred.

Besides the obvious influences caused by motion blur, it was also found that motion blur generates small change in the estimation of camera calibration parameters, compared with sharp images (Figure 3.7). The decreased accuracy of automatic target measurement can be illustrated by the change of the principal point position (Figure 3.7(a) and 3.7(b)) and the variation in calculated image size (Figure 3.7(c) and 3.7(d)). Figure 3.7(a) and 3.7(b) demonstrate that the position of the principal point varies from the position calculated for a sharp image set. This variation is $\approx 8\,\mu$m larger than one sensor pixel ($\approx 6.1\mu$m). Figure 3.7(a) and 3.7(b) show that the calculated image width has a tendency to increase with larger blur. This can be explained perhaps by the shake direction, which is along the x-axis. The failure of fully automatic detection and the requirement of semi-automatic target detection also implies that greater blur will cause increasing problems for automated photogrammetric processing.

The detection easily finds edges that are aligned in the direction of blur, because these edges are less affected. In contrast, edges perpendicular to the blur direction create a transitional smearing effect between the target and the background, making the detection of edges difficult. Subsequent estimation of the middle, between the start and end of the target, becomes inaccurate and the measured centre does not represent the true target centre. Due to this incorrect detection, subsequent measurements are imprecise, especially in the direction of blur. Measurements derived perpendicularly to



(a) Principle point in x direction.

(b) Principle point in y direction.

(c) Image width.

(d) Image height.

Figure 3.7: Results of camera calibration with blurred images.

the blur direction remain uninfluenced. As these images are blurred along the x-axis, x-coordinates are the most likely to be inaccurate. The image width is an example as the scaling of the image is influenced by the incorrect measured x-coordinates. It is perhaps surprising, that the principal point does not change its position on the x-axis but varies by about 8 μm on the y-axis. There is no obvious explanation for this observation other than simply just six images are unable to provide ideal geometry for reliable camera calibration. Another reason could be based on the close relationship between image size and principal point. It could also be that *Photomodeler* distributes the effect cause by blur in the image width only. Image size and principle point are close related so that probably a clear seperation of errors between image size and principal point is not possible.

Photogrammetry carried out with targets enables measurement accuracies of 1/20 pixel (Luhmann, 2014). As the variation in the principal point is larger than one pixel it can be assumed that measurements of 1/20 pixel precision cannot be achieved with blurred images. These imprecise measurements could influence subsequent calculations of 3D coordinates, compared to calculations achieved with sharp images. However, as camera calibration normally uses a larger set of calibration images, the influence of one blurred image will be insignificant as outliers can be detected and excluded.

Parameters for the radial lens distortion (k1, k2) and decentering lens distortion (p1, p2) were calculated during the calibration process. It was found that neither had any blur related dependency. However, both problematic automatic target detection and a change in the calibration results imply a deteriorating tendency with motion blur, as may be expected.

3.3 Dataset II - Influence of blur on 3D object coordinates

As the first dataset showed, blur does have an influence on camera calibration but effects are minimal as a tripod is usually involved. For a UAV image flight the accuracy of measured image coordinates and calculated 3D coordinates is more important. Consequently, a dataset was created to analyse the influence of blur on calculated 3D object coordinates.

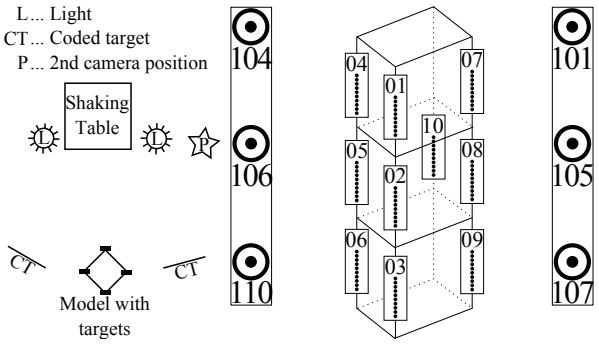
3.3.1 Setup and composition of the dataset

The second dataset was dedicated to the calculation of 3D coordinates, using targets appearing in blurred images. A Nikon D80 and in a second experiment, a Nikon D7000 DSLR camera, were fixed on the shake table facing a wall. An object was placed between the camera and the wall and was prepared with 130 signalised, non-coded targets. The targets helped to ensure that the same points on the object were measured and that subsequently the influence of blurred images could be analysed. Vertical bars were positioned on both sides of the object, each with three coded targets (Figure 3.8).

For this experiment a 24 mm lens was used and focused on the object, which was approximately 1.7 m away from the camera. The focus was fixed so that it could not change during the shaking process. The distance to the wall was 3.5 m. Out of focus blur was suppressed using an appropriate setting for the aperture $f/14$ ($f/13$ for D7000) to achieve an appropriate depth-of-field. The film speed was set to 100 ISO and an exposure time of 1/5 s (1/10 s for the D7000) was necessary.

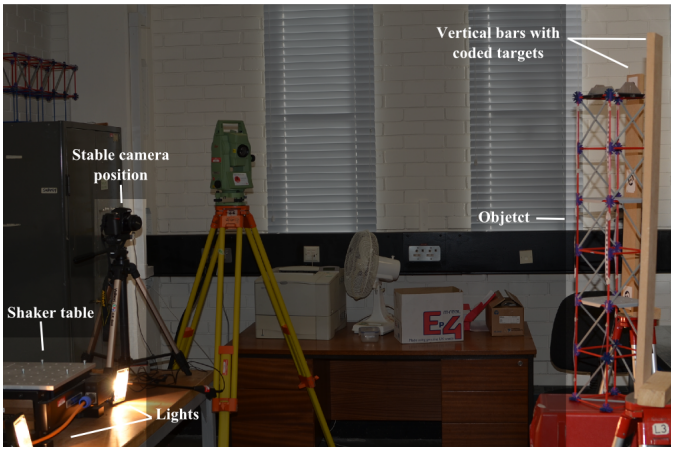
328 images were generated, (178 for D7000) each acquired with a wider range of camera displacements. The huge amount of manual measurements required, limited the total number of processed images to a subset of 6 images, which were equally distributed over the range of camera displacements from zero to 1.51 mm (0 to 0.86 mm for D7000).

The calculation of 3D coordinates in photogrammetry requires image coordinates to be acquired from at least two different images. Therefore, an image from a second stable position was included in the processing. This second image was taken from a position



(a) Sketch of experimental set up.

(b) Sketch of target point composition.



(c) Photo of laboratory set up.

Figure 3.8: Set up of second dataset.

beside the shake table, using a tripod and external shutter button to prevent any camera shake. The distance to the object was similar to the distance between the shake table and the object, to ensure that no adjustment to the focal length was required. The distance between the shake table camera position and the stable tripod camera position was 0.82 m.

The sharp image from the tripod was paired with each of the six blurred images and processed using PhotoModeler (Table 3.2).

As stated, 2D image coordinates have to be measured in at least two images. For the Nikon D80, 128 of the targets were visible from the shake table position as well as from the stable camera position. As the obstructed targets were different for both images it

Table 3.2: A list of parameters for the 3D object point dataset.

	Dataset 2
Camera to object distance	1.70 m
Size of target	9 mm
Number of coded targets	6
Number of not coded targets	130
Focal length	28.86 mm (40.11 mm)
Aperture	$f/14$ ($f/13$)
Frames per process	1 blurred + 1 sharp image
Number of camera displacements	6 (+1 blur repeated used for manual target measurements)

was possible to reference 126 targets only between both images. In the Nikon D7000 experiment, three targets were obstructed for the stable camera position and two for the shake table. This resulted in only 125 common targets that could be referenced between both images.

Additionally, both cameras underwent a camera calibration. This calibration was undertaken after the shaking process, using only sharp images of a camera calibration test field. The purpose of the camera calibration was to ensure precise knowledge about the camera parameters and to achieve precise results for the calculated 3D object coordinates.

3.3.2 Calculation of 3D coordinates with blurred images

At first automatic detection of targets was carried out using the sharp image. The detection, identification and measurement of all six coded targets and the 128 uncoded targets was successful. A few points were misdetected as targets, even when they were not signalised, but manual intervention excluded them from further processing. Automatic detection was also carried out using the blurred images from the shake table and Table 3.3 presents the result of fully-automatic detection. Similarly to the results found with the first dataset, the automatic detection of targets became more unreliable with increasing blur. It would appear that detection and referencing becomes increasingly impossible with increasingly blurred imagery (Table 3.3). It can be seen that

Table 3.3: Influence of camera displacement on automatic detection and referencing of not signalised targets for Nikon D80 camera.

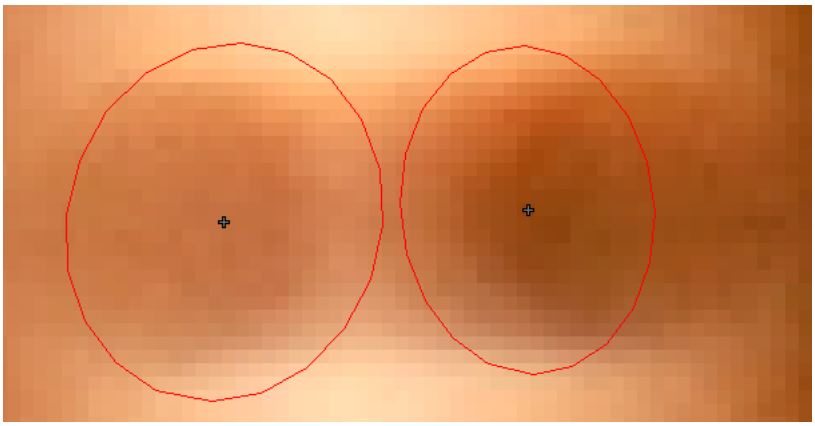
Camera displacement (mm)	Number of 'automatic target marking' targets out of 128	Automatic referenced targets out of 126
0	132 (4 not signalised)	125
0.20	130 (2 not signalised)	96 (4 incorrect)
0.32	135 (8 not signalised)	84 (3 incorrect)
0.49	126 (3 not signalised)	69 (4 incorrect)
0.99	0	80 (1 incorrect)
1.51	0	11 (7 incorrect)
1.51 (manual)	0	41 (3 incorrect)

the detection of signalised targets suffers significantly with larger camera displacements. With a displacement larger than 0.49 mm, detection fails completely. To ensure that all possible targets could be processed *PhotoModelers* 'sub-pixel target mode' tool was again used to detect the missing targets. Although initially it appeared successful, Figure 3.9(a) shows that the detection of circles can be actually inaccurate, due to blur. However, automated measurement generally derives a point close to the centre of the blurred target and the tool returns a coordinate to a sub-pixel precision. Difficulties encountered in automated target detection and measurement (Figure 3.9(b)), seem to be caused by the appearance of two target silhouettes caused by a large degree of blurring. Another explanation could be the combination of a black target, on a dark background. The high contrast of a black target on white background causes during the blurring that the target becomes paler. However, if the target is not just blurred with white background but also blurred over a darker area it regains darkness and appears darker but not as dark as the original black target. Figure 3.9(d) shows an example where the right silhouette appears darker than the left due to the darker background to the right of the target and two targets are detected. To analyse if semi-automatic measurement of blurred images is more precise than operator inputs, manual measurements were carried out. The most blurred image was taken and an operator measured all targets manually without computational support. The measurement was conducted as accurately as possible, in the middle of the blurred target, between both of the target silhouettes (Figure 3.9(c)). It would also be possible to use automated methods such as those presented by Boracchi et al. (2007), who calculated the middle between two targets silhouettes and returned a measurement for the centre or the two round silhouettes.

(a) Detection in image with 0.99 mm camera displacement.

(b) Detection in image with 1.51 mm camera displacement.

(c) Measurement in image with 1.51 mm camera displacement.



(d) Appearance of two target silhouettes, which the automatic tool detects both.

Figure 3.9: Detection of blurred targets with PhotoModeler's 'sub-pixel target mode' tool.

A next critical step is target referencing between images. As expected, referencing targets between images automatically becomes increasingly difficult as blur increases (Table 3.3). Referencing targets between a sharp image form the stable position and a non-blurred

image from the shake table returned 125 correct references from the 126 possible (four targets were not available in both images). A camera displacement of 0.20 mm returned only 92 correct references and four incorrect references. With increasing blur the number of correct references decreases significantly. Referencing is based on similarities between both images, which are more difficult to find with increasing blur. The remaining unreferenced targets were then referenced manually until all 126 targets were correctly referenced between both images.

A similar experiment conducted with the Nikon D7000 camera experienced similar problems to the D80 dataset and was solved in the same way (Table 3.4).

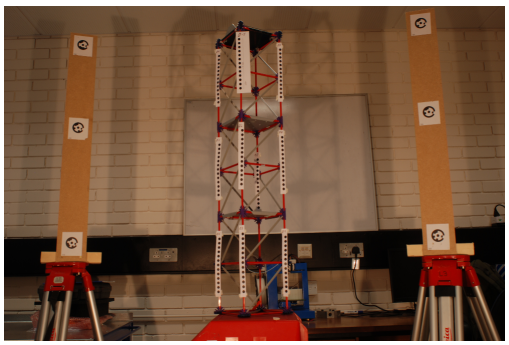
Table 3.4: Influence of camera displacement on automatic detection and referencing of not signalised targets for Nikon D7000 camera.

Camera displacement (mm)	Number of 'automatic target marking' targets out of 128	Automatic referenced targets out of 125
0	131 (6 not signalised)	103 (19 incorrect)
0.14	131 (6 not signalised)	96 (4 incorrect)
0.39	135 (8 not signalised)	101 (19 incorrect)
0.52	115 (2 not signalised)	68 (36 incorrect)
0.62	68 (2 not signalised)	61 (30 Incorrect)
0.86	0	102 (55 incorrect)
0.86 (manual)	0	125

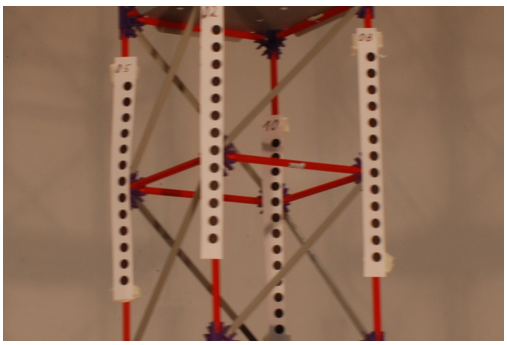
After successful referencing of the targets it was possible to calculate the 3D object coordinates for both the D80 and the D7000 dataset.

3.3.3 Result of 3D coordinate calculation with blurred images

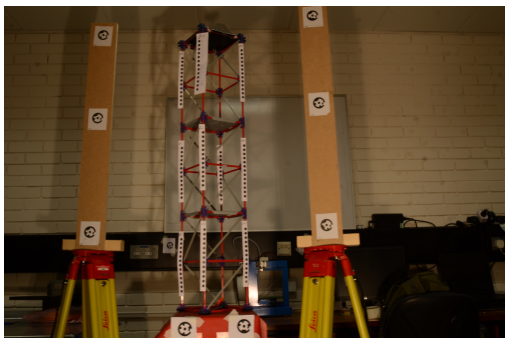
The second dataset used to calculate 3D coordinates using motion blurred images, supports the findings of the camera calibration dataset. A small degree of blur prevents fully-automated detection and requires some computer-assisted, manual or semi-automatic detection. With a camera baseline of 0.82 m and a camera-to-object distance of 1.70 m, an appropriate intersection angle for precise coordinate calculation is provided. Coordinate discrepancies between sharp and blurred image sets were calculated. This establishes that computed coordinates are inevitably influenced by increased motion blur. For a



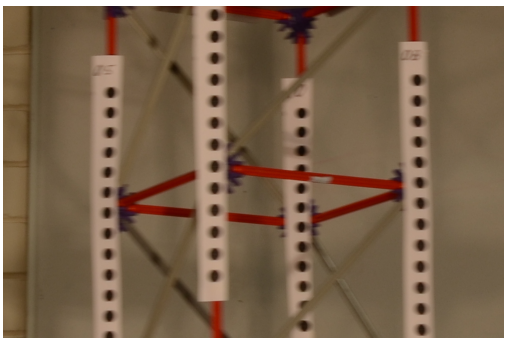
(a) Blurred image of D80 with 0.49 mm camera displacement.



(b) Inset of blurred image of D80 with 0.49 mm camera displacement.



(c) Blurred image of D7000 with 0.52 mm camera displacement.

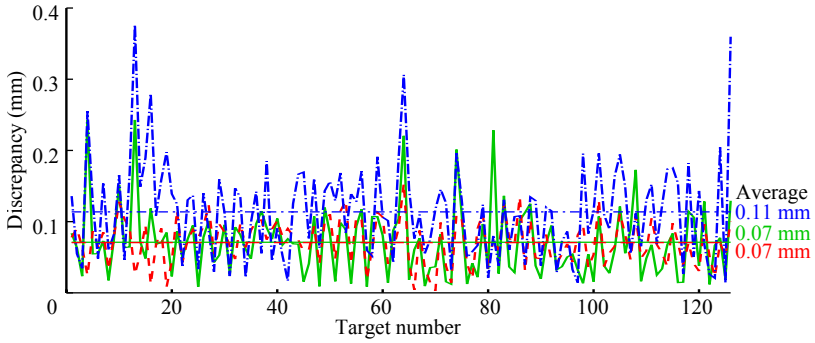


(d) Inset of blurred image of D7000 with 0.52 mm camera displacement.

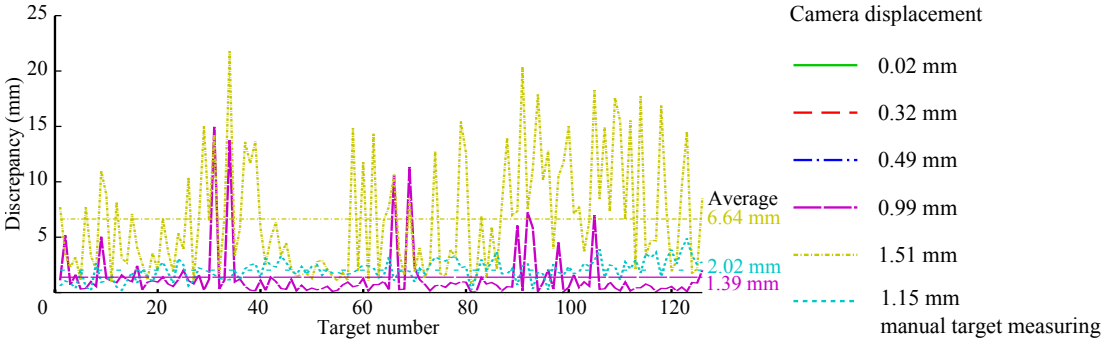
Figure 3.10: Comparison of similar camera displacement for D80 and D7000.

0.49 mm blurred image, the discrepancy was on average 0.11 mm (Figure 3.11(a)), which is small considering that one pixel has an approximate ground sampling distance of 0.35 mm. However, images with smaller levels of blur can have discrepancies of up to 0.4 mm, which is more than one pixel (Figure 3.10(a) and 3.11(a)). When blur is so large that automatic target detection fails, the accuracy of calculated coordinates decreases rapidly.

With the maximal camera displacement of 1.51 mm, which is clearly visible to the human eye (Figure 3.11(b)), the average discrepancy of the coordinates is 6.64 mm. This discrepancy is 19 times larger than for a camera displacement of 0.49 mm (Figure 3.11(b)). Also, the maximal discrepancy of 20 mm is 50 times larger than for a camera displacement of 0.49 mm (Figure 3.11(b)). The impact of blur is clear and is considered a significant problem, particularly considering the GSD of just 0.35 mm, which is $\frac{2}{3}$ the size



(a) Discrepancies for small camera displacements.



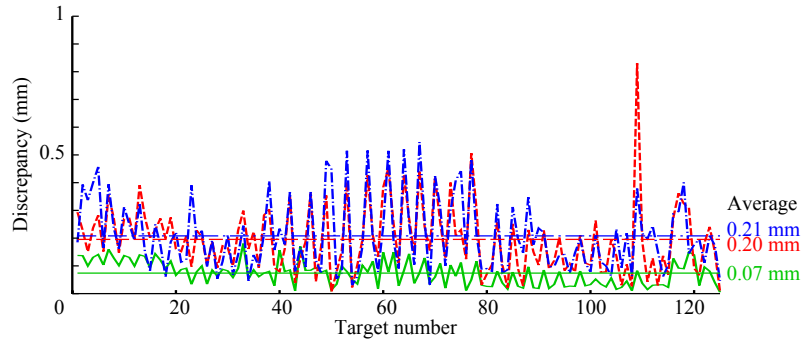
(b) Discrepancies for large camera displacements.

Figure 3.11: Discrepancies between blurred and sharp set for Nikon D80.

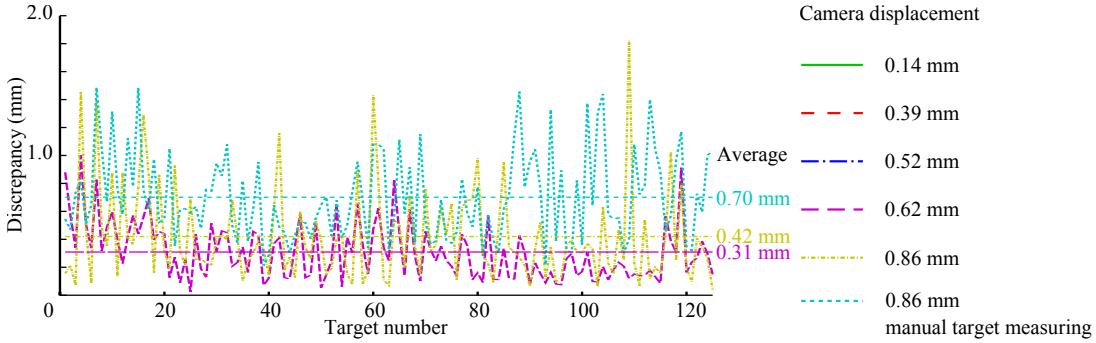
of the camera displacement. This test proves the deteriorating effect of image blur on the results of automated photogrammetric processing. It was found that the discrepancy between a sharp and manually processed image set was only 2.02 mm. This is three times smaller than for the semi-automated processed image set. The largest discrepancy with less than five millimetres was much smaller. This shows that automatic measurement is clearly inferior to manual measurement when visible blur is apparent.

Surprisingly, the results of the Nikon D7000 show similar results. With increasing camera displacement the discrepancy between the sharp and the blurred set increases (Figure 3.12). However, due to the smaller pixels on the sensor, similar displacements result in larger blur in the image (Figure 3.10). A displacement of around 0.5 mm has an average discrepancy of 0.11 mm for the D80 and 0.21 mm for the D7000 (Figure 3.11(a) and 3.12(a)).

It was also found that manual detection in the D7000 set produced inferior accuracies



(a) Discrepancies for small camera displacements.



(b) Discrepancies for large camera displacements.

Figure 3.12: Discrepancies between blurred and sharp set for Nikon D7000.

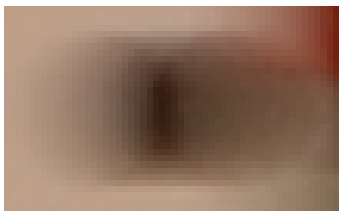


Figure 3.13: Blurred target of D7000 with 0.86 mm camera displacement.

than automatic detection. However, there were some subtle differences. The targets in this image are smeared, they do not create two separate, but two overlapping dots. The result is that the centre of the smeared target is significantly darker than the boundaries (Figure 3.13). This might enable target detection and measurement methods to find the centre of the target more accurately than a human operator. Methods for measuring the centre of a target remain effective because the gravity centre can be established precisely in the centre of the dark area.

Furthermore, it is noticeable that discrepancies of both sets, the D80 and the D7000 set, describe a zig-zag curve (Figure 3.11 and 3.13). As both datasets were created independently it is certain that the results are consistent. This could be caused by the detection method, which sometimes detects the smeared dot more to the right or more to the left. The side which is more often detected returns small discrepancies in the calculations. The targets detected more to the other side of the targets however, have larger discrepancies. It can be assumed that, if semi-automatic detection always measure the same side of a target, the discrepancies will be significantly smaller.

A camera displacement of 0.86 mm results in an average discrepancy of 0.42 mm compared to 0.2 mm GSD for the D7000; for the D80 a camera displacement of 0.99 mm results in an average discrepancy of 1.39 mm compared to a 0.4 mm ground sampling distance. In some applications an accuracy of 2 -3 pixels is acceptable. However, it can be assumed that larger camera-to-object distances, with the same baseline and camera displacement, will cause larger errors due to an increasingly smaller intersection angle. For UAV image flight is it also important to consider that all images contain a certain amount of motion blur due to the unstable camera platform. This means there is no perfectly sharp picture available and the calculated 3D coordinates are calculated using two blurred images. However, object coordinates are often calculated based on more than just two image coordinate measurements. This redundancy can help to suppress the influence of image blur and restore precise object coordinates. Nevertheless, it has been found that blurred images remain a significant problem for accurately detecting targets, which needs further investigation.

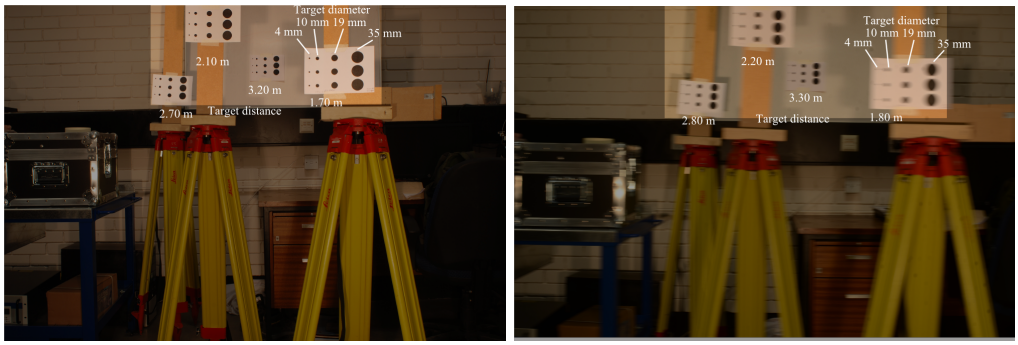
3.4 Dataset III - Influence of blur on target detection

While working with the two first datasets it was observed that the detection of targets in blurred images is the main difficulty posed to automatic processing. Fully automated detection of targets in blurred images fails even with a small amount of blur. Even semi-automatic detection, which requires operator intervention, is difficult and time consuming. The operator defines precisely a search mask around the proposed target area to ensure a detection. However, even if the detection appears successful it can be wrong and influence subsequent calculations. In extremely blurred images, where the camera displacement is larger than the target (Figure 3.9(d)), fully manual measurements appear to be the only method that ensures that measurements are made with reasonable precision.

3.4.1 Setup and composition of the dataset

The third dataset was acquired to determine the influence of blur on automatic target detection and to determine a threshold at which successful automatic detection fails. Therefore, circular targets of different sizes were positioned at different distances to the camera. The diameter of the targets were 0.4 mm, 1 mm, 1.9 mm and 3.5 mm. These targets were positioned in front of the camera with camera-to-object distances of 1.7 m, 2.1 m, 2.7 m and 3.2 m. Three targets of the same diameter were positioned at each distance, which resulted in a total number of 48 targets.

Two cameras were mounted on the shake table: a Nikon D80 with a 24 mm lens and a Nikon D7000 with a 28 mm lens. The higher camera resolution and focal length of the D7000 provided images with a higher geometric resolution than the D80 images. Both cameras were shaken at the same time to ensure exactly the same blur to both cameras. Due to the limited space on the table the D7000 needed to be fixed 100 mm further away from the targets. The focal length for the D80 was 25.97 mm and 29.31 mm for the D7000 (Table 3.5). All these factors, including the larger camera-to-object distance, different focal length and different pixel size were taken into account during subsequent analysis. The aperture was set for both cameras to $f/14$ with a film speed of 100 ISO



(a) Sharp image acquired with D80. (b) Blurred image acquired with D7000.

Figure 3.14: Setup of third dataset.

and a consequent exposure time of 1/5 s.

With this configuration, 135 images and 75 images were acquired with the D80 and D7000 respectively, all cameras experiencing displacements less than 2.41 mm. The discrepancy in the number of images was created by the need to manually trigger the D7000. Automatic camera triggering was possible for the D80 whilst manual camera operation was used for the D7000. This caused the difference in the number of exposures between cameras, with five times per shutter run for the D7000 compared to 10 times per run achieved with the D80. The shake table was set to a long period of constant velocity and both cameras were equipped with acceleration sensors. During post processing it was possible to exclude all D7000 images that were not acquired during a period of constant velocity (Section 2.2). Of the generated images, 14 images with different camera displacements were chosen for detailed processing from the D80 image set. The displacement for these images was between 0 mm to 2.42 mm. For the D7000, 15 images were chosen, with a camera displacement between 0 mm to 2.28 mm (Table 3.5).

3.4.2 Target detection in blurred images

The 48 targets used to conduct this test were chosen to have redundant information at different depth of the image. The pixel width of each target, at each distance was manually counted in the sharp image (Figure 3.14(a)). This included, the outmost left and right coordinate of the target which were also measured manually. These

Table 3.5: A list of parameters for the dataset to analyse the influence of blur on target detection.

	Dataset 3	
	Nikon D80	Nikon D7000
Camera to object distance	1.70 m;2.10 m;2.70 m;3.20 m	1.80 m;2.20 m;2.80 m;3.30 m
Size of target	4 mm; 10 mm; 19 mm; 35 mm	
Number of not coded targets	48	
Focal length	25.97 mm	29.31 mm
Aperture	$f/14$	
Number of camera displacements analysed	14	15

measurements were conducted by the same operator, on the same screen, with several breaks to rest their eyes so helping to ensure consistent measurement quality. Target width was calculated by using the difference between the measured left and the right coordinate and the average of the three targets in each size (four sizes) at each distance (four distances) determined (16 average values). Then it was calculated how large the targets should theoretically appear in the image. Equation 2.2 was applied, using target size in object space (t_O) focal length (c_k) and camera-to-object distance (h), generating the theoretical target size in image space (t_I), converted to pixels. This 'theoretical target size' was compared to the target size measured in the image. The discrepancies are represented in Table 3.6. For most targets these differences are around one pixel and it can be accounted for by spectral mixing and chromatic aberration. Chromatic aberration appears radial from the centre of a lens (Figure 3.15). To suppress the influence of chromatic aberration on the counting process, which is carried out in the horizontal direction in which the blur is being applied, targets were fixed in the centre top of the image (Figure 3.14). This causes, chromatic aberration to occur at the top and bottom of a target, but not on the left or right side of the target.

Automated target detection was then applied on the blurred images, to evaluate whether each target was detected successfully. For detection, the 'Automatic Target Marking' tool was used, which detects targets without any additional operator input. The detection tool began missing targets with a camera displacement of 0.59 mm (D80) and 0.42 mm (D7000). Detection of targets failed on all distances and target sizes with a camera displacement of 1.95 mm (D80) and 1.59 mm (D7000). Each image was analysed and the number of detected targets for each distance was evaluated. Unfortunately, the

(a) Comparison for targets in D80 image. Counted and (theoretical) size in pixels.

Target diameter (mm)	Camera to object distance (m)			
	1.7	2.1	2.7	3.2
4	$12.\overline{3}$ (10.02)	$9.\overline{6}$ (8.11)	$7.\overline{6}$ (6.31)	6 (5.32)
10	27 (25.05)	21 (20.28)	$15.\overline{6}$ (15.77)	$12.\overline{6}$ (13.31)
19	51 (47.59)	$39.\overline{6}$ (38.52)	29 (29.96)	$24.\overline{6}$ (25.28)
35	94 (87.67)	72 (70.97)	$51.\overline{6}$ (55.20)	44 (46.57)

(b) Comparison for targets in D7000 image. Counted and (theoretical) size in pixels.

Target diameter (mm)	Camera to object distance (m)			
	1.8	2.2	2.8	3.3
4	18 (14.67)	13 (11.88)	$11.\overline{6}$ (9.24)	$8.\overline{6}$ (7.80)
10	38 (36.68)	$29.\overline{6}$ (29.70)	23 (23.1)	$19.\overline{6}$ (19.49)
19	$72.\overline{6}$ (69.70)	58 (56.42)	$43.\overline{6}$ (43.89)	$36.\overline{6}$ (37.03)
35	133 (128.40)	104 (103.94)	$76.\overline{6}$ (80.84)	$66.\overline{6}$ (68.21)

Table 3.6: Comparison between counted and theoretical target size.

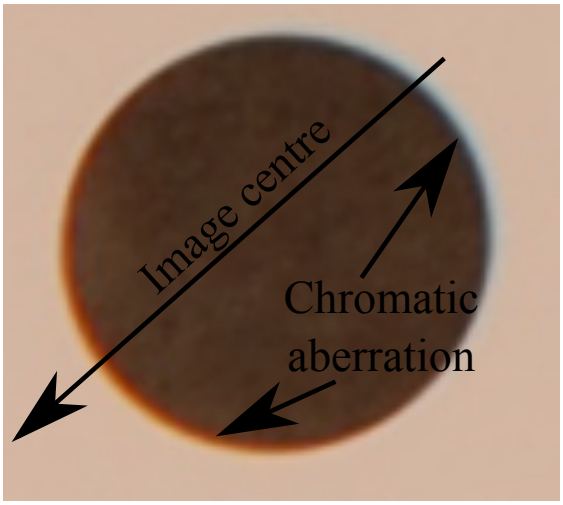
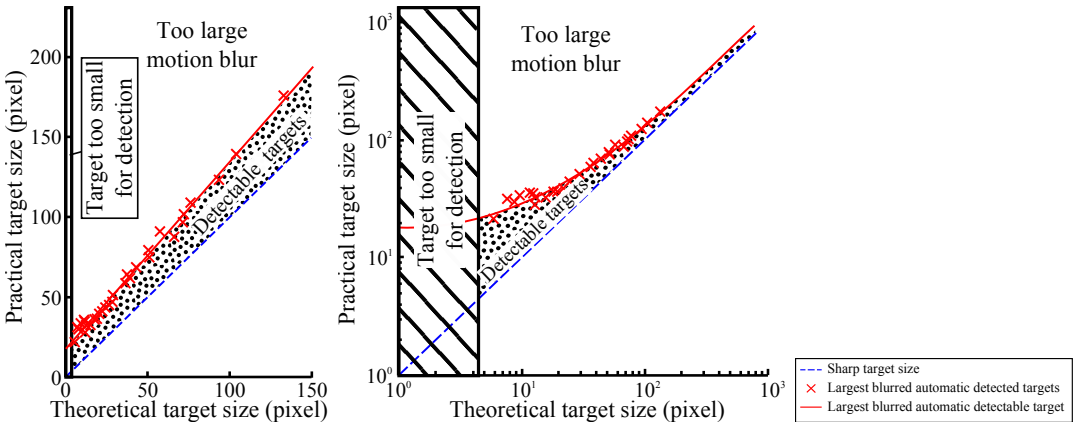


Figure 3.15: Chromatic aberration (blue and red) on black target. Inset of a 35 mm target with 1.7 m distance to the camera.

relationship between blur extent and successful target detection was only approximate. Only rarely were all targets of the same distance and size detected in one image but were then not detected in the subsequent more blurred image. Sometimes it even happened that a target was not detected in a less blurred image, but was detected in the following more blurred image (Section A.1). Due to the limited number of three targets, of one size, at one distance, it was decided that detection failed when all three targets were not detected any longer. The clear cut ensures that target detection would certainly fail for targets in real images at this size and distance. At this point the target pixel width was manually counted and was found to be larger than the original sharp target due to the motion blur.

3.4.3 Results of target detection in blurred images

Despite some indistinct patterns in performance, it was possible to find the threshold at which detection fails due to blur. By determining the practical pixel width in the blurred image and comparing it to the theoretical sharp target width it was possible to identify a linear dependency between sharp target size and blurred target sizes (Figure 3.16(a)). Figure 3.16 shows theoretical target size versus the size of targets when detection is no longer possible. The best-fitting linear dependency was established using least squares



(a) Linear axes scale. (b) Logarithmic axes scale.

Figure 3.16: Target size in a sharp image referenced to targets size in a blurred image when automatic detection failed. ($R^2 = 0.9914$)

matching and then used to formulate an equation that describes the degree to which a target can be blurred before automatic detection is unsuccessful. A theoretical target size in pixels (t_s) can be determined using information about the target size in object space (t_O), focal length (c_k), distance between the camera and target (h) and sensor pixel size px. Furthermore, the blurred target size (t_b) can be related linearly to its theoretical equivalent t_s:

$$t_s = \frac{t_O c_k}{p_s h}$$
$$t_b \leq 1.166 \cdot t_s + 16.794$$

$$\begin{aligned} t_s &\ldots \text{Theoretical target size in pixel} \\ t_b &\ldots \text{Blurred target size in pixel} \\ t_O &\ldots \text{Target size in object space} \\ c_k &\ldots \text{Focal length} \\ h &\ldots \text{Camera to object distance} \\ p_s &\ldots \text{Sensor pixel size} \end{aligned} \tag{3.4}$$

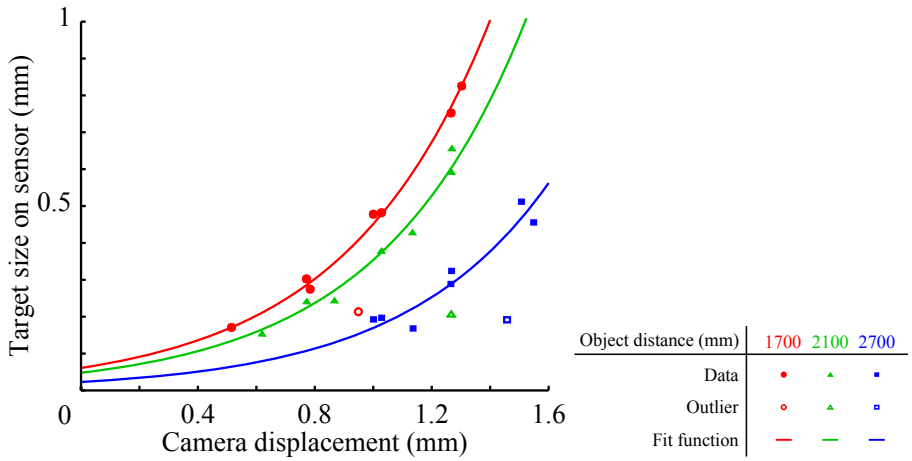


Figure 3.17: Size of blurred targets related to camera displacement.

From the tests conducted, targets which have a theoretical width of t_s, and appear in the image with a width between t_s and t_b, can be automatically detected. In Figure 3.16(b) it is possible to see that the ratio between the theoretical and blurred target sizes decreases for larger targets, which implies that there is a greater tolerance to the blurring of small targets than large targets. This is an important finding but can be practically explained by the circularity threshold used in the detection algorithm. This threshold is based on the roundness of a target and how many pixels are part of a round target. For large targets a small camera displacement will increase the percentage of pixels that do not support a round target and it is not accepted as a target. However, for a small target the percentage of pixels that do not support a round target is with the same displacement much smaller so that it is still accepted as round.

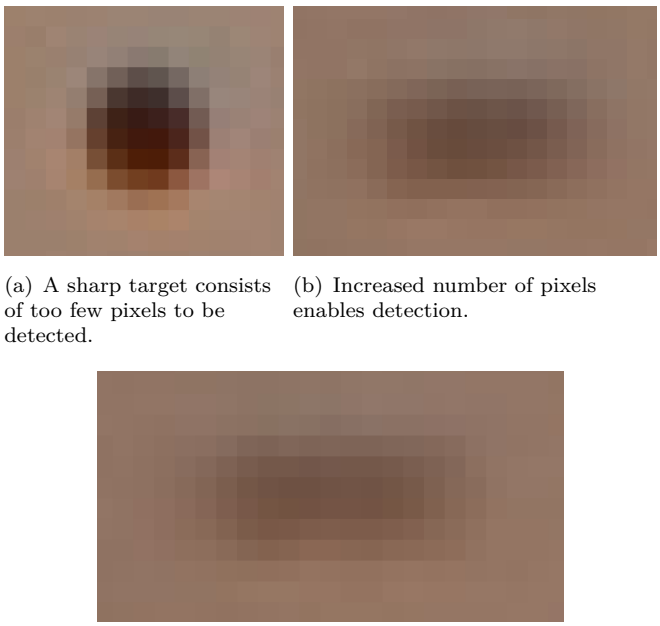
The size of blur is a direct result of camera displacement. Larger displacements cause targets to appear more smeared. Figure 3.16 illustrates how much displacement is required for target detection to fail, and how wide the target appears in the image. Comparing camera displacement with the size of the blurred targets proves that large targets can be absolute blurred more than small targets before detection fails. This means that they can experience greater camera displacements before detection fails. Figure 3.17 shows how target size, camera displacement and camera-to-object distance are related.

The dependency between camera displacement and successful target detection is exponential, which confirms that smaller targets tolerate more relative blur than larger ones, which means that relative to their small size they can be blurred more than large targets. A target with a theoretical width of 50 pixels can be detected until it becomes so blurred that its width increases to 75 pixels, suggesting that a 50 % blurring can be tolerated. However, a target measuring 100 pixels can be smeared to 133 pixels before detection fails, which is only an increase of just 33 %. These examples suggest that the 50-pixel target can only suffer a displacement of 25 pixels before detection is unsuccessful, whilst the larger 100-pixel target can resist detection failure due to blur up to 33 pixels (Figure 3.16). This represents a difference of 8 pixels of additional camera displacement that can be tolerated if larger targets are used (Figure 3.17). It also shows that an increasing object-to-camera distance results in a flatter exponential function. The outliers, which do not support the exponential function, represents the smallest targets with a size of 4mm. It would appear that the detection algorithm does not detect very small targets because they have too few pixels in the image to be recognised as a target (Figure 3.18(a)). However, once blurred they are represented by enough pixels to be accepted as a target (Figure 3.18(b)). Notwithstanding this, as the target becomes even more blurred, detection fails again because the target shape becomes too elliptical and does not appear as a circle (Figure 3.18(c)).

The exponential function (Equation 3.5) best represents the dependency between the camera-to-object distance, displacement and blurred target size (Figure 3.17). This function makes it possible to calculate the blurred target size (t_b) as dependent on the camera-to-object distance (h) and the camera displacement (d) for short camera-to-object distances:

$$t_b = 70000 \cdot h^{-1.85} \cdot e^{2d} \tag{3.5}$$

Equation 3.5 is a based on a least squares matching ($R^2 = 0.9815$) and has not been investigated for distances much greater than 3 m and should be treated with caution for the increased camera-to-object distances that may be more common with UAVs. The displacement d is the result of the camera velocity during exposure time (Equation



(a) A sharp target consists of too few pixels to be detected.

(b) Increased number of pixels enables detection.

(c) Too much blur do be detected.

Figure 3.18: Influence of blur on small targets.

3.6). Forward-motion displacement can be calculated using the exposure time and UAV velocity. In the case of rotations, the calculation depends on the position of the rotational axes. If the origin of the axes is coincident with the camera's perspective centre, it is possible to compute equation 3.6. As opposed to the roll ω and pitch ϕ, which only depend on the flight altitude, the yaw κ depends on the distance between the nadir position and the object s for which the displacement d is calculated:

$$\begin{aligned} d &= v_t \cdot ex \\ d &= h \cdot \tan v_{\omega/\phi} \cdot ex \\ d_\kappa &= s \cdot \tan v_\kappa \cdot ex \end{aligned} \tag{3.6}$$

where: d_κ is the displacement in yaw; $v_{t/\omega/\phi/\kappa}$ are the respective velocities in translation, roll, pitch and yaw; and ex is the exposure time.

Connecting equations 3.4 and 3.5 makes it possible to calculate the minimum target size

t_0 that should be used in a UAV image flight. To calculate it the following parameters are required: object distances h known a priori; focal length c_k; sensor pixel size px; and camera displacements d (Equation 3.7). These are normally already part of the flight planning process before every UAV image flight:

$$t_0 = \frac{70000 \cdot h^{-1.85} \cdot e^{2d} \cdot h - 16.794 \cdot h \cdot px}{1.166 \cdot c_k} \tag{3.7}$$

However, the inverse calculation to determine the displacement based on the size of a blurred target is not always valid. It would only be valid when the blurred target is on the threshold between successful and unsuccessful detection because this equation is based on the maximum detectable size of blurred targets. If a less blurred target is used in this calculation, the real camera displacement will be larger than the calculated one.

3.5 Dataset IV - Influence of blur on feature matching

The previous tests have demonstrated that photogrammetric target detection is influenced by blur. However, use of photogrammetric targets is normally restricted to establishing a coordinate system for a survey. The production of a dense digital surface model (DSM) never uses targets but well defined natural features are identified using automatic feature detection and referencing algorithms. These algorithms search for feature points, which are identifiable in other images. After detecting features in two or more images, the algorithm references the best match. Measuring 2D image coordinates for each feature in all images provides the possibility to generate a dense DSM. As the previous datasets have indicated, detecting targets is difficult with blurred images and it could be assumed that feature detection and matching method are also influenced by blur. This final test dataset has been used to assess the extent that motion blur affects detection and matching of natural features.

3.5.1 Setup and composition of dataset

The images for this dataset were taken from the camera calibration dataset (Section 3.2) using a subset of four images with 0 - 1.03 mm camera displacement. One sharp image was used, and detected feature points were referenced in the blurred images. Table 3.7 summarises the key characteristics of the image used.

Table 3.7: A list of parameters for the feature detection dataset.

	Dataset 4
Camera to object distance	1.60 m
Focal length	25.35 mm
Aperture	$f/8$
Frames per process	1 blurred + 1 sharp image
Number of camera displacements	4

3.5.2 Feature detection in blurred images

The first step in data processing involves feature detection achieved in each of the four blurred images and in the sharp image. Feature detection was carried out using OpenCV's '*SurfFeatureDetector*'-function which is the implementation of Bay et al. (2006) (OpenCV Dev Team, 2014). SURF is a well-established method that has been used for several years and is broadly accepted by the community. In a subsequent step the detected features were matched between the sharp and each of the blurred images, using OpenCV's '*BFMatcher*'. The function simply attempts to match every feature of one image against every feature of the other image, accepting the 'optimum' result (OpenCV Dev Team, 2014). Code 3.1 shows the program code implementation required for feature detection and matching.

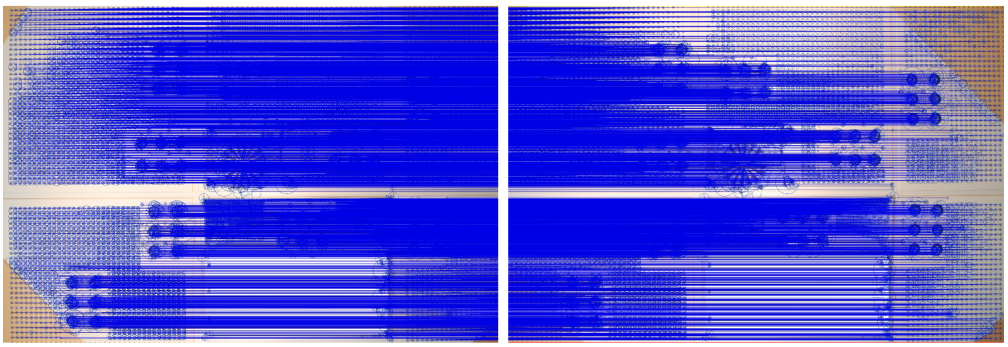
Listing 3.1: Feature detection and matching

```
1 // minHessian is a threshold to define if a feature is accepted or not.
2 int minHessian=400;
3 // Setting up feature detection method.
4 SurfFeatureDetector detector(minHessian);
5 // Definition of a vector to store features.
6 vector<KeyPoint> feature_blurred, feature_sharp;
```

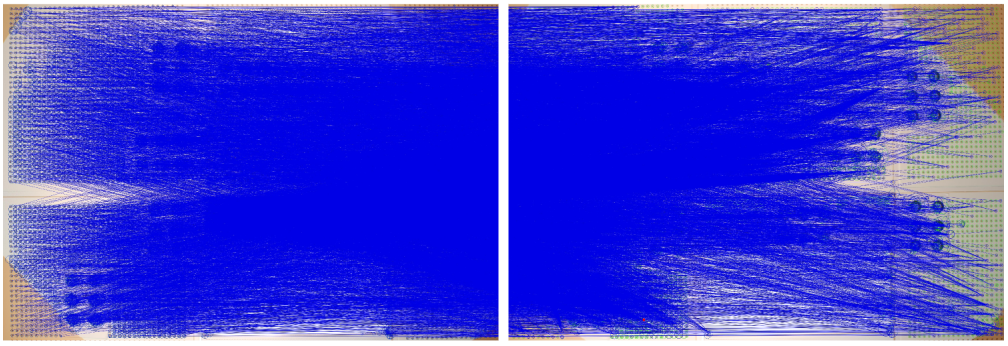
```
7  // Detection of feature points in a sharp and blurred image.
8  detector.detect(image_sharp, feature_sharp);
9  detector.detect(image_blurred, feature_blurred);
10 // Wrapper to make features accessible for BFMatcher.
11 SurfDescriptorExtractor extractor;
12 // Definition of matrix to store extracted information.
13 Mat descriptors_blurred, descriptors_sharp;
14 // Extraction of features to make them accessible for matching.
15 extractor.compute(image_sharp, feature_sharp, descriptors_sharp);
16 extractor.compute(image_blurred, feature_blurred, descriptors_blurred);
17 // Definition of vector to store matches.
18 vector<DMatch> matches;
19 // Setting up matching method.
20 BFMatcher matcher(NORM_L2SQR);
21 // Matching features between blurred and sharp image and storing the matches.
22 matcher.match(descriptors_sharp, descriptors_blurred, matches);
```

The basic algorithm will match all features, and therefore features in the destination image can be matched with multiple features in the original image. To filter out any incorrect matches, various filters need to be applied. These filters can be based on knowledge about the expected approximate position, orientation and scale of the features. For the laboratory dataset this information is available. The images are taken from a similar position with the same orientation of the camera. Hence, image coordinate, rotation and scale of the feature should be similar, with only small variations in position rotation and scale, due to the camera displacement process.

Figure 3.19 shows an example of two images with detected and matched feature points. Figure 3.19(a) shows two sharp images and correct matched features. Figure 3.19(b) demonstrates that multiple targets on the original image (left) are matched to a few features in the target image (right). This is shown by the divergent blue lines that connect origin and target features (green). These matches are incorrect because both images are acquired from the same position and with the same rotation and scale so the connection lines should be parallel (Figure 3.19(a)).



(a) Matching sharp and sharp image.



(b) Matching sharp and blurred image.

Figure 3.19: Matching (blue) feature points (green) between two images.

3.5.3 Results of feature detection in blurred images

A high number of feature points are detected in the sharp image but with increased blur there is an obvious decrease of 80 % in detected feature points. Matching these targets and filtering out incorrect matches results in a significantly decreased number of accepted targets (Table 3.8).

Table 3.8: Camera displacement vs. detected and correctly referenced feature points.

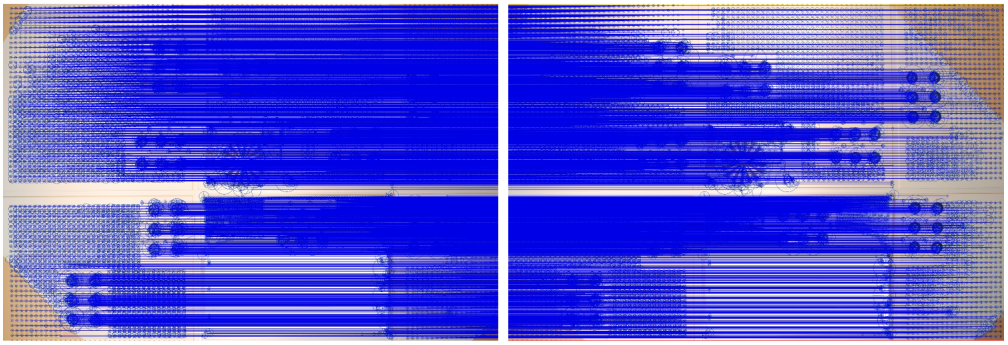
Camera displacement (mm)	Detected features (% of features of sharp image)	Correctly referenced features (% of detected features)
0	12214 (100 %)	12214 (100 %)
0.38	8847 (72 %)	1524 (17 %)
0.53	7370 (60 %)	224 (3 %)
1.03	2645 (22 %)	47 (2 %)

For real UAV images the parameters to assist the filtering process can be estimated using GNSS and IMU sensors, which may provide data with an appropriate level of accuracy. It would then be possible to estimate the image position of the blurred image, relative to the position of the sharper image.

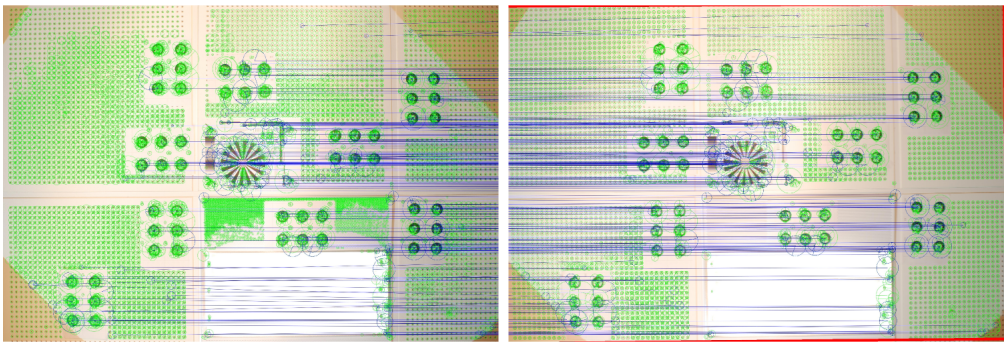
Figure 3.20 presents the results of feature matching for each level of blur after filtering of incorrect results. It is possible to see that the number of features (green) decreased significantly with increased blur and the number of correct matches is reduced. The number of correct matches would still be enough to calculate parameters for a perspective image transformation, but not sufficient for a dense DSM. Furthermore, the laboratory images have an overlap of nearly 100 %. Aerial images acquired by a UAV often provide a much smaller image overlap, this means there is a further reduction in the number of correctly matched features. The significant reduction of detected feature points and the problematic matching of these points proves that blur has a significant influence on feature detection and matching. It was found that minor blur, which is invisible to the naked eye, still disturbs this process.

3.6 Discussion

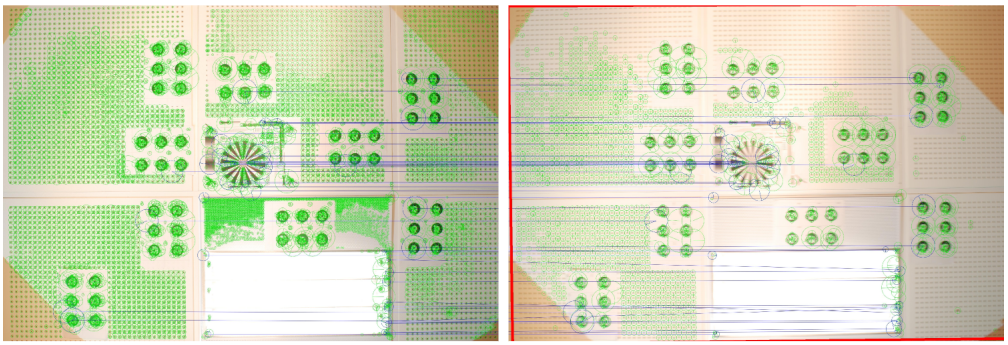
This chapter demonstrates some of the challenges associated with working with blurred images. Automatic processing procedures used in photogrammetry are clearly influenced by small motion blur and not only does the number of random feature points decreases significantly with blur but the number of detected signalised targets decreases also. Even images which appear sharp visually can cause difficulties if a small amount of blur is present. The number of measured targets directly influences the accuracy and ability of subsequent calculations, including camera calibration and 3D coordinate calculations. The detection of targets becomes more reliable with larger targets. These occupy more pixels which helps to accurately calculate the target centre. Furthermore, the same amount of blur causes a single small target to blur laterally, causing misdetection or multiple detection. However, targets sizes cannot become too large as very large targets are impractical for efficient photogrammetric field work.



(a) Filtered matches between sharp and sharp image.



(b) Filtered matches between sharp and blurred image (0.53 mm camera displacement).



(c) Filtered matches between sharp and blurred image (1.03 mm camera displacement).

Figure 3.20: Filtered (blue) feature points (green) between two images.

If automatic target detection is not possible, semi-automatic detection of targets using small ROIs can be successful. However, this is time consuming as these detection areas have to be defined both manually and accurately. Even with well-defined detection areas, detection procedures remain susceptible to increasing blur. Manual measurement of targets always remains possible, but the results are not as accurate as automatic measurements achieved using sharp images.

A limitation of the detailed tests conducted in this study has been the focus on close range images only. Care needs to be exercised in assessing the implications of these findings for UAV imagery, where flight altitudes are much larger than the distances used in these experiments, particularly where the camera displacements are larger due to the flight velocity. The effect is that small changes in one coordinate measurement due to blur will have correspondingly a larger effect on the calculated 3D coordinates. Furthermore, for the laboratory test only one blurred and one sharp image were used, while for a real UAV dataset it can be assumed that multiple images are available. However, for UAV images it is unlikely that they are absolutely sharp so that the image coordinate measurements are imprecise and would negatively influence subsequent calculations.

The human hand naturally introduce shake or jitter, with frequencies of 2 - 10 Hz and amplitudes of up to 1 mm. This creates the certainty that some blur is always present in hand held images as reported by Stiles (1976). Sachs et al. (2006) also reports on a 'drift of the hand' of up to 5 mm / s, which increases low levels of blur in hand held images even further. Based on the findings made here with much smaller camera displacements, it can be assumed that the push of a shutter release button on a hand held camera can cause significant blur in the image. This supports the commonly known rule across the photographer community to suppress image blur, which states that it is necessary to use shutter speeds faster than 1/focal length in seconds or at least 1/60 s (Emling, 2008). For UAVs the shutter speed is chosen significantly faster as they are acquired outside in daylight with shutter speeds of 1/250 s and faster.

Furthermore, this chapter supports findings that have been made using subjective human experiences, such as the work by Johnson and Casson (1995) that proved that blur influences acuity. The work described in this chapter has shown how blur influences photogrammetric image processing and that in comparison to human perception, image processing is more sensitive to blur. Also, it was shown that the detection of blurred simple structures, such as round targets, caused problems for the automated processes. Colombo et al. (1987) tested the ability of humans to detect structures in blurred images and found there was a decreasing ability for subjects to read text as blur increased. However, the legibility of text is a much more complicated task than the detection of round targets.

The results of this work support the desirability of excluding blurred images from photogrammetric processing. Gülch (2012) recommends elimination of blurred images as a first step in UAV image processing an issue resolved and explored upon in Chapter 4. The findings also support the work of Shah and Schickler (2012) who develop blur correction methods specifically for UAV applications. Lelégard et al. (2012) states that a blur larger than 2 pixels is a significant amount. However, the findings reported in this thesis suggest that a blur of just 2 pixels is actually too small to influence the detection, identification, referencing and measurement of targets.

3.7 Conclusion - Does blur disturb?

A range of difficulties are caused by image degradation in photogrammetry. Small camera displacements create motion blur and have a significant impact on the accuracy of subsequent calculations and processes. Even activating a shutter button on a hand held camera can result in significant camera movement that causes image blur. Fully-automated detection of targets in images that contain low levels of blur can be difficult and can influence further processing. This problem can be solved by using semi-automatic detection tools. A small amount of blur has no significant influence on calculation results, when targets are detected and measured successfully. However, if blur increases to such a degree that semi-automatic detection requires significant operator attention, subsequent calculation will return significantly inaccurate results. In these cases manual measurements can be carried out and still provide results with an acceptable accuracy. This is even possible in highly blurred images but manual intervention is very time consuming and prone to error. It is important to recognise that most of these tests that were conducted used signalised targets. When natural feature points were used fewer features were detected and referenced due to the influence of motion blur. The influence of blur upon natural features in subsequent processing can be assumed similar to those results achieved with signalised targets. Although blur might disturb most

image processing procedures, it can also be exploited for some applications, as identified by Boracchi (2009) and McCarthy et al. (2013). Boracchi (2009) uses blur to detect the motion of ping-pong balls, whilst McCarthy et al. (2013) uses blur captured during long exposures to detect the movement of structures subject to dynamic forces.

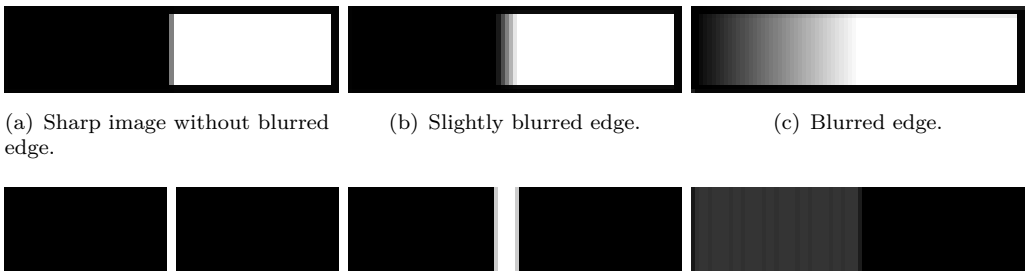
4 Blur detection

The previous chapter demonstrated that blur is a problem in photogrammetric image processing and that the detection of blurred images might be a useful first step to help solve the associated problems of blur. Photogrammetry in image processing may often involve the use of hundreds of images, collected during a UAV image flight. However, UAVs do not provide a stable platform which can result in blurred images in the datasets. The manual detection and exclusion of blurred images is an intensive, time consuming step that requires a human operator and is therefore prone to error. This chapter will assess current blur detection methods that are used in the detection of blurred images and then present a novel algorithm that was developed to detect and isolate blurred images.

4.1 Literature review

The previous chapters explained that blur prevention and hardware modification often provide information for deblurring images using image processing (Section 2.1.2.3). Before attempting to deblur an image it is useful to identify whether or not an image is blurred. Many studies do not include this preparatory step and only provide blurred images, which are used to test deblurring algorithms (Section 2.2).

A widely used approach in image processing is blurring an image to smooth it and make it more appealing from a purely visual perspective (OpenCV Dev Team, 2014). This can be done with image filters like a Gaussian filter or a Median filter and gives the viewer an impression of movement or draws their attention to specific objects (Nik Software, 2013). Detecting whether an image is blurred or sharp remains a complicated process



(a) Sharp image without blurred edge. (b) Slightly blurred edge. (c) Blurred edge.

(d) Edge detection result shows sharp edge. (e) Edge detection result shows a step gradient. (f) Edge detection result shows a flat gradient.

Figure 4.1: Example dataset with sharp and blurred edge. A high contrast between black and white indicates a sharp image a low gradient a blurred image.

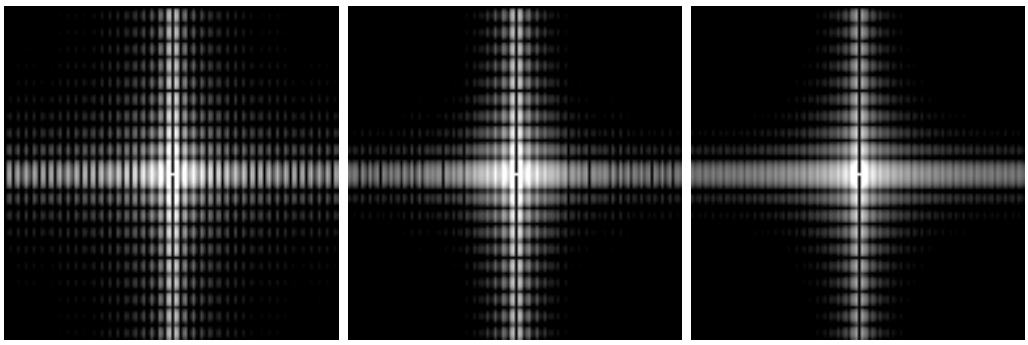
and has not been completely solved. There are a range of different approaches available that are used to detect blur. It is important to distinguish between blur detection approaches that necessitate additional data, such as another image or information from other sensors. Methods that do not use additional data are called 'no-reference blur estimation' (Crete et al., 2007). Many different approaches exist in this field, which deal with the question: 'What is blur and does it manifest itself in an image?' This question can be answered using two main methods that are used in 'no-reference blur estimation'. The first method detects blur based on edge detection and the second detects blur based on frequency analysis.

4.1.1 Blur detection based on edge detection

Edge detection is a widely used method to detect blur (Joshi et al., 2008; Narvekar and Karam, 2009; Ong et al., 2003). An edge in an image can be considered as a grey value difference between neighbouring pixels. Edge detection calculates the gradient between neighbouring pixels (Section 2.1.1.3), in sharp images this contrast is abrupt between two contrasting grey levels (Figure 4.1(a)) and edge detection would return a well-defined result for the edge (Figure 4.1(d)). With increasing blur the contrast decreases and becomes flatter (Figure 4.1(b)) and the edge detection result returns a flatter gradient over a larger area (Figure 4.1(e)). In case of largely blurred edges with very flat gradients (Figure 4.1(c)) edge detection returns barely visible results, or even invisible results due to this gradient being too flat (Figure 4.1(f)).

A method for blur detection which was developed by Ong et al. (2003) involves four steps. The first step is the calculation of the direction of the gradient. This describes in which direction the edge is orientated and on which sides of the edge the brighter and the darker values are located. The second step is edge detection itself. This is carried out with a Canny Edge detection (Ong et al., 2003), which returns the edge pixel but not the gradient between the neighbouring pixels. This is followed by the calculation of the edge-spread, which derives an actual representation of the blur. This is done by counting the pixels in both directions of the gradient from the position of the local extreme, downwards as long as the gradient pixels decrease and upwards as long as their intensity increases. The total number represents the edge-spread. The final stage involves determining a measure of image-quality. This value is calculated using the average edge-spread value of all the edges in the image and an additional parameter based on subjective ratings, derived from a small group of human subjects (Ong et al., 2003). The topic approach of Ong et al. (2003) describes how a blurring algorithm works, based on edge detection. Most algorithms are similar, but include more advanced techniques. A weakness of the Ong et al. (2003) method is that edge-spread is based on all edges, independent of their orientation. Edges oriented in the direction of blur can influence the result, even if they do not contain any useful information. The method by Narvekar and Karam (2009) calculates the number of edge pixels in sub-images, derived from a small part of the image. After counting the number of edges in the sub-image a decision is made whether or not further processing of the sub-image is required, or if insufficient information is available.

Just noticeable blur (JNB) is another method to determine if an edge is blurred or not (Ferzli and Karam, 2009). The JNB is based on just-noticeable distortion (Jayant et al., 1993) known as just noticeable difference (JND) (Ferzli and Karam, 2009). JND is defined as the minimal amount of intensity that has to be changed, in order to be noticeable by a sensor, relative to a background intensity (Jayant et al., 1993). The required difference can also be described in terms of contrast. The JNB is the minimum amount of perceived blur around an edge, given a contrast higher than the JND (Ferzli and Karam, 2009). There are several other methods for blur detection based on edges,



(a) Fourier transformation result for sharp image shows many high frequencies.

(b) Fourier transformation result for minor blurred image shows less high frequencies then (a).

(c) Fourier transformation result for image suffering from major blur shows no high frequencies.

Figure 4.2: Example of fast Fourier transformation on dataset with sharp and blurred edge shows influence of blur on frequencies. Each of the figures is point symmetric in the centre.

but they are either similar to the previously presented method, for specific special cases, or do not return sufficient results.

4.1.2 Blur detection based on frequency analysis

An alternative approach for blur detection which does not use edges involves analysing the image in the frequency domain (Rahtu et al., 2012). An image can be represented as a 2D function and described by its frequencies (Section 2.1.1.3). High frequencies do not appear in blurred images (Liu et al., 2008). The more an image is blurred the less high frequencies are present. Figure 4.2 shows the results of a fast Fourier transformation (FFT) derived from the image conveyed in Figure 4.1(a), 4.1(b) and 4.1(c). The centre is the origin of a coordinate system with lowest frequencies located in the centre and higher frequencies appearing at the boundaries. Figure 4.2(a) demonstrates that more high frequencies are present than in Figure 4.2(c), which is based on the blurred example (Figure 4.1(c)) and this absence of high frequencies can be used to detect blurred images.

Rahtu et al. (2012) uses the FFT to develop a blur resistant image analysis algorithm. After transforming the image from the spatial to the frequency domain, it is possible to divide the image into two components: magnitude and phase. Blur affects both

magnitude and phase. Using the assumption that the frequency domain is also influenced by textures and that the texture is not similar for the whole image, the analysis is only carried out for small patches of the image, to minimize the disturbing effect of textures.

The main weakness of a FFT approach is the computation time to convert the image from the spatial to the frequency domain and back again, which is significant longer than applying image filter. It was found that the transformation took about 2 times longer than applying standard image filters. Also, the analysis of the frequency domain is more complicated than the spatial domain, due to the radial structure of a frequency domain image. This requires transformations from Cartesian to radial coordinates and an offset of the origin from the upper left image corner to the image centre. Another problem is the visual interpretation of frequency images; their abstract nature requires a trained interpreter.

Blur detection based on the frequency domain of an image is based on analysing the number and magnitude of high frequencies in the frequency domain. The approach to analyse the frequencies differs for the various methods available.

4.1.3 Blur detection based on various other methods

There are a range of other blur detection methods, which may combine aspects of the edge or frequency approaches.

Edge sharpness detection A method developed by Crete et al. (2007) is based on the human perception of blur. This recognises that humans find it difficult to perceive differences between blurred and re-blurred images, but find it easy to distinguish between a sharp image that has been re-blurred. Figure 4.3 shows re-blurred images of the sharp examples Figure 4.1. The difference between the sharp image Figure 4.1(a) and the re-blurred image Figure 4.3(a) is easy to detect by human eyes. However, when comparing Figure 4.1(b) and the extremely blurred image Figure 4.1(c) to their re-blurred images Figure 4.3(b) and 4.3(c) the small differences are hard to perceive by the human eye. The differences between the blurred and re-blurred image are not as

(a) Sharp image re-blurred. (b) Slightly blurred image re-blurred. (c) Blurred image re-blurred.

Figure 4.3: Example showing re-blurred images.

visible, as between a sharp and a blurred image. However, this theory can still be used to detect blurred images.

Crete et al. (2007) uses the initial image and re-blurs this with a strong low-pass filter and consequently analyses the variation of the neighbouring pixels for both images separately. If variations between the original and the re-blurred image appear high, the original image can be considered sharp. If there is only a small difference in the variation, this suggests that the original image was already blurred. The algorithm involves six well defined steps.

1. Blur the image vertically and separately blur the image horizontally.

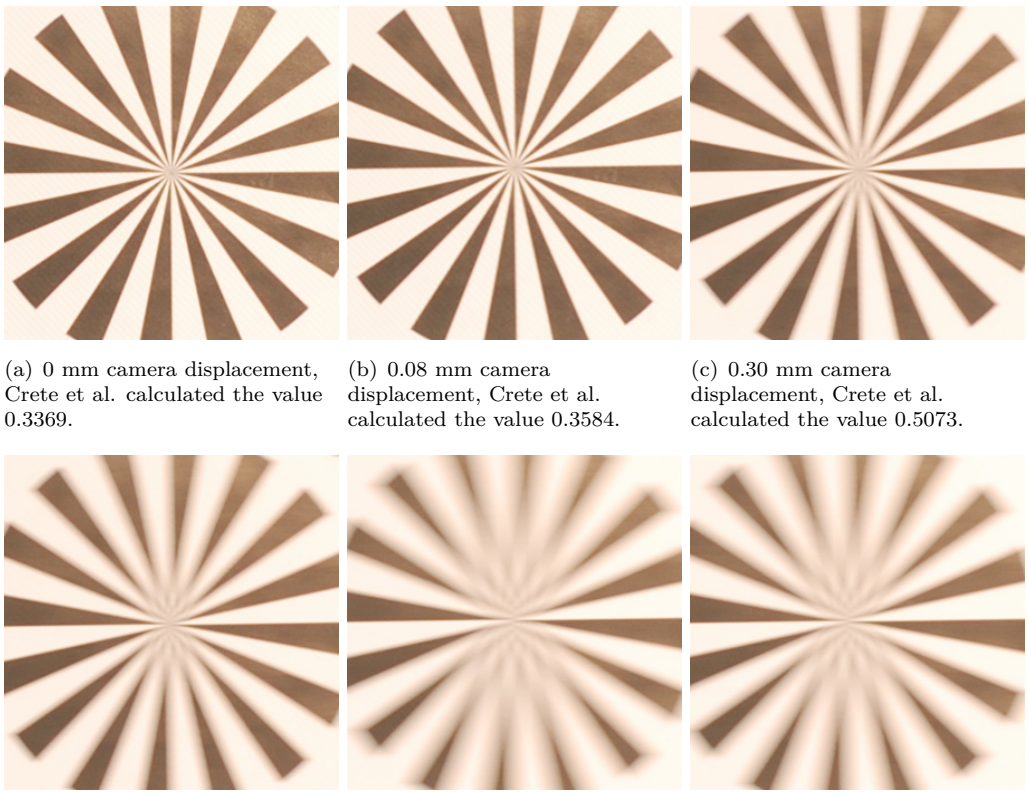
2. Compute the absolute variation between pixels both vertically and horizontally from the original. Compute the absolute grey value variation for the rows of the vertically re-blurred image. Compute the absolute grey value variation for the columns of the horizontally re-blurred image.

3. Calculate the difference between the vertical variation images and separately for the horizontal variation images.

4. Summation of all pixel values for both original variation images and the calculated difference images from step three.

5. Normalize the results retrieved in step four.

6. Select either the vertical, or the horizontal value as the blur value (dependent on which one is larger).



(a) 0 mm camera displacement, Crete et al. calculated the value 0.3369.

(b) 0.08 mm camera displacement, Crete et al. calculated the value 0.3584.

(c) 0.30 mm camera displacement, Crete et al. calculated the value 0.5073.

(d) 0.53 mm camera displacement, Crete et al. calculated the value 0.5442.

(e) 0.91 mm camera displacement, Crete et al. calculated the value 0.4845.

(f) 1.03 mm camera displacement, Crete et al. calculated the value 0.4485.

Figure 4.4: Test results of Crete et al. (2007).

As a last step, the authors evaluate the algorithm with a human test to match the computed blur value with a human related perception, the mean opinion score (MOS). The results show that MOS correlates with the computed blur of the algorithm. The algorithm has been made available online from Bao (2009) and was tested with images of the shake table dataset. The algorithm was tested using images from the test dataset with images corrupted by well-defined, one directional motion blur but the results were disappointing. The most blurred image (Figure 4.4(f)) was classified as less blurred than an image with three times less motion blur (Figure 4.4(d)). The extent of blur is particularly visible in the vicinity of the Siemens-Star (Figure 4.4). Beside the calculation time of several seconds, it was found that the algorithm returns unreliable results. Both represent serious problems when using huge image datasets, hence this method was believed to be unsuitable for blur detection in UAV image datasets.

4.1.4 Summary

Most of the reviewed methods in this section are computing intensive, for example the method proposed by Crete et al. (2007) and programmed by Bao (2009) took several seconds up to a minute to process one image. This currently renders these methods impractical and hence unusable for blur detection in UAV datasets containing hundreds of images. The tests identified in the literature, are very often undertaken on small image sets that have a wide range of backgrounds and a range of different blur sizes. Often this blur is generated mathematically with image filters or exhibits out-of-focus blur and not just motion blur. The results are often very subjective as they rely upon human perception. It has been shown that a subjective evaluation of image sharpness and blur is very dependent on the subjective observer (Sieberth et al., 2014). If the image content varies, the evaluation of blur becomes difficult, especially if there is no image for comparison. However, during a literature search it was found that most research focuses on prevention of blur or blur correction. Blur detection is rarely considered an independent topic.

4.2 Detecting Blurred Images

Deriving a statistic to represent the degree of blur in multiple images is rarely carried out independently, although some measures are used in blur prevention and correction for single images. Without quantification of blur on multiple images, the definition of a threshold value and the automatic exclusion of blurred images from datasets is impossible. Hence, an operator is required to manually identify blurred images and exclude them. Human detection and quantification of blur is dependent on the operator and is also prone to error. An automatic detection algorithm is therefore required, which can quantify blur in large image datasets and make them comparable to one another. The next section in this chapter will discuss the development of such an algorithm.

(a) Image for comparison.

(b) In comparison to (a) does it appear sharper.

(c) In comparison to (a) and (b) does this image appear sharpest.

Figure 4.5: Comparing of insets of UAV images.

4.2.1 The idea of the algorithm

An algorithm for blur detection was developed in this research. The algorithm is related to the edge sharpness detection algorithm developed by Crete et al. (2007) but significantly extends this and utilises concepts based on human perception. As previously mentioned, a person can best detect a blurred image when it is being compared to another image (Figure 4.5). If just Figure 4.5(a) is judged without using any comparison then is difficult to identify if the image is sharp or blurred. When compared to Figure 4.5(b), Figure 4.5(a) appears to be blurred and Figure 4.5(b) would be judged as sharp image. However, comparing Figure 4.5(c) to Figure 4.5(b) reveals that Figure 4.5(b) is actually blurred too. To enable a precise judgement the compared image should show the same area or parts of the same area to make judgement of blurred or sharp image possible.

Instead of using different images it is also possible to use one input image and blur it synthetically. The aim of the synthetic blur is to generate an image for comparison. Figure 4.3 demonstrates that it is possible for a human operator to differentiate between the sharp input image and the synthesised blurred image. However, if the input image is already blurred the differentiation is more difficult or may be incorrect. A stronger synthetic blur has to be applied to enable a human operator to differentiate between the blurred and re-blurred image. The differences between the blurred and the re-blurred image are an important support for a human to enable accurate visual blur detection. The approach of using differences between an image and a re-blurred image can be realised in an automatic algorithm and can be used to quantify blur in images (Section

4.1.3). However, realising this in an automatic algorithm requires more detailed processing steps (Section 4.2.2).

4.2.2 The algorithm

The blur detection algorithm aims to detect blurred images in UAV image datasets. A UAV dataset consists mostly of images with a similar texture and colour, typically representing fields, woods etc.. The two requirements of the algorithm should be that it can process the dataset quickly and that it can detect blurred images reliably.

Input The input to the program is a folder containing all images in a dataset. This folder can be specified by a folder path and accessed automatically. Each image in the folder is available and calculations conducted, and clearly the files must contain raster graphics so that the algorithm can process them. Beside the images themselves, no additional information is required, although, an output directory should be provided, to store results. With an input and output directory, all parameters are defined to run the program.

Flow of the program With the input directory defined, each image can be processed by the developed algorithm. The aim of the algorithm is to transfer the 'human detection of blur' into a computer algorithm. As presented, humans often use a second image for comparison to establish whether an image is blurred or not (Section 4.1.3). Figure 4.6 shows the basic steps of the program. These include:

1. Scaling down of the image resolution (Figure 4.7(b), Section 4.2.2.1).

2. Convert image to HSV colour space - comprise of a saturation, value, blue (SVB) image (Figure 4.7(c), Section 4.2.2.2).

3. Apply low-pass filter (artificial blur) to a copy of the image (Figure 4.7(d), Section 4.2.2.3).

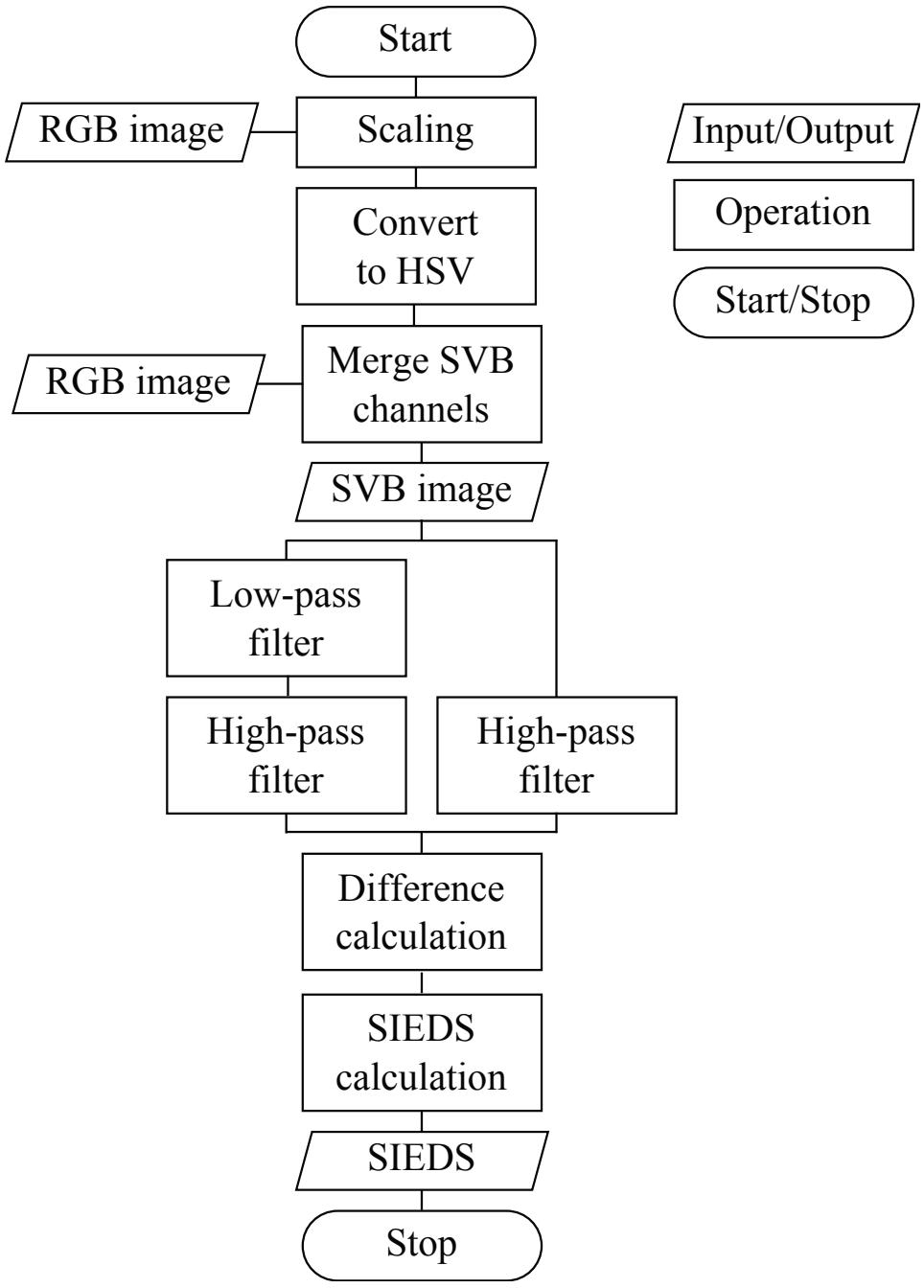
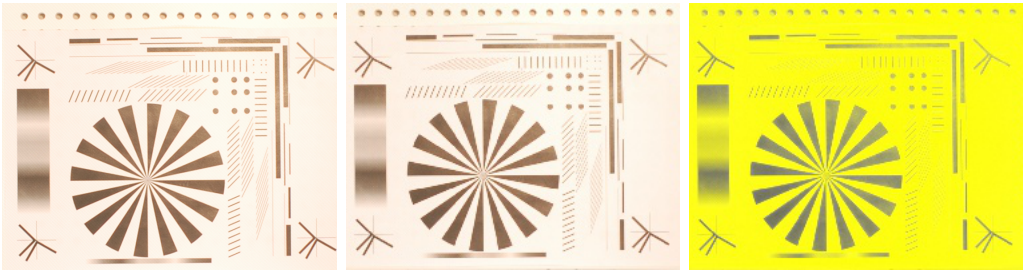


Figure 4.6: Flow chart of developed algorithm.

4. Apply high-pass filter (edge detection) on both: low-pass filtered copy and original SVB image (Figure 4.7(e), 4.7(f), Section).

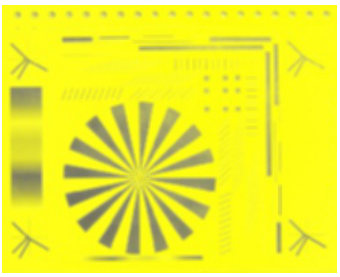
5. Calculate difference between both high-pass filtered images (Figure 4.7(g)).



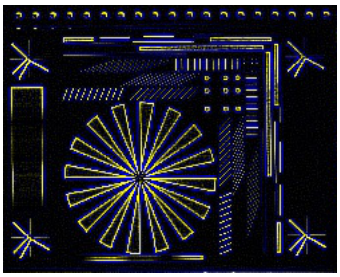
(a) Original input image.

(b) Scaled image.

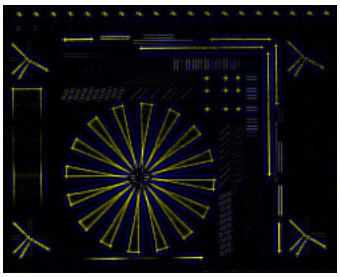
(c) RGB to SVB converted image.



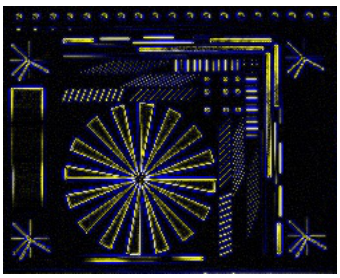
(d) Low pass filtered copy.



(e) High pass filtered original.



(f) High pass filtered copy.



(g) Difference image of high pass filtered images.

Figure 4.7: Comparing of insets of UAV images.

6. Calculate standard deviation of difference image (Section 4.2.2.5.

The calculated standard deviation of the difference image is named 'saturation image edge difference standard-deviation' (SIEDS). SIEDS is a single value representing how much an image is blurred.

Output A small SIEDS value represents a small standard deviation in the difference between the original SVB image and the low-pass filtered SVB image, while a large SIEDS value represents a large standard deviation in the difference between the original

SVB image and the low-pass filtered SVB image. The larger the SIEDS value, the more likely the original input image was initially sharp, while a small SIEDS value indicates that the input image was blurry. This result is similar to that of human perception. The perceived difference between a sharp and a re-blurred image is larger than the difference between a blurred and a re-blurred image.

A SIEDS value can be calculated for each image in the input directory and the calculated SIEDS values enable a precise judgement of how much an image is blurred related to other images in the set. However, the absolute calculated values will depend on the processing steps and image content.

4.2.2.1 Input image scaling

High resolution images contain vast information and inevitably require long processing time. The aim of the algorithm is to calculate the SIEDS value quickly in order to process all the images of a UAV dataset in a reasonable time. One method used to decrease calculation times is to reduce the number of pixels in the image (Figure 4.7) and is achieved by down scaling the image resolution, which reduces the pixel count (Code 4.1).

Listing 4.1: Scaling an image

```
1 //Function for rescaling the image with input of the image to scale and the scaling
      factor. Scaling factor \(1=100\,\)\%
2 Mat Fresize(Mat in, double scale){
3         //Definition of the output image.
4         Mat out;
5         //Resizing of the image by factor 'scale'.
6         resize(in, out, Size(), scale, scale);
7         //Returning scaled image back to function call.
8         return(out);
9         }
```

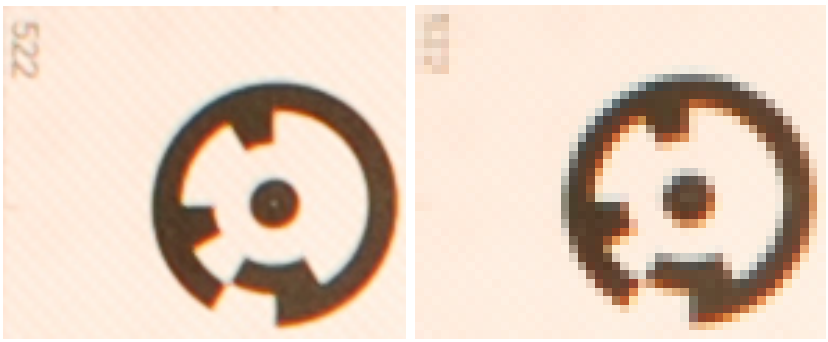
Scaling an image creates several advantages, beyond reducing calculation time, it also significantly reduces the required computer memory. The reduction of the pixel count has

another advantage. By scaling an image, multiple pixels are combined to just one pixel. When these combined pixels contain the same colour, the newly combined pixel will also have the same colour as all individual pixels. If the individual pixels have different colours, then the newly combined pixel will be a result of the interpolation of the different colours. When this newly combined pixel represents an edge it will have an impact on determining whether an image is blurred or not. This is useful because homogeneous areas cannot be used for blur detection. By reducing the number of pixels the number of 'insignificant pixels' representing homogeneous areas is also reduced. However, scaling does not influence edges, which remain important for blur detection.

Furthermore, scaling the image has an advantage of reducing the influence of other effects that appear similar to blur (Figure 4.8). Spectral mixing and optical errors such as chromatic aberration appear similar to blur and these errors can be reduced by scaling the image. In Figure 4.8(a) the effect of optical errors are clearly visible, the red and blue contour lines around the target do not exist, but are an effect caused by chromatic aberration. The effect of spectral mixing is also visible, as there is no strict edge between black and white but a gradient from black via grey to white. By scaling the image it is possible to reduce these effects. The observed gradient in Figure 4.8(a) occupies 8 pixels and scaling reduces this to just 3 pixels in Figure 4.8(b). The levels of scaling needed in an image can be determined by assessing the number of pixels that are initially influenced by these optical errors. In this study the influence of these effects was around 3 pixels in the centre of the image. Therefore, down scaling by a factor of three was used to reduce the impact of these effects to just one pixel.

One side effect that needs to be considered during scaling is that the process reduces the effect of motion blur. However, if a camera is displaced during image acquisition then the effect of motion blur maybe obscured by other effects, such as optical errors. When the effect of motion blur is smaller than the effect of optical errors, motion blur becomes undetectable, as it disappears behind optical effects. However, excessive scaling may result in total elimination of motion blur, making detection of blur impossible.

Scaling therefore clearly influences the detection of motion blur. If scaling is not used then too many pixels have to be processed and the density of usable edge pixels will be low. Subsequent calculations of the SIEDS value using only a small number of edges,



(a) High resolution image. Chromatic aberration and spectral mixing is clearly visible (red and blue contours around the target).

(b) Low resolution image after scaling. Chromatic aberration and spectral mixing is reduced to fewer pixels.

Figure 4.8: Effect of reducing image resolution.

compared to the pixel count, would result in the calculated standard deviation being very small, because many pixels will be zero. Subsequently this makes it difficult to use the standard deviation as a tool to differentiate between blurred and sharp images precisely. Therefore it is very important to use the correct level of scaling to identify smaller amounts of motion blur. However, by reducing the image resolution, the number of pixels is reduced and subsequently the image processing procedures are much faster than using the full resolution image. In this research the image was scaled down by factor 3. The criticality of this parameter will be discussed further in Section 4.3.1.

4.2.2.2 RGB to HSV

The first processing step after scaling an image is the conversion of RGB to the HSV colour space (Code 4.2, Figure 4.7). One of the main advantages with motion blur is that it is not dependent on the wavelength (colour) of the light, hence colour information is not significant for blur detection. Analysing an RGB image would require analysis of each channel separately, which would take significantly longer calculation times. To eliminate the colour information and reduce the image to just necessary information, the three channel RGB colour image has to be converted to a HSV colour space.

Listing 4.2: Colour space conversion

```
1 //Function to convert RGB to SVB image. RGB image as input.
```

```
2  Mat Fcalcsvg(Mat in){
3      //Creating a buffer for the SVB image. It needs to be the same size as the
           input image and needs to have three channels.
4      Mat_<Vec3d> svb(in.size().height, in.size().width, Vec3d(0,0,0));
5      //Loop to access every element of the image matrix
6      for(int n=0; n<in.size().height; n++){
7          for(int m=0; m<in.size().width; m++){
8              //Accessing pixel of the matrix and storing RGB values.
9              Vec3d rgb=in.at<Vec3d>(n, m);
10             //Separating red, green, and blue value of the current pixel
                   and normalising them.
11             double b=rgb.val[0]/255;
12             double g=rgb.val[1]/255;
13             double r=rgb.val[2]/255;
14             //http://cs.haifa.ac.il/hagit/courses/ist/Lectures/Demos/
                   ColorApplet2/t_convert.html and
                   http://www.shervinemami.info/colorConversion.html
15             // r, g, b = [0, 1]
16             // h = [0,360], s, v = [0,1]
17             // if s == 0, then h = 0 (undefined)
18             //Definition of variable to store largest and smallest RGB
                   value.
19             double max=0;
20             double min=0;
21             //Detection of the largest and smallest intensity in either
                   red, green or blue.
22             if(r>=g && r>=b){
23                 max=r;
24                 }
25             else if(g>=r && g>=b){
26                 max=g;
27                 }
28             else if(b>=r && b>=g){
29                 max=b;
30                 }
31             if(r<=g && r<=b){
32                 min=r;
```

```
33                          }
34                      else if(g<=r && g<=b){
35                          min=g;
36                          }
37                      else if(b<=r && b<=g){
38                          min=b;
39                          }
40                      //Value is the largest intensity of red, green or blue.
41                      double v=max;
42                      //Calculation of the discrepancy between largest and smallest
                            grey value.
43                      double delta=max-min;
44                      //If red, green and blue are not equal and larger than zero
                            then saturation can be calculated.
45                      if(delta!=0 && max!=0){
46                          s=delta/max;
47                          }
48                      else{
49                          s=0;
50                          }
51                      //Converting s, v, b = [0, 1] to [0, 255]
52                      s=s*255;
53                      v=v*255;
54                      b=b*255;
55                      //Collecting saturation, value and blue in a vector.
56                      Vec3d svbvector=Vec3d(s, v, b);
57                      //Writing SVB vector at current position in SVB image.
58                      svb(n, m)=svbvector;
59                      }
60              }
61          //Returning SVB image to calling function.
62          return(svb);
63          }
```

The HSV colour space only contains the colour information in the hue channel (Section 2.1.1.1). Hue does not contain any important information and is not of interest for

further processing. Both saturation and value contain information that are interesting for further processing, as they are both influenced by the extent of camera displacement. It has been observed that increasing image blur results in a reduction of saturation and value. This observation can be used to detect whether an image is blurred or sharp.

For the developed algorithm it was deceide to use the saturation channel for further analysis (4.2.2.5 and 4.3.2), however, most functions can only progress a single or three channel image, which made necessary that the image matrix was kept as either single or three channel image. Due to the aim to establish a calculation of an absolute camera displacement it was decided to keep a three channel image, which could provide additional information available in other than the saturation channel. To satisfy this requirement, the value channel which is also influenced by blur (4.3.2) and the blue channel, which is least influenced by ray bending, are added to the saturation channel. These three channel create the newly created SVB (saturation, value, blue) image. An SVB image cannot be converted back to an RGB image as there is no information available about other colours than blue. This technique helps to speed up the calculation process. Furthermore, the image now only contains information that is relevant for blur detection. The fact that the image cannot be converted back to an RGB image is not significant, as a true colour image is not required for the subsequent processing steps.

4.2.2.3 Re-blurring SVB image

After converting the colour space from RGB to SVB, subsequent image processing steps can be conducted. As determined earlier the human brain can differentiate easily between sharp and blurred images, but has difficulties in differentiating between a blurred and an even more blurred image (Section 4.1.3). This ability was identified as a processing step that can be realised in a computer algorithm. To enable a comparison between two images a more blurred image than the original has to be created. This can be done by applying a low-pass filter to the original input image.

Within the formulated algorithm there is a copy of a previously created SVB image (Code 4.3, Figure 4.7).

Listing 4.3: Creating copy of image matrix

```
1 //Image matrix to contain copy.
2 Mat copy;
3 //Copying SVB image.
4 svb.copyTo(copy);
```

A low-pass filter is applied to this copy, which is adding artificial blur to the image (Code 4.4). The copied image is now known to be more blurred than the original input image and can be used by a human operator to determine if and by how much the original image is blurred. Depending on the degree of added blur, it is easier to perceive the additional blur and to determine if the original input image was indeed blurred.

Listing 4.4: Low-pass filtering of an image matrix

```
1 //Function to low-pass/blur/smooth filter an image. Image and blur kernel sizes are
      required as input.
2 Mat Fsmooth(Mat in, int kernel){
3         //Creating image matrix for the output.
4         Mat out;
5         //Blurring the image.
6         blur(in, out, kernel);
7         //Return the low-pass filtered image
8         return(out);
9         }
```

The perception of additional blur largely depends on how much the original image was already blurred and how strongly the image was re-blurred. Adding additional blur to the copy will make the differentiation easier, as the discrepancy between the original and the re-blurred image will be much larger. It would also be possible to differentiate an image that already contains large blur from the re-blurred image and to detect small camera displacements. However, if the input image only contains a small amount of camera displacement it would be still possible to determine if the original image was blurred but it would not be possible to determine the amount of blur. This could be solved by re-blurring the input image with a small amount of additional blur, enabling more precise differentiation between an image with small camera displacements and the re-blurred image. However, the detection of severely blurred images will still be difficult

because the discrepancy between the input image and the re-blurred image is too small. The future use of the images will determine the amount of additional blur required.

The detection of blur needs to be fully automated and work independently from a human operator. Until now, detection of blur still requires a human operator to compare the original image and the re-blurred image and to make a judgement as to whether the original image was blurred. To achieve full automation, additional image processing steps are required to form the basis of a numerical measure of blur.

4.2.2.4 Edge detection and discrepancy calculation

Once an image containing additional blur has been created, a comparison can be carried out. To detect blur in an image a human concentrates mostly on the edges which are represented by the gradient between different grey values. The gradient between different grey values enables a human to judge whether an image is blurry or not. This approach can be implemented in a computer algorithm. To detect gradients in an image automatically a high-pass filter has to be applied to the original and to the re-blurred image, creating an edge image for both (Code 4.5, Figure 4.7). A high-pass filter enables edge detection by calculating the gradient between neighbouring pixels (Section 2.1.1.3).

Listing 4.5: High-pass filtering of an image matrix

```
1 //Function to calculate gradients in an image.
2 Mat Fcalcedges(Mat in){
3         //Creating the output image matrix.
4         Mat out;
5         //Calculating the gradients using a Laplace operator.
6         Laplacian(in, out, CV_64FC3);
7         //Returning the Laplace filtered image to the calling function.
8         return(out);
9         }
```

For this purpose, a 3 x 3 Laplace operator is used which calculates the gradient between one pixel in the centre of the filter matrix and the four adjacent pixels. This high-pass

filter is then applied to both, the original and the re-blurred image. Using a gradient to judge whether an image is blurred or not, would require the comparison of the gradient of each pixel to their neighbours. Additionally, it would be necessary to decide whether a pixel is part of an homogeneous area or part of a blurred edge. Both would appear in the edge image as low gradient. This approach is rather complicated, requires intensive calculations and would increase the calculation time significantly.

To avoid these extensive calculations a simple discrepancy image is calculated, which is the difference between the edge image of the original and the re-blurred edge image (Code 4.6, Figure 4.7). This image shows the discrepancy between gradients calculated for both, the original and re-blurred image. In homogeneous areas the discrepancy will be close to zero while the discrepancy at the edges will be significantly greater than zero. With the discrepancies calculated it is possible to finally calculate the SIEDS value.

Listing 4.6: Calculation of the discrepancy image.

```
1 //Function to calculate the discrepancy between both edge images. Input are both edge
      images.
2 Mat Fdiff(Mat original, Mat reblurred){
3         //Creating an output image matrix.
4         Mat out;
5         //Calculation of the absolute difference between both edge images.
6         absdiff(original, reblurred, out);
7         //Returning the calculated difference image to the calling function.
8         return(out);
9         }
```

4.2.2.5 SIEDS calculation

After calculation of the discrepancy image it is possible to carry out the last step to determine a single floating point number, which quantifies whether the image is sharp or blurred, the SIEDS value. For the SIEDS value it is important to understand the expected results of the previous processing steps. The gradients in the low-pass filtered image should be lower than the gradients calculated for the original image. Due to the added blur the edges are supposed to be 'smoother' and the gradients flatter.

Additionally, the extrema are 'flattened'. Hence, the standard deviation of the gradients should be smaller for the re-blurred image than for the original. However, neither the standard deviation for the original edge image, nor the re-blurred edge image provides a clear measure about the amount of blur in either image.

The discrepancy image is derived from the original and re-blurred edges. It is expected that the gradients are smaller than in the original image. How much smaller the gradients are depends on the amount of blur added to the re-blurred image and the amount of blur that existed in the original. If the input image was sharp, then the re-blurred image will have significantly smaller gradients. The discrepancy between the original and re-blurred images will therefore be large. However, if the original image exhibited blur then the re-blurred image will have similar but smaller gradients and the discrepancies between them will be small.

As a result, the average gradient discrepancies will be smaller or larger depending on the sharpness of the input image. Unfortunately, the average also depends on how many gradients are available in the image, due to the large number of small values that appear in the homogeneous areas. A rough texture with a large number of edges will create more gradients and increase the average. In comparison, an image with limited texture will return many values close to zero, hence a much smaller average. However, to improve calculations it was decided to use the standard deviation instead of the average grey value. This provides the advantage that the calculation is made independently of how steep the gradients are, and instead uses gradient variation (Code 4.7).

Listing 4.7: Calculation of SIEDS

```
1 //Function to calculate the average, variance and standard deviation for each channel
      of the SVB edge difference image. Input is SVB edge difference image, the file
      name of the image, output file to store calculated values.
2 double Fcalcmean(Mat in, string imagename, ofstream &pData, double *rgvpx){
3         //Buffer variable to carry out the calculation of average, variance and
              standard deviation.
4         double mean[18]={0};
5         //Loop to go through every pixel of the image.
6         for(int i=0; i<in.rows; i++){
7                 for(int j=0; j<in.cols; j++){
```

```
8                       //Accessing pixel of the matrix and storing SVB values.
9                       Vec3d svb=in.at<Vec3d>(n, m);
10                      //Buffer variable to store the currently processed value in
                            each channel
11                      int buf=0;
12                      //Loop through all three channels of the image
13                      for(int n=0; n<3; n++){
14                              buf=rgb.val[0];
15                              //Calculation of running average.
16                              mean[1+n*6]=((mean[0+n*5]*mean[1+n*6])+buf)
                                    /(mean[0+n*5]+1);
17                              mean[3+n*6]=mean[3+n*6]+buf;
18                              mean[4+n*6]=mean[4+n*6]+pow(buf, 2.0);
19                              //Counter of every pixel.
20                              mean[0+n*6]++;
21                              //Calculation of standard deviation.
22                              mean[2+n*6]=sqrt(1/mean[0+n*6]*
                                    (mean[4+n*6]-(1/(mean[0+n*6]+1)*
                                    pow(mean[3+n*6], 2.0))));
23                              //Test if pixel had a larger value than zero (Was a
                                    gradient available?).
24                              if(gv.val[n]>0){
25                                      //Counter of pixels that have larger values
                                            than zero.
26                                      mean[5+n*6]++;
27                                      }
28                              }
29                      }
30              }//Calculation of values finished for the image.
31      //Storage of the calculated values in file. First element is the file name.
32      pData<<"#"<<imagename;
33      //Loop over all channels.
34      for(int n=0; n<3; n++){
35              //Output to file are the channel number, number of pixel greater than
                    zero, average value and standard deviation.
36              pData<<"#"<<n+1<<"#"<<mean[5+n*6]<<"#"
                    <<mean[1+n*6]<<"#"<<mean[2+n*6];
```

```
37                  }
38          }
```

4.3 Results and limits of detection

The saturation image edge difference standard deviation (SIEDS) value is one single value used to represent the amount of blur for a single image. The value is either large when the image is sharp, or small when the image is blurred. Judgement as to whether an image is blurred or sharp is dependent on all the values which have been calculated for all images in a dataset. However, the settings of the algorithm will also have an influence on how large or small the calculated SIEDS values are. Using a normalised SIEDS value to determine a fixed threshold and to make datasets comparable is incorrect because most datasets will not contain images distributed normally over the range of motion blur. Some datasets might only contain small, acceptable blur for the complete dataset while other might only contain larger blurs. Normalised SIEDS values of an overall acceptable dataset would range SIEDS$= [0, 1]$ and the same for a dataset of not acceptable quality. However, the normalised values would cause the incorrect conclusion that both datasets contain acceptable and not acceptable images (Section 6.1.1.1).

4.3.1 Critical parameters for the algorithm

Influence of image scaling The algorithm can be modified by changing critical parameters which all have an influence on the calculation of the SIEDS value (Section 4.2.2). Modification of the scaling factor influences the calculation time for an image dataset. The greater the reduction, the shorter the calculation time. However, scaling images does not only influence the calculation time, but also the subsequently calculated SIEDS value. To assess the impact of scaling on the calculated SIEDS value, a dataset of 600 images was processed repeatedly, each run containing different extent of camera displacement. Figure 4.9 shows the calculated SIEDS values for different camera displacements and different scaled images. It is possible to see that without any image

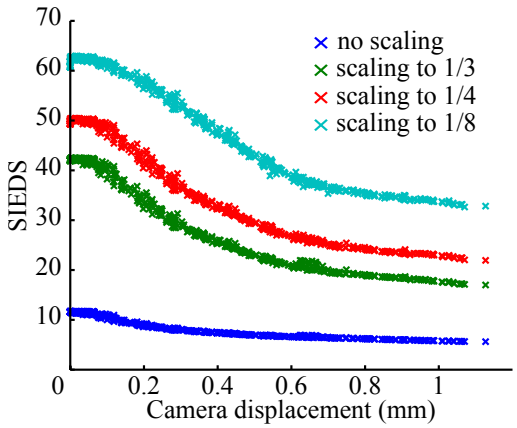


Figure 4.9: Influence of image scaling on calculated SIEDS.

scaling the calculated SIEDS value barely changed with increased camera displacement (Figure 4.9).

By scaling the image to a third of the original size the calculated SIEDS value changed significantly. It was found that the calculated SIEDS value is larger than without scaling. Furthermore, not only is the value increased, but the difference between the largest and smallest SIEDS value is 23 units. This is three times larger than for the unscaled images, which only had a difference of 7 units between the largest and smallest calculated SIEDS value. This large discrepancy enables precise differentiation between sharp images, images with small camera displacements and images with larger camera displacements. By scaling the image further the calculation time decreases further, while the calculated SIEDS value increases. However, the difference between the largest and smallest SIEDS value does not increase significantly and does not provide any advantages.

For the unscaled images the SIEDS value changed by just 7 units. Reducing the image size by a third, the change in the SIEDS value increases to 23 units, 24 units for a fourth and 26 for one eighth of the original image size. It is possible to see that the change in the SIEDS value does not increase significantly with decreasing resolution. Scaling results in a much faster calculation of SIEDS, but comes with the risk that small structures in the image, which could be used for blur detection, are degraded to a degree that they are not usable. Additionally, the laboratory images contain high contrast edges which are not common in natural images.

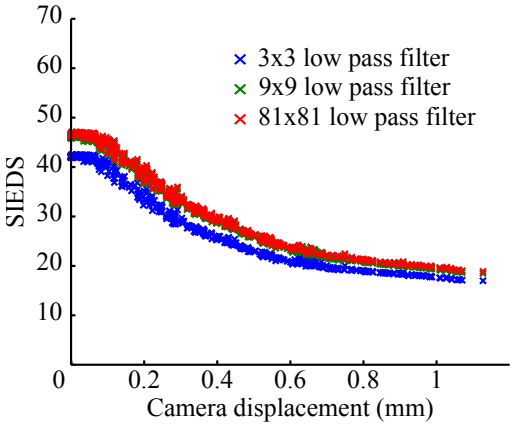


Figure 4.10: Influence of additional blur on calculated SIEDS.

Based on the quality of the equipment used, the effect of chromatic aberration, spectral mixing, image scale and the contrast of the edges, it was decided that a scaling factor of 3 should be used to return the best results.

Influence of additional blur Another setting that influences the calculation of the SIEDS value is the amount of blur that is added to the copy of the image. To visualise the difference in the calculated SIEDS values the images were degraded with different extents of additional blur (Figure 4.10). It was found that adding different low-pass filters increased the difference between the SIEDS value of a sharp image and of a blurred image. However, it was also found that the increased difference was not significant. For a 3 x 3 filter the difference is 23 units, while it is 24 units for a 9 x 9 low-pass filter. Even for a larger 81 x 81 low-pass filter the change did not prove significant. Larger low-pass filters also require longer calculation times and slow the calculation process.

4.3.2 Calculated SIEDS

The calculated SIEDS value is derived from the standard deviation of the saturation channel. The use of the value channel of the SVB image or any of the original RGB channels were also tested to see if they could be used instead of saturation (Figure 4.11). Figure 4.11 presents the "SIEDS" values calculated in dependency to camera displacements. Beside the SIEDS value based on saturation, "SIEDS" values based on

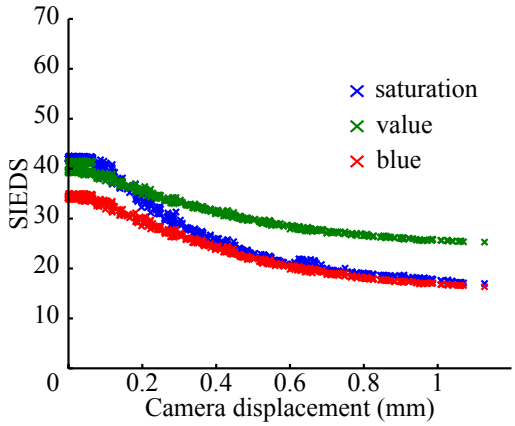


Figure 4.11: Different image channels used for SIEDS calculation.

the value and blue channel were calculated. The results showed that the difference between the largest and smallest SIEDS value is much larger for saturation than for the value and blue channels. The difference for the blue channel is about 16 units, while it is 15 units for value channel. The gradient for the saturation is much larger with 23 units, which enables greater distinction between the sharp and the blurred images. The saturation channel is most sensitive, which can be explained by considering blur in more detail. When an edge is blurred, the contrast reduces, whilst the colour remains the same. A change in contrast is represented in the saturation of colour, hence saturation is most influenced by blur. This strong influence on the saturation channel again supports the decision to select saturation as the best channel to quantify the sharpness of an image.

By calculating the SIEDS value of one image it is not possible to judge if an image is blurred or not. Just one single number does not have any significance. To be meaningful it needs to be set in context with other SIEDS values of other, similar images. Other similar images being acquired with the same camera and lens system at the same time and of similar terrain. As presented in Section 4.3.1, a change in the image resolution does affect the calculated SIEDS value. Based on this it can be assumed that cameras with different resolution will return different SIEDS values, even when the images show the same area, with the same amount of camera displacement. The different SIEDS values would imply that the images contain different amount of camera displacement, which would be incorrect. The same lens is important to ensure that the same optical

errors influence the images. A lens of high optical quality will often return a larger SIEDS value, due to the more accurate and sharper image produced by the lens. A lens of low quality will return images of lower quality and hence the calculated SIEDS value will be smaller. Differences caused by lens choice must not be misinterpreted when deciding between sharp or blurred images. The same camera and lens is normally used for one image dataset in photogrammetric processing, which reduces this particular difficulty.

To make a comparison between different SIEDS values, the images should be from the same camera dataset, taken under similar conditions and display similar terrain. Similar terrain implies areas that show one type of terrain (e.g. forest, agricultural). The requirement that the images show similar terrain is needed for the approach based on edges, which are inherently variable with different types of terrain. However, this is not a problem as UAV flights are normally acquired over one type of terrain to analyse this certain type of terrain. Large UAV datasets provide generally enough similar images to enable a comparison between these and to define a threshold between sharp and blurred imagery. Datasets with a variety of terrain can cause problems and should be avoided or datasets pre-processed by separating images by their terrain. Subsequently calculation of SIEDS values can be conducted on the separate image stacks.

The SIEDS value is not a random number. It is calculated as a value between zero and a value less than a half the bit depth of the edge difference image, which would be for a standard 8 bit image SIEDS= $[0, 127)$. The zero value can only be reached if, either every pixel of the image has the same grey value, or all neighbouring pixels have the same gradient, similar to a chessboard pattern. If all pixels have the same grey value then the image would not contain any edges (Figure 4.12). Without any edges blur detection would become impossible. In case of a homogeneous chessboard pattern, each pixel would have the same gradient, so that the difference between them is zero again and the standard deviation is zero too (Figure 4.13).

This shows that homogeneous images or repetitive patterns would not be suitable for blur detection using this method. This disadvantage should not be significant as UAV images normally show large variations in patterns and colours. Furthermore, these two types of images do not contain any useful information for photogrammetric procedures, as the unique identification of features for coordinate measurement would not be possible.

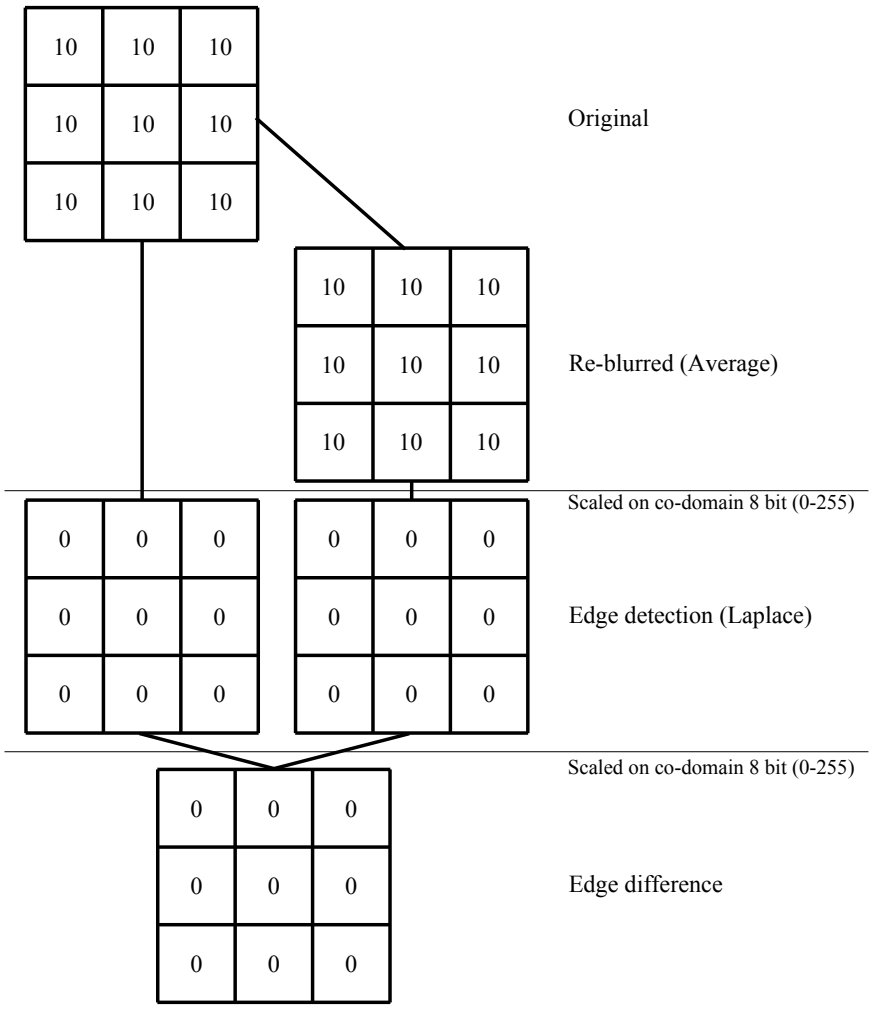


Figure 4.12: Calculation of SIEDS for homogeneous image would return 0.

The largest SIEDS value would be achieved with zero gradients for every first pixel and the largest gradients for every second pixel in the edge difference image. The standard deviation would be calculated as half the value between the largest and zero gradient. For an edge difference image of 8 bit with the largest value 255 representing a steep gradient the SIEDS value would be 127.5. However, an edge difference image containing zero gradients for every first pixel and largest gradients for every second pixel is most unlikely. Already the edge of the original and the re-blurred image, which are used to calculate the edge difference, would have to be zero in the one image and the largest value in the other image. However, the largest practically achievable SIEDS value is achieved by assuming a steep edge caused by one pixel. This SIEDS value would then be 103,

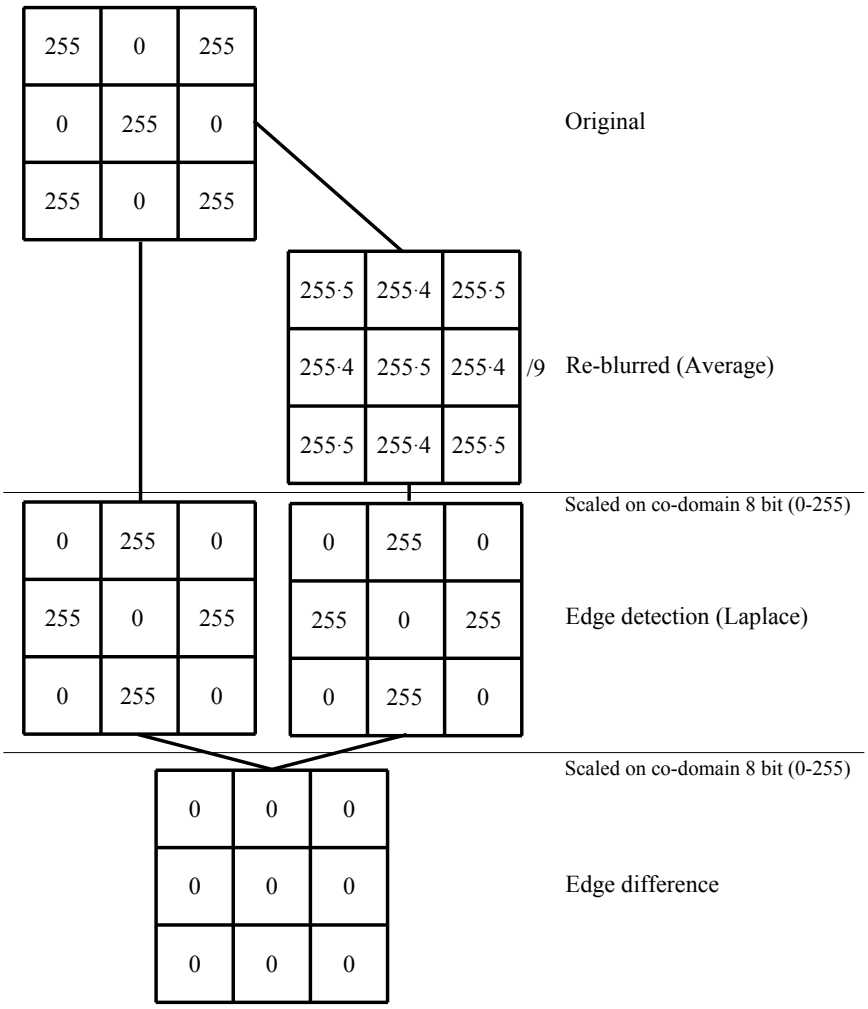


Figure 4.13: Calculation of SIEDS for homogeneous pattern would return 0.

which would be the maximum reachable value (Figure 4.14). However, natural images do not produce these SIEDS values as they provide more diverse gradients. During the development of the program it was found that values between $30-60$ are more typical.

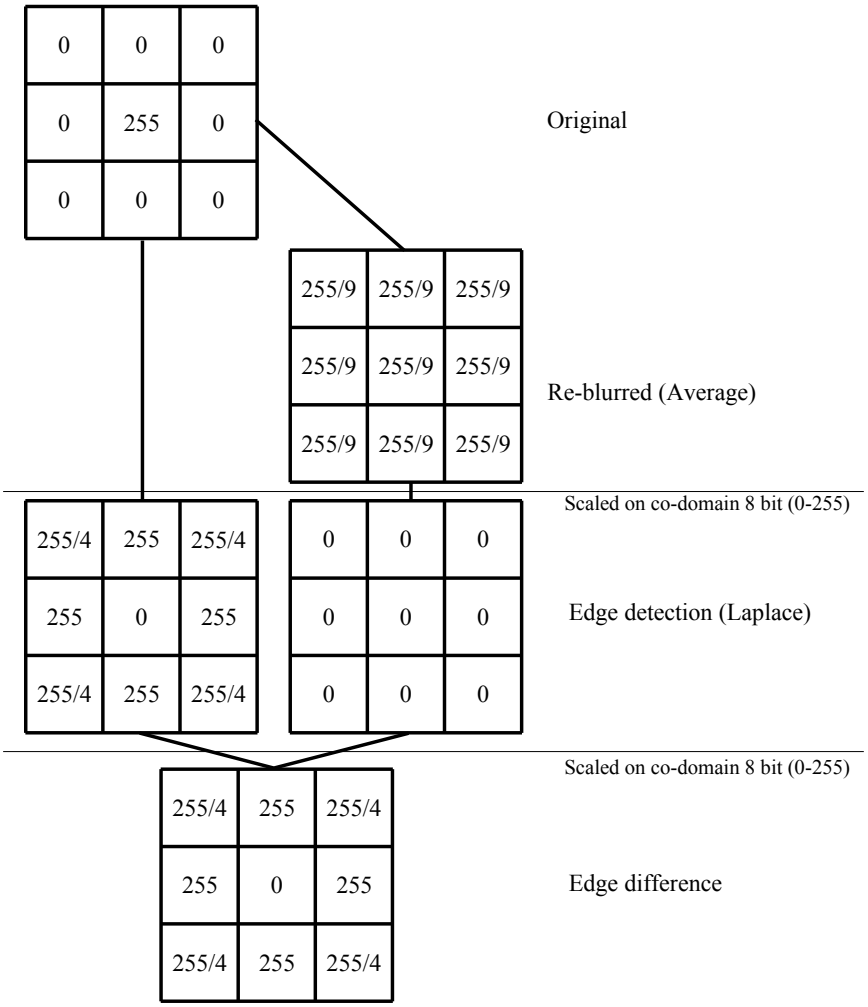


Figure 4.14: Calculation of SIEDS for steep edge would return 103.

4.3.3 Beyond SIEDS

The calculation of SIEDS values for images with known camera displacements reveals that the calculated standard deviation does not change linearly with increasing blur. However, it is obvious that the dependency between SIEDS and camera displacement is continuous and can be described by a function. The function best describing the curve is a special case of the damping function, the over-damped oscillation. (Equation 4.1).

$$b = e^{-\delta w}\left(\frac{\delta x_0}{\omega}\sinh(\omega w) + x_0\cosh(\omega w)\right) + n$$

$$\begin{aligned} b &\ldots \text{Blur.} \\ \delta &\ldots \text{Damping ratio.} \\ x_0 &\ldots \text{Start position.} \\ \omega &\ldots \text{Angular frequency.} \\ n &\ldots \text{y-offset from zero} \\ w &\ldots \text{SIEDS, derived from image} \end{aligned} \tag{4.1}$$

A damping function describes an oscillation, which is damped (Deutsch, 2015). Due to resistance the oscillation is reduced with every wave, which results in zero amplitude after a certain period of time. A special case of the damping function is the overdamped oscillation, which means that there is no complete wave before the end of oscillation. A practical example for this kind of function are damped doors. After opening they close but the rate of closure slows before they are shut.

It is not surprising that a damping function can be found in images affected by camera displacement. Images are the visualisation of 2D signals (Section 2.1.1.3), blur damps the signal and in the most extreme case, with an infinite camera displacement an image would appear as an homogeneous coloured 'blotch'. This 'blotch' would have the average colour of the area photographed in the image. The SIEDS value would be large when the image is sharp and high amplitudes are available. With increasing blur, edges in the

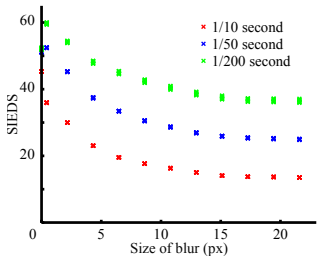


Figure 4.15: SIEDS calculated for images blurred using Matlab.

image would disappear and the SIEDS value would decrease.

The damping function helps to visualise the relationship between camera displacement and SIEDS. SIEDS can be used with the damping function to estimate the displacement of the camera. It was also found that the determination of the other unknown variables, the offset of the y-axis (n), the angular frequency (ω), the start position (x_0) and the damping ratio (δ) are possible, but dependent on various factors such as image size, number of edge pixels, average grey value and others.

4.3.4 SIEDS for mathematical blurred images

It is also possible to analyse images which are blurred using mathematical methods such as low-pass filters. The analysis found that images containing larger levels of blur return smaller SIEDS values. This is similar to images containing blur introduced by physical camera displacement. However, the relation between the mathematically applied blur and the calculated SIEDS value does not represent a damping function (Figure 4.15). By calculating the applied blur to SIEDS curves for the different exposure times it is possible to see that the functions are more similar to a logarithmic function. The reason for this is that mathematically blurred images reduce effects like noise, spectral mixing and other hardware impacts. It is also not possible to do sub-pixel blurring using mathematical blur methods. Sub-pixel displacement of the camera could provide new information that cannot be made available using mathematical blurring methods. Mathematical low-pass filters can also only use the information already captured in the image but not information that the object would provide.

4.3.5 Other methods to calculate SIEDS

There are also other methods to calculate SIEDS values, for example the standard deviation can be calculated based on the input image, minus the re-blurred image, without applying edge detection. The damping function was also applied here, but the gradient between the sharp images and images with large camera displacements was too small to guarantee correct differentiation between blurred and sharp image.

Another method to calculate SIEDS values includes application of the inverse process to a low-pass filter applied on the input image. It has been investigated if high-pass filtering instead of low-pass filtering could be useful. In this study, the copy of the original image was not treated with a low-pass but a high-pass filter. After applying the high-pass filter the high-pass filtered image was applied on the input image to generate an enhanced image used as comparison. Then both, the input image and the edge enhanced image were processed using a high-pass filter to find the edges and subsequent edge differences calculated. It was found that this procedure produced similar results to using a low-pass filter. However, the calculation of a high-pass filtered image and subsequent enhancing of the input image requires one additional step in the procedure to the calculation of SIEDS. This step takes additional time during the calculation procedure and was considered unnecessary.

4.4 Discussion and future work

The idea of using edge sharpness for blur detection and quantification is not entirely novel. Crete et al. (2007) used the edge sharpness approach in their research, however, real blurred images were used, which showed that the method did not return reliable results (Section 4.1.3). Despite some similarities to use additional blur for blur detection, the Crete et al. (2007) method is different from the method presented in this research (Table 4.1).

A similar approach is used also by *Agisoft PhotoScan* and their 'Estimate Image Quality' tool. This tool estimates image quality and returns values between 0 for blurred to 1

Table 4.1: Comparison of Crete et al. (2007) to developed method.

Step	Crete (2007)	Sieberth (2015)
1.		Scaling Image
2.		Convert to SVB
3.	Blur the image vertically and separately horizontally	Blur the image in both directions
4.	Compute the variation between vertically and horizontally pixels from the original and from the re-blurred images.	Detect edges in both, the original and re-blurred images.
5.	Calculate the difference between the vertical variation images and separately for the horizontal variation images.	Calculate the difference between the edge detection results.
6.	Summation of all pixel values for both original variation images and the calculated difference images from step three.	
7.	Normalize the results retrieved in step seven.	
8.	Select either the vertical, or the horizontal value as the blur value (dependent on which one is larger).	Calculate the Standard deviation for all pixels

for sharp (Agisoft LLC, 2013). *Agisoft PhotoScan* "[...] estimates the quality by the border sharpness, here each pixel has its own border." and "refers [the quality value] to the area of highest quality" (Pasumansky, 2014). However, the main difference between these two approaches is the conversion of RGB images to a different colour space, which is necessary to exclude the unimportant colour information and reduces the important saturation channel. Furthermore, the detection developed in this thesis is based on a difference image calculated using images that have been processed using a high-pass filter.

Compensation for rolling shutter or distorted images would be possible. A way to do so would be by weighting the values during calculation of SIEDS based on the camera and lens model. Another way would be by calculating the SIEDS not for the complete image but only for areas of interest.

The calculation of the SIEDS value is dependent on the camera displacement described by a damping function, that could be used to convert SIEDS to a physical value representing the camera displacement. However, the parameters required by the damping function and their dependencies should be researched further with various camera models, different camera displacements and image content. This could perhaps establish a model that would enable the calculation of actual physical camera displacement, based on a blurred image.

4.5 Conclusion

This chapter focused on the detection of blur in images and explains the method developed and demonstrate that it is possible to detect blur caused by linear camera displacement. The method is based on human detection of blur. Humans use an image for comparison in order to establish whether an image is blurred or not. By creating a comparable image using a low-pass filter this method is independent of additional images acquired from the same position with the same camera. However, the calculated SIEDS (saturation image edge difference standard-deviation) value on its own does not provide an absolute number to judge if an image is blurred or not. To do this the SIEDS values

of other similar images have to be compared. However, the need for other SIEDS values is not an issue, as the detection method is aimed at photogrammetric applications which require more than just one image. The developed algorithm ideally suited for large UAV imagesets as it is both, fast and reliable.

5 Blur correction

Blur correction is the next logical step after blur detection. This chapter will explain how blurred images can be deblurred. However, deblurring is a topic often researched and many different methods have been developed in recent years. To give an insight into existing methods a literature review is presented, explaining the most common methods used for deblurring. Subsequently, an algorithm will be presented which was developed to deblur UAV images which include photogrammetric targets.

5.1 Literature review

Finding techniques to deblur images has been an active research topic since the beginning of image acquisition. Since charge-coupled device (CCD) and complementary metal-oxide-semiconductor (CMOS) image sensors have been available, image enhancing has become a prime focus of research. Enhancing images to get sharper, more detailed results, is often desired by the film industry and is 'a dream' for every photographer. The film industry suggests that a sharp image provides information which is accurate to sub-pixel level. For example, it is possible to make text legible that would traditionally have been unreadable due to its too small size. This is sometimes referred to as 'super-resolution'. To create such 'super-resolution' images, additional information has to be provided (Bascle et al., 1996). One application of 'super-resolution' imagery is in medical imaging necessary to support medical decisions (Park et al., 2003). Another example is the use of 'super-resolution' mosaics created by videos captured from UAVs (Wang et al., 2008). However, both techniques require highly overlapping images acquired from slightly different positions, which are then merged to create a higher resolution representation. Although relevant, super resolution imaging is quite

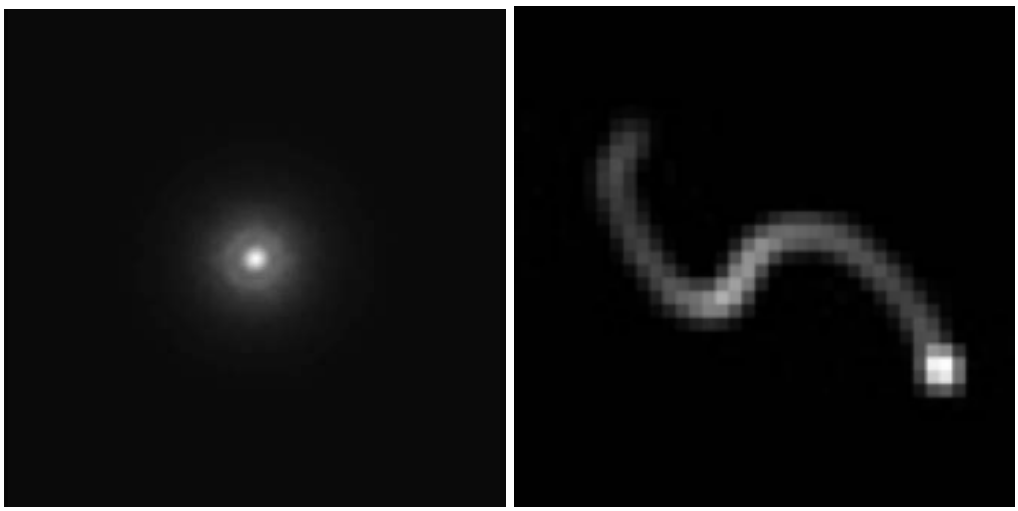
different to image deblurring. Deblurring, in contrast to 'super-resolution', aims to restore information in the image that has been degraded due to blur.

5.1.1 Blur correction

Blur correction is a step that is performed widely in image processing. Mathematically, blur can be represented by a convolution of an image and deblurring is often called 'image deconvolution' (Jia, 2007; MacAdam, 1970; Rav-Acha and Peleg, 2000; Yuan et al., 2007). If deconvolution is carried out without additional information from other sensors or images and only based on the single blurred image, then this process is called 'blind image deconvolution'. If additional sources of information are used it is described as 'non-blind image deconvolution' (Kundur and Hatzinakos, 1996).

5.1.1.1 Point spread function and blur kernel

The easiest way to deblur an image involves an original sharp image, which shows exactly the same scene as the blurred image. The original sharp image can be used to estimate the point spread function (PSF) of the blurred image, using edge detection algorithms (Charlmond, 1991). It is also possible to use image pairs to compute a blur kernel (Yuan et al., 2007). In both cases a PSF is calculated, which represents how pixels are distorted in an image (Nagy et al., 2004) (Figure 5.1(a)). The larger the spread of the point function, the larger is the image blur. However, this blur definition can only describe uniform blur, but not a more complex blur path. A more advanced approach uses a blur kernel, which can represent up to four dimensional (3 space and 1 time coordinate) (4D) movement of a pixel in time space (Figure 5.1(b)). It represents the path of a pixel during image exposure. The longer and more complex this path, the more complex the movement of the camera, respectively of the object. If the pixel was blurred with varying speed, the result is represented by different grey values of the path. A slow movement of the camera is represented by a high DN, whilst lower DN represents faster movements. Figure 5.1(b) represents a blur kernel in which the image was blurred in an 's' shape. Starting at the top left corner, motion speed was fast and then it slows down towards the middle of the path. After accelerating again, it slowed again and finally



(a) Example of PSF by Nagy et al. (2004) showing a uniform blur.

(b) Example of a blur kernel by Tai et al. (2010) showing a non-uniform blur in 3D (2 space, 1 time coordinate).

Figure 5.1: PSF and blur kernel as representation of blur in an image.

stopped in the bottom right, which is represented by the highest DN.

Precise inertial measurement units (IMUs) and global navigation satellite systems (GNSS) can be used to record actual camera displacement during image acquisition and to generate appropriate blur kernels. Assuming an image exposure time of $\frac{1}{400}$ s would require an IMU of at least 800 Hz to measure an approximate blur kernel for an image (Grenzdörffer et al., 2012). A blur kernel based on two measurements can only provide a linear representation of the motion and would not be able to represent more complex motion. Unfortunately, even if linear representation of motion blur is sufficient to represent typical UAV flight motion, such systems remain expensive and can rarely be used since they would exceed the payload of typical micro UAV platforms.

5.1.1.2 Blind deconvolution

In contrast to non-blind deconvolution, blind deconvolution has only the blurred image as an input to the process. No additional information is available, but the task of deriving a sharp image remain. The 'single image blind deconvolution' approach calculates a PSF or blur kernel using a single image to undertake deconvolution. Two methods for

deconvolution are the Wiener deconvolution (Wiener, 1950) and the Richardson-Lucy deconvolution (Lucy, 1974; Richardson, 1972), which both represent blind deconvolution methods proposed decades ago. However, both methods remain popular because they are simple and efficient. Both methods use Bayes's theorem, from the field of probability theory (Equation 5.1 (Leonard and Hsu, 1999)) (Fergus et al., 2006; Jia, 2007; Whyte et al., 2010; Yuan et al., 2007).

$$P(A\|B) = \frac{P(B\|A)P(A)}{P(B)}$$

$$\begin{aligned} P(A) &\ldots \text{Probability of } A. \\ P(B) &\ldots \text{Probability of } B. \\ (A\|B) &\ldots \text{Probability of } A \text{ given that } B \text{ is true.} \\ P(B\|A) &\ldots \text{Probability of } B \text{ given that } A \text{ is true.} \end{aligned} \tag{5.1}$$

Richardson (1972) states that the degraded image (h) is of the form $f \otimes g = h$, where f is the original image, g represents the PSF and $\otimes$ denotes the convolution operation. It is assumed that f, g and h are discrete probability-frequency functions. Bayes's theorem can be used to calculate f using an iterative approach and requires an initial value (f_0). However, noise in the blurred image can create difficulties, which can generate disturbance in the restored image. This problem remains unresolved (Fergus et al., 2006). Another significant problem that occurs using these methods are ringing artefacts which appear at high contrast edges. These approaches were examined and it was found that photogrammetric processes are negatively influenced by ringing artefacts, due to the appearance of additional structures, which appear similar to targets (Figure 5.2). Target detection algorithms therefore have significant difficulties in detecting targets correctly, due to these artefacts. This makes the use of these approaches, which create ringing artefacts, impossible for use in automated photogrammetric applications.

More advanced methods are also based on the probability calculation used by the Wiener and Richardson-Lucy deconvolution. Sun et al. (2014) uses image priors to conduct deblurring. They aim to improve deblurring and increase the reliability even with noise

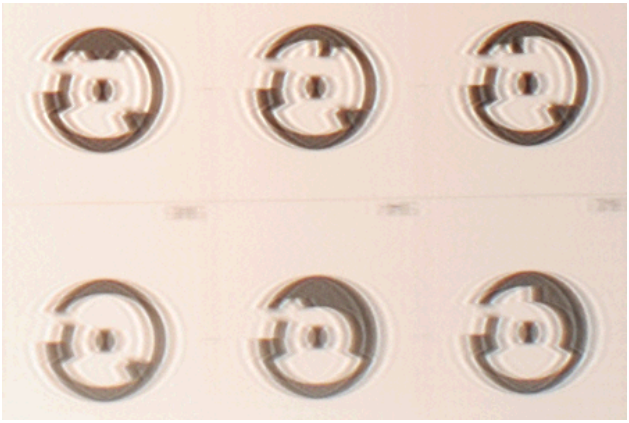


Figure 5.2: Lucy-Richardson deconvolution result (10 iterations). Ringing artefacts are clearly visible

present in the blurred image (Shan et al., 2008). There are also many other methods for blind deconvolution such as 'Fast Image Deconvolution using Hyper-Laplacian Priors' Krishnan and Fergus (2009) or Michaeli and Irani (2014), who uses recurrence of image patches. However, these methods are often computationally expensive and require significant time and memory, which is a major disadvantage for large UAV image datasets. However, the huge amount of images available in UAV datasets is an advantage, as the additional images can be used by non-blind deconvolution methods instead.

5.1.1.3 Non-blind deconvolution

Non-blind image deconvolution methods can be carried out in various ways and require additional knowledge. Additional information can be gained through a variety of methods including, other overlapping images (Agrawal et al., 2012), precise IMU measurements (Joshi et al., 2008), video cameras (Tai et al., 2008), fluttering shutters (Raskar et al., 2006) or colour channel dependent exposure times (Lelégard et al., 2012). The main aim of these methods is to establish either a blur kernel, which can be used for deblurring. The approach which utilises overlapping images appears useful and is based upon the idea that each overlap provides a small amount of additional information to generate a deblurred or even a super-resolution image.

Raskar et al. (2006) developed an approach to allow overlapping but slightly misaligned images to be used in deblurring. A small translation can be tolerated and interpreted as

motion blur, it does not need to be aligned because it can be described in the computed blur kernel. Larger translations still require conventional image alignment methods (Raskar et al., 2006).

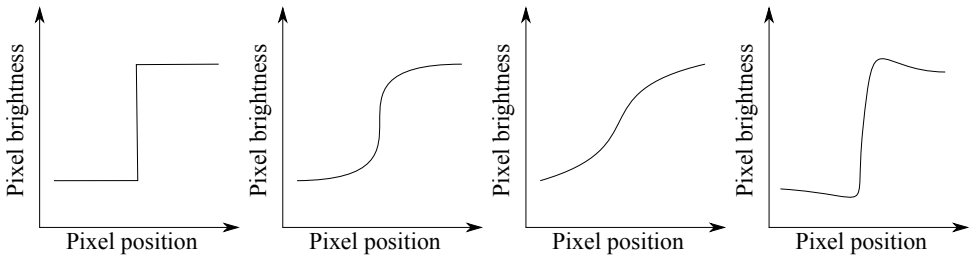
Another method that uses PSF was developed by Agrawal et al. (2012), it uses images of a blurred moving object with an associated PSF. A discrete Fourier transformation can be applied to the PSF. Due to the discrete Fourier transformation, the frequency domain is ill-posed and contains frequencies without information that can be detected. This can be done on images with different exposure times and information can be filled with the frequencies of other transformed PSFs. This composite PSF in the frequency space can then be inverted to the spatial domain and used for the deconvolution of the blurred object. However, the assumption is made that whilst the object maybe blurred, the image contains a static background. In addition, images require varying exposures and need to be captured from the same position.

Another method using Fourier transformation was developed by Jung et al. (2009). Jung et al. (2009) used the frequency domain of an image sequence to merge all frequencies of the same region together. This method does not recover lost frequencies but can combine existing frequencies. However, the approach by Jung et al.'s (2009) only works with fully aligned image sequences. It can be assumed that they use mathematical blurring algorithms or optical blur, to create the aligned image sequence, not real motion blur.

All of the methods described work with aligned images. However, deblurring of unaligned images is rarely tested and problems are expected.

5.1.1.4 Unsharp mask

Another method which is not real deblurring in the sense of deconvolution, does not require overlapping images, is fast and easy to apply and returns useful results, is the image enhancement method called 'unsharp mask' (McHugh, 2015; The GIMP Help Team, 2015). This technique improves the contrast at edges, which makes the image appear sharper. The unsharp mask approach is based on image processing techniques, such as edge detection and contrast enhancement and will be explained fully here.



(a) Well defined edge in a sharp image.

(b) Smoother edge in a blurred image.

(c) Edge smoothed further following use of a low pass filter.

(d) Improved edge representation (b-c).

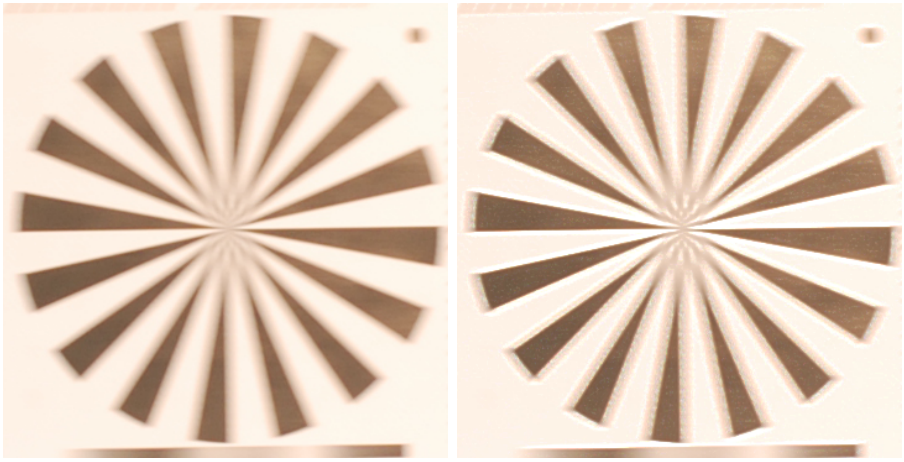
Figure 5.3: Unsharp mask image enhancement principle.

An edge in a blurred image is characterised by a softer gradient (Figure 5.3(b)) than the original sharp edge (Figure 5.3(a)). However, when applying a low-pass filter to the blurred image, the gradient becomes even flatter (Figure 5.3(c)). Subsequent calculation of the difference between a blurred gradient and the low-pass filtered gradient, results in a contrast enhanced and sharper appearing gradient (Figure 5.3(d), Code 5.1) (McHugh, 2015; The GIMP Help Team, 2015).

Listing 5.1: Unsharp mask method

```
1  //Function to apply unsharp mask approach. Input is only the blurred image.
2  Mat Funsharp(Mat in){
3          //http://stackoverflow.com/questions/19890054/
               how-to-sharpen-an-image-in-opencv
4          //Definition of variables to store image matrix.
5          Mat out;
6          //Re-blurring the input image with 3x3 kernel and high standard deviation for
               Gaussian equation.
7          GaussianBlur(in, out, Size(), 11);
8          //Calculating weighted sum of original and re-blurred image. The weighting of
               the input image is 1.5 and -0.5 for the re-blurred image
9          addWeighted(in, 1.5, out, -0.5, 0, out);
10         //Returning the enhanced image to the calling function.
11         return(out);
12         }
```

This process is not real image deblurring in terms of the understanding of image deconvolution,



(a) The original image. (b) The enhanced image.

Figure 5.4: Unsharp mask image enhancement example.

but it improves the image, enables better photogrammetric processing and improves further deblurring. It is possible to see in Figure 5.4 that the blur in the enhanced image and in the original image is still similar, but the improved contrast makes the image appear 'sharper'. However, this approach appears to assist subsequent photogrammetric processing (Table 6.4).

5.1.1.5 Summary

There are many image deblurring algorithms available, most utilising the original Wiener and Lucy-Richardson deconvolution and the basic idea of using probability theory to deblur images. However, these methods exhibit problems such as suppressing ringing artefacts efficiently and are less successful if the image is noisy. Some algorithms may return acceptable results, but they are often computationally expensive, take too long to deblur one image and require a higher performance processor. Most deblurring methods aim to simply restore a visually acceptable image. Photogrammetry requires a deblurring method which restores points to their correct positions within an image, in order to keep accurate image measurements.

5.2 Processing blurred images

After detecting if an image is blurred or not, the blurred image can be excluded from further processing. An alternative to exclusion is image deblurring to restore the image and then re-use for subsequent photogrammetric procedures. It is also important to minimise the number of excluded images to maintain a strong image configuration for precise coordinate calculations. There are several possible approaches available to deblur a photogrammetric image. One approach that can be used takes advantage of the image overlap available with UAV image sequences. However, this requires the ability to detect the overlap precisely, which is not always possible in the case of extensive blur. In such cases another method is required to restore an image in a way that they can be used for subsequent photogrammetric processing.

5.2.1 Frequency transfer method using image overlap

The frequency transfer method is one appropriate approach, which is similar to that used by Jung et al. (2009) (Figure 5.5). This approach was implemented and will be described in greater detail here. The method uses a sharp(er) overlapping image to deblur a blurred image. Initially, the image overlap between the sharp and blurred image has to be determined using feature detection or some other method. If the determination of image overlap fails, this method cannot be applied and other algorithms have to be used (Section 5.2.2).

Establishing the image overlap using feature detection and matching involves matching a number of features in both the sharp and blurred image. After successful establishing the relationship between the two images a perspective transformation is applied to the blurred image. Then both, the sharp and the transformed blurred images are cropped to just the overlapping area. Code sniplet 5.2 shows the calculation of parameters for a perspective transformation which was implemented in the developed program.

Listing 5.2: Perspective transformation parameter from matched features

```
1 //Following 'Feature detection and matching' code sniplet.
2 //Variables for key points/features.
```

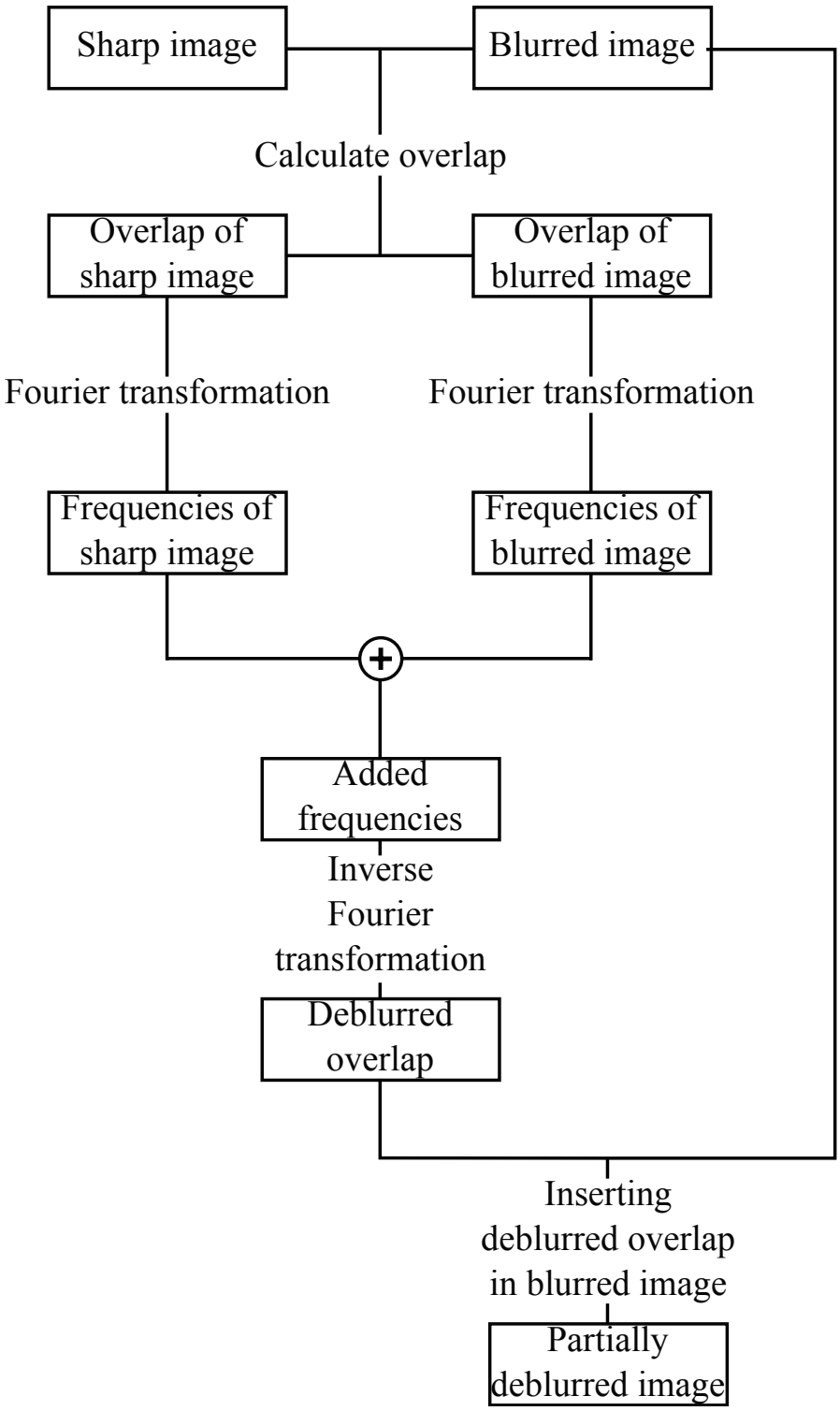


Figure 5.5: Flow of blur correction using Fourier transformation.

```
3  std::vector<Point2f> sharpvector;
4  std::vector<Point2f> blurredvector;
5  //Looping through all matches and separating features for sharp and blurred image.
6  for(int i=0; i<match.size(); i++){
7          //Get the keypoints from the good matches
8          sharpvector.push_back(feature_sharp[match[i].queryIdx].pt);
9          blurredvector.push_back(feature_blurred[match[i].trainIdx].pt);
10         }
11 //Calculating the perspective transformation parameter based on RANSAC method.
12 Mat H=findHomography(sharpvector, blurredvector, CV_RANSAC);
```

After determining the transformation parameter the overlapping image areas were calculated based on Code 5.3.

Listing 5.3: Calculating overlap area from perspective transformation parameters

```
1  //Function to calculate the overlapping area. Input is the matrix of the blurred image
       and the transformation matrix.
2  Rect Froi1(Mat image, Mat H){
3          //Variable for the rectangle representing the overlapping area.
4          Rect roi;
5          //Variable for the corners of the rectangle.
6          std::vector<Point2f> obj_corners(4);
7          //Variable for top and bottom point of the rectangle.
8          Point top;
9          Point bottom;
10         //Setting rectangle as image corners.
11         obj_corners[0]=cvPoint(0, 0);
12         obj_corners[1]=cvPoint(image.cols, 0);
13         obj_corners[2]=cvPoint(image.cols, image.rows);
14         obj_corners[3]=cvPoint(0, image.rows);
15         //Variable for the transformed corners.
16         std::vector<Point2f> scene_corners(4);
17         //Perspective transformation of the blurred image corners.
18         perspectiveTransform(obj_corners, scene_corners, H);
19         //Test if transformed coordinates are in or beyond blurred image boundaries.
               This needs to be done to define the exact area that is available in the
               image and not also the area outside the pictured area.
```

```
20          if(scene_corners[0].y>=scene_corners[1].y){
21                  top.y=scene_corners[0].y;
22                  }
23          else{
24                  top.y=scene_corners[1].y;
25                  }
26          if(scene_corners[0].x>=scene_corners[3].x){
27                  top.x=scene_corners[0].x;
28                  }
29          else{
30                  top.x=scene_corners[3].x;
31                  }
32          if(scene_corners[2].x<=scene_corners[1].x){
33                  bottom.x=scene_corners[2].x;
34                  }
35          else{
36                  bottom.x=scene_corners[1].x;
37                  }
38          if(scene_corners[2].y<=scene_corners[3].y){
39                  bottom.y=scene_corners[2].y;
40                  }
41          else{
42                  bottom.y=scene_corners[3].y;
43                  }
44          //Saving the corner coordinates of the overlapping area.
45          roi.x=top.x;
46          roi.y=top.y;
47          roi.width=bottom.x-top.x;
48          roi.height=bottom.y-top.y;
49          if(roi.height>image.rows){
50                  roi.height=image.rows-1;
51                  }
52          if(roi.width>image.cols){
53                  roi.width=image.cols-1;
54                  }
55          //Returning the overlapping area to the calling function.
56          return(roi);
```

```
57          }
```

Subsequently the perspective transformation can be conducted and the transformed image can be cropped to just the overlapping area (Code 5.4).

Listing 5.4: Warping the image using perspective transformation

```
1  //Function to warp the blurred image. Input are the blurred image matrix, the
       transformation matrix and the rectangle defining the overlapping area.
2  Mat Fwarp1(Mat image, Mat H, Rect roi){
3          //Variable for the perspective transformed overlapping area.
4          Mat matched;
5          //Size of output image required for warping function. Size set to same as
               original.
6          cv::Size size(image.cols, image.rows);
7          //Warping the image. Interpolation method is cubic interpolation.
8          warpPerspective(image, matched, H, size, INTER_CUBIC |
               WARP_INVERSE_MAP, BORDER_CONSTANT);
9          //Cropping the transformed image to the overlapping area.
10         matched=matched(Rect(0, 0, roi.width, roi.height));
11         //Returning the overlapping image.
12         return(matched);
13         }
```

Then the images are transferred to the frequency domain using a Fourier transformation. A disadvantage with the frequency domain approach is that any frequency operation is only able to work on single channel images, which makes it necessary to apply this method on each channel separately or to use a grey scale image instead of a three channel colour image (Code 5.5).

Listing 5.5: Fourier transformation

```
1  //Function for Fourier transformation. Input is the overlapping image matrix.
2  Mat Ffft(Mat image){
3          //Variable for output of Fourier transformed image matrix
4          Mat out;
5          //http://stackoverflow.com/questions/10269456/
               inverse-fourier-transformation-in-opencv
```

```
6         //http://docs.opencv.org/doc/tutorials/core/discrete_fourier_transform/
              discrete_fourier_transform.html
7         //Converting the RGB input image to a grey scale image.
8         Mat in;
9         cvtColor(image, in, CV_BGR2GRAY);
10        //Buffer variable.
11        Mat padded;
12        //Find optimal image size for Fourier transformation.
13        int m=getOptimalDFTSize(in.rows);
14        int n=getOptimalDFTSize(in.cols);
15        //Fourier transformations is more efficient for images of certain sizes. To
              perform a fast transformation the input image is reduced to this size.
16        copyMakeBorder(in, padded, 0, m-in.rows, 0, n-in.cols,
              BORDER_CONSTANT, Scalar::all(0));
17        //Fourier transformation return real and complex values which need to be stored.
18        Mat planes[]={Mat_<float>(padded), Mat::zeros(padded.size(), CV_32F)};
19        Mat complexI;
20        merge(planes, 2, complexI);
21        //Performing Fourier transformation.
22        dft(complexI, out, DFT_SCALE|DFT_COMPLEX_OUTPUT);
23        //Returning result of Fourier transformation to calling function.
24        return(out);
25        }
```

It is well established that high frequencies are absent in blurred images (Lelégard et al., 2012) (Section 4.1.2). The absence of high frequencies can be compensated by enhancing the blurred image using high frequencies extracted from the sharp image (Code 5.6).

Listing 5.6: Adding frequencies

```
1  //Function to add frequencies of two images. Input are the frequency images of sharp
       and blurred image.
2  Mat Faddfft(Mat fft1, Mat fft2){
3         //Variable to store the added frequencies. At first similar to the blurred
              frequency image.
4         Mat out=fft2.clone();
5         //Looping through all elements of the frequency matrix.
6         for(int i=0; i<fft1.size().height; i++){
```

```
7                 for(int j=0; j<fft1.size().width; j++){
8                         //Testing if amplitude in the sharp image is larger than in the
                              blurred image.
9                         if(fft1.at<Vec2f>(i, j)[0]>fft2.at<Vec2f>(i, j)[0]){
10                                //If sharp image has larger amplitude then writing
                                      sharp image amplitude to output matrix.
11                                out.at<Vec2f>(i, j)=fft1.at<Vec2f>(i, j)[0];
12                                }
13                        else{
14                                //If sharp image has smaller or equal amplitude then
                                      writing blurred image amplitude to output matrix.
15                                out.at<Vec2f>(i, j)=fft2.at<Vec2f>(i, j);
16                                }
17                        //Repeating procedure for the complexe values.
18                        if(fft1.at<Vec2f>(i, j)[1]<fft2.at<Vec2f>(i, j)[1]){
19                                out.at<Vec2f>(i, j)[1]=fft1.at<Vec2f>(i, j)[1];
20                                }
21                        else{
22                                out.at<Vec2f>(i, j)[1]=fft2.at<Vec2f>(i, j)[1];
23                                }
24                        }
25                }//End of loop.
26        //Returning added frequency domain matrix to calling function
27        return(out);
28        }
```

Afterwards the enhanced frequency domain can be transformed back to the spatial domain (Code 5.7).

Listing 5.7: Inverse Fourier transformation

```
1 //Function for inverse Fourier transformation. Input is the Fourier domain matrix
      containing the added frequencies.
2 Mat Fifft(Mat fft){
3         //http://stackoverflow.com/questions/10269456/
              inverse-fourier-transformation-in-opencv
4         //Variable for spatial domain image matrix.
5         Mat out;
```

```
6         //Inverse Fourier transformation.
7         dft( fft , out, DFT_INVERSE|DFT_REAL_OUTPUT);
8         //Returning spatial domain image to calling function.
9         return(out);
10        }
```

If a grey scale image is used, the deblurred image can be used to replace the intensity channel of the originally blurred image. Replacing the intensity channel is based on the idea of pan sharpening (Laben and Brower, 2000). Pan sharpening replaces the intensity channel of a colour image with the intensity channel of a high resolution grey scale image to achieve a higher resolution colour image.

After the frequency operations, the overlapping area needs to be transformed using an appropriate rotation, translation, scale and shear. Due to the fact that this deblurring method is only deblurring the overlapping areas and not the complete image, the deblurred area of the image needs to replace the overlapping area in the blurred image. The result is a partially deblurred image (Figure 5.6).

Figure 5.6 shows the effect of Fourier based deblurring. It is possible to see that the deblurred area of the image is much clearer and sharper than the area that has not been deblurred. The photogrammetric targets can be clearly recognised, the target codes can be correctly identified and the boreholes in the ceiling panels appear sharp. The inset shows a magnified comparison between the deblurred upper part and the bottom part of a target which contains a camera displacement of 1.03 mm. This direct comparison visualises how effective this deblurring approach is.

The frequency domain approach does not require knowledge about the path of camera displacement that caused the blur. However, as it requires the detection of overlapping areas, it is only applicable for camera displacements that do not significantly disturb feature detection and matching. Larger camera displacements, which exhibit a more complicated blur path can significantly influence the calculation of overlapping areas and make the calculation of the overlap impossible. If it is not possible to calculate the overlap when more advanced methods for image deblurring are required.

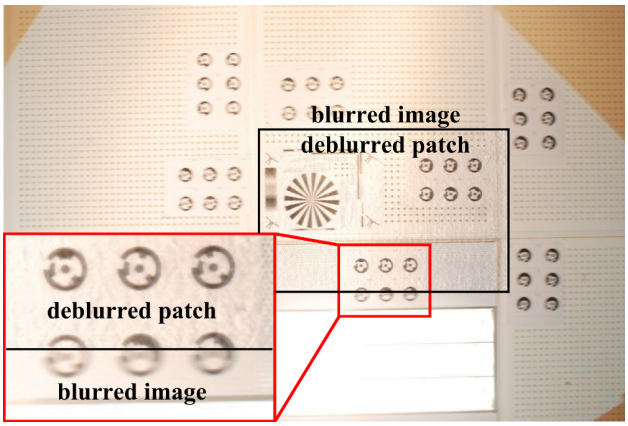


Figure 5.6: Deblurring using Fourier approach on an image with 1.03 mm camera displacement.

5.2.2 Edge shift - an overlap independent method

If image registration is unsuccessful, or no sharp image overlap is available then the frequency transfer method cannot be used and a different approach needs to be applied. A second method developed in this study focused on correcting edges in images, which represent an important visual component and are essential for target recognition. Edges indicate if an image is blurred or sharp (Chen et al., 2011) and help to identify how much an image is blurred. This method requires knowledge of a bur kernel, or PSF to deblur an image. However, it was found that existing deblur algorithms use the blur kernel to restore a colour image aesthetically. The developed edge shifting approach however, focuses on high contrast edges, but ignores colour information and image aesthetic (Figure 5.7).

High contrast edges used by this algorithm can include photogrammetric targets, which are often used for photogrammetric applications. In sharp images, targets are normally detected, identified and measured automatically and subsequent image registration can be conducted. Targets are designed to provide high contrast relative to the background, which makes them perfectly suitable for target detection procedures. Blur degrades these important high contrast edges. The degrading of the edge due to blur causes the gradient to decrease, but also causes a displacement of the edge, which is shifted by the size of the blur. Figure 5.8(a) represents the edges of photogrammetric targets in a sharp image, but in a blurred image these edges are displaced (Figure 5.8(b)). The decrease of

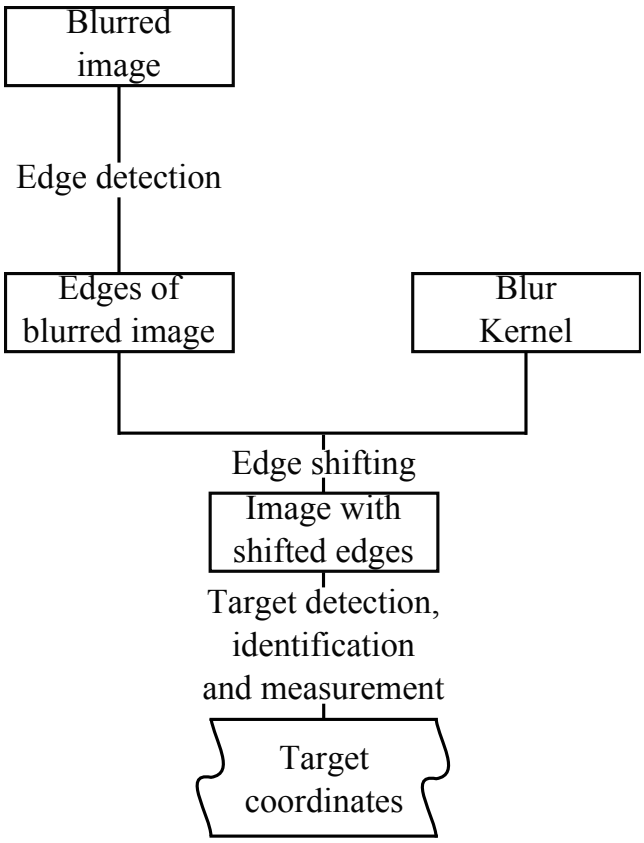
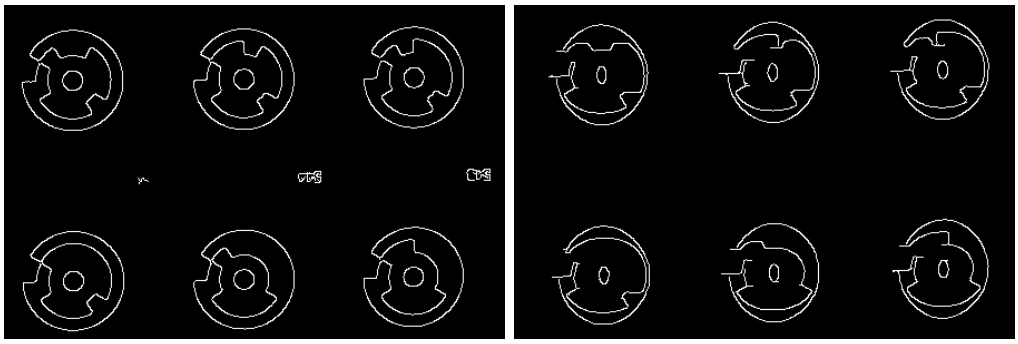


Figure 5.7: Flow of blur correction using edge shift.



(a) Edges of photogrammetric targets in a sharp image.

(b) Displaced edges of photogrammetric targets in a blurred image.

Figure 5.8: Displacement of edges in blurred images.

the gradient and the displacement of the edges can prevent automated photogrammetric processes. However, even in blurred images, high contrast edges remain detectable, provided they still appear stronger than other edges.

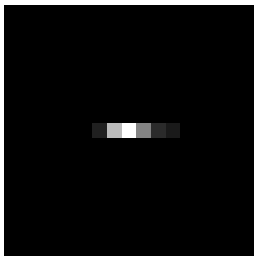


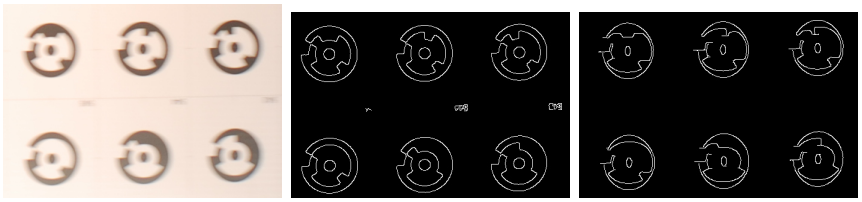
Figure 5.9: Blur kernel created for blurred image.

The algorithm developed in this research aims to shift the displaced edges back to their original position, providing the possibility to automatically detect the target and identify the target number based on the target code. With identified target number and measurement of the target, the calculation of image overlap or the calculation of coordinates should remain possible.

To achieve the repositioning of edges to their original sharp positions the blur kernel or PSF has to be known. Calculating the blur kernel is possible using a range of algorithms and the algorithm developed by Shan et al. (2008) was implemented in this study. Unfortunately, the full algorithm did not return any useful deblurring results and deblurred images could not be used for further processing. Even after correspondence with the author it was not possible to create an image that was acceptable deblurred. However, the returned blur kernel (Figure 5.9) appeared to be correct and was used to develop the novel edge shifting method that will be presented here. It should be recognised that blur kernels can also be produced by precise IMUs measurements or other image processing algorithms.

A high pass filter was initially applied to the blurred image, to detect the edges displaced, due to the original camera movement during image acquisition (Figure 5.8(b)). To restore the edges to their correct position, these edges needed to be displaced backwards along the path taken during the blurring process. This path is represented by the blur kernel.

To recover the original positions of edges each edge pixel had to be processed. The direction of movement to restore the original position of the pixels has to be identified by analysing the gradient of the edge. Figure 5.10 shows the blurred image (Figure



(a) Photogrammetric targets in a blurred image.

(b) Edges of photogrammetric targets in a sharp image.

(c) Edges of photogrammetric targets in the blurred image.

Figure 5.10: Direction of displacement of blurred edges.

5.10(a)), the edges of the sharp (Figure 5.10(b)) and blurred image (Figure 5.10(c)). It can be seen that camera displacement during exposure appears to displace the edges towards the darker pixels that represent the target bit pattern. Therefore, to restore the original positions the edge has to be displaced in the opposite direction towards the brighter area.

After establishing the direction of movement it has to be established how far the edge pixel has to be moved to reach its desired position. This can be established by the blur kernel which provides information about the extent of camera displacement and the displacement of the edge. However, moving the edge the full extent of the kernel path creates incorrect results, this is due to the fact that both, the edge on the left of an object as well as the edge on the right of an object are moved in opposite directions. Moving both edges by the full extent would therefore return incorrect results. Instead of moving them the full extend are they moved just half the length of the blur kernel.

After repositioning the edges to their original positions, it is possible to recognise the round targets. However, due to 'ripping' apart the left and the right edge of an object, the edges are no longer connected (Figure 5.11(a)). This problem needs to be detected and the holes filled to ensure that the contours of the circle are available. The detection can be done by finding edge pixels which are moved in both directions. This is indicated by brighter areas on both, the left and right of the pixel. After detection of such a pixel, the pixel can be displaced and the distance between the movement to the left and to the right filled with new edge pixels (Figure 5.11(b)). After filling the gap the contours

(a) Targets with holes due to shifting.

(b) Shifting holes filled.

Figure 5.11: Holes cause by 'ripping' apart edges.

are closed and target identification can be conducted. In this research target detection was carried out using the contours of the targets. If these match a specific size and shape they are accepted as targets, whilst objects that do not match a certain size or shape are excluded. The target were defined to be circular in this study. Each circular contour matching a certain size was then analysed and the binary target code read and translated to a decimal number to identify the target. To measure the target centre, the area of the target was selected and the first order image moment (gravity centre) was calculated.

With the target identified and the coordinate measured, it is possible to conduct subsequent photogrammetric procedures. For example, it is possible to incorporate the coordinates in a bundle adjustment to calculate 3D coordinates for the targets. Unfortunately, the edge shifting approach returns just an edge image without colour information, which limits certain use of the deblurred image. However, it would also be possible to achieve image registration, which would then enable the use of the frequency transfer method to generate a deblurred colour image.

5.3 Results and limits of correction

To prevent the negative influence of blurred images in post processing, blurred images should be detected and deblurred. Even if image blur is small it can be too severe for accurate detection and identification of photogrammetric targets and ground control points. A method to improve the image quality through 'deblurring' is therefore valuable. Two approaches have been investigated and their efficacy is examined here.

5.3.1 Fourier transformation approach

The frequency transfer method can correct some blur and is practicable if overlap is available and one image is sharp. Incorporating high frequencies derived from a sharp image into the blurred image appears successful. However, if the image used for deblurring is itself blurred, then although the enhanced result maybe better than the original blurred image, it will contain at least as much blur as the 'sharper' image originally contained.

PhotoModeler was used in this research to test the quality of deblurred images, specifically to detect and identify targets. An image with 1.03 mm camera displacement was deblurred initially using the Fourier approach that had been developed (Section 5.2.1). Subsequent target detection was executed and proved to be successful as *PhotoModeler* successfully detected 51 of 54 targets. In addition 49 of the 54 targets were correctly identified. In comparison, it was not possible to detect any targets in the original blurred image, demonstrating a major improvement (Figure 5.12, Section 3.2).

Feature detection and matching was also tested. A SURF algorithm was applied on the deblurred image, which detected 11,641 feature points. After matching and filtering incorrect features, 291 features were accepted as correct (Figure 5.13). This value was six times greater than that achieved using the original blurred image, demonstrating that deblurring can bring major improvements.

Adding frequencies to a blurred image introduces noise after transforming back from the frequency to the spatial domain. The insets in Figure 5.12 show a target sheet

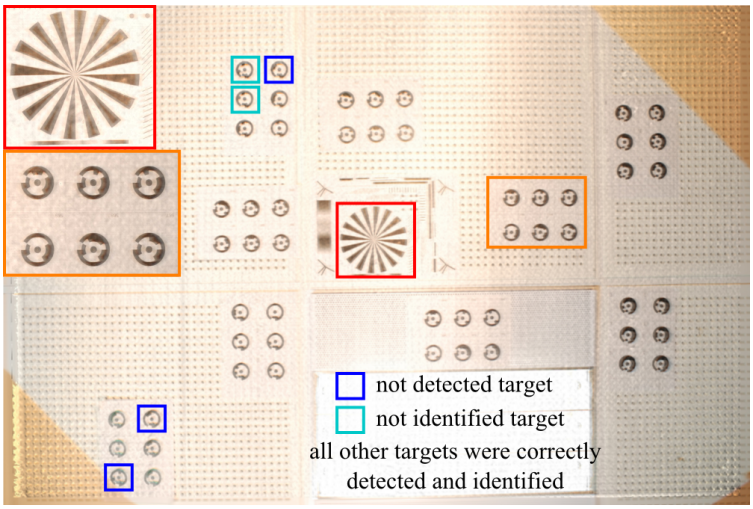


Figure 5.12: Detected and identified targets in deblurred image.

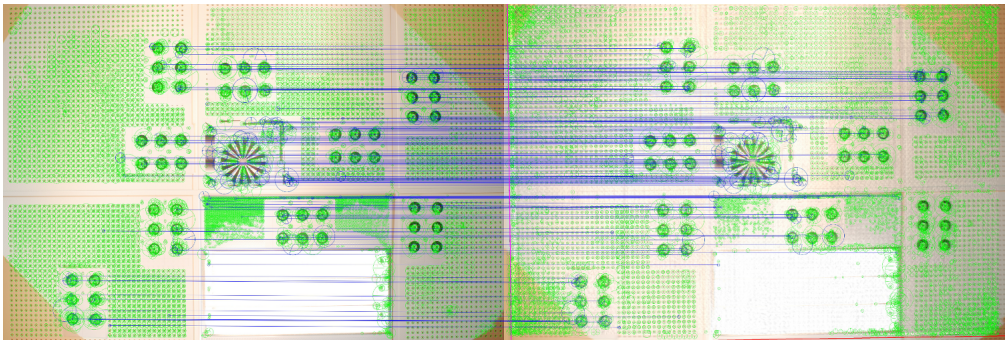


Figure 5.13: Detected and matched features in deblurred image.

and the Siemens star which both show that there is a pattern with brighter and darker 'blobs'. This pattern is insufficient to influence post processing, but does affect the visual appearance. This might prevent the use of the image for image mosaics or other visual products.

After joining the frequency domain images, the final stage involving image transformation can cause a 'smearing' effect. The image transformation is necessary to regain the correct position through, rotation, shear, scale and shift, but the interpolation of the correct pixel intensities for the rectified image can cause noticeable image degradation. Furthermore, deblurring using overlapping images acquired from a moving platform will always be limited by the sharpness of the image used for deblurring. Any slight

blur in the overlapping image will be replicated in the final result because the critical high frequencies are missing and cannot be integrated. A complete deblurring can only be generated if there is a perfectly sharp image containing all high frequencies. This frequency transfer is based on the original idea of Jung et al. (2009). However, the newly developed method uses images which are not highly overlapping or taken from the same camera point of view, which makes it widely applicable for aerial imagery.

Furthermore, it should be recognised that the Fourier approach is strictly only applicable for flat areas, which do not exhibit significant height differences. In cases of significant height variations, like high rise buildings or opencast pits, offsets will be generated due to relief displacement (Campbell and Wynne, 2012). A way to solve this would split the image into many small patches, which take account of the different heights.

This approach developed derived imagery suitable for further photogrammetric processing. However, it cannot be guaranteed that subsequent measurements in the deblurred image are of high or even sub-pixel accuracy. The geometric image transformation leads to errors, which depend on the accuracy of the transformation parameters. However, coordinate measurements are more likely to be successful using automated tools. One approach to deblur an image that is geometrically correct could be the edge shift approach.

5.3.2 Edge correction approach

The edge shifting approach appears to provide a method to restore high contrast edges in blurred images. It provides the opportunity to deblur targets and to detect and identify them (Figure 5.14). In Figure 5.14 it is possible to see that all 54 targets were detected. Out of these 54 targets, 47 were assigned with their correct label, 2 with an incorrect label. This is in contrast to the result achieved with *PhotoModeler* on the original blurred image, which detected only 3 targets and identified all of them incorrectly. This result shows a significant improvement (Section 3.2).

However, one of the main problems for this method is, that it requires a blur kernel. The blur kernel was not established in the developed algorithm but instead derived by using

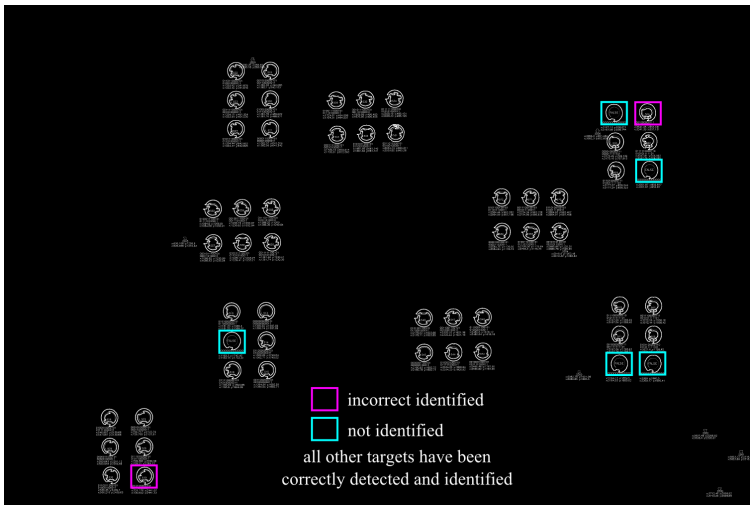


Figure 5.14: Edge shifting result for image with 0.53 mm camera displacement. 54 targets detected; 49 targets identified; 2 identified incorrectly; 47 identified correctly with their correct labels.

the algorithm of Shan et al. (2008). This can be solved by assuming a simplified blur kernel that is continuous and linear, or by using IMU and GNSS information recorded by the UAV. It is also possible to use blur kernels calculated by other algorithms.

If a blur kernel can be provided then the algorithm is able to deblur high contrast edges. High contrast edges are often used for image registration, as these can be clearly detected and measured in different images in a sequence. Standard deblurring methods based on the Wiener or Lucy-Richardson deconvolution, often produces ringing artefacts on high contrast edges, making automatic detection of targets and features challenging. In contrast, edge shifting does not produce these artefacts and makes automatic detection possible. It is also faster than standard deconvolution methods because it concentrates only on edges. However, the main problem is the loss of colour information, which would need to be restored afterwards using for example, a flood fill method. However, colour information is not always required for photogrammetric processing, which makes the absence of colour less important. More significant problems are the many special cases and exceptions, which can occur when applied to a wider range of imagery.

Handling these exceptions creates several challenges, such as the generation of gaps in the shapes caused by the shifting process. Shifting of two edges in different directions appears to 'rip' them apart and leaves holes in the structures (Section 5.2.2). Furthermore, edges

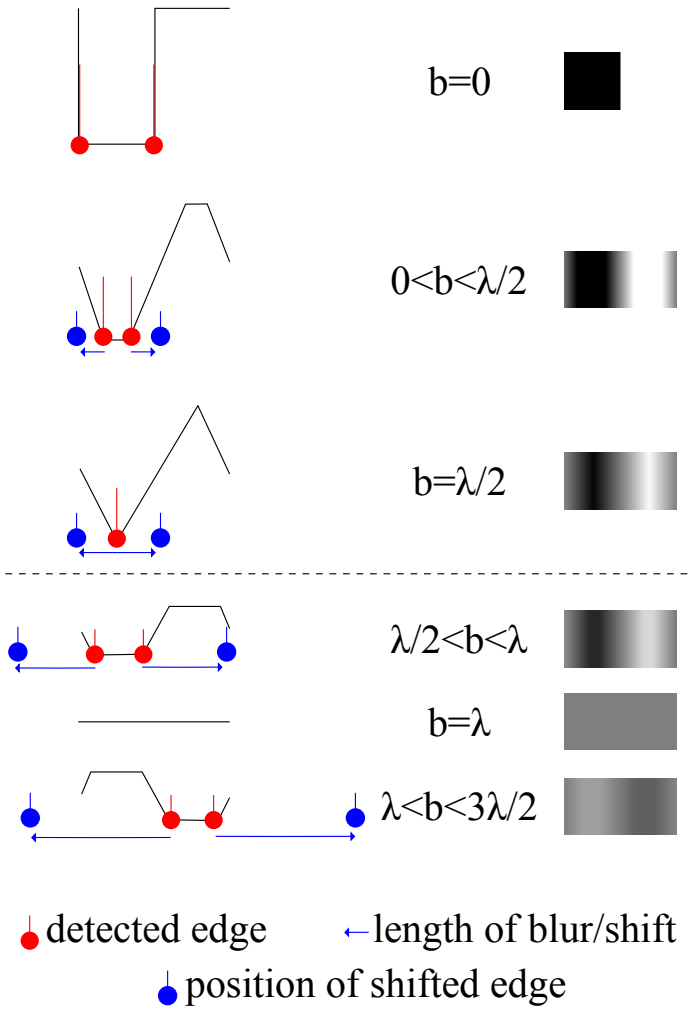


Figure 5.15: Edge shifting in blurred images. The left hand side exhibit waves degraded by different amounts of blur specified by the equation in the centre column. The right hand side shows real image examples.

which have completely disappeared cannot be recovered. Deblurring targets, which are smaller than the introduced amount of blur is problematic because of the number of special cases that need to be considered (Figure 5.15). Figure 5.15 shows the intensities of one pixel line on the left side. The pixels are either completely black or white. With increasing blur the intensities of the pixels change and this affects the positions of the edges. While the blur b is smaller than half the length of the plateaus (λ)only the edges change their pixel intensities. In the special case of $b = \lambda/2$, just one edge is detected which would be separated in opposite directions. With $\lambda/2 < b < \lambda$ the variation in the intensity of the pixels decreases constantly. Dark pixels would become brighter and

bright pixels would become darker. There would no longer be any white or black pixels but only a range of grey values. Also the edges would change their positions again. Deblurring these edges by half the blur length would be incorrect. They would not have to be shifted half the blur length but only half the blur length (b) minus half the length of the original plateau (λ). The problem is detecting these edges. To detect them it would be necessary to test if there is an edge in the opposite direction to the shift. It would also require testing along the shift direction to see if there is another edge along the blur path. If so, then the shift would have to be shorter than half the blur length. An additional complication occurs if the blur length is larger than the original plateau size. Then the amplitude is inverted which has to be considered during the shifting process.

Deblurring real images with different edge intensities, directions and blur length remains more challenging than in the high contrast test image. Despite these unsolved problems it is possible to achieve edge shifting using photogrammetric targets. This enables subsequent, sub-pixel accurate measurement of target coordinates, which can be used for photogrammetry. Furthermore, it might be possible to incorporate different blur in different positions of an image by calculating localised blur kernels. This would be necessary for blurred objects located close to the camera, which appear more blurred compared to objects further away.

This newly developed edge shifting method is completely different to previously published methods, because it does not concentrate on restoring an aesthetically acceptable image using probability theory (Lucy, 1974; Richardson, 1972; Shan et al., 2008; Wiener, 1950), but only edges. This provides the advantages of being fast as well as recovering the edge information, which is critical for photogrammetric coordinate measurements. Furthermore, this approach does not consume too much computational power and provides fast results, which could be used for post processing. However, this approach remains very complex, due to the number of exceptions for shifting edges.

5.4 Discussion

The deblurring methods developed in this research have some similarities and differences to deblurring algorithms developed previously. The Fourier transformation approach is a non-blind deconvolution method using image overlap for deblurring. Using image overlap makes it different from other method, which use precise IMU measurements (Joshi et al., 2008), video cameras (Tai et al., 2008) fluttering shutters (Raskar et al., 2006) or colour channel dependent exposure times (Lelégard et al., 2012). Agrawal et al. (2012) uses image overlap for deblurring. However, image overlap is used in their research to establish a point spread function, which is different to the Fourier transformation method developed upon this research.

The method published by Jung et al. (2009) is based on Fourier transformation for deblurring. However, it is still different to the method developed here as it uses only images for deblurring that are completely overlapping. The algorithm developed upon this research is different to Jung et al. (2009) because it does not require images that are completely overlapping. The large overlap, which is around 80 % along track and 60 % across track, should provide enough image material to carry out effective deblurring using the frequency transfer method.

The edge shifting approach is a blind deconvolution method. However, it does not use the models developed by Lucy (1974); Richardson (1972) or Wiener (1950) and does not use the Bayes's theorem, from the field of probability theory (Equation 5.1). This makes the method developed in this study different to many other deblurring methods developed previously (Fergus et al., 2006; Jia, 2007; Whyte et al., 2010; Yuan et al., 2007). Many other methods for blind deconvolution such as 'Fast Image Deconvolution using Hyper-Laplacian Priors' Krishnan and Fergus (2009) or Michaeli and Irani (2014), use recurrence of image patches. These do not have any similarity with the edge shifting approach developed in this research.

This high complexity is also the reason why the edge shifting approach was not applied to the same image as used for the frequency transfer method. The frequency transfer method was carried out on an image with 1.03 mm camera displacement ($\lambda/2 < b$).

The edge shifting approach was carried out on an image with just 0.53 mm camera displacement ($b < \lambda/2$). However, comparing frequency transfer and edge shifting method directly would also not be sensible as the frequency transfer method requires a second overlapping image and the result of deblurring strongly depends on the quality of the overlapping image. The edge shifting method does not require an overlapping image hence is not dependent on the quality of it. However, edge shifting requires a blur kernel, which is calculated based on the image itself. The results of both algorithms are dependent on the quality of the additional input data (overlapping image, blur kernel) but also on the image content and complexity, which causes that either method can return a more acceptable result even for the same blurred image.

5.5 Conclusion

Deblurring blurred images is a challenging topic and despite the availability of established algorithms, none of them currently provides a fast and reliable deblurring algorithm suitable for photogrammetric applications. This arises from the desire to achieve images that are aesthetically pleasing rather than geometrically correct. Also the aim to achieve this without using information of overlapping images or only using highly overlapping images is disadvantageous for photogrammetry. Photogrammetry often provides large datasets with overlapping images and these additional resources can be included in the process of deblurring an image. The developed frequency transfer method has been shown to successfully create images that can be used in subsequent photogrammetric processing. Furthermore, it is an easy and fast method that does not require high end computers. The edge shifting method has also been shown to provide a fast and reliable method that returns results of an appropriate quality. Furthermore, this shows that the concept of shifting blurred edges back to their original position works, but the development of a complete algorithms that can consider all special cases requires expert programming knowledge. This edge shifting approach appears to be a viable solution to the problem of deblurring images.

6 Application to UAV imagery

Blur in images has been shown to disturb photogrammetric processing, but can be both detected and corrected, particularly for laboratory images. However, laboratory images are very different to real images captured during UAV image acquisition, especially the camera to object distance and with camera displacements causing blur being typically larger. The image content is also very different to laboratory images. To prove the applicability of the developed methods described in previous chapters it is important to test the algorithms using typical UAV imagery.

6.1 Processing

Processing photogrammetric images was conducted using a range of datasets, including multiple datasets created by UAVs and for close range applications. The aim was to analyse images that were not specifically generated for examining blur detection and deblurring algorithms.

This chapter is divided in two key sections. The first (Section 6.1.1) is about blur detection in real images, the second (Section 6.1.2) concentrates on deblurring of blurred images. The step of analysing if blur disturbs was not possible prior to blur correction. This is because there are no perfectly sharp images available, which could be used to analyse the difference between sharp and blurred images during post processing.

	Fixed wing	Rotary wing (quad-copter)
Number images	195	97
Size	12Mega Pixel	10MegaPixel
Camera	Canon IXUS 220HS	Canon PowerShot S90
Focal length	4mm	6mm
ISO	˜160	˜100
Shutter speed	1/500	1/320
Aperture	f/2.7	f/4
Date	7 October 2012	13 September 2012
Processing time	˜1.4 seconds per image	˜1.4 seconds per image

Table 6.1: UAV datasets used for case application.

6.1.1 Blur detection

Detecting blur using more realistic imagery was conducted using both aerial and close range datasets.

6.1.1.1 UAV datasets

Various UAV datasets acquired by different UAVs were processed, including both fixed wing and rotary wing UAVs equipped with different cameras and using a range of camera settings. Two datasets were analysed in more detail. The two datasets chosen were taken with different UAV platforms, because fixed wing UAVs are expected to have a certain amount of motion blur in each image due to the forward motion. However, rotary wing UAVs can hover in one position, enabling acquisition of images without forward motion. The rotary wing quad-copter was a kit set that used a *Canon IXUS 500 HS*, 10.1 Megapixel camera (Almond, 2013), while the fixed wing UAV utilised a *SenseFly Swinglet Cam* system (Table 6.1) (Almond, 2013).

Table 6.1 provides an overview of the datasets. Both image sequences were acquired on sunny days which allowed both fast shutter and film speeds ensuring optimal image quality. Fortunately, the low aperture setting and reduced depth of field had minimal impact because the terrain did not exhibit any significant height differences. Both datasets were acquired of rural salt marsh in Abbotts Hall Farm (AHF) Great

(a) Example image from fixed wing dataset. (b) Example image from rotary wing dataset.

Figure 6.1: Examples of real application UAV images.

Wigborough, Essex, on the North bank of the Salcott Creek, a tributary to the Blackwater Estuary (Figure 6.1) (Almond, 2013).

Rotary wing dataset The first dataset was acquired using the quad copter and the SIEDS values show a large range of values, which allows easy separation of sharp and blurred images (Figure 6.2). Figure 6.2 demonstrates that the SIEDS value ranges between 70 and 25 exhibiting a range of sharp and blurred images. To evaluate the calculated SIEDS values appear sensible a random sample of images were chosen for visual analysis. The four chosen examples show that overall, the dataset appears visually to be of good quality, which can be closer assessed in the insets (Figure 6.3). The insets are all located towards the periphery of the image (Section 6.1.1.1). Figure 6.3(a) is visually the most blurred image. An increasing SIEDS values equates to an improving visual quality. This demonstrates applicability of the calculated SIEDS value and how well the blur detection method works. Furthermore, the fast calculation time of less than 2 seconds on a consumer laptop, shows that this method is potentially applicable during the actual image flight. This would enable the operator to decide immediately whether it is necessary to repeat image acquisition.

Fixed wing dataset The dataset acquired with the fixed wing UAV appeared to exhibit very high image quality also. However, application of the developed blur detection

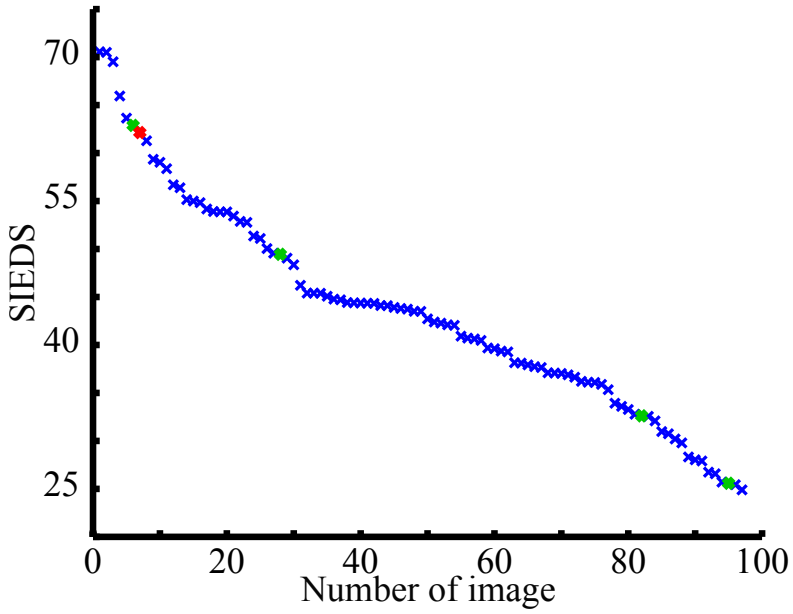


Figure 6.2: SIEDS calculated for rotary wing UAV images. The red mark shows the SIEDS value of Figure 6.1(b). The green marks show the SIEDS value of Figure 6.3

algorithm allowed different levels of sharpness to be detected (Figure 6.4). Figure 6.4 demonstrates that the SIEDS value ranges between 65 and 15 exhibiting a range of sharp and blurred images. To make the evaluation easier and meaningful three overlapping images were used with different SIEDS values (Figure 6.5).

Figure 6.4 suggest that all images are reasonably sharp. However, Figure 6.4 also implies that some images are of extraordinary quality, with SIEDS values above 60, while others are with lower quality, with SIEDS values around 15. The images chosen for comparison have SIEDS values suggesting that they are of similar quality, but the insets reveal differences between the images. The red inset in Figure 6.5(a) appears to be less sharp than in Figure 6.5(b), which was expected based on the result determined by the SIEDS value. This confirms that SIEDS represents the quality of an image. However, the same area in Figure 6.5(c) appears less sharp than Figure 6.5(b), even when the calculated SIEDS implies that this picture is of better sharpness than the two previous images. This can be explained by the position of the patch chosen for the inset. In Figure 6.5(b) the area of interest is towards the centre of the image while it is at the boundary in Figure 6.5(c). It might be that the camera and lens used to acquire the images introduces distortions at the image boundaries, causing the image to appear slightly

(a) Image with SIEDS=25.

(b) Image with SIEDS=32.

(c) Image with SIEDS=48.

(d) Image with SIEDS=63.

Figure 6.3: Example images for calculated SIEDS for rotary wing UAV images. The insets show a more detailed view.

out of focus. By choosing another patch (yellow inset), which is in the centre of Figure 6.5(c) and comparing it to the same area in 6.5(b), it reveals that Figure 6.5(c) is indeed sharper.

SIEDS is a value calculated for the entire image, which is problematic when specific areas of the picture are picked out. By processing just the small inset areas (marked red) the SIEDS value changes significantly. The SIEDS value for Figure 6.5(b) is the largest with 56, while Figure 6.5(c) (49) and 6.5(a) (47) are much smaller and the values much closer together. This matches with observations when judging the images visually (Table 6.2).

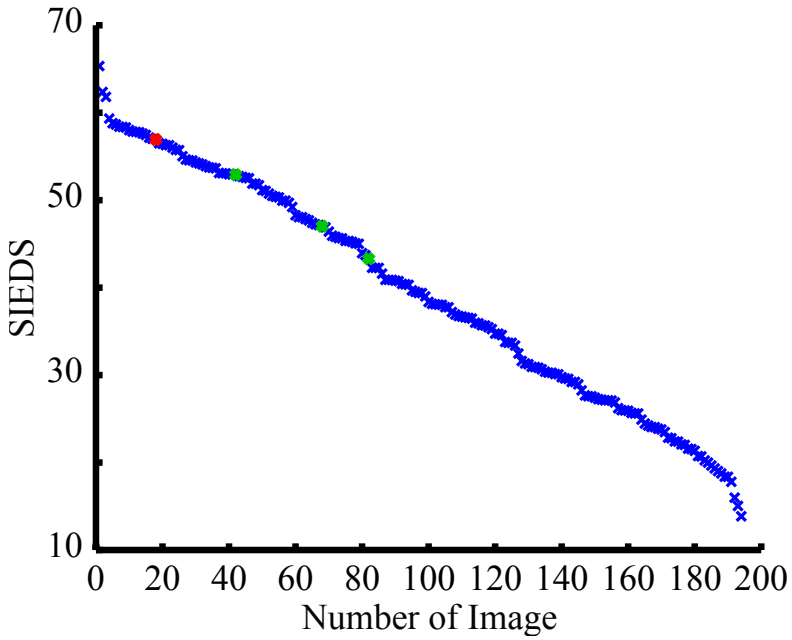


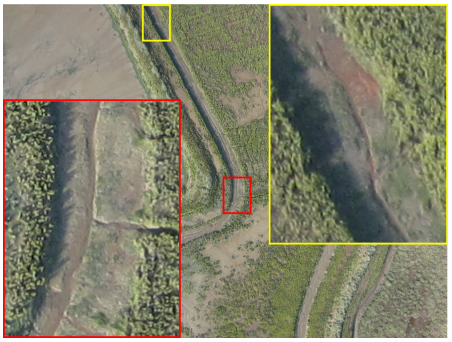
Figure 6.4: SIEDS calculated for fixed wing UAV images. The red mark shows the SIEDS value of Figure 6.1(a). The green marks show the SIEDS value of Figure 6.5

Image	SIEDS complete image	SIEDS inset
6.5(a)	42	47
6.5(b)	46	56
6.5(c)	53	49

Table 6.2: Comparison of SIEDS for complete image and image patches.

Inter-comparability SIEDS values are not inter-comparable. Dependent on the dataset is the absolute calculated value either large or small, which does not imply whether the image is sharp or not. Only the comparison within the dataset gives information of 'how blurry' an image is. Figure 6.6(a) has a SIEDS value of 50, which is smaller than the SIEDS value for Figure 6.6(b) with 58. However, also normalising does not return more reliable inter-comparability. For both examples in Figure 6.6 is the normalised value around 0.65, which would classify them as equally blurred. However, this cannot be confirmed visually.

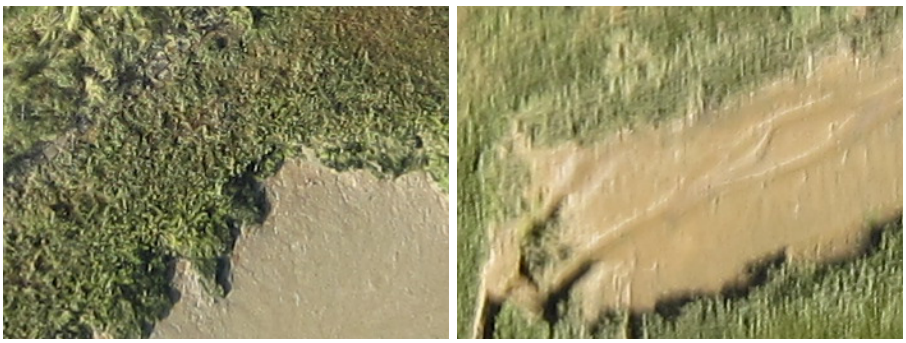
(a) Image with SIEDS=42.



(b) Image with SIEDS=46.

(c) Image with SIEDS=53.

Figure 6.5: Example images for calculated SIEDS for fixed wing UAV images. The insets show a more detailed view. The red insets show the same area in all three examples. The yellow inset was only available in two images.



(a) Image of fixed wing dataset. SIEDS=50. Normalised value=0.65

(b) Image of rotary wing dataset. SIEDS=58. Normalised value=0.65

Figure 6.6: Inter-comparability of UAV blur detection results. Neither the absolute SIEDS value nor the normalised value can be compared effectively.

	Terrestrial
Number images	111
Size	16Mega Pixel
Camera	NIKON D7000
Focal length	85
ISO	1000
Shutter speed	1/20
Aperture	f/5
Processing time	˜1.9 seconds per image

Table 6.3: Terrestrial datasets used for case application.

The lab based tests (Section 4) have demonstrated that detection of blurred images is possible. However, this section has proven that it is also possible to process aerial images and that the returned results can be used to separate sharp and blurred images. However, terrestrial images might be similar to aerial images, which makes testing the method on close range images an interesting opportunity.

6.1.1.2 Close range

The close range application involved an image sequence of vegetation, used to create a 3D model. The 3D model was required to analyse water flow characteristics and to investigate the potential of cheap endoscopic cameras to measure water flow, but was not part of this project. The images are unusual because they contain a large amount

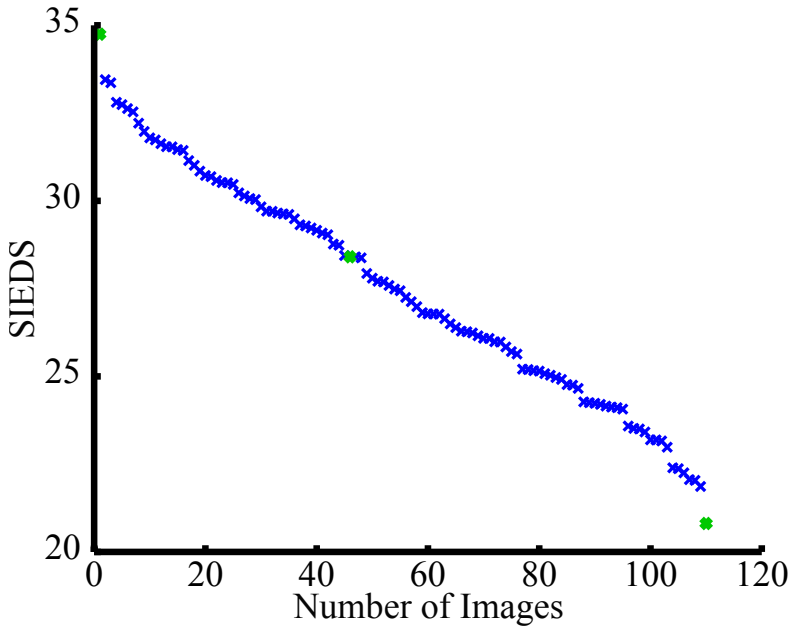
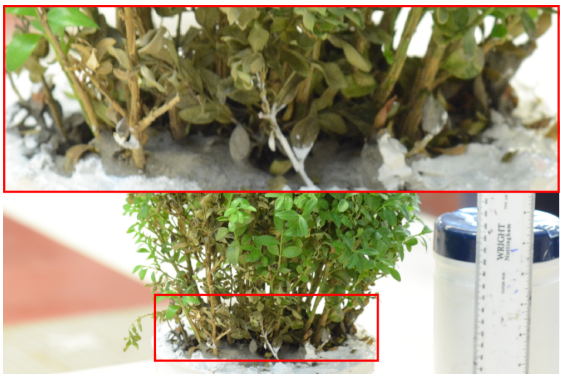


Figure 6.7: SIEDS calculated for terrestrial images. The green marks show the SIEDS value of Figure 6.8

of blurred background pixels while only the centre of the image, which contains the object, appears in focus (Figure 6.8(a)). This is caused by the dark light conditions in the laboratory, requiring a wide aperture opening which causes a narrow depth of field (Section 2.2.3). Furthermore, the images should not contain any motion blur because they are taken with a fixed camera on a tripod of a stationary, fixed object.

The calculation of the SIEDS value took less than 2 seconds per 16 Mega pixel image. Visualising the calculated SIEDS values created a similar graph to the one achieved with the UAV datasets (Section 6.1.1.1). However, the calculated SIEDS values exhibit a narrower range of just 14 units (Figure 6.7). This can be explained by the large areas which are out of focus, which is responsible for the lower SIEDS value.

(a) Image with SIEDS=21.



(b) Image with SIEDS=27.

(c) Image with SIEDS=35.

Figure 6.8: Example images for calculated SIEDS for terrestrial images. The insets show a more detailed view.

In the overview in Figure 6.7, the vegetation appears in focus, however, the insets reveals that the branches of the bush are not of the same level of sharpness. In Figure 6.8(a) the branches are less sharp than in Figure 6.8(b). Figure 6.8(c) appears to be sharpest, which confirms that the calculated SIEDS value represents the level of blur well.

6.1.1.3 Results of blur detection in real images

The visual confirmation of the results calculated by the blur detection program show that the algorithm returns useful results. The calculation speed is reasonably fast. This confirms that this method is applicable, not only in the office but potentially in the field during image acquisition. This allows the user to acquire new images if necessary and will then avoid the step of blur correction, which remains an unresolved topic in the community.

6.1.2 Blur correction

Although detecting and removing blurred images in a UAV sequence is valuable, the loss of individual frames may degrade the geometry of an image set, which is perhaps critical. An alternative to blurred image exclusion is blurred image correction, where instead of excluding blurred images it might be possible to correct them and restore photogrammetric geometry. To test the deblurring method developed in this study an image dataset was acquired using a fixed wing UAV (Figure 1.2(b) and Figure 6.9).

After executing blur detection, two overlapping images were chosen for further processing, each with different SIEDS values. In a first test, feature detection and matching was conducted on the dataset. It was possible to detect 2574 features in the sharp image but only 1711 features in the blurred image (Figure 6.10, Table 6.4). The difference may appear minor but is actually significant as the SIEDS values were similar. In a subsequent step, deblurring was carried out using an unsharp mask (Section 5.1.1.4) and the frequency transfer method (Section 5.2.1). To test if the images had improved, another feature matching process was conducted using the sharp and the deblurred images. It was found that the unsharp mask method returned a significant increase in

(a) Blurred image acquired by UAV (SIEDS=13.29)

(b) Sharp image acquired by UAV (SIEDS=14.85)

Figure 6.9: Two overlapping UAV images used for deblurring.

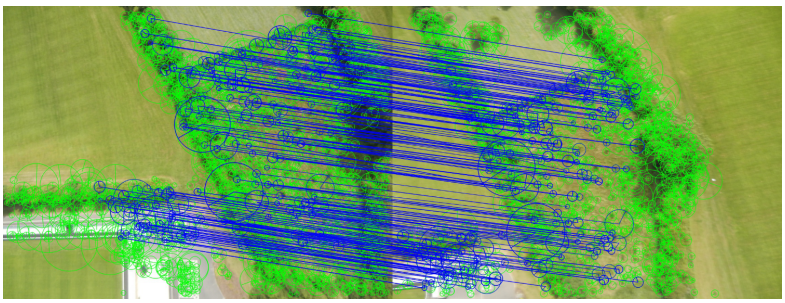
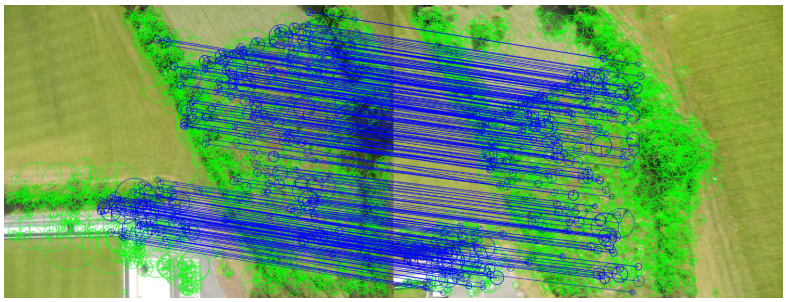


Figure 6.10: Feature matching in UAV images

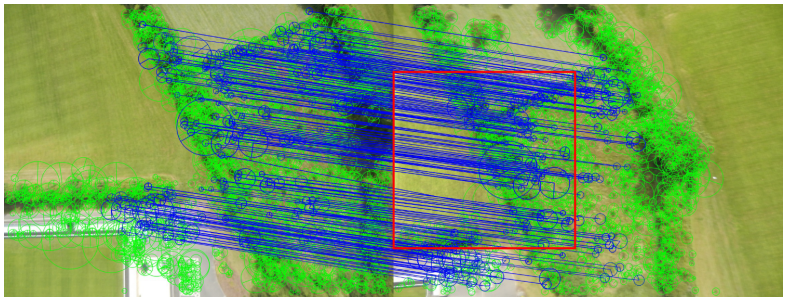
	Features detected	Accepted matched features
Sharp image	2574	
Blurred image	1711	406
Enhanced image	2859 (67% improvement)	478 (18% improvement)
Frequency transfer image	1896 (11% improvement)	415 (2% improvement)

Table 6.4: Feature detection in deblurred images.

feature points detected by 67 % to 2859 in the blurred image, which is more than the features detected in the supposed sharp image. The deblur approach using the frequency transfer method also increased the number of detected feature points by 11 % to 1896 features. The unsharp mask method appears to be much better than the frequency transfer method, at least in this case. However, the frequency transfer method was used on a much smaller overlapping area and not the complete image (Figure 6.11) and most of this area is occupied by green grass with limited features to detect (Figure 6.11(b)).



(a) Enhanced image using unsharp mask.



(b) Enhanced image using frequency transfer method. In red the enhanced area.

Figure 6.11: Feature detection and matching in enhanced images.

(a) Enhanced image using unsharp mask method.

(b) Enhanced image using frequency transfer method. In blue enhanced area.

Figure 6.12: Enhanced UAV images.

Figure 6.12 presents the deblurred images using the unsharp mask and frequency transfer methods. The insets in Figure 6.12(a) show that the unsharp mask image is still blurred and only has improved contrast to Figure 6.9(a). Figure 6.12(b) is sharper than Figure 6.9(a) because the frequency transfer method is more successful at deblurring the image.

Both methods are reasonably fast, requiring less than 3 seconds to process the entire 16 mega pixel image.

Image enhancement using the unsharp mask method improves the image without actually deblurring it. This therefore appears to represent an interesting approach for photogrammetric applications because it is simple, fast and applicable. By enhancing an image using the unsharp mask method, the ability to detect targets increases significantly, which makes photogrammetric post processing easier and faster. However, to deblur an image fully it still needs to be processed using a deblurring technique. The frequency transfer method appears to provide a method that is both fast and improves the image quality.

6.2 Conclusion

The practical work described in this application-focused chapter proves that both, blur detection and deblurring of UAV images is possible using the algorithms developed in this study. The fast processing time of the blur detection method and the representation of blur using a SIEDS value appears to be sufficient for real UAV image sets. Even high quality images can be assessed in more detail and enable the operator to identify the best quality images fully automatically. Manual filtering would be tedious and require many working hours. It would be prone to error and probably negatively affect the eyes of the operator. Full format aerial and terrestrial images can be assessed with this method and differentiated to exclude blurred images.

Although desirable, tests reveal that a SIEDS threshold value cannot exist as absolute values and are dependent upon image content, camera and lens. However, it is suggested that SIEDS is used to filter the lowest quality images. The user decides depending on location, geometry, coverage and image content if the image should be excluded. , or that a selection of images representing the range of calculated SIEDS values is examined by a human operator to establish a threshold for a dataset.

Not only is the detection of blurred images possible, but also the deblurring of images where overlapping images are available. Unfortunately, applying the frequency transfer

method to terrestrial images, with different points of view, can return incorrect results and is therefore not applicable. However, the image can be improved by image enhancement using the 'unsharp mask'. The unsharp mask approach is a method that could make photogrammetric post processing more successful, by increasing the contrast in the image. It is recommended that this easy and very fast method should be included in photogrammetric software.

7 Final conclusion

This thesis contributes to knowledge in the field of computer vision, image processing and photogrammetry. Blur is a widely researched topic in computer vision, however, most research concentrates on correcting blur and only rarely looks to determine the influence of blur on image processing, or in photogrammetric applications. Blur detection is rarely a totally independent topic and is often associated with blur correction. Mathematically blurred images are often used to develop and test these algorithms. However, for the research presented a newly developed method was used, which ensured that intentionally blurred images were available for development. Furthermore, these images were used to test existing algorithms to compare their results to results based on developments made in this research.

In this thesis blur was examined in three different ways. At first, it looked at and analysed how much blur 'disturbs' photogrammetric processing. It was found that even small camera displacements can cause problems in photogrammetric post processing. The main problem being that small camera displacements cause the failure of automatic target detection. It was found also that automation of photogrammetric procedures was problematic. However, semi-automatic or manual target detection remained possible and the accuracy of semi-automatic measured targets appeared to be acceptable, when there was only small camera displacement. Larger blur, which was clearly detectable by human eyes, created greater challenges even when targets were measured manually. Based on experiments using *PhotoModeler*, it was possible to state that blurred images compromise semi-automatic detection tools by introducing errors, which affected subsequent calculation of target coordinates, causing them to be inaccurate. This corruption effect was found to be predictable and it was possible to establish equations to calculate thresholds for acceptable camera displacements based on camera model, camera to object distance and

other camera parameter. This equation enables a better understanding and planning of UAV image flights.

A key focus in this thesis was the 'detection' of blurred images because detecting blurred images in larger image datasets is a major challenge when using UAV images. The process of looking at every image and detecting visually if an image is blurred is currently carried out manually. Manual blur detection requires a lot of time, is exhausting for the eyes and prone to error. An automatic algorithm to detect blurred images did not exist before and was developed as part of this research. The new algorithm increases processing speeds and is more reliable than manual human detection of blurred images. The approach used showed that it is possible to detect even small differences in blur, by computing a numerical measure of blur known as SIEDS. The high sensitivity enables the exclusion of blurred images even for very small camera displacements, however a limitation was that the processed images should show similar image content and needed to be acquired with the same optical system in order to avoid misinterpretation that could be caused by different SIEDS values. Despite these limitations the algorithm was also applicable for terrestrial images and could therefore be used on a much wider range of applications in the future. It was also possible to use the SIEDS value to calculate the actual extent of the camera displacement or to calculate in real time if an image is extensively blurred, or not. This allows the user the opportunity to obtain a replacement image during image acquisition and therefore provide only images with acceptable levels of sharpness.

The third and final area of research conducted was in the field of correcting blur as excluding blurred images was not always desirable. In order to retain images for processing, a number of techniques were used to enhance blurred images. One way presented was image enhancement using an 'unsharp mask'. This well-known approach was used to improve the image and make photogrammetric processing more successful. Another method used was to take the information provided by overlapping images and use the frequency transfer method. This method enhanced the blurred image by using the frequency content of the overlapping sharp image. Overlapping areas have to be detected first in order to use this technique, which can be problematic due to the existing blur. Furthermore, this method only corrected the overlapping area and the

degree of deblurring in the blurred image will depend strongly on the amount of blur that exists in the sharper overlapping image. Deblurring beyond the sharpness of the overlapping image was not practicable and more advanced and other time consuming image deblurring algorithms will be required.

The edge shifting approach developed in this research was found to be of value which appears to be a novel approach for image deblurring. It returned good results with the laboratory images that were tested but the approach has a high degree of complexity, which may limit its application. However, deblurring high contrast images and restoring photogrammetric targets was possible and did enable the identification and measurement of the target coordinates. In order to make this method more applicable for non laboratory applications, such as UAV images, more advanced programming is required.

7.1 Future work

This thesis contains new ideas and approaches for image blur detection and deblurring, which can be further progressed and improved. In this context, several computer algorithms were developed to detect whether an image is blurred and to correct blurred images. Unfortunately, due to time limitations, the developed algorithms should only be used by experienced users, because the algorithms were developed by an inexperienced programmer and might fail due to incorrect inputs or corrupt data. This limitation could be addressed in future work by improving the code, particularly catching and solving exceptions that might occur.

A specific development would be to analyse the connection between calculated SIEDS value and actual camera displacement. It was found that SIEDS and camera displacement can be related to one another by a modified damping function (Section 4.3.3). However, the relationship seems to be rather complex and dependent on a number of other factors. These factors could be determined and a calculation of actual camera displacement based on the SIEDS value could be made possible. If this link could be achieved then it would be possible to determine camera displacement values for each image. Calculating camera displacements for single images would then be possible. This would be of significant

interest to a wider consumer market and could lead to industrial collaboration.

The novel edge shifting technique was developed based on high contrast laboratory images. However, it is believed that it is possible to use this approach for real world images to restore edges. This would be particularly interesting for photogrammetric applications because they use edges for processing. Future research could analyse and improve this approach to reconstruct geometrically correct images. It might also be possible to restore colour images by deconvolving a colour image and enhancing it with a corrected edge image.

Finally, this research has achieved the original aim of analysing, detecting and correcting motion blur in digital images by accomplishing the objectives lined out in Section 1.2. Essential literature was reviewed, motion blurred images generated and their influence on photogrammetric processes was analysed. Furthermore, an algorithm for blur detection was developed and correction methods applied to them.

References

Adobe Developers Association, (1992), TIFF 6.0 Specifications. Technical report, Adobe Systems Incorporated, Mountain View, CA, US.
https://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf 12, 15

Adobe Systems Incorporated, (2015), Photoshop Elements Help. Internet, accessed: 04/05/2015.
https://helpx.adobe.com/photoshop-elements.html 19

Agence France-Presse, (2014), UN Surveillance Drone Crashes in Eastern DR Congo. Internet, accessed: 06/11/2014.
http://news.yahoo.com/un-surveillance-drone-crashes-eastern-dr-congo-201153425.html 1

Agisoft LLC, (2013), *Agisoft PhotoScan User Manual - Professional Edition, Version 1.0.0*. Agisoft LLC, 1st edition.
http://fieldofviewllc.com/wp-content/uploads/bsk-pdf-manager/27_AGISOFT PHOTOSCAN PRO USER GUIDE.PDF 134

Agrawal, A., Xu, Y. and Raskar, R., (2012), Multi-Image Deblurring. (8229244).
http://www.google.com/patents?hl=en&lr=&vid=USPAT8229244&id=8RogAg AAEBAJ&oi=fnd&dq=Multi-Image+Deblurring&printsec=abstract 140, 141, 163

Almond, F., (2013), *Using Unmanned Aerial Vehicles and Supervised Classification for Analysis of a Salt-Marsh Managed Realignment Site*. Master's thesis, Loughborough University. 166, 167

APS Dynamics Incorporated, (2015), APS 400 Electro-Seis. Internet, accessed: 03/06/2015.
http://www.apsdynamics.com/index.php?option=com_content&view=article&id=61%3Aaps-400-electro-seisr-&catid=38%3Aball-bearing-exiters&Itemid=63&lang=en 40

Associated Press, (2014), US Military Drone Makes 'Hard Landing' in Niger. Internet, accessed: 06/11/2014.
http://www.washingtontimes.com/news/2014/oct/21/us-military-drone-makes-hard-landing-in-niger/ 1

Atkinson, K.B., (2001), *Close Range Photogrammetry and Machine Vision*. 2nd edition, Whittles. 56

Bao, D.Q., (2009), Image Blur Metric. Internet, accessed: 10/06/2015.
http://www.mathworks.com/matlabcentral/fileexchange/24676-image-blur-metric 105, 106

Bascle, B., Blake, A. and Zisserman, A., (1996), Motion Deblurring and Super-Resolution from an Image Sequence. In: *European Conference on Computer Vision*, volume 1065, Springer Berlin Heidelberg.
http://www.springerlink.com/index/Y811H826Q1714346.pdf 136

Bäumker, M., Brechtken, R., Heimes, F.J. and Richter, T., (1999), Direkte Georeferenzierung mit dem Luftaufnahmesystem LEO. In: *Internationale Geodätische Woche*.

http://mobil.hochschule-bochum.de/fileadmin/media/fb_v/labore/photogrammetrie/Artikel/Veroeffentlichungen/Obergurgl_1999_Direkte_Georeferenzierung.pdf 29

Bay, H., Tuytelaars, T. and Gool, L.v., (2006), Surf: Speeded Up Robust Features. In: *European Conference on Computer Vision*, volume 3951, Springer Berlin Heidelberg.
http://link.springer.com/chapter/10.1007/11744023_32 26, 91

Bayer, B., (1976), Color Imagin Array. (3971065).
http://patft.uspto.gov/netacgi/nph-Parser?Sect2=PTO1&Sect2=HITOFF&p=1&u=/netahtml/PTO/search-bool.html&r=1&f=G&l=50&d=PALL&RefSrch=yes&Query=PN/3971065 10

BBC, (2014), UK to Deploy Armed Drones Against Islamic State in Iraq. Internet, accessed: 06/11/2014.
http://www.bbc.co.uk/news/uk-politics-29646699 1

Ben-Ezra, M. and Nayar, S.K., (2004), Motion-Based Motion Deblurring. *Pattern Analysis and Machine Intelligence*, 26(6).
http://www.ncbi.nlm.nih.gov/pubmed/18579930 36

Bethmann, F. and Luhmann, T., (2011), Least-Squares Matching with Advanced Geometric Transformation Models. *Photogrammetrie - Fernerkundung - Geoinformation*, 2011(2).
https://www.schweizerbart.de/papers/pfg/detail/2011/75743/Least_squares_Matching_with_Advanced_Geometric_Transformation_Models 57

Bidwell, A., (2014), Why Colleges Want Drones. Internet, accessed: 06/11/2014.
http://www.usnews.com/news/articles/2014/10/10/universities-use-drones-for-disaster-response-agriculture-and-energy-research 1

Boracchi, G., (2009), Estimating the 3D Direction of a Translating Camera from a Single Motion-Blurred Image. *Pattern Recognition Letters*, 30(7).
http://linkinghub.elsevier.com/retrieve/pii/S0167865509000312 98

Boracchi, G., Giusti, A. and Milano, P., (2007), Ball Position and Motion Reconstruction from Blur in a Single Perspective Image. In: *International Conference on Image Analysis and Processing*, IEEE.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4362762 74

Bradski, G. and Kaehler, A., (2008), *Learning OpenCV - Computer Vision with the OpenCV Library*. 1st edition, O'Reilly Media.
https://books.google.de/books?id=seAgiOfu2EIC&lpg=PA145&ots=hTE8cjfAOh&dq=opencv convolution equation&hl=de&pg=PA145#v=onepage&q=opencv convolution equation&f=false 21, 58

Brown, L.G., (1992), A Survey of Image Registration Techniques. *ACM Computing Surveys*, 24(4).
http://doi.acm.org/10.1145/146370.146374 26

Burger, W. and Burge, M.J., (2008), *Digital Image Processing*. 1st edition, Springer, New York, NY, USA.
http://books.google.co.uk/books?id=4gBUz_IkkSsC&lpg=PR1&dq=imageprocessing&hl=de&pg=PR4#v=onepage&q=image processing&f=false 17, 18, 19, 36

Campbell, J.B. and Wynne, R.H., (2012), *Introduction to Remote Sensing*. 5th edition, Guilford Press, New York. 159

Charlmond, B., (1991), PSF Estimation for Image Deblurring. *CVGIP: Graphical Models and Image Processing*, 53(4).
http://b.chalmond.free.fr/publi/CVGIP_PSF_1991.pdf 137

Chen, C., Chen, W. and Bloom, J.A., (2011), A Universal Reference-Free Blurriness Measure. *Image Quality and System Performance VIII*, 7867(78670B).
https://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.872477 152

Colombo, E.M., Kirschbaum, C.F. and Raitelli, M., (1987), Legibility of Texts: The Influence of Blur. *Lighting Research and Technology*, 19(3).
http://lrt.sagepub.com/cgi/doi/10.1177/096032718701900302 96

Colomina, I. and Molina, P., (2014), Unmanned Aerial Systems for Photogrammetry and Remote Sensing: A Review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 92. 32

Cooper, M.A.R. and Cross, P.A., (1988), Statistical Concepts and their Application in Photogrammetry and Surveying. *The Photogrammetric Record*, 12(71). 61

Crete, F., Dolmiere, T., Ladret, P. and Nicolas, M., (2007), The Blur Effect: Perception and Estimation with a New No-Reference Perceptual Blur Metric. *Human Vision and Electronic Imaging XII*, 6492(64920I).
http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1298489 VI, 100, 103, 104, 105, 106, 107, 132, 133

Deutsch, J., (2015), Mechanics Ebook. Internet, accessed: 21/04/2015.
http://deutsch.physics.ucsc.edu/6A/book/ 130

Dold, J., (1996), Influence of Large Targets on the Result of Photogrammetric Bundle Adjustment. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXI(B5).
http://www.isprs.org/proceedings/XXXI/congress/part5/119_XXXI-part5.pdf 56

Eisenbeiss, H., (2009), *UAV Photogrammetry*. Ph.D. thesis, ETH Zürich.
http://www.igp-data.ethz.ch/berichte/Blaue_Berichte_PDF/105.pdf 31

Eisenbeiss, H., (2011), Tutorial UAV-g Introduction. In: *UAV-g*.
http://www.geometh.ethz.ch/uav_g/tutorial/eisenbeiss_web 31

Emling, R., (2008), *Digitalfotografie - Das Grosse Einsteigerbuch*. 3rd edition, Humboldt.
https://books.google.com/books?id=NalOsEBP65UC&printsec=frontcover&dq=Digitalfotografie&hl=de&sa=X&ei=FzFvVYSxBYOL7AbjmoKQDw&ved=0CB0Q6AEwAA#v=onepage&q=Digitalfotografie&f=false 45, 46, 96

Eos Systems Inc., (2014), PhotoModeler - close-range photogrammetry and image-based modeling. Internet, accessed: 04/05/2015.
http://www.photomodeler.com/index.html 7

Ernstoff, M., Hoffman, W. and Winner, R., (1977), Liquid Crystal Dot Color Display. (4006968).
http://patft.uspto.gov/netacgi/nph-Parser?Sect2=PTO1&Sect2=HITOFF&p=1&u=/netahtml/PTO/search-bool.html&r=1&f=G&l=50&d=PALL&RefSrch=yes&Query=PN/4006968 10

Fergus, R., Singh, B., Hertzmann, A. and Roweis, S. T. Freeman, W.T., (2006), Removing Camera Shake from a Single Photograph. *ACM Transactions on Graphics*, 25(3).
http://dl.acm.org/citation.cfm?id=1141956 139, 163

Ferzli, R. and Karam, L.J., (2009), A No-Reference Objective Image Sharpness Metric Based on the Notion of Just Noticeable Blur (JNB). *Transactions on Image Processing*, 18(4).
http://www.ncbi.nlm.nih.gov/pubmed/19278916 101

Ford, A. and Roberts, A., (1996), Colour Space Conversions. Internet, accessed: 21/11/2014.
http://www5.informatik.tu-muenchen.de/lehre/vorlesungen/graphik/info/csc/COL_.htm 14

Galantucci, L.M., Percoco, G. and Dal Maso, U., (2008), Coded Targets and Hybrid Grids for Photogrammetric 3D Digitisation of Human Faces. *Virtual and Physical Prototyping*, 3(3).
http://www.tandfonline.com/doi/pdf/10.1080/17452750802259603 56

Green, G., (1828), *An Essay on the Application of Mathematical Analysis to the Theories of Electricity and Magnetism*. Hamilton, Adams and Co.
http://books.google.co.uk/books?id=GwYXAAAAYAAJ&hl=de&pg=PR3#v=onepage&q&f=false 58

Grenzdörffer, G., Niemeyer, F. and Schmidt, F., (2012), Development of Four Vision Camera System for a Micro-UAV. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX(B1).
http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B1/369/2012/isprsarchives-XXXIX-B1-369-2012.pdf 52, 138

Gülch, E., (2012), Photogrammetric Measurements in Fixed Wing UAV Imagery. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX(B1).
http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B1/381/2012/isprsarchives-XXXIX-B1-381-2012.pdf 97

Hattori, S., Akimoto, K., Fraser, C., Onu, T. and Imoto, H., (2000), Design of Coded Targets and Automated Measurement Procedures in Industrial Vision Metrology. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIII(B5).
http://www.isprs.org/proceedings/xxxiii/congress/part5/72_XXXIII-part5s.pdf 57

van den Heuvel, F.A., Kroon, R.J.G.A. and S., L.P.R., (1992), Digital Close-Range Photogrammetry Using Artificial Targets. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXIX(B5).
http://www.isprs.org/proceedings/XXIX/congress/part5/222_XXIX-part5.pdf 57

Hofmann, O., (2005), Calibration and Georeferencing of Aerial Digital Cameras. In: *Photogrammetric Week '05*, Wichman Verlag, Heidelberg.
http://www.ifp.uni-stuttgart.de/publications/phowo05/140hofmann.pdf 59

Hunt, R.W.G., (2004), *The Repoduction of Colour*. 6th edition, John Wiley & Sons.
http://books.google.co.uk/books?id=Cd_FVeuO10gC&lpg=PA41&dq=The Reproduction of Colour&hl=de&pg=PR4#v=onepage&q=The Reproduction of Colour&f=false 14

Inkscape, (2015), Inkscape. Internet, accessed: 04/05/2015.
https://inkscape.org/en/ 7

Jayant, N., Johnston, J. and Safranek, R., (1993), Signal Compression Based on Models of Human Perception. *Proceedings of the IEEE*, 81(10).
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=241504 101

Jefferies, C., (2014), Introducing Pars: The Search and Rescue Drone Helicopter. Internet, accessed: 06/11/2014.
http://www.mby.com/news/introducing-pars-the-search-and-rescue-drone-helicopter/ 1

Jerri, A., (1977), The Shannon sampling theorem - Its various extensions and applications: A tutorial review. *IEEE*, 65(11). 37

Jia, J., (2007), Single Image Motion Deblurring Using Transparency. In: *Conference on Computer Vision and Pattern Recognition*, IEEE.
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4270054 137, 139, 163

Johnson, C. and Casson, E., (1995), Effects of Luminance, Contrast, and Blur on Visual Acuity. *Optometry and Vision Science*, 72(12).
http://journals.lww.com/optvissci/Abstract/1995/12000/Effects_of_Luminance,_Contrast,_and_Blur_on_Visual.4.aspx 96

Joshi, N., Kang, S.B., Zitnick, C.L. and Szeliski, R., (2010), Image Deblurring Using Inertial Measurement Sensors. *ACM Transactions on Graphics*, 29(4).
http://portal.acm.org/citation.cfm?doid=1778765.1778767 37, 39, 40

Joshi, N., Szeliski, R. and Kriegman, D.J., (2008), PSF Estimation Using Sharp Edge Prediction. In: *Conference on Computer Vision and Pattern Recognition*, IEEE.
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4587834 100, 140, 163

Jung, S.W., Kim, T.H. and Ko, S.J., (2009), A Novel Multiple Image Deblurring Technique Using Fuzzy Projection Onto Convex Sets. *IEEE Signal Processing Letters*, 16(3).
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4776576 141, 144, 159, 163

Keshava, N. and Mustard, J.F., (2002), Spectral Unmixing. *IEEE Signal Processing Magazine*, 19(1).
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=974727 16

Killalea, D., (2014), Built to Burn: Inside a Raging Inferno and the Technology Fire Crews are Using to Beat Nature. Internet, accessed: 06/11/2014.
http://www.news.com.au/technology/environment/built-to-burn-inside-a-raging-inferno-and-the-technology-fire-crews-are-using-to-beat-nature/story-e6frflp0-1227113543737 1

Kraus, K., (2004), *Photogrammetrie*. 7th edition, Walter de Gruyter.
http://books.google.de/books?id=L5QcP5QkUm4C&lpg=PA273&hl=de&pg=PA273#v=onepage&q&f=false 28, 29

Kraus, K., (2007), *Photogrammetry - Geometry from Images and Laser Scans*. 2nd edition, Walter de Gruyter.
http://books.google.de/books?id=sjP-IvYCYkQC&lpg=PP1&dq=Photogrammetry%20-%20Geometry%20from%20Images%20and%20Laser%20Scans&hl=de&pg=PP1#v=onepage&q&f=false 2, 27, 29, 59

Krishnan, D. and Fergus, R., (2009), Fast Image Deconvolution Using Hyper-Laplacian Priors. *Advances in Neural Information Processing Systems (NIPS)*.
http://machinelearning.wustl.edu/mlpapers/paper_files/NIPS2009_0341.pdf 140, 163

Kundur, D. and Hatzinakos, D., (1996), Blind Image Deconvolution. *IEEE Signal Processing Magazine*, 13(3).
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=489268 137

Kurz, T.H., Buckley, S.J. and Howell, J.A., (2011), Integration of Panoramic Hyperspectral Imagin with Terrestrial Lidar Data. *The Photogrammetric Record*, 26(134).
http://onlinelibrary.wiley.com/doi/10.1111/j.1477-9730.2011.00632.x/pdf 15

Laben, C. and Brower, B., (2000), Process for Enahncing the Spatial Resolution of Multispectral Imagery Using Pan-Sharpening. (6011875).
https://docs.google.com/viewer?url=patentimages.storage.googleapis.com/pdfs/US6011875.pdf 151

Lelégard, L., Delaygue, E., Brédif, M. and Vallet, B., (2012), Detecting and Correcting Motion Blur From Images Shot with Channel-Dependent Exposure Time. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, I(3).
http://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/I-3/341/2012/isprsannals-I-3-341-2012.pdf 36, 37, 97, 140, 149, 163

Leonard, T. and Hsu, J.S.J., (1999), *Bayesian Methods: An Analysis for Statisticians and Interdisciplinary Researchers*. 1st edition, Cambridge University Press.
https://books.google.com/books?id=KBOZsRZP6H4C&dq=Bayes+theorem&hl=de&source=gbs_navlinks_s 139

Lillesand, T.M., Kiefer, R.W. and Chipman, J.W., (2008), *Remote Sensing and Image Interpretation*. 6th edition, John Wiley & Sons. 10, 15, 16

Liu, R., Li, Z. and Jia, J., (2008), Image Partial Blur Detection and Classification. In: *Conference on Computer Vision and Pattern Recognitio*, IEEE.
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4587465 102

Loughborough University, (2011), Corporate Identity Guidlines. Internet, accessed: 17/12/2014.
http://www.lboro.ac.uk/designandprint/downloads/corporate-identity/lboro-corporate-identity-guidelines.pdf 38

Lowe, D.G., (2004), Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2).
http://link.springer.com/10.1023/B:VISI.0000029664.99615.94 26

Lucy, L.B., (1974), An Iterative Technique for the Rectification of Observed Distributions. *Astronomical Journal*, 79(6).
http://adsabs.harvard.edu/full/1974AJ.....79..745L 139, 162, 163

Luhmann, T., (2010a), Close Range Photogrammetry for Industrial Applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6).
http://dx.doi.org/10.1016/j.isprsjprs.2010.06.003 55

Luhmann, T., (2010b), *Erweiterte Verfahren zur Geometrischen Kamerakalibrierung in der Nahbereichsphotogrammetrie*. Number 645 in C, Deutsche Geodätische Kommission, Munich, Germany.
http://dgk.badw.de/fileadmin/docs/c-645.pdf 59

Luhmann, T., (2014), Eccentricity in Images of Circular and Spherical Targets and its Impact on Spatial Intersection. *The Photogrammetric Record*, 29(148).
http://onlinelibrary.wiley.com/doi/10.1111/phor.12084/pdf 54, 55, 57, 58, 70

Luhmann, T., Robson, S., Kyle, S. and Boehm, J., (2014), *Close Range Photogrammetry*. Walter De Gruyter. 28, 29, 55, 59

MacAdam, D.P., (1970), Digital Image Restoration by Constrained Deconvolution. *Journal of the Optical Society of America*, 60(12).
http://www.opticsinfobase.org/abstract.cfm?id=54221 137

Maison Nicéphore Niépce, (2013), Invention of Photography. Internet, accessed: 12/12/2014.
http://www.niepce.org/pagus/pagus-inv.html 32

McCarthy, D.M.J., Chandler, J.H. and Palmeri, A., (2013), Monitoring Dynamic Structural Tests Using Image Deblurring Techniques. *Key Engineering Materials*, 569-570.
http://www.scientific.net/KEM.569-570.932 98

McHugh, S., (2015), Cambridge in Colour. Internet, accessed: 09/04/2015.
http://www.cambridgeincolour.com/ 141, 142

Meier, H.K., (1976), Über die Geometrische Genauigkeit von Luftbildkammern. *Schriftreihe des Instituts für Photogrammetrie der Universität Stuttgart*, 2(2), accessed: 25/12/2014.
http://www.ifp.uni-stuttgart.de/publications/phowo75/meier.pdf 59

Merrill, J., (2014), Government Use of Surveillance Drones is 'Probably Illegal'. Internet, accessed: 06/11/2014.
http://www.independent.co.uk/news/uk/politics/government-use-of-surveillance-drones-is-probably-illegal-9789296.html 1

Michaeli, T. and Irani, M., (2014), Blind Deblurring Using Internal Patch Recurrence. In: *European Converence on Computer Vision*, Springer International Publishing.
http://link.springer.com/chapter/10.1007%2F978-3-319-10578-9_51#close 140, 163

Molina, P., Colomina, I., Vitoria, T., Silva, P.F., Skaloud, J., Kornus, W., Prades, R. and Aguilera, C., (2012), Searching Lost People With UAVs: The System And Results Of The Close-Search Project. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX(B1).
http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B1/441/2012/isprsarchives-XXXIX-B1-441-2012.pdf 32

Morimoto, C. and Chellappa, R., (1996), Fast Electronic Digital Image Stabilization. In: *Proceedings of 13th International Conference on Pattern Recognition*, IEEE.
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=546956 36

Murray, J. and van Ryper, W., (1996), *Encyclopedia of Graphics File Formats*. 2nd edition, O'Reilly Media, accessed: 18/11/2014.
http://www.fileformat.info/resource/book/1565921615/index.htm 10, 11

Nagy, J.G., Palmer, K. and Perrone, L., (2004), Iterative Methods for Image Deblurring: A Matlab Object-Oriented Approach. *Numerical Algorithms*, 36(1).
http://www.springerlink.com/openurl.asp?id=doi:10.1023/B:NUMA.00000277 62.08431.64 137, 138

Narvekar, N.D. and Karam, L.J., (2009), A No-Reference Perceptual Image Sharpness Metric Based on a Cumulative Probability of Blur Detection. In: *Quality of Multimedia Experience*, IEEE.
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5246972 100, 101

NASA, (2015), Landsat 8. Internet, accessed: 04/05/2015.
http://landsat.gsfc.nasa.gov/?p=5081 15

Nasse, H.H., (2008), *Wie liest man MTF- Kurven?* Carl Zeiss.
http://www.zeiss.com/content/dam/Photography/new/pdf/de/cln_archiv/cln30_de_web_special_mtf_01.pdf
33

Niemeier, W., (2007), *Ausgleichungsrechnung*. Walter de Gruyter.
https://books.google.de/books?id=q7SNDUn4NhcC&lpg=PA161&ots=BTYqPk407F&dq=niemeyer ausgleichung&hl=de&pg=PR7#v=onepage&q=niemeyer ausgleichung&f=false 21, 60, 61

Nik Software, (2013), Snapseed. Internet, accessed: 05/07/2013.
http://www.snapseed.com/home/learn/mobile/creative-adjustments/ 99

Ong, E., Lin, W., Lu, Z., Yang, X., Yao, S., Pan, F., Jiang, L. and Moschetti, F., (2003), A No-Reference Quality Metric for Measuring Image Blur. In: *International Symposium on Signal Processing and Its Applications*, volume 1, IEEE.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1224741 100, 101

OpenCV Dev Team, (2014), OpenCV Documentation. Internet, accessed: 09/12/2014.
http://docs.opencv.org/ 7, 19, 21, 22, 24, 25, 58, 59, 91, 99

Oxford University Press, (2014), Image: Definition of Image in Oxford Dictionary (British & World English). Internet, accessed: 07/11/2014.
http://www.oxforddictionaries.com/definition/english/image 10

Pacey, R. and Fricker, P., (2005), Forward Motion Compensation (FMC) - Is It the Same in the Digital Imaging World? *Photogrammetric Engineering and Remote Sensing*, 71(11).
http://www.asprs.org/a/publications/pers/2005journal/november/highlight1.pdf 35

Park, S., Park, M. and Kang, M., (2003), Super-Resolution Image Reconstruction: A Technical Overview. *Signal Processing Magazine*, 20(3).
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1203207 136

Pasumansky, A., (2014), Estimate Image Quality - Inconsistent Results. Internet Forum, accessed: 21/04/2015.
http://www.agisoft.com/forum/index.php?topic=1924.msg10246#msg10246 134

PCB Piezotronics, (2015), PCB Piezotronics, Inc.: Sensors to Measure Vibration, Acoustics, Force, Pressure, Load, Strain, Shock and Torque. Internet, accessed: 16/11/2015.
http://www.pcb.com/ 48

Pennebaker, W.B. and Mitchell, J.L., (2004), *JPEG Still Image Data Compression Standard*. 8th edition, Kluwer Academic Publishers.
http://books.google.co.uk/books?id=AepB_PZ_WMkC&printsec=copyright&hl=de#v=onepage&q&f=false
12, 13

Peterman, V. and Mesarič, M., (2012), Land Survey from Unmaned Aerial Vehicle. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX(B1).
http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B1/447/2012/isprsarchives-XXXIX-B1-447-2012.pdf 32

PIEneering, (2012), Aspects of Accuracy in UAS Photogrammetry. Technical report, Parallel Image Engineering.
32

QuickMTF, (2015), QuickMTF. Internet, accessed: 06/11/2014.
http://www.quickmtf.com/infopanel.html 34

Rahtu, E., Heikkilä, J., Ojansivu, V. and Ahonen, T., (2012), Local Phase Quantization for Blur-Insensitive Image Analysis. *Image and Vision Computing*, 30(8).
http://linkinghub.elsevier.com/retrieve/pii/S0262885612000510 102

Raskar, R., Agrawal, A. and Tumblin, J., (2006), Coded Exposure Photography: Motion Deblurring Using Fluttered Shutter. *ACM Transactions on Graphics*, 25(3).
http://dl.acm.org/citation.cfm?id=1141957 37, 140, 141, 163

Rav-Acha, A. and Peleg, S., (2000), Restoration of Multiple Images with Motion Blur in Different Directions. In: *Workshop on Applications of Computer Vision*, IEEE.
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=895398 137

Remondino, F., Barazzetti, L., Nex, F., Scaioni, M. and Sarazzi, D., (2012), UAV Photogrammetry for Mapping and 3D Modeling - Current Status and Future Perspectives. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-1(C22).
http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXVIII-1-C22/25/2011/ 32

Richardson, W.H., (1972), Bayesian-Based Iterative Method of Image Restoration. *Journal of the Optical Society of America*, 62(1).
http://www.opticsinfobase.org/abstract.cfm?URI=josa-62-1-55 139, 162, 163

Roelofs, G., (2014), Portable Network Graphics. Internet, accessed: 20/11/2014.
http://www.libpng.org 12

Sachs, D., Nasiri, S. and Goehl, D., (2006), Image Stabilization Technology Overview, accessed: 26/06/2015.
http://www.digikey.de/Web Export/Supplier Content/invensense-1428/pdf/invensense-image-stabilization-technology.pdf?redirected=1 35, 36, 96

Schneider, D., (2010), *Geometrische und Stochastische Modelle für die Integrierte Auswertung Terrestrische Laserscanndaten und Photogrammetrische Bilddaten*. Ph.D. thesis, Dresden University of Technology.
http://dgk.badw.de/fileadmin/docs/c-642.pdf 60

Schowengerdt, R.A., (2007), *Remote sensing - Models and Methods for Image Processing*. 3rd edition, Elsevier Inc. 37

Shah, C.A. and Schickler, W., (2012), Automated Blur Detection And Removal In Airborne Imaging Systems Using IMU Data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX(B1).
http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B1/321/2012/isprsarchives-XXXIX-B1-321-2012.pdf 37, 97

Shan, Q., Jia, J. and Agarwala, A., (2008), High-Quality Motion Deblurring from a Single Image. *ACM Transactions on Graphics*, 27(3).
http://portal.acm.org/citation.cfm?doid=1360612.1360672 140, 154, 160, 162

SharkD, (2010), User:SharkD. Internet, accessed: 25/11/2014.
http://commons.wikimedia.org/wiki/User:SharkD 14

Sheikh, H.R., Sabir, M.F. and Bovik, A.C., (2006), A Statistical Evaluation of Recent Full Reference Quality Assessment Algorithms. Internet, accessed: 26/06/2015.
http://live.ece.utexas.edu/research/quality/subjective.htm 39

Shortis, M.R. and Seager, J.W., (2014), A Practical Target Recognition System for Close Range Photogrammetry. *The Photogrammetric Record*, 29(147).
http://onlinelibrary.wiley.com/doi/10.1111/phor.12070/pdf 55

Sieberth, T., Wackrow, R. and Chandler, J.H., (2014), Motion Blur Disturbs - The Influence of Motion-Blurred Images in Photogrammetry. *The Photogrammetric Record*, 29(148).
http://onlinelibrary.wiley.com/doi/10.1111/phor.12082/abstract 106

Singh, D., (2010), *Fundamentals of Optics*. 1st edition, PHI Learning Private Limited.
http://books.google.de/books?id=L5QcP5QkUm4C&lpg=PA273&hl=de&pg=PA273#v=onepage&q&f=false 25

Skahill, P., (2014), If Police in Connecticut Use Drones for Surveillance, Do They Need a Warrant? Internet, accessed: 06/11/2014.
http://wnpr.org/post/if-police-connecticut-use-drones-surveillance-do-they-need-warrant 1

Steinhardt, J., (1936), Intensity Discrimination in the Human Eye. *The Journal of General Physiology*, 20(2).
http://jgp.rupress.org/content/20/2/185.full.pdf+html 15

Stiles, R.N., (1976), Frequency and Displacement Amplitude Relations for Normal Hand Tremor. *Journal of Applied Physiology*, 40(1).
http://www.ncbi.nlm.nih.gov/pubmed/1248981 96

Sun, L., Cho, S., Wang, J. and Hays, J., (2014), Good Image Priors for Non-blind Deconvolution : Generic vs Specific. In: *European Conference on Computer Vision*, Springer International Publishing.
http://link.springer.com/chapter/10.1007%2F978-3-319-10593-2_16 139

Sun Microsystems Incorporated, (1999), Programming in Java Advanced Imaging. Internet, accessed: 12/12/2014.
http://www2.hs-fulda.de/caelabor/inhalte/java/j3d/j3d_seminar/19/JAI Guide von Sun/JAITOC.fm.html 25

Tai, Y.W., Du, H., Brown, M.S. and Lin, S., (2008), Image/Video Deblurring Using a Hybrid Camera. In: *Conference on Computer Vision and Pattern Recognition*, IEEE.
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4587507 140, 163

Tai, Y.W., Du, H., Brown, M.S. and Lin, S., (2010), Correction of Spatially Varying Image and Video Motion Blur Using a Hybrid Camera. *Transactions on Pattern Analysis and Machine Intelligence*, 32(6).
http://www.ncbi.nlm.nih.gov/pubmed/20431128 138

The GIMP Help Team, (2015), GIMP Documentation. Internet, accessed: 2015/04/09.
http://docs.gimp.org 141, 142

The GIMP Team, (2015), GIMP - The GNU Image Manipulation Program. Internet, accessed: 04/05/2015.
http://www.gimp.org/ 8, 23

The MathWorks, Inc., (2015), MATLAB and Simulink for Technical Computing - MathWorks. Internet, accessed: 04/05/2015.
http://uk.mathworks.com/index.html?s_tid=gn_logo 7

Triggs, B., McLauchlan, P.F., Hartley, R.I. and Fitzgibbon, A.W., (2000), Bundle Adjustment - A Modern Synthesis. In: *International Conference on Computer Vision*, volume 1883.
http://link.springer.com/chapter/10.1007/3-540-44480-7_21 29, 60

Wang, Y., Fevig, R. and Schultz, R.R., (2008), Super-Resolution Mosaicking of UAV Surveillance Video. In: *International Conference on Image Processing*, IEEE.
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4711762 136

Whyte, O., Sivic, J., Zisserman, A. and Ponce, J., (2010), Non-Uniform Deblurring for Shaken Images. In: *Computer Vision and Pattern Recognition*, IEEE.
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5540175 139, 163

Wiener, N., (1950), *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. 2nd edition, Press of MIT and John Wiley & Sons Inc.
http://babel.hathitrust.org/cgi/pt?id=uc1.b4062686;view=1up;seq=12 139, 162, 163

Wijenayake, U., Choi, S.I. and Park, S.Y., (2014), Automatic Detection and Decoding of Photogrammetric Coded Targets. In: *Conference on Electronics, Information and Communications*, IEEE.
http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6914413&queryText% 3DAutomatic+Detection+and+Decoding+of+Photogrammetric+Coded+Targets 58

Wilson, N., (2014), UK and France Sign £120m Deal for Military Drones Study. Internet, accessed: 06/11/2014.
http://www.ibtimes.co.uk/uk-france-sign-120m-deal-military-drones-study-1473362 1

www.eye.de, (2014), www.eye.de. Internet, accessed: 25/11/2014.
http://eye.de/ 15

www.red.com, (2015), Global and Rolling Shutter. Internet, accessed: 09/12/2014.
http://www.red.com/learn/red-101/global-rolling-shutter 34

Xia, R.B., Zhao, J.B., Liu, W.J., Wu, J.H., Fu, S.P., Jiang, J. and Li, J.Z., (2012), A Robust Recognition Algorithm for Encoded Targets in Close-Range Photogrammetry. *Journal of Information Science and Engineering*, 28(2).
http://journal.iis.sinica.edu.tw/paper/1/100141-3.pdf?cd=FA6C7140A358B26E6 57, 58

Yuan, L., Sun, J., Quan, L. and Shum, H.Y., (2007), Image Deblurring with Blurred/Noisy Image Pairs. *ACM Transactions on Graphics*, 26(3).
http://research.microsoft.com/en-us/um/people/jiansun/papers/Deblurring_SIGGRAPH07.pdf 137, 139, 163

Zhou, G., Yang, J., Li, X. and Yang, X., (2012), Advances Of Flash LIDAR Development Onboard UAV. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX(B3).
http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B3/193/2012/isprsarchives-XXXIX-B3-193-2012.pdf 32

A Additional material

A.1 Blur disturbs

Table A.1: Number of detected targets (out of 3) in Nikon D80 dataset

0.59mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1700	2100	2700	3200
4	3	2	3	2
10	3	3	3	3
19	3	3	3	3
35	3	3	3	3
0.78mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1700	2100	2700	3200
4	3	3	3	3
10	0	2	3	2
19	3	3	3	3
35	3	3	2	3
0.95mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1700	2100	2700	3200
4	0	2	3	3
10	0	0	1	2
19	1	2	3	3
35	3	3	3	3
1.11mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1700	2100	2700	3200
4	0	3	3	2
10	0	0	0	1
19	0	0	3	3
35	3	3	3	3

1.27mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1700	2100	2700	3200
4	0	0	2	0
10	0	0	0	0
19	0	0	0	3
35	0	0	3	3
1.46mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1700	2100	2700	3200
4	0	0	1	0
10	0	0	0	0
19	0	0	0	0
35	0	0	1	3
1.64mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1700	2100	2700	3200
4	0	0	0	0
10	0	0	0	0
19	0	0	0	0
35	0	0	0	3
1.95mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1700	2100	2700	3200
4	0	0	0	0
10	0	0	0	0
19	0	0	0	0
35	0	0	0	0

Table A.2: Number of detected targets (out of 3) in Nikon D7000 dataset

0.42mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1800	2200	2800	3300
4	3	3	3	3
10	3	3	3	3
19	3	3	3	3
35	3	3	3	3
0.51mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1800	2200	2800	3300
4	0	2	3	3
10	3	3	3	3
19	3	3	3	3
35	3	3	3	3
0.72mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1800	2200	2800	3300
4	0	0	2	3
10	1	2	3	3
19	3	3	3	3
35	3	3	3	3
0.82mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1800	2200	2800	3300
4	0	0	3	2
10	0	0	3	2
19	3	3	3	3
35	3	3	3	3
1.00mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1800	2200	2800	3300
4	0	0	2	2
10	0	0	0	0
19	0	2	3	3
35	3	2	3	3

1.27mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1800	2200	2800	3300
4	0	0	0	0
10	0	0	0	0
19	0	0	0	0
35	1	0	3	1
1.34mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1800	2200	2800	3300
4	0	0	0	0
10	0	0	0	0
19	0	0	0	0
35	0	0	2	0
1.51mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1800	2200	2800	3300
4	0	0	0	0
10	0	0	0	0
19	0	0	0	0
35	0	0	1	0
1.59mm camera displaced				
Target size (mm)	Camera to object distance (mm)			
	1800	2200	2800	3300
4	0	0	0	0
10	0	0	0	0
19	0	0	0	0
35	0	0	0	0

B Publications

B.1 Double-blind peer reviewed

THE PHOTOGRAMMETRIC RECORD

The Photogrammetric Record 29(148): 434–453 (December 2014)
DOI: 10.1111/phor.12082

MOTION BLUR DISTURBS – THE INFLUENCE OF MOTION-BLURRED IMAGES IN PHOTOGRAMMETRY

T. Sieberth (T.Sieberth@lboro.ac.uk)
R. Wackrow (R.Wackrow@lboro.ac.uk)
J. H. Chandler (J.H.Chandler@lboro.ac.uk)
Loughborough University, Loughborough, UK

Abstract

Unmanned aerial vehicles (UAVs) have become an interesting and active research topic for photogrammetry. Current research is based on images acquired by UAVs which have a high ground resolution and good spectral resolution due to low flight altitudes combined with a high-resolution camera. One of the main problems preventing full automation of data processing of UAV imagery is the unknown degradation effect of blur caused by camera movement during image acquisition. The purpose of this paper is to analyse the influence of blur on photogrammetric image processing. Images with precisely known motion blur were produced to determine the effect. It was found that even small blurs affect normal photogrammetric processes significantly. Although operator intervention might be time consuming, it can ensure that the results are still of acceptable accuracy.

Keywords: automation, blur, bundle adjustment, camera calibration, image processing, photogrammetry, UAV

Introduction

A constraint enforced on the acquisition of early photographs used in photogrammetry was a stable camera position and a stationary object as the basic requirements for sharp images. Exposure times of many days were required to obtain an image (Maison Nicéphore Niépce, 2013). Today, professional photographers use a tripod and remote shutter release to prevent camera movement during image acquisition.

Unfortunately, unmanned aerial vehicles (UAVs) rarely provide a stable camera position. UAVs are affected by wind, turbulence, sudden input by the operator and also by the flight movement of the aircraft itself. However, their good manoeuvrability and flight path control, combined with endurance, flight range and low cost, make UAVs an appropriate platform for a range of different applications (Eisenbeiss, 2009). Nevertheless, limited payload, regulatory restrictions and vulnerability of the UAV platform encourage the use of low-cost sensors, which dictates the use of consumer-grade cameras (Eisenbeiss, 2011; Eisenbeiss and Sauerbier, 2011). Unfortunately, the problem remains that the high spatial resolution of an image is often degraded due to motion blur. Fig. 1 represents an image acquired using a fixed-wing UAV at an altitude of approximately 100 m above

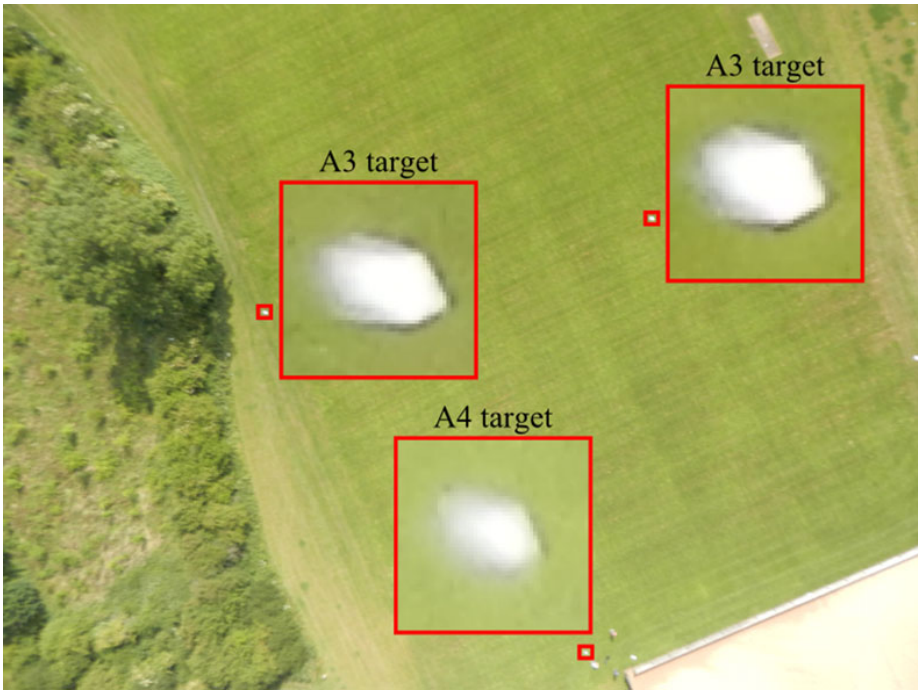


FIG. 1. A 16-megapixel image acquired by a UAV. The three insets show 20-times enlargement of the targets. Targets are blurred due to the forward motion of the UAV.

ground. A Nikon Coolpix S800c was used with the film speed set to ISO 125, which was chosen to achieve the best image quality with the least noise. The magnified insets in Fig. 1 represent blurred targets caused by forward motion of the UAV. However, rotational movements appear to have an even larger influence on the image sharpness than translational movements (Grenzdörffer et al., 2012).

Image sharpness degraded by motion blur is, amongst other effects, visible by a reduction in contrast, which can also be caused by noise or other processes influencing the radiometry of an image. Many of these influences are part of the hardware processing after the image is "on the camera sensor" and would influence a sharp image in the same way as a blurred image. Expressing the sharpness of a motion-blurred image, using a modulation transfer function (MTF) or similar sharpness curves, is problematic as they would return different results depending on the direction of measurement. To overcome this problem, motion blur is expressed in a scale-invariant manner as the physical movement of the camera body in millimetres.

This paper describes the effect of image blur on automatic image processing. Motion-blurred images have been generated to investigate their effect on automatic camera calibration and the functionality of coordinate calculations. The purpose is to define a threshold representing the maximum amount of image blur which is acceptable for automatic processing.

RELATED WORK

Photogrammetry is the science of reconstructing "the position, orientation, shape and size of objects from pictures" (Kraus, 2007, page 1). Aerial photogrammetry uses object

information in images acquired by airborne platforms ranging from aircraft to balloons. A major difference to close range photogrammetry is the normally unstable camera position due to a moving platform. The recently developed UAVs, which are increasingly popular for image flights, are especially vulnerable to wind and turbulence, and are sensitive to user navigational input.

For a successful image flight it is necessary to carefully prepare a flight plan beforehand. To provide appropriate image geometry for 3D measurements a recommended image fore-and-aft overlap (forward – along track) of 60% and a lateral overlap (sidelap – across track) of 20% should be used (Kraus, 2004; Luhmann et al., 2014). To calculate accurate 3D coordinates (X, Y, Z) for an object point, it is necessary to precisely measure the image coordinates (x, y) in at least two images. Today, fully automated data processing is demanded by an increasing number of users.

A series of methods have been developed in previous work to detect if an image is sharp or blurred. It is recognised that improving the image sharpness using deblurring algorithms is an important topic in computer vision and image processing. A widely used application for automatic blur detection is the "autofocus" system in cameras, which should prevent the user from taking optically blurred images caused by an inappropriate focal length setting (Kim and Paik, 1998). As optical blur can be prevented using these systems, other methods are required to suppress blur due to motion, which should be carefully distinguished as a discrete type of blur. Motion blur is often caused by human hand movement (jitter). This jitter has frequencies of 2 to 10 Hz with amplitudes of up to 1 mm (Stiles, 1976). Sachs et al. (2006) note that there is also a "drift of the hand" of up to 5 mm/s and that commercial systems for shake reduction use gyroscopes to prevent motion blur. Methods developed for aircraft use precise inertial measurement unit (IMU) and global navigation satellite system (GNSS) information to reduce angular and forward-motion blur (Pacey and Fricker, 2005). Such systems are expensive and cannot always be used since these would exceed the payload of typical UAV platforms. Furthermore, the IMU and GNSS sensors incorporated for flight stabilisation are not accurate enough and lack sufficient acquisition frequency to determine if an image is blurred or not. However, both sensors can provide additional information for blur detection algorithms such as the approximate path followed when the image was blurred.

Algorithms used for blur detection often use edge detection or frequency analysis. Edge detection methods focus on the spread and gradient of an edge (Ong et al., 2003; Joshi et al., 2008; Narvekar and Karam, 2009). In the case of a widely spread edge of low gradient it is first assumed that the image is blurred. This edge smear due to blur is also used in image frequency analysis methods. When an image has extensively smeared edges in the spatial domain it is characterised by the disappearance of high frequencies in the frequency domain (Liu et al., 2008; Rahtu et al., 2012). A problem with most blur detection methods is the presumption that the image is blurred. Another more important problem is that these blur detection methods are developed using mathematically blurred images without any relation to geometrical motion blur or random hand-held camera shake. In such circumstances the amount of blur is unknown and has to be evaluated subjectively. In an example presented by Shaked and Tastl (2005) (Fig. 2), image (a) is declared sharper than image (b). However, image (b) is at a different scale and contains more detail than (a), as it is possible to identify individual hair strains or differentiate between the shirt collar and the shirt itself. It is obvious that although subjective evaluation of blur is useful, it can be wrong. This explains the desire to develop a

(a) (b)

FIG. 2. A portrait taken with different cameras in a similar scene by Shaked and Tastl (2005).

measure of blur, by generating images with a precisely known image blur which is not influenced by human perception.

METHOD DEVELOPMENT

To quantify the impact of image blur on automatic image processing, images with a precisely known blur are required. In the study described in this paper, three datasets were produced using a shaker table (Fig. 3), which is a device normally used by construction engineers to test the strength of building materials and their resistance against earthquakes or other vibrating influences. In this study, a Nikon D80 camera was mounted on the shaker table and shaken with a known amplitude and frequency. Exposures were synchronised to this movement and images with precisely known blur could therefore be captured (Sieberth et al., 2013).

The first dataset consisted of convergent images of a calibration field comprising 54 coded targets and a Siemens Star in the image centre (Fig. 4(a)). The Siemens Star provides a directly visual way to evaluate both the amount and direction of blur. Coded targets were used to allow for fully automatic camera calibration.

The second dataset consisted of images of a 3D model which include 130 signalised targets and 6 coded targets. The coded targets were located around the 3D model to establish a stable local coordinate system and to evaluate the accuracy of automatically calculated 3D coordinates of the target points (Fig. 4(b)).

The third dataset consisted of differently sized signalised targets obtained with different camera-to-target distances and acquired using both a Nikon D80 and a Nikon D7000

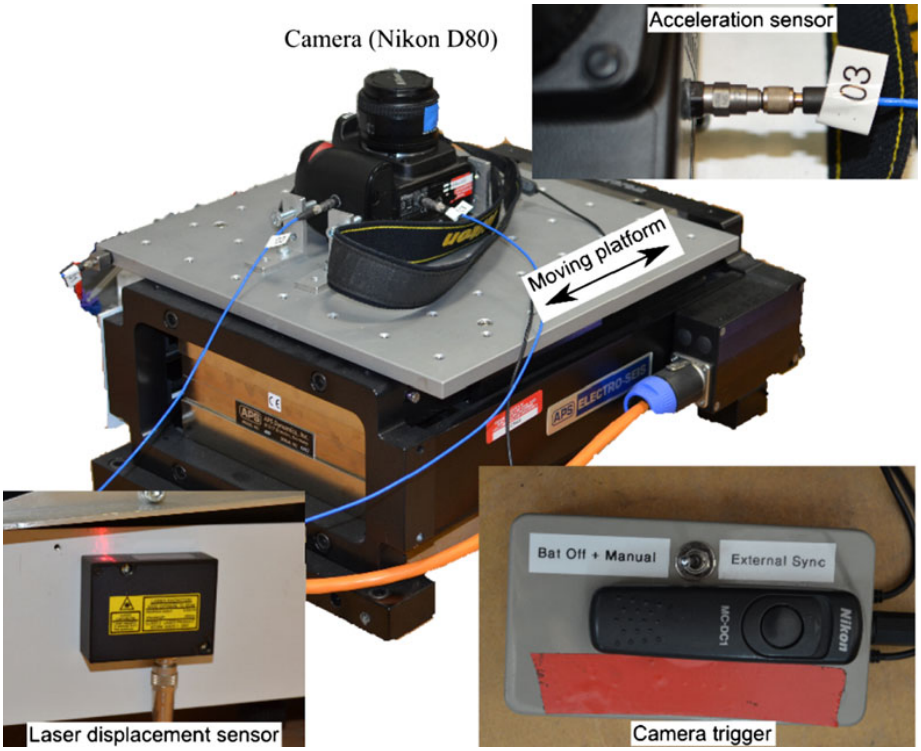
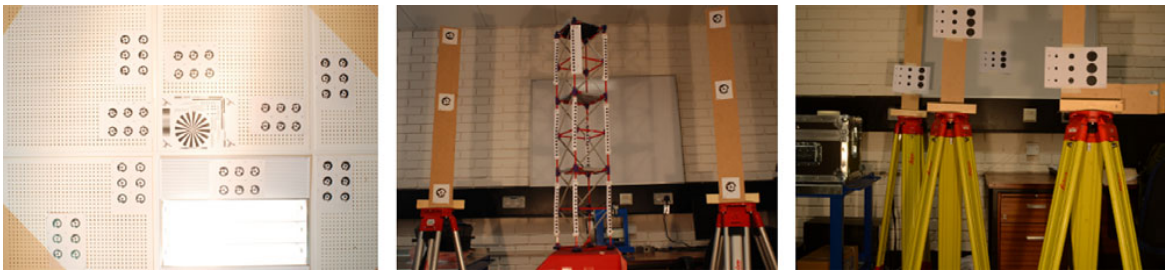


FIG. 3. Representation of the experimental set-up of the shaker table (Sieberth et al., 2013).



(a) (b) (c)

FIG. 4. Images used in the three datasets: (a) sharp image from the first dataset with the camera pointing towards a ceiling; (b) sharp image from the second dataset with the camera facing horizontally towards a wall; (c) sharp image from the third dataset with the camera also pointing horizontally towards a wall.

camera (Fig. 4(c)). This dataset was used to assess the accuracy of detection of target points.

Terminology

For analysing these datasets it is important to distinguish between the meaning of the terms "measurement", "detection", "identification" and "referencing". "Detection" is the process of finding a target in an image. As targets used in photogrammetry are often of a circular shape, a detection algorithm looks for round objects in the image. It finds the boundaries of a circle and then uses techniques to calculate the centre of the circle (Luhmann, 2014). The detection process can be carried out in two ways: fully automatic,

where the algorithm processes the complete image trying to find targets; or semi-automatic, where the operator defines a region of interest and adjusts search parameters. The latter case was the approach mostly adopted to detect targets in this study. The process of detection is important as the results of the recognised target are subsequently used to calculate the centre of the detected area and derive a "measurement" which is hopefully of sub-pixel precision and represents the actual measured location of the target in the plane of the image. However, if the target is not detected, or incorrectly detected, then no automatic measurement is possible or the measured coordinates will be incorrect. If automatic "measurements" are unsuccessful it is still possible to manually measure coordinates in the image. However, manual measurements are rarely practicable and it is more important to successfully detect a target to ensure automation and sub-pixel measurements.

With an automatically detected and measured target it might be also possible to carry out identification. The term "identification" will be used in this paper to refer to the process of assigning an identifying integer (ID) to a target. This is normally achieved automatically using a coded bit pattern surrounding the target and the code needs to be clearly readable to prevent incorrect identification (Shortis and Seager, 2014). An identified target can then be referenced. "Referencing" is the process of connecting identical targets across multiple images. This can be achieved by using the ID of identified targets.

Influence of Blur on Camera Calibration

The first dataset was used for automatic camera calibration calculation (Table I). To generate a complete image set for camera calibration, images from four stable positions around the shaker table were acquired, each with four different orientations (Fig. 5). These 16 calibration images were acquired with the camera mounted on a tripod and triggered remotely to avoid the introduction of any blur from this source. Around 2000 images with varying blur were taken using the shaker table. Of these 2000 images, 13 images with a camera displacement ranging from no movement up to 1·03 mm were chosen for further analysis.

The process of camera calibration was conducted using the software "PhotoModeler Scanner" (Version 2013·0·3·1131 (64 bit) dated 17th July 2013), which required a minimum of six images for processing. These six images were composed of a subset of five images from the stable camera positions and a sixth image from the shaker table position. A total of 13 image sets with each of the chosen blurred images were processed. Either fully automatic camera calibration was possible or manual intervention was required, depending on the amount of image blur introduced by the shaker table. Intervention involved operator input to semi-automatically detect and measure the blurred targets, which the algorithm was unable to detect, identify and measure.

TABLE I. A list of parameters for the camera calibration dataset.

	Dataset 1
Camera-to-object distance	1·80 m
Size of targets	8 mm
Number of coded targets	54
Focal length	25·35 mm
Aperture	f/8
Frames per process	6
Number of camera displacements analysed	13

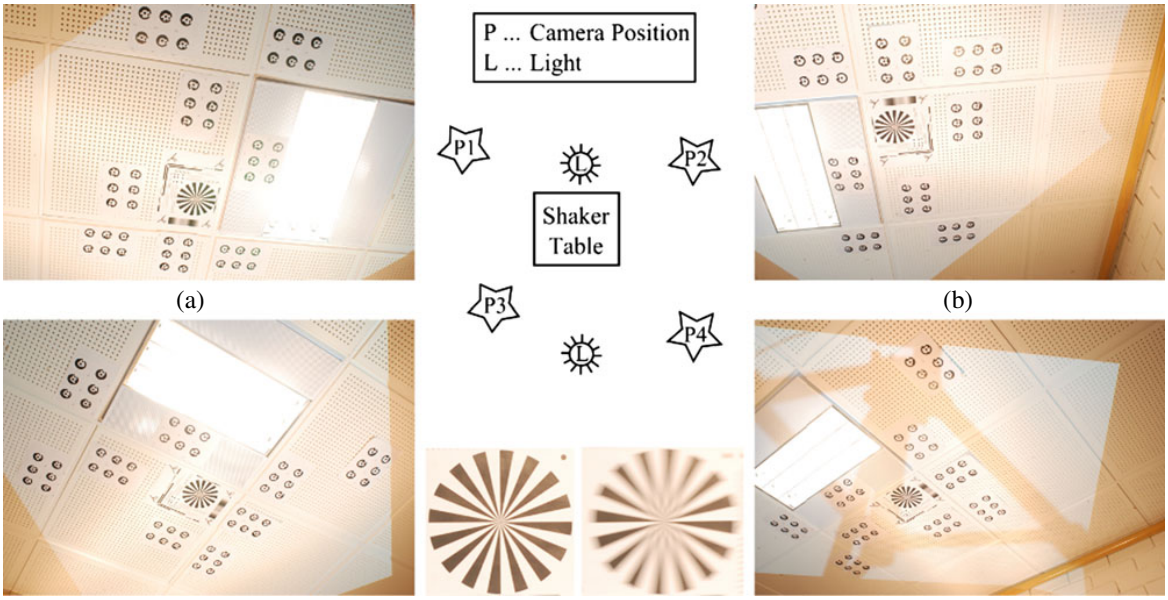


FIG. 5. Sketch of the shaker table set-up: (a) to (d) images from the four stable camera positions P1 to P4; (e) Siemens Star in a sharp image; (f) Siemens Star of the most blurred image with 1·03 mm camera displacement (168·6 sensor pixels).

Automated camera calibration becomes increasingly problematic as PhotoModeler is unable to automatically detect targets in the blurred image. Initial problems for fully automatic target detection occurred with a camera displacement larger than 0·263 mm (Fig. 6(a)), where the algorithm detected only 46 out of 54 targets. With a displacement of more than 0·377 mm (Fig. 6(b)) the number of detected targets was reduced to a point where calibration was no longer possible (Fig. 6(c)).

Additionally, referencing targets identified in the blurred images to targets detected in the sharp images was incorrect due to wrongly identified target numbers. Fig. 7(a) shows

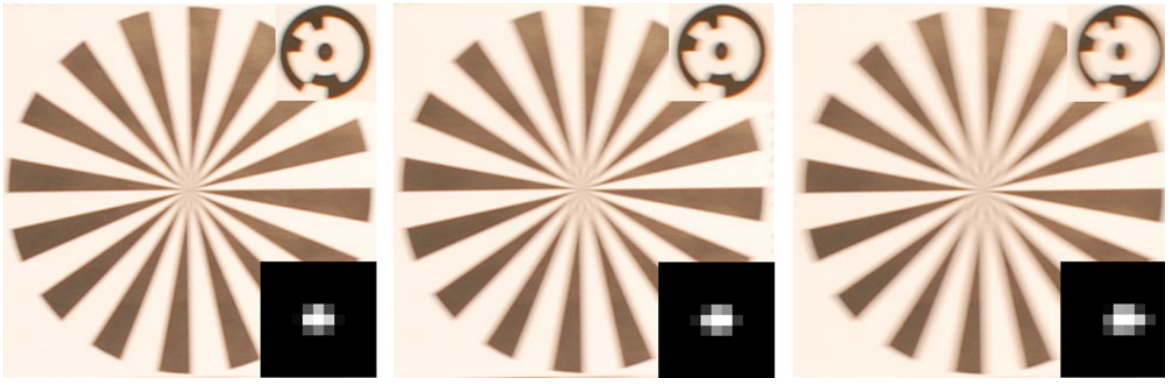


FIG. 6. Siemens Stars and coded targets affected by different motion: (a) camera displaced by 0·263 mm; (b) camera displaced by 0·377 mm (the last fully automatically processed image); (c) with the camera displaced 0·529 mm it is not possible to carry out fully automatic camera calibration. The insets (bottom right in each case) show the blur kernel calculated for these images.

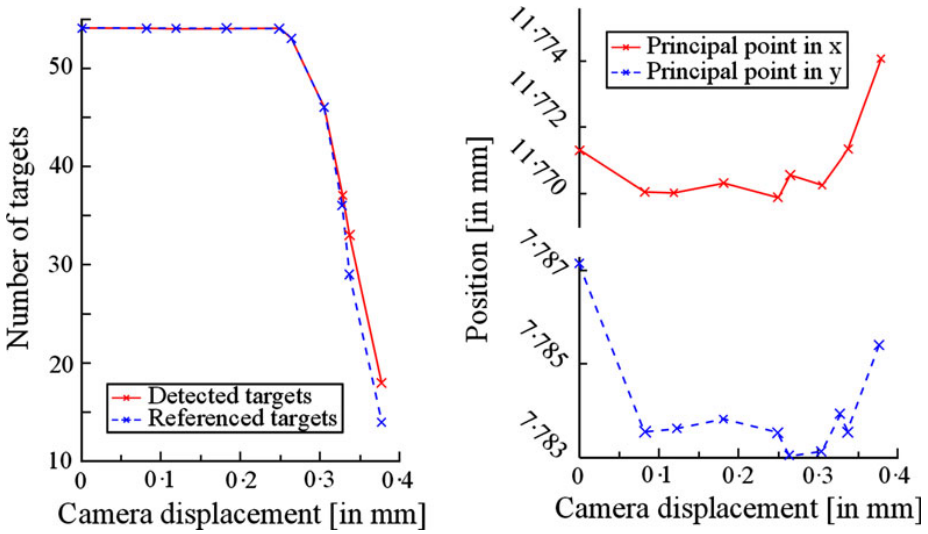


FIG. 7. (a) Number of detected and referenced targets and its dependency upon camera displacement. (b) Influence of camera displacement upon the calculated principal point position.

the number of detected targets and how many of them were referenced. Fig. 7(b) shows the estimated position of the principal point; it appears that it changes its position with increasing blur, although this may be due to the decreasing number of detected targets. To ensure that the camera calibration results were not just influenced by the decreased number of detected targets, semi-automatic detection and measurement of targets was performed using the "sub-pixel target mode" provided by PhotoModeler (Fig. 8). Unfortunately, identification fails completely because it is no longer possible to read the code. However, the "sub-pixel target mode" tool can be applied precisely on the centre dot of a coded target (Fig. 8(d)) for accurate detection and measurement (Figs. 8(e) and (f)).

If the search box of the targeting mode was not applied precisely (Fig. 8(a)), target detection and subsequent target measurement were unsuccessful (Figs. 8(b) and (c)). Manual creation of the detection box at the correct position with the right size is therefore prone to error and is also time consuming. Thus, referencing these targets needs to be carried out manually too. After this manual intervention, all targets have been detected and it was possible to assess the impact of blur created by camera displacement alone. With all targets correctly detected, calibrations can be conducted under conditions similar to a camera calibration using sharp images. Corrections calculated during the process can be distributed to all targets and variations in results can be attributed to blur.

Influence of Blur on Coordinate Calculation

A second dataset (Table II) was generated to assess the influence of blurred targets on coordinate calculations. A Nikon D80 camera was calibrated and used to acquire data for this specific study. A sharp image from a stable camera position beside the shaker table was acquired. Afterwards the camera was fixed on the shaker table and 328 images with varying blur were acquired. A subset of six images, with camera displacements from no movement up to 1·51 mm were used. The 3D coordinates of control points, the sharp image from the position next to the table and one of the six chosen blurred images from the shaker table position were included to evaluate the influence of blur on coordinate calculations.

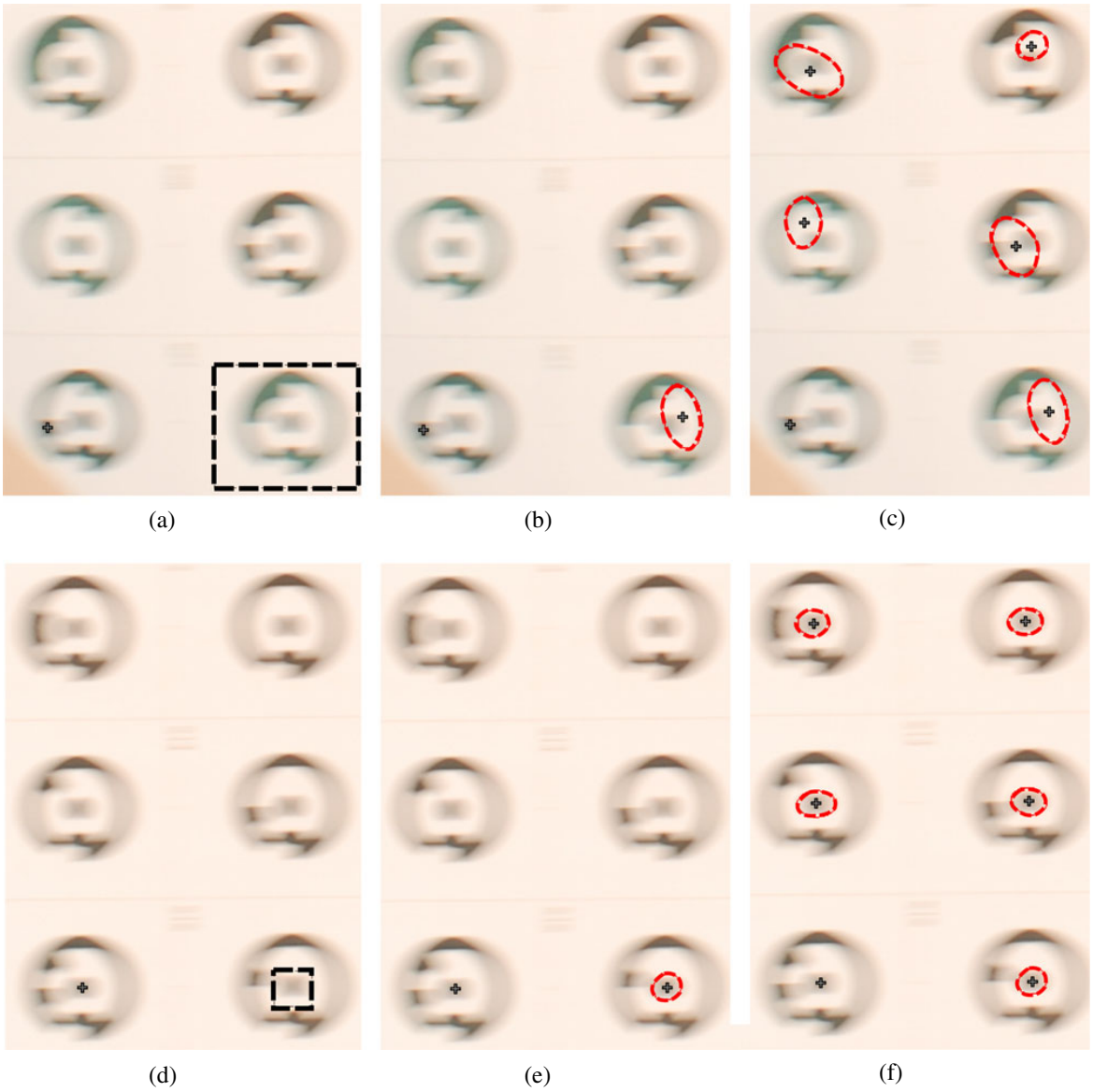


FIG. 8. Example of target detection on the most blurred image: (a) and (d) sub-pixel targeting area which was used; (b) and (e) the result of the detection and measurement; (c) and (f) the result of similar tests with search areas equivalent to those shown in (a) and (d). More successful detection and measurement can be achieved by using smaller target areas (d).

At first, the coded control points had to be detected, identified and measured. In addition to the detection and identification of the coded targets, it is now necessary to detect the uncoded targets on the model (Fig. 4(b)). Similarly to the results found with the first dataset, automatic detection of targets becomes unreliable with increasing blur. Eventually, detection and referencing between sharp and blurred images becomes impossible. When problems occurred using the "automatic target marking" tool, semi-automatic detection of each point was performed using PhotoModeler's "sub-pixel target mode" tool. Although appearing successful, Fig. 9(a) shows that the detection of circles can be inaccurate due to the blur. However, automatic measurement generally derives a point close to the centre of the blurred target. Difficulties encountered in automatic target detection and measurement shown in Fig. 9(b) seem to be caused by the appearance of two target silhouettes caused by

TABLE II. A list of parameters for the coordinate calculation dataset.

	Dataset 2
Camera-to-object distance	1·70 m
Size of targets	9 mm
Number of coded targets	6
Number of uncoded targets	130
Focal length	28·86 mm
Aperture	f/14
Frames per process	1 blurred + 1 sharp image
Number of camera displacements analysed	6 (+1 blur repeatedly used for manual target measurements)

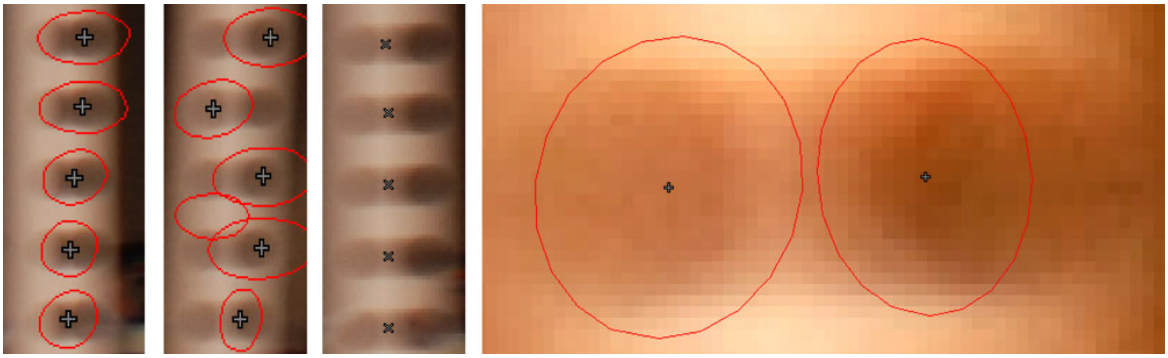


FIG. 9. Example of PhotoModeler's "sub-pixel target mode" tool on the most blurred images of the second dataset: (a) detection with 0·99 mm camera movement appears to be successful; (b) detection with 1·51 mm camera movement is strongly affected by blur; (c) manual measurement appears to be more accurate but is also more time consuming; (d) appearance of two dots per target compromises the accuracy of the automatic detection.

a large degree of blurring. An alternative explanation is the combination of a black target on a dark background. Fig. 9(d) represents an example where the right silhouette appears to be darker than the left because of the darker background at the right-hand side of the target. Due to the motion, the black background is mixed with the target and appears as a second, darker, target. This appearance of two targets can be solved by manual measurement (Fig. 9(c)) or by using methods which detect both circles and calculate the middle between both (Boracchi et al., 2007).

A critical aspect is target referencing between images. Referencing targets between images automatically becomes increasingly difficult with larger blurs (Table III). This is caused by the problem of finding similar features in both images to generate a unique identification because blurring makes identification of suitable features more difficult. Features are normally defined by edges but due to blurring they are not similar in both images. Finally, 3D coordinates can be determined and the effect of various degrees of blur can be established in the object space.

Influence of Blur on Target Detection

The third dataset was acquired to analyse the influence of blur on automatic target detection (Table IV). Targets of different sizes were placed at different distances, and

TABLE III. Influence of camera movement on automatic detection and referencing of uncoded signalled targets.

Camera movement (mm)	*Number of "automatic target marking" targets out of 128*	*Automatic referenced targets out of 126*
0	132 (4 wrong)	125
0·2012	130 (2 wrong)	96 (4 wrong)
0·3176	135 (8 wrong)	84 (3 wrong)
0·4923	126 (3 wrong)	69 (4 wrong)
0·9900	0	80 (1 wrong)
1·5134	0	11 (7 wrong)
1·5134 (manual measured targets)	0	41 (3 wrong)

TABLE IV. A list of parameters for the dataset to analyse the influence of blur on target detection.

	Dataset 3
Camera-to-object distances	1·70 m; 2·10 m; 2·70 m; 3·20 m
Size of targets	4 mm; 10 mm; 19 mm; 35 mm
Number of uncoded targets	48
Focal lengths	25·97 mm (Nikon D80); 29·31 mm (Nikon D7000)
Aperture	f/14
Number of camera displacements analysed	14 (Nikon D80); 15 (Nikon D7000)

images with different camera models and lenses were acquired. Two cameras were mounted on the shaker table: a Nikon D80 with a 24 mm lens and a Nikon D7000 with a 28 mm lens. The higher camera resolution and focal length of the D7000 provided images with a higher geometric resolution than the D80 images. Targets with diameters between 4 and 35 mm were placed at distances between 1·70 and 3·20 m from the camera (Fig. 10).

At first, the pixel width of each target at each distance was manually counted in the sharp image (Fig. 10(a)). Then automatic detection of signalised points was applied which proved to be successful for all targets except for the smallest targets at the greatest distance (4 mm targets at 3·20 m). At this distance the targets appear to be too small to be detected.

Then automated target detection was applied on the blurred images (Fig. 10(b)), to evaluate whether each target was detected. If detection failed for all three targets of the same size at the same distance, the target pixel width due to motion blur was manually counted. The discrepancy between blurred and sharp target widths was related to the displacement necessary to cause automatic detection to fail.

EVALUATION OF EXPERIMENTAL RESULTS

The prime purpose of the experimental work and subsequent evaluation was to investigate whether blur disturbs normal photogrammetric procedures or not.

Camera Calibration Dataset

The camera calibration dataset (dataset 1) demonstrates that blur certainly has an influence on the level of automation achievable during camera calibration in two ways. First, misidentification of targets occurs, which can result in incorrect referencing of targets between the images. Second, misdetections and subsequently mismeasurement influences the coordinate calculation. Processing 54 photogrammetric targets which are blurred resulted in a

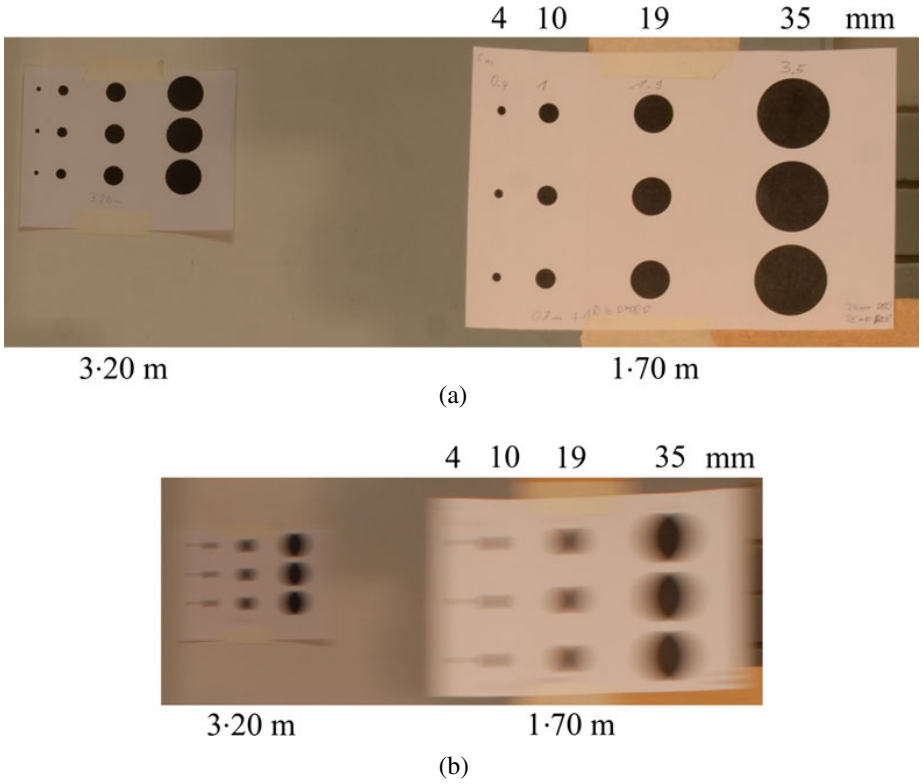


FIG. 10. Example of blurred targets (4 to 35 mm diameter) at different distances: (a) images acquired with a Nikon D7000 camera showing, on the right, a short distance (1·70 m) image and, on the left, far distance (3·20 m) targets in a sharp image; (b) images acquired by a Nikon D80 camera showing equivalent targets in a blurred image. The different sizes of the images are a result of different ground sampling distances.

small change in the camera calibration parameters compared with sharp images (Fig. 7(b)). The decreased accuracy of automatic target measurement can be illustrated by the change of the principal point position (Fig. 11(a)) and the variation in calculated image size (Fig. 11(b)). The figure shows that the calculated image width has a tendency to increase with larger blur. This can be explained perhaps by the shake direction, which is along the x axis.

A detection algorithm attempts to detect the edges of a target. It will find edges that are aligned in the direction of blur easily, because these edges are less affected by blur. In contrast, edges perpendicular to the blur direction create a transitional effect between the target and background, making the detection of edges difficult. Subsequent estimation of the middle, between the start and end of the target, becomes inaccurate and the measured centre does not represent the true target centre. Due to this incorrect detection, subsequent measurements are imprecise, especially in the direction of blur. Measurements derived perpendicularly to the blur direction remain uninfluenced. As these images are blurred along the x axis, x coordinates are the most likely to be inaccurate. It is, perhaps, surprising that the principal point does not change its position on the x axis but varies by about 8 μm on the y axis. There is no obvious explanation for this observation.

These variations in principal point position and estimated image size are insignificant and were not considered as a problem. However, these imply a deteriorating tendency with blur as may be expected. The failure of fully automatic detection and the requirement of semi-automatic target detection also implies that greater blur will cause increasing problems for automated photogrammetric processing.

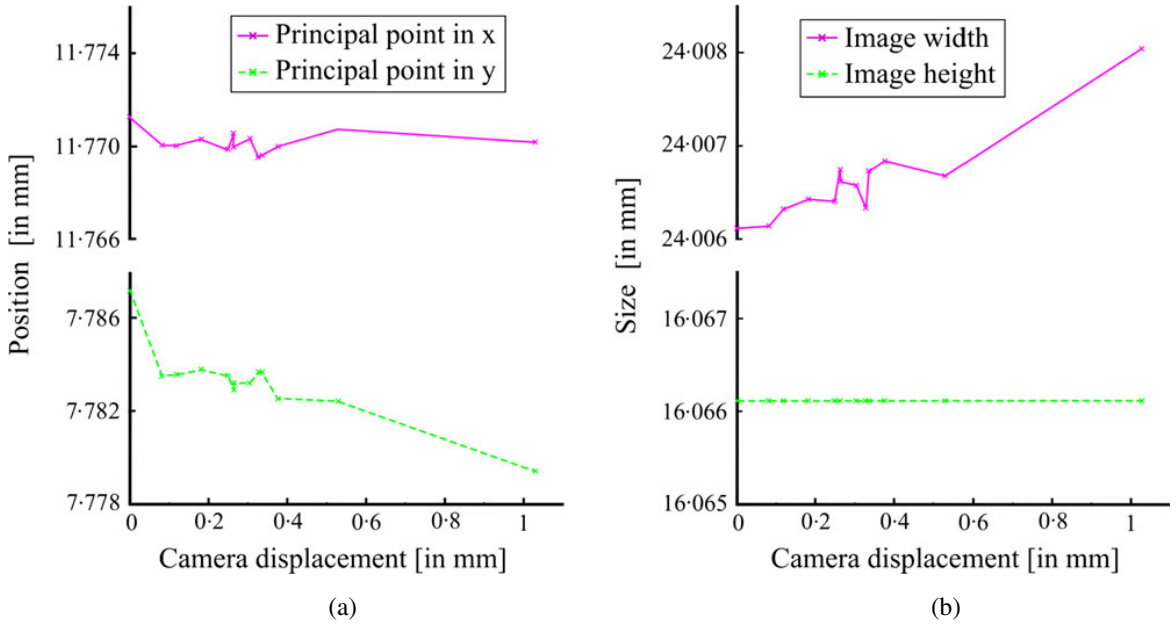


FIG. 11. Influence of image blur on camera calibration results. (a) The movement of the principal point. Contrast this with Fig. 7(b), where the principal point makes a "jump" at a camera displacement of 0·39 mm. (b) Change of the image size with camera displacement.

Coordinate Calculation Dataset

The second dataset used to calculate 3D coordinates also supports the findings of the camera calibration dataset. A small degree of blur prevents fully automatic detection and requires manually assisted or semi-automatic detection. A baseline of 0·82 m and distance of 1·70 m should provide an appropriate intersection angle for precise coordinate calculation. However, the calculated coordinates are inevitably influenced by increased motion blur. As expected, increasing blur creates larger discrepancies between the sharp and blurred coordinates (Fig. 12). Images with a small but apparently invisible blur of up to 0·5 mm are influenced less and create discrepancies in the object space of up to 0·4 mm (Fig. 12(a)). The actual camera displacement of 0·5 mm, divided by the pixel size on the sensor, equals 82 sensor pixels of blur. A 1·5 mm camera displacement is clearly visible to the human eye. This blur results in coordinate discrepancies of up to 20 mm between sharp and blurred images (Fig. 12(b)). In comparison, a fully manual target measurement results in a discrepancy of only 2 mm. This shows that, in the case of visible blur, automatic measurement is clearly inferior to manual measurement.

It can be assumed that larger camera-to-object distances, with both the same baseline and camera displacement, will cause larger errors due to an increasingly smaller intersection angle. A test using the higher resolution Nikon D7000 camera shows the same result of increasing discrepancies for increasingly blurred images.

Target Detection Dataset

While working with the first two datasets, it was observed that target detection in blurred images can be time consuming if semi-automatic detection and measurement is required. Tests reveal that relatively small camera displacements cause automatic target

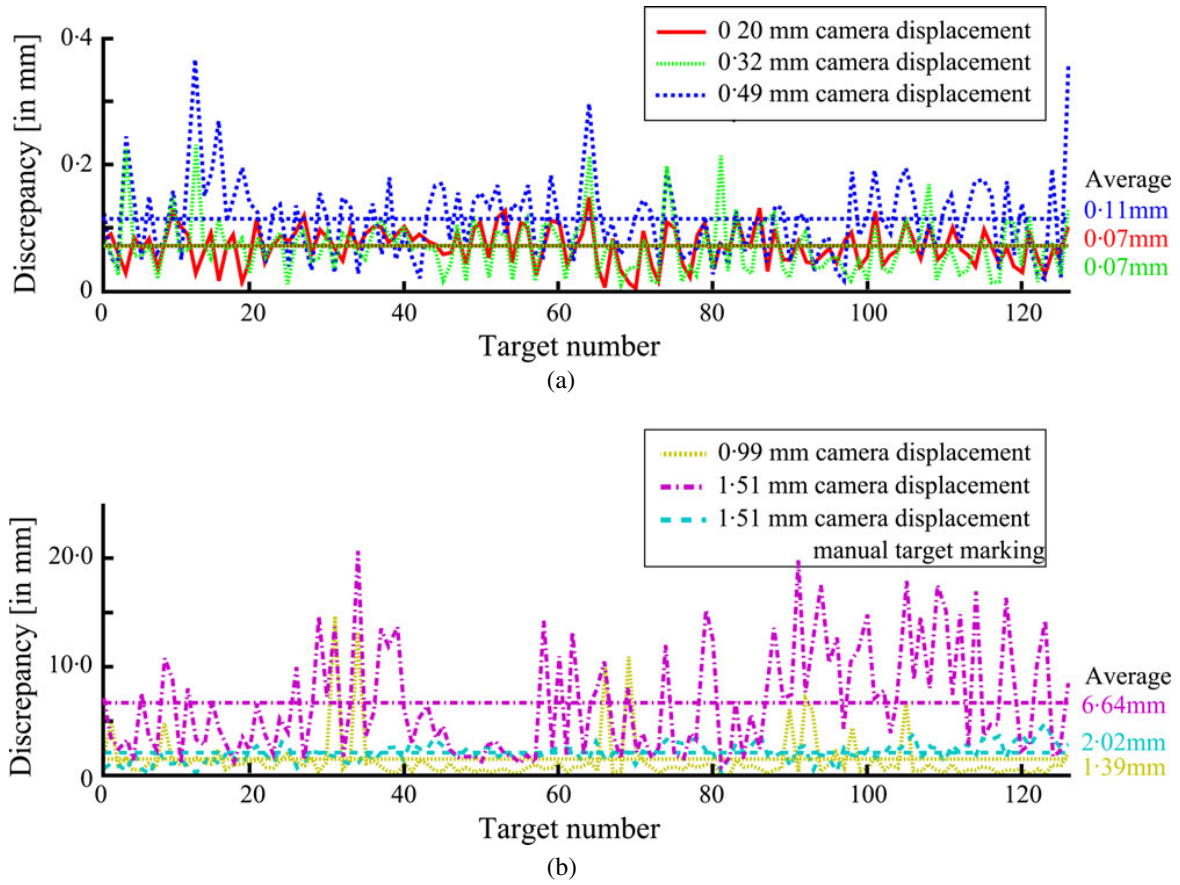


FIG. 12. Discrepancy between blurred and sharp image sets. The impact of large camera displacements (b) is 20 to 70 times larger than with small camera displacements (a).

detection to fail. The third dataset was used to determine a threshold beyond which blur disturbs the automatic detection of signalised targets (Fig. 13). The theoretical pixel size of a target is based on the focal length, the camera-to-object distance and the image sensor. In practice, the pixel width of a target can be counted on the image. In a sharp image, the theoretically calculated size and the actual size on the image should be equal. If the target is blurred, the pixel width in the image becomes larger than the theoretical size.

From the data captured it is possible to find the threshold at which detection fails due to blur. Determining the practical pixel width on the image and comparing it to the theoretical target width should make it possible to identify a linear dependency between sharp and blurred target sizes (Fig. 13(a)). The best-fitting linear dependency can be used to formulate an equation that describes the degree to which a target can be blurred before automatic detection is unsuccessful. A theoretical target size in pixels t_s can be determined easily using information about the target size in object space t_0, focal length c_k, distance between the camera and target h and sensor pixel size p_s. Furthermore, the blurred target size t_b can be related linearly to its theoretical equivalent t_s:

$$t_s = \frac{t_0 c_k}{h p_s}$$
$$t_b \leq 1{\cdot}166 t_s + 16{\cdot}794. \tag{1}$$

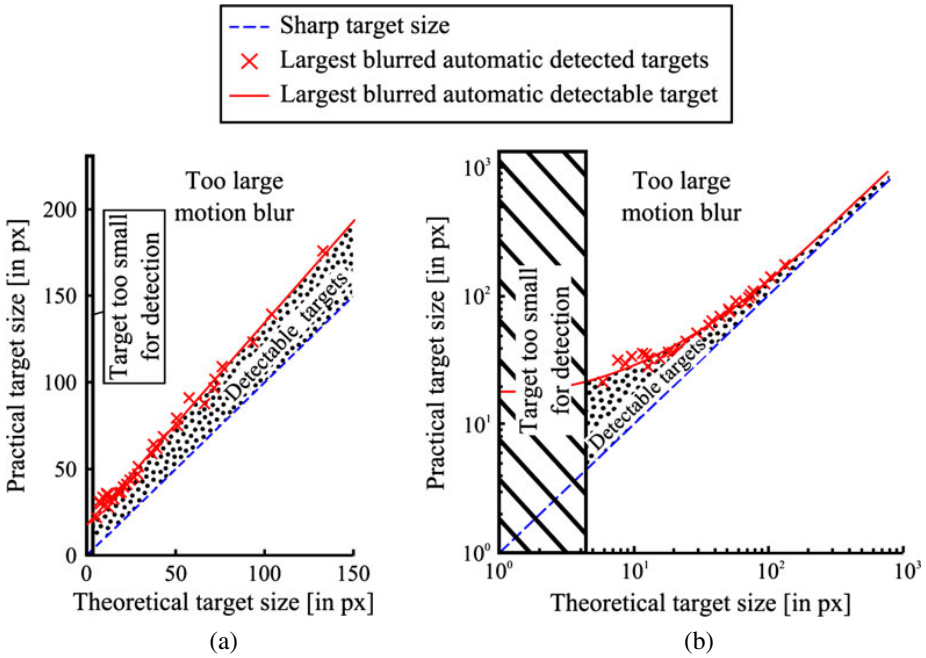


FIG. 13. Target size in a sharp image referenced to target size in a blurred image where automatic detection failed. Different camera models and lenses were used: (a) linear axes scale; (b) logarithmic axes scale.

From the tests conducted, targets which have a theoretical width of t_s, and appear in the image with a width between t_s and t_b, can be automatically detected. In Fig. 13(b) it is possible to see that the ratio between the theoretical and blurred target sizes decreases for larger targets. This implies that there is a greater tolerance to the blurring of small targets than that of large targets. A possible explanation is the threshold used in the detection algorithm. This threshold is based on the roundness of a target and how many pixels are part of a round target. For large targets an increased number of pixels do not support a round target and it is not accepted as a target.

The size of a blur is a direct result of camera displacement. Larger displacements cause targets to appear more smeared. Fig. 13 illustrates how much displacement is required for target detection to fail, and how wide the target appears in the image. Comparing camera displacement with the size of the blurred targets proves that large targets can be blurred more than small targets. Fig. 14 shows how target size, camera displacement and camera-to-object distance are related. The dependency between camera displacement and successful target detection is exponential, which shows that smaller targets tolerate more blur than larger targets. A target with a theoretical width of 50 pixels can be detected until it becomes so blurred that its width increases to 75 pixels, suggesting that a 50% blurring can be tolerated. However, a target measuring 100 pixels can be only smeared to 133 pixels before detection fails, which is only an increase of 33%. These examples suggest that the 50-pixel target can only suffer a displacement of 25 pixels before detection is unsuccessful, whilst the larger 100-pixel target can resist detection failure due to blur up to 33 pixels. This represents a difference of 8 pixels of additional camera displacement that can be tolerated if larger targets are used (Fig. 14). It also shows that an increasing object-to-camera distance results in a flatter exponential function. The outliers, which do not support the exponential function, are the smallest targets with a size of 4 mm. It would appear that the detection algorithm does not detect very small targets because they have too few pixels in the image

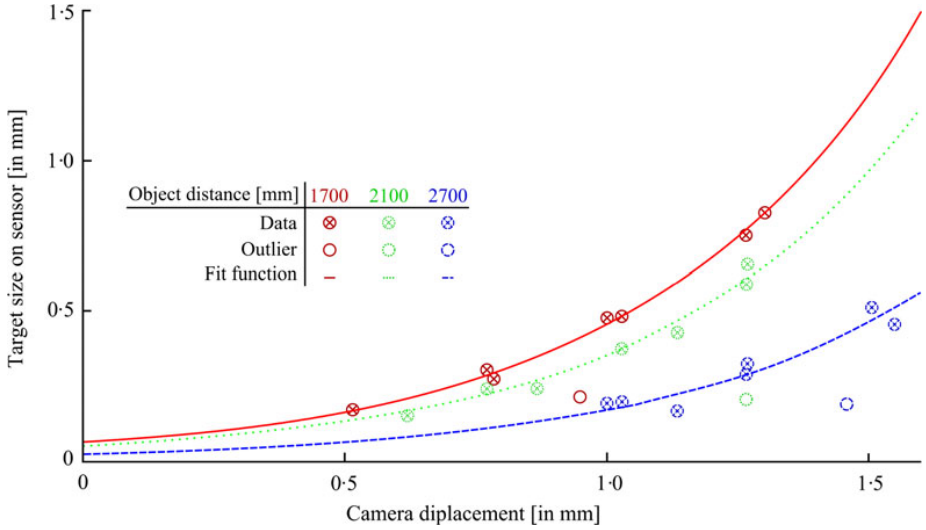


FIG. 14. Size of blurred targets related to displacement of the camera during shaking.

to be recognised as a target (Fig. 15(a)). However, once blurred they are represented by enough pixels to be accepted as a target (Fig. 15(b)). Notwithstanding this, as the target becomes more blurred, the detection fails again because the target shape becomes too elliptical and does not appear as a circle (Fig. 15(c)).

The function best representing the dependency between the camera-to-object distance, displacement and blurred target size, as shown in Fig. 14, is the exponential function shown in equation (2). This function makes it possible to calculate the blurred target size t_b as dependent on the camera-to-object distance h and the camera displacement d for short camera-to-object distances:

$$t_b = 70000h^{-1{\cdot}85}e^{2d}. \tag{2}$$

Equation (2) has not been investigated for $h \gg 3$ m and should be treated with caution for the increased camera-to-object distances that may be more common with UAVs. The displacement d is the result of the camera velocity during exposure time (equation (3)). Forward-motion displacement can be calculated using the exposure time and UAV velocity. In the case of rotations, the calculation depends on the position of the rotational axes. If the

(a) (b) (c)

FIG. 15. Influence of blur on small targets: (a) sharp target consists of too few pixels to be detected; (b) the same target when blurred, causing it to have an increased number of pixels; this target was detected by PhotoModeler; (c) the same target when even more blurred and no longer detectable by PhotoModeler.

origin of the axes is coincident with the camera's perspective centre, a calculation is possible using equation (3). As opposed to the roll ω and pitch φ, which only depend on the flight altitude, the yaw κ depends on the distance between the nadir position and the object s for which the displacement d is calculated:

$$\begin{aligned} d &= v_t \cdot ex \\ d &= h\tan(v_{\omega/\varphi}) \cdot ex \\ d_\kappa &= s\tan(v_\kappa) \cdot ex \end{aligned} \tag{3}$$

where d_κ is the displacement in yaw; $v_{t/\omega/\varphi/\kappa}$ are the respective velocities in translation, roll, pitch and yaw; and ex is the exposure time.

Connecting equations (1) and (2) makes it possible to calculate the minimum target size t_0 that should be used for the following parameters that are normally part of the flight planning: object distances h known a priori; focal length c_k; sensor pixel size p_s; and camera displacements d:

$$t_0 \geq \frac{70000h^{-1{\cdot}85}e^{2d}h - 16{\cdot}794hp_s}{1{\cdot}166c_k}. \tag{4}$$

However, the inverse calculation of determining displacement based on the size of a blurred target is not always valid. It would only be valid when the blurred target is on the threshold between successful and unsuccessful detection because this equation is based on the maximum size of blurred targets. If a less blurred target is used in this calculation the real camera displacement will be larger than the calculated displacement.

Discussion of Results and Implications for Photogrammetric Imagery

The study reported in this paper demonstrates that working with blurred images causes a range of challenges. First, it is difficult for the human eye to quantify the amount of blur in an image, especially if there is no image available for comparison. Second, automatic processing is clearly influenced by small motion blur and the number of detected signalised targets decreases. Even images of apparently high visual quality can cause problems if a small amount of blur is present. The number of measured targets directly influences the accuracy and ability of subsequent calculations including camera calibration and bundle adjustment. The detection of targets becomes more reliable with larger targets, which include more pixels to calculate accurately the target centre. Furthermore, the same amount of blur causes a single target to blur laterally, causing misdetection or multiple detection. However, very large targets are impractical for efficient photogrammetric data processing.

If automatic target detection is not possible, semi-automatic detection of targets over small areas can be successful. However, this is time consuming as these detection areas have to be defined manually and accurately. Even with well-defined detection areas, the detection fails with increasing blur. Manual measurement of targets is possible but the results are not as accurate as automatic measurements on sharp images.

A limitation of the detailed tests conducted in this study has been the focus on close-range images. Care needs to be exercised in assessing the implications of these findings for UAV imagery where flight altitudes are much larger than the distances used in these experiments and camera displacements are larger due to the flight velocity. Furthermore, the

ratios between the object distance and baseline are smaller than in the conducted experiments, which results in glancing ray intersections. The effect is that small changes in one coordinate measurement due to blur will have a correspondingly larger effect on the calculated 3D coordinates.

This study supports the findings made using subjective human opinions. Johnson and Casson (1995) proved that blur influences acuity. The current study also shows that blur influences photogrammetric image processing. Compared with human perception, image processing is more sensitive to blur. In addition, the detection of blurred simple structures, such as round targets, causes problems for automatic processes. Colombo et al. (1987) tested humans on the detection of structures in blurred images and found a decreasing ability to read text with increasing blur. However, the legibility of text is a much more complicated task than the detection of round targets.

This work supports the desirability of excluding blurred images from photogrammetric processing. Gülch (2012) recommends elimination of blurred images as a first step in UAV image processing. The findings also support the work of Shah and Schickler (2012) who develop blur correction methods specifically for UAV applications. Lelégard et al. (2012) state that a blur larger than 2 pixels is a significant amount. However, the findings reported in this paper suggest that a blur of just 2 pixels is actually too small to influence the detection, identification, referencing and measurement of targets.

Conclusions

A range of difficulties concerning photogrammetric applications is caused by image degradation due to motion blur. Small camera displacements have a significant impact on the accuracy of subsequent calculations and processes. Activating a shutter button on a camera can result in significant camera movement causing image blur. Fully automatic detection of targets in images containing small blurs can be difficult and can influence further processing. This problem can be solved by using semi-automatic detection tools. A small amount of blur has no significant influence on calculation results when targets are detected and measured successfully. If blur increases to such a degree that semi-automatic detection requires significant operator input, subsequent calculation will return significantly inaccurate results. However, manual measurements can be carried out and results of acceptable accuracy can be achieved, even in highly blurred images. It is also important to recognise that the tests were conducted only using signalised targets. It can be assumed that using natural feature points results in fewer detected features and feature referencing can be erroneous or even becomes impossible. Using natural features for processing blurred images would be an interesting topic for future work.

Although blur might disturb most image processing procedures it can be also exploited for some applications, as identified by Boracchi (2009) and McCarthy et al. (2013).

References

Boracchi, G., 2009. Estimating the 3D direction of a translating camera from a single motion-blurred image. *Pattern Recognition Letters*, 30(7): 671–681.

Boracchi, G., Caglioti, V. and Giusti, A., 2007. Ball position and motion reconstruction from blur in a single perspective image. *14th International Conference on Image Analysis and Processing ICIAP 2007*, Modena, Italy. Pages 87–92.

Colombo, E. M., Kirschbaum, C. F. and Raitelli, M., 1987. Legibility of texts: the influence of blur. *Lighting Research and Technology*, 19(3): 61–71.

Eisenbeiss, H., 2009. *UAV photogrammetry*. Doctoral thesis 18515 ETH, Zurich, Switzerland. 235 pages.

EISENBEISS, H., 2011. *Tutorial on UAV-g: introduction*. Presentation at UAV-g 2011, Zurich, Switzerland. http://www.geometh.ethz.ch/uav_g/tutorial/eisenbeiss_web [Accessed: 7th October 2014].

EISENBEISS, H. and SAUERBIER, M., 2011. Investigation of UAV systems and flight modes for photogrammetric applications. *Photogrammetric Record*, 26(136): 400–421.

GRENZDÖRFFER, G., NIEMEYER, F. and SCHMIDT, F., 2012. Development of four vision camera system for a micro-UAV. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39(B1): 369–374.

GÜLCH, E., 2012. Photogrammetric measurements in fixed wing UAV imagery. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39(B1): 381–386.

JOHNSON, C. A. and CASSON, E. J., 1995. Effects of luminance, contrast, and blur on visual acuity. *Optometry and Vision Science*, 72(12): 864–869.

JOSHI, N., SZELISKI, R. and KRIEGMAN, D. J., 2008. PSF estimation using sharp edge prediction. *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, USA. 8 pages. http://dx.doi.org/10.1109/CVPR.2008.4587834

KIM, S. K. and PAIK, J. K., 1998. Out-of-focus blur estimation and restoration for digital auto-focusing system. *Electronics Letters*, 34(12): 1217–1219.

KRAUS, K., 2004. *Photogrammetrie*. Seventh edition. De Gruyter, Berlin, Germany. 516 pages.

KRAUS, K., 2007. *Photogrammetry – Geometry from Images and Laser Scans*. Second edition. De Gruyter, Berlin, Germany. 459 pages.

LELÉGARD, L., DELAYGUE, E., BRÉDIF, M. and VALLET, B., 2012. Detecting and correcting motion blur from images shot with channel-dependent exposure time. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(3): 341–346.

LIU, R., LI, Z. and JIA, J., 2008. Image partial blur detection and classification. *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, USA. 8 pages. http://dx.doi.org/10.1109/CVPR.2008.4587465

LUHMANN, T., 2014. Eccentricity in images of circular and spherical targets and its impact on spatial intersection. *Photogrammetric Record*, 29(148): 417–433.

LUHMANN, T., ROBSON, S., KYLE, S. and BOEHM, J., 2014. *Close-Range Photogrammetry and 3D Imaging*. Second edition. De Gruyter, Berlin, Germany. 684 pages.

MAISON NICÉPHORE NIÉPCE, 2013. http://www.niepce.org/pagus/pagus-inv.html [Accessed: 24th June 2013].

MCCARTHY, D. M. J., CHANDLER, J. H. and PALMERI, A., 2013. Monitoring dynamic structural tests using image deblurring techniques. *Key Engineering Materials, Damage Assessment of Structures X*, 569–570: 932–939.

NARVEKAR, N. D. and KARAM, L. J., 2009. *A no-reference perceptual image sharpness metric based on a cumulative probability of blur detection*. International Workshop on Quality of Multimedia Experience, San Diego, California, USA. Pages 87–91.

ONG, E., LIN, W., LU, Z., YANG, X., YAO, S., PAN, F., JIANG, L. and MOSCHETTI, F., 2003. A no-reference quality metric for measuring image blur. *Seventh International Symposium on Signal Processing and its Applications*, Paris, France. 1: 469–472.

PACEY, R. and FRICKER, P., 2005. Forward motion compensation (FMC) – is it the same in the digital imaging world? *Photogrammetric Engineering & Remote Sensing*, 71(11): 1241–1242.

RAHTU, E., HEIKKILÄ, J., OJANSIVU, V. and AHONEN, T., 2012. Local phase quantization for blur-insensitive image analysis. *Image and Vision Computing*, 30(8): 501–512.

SACHS, D., NASIRI, S. and GOEHL, D., 2006. *Image stabilization technology overview*. http://www.invensense.com/jp/mems/gyro/documents/whitepapers/ImageStabilizationWhitepaper_051606.pdf [Accessed: 29th October 2013].

SHAH, C. A. and SCHICKLER, W., 2012. Automated blur detection and removal in airborne imaging systems using IMU data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39(B1): 321–323.

SHAKED, D. and TASTL, I., 2005. Sharpness measure: towards automatic image enhancement. *IEEE International Conference on Image Processing*, Genoa, Italy. I: 937–940.

SHORTIS, M. R. and SEAGER, J. W., 2014. A practical target recognition system for close range photogrammetry. *Photogrammetric Record*, 29(147): 337–355.

SIEBERTH, T., WACKROW, R. and CHANDLER, J. H., 2013. Automatic isolation of blurred images from UAV image sequences. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(1/W2): 361–366.

STILES, R. N., 1976. Frequency and displacement amplitude relations for normal hand tremor. *Journal of Applied Physiology*, 40(1): 44–54.

Résumé

Les drones sont devenus un sujet de recherche intéressant et dynamique en photogrammétrie. Les recherches actuelles s'intéressent à des images acquises depuis des drones qui permettent une haute résolution spatiale et spectrale grâce à un vol à basse altitude et à des caméras à haute résolution. L'un des principaux problèmes empêchant l'automatisation du traitement des images acquises par drone est la méconnaissance de l'effet du phénomène de filé produit par le mouvement de la caméra durant l'acquisition des images. L'objet de cet article est d'analyser l'influence du filé sur le traitement photogrammétrique des images. Des images ont été produites avec un effet de filé parfaitement connu pour en déterminer l'influence. On constate que même des filés très limités affectent de manière significative le traitement photogrammétrique. Une intervention de l'opérateur, bien que coûteuse en temps, permet de garantir des résultats de qualité acceptable.

Zusammenfassung

Unbemannte Luftfahrzeuge (UAV) sind ein interessantes und aktuelles Thema in der photogrammetrischen Forschung. Die mit den UAVs aufgenommenen Bilder weisen, aufgrund der geringen Flughöhe und der Nutzung hochauflösender Kameras, eine hohe geometrische und spektrale Auflösung auf. Es wird angenommen, dass die Bewegungsunschärfe die vollautomatische photogrammetrische Auswertung von UAV Bildern beeinträchtigt. Bisher ist jedoch unbekannt, in welchem Maße unscharfe Bilder den Prozess beeinflussen. Ziel dieser Veröffentlichung ist es, den Einfluss von Bewegungsunschärfe auf photogrammetrische Prozesse zu untersuchen. Dazu werden Bilder mit genau bekannter Bewegungsunschärfe erzeugt, um deren Einfluss auf photogrammetrische Operationen zu analysieren. Es wurde herausgefunden, dass schon eine geringe Unschärfe den Verlauf photogrammetrischer Prozeduren negativ beeinflusst. Manuelle Eingriffe können zwar akzeptable Ergebnisse sicherstellen, sind jedoch sehr zeitintensiv.

Resumen

Los vehículos aéreos no tripulados (UAV) se han convertido en un interesante y activo tema de investigación en fotogrametría. La presente investigación está basada en imágenes captadas desde UAVs que tienen una alta resolución espacial y buena resolución espectral gracias a la combinación de vuelos a baja altitud con cámaras de alta resolución. Uno de los principales problemas que dificultan la automatización del proceso de datos de imágenes de UAV es el efecto de la degradación en la definición ocasionado por el movimiento de la cámara durante la adquisición. El propósito de este trabajo es analizar la influencia de esta falta de definición en el proceso de imágenes fotogramétricas. Se producen imágenes con degradación de la definición causado por el movimiento conocida para determinar su efecto. Se encontró que incluso pequeñas degradaciones de la definición afectan significativamente a los procesos normales fotogramétricos. Aunque la intervención manual conlleva mucho tiempo sin embargo puede garantizar resultados aceptables.

摘要

无人机已经成为摄影测量领域一个非常活跃的研究方向,现在的研究主要集中在利用低空飞行无人机配备的高分相机获取高空间分辨率和高光谱分辨率遥感影。现在阻碍无人机影像全自动化处理的主要问题是在影像采集过程由于相机运动引起的模糊。本文分析了该类模糊对摄影测量影像处理的影响,结果表明,小的模糊对摄影测量处理的影响是非常大的。虽然操作员人机交互可能比较耗费时间,但是齐可以确保处理结果达到可接受的摄影测量精度。

B.2 Peer reviewed

UAV Image Blur - Its Influence and Ways to Correct it

T. Sieberth [a], R. Wackrow [a], J. H. Chandler [a]
[a] Loughborough University, School of Civil and Building Engineering, Loughborough University, LE11 3TU, United Kingdom
(T.Sieberth, R.Wackrow, J.H.Chandler)@lboro.ac.uk

KEY WORDS: Blur, Correction, Detection, Digital, Image processing, Photrogrammetry, UAV

ABSTRACT:

Unmanned aerial vehicles (UAVs) have become an interesting and active research topic in photogrammetry. Current research is based on image sequences acquired by UAVs which have a high ground resolution and good spectral resolution due to low flight altitudes combined with a high-resolution camera. One of the main problems preventing full automation of data processing of UAV imagery is the unknown degradation effect of blur caused by camera movement during image acquisition.

The purpose of this paper is to analyse the influence of blur on photogrammetric image processing, the correction of blur and finally, the use of corrected images for coordinate measurements. It was found that blur influences image processing significantly and even prevents automatic photogrammetric analysis, hence the desire to exclude blurred images from the sequence using a novel filtering technique. If necessary, essential blurred images can be restored using information of overlapping images of the sequence or a blur kernel with the developed edge shifting technique. The corrected images can be then used for target identification, measurements and automated photogrammetric processing.

1. INTRODUCTION

A constraint enforced on the acquisition of early photographs used in photogrammetry was a stable camera position and a stationary object as the basic requirements for sharp images. Exposure times of many days were required to obtain an image (Maison Nicéphore Niépce, 2013}. Today, professional photographers use a tripod and remote shutter release to prevent camera movement during image acquisition. Unfortunately, unmanned aerial vehicles (UAVs) rarely provide a stable camera position. UAVs are affected by wind, turbulence, sudden input by the operator and also by the flight movement of the aircraft itself as well as vibrations of the engines. Hence, the acquired image sequence can contain blurred imagery which disturbs automatic post processing. Filtering these blurred images has to be carried out manually, which is exhausting for the eyes, prone to errors and time extensive. By filtering out blurred images the number of images is reduced and can negatively influence the result of post processing. This paper describes a series of experimental work: how a dataset of images was created with known blur characteristics; the effect of blurred imagery on photogrammetry and image processing; how to correct blurred images and subsequent image operations.

2. RELATED WORK

Photogrammetry is the science of reconstructing 'the position, orientation, shape and size of objects from pictures' (Kraus, 2007). To provide appropriate image geometry for 3D measurements a recommended along track overlap of 60% and a lateral overlap of 20% should be used (Kraus, 2004; Luhmann et al., 2006}. To calculate accurate 3D coordinates (X, Y, Z) for an object point, it is necessary to precisely measure the image coordinates (x, y) in at least two images. Today, fully automated data processing is demanded by an increasing number of users. Blur is expected to influence the accuracy and ability of image coordinate measurements, which is analysed in this study.

A series of methods have been developed in previous work to detect if an image is sharp or blurred. It is recognised that improving the image sharpness using deblurring algorithms is an important topic in computer vision and image processing. A widely used application for automatic blur detection is the 'auto-focus' system in cameras, which should prevent the user from taking optically blurred images caused by an inappropriate focal setting (Kim et al., 1998). As optical blur can be prevented using these systems, other methods are required to suppress blur due to motion, which should be carefully distinguished as a discrete type of blur. Motion blur is often caused by human hand movement (jitter). This jitter has frequencies of 2-10 Hz, with amplitudes of up to 1 mm (Stiles, 1976). Sachs et al. (2006) note that there is also a 'drift of the hand' of up to 5 mm/s and that commercial systems for shake reduction use gyroscopes to prevent motion blur.

Methods developed for aircraft include precise inertial measurement unit (IMU) and global navigation satellite system (GNSS) information, to reduce angular and forward motion blur (Pacey and Fricker, 2005). Both sensors can provide additional information for blur detection algorithms such as the approximate path followed when the image was blurred. Precise IMUs are also used to generate blur kernels which represent how an image is blurred. The kernel visualises how a point in an image would be blurred, which path it would have taken and which motion speed it would have had. IMUs used by most UAVs are too imprecise to establish a basic precise blur kernel. Assuming an image exposure time of 1/400 s would require an IMU of at least 800 Hz to measure a basic blur kernel for an image (Grenzdörffer et al., 2012). A blur kernel based on two measurements would be only a linear representation of the motion and would not represent more precise motion. However, even if linear representation of motion blur is sufficient to represent typical UAV flight motion, such systems are expensive and cannot always be used since these would exceed the payload of typical micro UAV platforms. Furthermore, the IMU and GNSS sensors incorporated for flight stabilisation are not accurate enough and lack sufficient acquisition frequency to

determine if an image is blurred or not. This makes detection of motion blur during the post processing stage often necessary.

Blind blur detection algorithms normally use edge detection or frequency analysis. Edge detection methods focus on the spread and gradient of an edge (Joshi et al., 2008; Narvekar and Karam, 2009; Ong et al., 2003). In the case of a widely spread edge of low gradient it is assumed that the image is blurred. This edge smear due to blur is also used in image frequency analysis methods. When an image has extensively smeared edges in the spatial domain, it is characterised by the disappearance of high frequencies in the frequency domain (Liu et al., 2008; Rahtu et al., 2012). A problem with most blur detection methods is the presumption that the image is blurred and that these methods are developed using mathematically blurred images without any relation to geometrical motion blur or random hand held camera shake (Levin et al., 2009). In such circumstances the amount of blur is unknown and has to be evaluated subjectively. It is obvious that although subjective evaluation of blur maybe useful as fast and omnipresent method to evaluate images, it can be wrong. This explains the desire to develop a measure of blur, by generating images with a precisely known image blur which is not influenced by human perception. Using a blur kernel to represent blur is possible. However, comparing the blur between different images based on blur kernels is difficult due to the complexity, their shape and extent (Sieberth et al., 2013).

There are various methods for image deblurring which have been published over the last decades. Image deblurring approaches can be separated into blind and non-blind deconvolution. Non-blind image deconvolution uses a-priori knowledge about the blur in the image. Blind deconvolution has only the blurred image and no additional information but the task of deriving a sharp image remains. Wiener deconvolution (Wiener, 1950) and Richardson-Lucy deconvolution (Lucy, 1974; Richardson, 1972) are blind deconvolution methods proposed decades ago and remain popular because they are both simple and efficient. A significant problem that occurs using these methods are ringing artefacts at high contrast edges (Fig. 1). More advanced methods are often based on a probability calculation, which is also used by Wiener and Richardson-Lucy deconvolution such as for example (Sun et al., 2014). They aim to improve deblurring and the capability to work reliably, even with noise in the blurred image (Shan et al., 2008). There are also many other methods such as (Krishnan and Fergus, 2009) who does 'Fast Image Deconvolution using Hyper-Laplacian Priors' or (Michaeli and Irani, 2014) who uses recurrence of image patches.

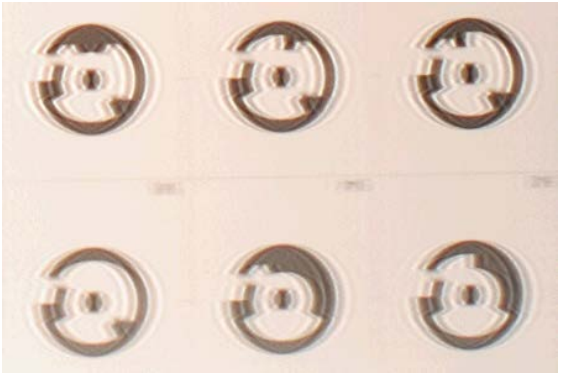


Figure 1. Image deblurred with Lucy-Richardson deconvolution (10 Iterations). The ringing artefacts are clearly visible.

Non-blind image deconvolution methods can be carried out in various ways and requires additional knowledge. Additional information can be gained through a variety of methods including, other overlapping images (Agrawal et al., 2012), precise IMU measurements (Joshi et al., 2008), video cameras (Tai et al., 2010), fluttering shutters (Raskar et al., 2006) or colour channel dependent exposure times (Lelégard et al., 2012). The main aim of these methods is to establish an appropriate blur kernel, which can be used by established deblurring algorithms.

3. PRACTICAL WORK

The practical work described here is focused on three different problems associated with blur. First, how blurred images were created; second, how blur disturbs photogrammetic image processing; and finally, how blur may be corrected.

3.1 Generating blurred images

A mathematical low-pass filter could be used to simulate motion blur but would decrease noise and other hardware influences and return a non-realistic result (Sieberth et al., 2013). Instead of using image processing a shaker-table was used to acquire motion blurred images, which are influenced by typical degrading effects introduced by the camera hardware (Fig. 2).

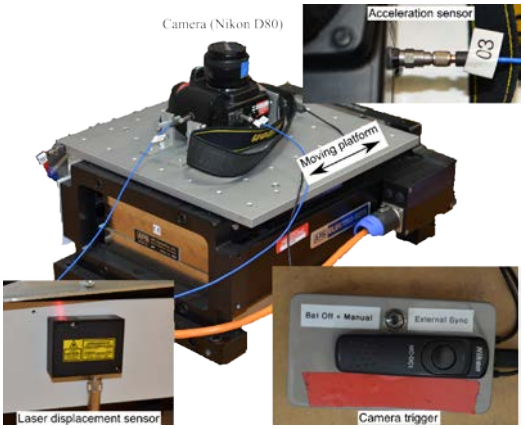


Figure 2. Shaker table with camera mounted on it.

A shaker table is a platform used by construction engineers to investigate the likely response of a building to vibrating influences like earth quakes. It can be displaced with a known magnitude and frequency and movement of objects mounted on it can be monitored with displacement and acceleration sensors. Here the device was used to mount a camera (Fig. 2). If the device is fitted with acceleration sensors, it is possible to determine precisely when the camera shutter opened and closed due to vibrations caused by the shutter movement. This can be combined with the displacement sensor allowing an accurate displacement of the camera to be calculated for each image (Fig. 3). However, the vibrations caused by the moving shutter remain small and do not significantly influence the camera (Sieberth et al., 2013). By using the displacement sensor, camera displacement can be measured precisely during image exposure. This precise measure provides a higher control for further processing.

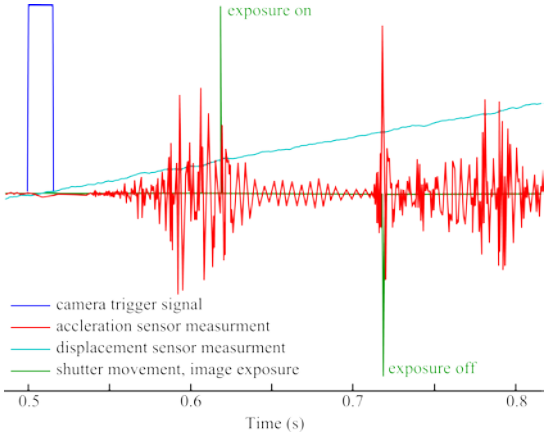


Figure 3. Image exposure with camera displacement.

3.2 Blur disturbs photogrammetry

A set of generated motion blurred images was used to analyse the effect of blur on photogrammetric data processing. This was achieved by processing blurred images, of high contrast targets, using photogrammetric and image processing procedures (Fig. 4). A range of typical photogrammetric procedures were examined and tested. The first concentrated on camera calibration used to determine the interior geometry of the camera (Sieberth et al., 2014b). The second procedure assessed the detection and registration of feature points in sharp and overlapping blurred images (Sieberth et al., 2014a).

3.3 Blur correction

After quantifying how much feature detection is influenced by blur and if image overlap can be calculated successfully, it can be decided which technique for deblurring can be used. One method developed for deblurring in this research uses information provided by an overlapping image determined after image registration. However, if an appropriate image overlap is unavailable, deblurring needs to be carried out using an independent approach.

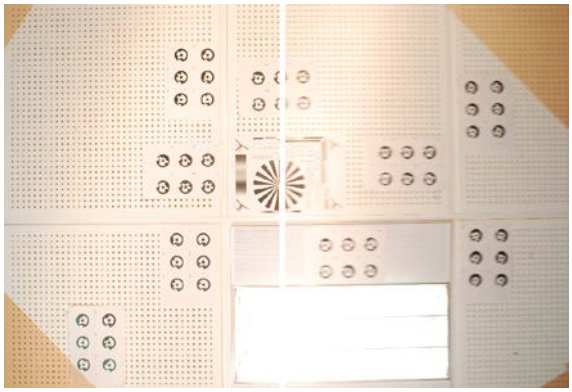


Figure 4. Sharp and blurred image (1.03 mm camera displacement) of the dataset.

3.3.1 Frequency transfer method using an overlapping image

The first method, using an image overlap is the frequency transfer method. This method uses a sharp overlapping image to deblur a blurred image. A perspective transformation is applied to the blurred image and both the sharp and the transformed blurred image are cropped to just the overlapping area. Then the images are transferred to the frequency domain using a Fourier transformation. It is well established that high frequencies are absent in blurred images (Lelégard et al., 2012). The absence of high frequencies can be compensated by enhancing the blurred image using high frequencies extracted from the sharp image. Afterwards, the enhanced image is transferred back to the spatial domain. As this frequency operation only works on single channel images, it is either possible to apply this method on each channel separately or to use just a grey scale image. If a grey scale image is used, the enhanced image can replace the intensity channel of the originally blurred image (Fig. 5). The approach relies upon an overlapping sharp image to correct the blurred image (Sieberth et al., 2014a) and if unavailable then an alternative approach is required.

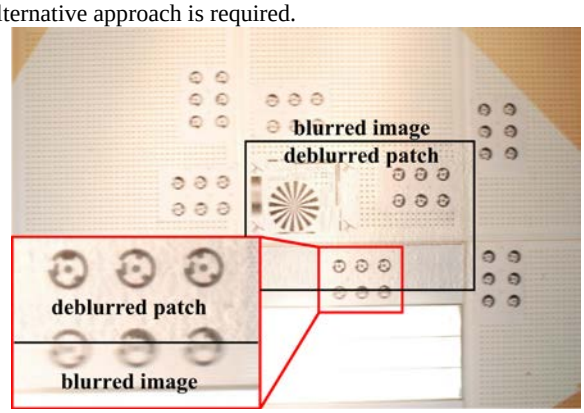
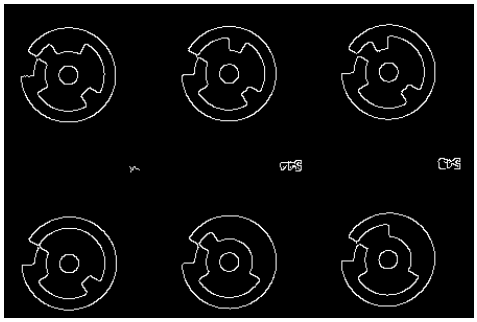


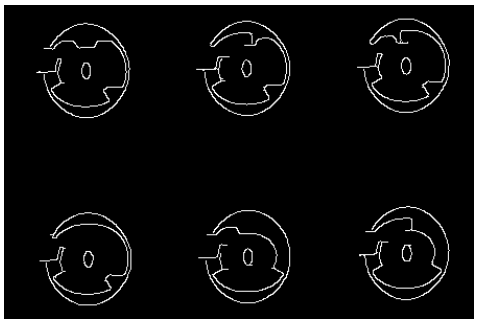
Figure 5. Deblurring using frequency transform method with 1.03 mm camera displacement.

3.3.2 Edge shift - an overlap independent method

If image registration is unsuccessful or no sharp image overlap is available then the frequency transfer method cannot be used and a different approach needs to be applied. The developed edge shifting approach focuses on high contrast edges. Suitable high contrast edges can include photogrammetric targets, which are often used for photogrammetric applications. They can normally be detected, identified and measured automatically in every image and the image registration can be conducted. Blur can prevent this automated process. These high contrast boundaries enable edge detection methods to find an edge in the image even if it is blurred. However, the blur causes a displacement of the edge (Fig. 6(b)), which is shifted by the size of the blur (Fig. 6). Shifting the displaced edges back to their original position provides the possibility to automatically detect the target and identify the target number based on the target code. To measure the target, the area of the target can be masked and the image moment (gravity centre) can be calculated, to represent the centre of the target (Fig. 6(c)). Using the target ID and coordinate measurement, allows the registration and orientation of the image. Unfortunately, the edge shifting approach returns just an edge image without colour information. However, after image registration it is possible to use the frequency transfer to generate a colour image when an overlapping image exists, which is sharper than the just processed image alone.



(a) Original sharp targets.



(b) Blurred targets (0.53 mm camera displacement).

(c) Restored targets with target number in the centre, binary code and measurements (centre of target was neglected)

Figure 6. Distortion of targets and target code due to blur to elliptic shapes.

4. RESULTS AND DISCUSSION

The presented methods are effective and compensating for various stages of motion blur. All methods involved in this progress return novel and interesting results, which are explained further below.

4.1 Motion blurred images

Generating blurred images using a shaker table provides an accurate method to produce images with known characteristics. The fitted acceleration sensors are sensitive enough to recognise camera displacement and also the much smaller shake of the camera body caused by the shutter opening and closing. It was found that the vibration caused by the shutter is not significant, so that it can be neglected from further blur calculations (Sieberth et al., 2013). For further calculation of blur, the shaker table provides the opportunity to create many images with different blur characteristics under stable laboratory conditions. The same scene can be photographed with the same light conditions and with a displacement defined a-priori. However, one feature of the shaker table is that it is only able to simulate blur in one direction. This is appropriate because blurred images due to flight motion only contain one directional movement and not complicated blur paths. Forward motion as well as pitch and roll result in linear motion blur in the image. However, the combination of yaw rotation with forward motion would cause an almost linear blur. Only rotations around the yaw axis would cause a radial blur. Linear motion is also supported by the short exposure time used by UAVs and it can be assumed that complicated motions rarely influence the camera. A disadvantage is the limited displacement range of the table, which is less than 150 mm. In contrast to potential displacements created by UAVs, this displacement is very small. Although, real UAV images exhibit larger displacements, these occur at larger camera to object distances. Fortunately, these are equivalent to smaller displacements acquired at short object distances. Based on this assumption, the conducted laboratory experiments can be considered as representative. A major advantage of the methodology used is that the dataset is characterised by images with real motion blur, in contrast to other datasets which contain mathematically generated motion blur (e.g. 'LIVE Database for image and Video Quality Assessment' (Sheikh et al., 2005)). The images created using the shaker table contain known blur defined by the camera displacement. This provides a higher control for following processing steps as calculated blur kernels can be compared to the true camera displacement.

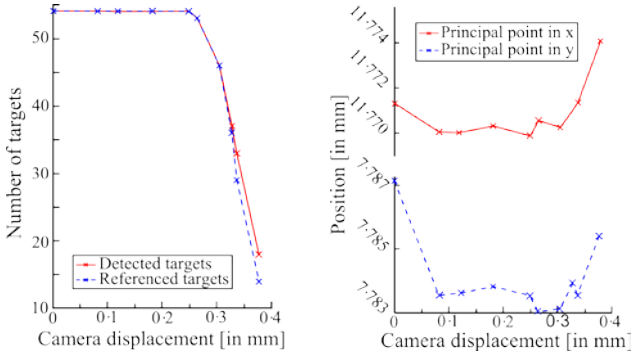
4.2 Blur disturbs

After generating images with known motion blur, a range of tests were conducted. The first aimed to analyse whether and by how much blur disturbs normal photogrammetric processes (Sieberth et al., 2014b). It was found that it is difficult for the human eye to precisely quantify the amount of blur and assess whether it will disturb procedures adversely. This becomes even more difficult if there is no sharp image available for visual comparison. Even images of apparently high visual quality can cause problems, if a small amount of blur is present. The number of measured targets directly influences the accuracy and ability of subsequent calculations (Fig. 7). Measurements of blurred signalised targets become inaccurate and incorrect (Fig. 7(a)) therefore degrading subsequent procedures such as the calculation of the principal point (Fig. 7(b)). It can be assumed that detection and measurement of unsignalised targets is worse and again negatively influences the results. Additionally, feature detection algorithms are also affected, which can be identified in terms of both the number of detected features and correctly referenced features (Table 1).

Camera displacement [mm]	Detected feature points	Correct referenced feature points
0	12214 (100%)	12214 (100%)
0.377	8847 (72%)	1524 (17%)
0.529	7370 (60%)	224 (3%)
1.028	2645 (22%)	47 (1.8%)

Table 1. Camera displacement while camera shutter is open vs. detected and correctly referenced feature points.

Table 1 presents how many features were detected in images containing different camera displacements. With increasing displacement the number of detected features decreases. Furthermore, Table 1 does show how many of the detected features were referenced correctly to features detected in a sharp image. The number of correct referenced features decreases significantly with increasing camera displacement.



(a) Detected and referenced targets.

(b) Influence of camera displacement during image exposure upon the calculated principal point position.

Figure 7. Negative influence of blur on target detection and camera calibration (Sieberth et al., 2014b).

The results supports the findings of Johnson and Casson (1995) who proved that human perception is influenced by blur. Furthermore, it shows also that blur influences photogrammetric image processing and in contrast to human perception, image processing procedures are more sensitive to blur. The detection of blurred simple structures such as round targets certainly causes problems for automatic procedures.

Colombo et al. (1987) tested humans on the detection of structures in blurred images and found a decreasing ability to read text with increasing blur. However, the legibility of text is a much more complicated task than detection of round targets.

The results support the desirability to exclude blurred images from photogrammetric processing. Gülch (2012) recommends elimination of blurred images as a first step in UAV image processing. The findings also support work of Shah and Schickler (2012) who developed blur correction methods specifically for UAV applications. Lelégard et al. 2012 states that blur larger than 2 pixels is significant. Our findings would contradict this, suggesting that a blur of just 2 pixels is actually too small to influence detection, identification, referencing and measurement.

4.3 Blur correction

To prevent the negative influence of blurred images on post processing, blurred images should be detected and deblurred. The frequency transfer method appears suitable to correct some blur, if overlaps are available and one image is sharp. If the image used for deblurring is itself blurred, then the enhanced result maybe better than the original blurred image but will contain at least as much blur as the 'sharper' image originally contained. Adding frequencies to a blurred image also introduces some noise after transforming from the frequency domain back to the spatial (Fig. 5). In the subsequent step of image transformation, which is necessary to regain the correct position, the rotation, shear and scale and the interpolation of the correct pixel intensities for the rectified image, can cause a 'blurring' effect. It could also happen that the transformation parameters of the images are unknown, which makes deblurring using an overlapping image, impossible. Overlapping images are often available in photogrammetric procedures, however, it can also happen that not enough overlapping images are available and the frequency transfer method cannot be applied. This frequency transfer is based on the idea of Jung et al. (2009). However, our method presented here uses images which are not highly overlapping or taken from the same camera point of view, which makes it applicable for aerial imagery.

The newly developed edge shifting approach is a method to deblur images containing high contrast edges. High contrast edges are often used for image registration, as these can be clearly detected and measured in image sequences. Standard deblurring methods like Wiener deconvolution or Lucy-Richardson deconvolution, produces ringing artefacts on high contrast edges making automatic detection of features significantly difficult (Fig. 1). In contrast, edge shifting does not produce these artefacts and makes automatic detection possible. It is also faster than standard deconvolution methods because it concentrates on edges only. However, the main problem is the loss of colour information, which needs to be restored afterwards by using a flood fill method. More significant problems are the many special cases and exceptions which occur. Handling these exceptions creates a range of challenges. It is possible that the shifting process generates holes in the shapes. This is caused by shifting two edges apart from one another, which gives the appearance of 'ripping' them apart. Furthermore, edges which have completely disappeared cannot be recovered. Deblurring targets which are smaller than the blur also need to be considered as special cases. Deblurring real images with different edge intensities, directions and blur length remains more challenging than in the high contrast test image. Despite these unsolved cases it is possible to achieve edge shifting using photogrammetric targets, to enable sub-pixel accurate measurement of target coordinates. Furthermore, it might be possible to incorporate different blur in different positions of an image by calculating localised blur kernels. This would be necessary for blurred objects close to the camera, which appear more blurred compared to objects further away. The method developed here is completely different to previously published methods, because it does not concentrate on restoring colour information using probability theory (Lucy, 1974; Richardson, 1972; Shan et al., 2008; Wiener, 1950) but only edges. This provides the additional advantage of being much faster than methods previously published.

5. CONCLUSION

The study reported in this paper shows that it is difficult to generate blurred images that contain a known path and amount of camera displacement. However, a shaker table was found to be suited and preferred to simply simulating blur using a low pass filter. Precisely blurred images are required to analyse if and when blur degrades a photogrammetric processing. It was shown that even small camera displacements have a significant impact on the accuracy of subsequent photogrammetric image processing. A threshold can be defined, which can be used to filter blurred images and/or apply deblurring algorithms (Sieberth et al., 2014b). If deblurring is necessary or required, then overlapping images can be used to achieve reasonably sharp results for further processing. If overlapping images are unavailable then it is still possible to use the 'edge shifting' approach for image deblurring. This novel approach can restore images, provided that high contrast edges are available.

ACKNOWLEDGEMENTS

I want to thank RSPSoc for sponsoring my attendance at this conference.

REFERENCES

Agrawal, A., Xu, Y. and Raskar, R., 2012. Multi-image deblurring. US Patent 8,229,244.

Colombo, E. M., Kirschbaum, C. F. and Raitelli, M., 1987. Legibility of texts: The influence of blur. *Lighting Research and Technology*, 19(3), pp. 61-71.

Grenzdörffer, G., Niemeyer, F. and Schmidt, F., 2012. Development of four vision camera system for a micro-UAV. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Melbourne, Australia, Vol. XXXIX, Part B1, pp. 369-374.

Gülch, E., 2012. Photogrammetric measurements in fixed wing UAV imagery. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Melbourne, Australia, Vol. XXXIX, Part B1, pp. 381-386.

Johnson, C. and Casson, E., 1995. Effects of luminance, contrast and blur on visual acuity. *Optometry and Vision Science*, 72(12), pp. 864-869.

Joshi, N., Szeliski, R. and Kriegman, D. J., 2008. PSF estimation using sharp edge prediction. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8.

Jung, S.-W., Kim, T.-H. and Ko, S.-J., 2009. A novel multiple image deblurring technique using fuzzy projection onto convex sets. *IEEE Signal Processing Letters*, 16(3), pp. 192-195.

Kim, S. K., Park, S. R. and Paik, J. K., 1998. Simultaneous out-of-focus blur estimation and restoration for digital auto-focusing system. *IEEE Transactions on Consumer Electronics*, 44(3), pp. 1071-1075.

Kraus, K., 2004. *Photogrammetrie*. Walter de Gruyter, Berlin, Germany.

Kraus, K., 2007. *Photogrammetry - Geometry from images and laser scans*. Walter de Gruyter, Berlin, Germany.

Krishnan, D., Fergus, R., 2009. Fast image deconvolution using hyper-laplacian priors. *Advances in Neural Information Processing Systems*, 22. 1-9.

Lelégard, L., Delaygue, E., Brédif, M. and Vallet, B., 2012. Detecting and correcting motion blur from images shot with channel-dependent exposure time. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, I(3), pp. 341-346.

Levin, a., Weiss, Y., Durand, F. and Freeman, W., 2009. Understanding and evaluating blind deconvolution algorithms. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1964-1971.

Liu, R., Li, Z. and Jia, J., 2008. Image partial blur detection and classification. In: 2008 *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8.

Lucy, L. B., 1974. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79, pp. 745-754.

Luhmann, T., Robson, S., Kyle, S. and Harley, I., 2006. *Close range photogrammetry*. Whittles Publishing.

Maison Nicéphore Niépce, 2013. *Invention of photography*. http://www.photo-museum.org/en/ (11/03/2015)

Michaeli, T., Irani, M., 2014. Blind deblurring using internal patch recurrence. *Proceedings of the European Conference on Computer Vision*, pp. 783-798.

Narvekar, N. D. and Karam, L. J., 2009. A No-reference perceptual image sharpness metric based on a cumulative probability of blur detection. *Quality of Multimedia Experience*, pp. 87-91.

Ong, E., Lin, W., Lu, Z., Yang, X., Yao, S., Pan, F., Jiang, L. and Moschetti, F., 2003. A no-reference quality metric for measuring image blur. *Seventh International Symposium on Signal Processing and Its Applications*, pp. 491-498.

Pacey, R. and Fricker, P., 2005. Forward motion compensation (FMC)-is it the same in the digital imaging world? *Photogrammetric Engineering and Remote Sensing*, 71(11), pp. 1241-1242.

Rahtu, E., Heikkilä, J., Ojansivu, V. and Ahonen, T., 2012. Local phase quantization for blur-insensitive image analysis. *Image and Vision Computing*, 30(8), pp. 501-512.

Raskar, R., Agrawal, A. and Tumblin, J., 2006. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Transactions on Graphics*, 25(3), pp. 795-804.

Richardson, W. H., 1972. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1), pp. 55-59.

Sachs, D., Nasiri, S. and Goehl, D., 2006. Image stabilization technology overview.

Shah, C. A. and Schickler, W., 2012. Automated blur detection and removal in airborne imaging systems using IMU data. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Melbourne, Australia, Vol. XXXIX, Part B1, pp. 321-323.

Shan, Q., Jia, J. and Agarwala, A., 2008. High-quality motion deblurring from a single image. *ACM Transactions on Graphics*, 27(3), pp. 73:1-73:10.

Sheikh, H. R., Wang, Z., Cormack, L. and Bovik, A. C., 2005. LIVE image quality assessment database release 2. http://live.ece.utexas.edu/research/quality (11/03/2015)

Sieberth, T., Wackrow, R. and Chandler, J., 2013. Automatic isolation of blurred images from UAV image sequences. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Rostock, Germany, Vol. XL-1, Part W2, pp. 361–366.

Sieberth, T., Wackrow, R. and Chandler, J. H., 2014a. Influence of blur on feature matching and a geometric approach for photogrammetric deblurring. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Zurich, Switzerland, Vol. XL-3, pp. 321-326.

Sieberth, T., Wackrow, R. and Chandler, J. H., 2014b. Motion blur disturbs - The influence of motion-blurred images in photogrammetry. *The Photogrammetric Record*, 29(148), pp. 434-453.

Stiles, R. N., 1976. Frequency and displacement amplitude relations for normal hand tremor. *Journal of Applied Physiology*, 40(1), pp. 44-54.

Sun, L., Cho, S., Wang, J., Hays, J., 2014. Good image priors for non-blind deconvolution: generic vs specific. *Proceedings of the European Conference on Computer Vision*, 2014. pp. 1-16.

Tai, Y.-W., Du, H., Brown, M. S. and Lin, S., 2010. Correction of spatially varying image and video motion blur using a hybrid camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6), pp. 1012-1028.

Wiener, N., 1950. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. Press of MIT and John Wiley & Sons Inc., New York, United States.

INFLUENCE OF BLUR ON FEATURE MATCHING AND A GEOMETRIC APPROACH FOR PHOTOGRAMMETRIC DEBLURRING

T. Sieberth [a, *], R. Wackrow [a], J. H. Chandler[a]

[a] Loughborough University, School of Civil and Building Engineering, Leicestershire, LE11 3TU, United Kingdom - (T.Sieberth, R.Wackrow, J.H.Chandler)@lboro.ac.uk

KEY WORDS: Blur, Computer, Vision, Feature, Matching, Photogrammetry, UAV

ABSTRACT:
Unmanned aerial vehicles (UAV) have become an interesting and active research topic for photogrammetry. Current research is based on images acquired by a UAV, which have a high ground resolution and good spectral and radiometric resolution, due to the low flight altitudes combined with a high resolution camera. UAV image flights are also cost efficient and have become attractive for many applications including change detection in small scale areas.

One of the main problems preventing full automation of data processing of UAV imagery is the degradation effect of blur caused by camera movement during image acquisition. This can be caused by the normal flight movement of the UAV as well as strong winds, turbulence or sudden operator inputs. This blur disturbs the visual analysis and interpretation of the data, causes errors and can degrade the accuracy in automatic photogrammetric processing algorithms.

The aim of this research is to develop a blur correction method to deblur UAV images. Deblurring of images is a widely researched topic and often based on the Wiener or Richardson-Lucy deconvolution, which require precise knowledge of both the blur path and extent. Even with knowledge about the blur kernel, the correction causes errors such as ringing, and the deblurred image appears "muddy" and not completely sharp. In the study reported in this paper, overlapping images are used to support the deblurring process, which is advantageous. An algorithm based on the Fourier transformation is presented. This works well in flat areas, but the need for geometrically correct sharp images may limit the application. Deblurring images needs to focus on geometric correct deblurring to assure geometric correct measurements.

1. INTRODUCTION

Constraints enforced on the acquisition of photographs for photogrammetry normally include a stable camera position and a stationary object. Unfortunately, lightweight small scale unmanned aerial vehicles (UAV) rarely provide a stable camera position. UAVs are easily affected by wind, gusts, turbulence or sudden operator inputs. However, their good manoeuvrability and flight path control combined with endurance, flight range and low cost make UAVs applicable for a range of different tasks (Eisenbeiß, 2009). The limited payload, regulatory restrictions and vulnerability of the UAV platforms encourage the use of low cost sensors, which often dictates the use of consumer grade cameras (Eisenbeiß, 2011). Unfortunately, the problem remains that the high spatial resolution of an image is often degraded due to motion blur. Since optical blur can be reduced using automatic focusing methods, motion blur remains a challenge. Resolving motion blur is an important and often researched topic in signal and image processing. The focus of many deblurring methods is often on spectral, radiometrical and geometrical deconvolution, which requires extensive calculations.

1.1 Aims and Objectives

Photogrammetry is defined as finding 'position, orientation, shape and size of objects from pictures' (Kraus, 2007). This is achieved by measuring coordinates of objects appearing in images. These measurements are mostly based on edge detection in the images because edges can often be detected and identified in overlapping images. In blurred images edges can disappear, change their position or become difficult to identify due to the degrading effect of motion blur (Fig 1).
Most methods use one blur kernel which represents the camera path during image acquisition. Using this general blur kernel for all pixels in the image is inaccurate because objects with a short camera to object distance are blurred more than objects further away. Also sensor rotation can change the blur kernel as the kernel is shorter for objects close to the rotation axis and larger

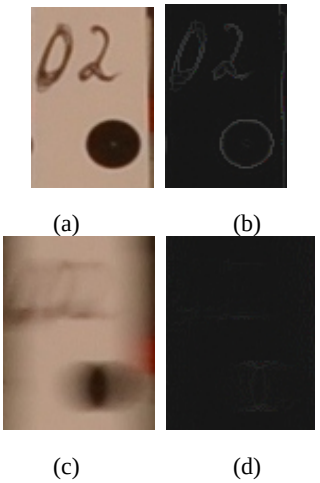


Figure 1. Example of a photogrammetric target. (a) Sharp image without any camera movement during image exposure. (b) The target is easy identifiable due to the well-defined edge between black and white. (c) Image blurred due to motion during image exposure. (d) The target is difficult to identify due to the difficulty to define the target outline.

* Corresponding author.

for objects far away from the rotation axis. This needs to be considered for generating a precise deblurring method capable of working in 3D.

This paper aims to analyse how much blur influences the detection of image overlap using image processing. Furthermore, image deblurring approaches suitable for photogrammetric purposes will be examined and discussed.

2. RELATED WORK

Aerial photogrammetry uses object information in images acquired by airborne platforms ranging from aircrafts to balloons. A platform recently developed and increasingly popular for image flights are lightweight UAVs, which are vulnerable to gusts, turbulence and are sensitive to operator inputs.

For a successful image flight it is necessary to carefully prepare the flight plan beforehand. To provide appropriate image geometry for 3D measurements a recommended image overlap of 60% along track and 20% across track should be used (Kraus 2004, Luhmann 2006). To calculate precise 3D coordinates for an object point, it is necessary to precisely measure the image coordinates in at least two images (Kraus, 2004). Additionally the exterior orientation and interior parameter of the camera are required. These parameters are required to fulfil the collinearity equation, which enables the 3D coordinates of points to be determined (Luhmann, 2006).

To carry out point measurements in multiple images automatically, feature points (signalised and unsignalised) need to be identified across frames. This is today based on the Scale-Invariant Feature Transform (SIFT), Speed-Up Robust Features (SURF) or similar algorithms. These algorithms detect suitable candidate features in the images, which are invariant to scaling, translation and rotation (Lowe, 2004, Bay et al., 2006). Similar features detected in two overlapping images can be referenced between them using least squares matching or other similar methods (Brown, 1992). Due to blur the same features can appear differently in the images and matching these becomes increasingly difficult. Even measuring well defined targets using automatic measurement methods and matching them between two images is influenced or even impossible (Sieberth et al., unpublished). These difficulties suggest the need to explain the influence of blur in photogrammetric processes.

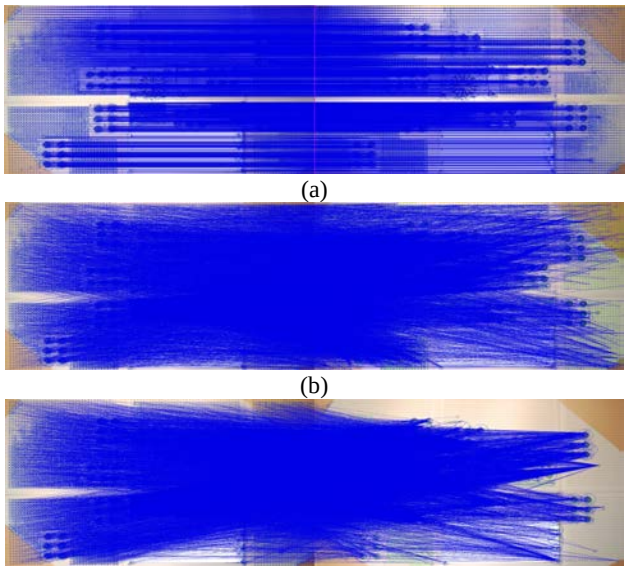
There are various methods for image deblurring which have been published over the last decades. Image deblurring approaches can be separated into blind and non-blind deconvolution. Non-blind image deconvolution uses a-priori knowledge about the blur in the image. Blind deconvolution has only the blurred image and no additional information but the task of deriving a sharp image remains. Wiener deconvolution (Wiener, 1949) and Richardson-Lucy deconvolution (Richardson, 1972, Lucy, 1974) are blind deconvolution methods proposed decades ago and remain popular because they are both simple and efficient. A significant problem that occurs using these methods are ringing artefacts at steep edges. More advanced methods are often based on a probability calculation, which is also used by Wiener and Richardson-Lucy deconvolution. They aim to improve deblurring and the capability to work reliably, even with noise in the blurred image (Shan et al, 2008).

Non-blind image deconvolution methods can be carried out in various ways and requiring additional knowledge. The additional information can be gained through other overlapping images (Agrawal, 2012), precise IMU measurements (Joshi, 2008), video cameras (Tai et al., 2008), fluttering shutters (Raskar et al., 2006) or colour channel dependent exposure times (Lelégard, 2010). The main aim of these methods is to establish an appropriate blur kernel, which can be used by established deblurring algorithms.

3. METHOD DEVELOPMENT

3.1 Feature Detection

To analyse how critical motion blur influences feature detection, images with precisely known blur were generated. Blurred images were acquired using a Nikon D80 SLR camera, and a shaking table using a methodology described more fully in a paper by Sieberth et al. (2013). Briefly, this involved generating four images with different camera displacements each of between 0 to 1 mm. Then these images were processed using the SURF function provided by OpenCV (SurfFeatureDetector), to detect feature points in the images (OpenCV, 2014). In the following step a “brute force matching method” (BFMatcher) was used to connect detected feature points. The connection was made between the feature points in the sharp image and each of the blurred images. The connection lines between sharp and blurred image should be parallel as shown in Figure 2 (a). Many of the returned matches were incorrect and additional filtering of the matches was required. Figure 2 shows the sharp image on the left and a blurred image on the right, with blue lines show the connection between sharp and blurred feature points. Figure 2 demonstrates that increasing camera displacement influences both the number of feature points and correctness of feature matching. In (b) and (c) the connections are converging on a few features which must be incorrect because the blurred image is made with the same orientation and from the same position as the sharp image and shows the same scene. Filtering out the incorrect matches can be done using an approximate position and orientation of the images. In real UAV images this can be extracted from recorded GNSS and IMU data. Using the flight altitude and the camera perspective centre for both images the offset between the images can be calculated. By using this method it can be

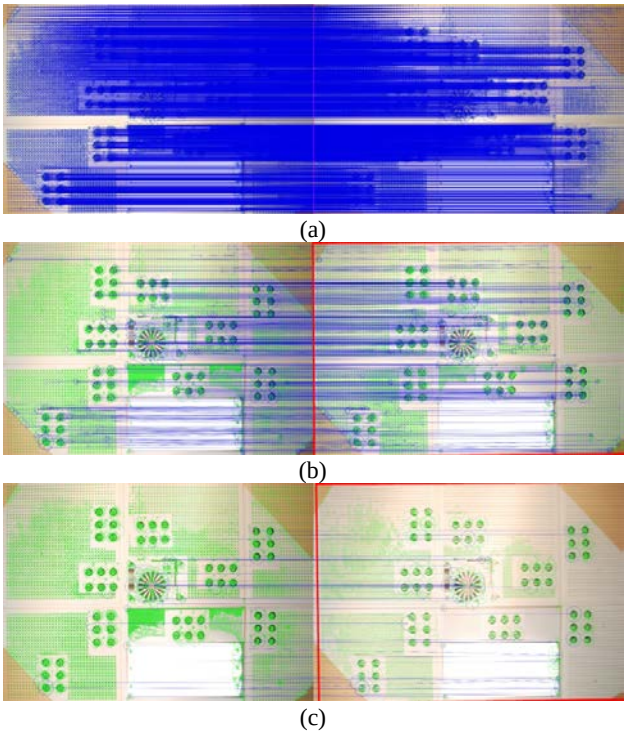


(a)

(b)

(c)

Figure 2. Influence of image blur on automatic feature detection and connection using SURF and brute force matching. Left image is a sharp reference image. Blue lines and circles indicate feature points (a) Right image without camera displacement. (b) Right image with 0.377 mm camera displacement. (c) Right image with 1.028 mm camera displacement.



(a)

(b)

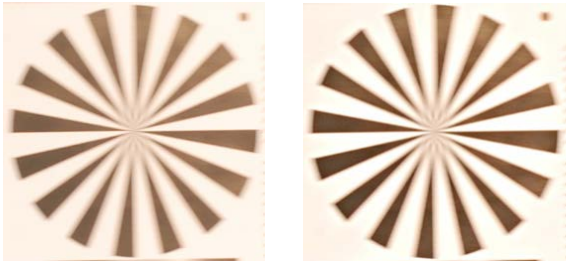
(c)

Figure 3. Results of automatic feature detection and connection using SURF and Brute Force matching after filtering out incorrect matches using sensor position. Left image is a sharp reference image. Blue lines and circles indicate connected feature points, green indicate detected feature points, red indicates image overlap. (a) Right image without camera displacement. (b) Right image with 0.377 mm camera displacement. (c) Right image with 1.028 mm camera displacement.

estimated at which position the matched feature should be in the blurred image. If the matched feature is found at a different position, this match can be assumed to be incorrect and ignored. If the feature was matched at the right position, the orientation of the feature has to match with the orientation given by the IMU. However, it is important to recognise that GNSS and IMU can only provide an approximate value for the translation and rotation of the images. This problem can be tackled by defining an appropriate threshold for the discrepancy between calculated and actual position, rotation and scale of the matched feature.
Figure 3 demonstrates the number of acceptably matched feature points, which is significantly less than the total matches made. These matches can be considered correct as the connection lines between sharp and blurred image are parallel. Even with this small number of matches it is still possible to calculate transformation parameters to detect the relative translation and rotation of sharp and blurred image. These transformation parameters are necessary to define the image relationship required to carry out standard photogrammetric measurements. If it is not possible to match enough features between the sharp and blurred image, feature detection can perhaps be improved by enhancing the blurred image.

3.1.1 Improving Feature Detection

Enhancing the blurred image can result in improved results for image processing and this does not require additional information. A fast, easy and reliable method to achieve this was investigated using an “Unsharp Mask Filter”. This approach blurs the input image and then subtracts the input



(a) (b)

Figure 4. Figure (a) shows the blurred image before enhancement. (b) represents image (a) after enhancement. The image appears with stronger contrast but not sharp.

image from the result (GIMP, 2014). An enhanced image can be generated, which appears sharper than the original input image and with stronger contrast (Fig 4). Figure 4 represents a Siemens star to visualise the blur. Figure 4 (a) demonstrates that before enhancement it is easily possible to see the blur, especially in the centre of the star. Figure 4 (b) demonstrates that after enhancement the image appears sharper, but at the centre of the star it is possible to see that blur remains. The apparent higher contrast gives the impression of sharpness. However, with higher contrast it is possible that feature detection algorithms find feature points easier.

3.2 Image Deblurring

The effectiveness of initial feature detection, controls which image deblurring technique is best used. If the overlap calculation is successful, the information provided by the overlapping image can be used for deblurring. If the overlap is not successfully calculated, deblurring requires a different approach.

3.2.1 Fourier Domain Approach

A Fourier transformation can be adopted if points are successfully matched. As all images are taken from a moving platform it can be assumed that all images contain a certain amount of blur due to forward motion. However, some contain significantly more blur due to gusts, turbulences or operator inputs. If there is an overlap between a sharp and blurred image, then the overlap area in the blurred image can be deblurred.
In the first step the overlapping areas need to be transformed so that they have the same rotation, translation scale and shear. The parameters for this can be based on the results of SURF matching or manual feature detection and matching. After this the deblurring process can commence.
The frequency domain of the overlap is calculated for both images using a discrete Fourier transformation (DFT). The frequency domain of the more blurred image contains less high frequency elements than the sharper image. As both images show the same area, deblurring can be conducted by integrating the high frequencies of the sharper image into the frequency domain of the blurred image.
Transforming the frequency domain representation of the blurred image, (now enhanced with the high frequencies of the sharper image,) back to the spatial domain, then generates a less blurry image. As DFT is only possible for grey scale images it is necessary to either correct all channels separately or calculate a single grey scale representation. If a one channel grey scale image is used a final step transferring this back to a three channel colour image is required. The approach used is based

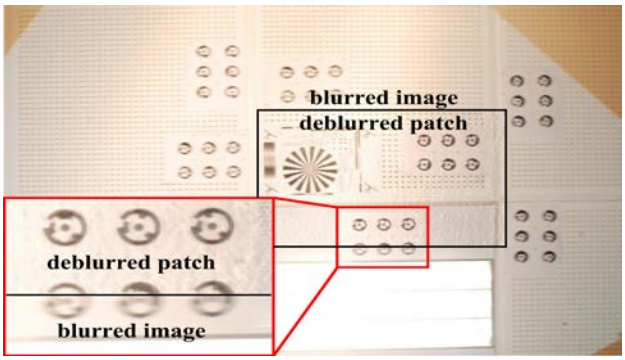


Figure 5. A deblurred patch is presented in the centre of the image. It is possible to see the photogrammetric targets are sharper and better to read and the boreholes in the ceiling panels are good to see. The inset shows a comparison between the deblurred upper part and the bottom part of a target which contains a camera displacement of 1mm and visualizes how effective the deblurring is.

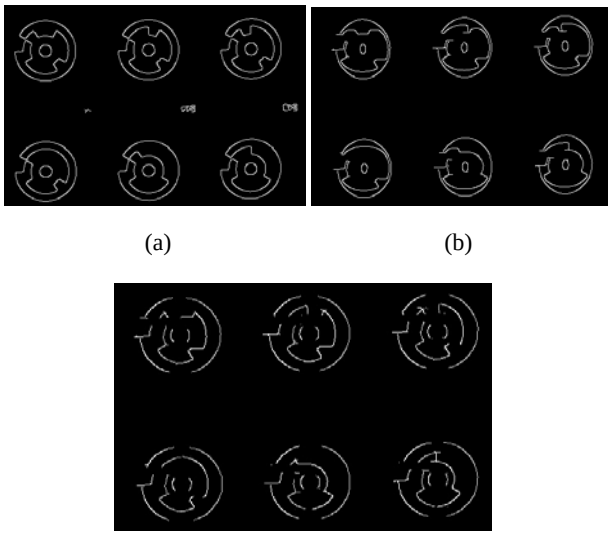
on the idea of pan sharpening by replacing the intensity channel of an image with the deblurred image (Laben, 2000).
After the deblurring process, the overlapping area needs to be transformed using an appropriate rotation, translation, scale and shear. As this deblurring is only possible for the overlapping area and not the complete image, the deblurred area of the image needs to replace the overlapping area in the blurred image. The result is a partially deblurred image (Fig 5).

3.2.2 Edge Correction Approach

Another approach developed in this study focused on correcting edges in images, which represent an important visual component. Edges indicate if an image is blurred or sharp (Chen, 2011) and help to identify how much an image is blurred. Also point spread functions and blur kernels can be calculated, based on edges or are part of additional information provided in other ways.
Knowing this, the image can be reduced to its edges by using a standard edge detection algorithm like a Laplace filter or similar. In a blurred image, the edges are displaced from their 'sharp' position due to motion of the camera during exposure (Fig 6 (b)). Also the intensity of the detected edge in a blurred image is much less than in a sharp image. This shift can be reversed by knowing the blur kernel so that the edge can be shifted back to its original position (Fig 6 (c)). How large the blur kernel is can be determined using overlapping images, IMU data or outputs of point spread calculation methods. Figure 6 (a) presents edges detected in a sharp image and Figure 6 (b) how the edges appear in a blurred image. The blurred edges are not round but oval. Figure 6 (c) is a re-sharpened image using the edge shift approach. It is possible to identify that the targets in the re-sharpened image are now circular in shape. Additionally, it is possible for the human brain to interpret elements of the original code. The errors which appear in this edge corrected image will be discussed further in the results section (4.2.2).

4. RESULTS AND DISCUSSION

The accuracy of the non-blind deconvolution methods used here depends upon which algorithm is used and the quality of the additional information. This additional information is dependent upon the feature detection and matching which is itself influenced by blur. It is important to reassess both, the feature detection and deblurring approaches used.



(a) (b)

(c)

Figure 6. Figure (a) represents edges of circular targets in a sharp image. (b) shows the same edges in a blurred image. They do not appear as circles any longer. (c) is a preliminary result of the deblurring process based on the blurred edges (b) and knowledge about the blur path.

4.1 Feature Detection

As expected, it was found that with increasing blur the total number of detected feature points decreases (Table 1). For example, a camera displacement of 0.377 mm during exposure, results in just 72% of the original feature points being detected. Human hand jitter has frequencies of 2-10 Hz with an amplitude of up to 1 mm (Stiles, 1976) which is a likely cause for much motion and has to be considered as a significant influence.
The results after filtering out the incorrect matches using IMU and GNSS information, show that blur has an extreme influence on matching methods. Only a fraction of features were correctly connected between sharp and blurred image. In the case of a blurred image with a 1 mm displaced camera, only 47 matches were accepted (Table 1).

Camera displacement [mm]	Detected Feature Points	Accepted Feature Points after Filtering
0.0	12214 (100%)	12214 (100%)
0.377	8847 (72%)	1524 (17%)
0.529	7370 (60%)	224 (3%)
1.028	2645 (22%)	47 (1.8%)

Table 1. Impact of increasing image blur and feature detection. With increasing image blur the number of detected and accepted feature points reduces rapidly.

The feature detection shows that blur influences image processing, which confirms findings made by Sieberth et al. (unpublished). The more an image is blurred the fewer features are detected using SURF and the number of matched features becomes increasingly incorrect. However, the test images were taken with a short camera to object distance which questions the applicability for larger camera to object distances, which are around 100 m for typical UAV image flights. Therefore the camera displacement for a typical UAV image flight should be calculated. Normal UAV flight speed is supposed to be 54 km/h and an image exposure time of 1/400 s (used by Grenzdörffer et. al (2012)), implies that the camera should experience a displacement of 37 mm during exposure. This is 70 times more

than in the laboratory test with a displacement of 0.53 mm. The camera to object distances in the lab test was 1.8 m, 55 times shorter than normal UAV flight altitude of 100 m. The ratios show that the lab tests are actually comparable with a typical UAV image flight because the camera displacement is as many times larger as the camera to object distance. Furthermore, this calculation does not consider that angular movements of UAVs which are much faster and cause more extensive image blur (Grenzdörffer et al., 2012). The test images also provide a texture with high contrast enabling SURF to find many feature points. SURF is the recommended method for UAV images (Gülch, 2012) but will experience problems with blurry images of low contrast areas (e.g. grass). Additionally, image overlap in the test image is nearly 100% but will be around only 60% in real scenes. This reduced overlap will reduce the feature points which are available in both images and reduces the chance for correct calculation of transformation parameters. In addition, the laboratory images do contain only linear movement without angular rotations. This simplifies the feature matching which will be significantly more difficult with more challenging camera displacements and rotations.

Enhancing the blurred image certainly improves the image. Instead of only 2645 features, twice as many features can be found and matched in an image with as much as 1 mm camera displacement. Out of these 5844 matched features 91 matches were accepted as correct, twice as many as before. This shows, that fast and easy to compute image enhancements can improve significantly the image quality necessary for photogrammetry.

4.2 Deblurring

Even if blur is insufficient to prevent successful image matching, it may be too severe for accurate detection and identification of photogrammetric targets and ground control points. A method to improve the image quality through "deblurring" is therefore valuable. Two approaches have been investigated.

4.2.1 Fourier Approach

The Fourier approach (Section 3.2.1) inserts high frequencies derived from a sharp image into the blurred and appeared successful (Fig 5). Figure 5 shows that photogrammetric targets in the deblurred patch can be identified easily and in comparison to targets in the remaining blurred part of the image, they are now sharp and have high contrast. A deblurred image for feature detection and matching is presented in Figure 7. The blurred image had a camera displacement of 0.53 mm. After deblurring, SURF returns 11725 feature points. After matching and filtering incorrect features, 1124 were accepted as correct. This is five times more than with the original blurred image.

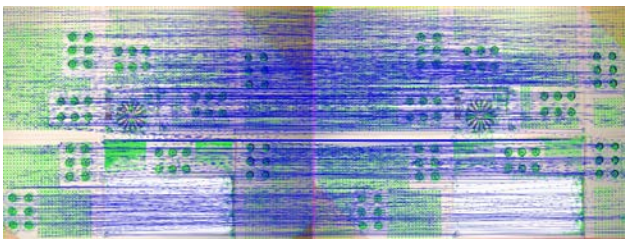


Figure 7. SURF feature detection applied on a deblurred image. On the right a sharp image. On the left a deblurred image which had a camera displacement of 0.53 mm.

However, the approach using Fourier transformation is only applicable for flat areas, which do not exhibit significant height differences. In cases of significant height variations, like high rise buildings, tall trees or opencast pits, offsets will be generated due to relief displacement (Campbell and Wynne, 2012). A way to solve this is by using many small image patches, which take account for different heights.

Adding frequencies to a blurred image also causes some noise after transforming from the frequency back to the spatial domain. In the subsequent step of image transformation necessary to regain the correct position, rotation, shear and scale, the interpolation of the correct pixel intensities for the rectified image can cause a 'blurring' effect. Furthermore, deblurring using overlapping images acquired from a moving platform will only be as good as the 'sharper' image. If the overlapping image is completely sharp then the deblurred images will be sharp. If the overlapping image is slightly blurry the deblurred image will be blurry too because the critical high frequencies are missing and cannot be integrated to the blurred image. A complete deblurring can only be generated if there is a perfectly sharp image containing all high frequencies.

From a photogrammetric point of view this approach achieves images suitable for further processing. However, it cannot be guaranteed that subsequent measurements in the deblurred image are of high or even sub-pixel accuracy. Due to the geometric image transformation, errors can be introduced which depend on the accuracy of the transformation parameters. This shows the demand for a geometric correct approach, which guarantees correct photogrammetric measurements. This could be possible by using the edge shift approach.

4.2.2 Edge Shifting Approach

The edge shifting approach is a novel method developed here and shows promising results. The main problem is the disregard of colour information, which needs to be restored afterwards. This should be acceptable as the colour just needs to refill the shapes generated by the shifted edges. A more significant problem are the many special cases and exceptions, which can occur. Handling these exceptions is connected with a range of difficulties. It is possible that the shifting process generate holes in the shapes (Figure 6 (c)). This is caused by shifting two edges apart from one another and effectively 'ripping' them apart. Furthermore, edges which have completely disappeared cannot be recovered. The complexity of real images with different edge intensities, directions and blur length is much higher than in this high contrast test image.

However, deblurring of high contrast areas can also be usable for aerial images because aerial photogrammetric targets normally provide high contrast. It should be possible to achieve edge shifting with photogrammetric targets and get acceptable results which can be then accurately measured. Subsequent target measurements can return sub-pixel accuracies due to a sub-pixel precise shifting process. Furthermore, it might be possible to incorporate different blur in different positions of an image by calculating localised blur kernels. This would be necessary for blurred objects close to the camera, which appear more blurred compared to objects further away from the camera. Theoretically this approach is achievable and does not consume too much computational power. However, this approach is very complex due to the complexity and exceptions for shifting edges. It will be difficult to implement such an approach.

5. CONCLUSION

This study has shown that image blur caused by camera displacement during image exposure has a clear and significant influence on photogrammetric processing. Successful feature detection and matching becomes increasingly difficult with increasing image blur. Enhancing a blurred image might improve feature detection at low computational cost and is easy to implement. Reversing image blur in a photogrammetrically correct and precise way is more difficult and cannot be solved with conventional deblurring methods. The described approach here using Fourier transformation returns good deblurred result for flat areas. However, the dependency on sharp images for deblurring makes this approach difficult to apply. Additionally, the unavailability of 3D object coordinates required to correct blur explicitly for varying camera to object distances, limits this approach. The problem of geometrically correct deblurring might be solvable using the edge shift approach. However, this approach will need further development.

REFERENCES

Agrawal, A., Xu, Y., Raskar, R., 2012. Multi-image deblurring. US Patent 8,229,244.

Bay, H., Tuytelaars, T., Gool, L. Van, 2006. *Surf: Speeded up robust features*, in: Computer Vision - ECCV 2006. Springer Berlin Heidelberg, pp. 404-417.

Brown, L.G., 1992. A survey of image registration techniques. *ACM Computing Surveys*, 24(4), pp. 325-376.

Campbell, J.B., Wynne, R.H., 2012. *Introduction to Remote Sensing*, Fifth ed. Guilford Press, New York.

Chen, C., Chen, W., Bloom, J. a., 2011. *A universal reference-free blurriness measure*. Society of Photo Optical Instrumentation Engineers. 7867(78670B-2), pp. 1–14.

Eisenbeiß, H., 2009. *UAV photogrammetry*. ETH Zürich, Zürich.

Eisenbeiß, H., 2011. Tutorial UAV-g Introduction. *UAV-G*. 2011.

GIMP, 2014. Unsharp Mask. http://docs.gimp.org/en/plug-in-unsharp-mask.html (20 Jun. 2014).

Grenzdörffer, G., Niemeyer, F., Schmidt, F., 2012. Development of Four Vision Camera System For A Micro-UAV. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX-B1, pp. 369-374.

Gülch, E., 2012. Photogrammetric Measurements in Fixed Wing UAV Imagery. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX-B1, pp. 381-386.

Joshi, N., Szeliski, R., Kriegman, D.J., 2008. PSF estimation using sharp edge prediction. *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1 – 8.

Kraus, K., 2004. *Photogrammetrie*, 7th ed. Walter de Gruyter, Berlin.

Kraus, K., 2007. *Photogrammetry - Geometry from Images and Laser Scans*, Second ed. Walter de Gruyter, Berlin.

Laben, C.A., Brower, B.V., 2000. Process for Enahncing the Spatial Resolution of Multispectral Imagery Using Pan-Sharpening. US Patent 6,011,875.

Lelégard, L., Delaygue, E., Brédif, M., Vallet, B., 2012. Detecting and correcting motion blur from images shot with channel-dependent exposure time. *International Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3, pp. 341-346.

Lowe, D.G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), pp. 91-110.

Lucy, L.B., 1974. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79, pp. 745-754.

Luhmann, T., Robson, S., Kyle, S., Harley, I., 2006. Close Range Photogrammetry. Whittles Publishing.

OpenCV Developer Team, 2013. OpenCV API Reference - OpenCV 2.4.6.0 http://docs.opencv.org/modules/refman.html (20 Jun. 2014)

Raskar, R., Agrawal, A., Tumblin, J., 2006. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Transactions on Graphics*, 25(3) , pp. 795-804.

Richardson, W.H., 1972. Bayesian-Based Iterative Method of Image Restoration. *Journal of the Optical Society of America*, 62(1), pp. 55-59.

Shan, Q., Jia, J., Agarwala, A., 2008. High-quality motion deblurring from a single image. *ACM Transactions on Graphics*, 27(1), pp. 1-10.

Sieberth, T., Wackrow, R., Chandler, J.H., 2013. Automatic Isolation of Blurred Images from UAV Image Sequences. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1/W2, pp. 361-366.

Sieberth, T., Wackrow, R., Chandler, J.H., unpublished. Blur Disturbs – Influence of Blurred Images in Photogrammetry. *Submitted to The Photogrammetric Record*.

Stiles, R.N., 1976. Frequency and displacement amplitude relations for normal hand tremor. *Journal of Applied Physiology*, 40(1), pp. 44 - 54.

Tai, Y.-W., Du, H., Brown, M.S., Lin, S., 2008. Image/video deblurring using a hybrid camera, *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1-8.

Wiener, N., 1949. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Wiley, New York.

ACKNOWLEDGEMENTS

I want to thank the RSPSoc for sponsoring my attendance at this conference.

AUTOMATIC ISOLATION OF BLURRED IMAGES FROM UAV IMAGE SEQUENCES

T. Sieberth [a, *], R. Wackrow [a], J. H. Chandler [a]

[a] Loughborough University, School of Civil and Building Engineering, Loughborough LE11 3TU, UK - (T.Sieberth, R.Wackrow, J.H.Chandler)@lboro.ac.uk

KEY WORDS: Automation, Blur, Correction, Development, Digital, Image processing, Photogrammetry, UAV

ABSTRACT:

Unmanned aerial vehicles (UAV) have become an interesting and active research topic for photogrammetry. Current research is based on images acquired by an UAV, which have a high ground resolution and good spectral and radiometrical resolution, due to the low flight altitudes combined with a high resolution camera. UAV image flights are also cost effective and have become attractive for many applications including change detection in small scale areas.

One of the main problems preventing full automation of data processing of UAV imagery is the degradation effect of blur caused by camera movement during image acquisition. This can be caused by the normal flight movement of the UAV as well as strong winds, turbulence or sudden operator inputs. This blur disturbs the visual analysis and interpretation of the data, causes errors and can degrade the accuracy in automatic photogrammetric processing algorithms. The detection and removal of these images is currently achieved manually, which is both time consuming and prone to error, particularly for large image-sets. To increase the quality of data processing an automated filtering process is necessary, which must be both reliable and quick.

This paper describes the development of an automatic filtering process, which is based upon the quantification of blur in an image. A "shaking table" was used to create images with known blur during a series of laboratory tests. This platform can be moved in one direction by a mathematical function controlled by a defined frequency and amplitude. The shaking table was used to displace a Nikon D80 digital SLR camera with a user defined frequency and amplitude. The actual camera displacement was measured accurately and exposures were synchronized, which provided the opportunity to acquire images with a known blur effect. Acquired images were processed digitally to determine a quantifiable measure of image blur, which has been created by the actual shaking table function. Once determined for a sequence of images, a user defined threshold can be used to differentiate between "blurred" and "acceptable" images.

A subsequent step is to establish the effect that blurred images have upon the accuracy of subsequent measurements. Both of these aspects will be discussed in this paper and future work identified.

1. INTRODUCTION

Research involving UAV's is a current research topic (Aerometrex, 2012). The evolution from manned to unmanned aerial vehicles has progressed rapidly using advancing technology, faster and more powerful computers, increasing knowledge about aerodynamics and the development of newer, lighter and more robust materials. The removal of an on-board human makes a UAV both time and cost efficient. Pilot training, registration and airport fees are not required and the usage of SLR cameras instead of aerial imaging systems reducing costs significantly. UAV's are of particular interest for military use as human life is not endangered (Shane and Shanker, 2011).

1.1 Motivation

One of the main problems influencing image quality, beside the flight altitude and the camera model, is the blur of the image. Blur caused by the forward movement of the UAV can be computed using the following equations:

$$\frac{L}{H} = \frac{l}{c} \qquad L = \frac{H \cdot l}{c} \qquad m = v \cdot ex \qquad b = \frac{m}{L} \tag{1}$$

where c = focal length
H = flight altitude
ex = exposure time
l = pixel size

* Corresponding author.

L = ground sample distance
v = vehicle velocity
m = forward movement during exposure
b = blur in pixel

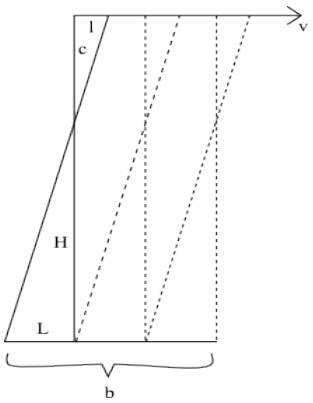


Figure 1. Sketch of image blur due to UAV movement. Not included are angular rotations which introduce a bigger blurring effect.

Figure 2. Motion blur example in high resolution UAV image.

Image blur introduced by the forward motion of the UAV can degrade the quality of data, even if a very short exposure time and high flight altitude is used (Fig1 and 2) for example a flight altitude of 100m, a flight speed of 11m/s and an exposure of 1/100s generates blur equal to 4 pixel.

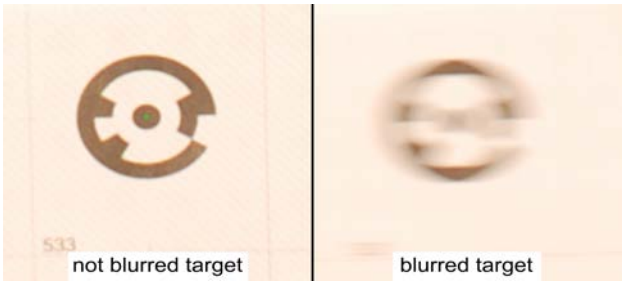


Figure 3. Target cannot be detected automatically if it is significantly blurred.

However, turns and turbulences cause a more significant blur effect on image quality. These blurred images cannot be processed using automatic image processing software because the software can fail to identify control targets or tie points (Fig 3). Manual processing therefor becomes necessary, which is time consuming and prone to error.

If an automated procedure could be developed to filter out blurred images, data processing efficacy would be improved.

1.2 Aims and objectives

The main aim of this study is to filter out blurred images from datasets automatically prior to image processing using a developed software algorithm. As an initial step, images with a defined blur have been captured, which have been used to develop and test a new algorithm. This algorithm is capable of quantifying image blur, which is used to compute a threshold value. This threshold value can be used for detecting and excluding blurred images from flight datasets. It is recognised that the on-board autopilot of a UAV can be used to improve flight quality and additional data derived from Global Navigation Satellite System (GNSS) and Inertial Navigation System (INS) could be used in identifying blurred images. However, the strength of the approach developed here is that only images are required.

1.3 Current research

This paper describes the current development of an algorithm to detect motion blur in images. Existing blur detection algorithms often require additional information associated with both the image and image acquisition. Many of these are using information of INSs (Shah, 2012), whilst others include a video camera (Ben-Ezra and Nayar, 2004, Tai et. al., 2008, Agrawal et. al., 2012). An INS which is included in the UAV autopilot does normally not have a measuring frequency that is high enough to calculate a blur kernel. An additional video camera is often too heavy and requires additional modification and calibration for the calculation of a blur kernel. A blur kernel represents precisely the three dimensional movement of a camera during image acquisition. It can be used to separate images with a small blur, from images with a larger movement (big blur). Also other hardware modifications like “Forward Motion Compensation” (Pacey and Fricker, 2005) or image stabilisation are difficult to include in a light weight UAV or a low cost camera. In the case of a “Forward Motion Compensation” the whole camera, the film or the sensor is moved in the direction of travel during image acquisition to compensate blur effects. Image stabilisation uses a mathematical model to compensate the motion blur effect. Both methods are only efficient for small blurs but cannot compensate significant image blur caused by UAV turns or turbulences. Other detection algorithms modify the exposure time to detect blur effects, which is difficult to realise for low cost high resolution cameras. A modification can be done for each of the RGB channels, which results in a different amount of blur for each channel (Lelégard et. al., 2010). Another approach is the usage of an unblurred reference image (Paramanand and Rajagopalan, 2012, Charlmond, 1991), which cannot be used for moving UAVs. Other algorithms which are not based on additional information or hardware modifications work often with edge detection (Ong et. al., 2003, Joshi et. al., 2008, Narvekan and Karam, 2009) and frequency analyses (Clark Jones, 1958, Liu et. al., 2008). Blurred edges are not well defined in an image, which can also be seen in the frequency space by a decreased number of high frequencies. The number of high frequencies is reduced and the number of low frequencies is increased. These algorithms apply higher control using an existing dataset of known camera movement and the resulting blur, which is difficult to achieve for UAV imagery.

2. METHODOLOGY

The main problem of existing algorithms is that the blur in the images cannot be quantified, which prevents defining a threshold value to automatically exclude blurred images from datasets. An operator is required to manually identify blurred images, which is subjective. The subsequent sections describe the development of a new automated method to quantify image blur.

2.1 Image blurring using a shaking-table

To acquire images with a defined blur the movement of the camera and the point of time of the image exposure have to be known, which can be achieved using a shaking table. The movement of this table can be controlled with high precision. Normally it is used by construction engineers to test the strength of building materials and their resistance against earthquakes or other vibrating influences. Usually the table moves with a very high frequency but a low amplitude, which represent acceleration and velocity. The displacement can be measured accurately using a laser sensor. Additionally, the acceleration of the table and consequently objects on the table can be determined using acceleration sensors (G-Sensor).
By fixing the camera on the table and measuring both the displacement and the acceleration of the platform, images with a known blur can be generated (Fig 4). A test field comprised of fixed photogrammetric targets was established on a horizontal ceiling located approximately 1.7m above the shaking table. The camera was equipped with a 24mm lens providing an image scale of 1:70. A horizontal camera displacement of a single pixel (6.1µm) produced by the shaking table equates to a blur of 0.4mm in the object space.

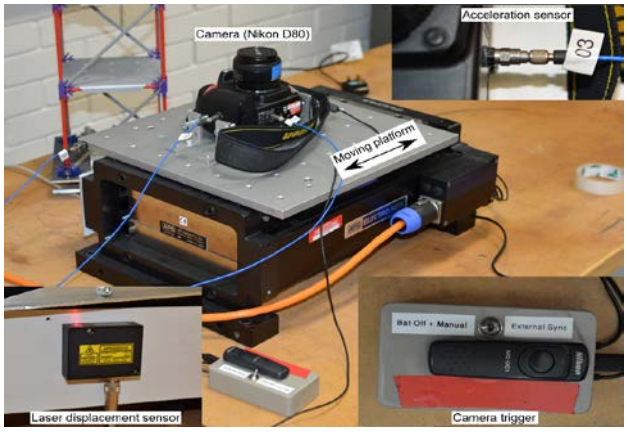


Figure 4. Shaking table setup.

Additionally, a G-Sensor is attached to the camera body in direction of the movement of the table. A second one is attached to the camera body to ensure that the camera is only moving in one direction and a third sensor is attached to the platform. These sensors are primary used to gather additional information and control of the movement of the table and the camera.
The movement of the table is based on a user defined input function describing the velocity (amplitude) of the platform. It is synchronised with the camera shutter to acquire images at a time of constant velocity (Fig 5), which generates homogenous blurred images. Additionally, measurements acquired using the displacement sensor can be compared to observations determined using the G-Sensor. It is noticeable that the G-Sensor measures a high acceleration immediately after the camera trigger signal, due to open and closing of the mirror.

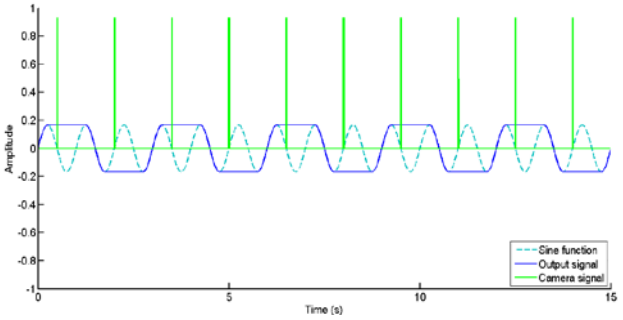


Figure 5. Shaking table signal with camera triggering signal.

No additional camera movement was registered by the G-Sensor, showing that the camera is not affected by vibrations during image exposure (Fig 6). The image exposure is between the two signals in green (positive, start of exposure, negative, stop of exposure), delayed after the real triggering signal but during a period of constant displacement.

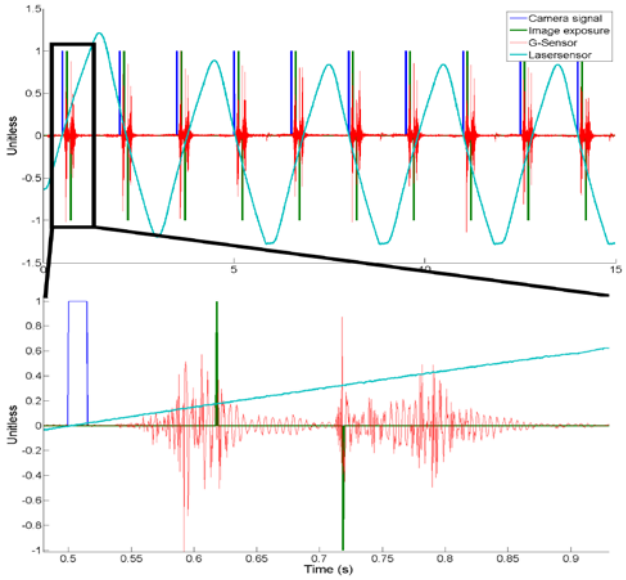


Figure 6. Measurements and camera release during shaking process.

After the shaking process, the signals can be evaluated and the movement of the camera during the exposure can be calculated. The producible blur is limited by both, the minimal movement speed (0.001m/s) and maximum displacement (0.15m) of the platform and needs to be combined with the time of image exposure. A longer exposure time requires a slower movement of the table and consequently a shorter exposure time requires a faster movement of the table platform. This provides the opportunity to generate images with both, different blur and different brightness, which is based on the exposure time. Additionally, background textures were projected onto the ceiling to vary the image content (Fig 7).
The complete dataset involved capturing nearly 2000, images consisting of 7 different exposure times and 10 different background textures. To visualise the blur a special test pattern was used. This pattern consists of lines of different width and orientation as well as a Siemens-Star (Fig 7). The advantage of the star is characterised by the possibility to increase visualization of the direction of movement in contrast to using a simple circular target.

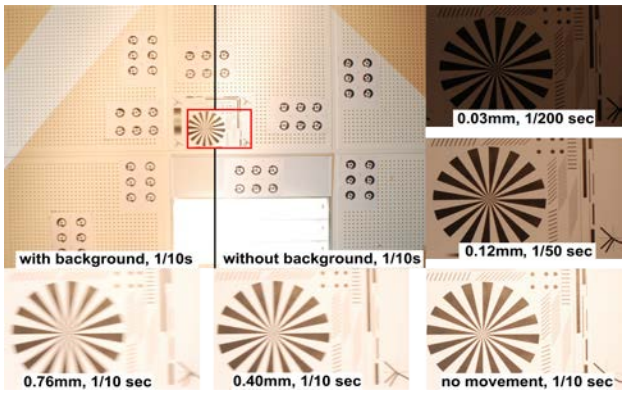


Figure 7. Comparison of produced blurred images with movement of camera and image exposure time. Pixel resolution 0.43mm.

2.2 Image blurring by image processing

Another method used for this paper to generate blurred images is the use of image processing software with image filters, for example a Gaussian filter. The aim of these filters is to "smooth" the image by blurring it. This results in an image distortion, which can appear similar to true optical blur. MATLAB also provides a specific filter, which simulates motion blur. The filter represents a vector, which is defined in both size and direction of movement (MathWorks 2013). This filter has been applied on images without any a-priori movement, which assures that no additional blur is introduced by the movement of the camera (Fig 8).

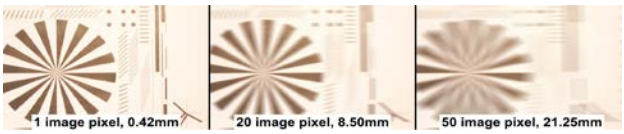


Figure 8. Comparison of produced blurred images with MATLAB filter.

2.3 Blur detection

Images with a known blur, generated using the methods described in section 2.1 and 2.2 were processed to compute a "blur value" using a newly developed mathematical algorithm. However, it is necessary to investigate and exclude unwanted systematic effects.

To ensure that the movement of the mirror does not influence the computed blur in the images, the G-Sensor measurements of the camera body were analysed. Integrating twice over the time, acceleration was converted to distance and the camera shake visualised. This generated a movement of just $0.7 \cdot 10^{-5}$µm, in comparison to a pixel size of 6µm. It would therefore be concluded that the movement caused by the mirror is insignificant and the computed blur for each image was based on the actual value measured by the laser displacement sensor (Fig 9).

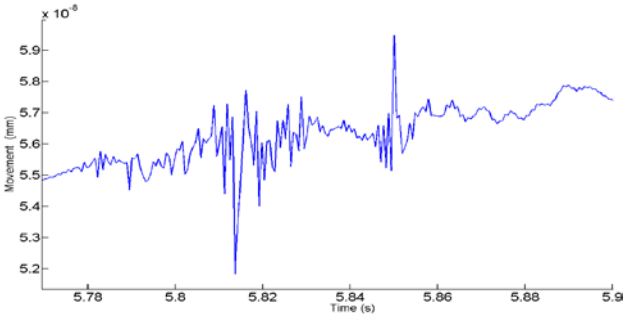


Figure 9. Camera shake due to mirror movement.

After ensuring that the mirror movement is not significantly influencing the image blur, it is possible to analyse the images and calculate a value which is related to the blur using the developed mathematical method. This blur related value is called "sieds". Figure 10 represents the sieds and demonstrates that the calculated sieds depends on the movement of the camera.

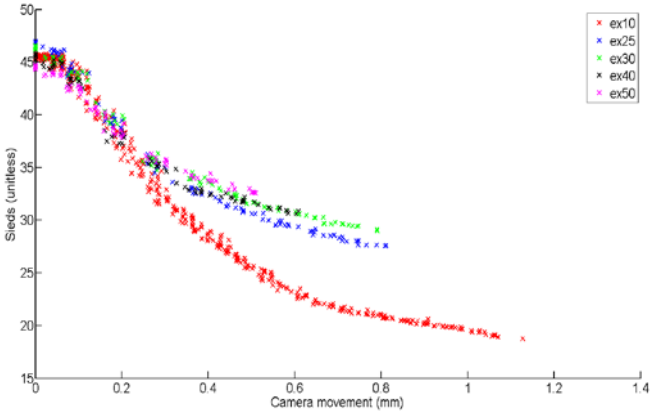


Figure 10. Camera displacement in comparison to sieds.

The distribution of the results indicates that the calculated sieds is related to image blur via a modified damping function (Eq 2).

$$b = e^{-\delta w} \cdot \left(\frac{\delta x_0}{\omega} \cdot \sin(\omega w) + x_0 \cos(\omega w) \right) + n \qquad (2)$$

where b = blur
δ = damping ratio
x_0 = start position
ω = angular frequency
n = y-offset from zero
w = sieds, derived from image

To set the unknown parameters for the damping function, which are damping ratio (δ), start position (x_0), angular frequency (ω) and offset from the y-axis (n), it is required to include other information provided in the image, which are independent of the amount of blur. A blur independent value is the average grey-value of an image.

The calculation of the sieds is also possible for the images blurred by image processing. But this graph represents a logarithmic function and not a damping function (Fig 11).

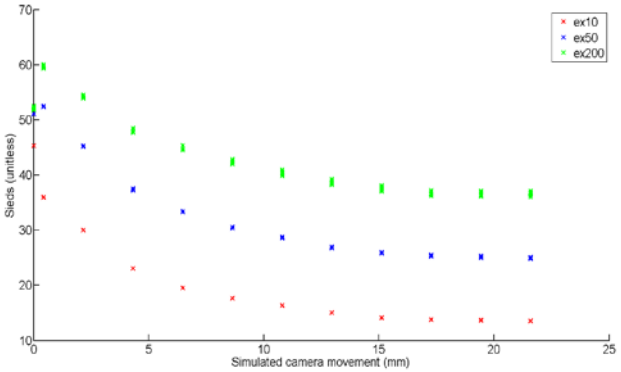


Figure 11. Evaluation of simulated motion blur with MATLAB.

The intercept on the y-axis is much higher, except for 1/10s exposure, than in the images blurred with the shaking table. An explanation can be found through closer examination of the blurred results. It is possible to see clear differences in the centre of the images. In the real blurred image (Fig 12a) the centre is the darkest part of the smeared dot. In the MATLAB blurred image it is (Fig 12c) not the centre of the dot which is darkest, but there are two shadows which appear around the centre. It is known that the shaking table image (12a) was acquired during a constant velocity, whilst the still image (Fig 12b) was captured without any movement. By comparing this to the MATLAB processed image (12c), it appears that the image was not blurred with a constant velocity but with acceleration.

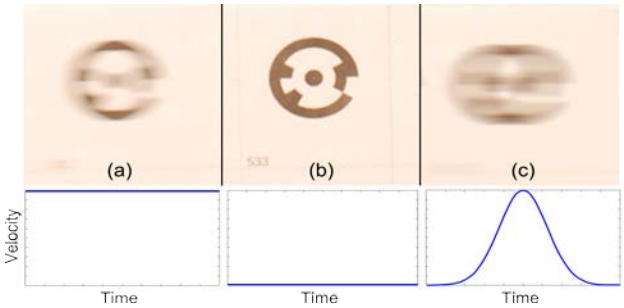


Figure 12. Difference between real blur with constant velocity (a), still image (b) and simulated MATLAB blur with acceleration (c).

2.4 Discussion

By calculating the blur independent parameters and including these in the modified damping function together with the computed blur related sieds, it is possible to calculate the blur for an image. A first test of this process was conducted using a mixed dataset of images with a known blur, which demonstrates that it is possible to calculate blur for images.
However, these results demonstrated this is not as accurate for short exposures as for images with a longer exposure time (Fig 13).

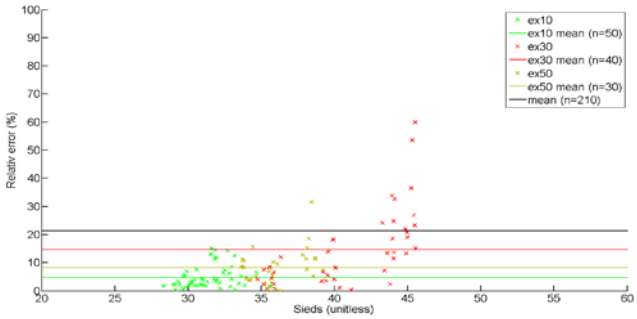


Figure 13. Relative error of the calculated blur.

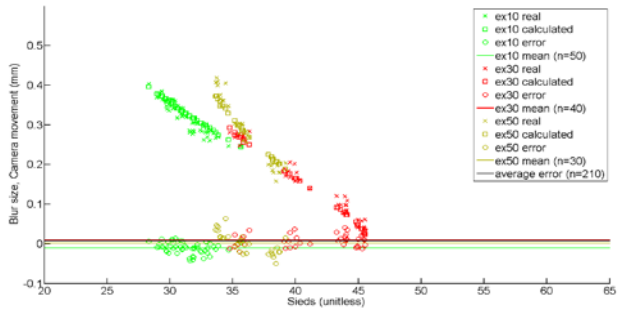


Figure 14. Absolute error of the calculated blur.

This is perhaps expected as images with a short exposure do only contain small blurs. Closer examinations of Figure 10 reveals why it is more difficult to calculate small blur effects precisely. In the area of small blur the gradient of the damping function appears very flat. Small variations in the computed sieds have a large impact on the final estimated blur.
A closer examination of the absolute values reveals that the images with an exposure of 1/200s are set to zero but all the other calculated blur values are close to the real blur (Fig 14).
In the 1/10s exposure, the calculated blur fits closely to the real blur of the images. Although the mean error of 7.65µm (compared with 6.1µm pixel size) shows that the calculation based on the images is very precise.
The algorithm was also applied on a real UAV image dataset. This demonstrates that the worst images are blurred to an equivalent of 95px (Fig 15).

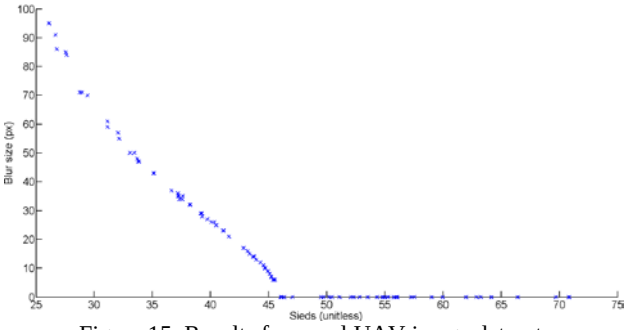


Figure 15. Results for a real UAV image dataset.

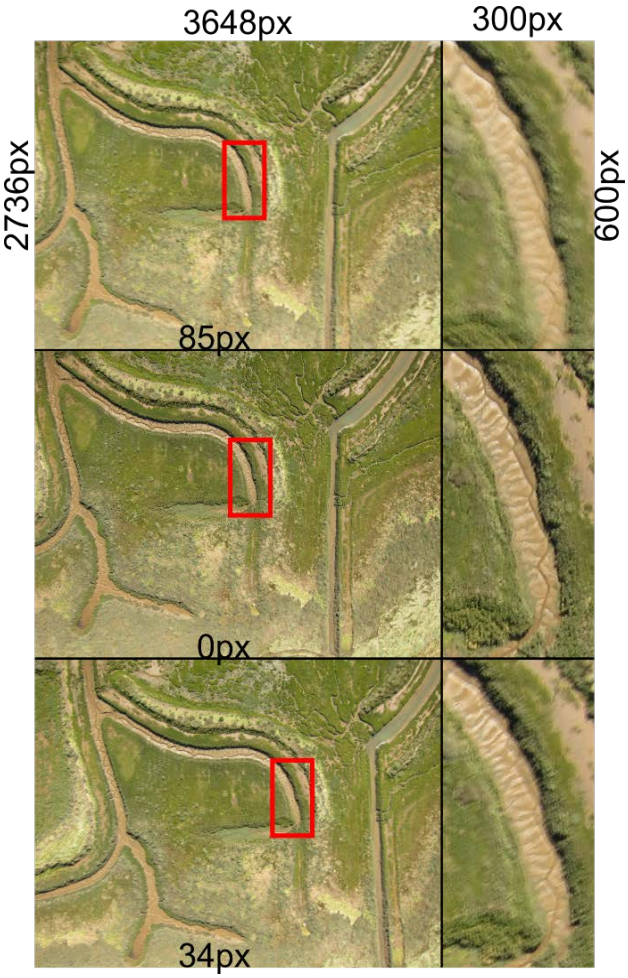


Figure 16. Blur detection result for image sequence.

The graph shows that it is possible to differentiate images with a different amount of blur. The examples from Figure 2 are classified with 34px and 0px blur. To give an impression which kind of blur can be differentiated a second example is shown in Figure 16.
This demonstrates how effective the blur detection algorithm can be to detect blur in images. Even images with a very homogenous background, where a human cannot identify blur without a reference image, the algorithm is capable to detect blur. The calculation speed for nearly 200 images, with 3600px to 2700px is just three minutes using a laptop with 4 (2.5GHz) CPUs and 4GB RAM. This is clearly acceptable.

3. CONCLUSION

This paper has outlined the development of an algorithm to detect blurred images in UAV image datasets. This makes it possible to exclude these blurred images or use blur correction algorithms to improve further data processing.
The algorithm does not contain any GNSS or INS data and does not include any information about the neighbouring images. This represents a very benefit of the developed algorithm and allows application beyond simple UAV image filtering.
The main problems which are not solved yet are the influence on the calculation due to the image size and the correlation of the detected blur size to a real movement of the camera. The algorithm does also not contain any blur kernel calculations which are normally required to correct blur in images. The algorithm is efficient, being fast and reliable and separates blurred images in a UAV image dataset, as long as the images have the same size and the same exposure time.

3.1 Future work

For the future is it planned to improve the accuracy of the blur calculation by taking account of the image size and other influencing factors and correlating the calculated movement to real world values. Additionally information of the low cost GNSS and INS sensors will be included to see if improve results justify the increased system complexity.

4. REFERENCES

Aerometrex, 2012. Aerial Photogrammetry from UAVs, Kent Town, Australia http://aerometrex.com.au/blog/?p=521 (03 Apr. 2013).

Agrawal, A., Xu, Y., & Raskar, R., 2012. Multi-image deblurring. *US Patent 8,229,244*.

Ben-Ezra, M., Nayar, S. K., 2004. Motion-based motion deblurring. *Pattern analysis and machine intelligence*, 26(6), pp. 689-698.

Charlmond, B., 1991. PSF Estimation for Image Deblurring. *CVGIP: Graphical Models and Image Processing*, 53(4), pp. 364-372.

Clark Jones, R., 1958. On the Point and Line Spread Functions of Photographic Images. *Journal of the Optical Society of America*, 48(12), pp. 934-937.

Joshi, N., Szeliski, R., Kriegman, D. J., 2008. PSF estimation using sharp edge prediction. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8.

Lelégard, L., Brédif, M., Vallet, B., & Boldo, D, 2010. Motion blur detection in aerial images shot with channel-dependent exposure time. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVIII (3A), pp. 180-185.

Liu, R., Li, Z., Jia, J., 2008. Image partial blur detection and classification. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8.

MathWorks, 2013. fspecial, Create predefined 2-D filter http://www.mathworks.co.uk/help/images/ref/fspecial.html?searchHighlight=fspecial (03 Apr. 2013).

Narvekar, N. D., Karam, L. J., 2009. A No-Reference Perceptual Image Sharpness Metric Based On A Cumulative Probability Of Blur Detection. *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop*, pp. 87-91.

Ong, E., Lin, W., Lu, Z., Yang, X., Yao, S., Pan, F., Jiang, L., 2003. A no-reference quality metric for measuring image blur. *Signal Processing and Its Applications, 2003*. Proceedings. *Seventh International Symposium*, 1, pp. 469-472.

Pacey, R., Fricker, P., 2005. Forward motion compensation (FMC) - Is It The Same In The Digital Imaging World? *Photogrammetric Engineering and Remote Sensing*, 71(11), pp. 1241-1242.

Paramanand, C., Rajagopalan, A. N., 2012. Depth from motion and optical blur with an unscented Kalman filter. *IEEE transactions on image processing*, 21(5), pp. 2798–2811.

Raskar, R., Agrawal, A., Tumblin, J., 2006. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Transactions on Graphics*, 25(3), pp. 795-804.

Shane, S., Shanker, T., 2011. The New York Times, Strike Reflects U.S. Shift to Drones in Terror Fight, Washington, United States http://www.nytimes.com/2011/10/02/world/awlaki-strike-hows-us-shift-to-drones-in-terror-fight.html?ref=unmanneda erialvehicles&_r=0 (03 Apr. 2013).

Shah, C. A., Schickler, W., 2012. Automated Blur Detection And Removal In Airborne Imaging Systems Using IMU Data. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Melbourne, Australia, Vol. XXXIX, pp. 321–323.

Tai, Y.-W., Du, H., Brown, M. S., Lin, S., 2008. Image/video deblurring using a hybrid camera. *Conference on Computer Vision and Pattern Recognition*, pp. 1-8).

4.1 Acknowledgements

I want to thank Fiona Almond for the UAV image flight dataset. I want to thank also the RSPSoc for sponsoring my attendance at this conference.

C Patent

In context of this research the novelty of the blur detection method was analysed. Loughborough University Enterprise Office guided and supported the steps that had to be carried out. It is still in the evaluation process if a patent application is filed. The actual step is to contact companies and evaluate if there is any demand for a blur detection algorithms. If there is demand for the program the next step would be the improvement of the program and start of the patent filing process.

C.1 Coarse patent search

The patent search was conducted on national and international basis. Table C.1 shows the search carried out with ESPACENET the search engine of the European Patent Office (EPO). The database includes patents of the EPO as well as of the member state patent offices. Additionally, several patents of the World Intellectual Property Organisation (WIPO) and other non European states are in the database. The database of the United States Patent and Trademark Office (USPTO) was also searched (Tabel C.2). The results of the search were examined based on a fast read of the abstract. 567 patents were examined this way. For 183 patents was a more detailed examination required:

- RU2312395 (C1)
- WO2007057808 (A3), WO2007057808 (A2)
- US6628842
- JP2005141497 (A), JP4133746 (B2)
- US7961953
- WO2012080643 (A1)
- US8200024
- US8406510
- JP2004336752 (A), US7379091 (B2)
- JP2007184787 (A)
- US7257273
- US7711199
- US5103254
- US7295232
- CN102572466 (A)
- CN102801993 (A), US2012301012
- DE102011083745 (A1)
- JP2004336751 (A)
- JP2012195668 (A)
- JP2013066142 (A), EP2565843
- US2011069190 (A1)
- US2011081088 (A1), US8290281 (B2)
- US2012162527 (A1)
- US6344876
- US6493023
- US6798910
- US7061524
- US8139884
- US8159552
- US8260077
- US8280182

- US8284296
- US8295565
- US8433153
- WO03012725 (A1)
- WO2005114577 (A1)
- WO2010105212 (A1)
- WO2013148566 (A1)
- CA2142570 (A1)
- CN101765019 (A), CN101765019 (B)
- CN101873508 (B), CN101873508 (A)
- CN102480633 (A)
- CN102521821 (A)
- CN102595171 (A)
- CN102986221 (A)
- EP2284764 (B1), EP2284764 (A1)
- EP2642245 (A1)
- GB2485478 (A) also US8532421
- JP2005328117 (A), JP4389656 (B2)
- JP2007180912 (A)
- JP2007240733 (A)
- JP2008236645 (A)
- JP2009217076 (A), JP5178250 (B2)
- JP2010220023 (A)
- JP2011044839 (A)
- JP2011103631 (A)
- JP2011130268 (A)
- JP2013019886 (A)
- JP2013074397 (A)
- JP2013105078 (A)
- JP3143484 (B2), JPH04268873 (A), US5258783
- JP4466430 (B2), JP2006279359 (A)
- JP4697461 (B2), JP2008118289 (A)
- JP4883468 (B2), JP2007129491 (A)
- JP4915166 (B2), JP2008040705 (A)
- JPH01215547 (A)
- JPH03254286 (A)
- JPH04315379 (A), JP2819852 (B2)
- JPH04318776 (A)
- JPH0556329 (A)
- JPH0865656 (A), JP3471436 (B2)
- JPH09312861 (A)
- JPS53126952 (A)
- JPS57129421 (A)
- JPS62209686 (A)
- KR20020014986 (A)
- KR20020056631 (A), KR100658601 (B1)
- KR20070071786 (A)
- KR20120019164 (A)
- TW201023000 (A), US2010079602
- US2002003581 (A1), US7042507 (B2)
- US2002109701 (A1), US6618054 (B2)
- US2004024296 (A1)
- US2004120598 (A1), US7181082 (B2)
- US2007024627 (A1), US7483550 (B2)
- US2007070250 (A1), US7570309 (B2)
- US2009273676 (A1), US8204335 (B2)
- US2011037861 (A1), US8363115 (B2)
- US2011090345 (A1)
- US2012300025 (A1)
- US2013162629 (A1)
- US2013194486 (A1)
- US2013250060 (A1)
- US3938386
- US4523842 (A)
- US4772117
- US5311132 (A)
- US5596366
- US5600574, MX9604557
- US5623705
- US5634145
- US5712474
- US5745801
- US5748994
- US5758202
- US5758203
- US5771403
- US5831671
- US5881324
- US5943169
- US5973319
- US5978598
- US6067419
- US6122447
- US6163651
- US6219468
- US6263161
- US6285799 (B1)
- US6366735
- US6393215
- US6408135

- US6426755 (B1)
- US6522837
- US6734903
- US6754445
- US6778210
- US6922524
- US6985126 (B2), US2003146885 (A1)
- US7066597
- US7110024 (B1)
- US7260270
- US7283676 (B2), US2003118233 (A1)
- US7317445 (B2), US2005168492 (A1)
- US7403710
- US7469099
- US7519231 (B2), US2008013861 (A1)
- US7542088
- US7596307
- US7606477
- US7636107
- US7769219
- US7773825
- US7796872
- US7826731
- US7884854 (B2), US2009015719 (A1)
- US8014583 (B2), US2010208961 (A1)
- US8027582, JP2011128623
- US8049782
- US8063920 (B2), US2008042953 (A1)
- US8068639
- US8085305
- US8098333 (B2), US2009002559 (A1)
- US8150250
- US8184123 (B2), US2009309895 (A1)
- US8213504
- US8218888
- US8259226
- US8274570
- US8284295
- US8289438
- US8319898
- US8331713
- US8369697
- US8405708 (B2), US2010002073 (A1)
- US8411195
- US8462266
- WO0188854 (A3), WO0188854 (A2)
- WO0215786 (A1), WO0215786 (A9)
- WO0215788 (A1), WO0215788 (A9), US2004024296
- WO0229718 (A2), WO0229718 (A3)
- WO03044740 (A1)
- WO2005001541 (A1)
- WO2008131438 (A2), WO2008131438 (A3)
- WO2010104549 (A1), US8405770
- WO2013107037 (A1)
- WO9848381 (A1)
- JP2011128623 (A), US8027582
- MX9604557 (A), US5600574
- US2008309771 (A1), US8284295 (B2), US8284295
- US2011063494 (A1), US8284296 (B2), US8284296
- US7397500, JP2004336751
- US8532421, GB2485478

C.2 Detailed patent search

More detailed examinations were based on the abstract and a search for specific key words on the complete patent text. The key words represented essential steps of my method. A variety of keywords are:

- blur;
- edge;
- grad... for gradient;

- degra... for degradation, degrading;
- high... for highpass, high-pass;
- low... for lowpass, low-pass.

If this search returned significant results the patent was compared in detail against the method proposed in this research. A minor similarity was given in 16 %, major similarities in 4% of the examined patents. No patent was completely similar and significant differences were found with all.

- WO2007057808 (A3), WO2007057808 (A2)
- US6628842
- JP2005141497 (A), JP4133746 (B2)
- US7961953
- WO2012080643 (A1)
- US8200024
- US8406510

Patent number RU2312395 seams to have similarities but was only available in russian. Because of this it was not possible to examine this patent in more detail.

Table C.1: Search details for ESPACNET

#	Searchtype	Searchfield	Keyword	Hits	Used	
	Advanced	Title&Abstract	blur	10887		Not worked through
	Advanced	Title&Abstract	image blur	7459		Not worked through
	Advanced	Title&Abstract	image blur detection	956		Not worked through
2	Advanced	Title&Abstract	multi image blur detection	11	9	
3	Classification search		blur			
	Advanced	Title&Abstract	blur	0	0	
		IPC	H04N5/00 G06T5/00 G06T7/00 H04N13/00 G02B27/00 G03B5/00 G06K9/00 H01L51/00 G09G3/00			
	Advanced	Title&Abstract	blur	2	0	
		IPC	H01L51/00			
4	Advanced	Title&Abstract	blur	9	7	
		IPC	H04N5/00			
5	Advanced	Title&Abstract	blur	10	5	
		IPC	G09G3/00			
6	Advanced	Title&Abstract	blur	30	7	
		IPC	G02B27/00			
7	Advanced	Title&Abstract	blur	44	18	
		IPC	H04N13/00			
	Advanced	Title&Abstract	blur measure	70		Not worked through
8	Advanced	Title&Abstract	image blur measure	46	19	
	Advanced	Title&Abstract	blur detection	1202		Not worked through
9	Advanced	Title&Abstract	image blur detection	955	1	only page 1 based on title
10	Advanced	Title&Abstract	digital image blur detection	76	30	

#	Searchtype	Searchfield	Keyword	Hits	Used	
11	Advanced	Title&Abstract	digital image motion blur detection	8	6	
	Advanced	Title&Abstract	image quality measure	967		Not worked through
12	Advanced	Title&Abstract	digital image quality measure	76	14	
	Advanced	Title&Abstract	blur	144		Fast harvest and based on results of first 2 pages results included in advanced search
		IPC	G06T7/00			
	Advanced	Title&Abstract	blur	171		Fast harvest and based on results of first 2 pages results included in advanced search
		IPC	G06K9/00			
	Advanced	Title&Abstract	blur	474		Fast harvest and based on results of first 2 pages results included in advanced search
		IPC	G06T5/00			
	Advanced	Title&Abstract	blur	1667	Not applicable	Adjustment of optical system relative to image or object surface other than for focusing
		IPC	G03B5/00			

Table C.2: Search details for USPTO

#	Searchtype	Searchfield	Keyword	Hits	Used	
						search from 1976 as before there where no digital images
	Advanced	ABST/	blur	1108		Not worked through
	Advanced	ABST/	(image AND blur)	881		Not worked through
13	Advanced	ABST/	((image AND blur) AND detection)	104	62	
14	Advanced	ABST/	(blur AND measure)	13	4	
15	Advanced	ABST/	(blur AND detection)	119	8	
	Advanced	ABST/	(quality AND measure)	1098		Not worked through
	Advanced	ABST/	((image AND quality) AND measure)	150		Not worked through
16	Advanced	ABST/	(((digital AND image) AND quality) AND measure)	19	5	

D Program documentation

For the program code please see the attached digital versatile disc (DVD).