Aalto University
School of Engineering

Automated texturing of 3D models using georeferenced images

Master's Thesis, Department of Surveying and Planning, School of Engineering, Aalto University.

Helsinki, 9th February 2015

Bachelor of Science
Mika Kekäläinen

Supervisor: Professor Henrik Haggrén
Instructor: M.Sc. (Tech.) Lingli Zhu

Aalto University
School of Engineering

Aalto University, P.O. BOX 11000, 00076
AALTO
www.aalto.fi

Aalto University
School of Engineering
Degree Programme in Geomatics

ABSTRACT OF
MASTER'S THESIS

Author: Mika Erik Kalevo Kekäläinen

Title: Automated texturing of 3D models using georeferenced images

Date: 9th February, 2015 **Pages:** 60

Major: Photogrammetry and Remote Sensing **Code:** M3006

Supervisor: Professor Henrik Haggrén

Advisor: M.Sc. (Tech.) Lingli Zhu

The purpose of this thesis was to form automated texturing of 3D models from georeferenced airborne images made with the Pictometry technique. As an exemplary application, a 3D building model was generated. The study aimed to texture the building models based on photogrammetry knowledge, to be realized with Matlab software. The data in this thesis were bought from Blom and it consisted of images and image parameters. National Land Survey (NLS) airborne laser scanning (ALS) point clouds of the research area were also employed in this thesis.

The first step was the investigation of the Pictometry -technology adopted, followed by an examination of the oblique and nadir view images and the parameters and their units of measurements in the data files from several research articles.

The object space coordinates were estimated from the images with the inversion of the collinearity equation. The collinearity equation was applied, for example, to visualize and verify the right position of the three-dimensional (3D) coordinates in the image. The original building point cloud data were in the coordinate system of valtion vanha järjestelmä (VVJ), eventually transformed to the current ETRS-TM35FIN coordinate system with the height system N2000.

Unfortunately, the attempt to generate a fully automatic Matlab program from the original data was unsuccessful because the orthophoto of the façade was not the complete façade. The obtained orthophoto had been transformed in the vertical and horizontal direction compared to what it was expected to be. It was possible to generate a semi-automatic system from the original data but the corrections to the coordinates needed to be defined for each image separately. From the semi-automatic system, it was possible to make realistically-looking three-dimensional models of buildings, with the location of the building positioned in the real coordinates.

Another approach to generate automated texturing was to recalculate the internal orientation (IO) and the external orientation (EO) parameters with other programs. This was done because there were tens of metres of error in the coordinate measurements compared with the real coordinates, and tens of pixels of an error in the image coordinates compared with the real image coordinates. Applying these regenerated orientation parameters enabled automated texturing from the known model vertex coordinates or the coordinates measured from the image.

Keywords: Oblique imaging, texturing, 3D modelling, automation, image orientations

Language: English

Aalto-yliopisto
Insinööritieteiden korkeakoulu

Aalto University, P.O. BOX 11000, 00076
AALTO
www.aalto.fi

Aalto-yliopisto
Insinööritieteiden korkeakoulu
Geomatiikan koulutusohjelma

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Mika Erik Kalevo Kekäläinen

Työn nimi: Automaattinen teksturointi georeferoiduilta kuvilta 3D malliin

Päiväys:	9. Helmikuuta, 2015	**Sivumäärä:**	60
Pääaine:	Fotogrammetria ja kaukokartoitus	**Koodi:**	M3006
Valvoja:	Professori Henrik Haggrén		
Ohjaaja:	M.Sc. (Tech.) Lingli Zhu		

Tämän tutkimuksen ensisijainen tarkoitus oli muodostaa Pictometry -tekniikalla otetuista georeferoiduista ilmakuvista rakennuksien tekstuuri automaattisesti. Tutkimuksen tavoitteena oli tehdä mahdollisimman automaattisesti Matlab -ohjelmalla saaduista lähtötiedoista fotorealistinen julkisivun tekstuuri. Työssä käytetty data sisältää Blomilta hankittuja kuvia ja kuviin liittyviä parametritietoja. Lisäksi työssä käytetään Maanmittauslaitoksen (MML) ilmasta laserkeilattua (ALS) pistepilviaineistoa.

Tutkimus aloitettiin perehtymällä Pictometry -tekniikkaan, hankittuihin kuviin, kuvien parametreihin ja parametrien yksiköihin useiden tieteellisten artikkeleiden avulla.

Kohdekoordinaattien mittaaminen kuvilta suoritettiin käänteisellä kollineaarisuusyhtälöllä. Kollineaarisuusyhtälöä käytettiin muun muassa kohdekoordinaattien visuaaliseen tarkistamiseen kuville. Alkuperäinen rakennusten pistepilviaineisto on valtion vanhassa järjestelmässä (VVJ), ja se on muutettu ETRS-TM35FIN -koordinaatistoon ja korkeusjärjestelmään N2000.

Täysin automaattista Matlab -ohjelmaa ei pystytty rakentamaan, koska rakennuksen julkisivun irrottaminen kuvalta kohteen koordinaattien avulla ei muodostanut haluttua julkisivua, vaan irrotetulla ortokuvalla oli tavoiteltuun ortokuvaan nähden siirto sekä vaaka- että pystysuunnassa käytettäessä alkuperäistä dataa. Puoliautomaattinen järjestelmä oli mahdollista tehdä, mutta se vaati jokaiselle kuvalle jonkinlaisen siirron. Puoliautomaattisena Matlab -ohjelmana hankituilla kuvilla oli kuviin liittyvien parametrien avulla mahdollista tehdä realistisen näköinen ja todellisessa koordinaatistossa sijaitsevan rakennuksen kolmiulotteinen malli.

Täysin automaattisen tekstuurin muodostamiseksi orientointiparametrit tehtiin uudelleen käyttämällä muita ohjelmia. Syynä tähän oli se, että mittauksissa esiintyvät kohdekoordinaattien virheet todellisiin kohteen koordinaatteihin nähden olivat kymmeniä metrejä ja kuvilta mitattujen kuvapisteiden sijainti oli myöskin kymmenien pikseleiden päässä oikeasta kohteesta. Automaatio ja tekstuurin muodostaminen joko tunnettujen kulmien koordinaateista tai pisteitä mittaamalla kuvilta oli mahdollista uusia orientointiparametreja käyttämällä.

Asiasanat: Viistokuvaus, teksturointi, 3D -mallinnus, automaatio, kuvan orientointi

Kieli: Englanti

Preface

This thesis has mostly been done at the Finnish Geodetic Institute (FGI) in the department of Remote Sensing and Photogrammetry. I would like to thank Prof. Juha Hyyppä for the opportunity to work on my master's thesis at the 3D Modelling and Virtual Worlds research group, which is part of the Centre of Excellence in Laser Scanning. I also want to thank my adviser Lingli Zhu for having time for me whenever I knocked on her door, even if she was busy with her own research. Also many colleagues at the FGI were very helpful; for example Tomi Rosnell and Anssi Krooks. Discussions with other people working on their master thesis (R.Näsi, S.Simola and R.Kaijaluoto) at FGI were also helpful. I want to thank Milka Nuikka for making it possible to test an idea I had with a good computer at the Institute of Photogrammetry and Remote Sensing in the Aalto University School of Engineering in autumn 2014. Even if it was a small task, it did have a big influence in getting my thesis finished.

Huge thanks to Eve my girlfriend. There were times when I thought that this thesis would never be ready. I do also appreciate all the comments I got when you read my thesis during the flight home from a conference in California.

When I was studying either at Helsinki or Aalto University, I occasionally thought what a priviledge it was. When my grandfather Niilo Taskinen was at age of 25, it was the year 1943, and he was fighting in JR1 for the independency of Finland. Finally, I want to thank my mother Ritva to whom this thesis is dedicated. I have always gotten from you the positive motivation and spirit to try even if the possibility to succeed has not always been good. So thanks mom, although you are not here anymore to see me graduate!

Herttoniemi 02.02.2015
Mika Kekäläinen

Contents

Acronyms

DTM	Digital Terrain Model
DEM	Digital Elevation Model
Lidar	Light detection and ranging
ALS	Airborne Laser Scanner
TIN	Triangular irregular net
EFS	Electronic Field Study
GPS	Global Positioning System
DGPS	Differential Global Positioning System
IMU	Inertial measuriment unit
INS	Inertial system
INU	Inertial unit
AT	Automatic triangulation
CCD	Charge Coupled Devices
RMS	Root mean square
VVJ	Valtion Vanha Järjestelmä
ETRS-TM35FIN	European Terrestrial Reference System - Universal Transverse Mercator (projection) 35 Finland
ETRS-GK24	European Terrestrial Reference System - Gauss Krüger central meridian 24
EO	External orientation
IO	Internal orientation
PP	Principal point
PPA	Principal point of the autocollimator
PD	Principal distance
FC	Fiducial centre
UL	Upper left
UR	Upper right
LL	Lower left

LR	Lower right
RGB	Red, green, blue
GSD	Ground sample distance
OBJ	Object (file)
MTL	Material Template Library
FGI, GL	Finnish Geodetic Institute, Geodeettinen laitos
NLS, MML	National Land survey, Maanmittauslaitos
FKS	Fotogrammetrian ja Kaukokartoituksen Seura ry
VisualSFM	Visual structure from motion system (Wu, 2011; Wu et al., 2011)
SURE	photogrammetric surface reconstruction from imagery (Rothermel et al., 2012)

List of Figures

List of Tables

1 Introduction

The main idea of this thesis was to form automated texturing to make building façades from oblique images. The textures of the buildings have to be transformed to the ortho view because the buildings are in the perspective view in the oblique images. By making this as automatic as possible is the ideal situation. As an exemplary application, a 3D building model was generated from the used data.

The automated texturing to produce a 3D model of a building was performed with the following steps:

1 Find all the images containing a chosen building.

2 Choose the image which represents the façade in best possible way. This means picking the image which is in the the greatest angle with respect to the façade.

3 From the image chosen to represent the given façade, generate a texture of the façade.

4 Repeat the previous steps for the other façades, until the whole building is finished.

5 Generate OBJ and MTL files for a 3D model of the building by using the textures of the façades.

The above procedure can be performed automatically by coding the steps and running a suitable program such as Matlab, which was used in this thesis.

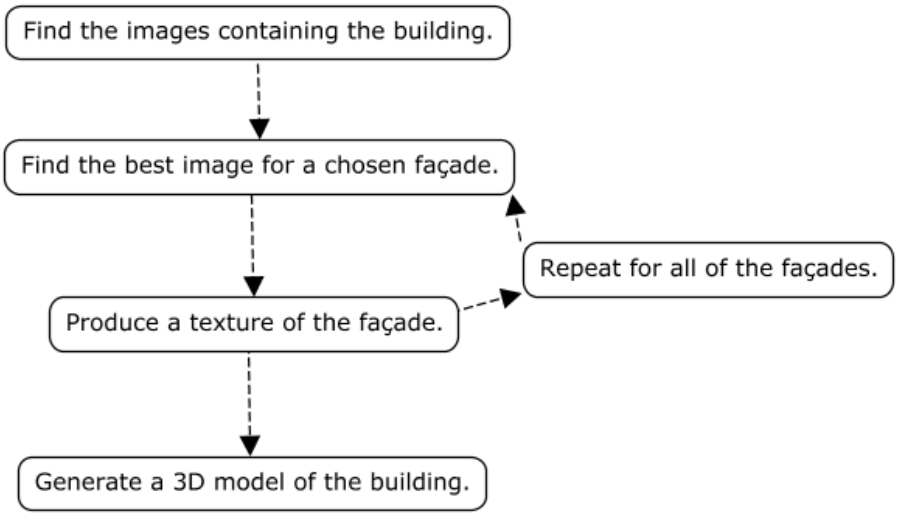

Figure 1: A simplified algorithm of the texturing process.

This master's thesis is mainly concentrated on using aerial images made with patented technology called Pictometry (Schults et al., 2008). The basic idea of the aerial images gained by the used technology is that there are multiple cameras in the airplane instead of only one. The cameras are positioned to take images in an angle of 45 degrees

and in four different main cardinal points (north, south, east and west). One camera is also needed to get the nadir images. The reason for using these images is to get images of better quality from the buildings' façades than normal aerial nadir images could produce. Oblique images, unlike orthogonal images, display also the sides of terrestrial features, such as buildings and or mountains. It is more intuitive to look at oblique images rather than orthophotos or true orthophotos. It is even possible for casual observers to recognize terrestrial features and other objects from oblique images (Schults et al., 2008).

Blom Group has an exclusive license to apply Pictometry technology in Europe. Data capturing with this technology will be executed for all cities with more than 50000 inhabitants in Europe. From October 2006, Blom Group has modified the standard approach to 60 per cent along track overlap, which allows for stereo viewing. Blom uses a business model where first the images are acquired and then it is approached to potential customers (Lemmens et al., 2007, p.1-3).

Some other companies which produce oblique images are Woolpert, the Sanborn Map Company, Fugro EarthData, Geospan and ControlCam. Also Microsoft Bing and Google have the possibility to view some cities from oblique view (Kalinski, 2013). Nokia HERE had a beta version in the autumn of 2014.

The test area for the texturing of this work was in the Keilaniemi area in Espoo, Finland. Images have to be georeferenced when they are projected onto the building façades as textures. "The primary requirement of a georeference is that it must be unique, so that there is only one location associated with a given georeference." (Longley et al., 2005, p.110) What this basically means is that every pixel of the aerial images has only one exact coordinate in the object space.

When the oblique image has accurate external orientation (EO) parameters, the projected boundaries of the façade should match the corresponding building edges in the image. When one uses the EO parameters from GPS/IMU directly, the boundaries may not be exactly the same with the actual edges of the building. One must use accurate EO parameters to create an accurate texture for 3D model. Usually, automatic triangulation (AT) is used to obtain accurate EO parameters of images (Wang et al., 2008).

Automation is needed for reducing the time and effort of the processing from unrectified and un-calibrated images to the end of producing textured 3D models. No automatic corner or edge detection methods are used in this thesis. All measurements from the buildings' vertices were made either by the user from the image (semi-automatic) or by using existing knowledge of the location of the buildings' roofs and height of the building. This data of the buildings' vertices is produced by light detection and ranging (Lidar) from an airplane.

This thesis contains six chapters: "Introduction", "Background", "The theory of basic photogrammetry", "3D models and texturing", "Results" and "Conclusion".

"Background" contains information about orthophotos, aerial photogrammetry and Pictometry. In "The theory of basic photogrammetry", I explain for example how to plan an aerial image campaign and what is the relation between the image and object coordinates. The motivation for this chapter comes from Kraus: “Knowledge of the mathematics underlying photogrammetry is essential to the understanding about the various photogrammetric techniques and processes” (Kraus, 1993, p. 4). "3D models and texturing" contains information about the data and how the data is used.

This thesis was written with LaTeX(Pihlajamäki and Pekonen, 2011; Suoranta et al., 2011).

2 Background

This chapter contains comparison of orthophotos and true orthophotos, the basic ideas of how Aerial photogrammetry with the Pictometry technique works and information about how the Pictometry cameras are calibrated. The programs that have been used to make this thesis are also introduced in this chapter.

Before digitalization and Charge Coupled Devices (CCD), aerial images were taken by analogous methods with cameras using film or glass. The use of digital images has made the processing of images much faster. 3D models from aerial images is not a new concept. It is possible to use stereo image pairs to produce them. An easy way to produce three-dimensional illusions from any digital image is to use an anaglyph method. An image made by the anaglyph method needs anaglyph glasses which have one red and one green lens. The basic idea of the anaglyph image is that there is a red and green version of the same image with a small vertical parallax between the two images, which is the key point of anaglyph images. The same idea is used with stereo image pairs which may also be grayscale images.

Oblique images give more texture in a better angle compared to nadir view images. By having a better angle of view, the 3D models will be more reliable to texture. It should also be easier to figure out textures from problematic parts of the buildings like balconies and other small bays of the building.

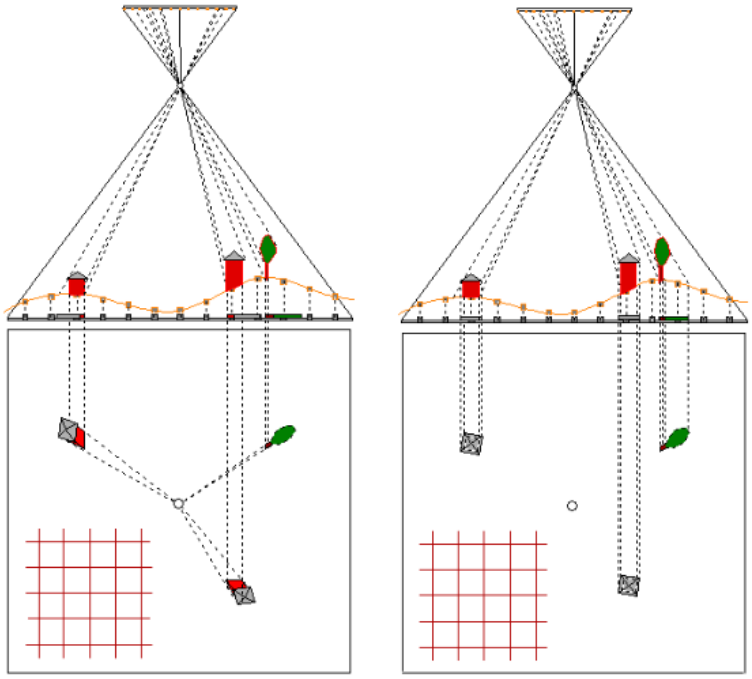

Figure 2: The difference of an orthophoto and a true orthophoto (Honkavaara and Haggrén, 2002).

On the left hand side of figure 2, there is an orthophoto and on the right hand side a true orthophoto. The basic difference between normal and true orthophotos is that

in true orthophotos, digital surface model (DSM) instead of digital elevation model (DEM) is used during the rectification process. DEM is a model which contains no vegetation or buildings. In DSM there are buildings, canopies of trees and bridges which are included in the model. The term Digital terrain model (DTM) is also used to refer to DEM data stored and modeled directly from the points.

The images in figure 2 are already georeferenced because the red coloured grid in both images down in the left corner is regular. With ungeoreferenced images, that red grid would be unregular and curved.

In figure 3, there are cubes presenting how the light affects the normal and true orthophotos. It shows a good visualization of the differences of the building façades between normal and true orthophotos.

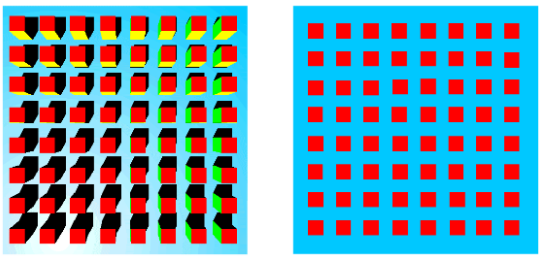

Figure 3: The difference of an orthophoto and a true orthophoto from nadir point of view (Honkavaara and Haggrén, 2002).

A true orthophoto contains the height of the buildings which means that it is good for producing maps or roof textures. Oblique images are needed to generate the textures for façades of buildings. The angle in orthophotos is not very good for facades and therefore aerial or terrestrial images with a larger oblique angle are used.

2.1 Used programs

At first, most of the work was done with Matlab and iWitness, according to the aim of this work, which was to generate a fully automatic Matlab program for texturing. iWitness is a MS Windows^{TM} based software program for measuring and modelling objects and scenes. iWitness uses the technology of close-range photogrammetry, where the three-dimensional (3D) coordinates (X, Y, Z) of feature points on an object photographed from two or more locations can be accurately determined by making two-dimensional measurements on digital images (DeChant Consulting Services - DCS Inc., 2014).

iWitness needs internal orientation for the cameras used. At least an initial guess for internal orientation parameters is needed when the program updates them during bundle adjustment. For the bundle adjustment, one has to measure tens of (the same object's) points from each image in the project. The measured points might have to be moved a bit or even to unlink from the calculation if the root mean square (RMS) error is large at some single point. When all the points have been measured and we

are happy to have small enough RMS errors, one has to bring some control points to the block. All the image external orientation parameters used will have their control points put in the same coordinate system as the buildings are in the point cloud. There is more discussion about control points in section 3D models and texturing.

Other programs which were used in this thesis, for comparison of the gained results, are VisualSFM (Wu, 2011; Wu et al., 2011) and SURE (Rothermel et al., 2012). VisualSFM was used to regenerate EO parameters for the project images and SURE for making a denser point cloud of the research area from the gained VisualSFM results. This point cloud can for example be compared with the airborne laser scanned (ALS) point cloud in a software CloudCompare. More about VisualSFM and SURE results is discussed in section Other software.

FugroViewer and CloudCompare were also used to investigate the quality of the transformed point cloud from the research area. Control points are chosen from the point cloud and tested with the National Land Surveys (NLS) internet software Karttapaikka (National Land Survey, 2013a). Also Paikkatietoikkuna which is quite handy for example investigating the most recent orthophotos from the area, was used. It is mainly a NLS service but there are many co-operation partners in the project like Finnish Geodetic Institute (FGI) (National Land Survey, 2013a).

Matlab was used to fasten the work by finding the needed parameters from the data, using a list with only the name of the needed image. It was also used to estimate the object coordinates by the collinearity equation, using the external orientation calculated earlier together with the known internal orientation.

2.2 Aerial photogrammetry and Pictometry

Pictometry oblique images have been widely used in various applications such as public safety, tax assessment, urban planning and 3D city modelling (Wang et al., 2008). Pictometry technology is patented in the United States at September 9 2008 (Schults et al., 2008). The basic idea of the camera system based on Pictometry technology used in this thesis is that, instead of using one camera in the airplane, five cameras are used. One camera is at a nadir point of view meaning that it is aimed straight down as normally in the aerial photogrammetry. Four other cameras are turned in the main cardinal points: east, west, south and north. These four cameras are approximately in 40 degree oblique position from the nadir z-axis (Lemmens et al., 2007). In figure 4, there is a drawing from the patent of the Pictometry technology about how this camera system takes images from multiple angles simultaneously.

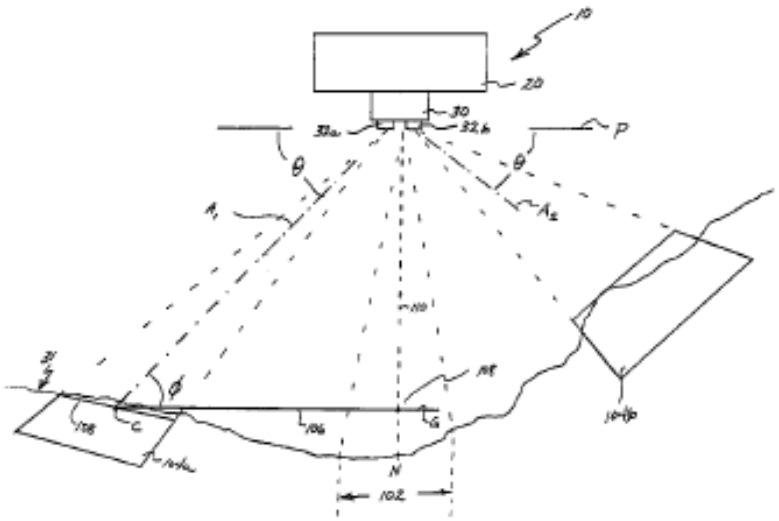

Figure 4: This figure illustrates one embodiment of a platform or vehicle carrying an image-capturing system of the present invention, and shows exemplary orthogonal and oblique images taken thereby (Schults et al., 2008).

In the patent of the Pictometry technology, there is also a drawn process image about how the images are captured and what kind of other equipment is used, figure 5.

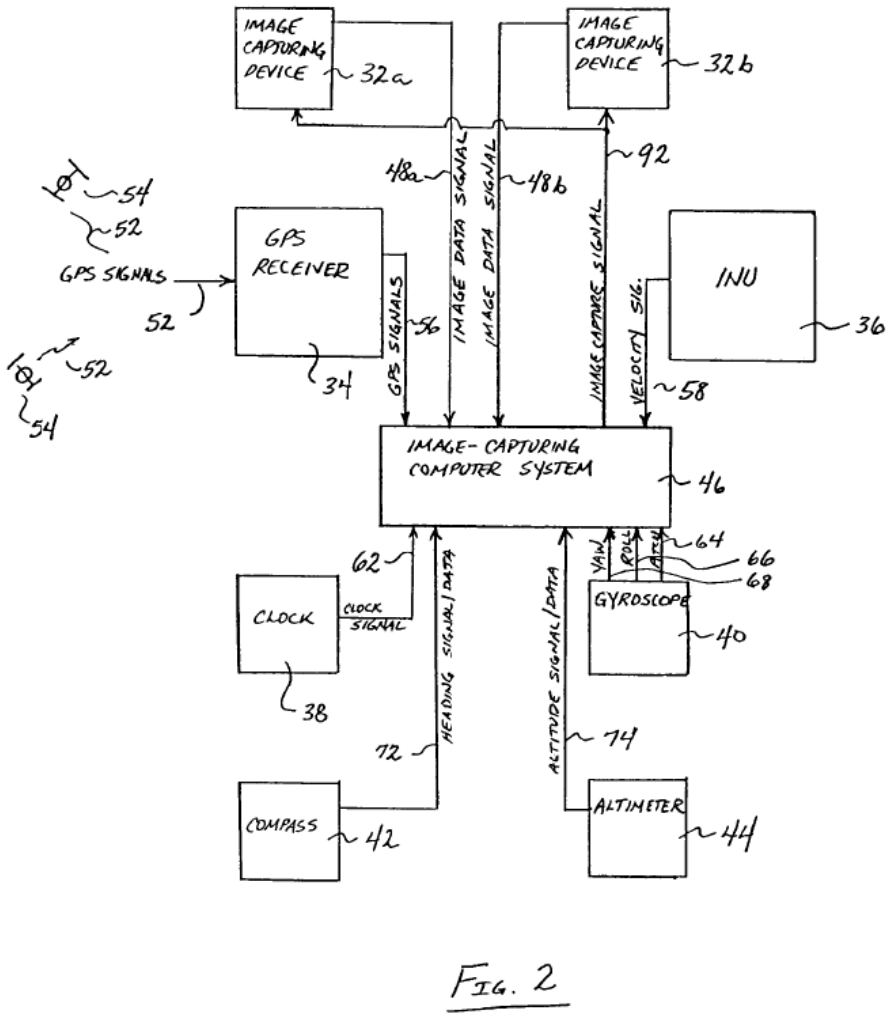

FIG. 2

Figure 5: Diagrammatic view of the image-capturing system of figure 4 (Schults et al., 2008)

As usual in an image-capturing system, except cameras, there are a global position system (GPS), an inertial measurement system (IMS) or an inertial unit (INU), a gyroscope, a clock, a compass and an altimeter in the system.

The images of this project are captured by Penta DigiCams with the sensor size of 16 megapixels (4872 times 3248). Digital cameras take the images which are then saved to the hardware with the other positioning data. More recent versions of this camera system are from 39 megapixels (7216 times 5412) to 60 megapixels (8964 times 6716). Because the camera used in the images of this project is somewhat old in today's photogrammetry high end use it was somewhat hard to find certain properties of the cameras. These modern cameras with bigger CCD sensors will give higher resolution images, aerial images and further, for texturing of 3D models or in other possible usage.

There is also a flight management system. It is a flight planning software which determines for example flight lines and control image overlaps, before and during the

flight for both vertical and oblique images. Electronic Field Study (EFS) has been developed at Pictometry. "Both vertical and oblique images can easily be viewed in EFS, and spatial measurements such as distance and height of objects on the ground can be easily performed on both oblique and vertical images. The results can be exported into ArcGIS to update directly the existing geospatial information in the database" (Wang et al., 2008).

"At Pictometry, a calibration system has been developed for calibration of its digital cameras and the system was provided to EROS center of USGS at Sioux Falls for establishing a calibration system for calibration of various digital mapping cameras in mapping community" (Wang et al., 2008). Basically, Pictometry's calibration system includes an indoor calibration cage with evenly distributed targets as shown in figure 6. Calibration is done with the software Australis (Fraser and Edmundson, 2000).

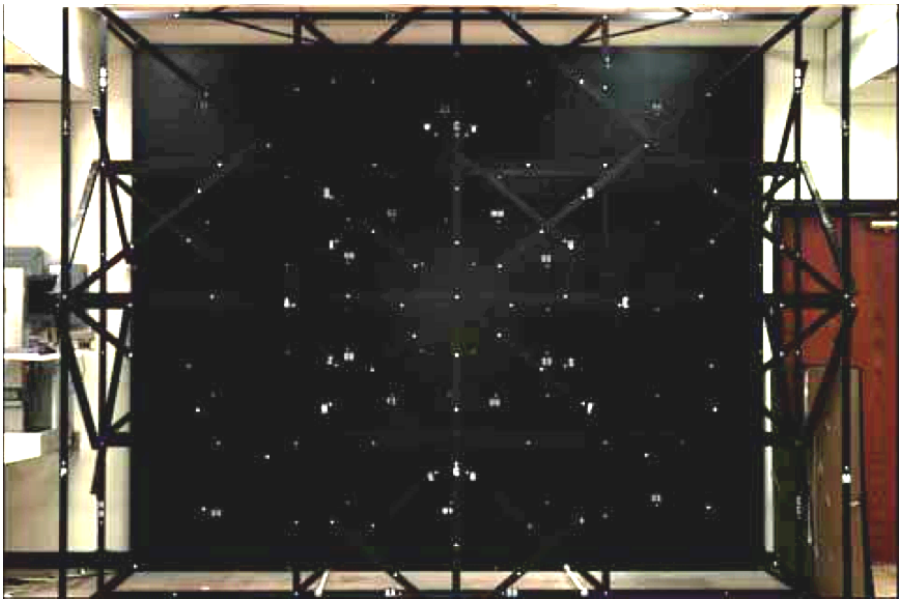

Figure 6: Calibration cage (Wang et al., 2008)

In calibration, the camera to be calibrated takes multiple images against the calibration cage from different locations to form a good network. Australis is then used to measure the targets in the images automatically and accurately as well as to perform a free network bundle adjustment to compute the IO parameters and distortion coefficients of the camera used (Wang et al., 2008).

The advantage of Pictometry's calibration system lies on its efficiency and reliability. It is very easy and economic to run calibration of digital cameras with Pictometry's calibration system, compared to other approaches such as the in-situ calibration method. Figures 7 and 8 show the radial lens distortion of a Pictometry digital camera obtained at calibrations repeated in a short time period. It can be seen that the results are consistent and the difference between two calibrations is very small. It is very important that each digital camera is calibrated regularly so that the changes of its IO parameters

between two consecutive calibrations are within the defined tolerance and do not affect the mapping accuracy (Wang et al., 2008).

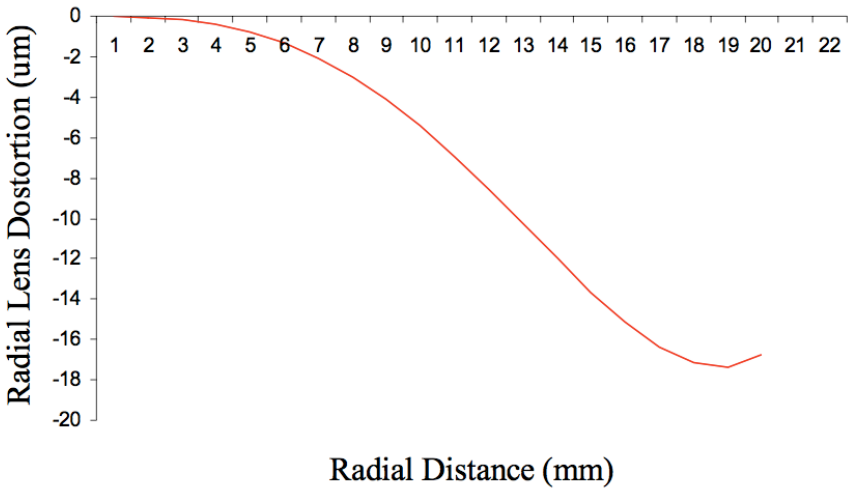

Figure 7: Radial lens distortion (Wang et al., 2008)

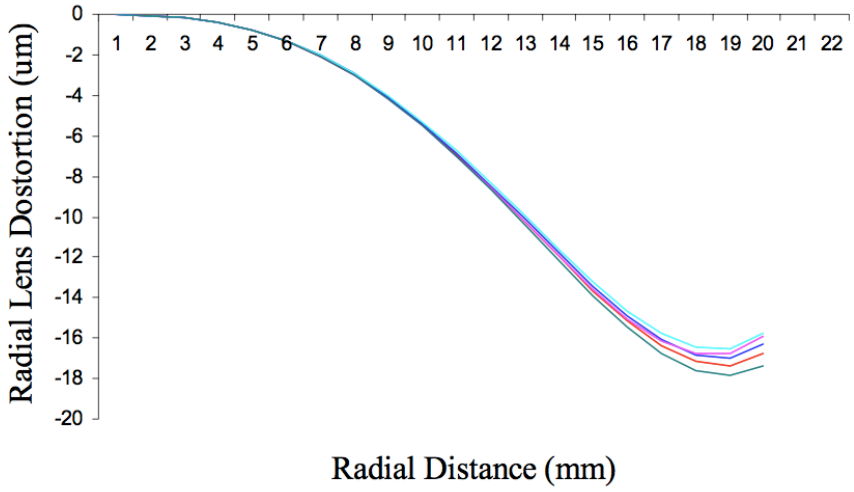

Figure 8: Radial lens distortion at repeated calibrations (Wang et al., 2008)

The measured distances, planimetric coordinates, elevations, heights, and areas depend on the calibration of the camera system, the accuracy of the outer orientation of the images and on the quality of the DTM. The tilt of oblique images is bigger than 5 gon and may reach 55 gon. Oblique images have therefore a varying scale. The area which the image pixels cover on the ground is of different size and form. The image scale is variable even in flat terrain and the sides of the image are of different length.

The scale number of a target point can be determined by the equation

$$m_\tau = \frac{h \cdot \cos(\beta - t)}{c \cdot \cos \beta}, \tag{1}$$

where m_τ is the scale number of a target point (T), h the flying height above ground, c the camera constant, β the angle between a direct line from the lens to the target and the vertical, and t the tilt of the camera axis. These parameters are depicted in figure 9. The scale number at the front and back of the oblique image can be approximately determined using $\beta_{back} = \tau - \alpha(back)$ and $\beta_{front} = \tau + \alpha(front)$, where α is the half of the field of view. The scale number at the principal point of the image can be determined by the formula

$$m_{pp} = \frac{h}{c \cdot \cos(t)}. \tag{2}$$

If image is highly oblique then it contain the horizon and the nadir point is then often outside of the image. Oblique images give a good view on façades of houses and into the backyards. Objects like lampposts can easily be recognized but behind buildings and other elevated objects are invisible areas. In some parts of the image, quality may not be optimal due the sun reflection (hot spot) and due to atmospheric effects (absorption and diffusion) (Höhle, 2008).

The measurement of objects based on oblique images requires an accurate interior orientation of the camera. Even small errors in the location of the principal point will have great influence on the results. In addition, the exterior orientation of the image has to be known accurately.

The third requirement for accurate measurements with single images is the DTM. The measurements have to take into account that the measurements of coordinates, areas, etc. are correct on the ground (bare Earth) only. Also possible errors caused from the performance of the GPS/IMU in the direct georeferencing process have to be considered. The pointing to objects with a cursor will be of different quality depending on the varying resolution and pixel form (Hamruni, 2010; Höhle, 2008, p.8-9,p.82-83).

The determination of heights of elevated objects can be carried out by measuring radial displacements. The heights may be determined approximately by the equation

$$dh \approx dr' \cdot m_T \cdot \frac{1}{cos(t) \cdot cos(\beta)}, \tag{3}$$

where dh is the height of elevated object, dr' the radial displacement of an elevated object, and m_T the scale number at the bottom of the elevated object. The scale number m_T is calculated by equation (1). The angle β is determined as $\beta = t + \tau$, with $\tau = \arctan\left(\frac{r'}{c}\right)$ (Höhle, 2008).

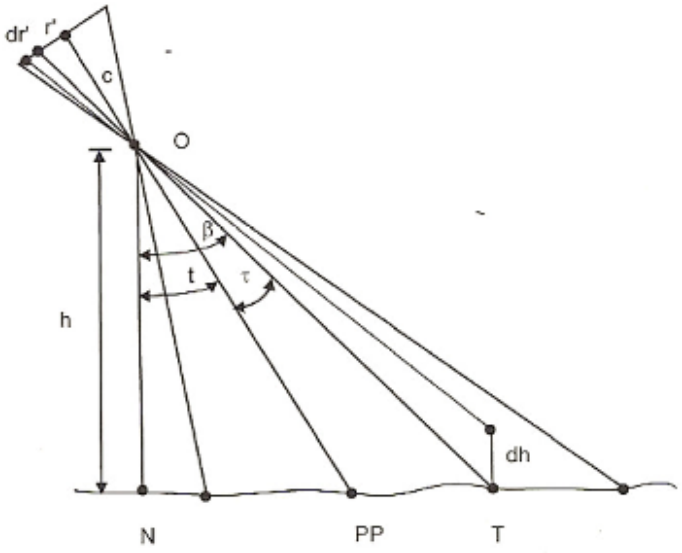

Figure 9: Determination of heights of elevated objects. (Höhle, 2008)

Georeferencing of the images is achieved by acquiring data of the differential GPS (DGPS) and an IMU of Applanix. The camera system is calibrated so that the interior and relative orientation between the five cameras is known to the manufacturer. The user (license buyer), however, has only knowledge of approximate values.

The database of the imagery is organized in sectors and blocks containing 5 by 5 sectors. Each image is stored in a file which name indicates the location (sector name), type of image (oblique or ortho), the direction of shooting (north, south, east and west) and the date of photography. A service provider has rectified the vertical images into ortho images (Höhle, 2008).

3 The theory of basic photogrammetry

In this chapter, I explain how to plan an aerial image campaign, what is the rotation matrix in two and three dimensions and how to define it. I also explain the relation between the image and object coordinates, calibration, internal and external orientation parameters and finally the important equations between the collinearity and the inverse collinearity. The motivation for this chapter comes from Kraus: "Knowledge of the mathematics underlying photogrammetry is essential to the understanding about the various photogrammetric techniques and processes" (Kraus, 1993, p. 4).
This sentence is more and more important to remember these days since all new photogrammetric techniques are based on the earlier theories.

3.1 Scale

Scale is usually familiar from maps, at least it used to be before global position service (GPS) navigators became popular in the cars and mobile devices. Basic map (peruskartta) in Finland is on the scale of 1 : 25000. On the basic map on the scale of 1:25 000 one map sheet covers an area of 12 x 24 km. One centimetre on the map corresponds to 250 metres in the terrain (National Land Survey, 2013b). Vertical photographs over flat terrain have a simple characterization of scale. The ratio of image distance and object distance gives the scale of the photograph:

$$scale = \frac{image\ distance}{object\ distance}. \tag{4}$$

From the geometry of the figure, the scale can also be expressed as the ratio of the focal length f and the flying height above terrain.

$$scale = \frac{f}{H - h}. \tag{5}$$

Here H is the flying height from the datum and h is the distance between datum and terrain.

3.2 Image coordinate system

The optical axis of a camera passes through the perspective centre and intersects the image plane at a point called the principal point (PP) or the principal point if autocollimator (PPA). The distance from the perspective centre to the image plane measured along the optical axis is the principal distance (PD). For aerial cameras and other fixed-focus cameras, the principal distance is equal to the focal length f. For close-range cameras, the principal distance is greater than the focal length and it changes with focus setting (Mikhail et al., 2001, p. 20). The images used in this thesis are aerial images and the focal length is kept as the principal distance.

Fiducial marks are no more needed since the digital image size is the size of CCD-sensor used. The fiducial centre (FC) is still defined to be the image centre in pixel coordinates. The small offset from FC to PPA is determined during camera calibration.

3.3 Aerial image planning

Fotogrammetrian ja Kaukokartoituksen Seura ry (FKS) has made a guide of recommendations for aerial photogrammetry and measurements tasks in 1995. This guide is made for analogical film cameras but the knowledge of planning an aerial imaging survey has not been changed. All of these guides are said to apply mainly to normal stereo image pair matching and the camera position is assumed to be in nadir view. Multiple parts of the guide are still useful for five camera systems with oblique images like Pictometry.

3.3.1 Height and direction planning

The flying height is determined with the used camera and from the needed scale. Fly lines are defined according to the photogrammetry measurement task, airplanes technical properties and economic efficiency. Image block area is made primarily from east to west unless the pattern or model of the research area requires something else.

The furthest sides of the fly lines have to exceed the boundaries of the research area by at least 10% times flight height plus 1cm in the image scale. In more accurate photogrammetry measurements, the flight lines are supposed to reach approximately 60% over the research area. Both ends of the fly lines have to reach over the research area by at least one image base, and in higher accuracy photogrammetry measurements, by a whole image over the research area.

A map of the aerial image plan with at least the following information (Fotogrammetrian ja Kaukokartoituksen Seura ry, 1995; National Land Survey, 2003):

- Position of the area (municipality and number of maps).
- Purpose of aerial imaging.
- Scale and hight for aerial imaging and medium height of area.
- Camera to be used.
- Film type (not needed if digital camera).
- Who has made the signals in the ground and how wide they are.
- Timetable.

3.3.2 Flying conditions

The elevation angle of the Sun has to be over 30° at the moment of the images taken. If it is necessary, for example when leafs are being opening in the spring or for some other reason, it can be agreed to be taken some other value. The vision is supposed to be good. No clouds, shadows from clouds or smoke should be seen in the sky. If the image task accepts exceptions, it needs to be agreed beforehand (Fotogrammetrian ja Kaukokartoituksen Seura ry, 1995).

3.3.3 Tolerance of gained images

The greatest allowed exception of the height of result images is ±5% of the planned height level. The realized flight lines can have the maximum of an ±10% exception of the planned height level. Forward overlap (or overlap) should be 60% ± 5% and in some exceptions 80% ± 5%. Sidelap is normally 30% ± 15% and in more accurate photogrammetry tasks 60% ± 15%. The flight line direction should not have been changed over $5gon(= 4.5°)$. Yaw of the image corner points is not supposed to be more than one centimetre in the image scale. Pitch and roll angle of images is not supposed to be over $4gon(= 3.6°)$ (Fotogrammetrian ja Kaukokartoituksen Seura ry, 1995).

3.3.4 Conditions of the need to renew the flight

The flight should be renewed partly or completely if an image contains clouds or shadows of clouds, the tolerances of gained images are not acceptable or the quality of images is not good enough. The person who is responsible for the images in the airplane is supposed to check after the images have been taken that the image block and image quality is in tolerance of what is wanted. If some of the image block is missing, a new flight mission has to be arranged. If new images are needed to be taken, their quality must be so good that there is no more reason for another flight mission (Fotogrammetrian ja Kaukokartoituksen Seura ry, 1995).

3.4 Rotation in the plane and space

If nothing else is mentioned, this section is based on the following Kraus's books: "Photogrammetry: Fundamentals and Standard Processes" (Kraus, 1993) and "Photogrammetry: Advanced Methods and Applications" (Kraus, 1997).

Consider a point $P(x, y)$ in a xy-coordinate system rotated by angle α relative to the XY-coordinate system. We require the coordinates X and Y of the point P in this second system to be

$$\begin{cases} X = xcos(\alpha) - ysin(\alpha) \\ Y = xsin(\alpha) + ycos(\alpha) \end{cases} \tag{6}$$

These equations can be written in matrix form as

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{7}$$

This, in turn, can be written in short form, with the symbols for vectors and matrices in bold script, as

$$\mathbf{X} = \mathbf{Rx},$$

where R is the rotation matrix

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \tag{8}$$

Let vectors i and j be unit vectors along the coordinate axes x and y. Components of the unit vectors i and j in the XY-system and in the rotation matrix are

$$i = \begin{bmatrix} cos(\alpha) \\ sin(\alpha) \end{bmatrix}, \quad j = \begin{bmatrix} -sin(\alpha) \\ cos(\alpha) \end{bmatrix} \tag{9}$$

$$\mathbf{R} = (\mathbf{i}, \mathbf{j}).$$

The two mutually perpendicular unit vectors must, however, satisfy the following conditions for orthogonality:

$$\begin{cases} \mathbf{i^T i} = cos^2\alpha + sin^2\alpha = r_{11}^2 + r_{21}^2 = 1 \\ \mathbf{j^T j} = sin^2\alpha + cos^2\alpha = r_{12}^2 + r_{22}^2 = 1 \\ \mathbf{i^T j} = -cos(\alpha)sin(\alpha) + sin(\alpha)cos(\alpha) = r_{11}r_{12} + r_{21}r_{22} = 0. \end{cases} \tag{10}$$

A matrix which satisfies the orthogonality conditions is called an orthogonal matrix. $\mathbf{R}$ is an orthonormalised matrix, whose determinant $det(\mathbf{R}) = 1$. If $det(\mathbf{R}) = -1$, we have a rotation and a mirror reversal. Since the four elements of the rotation matrix must satisfy the three orthogonality conditions, only one parameter is independent. In general, this is the angle of rotation, α.

3.4.1 Inversion of the rotation matrix.

Definition 3.1. *Multiplication of the inverted matrix* $\mathbf{R^{-1}}$ *by the matrix* $\mathbf{R}$ *gives the unit matrix I*

$$\mathbf{R^{-1}R} = \mathbf{I} = \mathbf{RR^{-1}}. \tag{11}$$

Lemma 3.2. *When R is orthogonal, multiplication of the transposed matrix* $\mathbf{R^T}$ *by the matrix* $\mathbf{R}$ *gives the unit matrix I*

$$\mathbf{R^T R} = \mathbf{I}. \tag{12}$$

Proof. *Using equations* (9) *and* (10)

$$\mathbf{R^T R} = \begin{bmatrix} \mathbf{i^T} \\ \mathbf{j^T} \end{bmatrix} \begin{bmatrix} \mathbf{i} & \mathbf{j} \end{bmatrix} = \begin{bmatrix} \mathbf{i^T i} & \mathbf{i^T j} \\ \mathbf{j^T i} & \mathbf{j^T j} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I}. \tag{13}$$

Theorem 3.3. *According to definition 3.1 and lemma 3.2 we get the important relation for the rotation matrix:*

$$\mathbf{R^{-1}} = \mathbf{R^T}. \tag{14}$$

3.4.2 Spatial rotation matrix

The three independent parameters are the three angles of rotation, ω, ϕ and κ, about the three coordinate axes. An counterclockwise rotation is defined to be positive here.

Rotation in 3D-space: the rotation matrix reads

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{15}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{R} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{16}$$

$$\mathbf{X} = \mathbf{Rx}. \tag{17}$$

The primary rotation omega (ω) of the xyz-system is defined about the X-axis as

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\omega) & -sin(\omega) \\ 0 & sin(\omega) & cos(\omega) \end{bmatrix} \begin{bmatrix} x_\omega \\ y_\omega \\ z_\omega \end{bmatrix} = \mathbf{R}_\omega \mathbf{x}_\omega. \tag{18}$$

The secondary rotation phi (ϕ) of the xyz-system is defined about the y_ω-axis as

$$x_\omega = \begin{bmatrix} cos(\phi) & 0 & sin(\phi) \\ 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \end{bmatrix} \begin{bmatrix} x_{\omega\phi} \\ y_{\omega\phi} \\ z_{\omega\phi} \end{bmatrix} = \mathbf{R}_\phi \mathbf{x}_{\omega\phi}. \tag{19}$$

We have the relation

$$\mathbf{X} = \mathbf{R}_\omega \mathbf{R}_\phi \mathbf{x}_{\omega\phi}. \tag{20}$$

The tertiary rotation kappa (κ) of the xyz-system is defined about the $z_{\omega\phi}$-axis as

$$x_{\omega\phi} = \begin{bmatrix} cos(\kappa) & -sin(\kappa) & 0 \\ sin(\kappa) & cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\omega\phi\kappa} \\ y_{\omega\phi\kappa} \\ z_{\omega\phi\kappa} \end{bmatrix} = \mathbf{R}_\kappa \mathbf{x}. \tag{21}$$

We also have the relation

$$\mathbf{X} = \mathbf{R}_\omega \mathbf{R}_\phi \mathbf{R}_\kappa \mathbf{x}. \tag{22}$$

The three partial matrices $\mathbf{R}_\omega, \mathbf{R}_\phi$ and $\mathbf{R}_\kappa$ are multiplied together in two steps. First, we form $\mathbf{R}_{\phi\kappa} = \mathbf{R}_\phi \mathbf{R}_\kappa$ and then $\mathbf{R}_{\omega\phi\kappa} = \mathbf{R}_\omega \mathbf{R}_{\phi\kappa}$ as

$$\mathbf{R}_{\phi\kappa} = \mathbf{R}_\phi \mathbf{R}_\kappa = \begin{bmatrix} cos(\phi) & 0 & sin(\phi) \\ 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \end{bmatrix} \begin{bmatrix} cos(\kappa) & -sin(\kappa) & 0 \\ sin(\kappa) & cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{\phi\kappa} = \begin{bmatrix} cos(\phi)cos(\kappa) & -cos(\phi)sin(\kappa) & sin(\phi) \\ sin(\kappa) & cos(\kappa) & 0 \\ -sin(\phi)cos(\kappa) & sin(\phi)sin(\kappa) & cos(\phi) \end{bmatrix}$$

$$\mathbf{R}_{\omega\phi\kappa} = \mathbf{R}_\omega \mathbf{R}_{\phi\kappa} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\omega) & -sin(\omega) \\ 0 & sin(\omega) & cos(\omega) \end{bmatrix} \begin{bmatrix} cos(\phi)cos(\kappa) & -cos(\phi)sin(\kappa) & sin(\phi) \\ sin(\kappa) & cos(\kappa) & 0 \\ -sin(\phi)cos(\kappa) & sin(\phi)sin(\kappa) & cos(\phi) \end{bmatrix}$$

$$\mathbf{R}_{\omega\phi\kappa} = \begin{bmatrix} cos(\phi)cos(\kappa) & -cos(\phi)sin(\kappa) & sin(\phi) \\ sin(\omega)sin(\phi)cos(\kappa)+cos(\omega)sin(\kappa) & -sin(\omega)sin(\phi)sin(\kappa)+cos(\omega)cos(\kappa) & -sin(\omega)cos(\phi) \\ -cos(\omega)sin(\phi)cos(\kappa)+sin(\omega)sin(\kappa) & cos(\omega)sin(\phi)sin(\kappa)+sin(\omega)cos(\kappa) & cos(\omega)cos(\phi) \end{bmatrix} \tag{23}$$

(Kraus, 1993, p.4-9 and 379-381).
The transpose of this rotation matrix is

$$\mathbf{R}^{\mathbf{T}}_{\omega\phi\kappa} = \begin{bmatrix} cos(\phi)cos(\kappa) & cos(\omega)sin(\kappa)+sin(\omega)sin(\phi)cos(\kappa) & sin(\omega)sin(\kappa)-cos(\omega)sin(\phi)cos(\kappa) \\ -cos(\phi)sin(\kappa) & cos(\omega)cos(\kappa)-sin(\omega)sin(\phi)sin(\kappa) & sin(\omega)cos(\kappa)+cos(\omega)sin(\phi)sin(\kappa) \\ sin(\phi) & -sin(\omega)cos(\phi) & cos(\omega)cos(\phi) \end{bmatrix} \tag{24}$$

$$\mathbf{R}^{\mathbf{T}} = \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \tag{25}$$

When the original data from Blom is used for automation, or any calculation at all, equation (26) below is used. There is no difference in the results telling which equation to use but it is essential to decide how to build the rotation matrix and stick with the selection made. Since there are also external calibration parameters produced with the iWitness software, another rotation matrix is also needed. Because iWitness produces angle parameters in radians, those angle parameters need to be converted to degrees with the basic equation: $2\pi(radian) = 360^o(degree)$. In the elevation angle ν $90^o(degree)$ is added before calculating the rotation matrix

$$\begin{bmatrix} cos(\alpha)cos(\kappa)-sin(\alpha)cos(\nu)sin(\kappa) & sin(\alpha)cos(\kappa)+cos(\alpha)cos(\nu)sin(\kappa) & sin(\nu)sin(\kappa) \\ -cos(\alpha)cos(\kappa)-sin(\alpha)cos(\nu)sin(\kappa) & -sin(\alpha)cos(\kappa)+cos(\alpha)cos(\nu)sin(\kappa) & sin(\nu)sin(\kappa) \\ sin(\alpha)sin(\nu) & -cos(\alpha)sin(\nu) & cos(\nu) \end{bmatrix} \tag{26}$$

(Rönnholm, 2008)

This rotation matrix is developed for terrestrial photogrammetry and it is known as (α, ν, κ) rotation, where α is the azimuth of the image direction, ν is the tilt of the image and κ is the image rotation. This (α, ν, κ) rotation matrix is constructed so that the alpha(α) rotation is counterclockwise when the nyy(ν) and kappa(κ) rotation are clockwise. The alpha(α) rotation is counterclockwise because the compass directions are used directly (Rönnholm, 2002). We will use equation (25) for both the iWitness and the Blom rotation matrices. The matrix which is used is specified by the used data.

3.4.3 Relation between image and object coordinates

Kraus's notations for collinearity are the following:

$$\begin{cases} \xi = \xi_0 - c\dfrac{r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} \\ \eta = \eta_0 - c\dfrac{r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} \end{cases} \tag{27}$$

where ξ_0 and η_0 are the image coordinates of the principal point (PP), c is the principal distance, X_0, Y_0 and Z_0 are the object coordinates of the camera position, and r_{ij}, where $j, i \in \{1, 2, 3\}$, are the parameters of the rotation matrix (with the angles ω, ϕ and κ).

It can be also written in the more intuitive matrix version

$$\begin{bmatrix} \xi - \xi_0 \\ \eta - \eta_0 \\ -c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix} \tag{28}$$

Kraus's notations for inversion of collinearity are

$$\begin{cases} X = X_0 + (Z - Z_0)\dfrac{r_{11}(\xi - \xi_0) + r_{12}(\eta - \eta_0) - r_{13}c}{r_{31}(\xi - \xi_0) + r_{32}(\eta - \eta_0) - r_{33}c} \\ Y = Y_0 + (Z - Z_0)\dfrac{r_{11}(\xi - \xi_0) + r_{12}(\eta - \eta_0) - r_{13}c}{r_{31}(\xi - \xi_0) + r_{32}(\eta - \eta_0) - r_{33}c} \end{cases} \tag{29}$$

and they can also be written in the more intuitive matrix version

$$\begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} \xi - \xi_0 \\ \eta - \eta_0 \\ -c \end{bmatrix} \tag{30}$$

Equation (30) is same as equation (28) when the rotation matrix from equation (28) is divided to the other side according to theorem 3.3.

3.4.4 Calibration

Camera calibration means that we need to figure out the IO parameters. Before we can use any of the gathered images for measuring features in the object space, we need to perform calibrations. Calibration parameters give us also knowledge of the quality of the cameras which are used in the project. The root mean square (RMS) gives us the maximal possible difference for the measured values. When working with iWitness, the program calculates us automatically the accuracy values from the measured points, whereas using data directly from Blom, we already have the calibration files for the cameras and the values for the accuracy of the images.

When the same internal orientation is used in iWitness, the external orientation which is produced from control points should be the same as in the data of Blom. VisualSFM is supposed to do automatical external orientation while using either points from the ground or some of the cameras as controls. It can be used without any given internal parameters but the results will be better if some parameters are given.

As a warning, it must be said that images for calibration and images for modelling should not have very long temporal difference. After the external orientation is performed, it can be seen in iWitness if from some of the cameras all the measured points have higher RMS values than from the other cameras. This might indicate that the camera with high RMS values has been changed or broken somehow after calibration.

3.4.5 Internal orientation

The primary parameters in the internal orientation (IO) are the focal length and a principal point for each separate camera. We also get parameters for the correction of the distortion of each image. Focal length is the distance between the CCD sensor and the centre of the lens or lens system. One camera is in the nadir view and its focal length is approximately 120 millimetres. Other cameras which are in an oblique angle have focal lengths approximately 170 millimetres.

The principal point is the exact image point where the linear ray goes through in the collinearity equation. It is usually quite near the centre of the image and is calculated in the accuracy of partial pixels.

There are four types of different distortion parameters: radial, decentring, image plane unflatness and in-plane image distortion (Fraser, 1997). Radial distortion has the greatest magnitude of these distortion parameters. It can be seen in the image as a

"barrel" or "pincushion" effect.

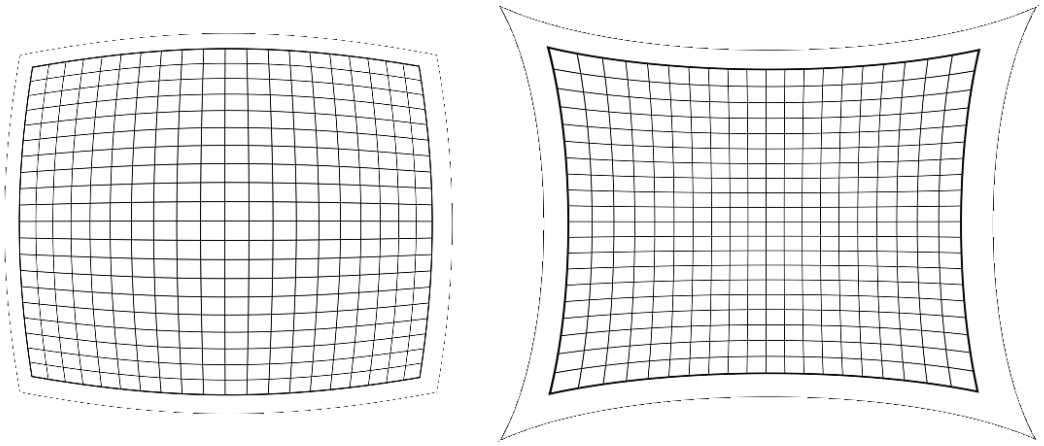

Figure 10: The difference in barrel and pincushion distortions. ©Mika Kekäläinen

In usual cases, as in this project, we only need three terms K1, K2 and K3 to define radial distortion. In projects which need less accuracy the K1 term might be enough. With a fisheye lens, more correction terms, even five, might be needed. We also need the radial distance r from the centre of the image for the equation. Decentring distortion parameters correct the error in the lens elements along the optical axis. Unflatness of the image plane or out-of-plane distortion with CCD sensors can be used as zero. This was a possible problem with an old analogical camera where there was film inside.

In this project, it was not possible to measure the surface topography directly. This is not a very major error and it is not usually appended to the physical formula of distortion. In-plane image distortion fixes scaling between horizontal and vertical pixel spacing. It also fixes skewness between the x and y axis (Fraser, 1997). The following equations are commonly used to formulate the physical model in analytical photogrammetry (Brown, 1971).

The radial distortion is

$$D_{radial} = K_1 r^3 + K_2 r^5 + K_3 r^7 \tag{31}$$

where K_1, K_2 and K_3 are coefficients of the radial distortion and r is a radial distance from the optical axis of the image: $r = \sqrt{((x - x_p)^2 + (y - y_p)^2)}$.
The decentering distortion is

$$D_{decentering} = P(r^2 + 2(x - x_p)^2) + 2(x - x_p)^2(y - y_p)^2. \tag{32}$$

Corrections to the image coordinates (x, y) are

$$\begin{cases} x' = x + \bar{x}(K_1 r^2 + K_2 r^4 + K_3 r^6 + \cdots) + [P_1(r^2 + 2\bar{x}^2) + 2P_2 \bar{x}\bar{y}][1 + P_3 r^2 + \cdots] \\ y' = y + \bar{y}(K_1 r^2 + K_2 r^4 + K_3 r^6 + \cdots) + [2P_1 \bar{x}\bar{y} + P_2(r^2 + 2\bar{x}^2)][1 + P_3 r^2 + \cdots] \end{cases}$$

(Brown, 1971).

Where (x, y) is the point in the image and $\bar{x} = x - x_p$ and $\bar{y} = y - y_p$ and (x_p, y_p) is the principal point.
A useful means of representing the magnitude of decentring distortion is via the profile function $P(r)$ which is obtained from the parameters P_1 and P_2 as

$$P(r) = (P_1^2 + P_2^2)^{1/2} r^2 \tag{33}$$

(Fraser, 1997).

"The maximum magnitudes for the radial and tangential components of decentring distortion are then obtained as $3P(r)$ and $P(r)$, respectively. For lenses employed with digital cameras, the magnitude of decentring distortion, as determined in a self-calibration, rarely exceeds $10\mu m$ at the extremity of an image format. Decentring distortion also varies with focusing but the resulting image coordinate perturbations are typically very small and the distortion variation is universally ignored in analytical photogrammetry" (Fraser, 1997).

"A significant extend decentring distortion effects can be compensated for a shift in the principal point (and an effective tilting of the optical axis). The projective compensation can usually be anticipated with CCD cameras and hence a self-calibration may indicate that the lens be treated as if it were largely free of decentring distortion" (Fraser, 1997). "Although the profile function $P(r)$ may only reach a maximum value approaching half to one pixel, decentring distortion cannot be ignored in high accuracy close range digital photogrammetric measurements" (Fraser, 1997).

Since there are only aerial images used in this thesis, the decentering distortion is left out of the equation which corrects the image distortions. Exponents for the coefficients of radial distortion K_1, K_2 and K_3 are chosen to be 1, 3 and 5 so that

$$\begin{cases} x_{radial} = (x - x_p)(K_1 r^1 + K_2 r^3 + K_3 r^5) \\ y_{radial} = (y - y_p)(K_1 r^1 + K_2 r^3 + K_3 r^5) \end{cases}$$

Even after all the distortion parameters are corrected, it might be wise to avoid using image points from the edges of the images in the measurements. The larger is the distance r from the absolute centre of the image the larger is the correction. In figure 10, there is an example of this phenomenon.

3.4.6 External orientation

External orientation (EO) produces the position (X_0, Y_0, Z_0) and the rotation angles (ω, ϕ, κ) for each separate camera. The collinearity equation define an object point (X_i, Y_i, Z_i), its image point (x_i, y_i) and the projective centre of a camera (X_0, Y_0, Z_0) are on the same linear ray. This basically means that all points and straight lines on the ground are also points and straight lines in the image.

The object point can be written as

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + \lambda \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \\ -c \end{bmatrix} \tag{34}$$

where λ is scale parameter which is divided away when formulating the collinearity equation, and r_{ij}, where $i, j \in \{1, 2, 3\}$, are the parameters of the rotation matrix.

There are a couple of ways to generate angles from the coordinate system. The most common one is to use (ω, ϕ, κ) where the coordinate axes are in the right hand model and all rotations are clockwise (or counterclockwise) towards the coordinate axis. Another method which is often used in aerial photogrammetry is (α, ν, κ) rotation, where α is the azimuth of the image direction, ν is the tilt of the image and κ is the image rotation.

The collinearity equations (27) and inverse collinearity equations (29) with the notation of equation (34) can be written as

$$\begin{cases} x - x_0 = -c\dfrac{r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} \\ y - y_0 = -c\dfrac{r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} \end{cases} \tag{35}$$

$$\begin{cases} X = X_0 + (Z - Z_0)\dfrac{r_{11}(x - x_0) + r_{21}(y - y_0) + r_{31}(-c)}{r_{13}(x - x_0) + r_{23}(y - y_0) + r_{33}(-c)} \\ Y = Y_0 + (Z - Z_0)\dfrac{r_{12}(x - x_0) + r_{22}(y - y_0) + r_{32}(-c)}{r_{13}(x - x_0) + r_{23}(y - y_0) + r_{33}(-c)} \end{cases} \tag{36}$$

3.5 Space intersection

Space intersection could be used when internal and external orientation is known. The space intersection image coordinates (x, y) from two or more images are measured and the result is (X, Y, Z), i.e., object coordinates. When the unknown object coordinates (X, Y, Z) and the image coordinates (x, y) are measured from $k \geq 2$ images, $2k$ equations are produced. From these $2k$ equations it is possible to calculate object space coordinates without any iteration and initial approximate values of the object space coordinates (Jokinen, 2013, p.61-62). This would be a simple linear explicit model: $l = Ax$. As an example, if four images would be used, then $k = 4$, the design matrix A would be of size 8×3, the vector x would be of size 3×1 and the matrix l would be of size 8×1 :

$$A = \begin{bmatrix} x_a r_{31a} + c_a r_{11a} & x_a r_{32a} + c_a r_{12a} & x_a r_{33a} + c_a r_{13a} \\ y_a r_{31a} + c_a r_{21a} & y_a r_{32a} + c_a r_{22a} & y_a r_{33a} + c_a r_{23a} \\ x_b r_{31b} + c_b r_{11b} & x_b r_{32b} + c_b r_{12b} & x_b r_{33b} + c_b r_{13b} \\ y_b r_{31b} + c_b r_{21b} & y_b r_{32b} + c_b r_{22b} & y_b r_{33b} + c_b r_{23b} \\ x_d r_{31c} + c_d r_{11d} & x_d r_{32d} + c_d r_{12d} & x_d r_{33d} + c_d r_{13d} \\ y_d r_{31c} + c_d r_{21d} & y_d r_{32d} + c_d r_{22d} & y_d r_{33d} + c_d r_{23d} \\ x_e r_{31d} + c_e r_{11e} & x_e r_{32e} + c_e r_{12e} & x_e r_{33e} + c_e r_{13e} \\ y_e r_{31d} + c_e r_{21e} & y_e r_{32e} + c_e r_{22e} & y_e r_{33e} + c_e r_{23e} \end{bmatrix}$$

$$x = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$l = \begin{bmatrix} (x_a r_{31} + c_a r_{11})X_{0a} + (x_a r_{32} + c_a r_{12})Y_{0a} + (x_a r_{33} + c_a r_{13})Z_{0a} \\ (y_a r_{31} + c_a r_{21})X_{0a} + (y_a r_{32} + c_a r_{22})Y_{0a} + (y_a r_{33} + c_a r_{23})Z_{0a} \\ (x_b r_{31} + c_b r_{11})X_{0b} + (x_b r_{32} + c_b r_{12})Y_{0b} + (x_b r_{33} + c_b r_{13})Z_{0b} \\ (y_b r_{31} + c_b r_{21})X_{0b} + (y_b r_{32} + c_b r_{22})Y_{0b} + (y_b r_{33} + c_b r_{23})Z_{0b} \\ (x_d r_{31} + c_d r_{11})X_{0d} + (x_d r_{32} + c_d r_{12})Y_{0d} + (x_d r_{33} + c_d r_{13})Z_{0d} \\ (y_d r_{31} + c_d r_{21})X_{0d} + (y_d r_{32} + c_d r_{22})Y_{0d} + (y_d r_{33} + c_d r_{23})Z_{0d} \\ (x_e r_{31} + c_e r_{11})X_{0e} + (x_e r_{32} + c_e r_{12})Y_{0e} + (x_e r_{33} + c_e r_{13})Z_{0e} \\ (y_e r_{31} + c_e r_{21})X_{0e} + (y_e r_{32} + c_e r_{22})Y_{0e} + (y_e r_{33} + c_e r_{23})Z_{0e} \end{bmatrix}$$

The indices a, b, d and e refer to four different images, the values of r come from a rotation matrix, (x, y) is the image coordinate, c is focal length and (X_0, Y_0, Z_0) is the projective centre of a camera. The normal equation is

$$A^T A\hat{x} = A^T l \tag{37}$$

and the estimated parameters from the normal equation are given by

$$\hat{x} = (A^T A)^{-1} A^T l \tag{38}$$

which are the object space 3D coordinates. However, this was never used due to the small amount of images. Instead, equation (39) below was used, with two images to calculate the three-dimensional object space coordinates.

Theorem 3.4. *Object space coordinates can be discovered from two images, if IO and EO parameters of those two images are known. They are given by*

$$\begin{cases} X = X_0 + (Z - Z_0)A \\ Y = Y_0 + (Z - Z_0)B \\ X = X_1 + (Z - Z_1)C \\ Y = Y_1 + (Z - Z_1)D \end{cases} \tag{39}$$

In equation (39)*, (X_0, Y_0, Z_0) and (X_1, Y_1, Z_1) are the two camera coordinates, (X, Y, Z) are the coordinates of the unknown object and the parameters {A, B, C, D} are a simplified notation of the inverse collinearity equations* (29) *part where is rotations and image coordinates. Equation* (36) *is the inverse collinearity equation with all the parameters written out. As an example, here is a written parameter A:*

$$A = \frac{r_{11}(x - x_0) + r_{21}(y - y_0) + r_{31}(-c)}{r_{13}(x - x_0) + r_{23}(y - y_0) + r_{33}(-c)}$$

Proof. *First, the height parameter Z is calculated from the combination of the first and the third row of equation* (39) *as*

$$\begin{aligned} X_0 + (Z - Z_0)A &= X_1 + (Z - Z_1)C \\ X_0 + ZA - Z_0A &= X_1 + ZC - Z_1C \\ ZA - ZC &= X_1 - Z_1C - X_0 + Z_0A \\ Z(A - C) &= (X_1 - Z_1C) - (X_0 - Z_0A) \\ Z &= \frac{(X_1 - Z_1C) - (X_0 - Z_0A)}{(A - C)} \end{aligned}$$

With the parameter Z, we can write the parameters X and Y for example by substituting Z in the first and second row of equation (39) as

$$
\begin{aligned}
X &= X_0 + (Z - Z_0)A \\
X &= X_0 + \left(\frac{(X_1 - Z_1C) - (X_0 - Z_0A)}{(A - C)} - Z_0\right)A \\
X &= X_0 + \Big(X_1 - Z_1C - (X_0 - Z_0A)\Big)\frac{A}{(A - C)} - Z_0A \\
X &= X_0 - Z_0A + \Big(X_1 - Z_1C - (X_0 - Z_0A)\Big)\frac{A}{(A - C)}
\end{aligned}
$$

$$
\begin{aligned}
Y &= Y_0 + (Z - Z_0)B \\
Y &= Y_0 + \left(\frac{(X_1 - Z_1C) - (X_0 - Z_0A)}{(A - C)} - Z_0\right)B \\
Y &= Y_0 + \Big(X_1 - Z_1C - (X_0 - Z_0A)\Big)\frac{B}{(A - C)} - Z_0B \\
Y &= Y_0 - Z_0B + \Big(X_1 - Z_1C - (X_0 - Z_0A)\Big)\frac{B}{(A - C)}
\end{aligned}
$$

Thus, the three-dimensional object space coordinates can be calculated from two images with different image coordinates. If the height of the object is known then only X and Y are needed to be calculated. To summarize the parameters are given by

$$
\begin{cases}
X = X_0 - Z_0A + \Big(X_1 - Z_1C - (X_0 - Z_0A)\Big)\dfrac{A}{(A - C)} \\
Y = Y_0 - Z_0B + \Big(X_1 - Z_1C - (X_0 - Z_0A)\Big)\dfrac{B}{(A - C)} \\
Z = \dfrac{(X_1 - Z_1C) - (X_0 - Z_0A)}{(A - C)}
\end{cases}
\tag{40}
$$

4 3D models and texturing

In this chapter, I describe how to transform the Espoo city local coordinate system to ETRS-TM35FIN. There is also a map from the research area as well as information about the data parameters and statistics of when and by which camera the Pictometry images were taken. I describe how to find and choose the best image for a certain building façade and prove the method for calculating object space coordinates from two images when IO and EO parameters are known. Then I explain how to make an orthophoto of a façade before making a textured 3D model. In the end of this chapter, I introduce the 3D model and present what was done with other software: Terra, iWitness, VisualSFM, SURE and Nokia HERE.
The used aerial point clouds in which all the vertices and faces are taken from an Espoo city local coordinate system called Helsingin järjestelmä or Vanha valtion järjestelmä (VVJ) and the height system is N60. In the VVJ coordinate system Hayford ellipsoid and Gauss-Krüger projection were used.

In the Blom material, EO parameters of the cameras are in ETRS-TM35FIN coordinates and in the height N2000. Both ALS dataset and the Pictometry image data from Blom have to be in the same coordinate system. Therefore, all coordinates from the ALS point clouds have to be converted into the same coordinate system with the Blom camera parameters. The easiest way for this is to transform points from VVJ into ETRS-TM35FIN with N2000.

Espoo city VVJ coordinates are converted with an affine transformation or Helmert transformation. The height parameter is updated from N60 to N2000 by adding the constant +247mm to the original height. Coordinates are transformed at first from VVJ to ETRS-GK24 and then to ETRS-TM35FIN.

The conversion of the Espoo city local coordinates VVJ to ETRS-GK24 is given by

$$\begin{cases} X = a + cx + dy \\ Y = b + ex + fy \\ Z = z + 247, \end{cases} \tag{41}$$

where x = Espoo VVJ(X) and y = Espoo VVJ(Y) (City of Espoo, 2011).
We need to use the Espoo mainland (Manner-Espoon) ETRS-GK24 conversion (City of Espoo, 2011):

$$\begin{cases} a = 6599858.007479810200000 \\ b = 24499824.978235636000000 \\ c = 0.999998786628487 \\ d = 0.000020762261526 \\ e = -0.000014784506306 \\ f = 0.999996546603269 \end{cases}$$

and we use the most accurate conversion (i.e., the conversion priority is one) (City of Espoo, 2011). This still needs to be transformed to ETRS-TM35FIN with for example the following online internet tool: www.coordtrans.fgi.fi.

To summarize, the process of coordinate transformation is

$$VVJ \;\rightarrow\; ETRS-GK24 \;\rightarrow\; ETRS-TM35FIN.$$

The building vertices which are used as control points in the ETRS-TM35FIN coordinates are listed in table 1.

ID	E(m)	N(m)	High(m)
1	379608.98	6672739.22	21.80
2	379668.38	6672760.70	21.98
3	379647.17	6672819.91	21.77
4	379587.18	6672799.07	21.83
5	379657.49	6672843.52	78.69
6	379675.88	6672849.88	78.70
7	379661.53	6672889.50	78.61
8	379643.13	6672883.00	78.73
9	379622.11	6672569.56	37.96
10	379641.42	6672576.82	37.89
11	379597.19	6672699.83	37.84
12	379577.39	6672691.05	37.49
13	379535.18	6672456.39	22.71
14	379529.01	6672477.92	23.02
15	379526.38	6672485.23	35.95
16	379468.35	6672448.55	34.63
17	379499.48	6672443.91	22.86
18	379438.78	6672932.50	36.74
19	379454.72	6672949.00	36.80
20	379442.09	6672961.50	36.81
21	379428.06	6672951.00	36.81
22	379696.91	6672857.50	19.51
23	379717.81	6672865.00	19.57
24	379694.81	6672928.50	19.51
25	379674.75	6672921.00	19.40

Table 1: Control point coordinates.

In general cases, photogrammetric methods are more accurate in determining building outlines. Building outlines and length deviations are better obtained from image measurements than point clouds, but the target height deviation is better from point clouds (Kaartinen et al., 2005a,b).

Control point coordinates in table 1 are measured from a dense ALS point cloud from the research area. Because the coordinates are measured one by one with computer mouse from the point cloud, there is a small possibility of not measuring the right point or the best point. An example of a good point is a corner of a known building. Control points should also be measured in as many images in the project as possible.

The use of the control points defines the external orientation parameters in iWitness. If only data from Blom was used then the role of the control points was more of to test

that the written Matlab codes work properly. This means that one either measures a control point from two images and calculates its 3D position with the inverse collinearity equation (29), or uses a control point and the position of the camera to calculate the point in the image with the collinearity equation (27).

4.1 Map of the research area

The area of interest is Keilaniemi and Keilalahti region in the eastern part of the city of Espoo. The map containing the research area is in figure 11.

Figure 11: Map of the Keilaniemi research area (National Land Survey, 2010).

The object space coordinates of the map in figure 11 are at the upper left (UL) point north: 6673837 and east: 378731 and on the lower right (LR) point north: 6672005 and east: 380670. In the maps which NLS produces, this area can be found in the map leafs L4131H and L4133B (National Land Survey, 2005).

As it can be seen from figure 11, there is a lot of seawater in the map area. Buildings on the east side of the area between Karhusaarentie and Keilalahti bay are mostly

high office buildings. The west side of the area from Karhusaarentie consists mostly of apartment houses and there is also a forest park of pinewoods and deciduous trees.

4.2 Data

The original images were in *.psi format and they were transformed to both *.jpg and *.tif images with IfranView version 4.37. The arrangement of cameras in the airplane is depicted in figure 12. It is a demonstration of how the photo shooting is performed in practice; figures of the cameras Blom has used are not available: "Although Pictometry is the leading player in this field, they will not release any details of their imaging technology" (Petrie, 2008).

Figure 12: An IGI Penta-DigiCAM system with a single vertical camera (equipped with an f=28 mm lens) in the centre. It is flanked by the four tilted cameras (each with an f=150mm lens) that generate the surrounding oblique images of the ground. An AEROcontrol IMU unit has been mounted directly over the vertical camera (Petrie, 2009).

Only images from Keilaniemi (or Keilalahti) area were taken in use but there is a possibility to use larger area and produce more buildings to textured 3D models. There are temporal differences in the images which means multiple flights over the research area. For having more reliable orientation parameters, these differences in the flight date were undesired. After finding all the images from the research area and comparing the date, the final images to use in my project were chosen. From nadir view there are five images and from oblique view two or three from each camera. Overall 15 images were used. There are more images from the nadir view because the airplane flew both directions from north to south and south to north. It would have been better if there would have been more images, especially oblique ones, with the same date.

Date	North	East	South	West	nadir
15.07.2010	41	28	7	30	97
22.07.2010	20	5	6	7	21
07.08.2010	0	0	23	0	0
16.08.2010	13	0	0	0	52
17.08.2010	0	15	36	15	0
30.08.2010	19	58	26	59	201
Sum:	93	106	98	111	371
Good:	61	33	13	37	118

Table 2: Amount of images for each camera.

We did not use the nadir view images since the main aim was to produce textures of building façades. Only the images which are marked as green in table 2, taken 15th and 22nd of July, are used. The images which have the date stamp 7th, 16th or 17th August are not used since there are no images from all of the cameras. The images taken 30th August have neither been used since the temporary difference is over a month.

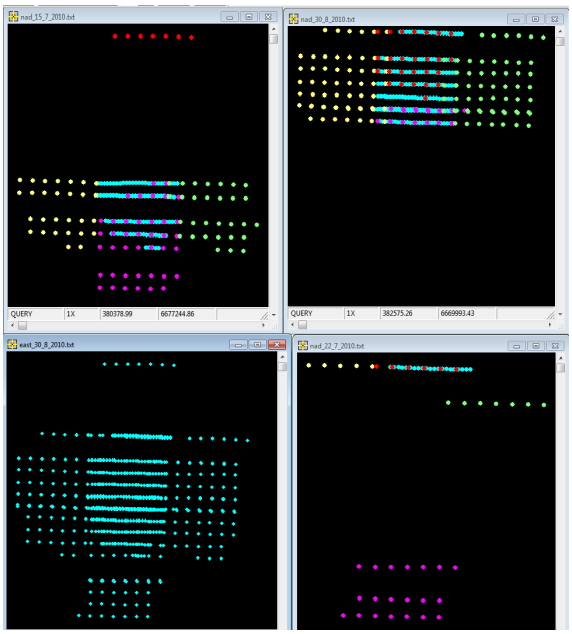

Figure 13: Colored points of all used camera position. (red=north, green=east, pink=south, yellow=west and blue=nadir direction.)

In figure 13, positions of the cameras from 15.07.2010, 30.08.2010 and 22.07.2010 are coloured in the upper left, upper right and lower right parts of the image, respectively. The lower left part of the image is a combined image of these three cases.

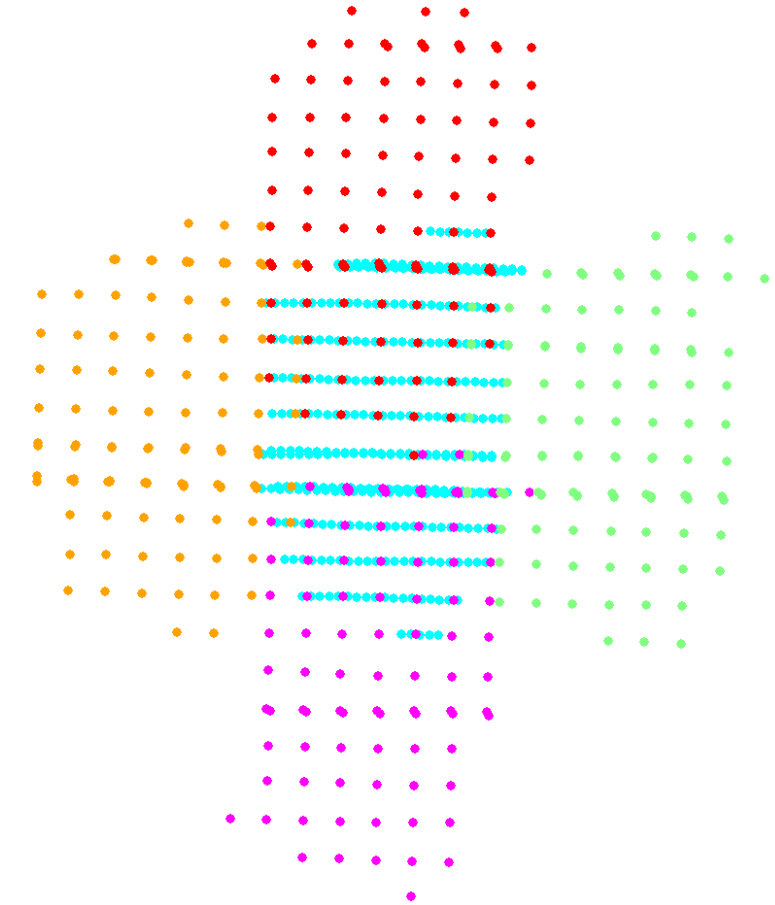

Figure 14: Coloured points depicting all the used camera positions.

In figure 14, the (X, Y, Z) coordinates of all of the cameras lie in the same point cloud. Red describes north, pink south, green east, orange west and blue the nadir camera position. When all the positions of the cameras are plotted, the flight lines and camera positions look quite good. However, the problem is that even if we have images this regular the difference in flight days makes the use of all images not so ideal.

In figure 15, there is a sample of images taken 30th August 2010 on the left hand side and 15th July 2010 on the right hand side. There is a crane making a new building in a newer image on the left hand side, as it is pointed out by white squares. There are also radiometrical differences between these images. It seems that the weather conditions are radically different from sunshine to heavy clouds.

Figure 15: The difference between two separate flight results.

The image irradiance of a point in the image plane is defined to be the power per unit area of radiant energy falling on the image plane. Radiance is outgoing energy; irradiance is incoming energy.

Two factors determine the radiance reflected by a patch of a scene surface:

- the illumination falling on the patch of the scene surface
- the fraction of the incident illumination that is reflected by the surface

(Ramesh et al., 1995, p.257-258).

The reflectance from the same objects does not change but the difference in the lighting affects the radiance which affects the radiometric model of the digital image. Therefore, it is more suitable to use images which have a very small temporal difference. Temporal difference will cause even more problems when more automation is used in the calculation to get the orientation parameters. This is possibly one reason why there were problems in gaining just one point cloud, instead of several different point clouds, from the images by using VisualSFM.

For all the images there is a huge data file where all kinds of different parameters of each image are stored. Table 3 is an example of all of the image parameters. Unfortunately, there is no description file of the used field names and parameter units. The fields which are used in the thesis are: ImageName, ShotOrient, ShotDate, GSD, ImageRows, ImageCols, ULx, ULy, URx, URy, LRx, LRy, LLx, LLy, Alt, FocalLen, K1, K2, K3, PPx, PPy, Camera, CameraX, CameraY, Omega, Phi and Kappa.

Fieldname	Data
ImageName	FIHELS045027NeighObliq50760E_100830
ImagePath	FIHELS041026 cluster/FIHELS045027/
ImageExt	psi
ShotLevel	N
ShotType	Ob
ShotOrient	E
ShotDate	2010/08/30 07:46:09
GSD	0.1099
ImageRows	3248
ImageCols	4872
ULx	380675.5719
ULy	6673288.607
URx	380617.2878
URy	6672652.96
LRx	380001.844
LRy	6672717.483
LLx	380048.3151
LLy	6673250.899
Flags	3
CameraLat	1.050251015
CameraLon	0.432909345
Bearing	3.130982043
Roll	0.898284075
Declin	0.059402556
Alt	1701.171858
Elevation	0.43
FPx	36.0528
FPy	24.0352
FocalLen	169.8702
K1	-2.68E-11
K2	1.00E-17
K3	-1.03E-24
PPx	0.001589336
PPy	0.002537944
Camera	[3]
CameraX	378167.8287
CameraY	6672921.297
Omega	0.030999773
Phi	-0.899575446
Kappa	-1.607110536
BalancedFL	169.8676564
BalancedK0	-1.50E-05
BalancedK1	4.89E-07
BalancedK2	-3.34E-09
BalancedK3	6.28E-12
Project	FIHELS_FP014_IOD_1191A_100830
System	Universal Transverse Mercator
SubSystem Zone	35N (24 E to 30 E)
Datum	WGS 1984
Units	Metre

Table 3: Example of the original data of one image.

4.3 Finding images from research area

At first, one has to find all of the images which have buildings from the desired area. This selection of images can be done manually or automatically. The manual searching operation can be performed by looking from an image folder which images look like they might be in the area, or at least this is a way to reject some folders of images where there is only seawater. A more efficient technique is to use the upper left (UL) and lower right (LR) coordinates from Blom data files, which indicate the pixel coordinates of the image UL and LR in the object space.

The use of these UL and LR is not so simple as it sounds because the resulting images from the use of Pictometry technology are in multiple directions and there have been flights in both north to south and south to north directions. Only images in the north direction are in a correct direction and all other images have to be converted by using either different corners like an upper right (UR) and lower left (LL) or by changing the order of the (x, y) coordinates of UL and LR.

There is one column in the data files which tells the direction of the upper edge of images. With the knowledge of the direction of the upper edge it is possible to choose the right parameters of each image and therefore automatically find from the object space coordinates which images can be used.

Another thing is to open the chosen images and think about the quality and possible errors with the gained result. As it was seen earlier in table 2, some of the images from the right area were left out because of the temporal difference and the lack of cameras used at the same time during the flight.

I have not used the images in this thesis unless all five cameras were used at the same time. The idea of Pictometry technology (Schults et al., 2008) is to generate images with five cameras together at the same flight, four from oblique view and one from ortho view.

4.4 Choice of the best image

To get the best image for usage in the building modelling one has to find the image which is the nearest and in the smallest angle to the normal vector of the building façade. This was done by Matlab since it has many good existing tools for that. From one image, at least three image points from the desired objects or building's vertices were measured to get the level of the façade.

Because the camera parameters are in the ETRS-TM35FIN coordinates and in the height N2000, the normal vector and the centre point of the façade have to be in the same 3D coordinate system. This is possible to do either manually or with the Matlab Ginput tool (The MathWorks, Inc, 1994). The automatic version needs some existing data of building vertex coordinates in a right coordinate system. No kind of corner and/or edge detection has been used to find a vertex.

Since we know the internal and external orientation, the image coordinates were estimated with the known equations of collinearity with the properties of coplanarity to the object space coordinates. For the level, a central point of the points measured from an image was calculated. At the central point, a normal vector was calculated which is in 90 degree angle against the formed level. A normal vector can be formed for example by making a cross product between the vectors from the centre point to two measured corner points.

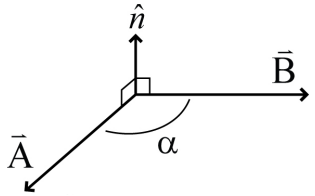

Figure 16: Cross product (Kirby, 2009).

When all the vectors are in the same coordinates, we can calculate a camera vector to the centre point. The angle between the camera vector and the normal vector is calculated with the basic equation

$$cos(\alpha) = \frac{(k * l)}{(|k| * |l|)}, \tag{42}$$

where k is the normal vector from the centre point and l is the camera vector from the centre point to the camera position coordinates. Angle measurement is done from two image pairs. This needs to be done for all the images where that part of a building's façade is shown. As a result, the code gives the matrix of the calculated angles, possible errors with bad images and offers the best image number or name for the use in the next step.

Any kind of method with least square adjustments involved has not been used. If the bundle (in iWitness) is strong enough, two images should be enough for producing accurate results (redundant images are the most reliable). One problem was the small amount of images for façades. If there is no more than one to three images from a certain façade, it is not meaningful to use least square methods. Multiple iteration might give better results but this was not the aim of this thesis.

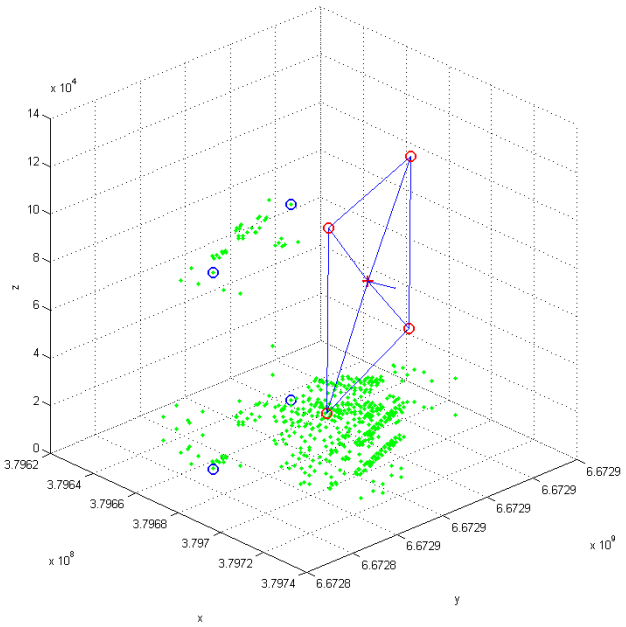

Figure 17: Demonstration of a normal vector in the situation where image points are measured manually from an image. All the axis are in millimetres.

The green points in figure 17 come from an ALS point cloud. The red circles are 3D points measured from an image with the inversion of collinearity. The red cross is the centre of red circles and the position of a normality vector. The blue circles are the points where the red circles are supposed to be. The blue arrows or lines are vectors from one 3D point to another, and also the normal vector is drawn from the centre point. The difference between a measured 3D point and the real position of the 3D point is approximately 56 metres in the east direction, 1 metre in the north direction and 43 metres in the height direction. This difference between measured coordinates and reality can still be seen in the image but it is not very easy to recognize.

In figure 18, the image which was used to measure vertex points is given. The measured points are drawn in the same image by red circles and one cross. The corner points of the building are very close to the reality. The red cross in the centre is the only point which does not seem to be in the realistic centre of the four corner points.

Figure 18: Vertices and centre point in the image.

Because some of the measurements look good when they are drawn in the figure, there was no reason to test where the measured coordinates are in reality until problems occurred in producing orthophotos of certain façades. These measurements were actually so good that it was possible to make a red, green and blue (RGB) coloured point cloud from a nadir view image.

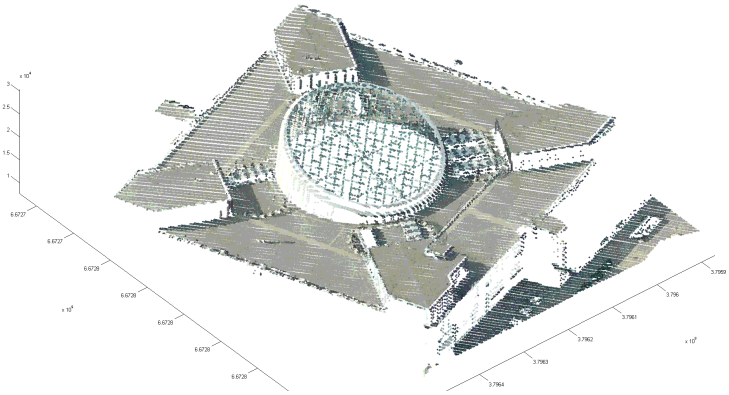

Figure 19: ALS point cloud with RGB colours from the Pictometry image (axis scale are millimetres).

Figure 19 is made from the nadir view image and from the centre of the image where there is smaller radial distortion than edges or corners to be fixed. Part of the ALS point cloud points are coloured with the RGB values from the image by using the collinearity equations (27) to find the point from the image and inversion collinearity equation (29) to produce a three-dimensional figure of the point cloud with RGB coloured points.

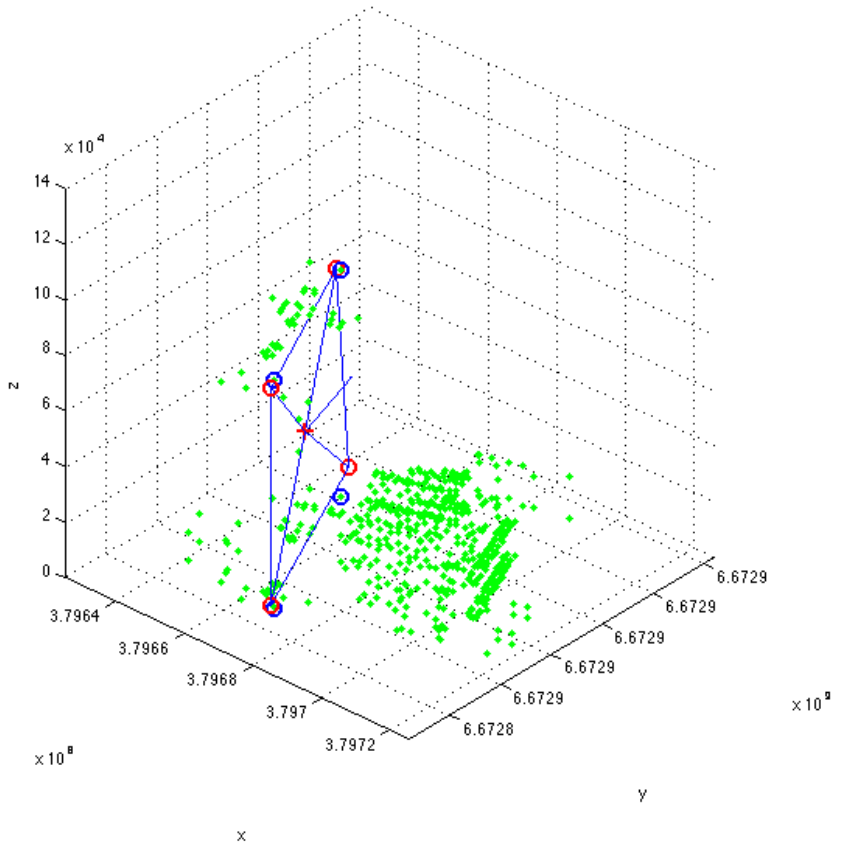

Figure 20: Demonstration of a normal vector in the situation where the image points are measured manually from the image with SURE data used for orientation. All units are in millimetres.

In figure 20, the situation where the image points are measured manually from the image with SURE data used for orientation is depicted. This is to be compared with figure 17 where the errors are much greater. As before, the green points come from an ALS point cloud. The red circles are 3D points measured from an image with the inversion of collinearity when the orientation parameters from SURE are used. The red cross is the centre of red circles and the position of a normality vector. The blue circles are the points where the red circles are supposed to be. The blue arrows or lines are vectors from one 3D point to other, and the normal vector is drawn from the centre point. The difference between a measured 3D point and the real position of the 3D point is less than 2 metres. The accuracy depends on how well were the measurements made from an image and from the used building vertex coordinates.

4.5 Orthophoto from the façade

When the best images are chosen for each façade, the next thing to do is either to use the existing data of the building's vertices in the right coordinates or to measure the vertices from the images. The degree of automation is higher if existing data points are used. With manually measuring points from a chosen image, it is easy to make more complex shapes with multiple triangles than using large squares.

The Matlab function which makes the orthophoto of each separate façade needs the list of vertices and the vertices' triangulations and the knowledge of the internal and external orientation parameters of the image. The distortion parameters of the image are included in the internal parameters.

Because something is making an error in the measurements, as can be seen in figure 17, one cannot produce the automatic orthophoto from the original parameters. The potential error of missing one decimal of the pixel size in the calibration files does not improve the tens of metres of errors in the measurements. In section Results, there is more discussion about the errors.

After the range of the error had appeared with the 3D figures like figure 17 or had been found out with NLS online map service (National Land Survey, 2013a) a lot of time was used to figure out the reason for this. Since no answer was found, another external orientation version was done with iWitness and SURE from VisualSFM results, using the written Matlab functions. For iWitness, it seemed that there were not enough good images to use and the external orientation parameters did not improve. (This still does not mean that there would not be errors in the Matlab functions, for example in the indices.) There is more discussion about this in the next section Results.

When using the original parameters or iWitness parameters, the problem was that if the points were measured from an image, the resulting image was not an orthophoto. If the 3D coordinates were used, the resulting image would be an orthophoto but there was transfer in it. This shift against the real position of a building façade was big enough that the resulting images were more or less useless for producing textures to the 3D models. When using *.ori files generated by SURE it was possible to generate orthophotos from a façade with small amount of pixel error.

The texturing of the 3D model was done with a little modification to the Matlab function *orthoimage_ terrestial.m* made by my advisor Lingli Zhu. The main idea of the orthophotos is to convert new images which look like they would have been taken perpendicular towards the façade. The reason for this is that in the final texturing of the 3D model, the building which is modelled would look good from all angles. If the oblique images would be used straight away, textures would also be oblique. Basically, this means that windows and balconies would be in a skew position instead of direct. An example of possible skew in the windows and balconies can be seen in the two images figure 21.

Figure 21: Example of an image and an orthophoto from the used data.

Since there is something that makes measurements from images to go wrong when using the original parameters or iWitness parameters, one has to at least crop the images before they can be used further. Cropping is not enough in most cases since parts of the walls are missing from the orthophotos. In this stage, making a realistic texturing of 3D building models of any kind was not possible without moving points for measurements manually. This manual procedure was mainly done to obtain some kind of a model to prove that it is possible to do automated texturing from a zero point to the *.obj file containing realistic looking texturing of 3D model with Matlab. The original orthophotos were saved before the cropping operation.

After the images were cropped, the cropped orthophotos were modified by the image size because it is more efficient to computers to calculate images whose sizes are $2^n * 2^m$, where $n, m \in \{1, 2, 3, \dots\}$, than irregular sized orthophotos. After optimization, the final orthophotos were saved in a folder named by the address of building or as "building_1".

The optimization algorithm, which goes through the image coordinates, is quadratic and its complexity class is $O(n^2)$ (Floréen, 2013, p.51-52,p.92). Quadratic algorithms usually have two nested for-loops

$$\begin{aligned} &for\ i = 1\ to\ n \\ &\quad for\ j = 1\ to\ m \\ &\quad\quad \dots \end{aligned}$$

where n and m are image sizes (width and height) in pixels. The image coordinates are investigated pixel by pixel beginning from the upper left coordinate. Every column of each row is investigated with the quadratic algorithm. When RGB images are used, the algorithm is cubic and its complexity class is $O(n^3)$ (Floréen, 2013, p.51-52,p.92). Cubiq algorithms usually have three nested for-loops

$$\begin{aligned} &for\ i = 1\ to\ n \\ &\quad for\ j = 1\ to\ m \\ &\quad\quad for\ k = 1\ to\ l \\ &\quad\quad\quad \dots \end{aligned}$$

where n and m are image size (width and height) in pixels and $l \in \{1, 2, 3\}$ is the RGB value of the image I.

4.6 3D models

Photogrammetry and remote sensing are defined as image-based modelling techniques, which allow for the extraction of both geometrical and semantic information from images. Image-based modelling is the most widely used method for geometric surface of architectural objects or for precise terrain and city modelling. Image-based modelling methods use 2D image measurements (correspondences) to recover 3D object information. 3D city modelling has been an active research area in digital photogrammetry for more than a decade and a number of methods and systems have been developed for creating 3D city models from digital images and other auxiliary data automatically or semiautomatically. The two major steps involved in generating 3D city models are the creation of building geometry and adding textures (Hamruni, 2010, p.15-16).

Since the mid-1990s, the demand for 3D city models is continuously growing and has meanwhile become commonplace. Especially Internet applications like Google Earth, Bing Maps or HERE and other advanced navigation systems carry 3D city models for a huge group of potential users. "The spread and wide range of applications also result in more heterogeneous demands ranging from the provision of input data for the photorealistic real-time visualization of terrestrial walkthroughs to interpreted scenes for advanced search and navigation" (Haala and Kada, 2010).

According to (Varshosaz, 2003), the reasons to use real textures in façades can be written in a few points:

- Improved realism and ease of access: incorporating real textures not only increases the realism of the model but also enables the viewing of details which are absent from the geometric model. Objects can instantly be recognised and accessed as required.

- Improved interpretability: when models are enhanced using real textures, the relationship between objects can easily be perceived. This not only includes features on the façades of buildings such as doors and windows, but also the surrounding buildings and features like telephone boxes, etc.

- Making instant measurements: a geometric model rendered with real textures can be used as a base to perform measurements on objects that may not even be included in the geometric model. The accuracy of these measurements, however, depends on the accuracy of the geometric model and the way the measurements are carried out.

3D models are generated as Wavefront *.obj file with *.mtl file. The OBJ format basically needs all vertex coordinates (v), vertex texture coordinates (vt) and façades (f). Vertices' normal coordinates can be given (vn) but those can be calculated also in the Meshlab program. Vertices' coordinates are the used buildings' real coordinates in Espoo. Vertices' texture coordinates are in this simple case a square. Façades compile

a right vertex line number with right vertex and optionally also with the right normal line number. Before the façade line (f) one has to write which orthophotos are used with the line: “usemtl new_material_1”.

If the used vertex coordinates are wrong, the results can be seen with a combination of the building’s 3D model and the point cloud of the same area. In the example of figure 22, vertex points which form the base of the building are at the sea level instead of the same level with the point cloud of the surface.

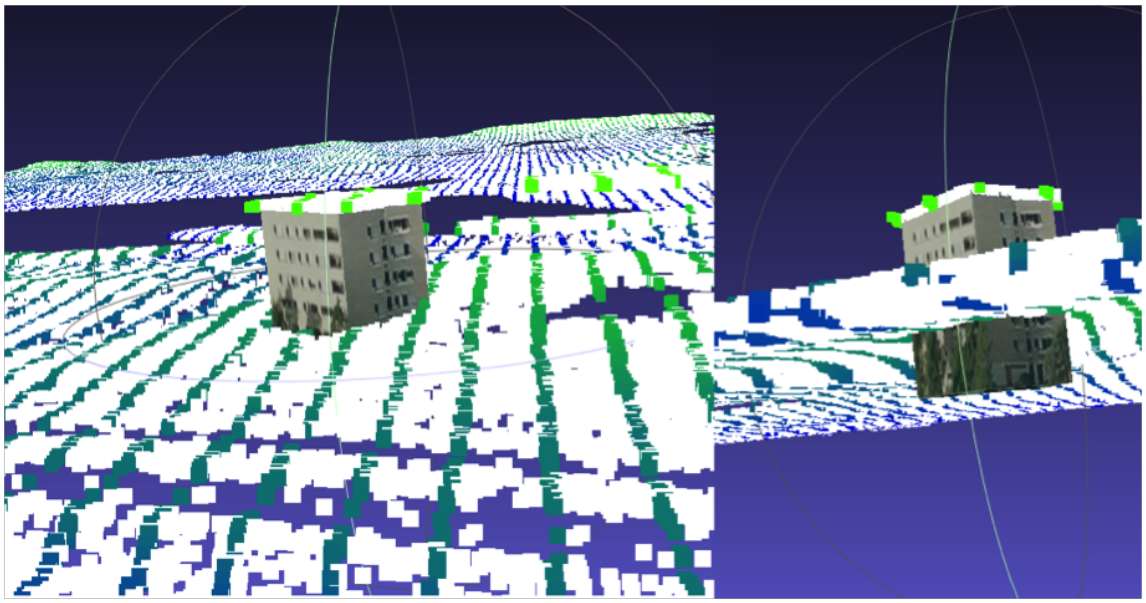

Figure 22: A 3D model of one building with point cloud of the terrain (DTM). (Notice that the base of the building is set to be on sea level and not on the real terrain model.)

To generate textures as the *.obj file one needs a *.mtl file. In the *.mtl file, each image is generated with the basic settings. Every new material begins with “newmtl new_material_1” where new_material_1 is the name used in the *.obj file. In the *.mtl file each material’s properties can be modified separately if needed. The most important issue is to create a correct path for each image used in the command “map_Kd northWall_texture.jpg” if the images are not saved in the same place as the *.obj and *.mtl files are. To be able to use the generated *.mtl file one, has to write in the *.obj file command “mtllib building1.mtl” where building1.mtl is the name of the *.mtl file saved. Look for appendix 3 for examples of *.obj and *.mtl file done with the function export_obj_by_mke.m in Matlab.

In figure 24, there is the scene of two buildings with the façade textures. There are also other grey coloured buildings in the back of the image. In this version, roofs are missing textures because those can be easily made from the nadir view images or from NLS open source true orthophotos.

This *.mtl file is done as one image or texture for each façade and that is the reason why there are so many newmtl sections. Another way to do to this is to generate one image named as texture map, where all the textures for all the façades of the building

which is under progress are stored. The optimized size for the texture map would be for example $2^{11} \times 2^{11} = 2048 \times 2048$ (pixels).

Figure 23: A texture map of the building's location in Espoo Harjuviita 20-22.

In figure 23, there are the same textures as in figures 22 and 24. This texture map is quite simple since all images are in rectangular shape. The buildings Harjuviita 20-22 are also quite simple ones, and hence the use of rectangular shape textures is justified.

With more complex buildings, one has to generate more complicated textures to obtain a more accurate 3D model. Objects which are circles are somewhat more difficult to generate. The easiest way would be generating several narrow triangles which together form approximately the circle façade. The order of the texture images needs some optimization if they have different sizes or amount of vertices.

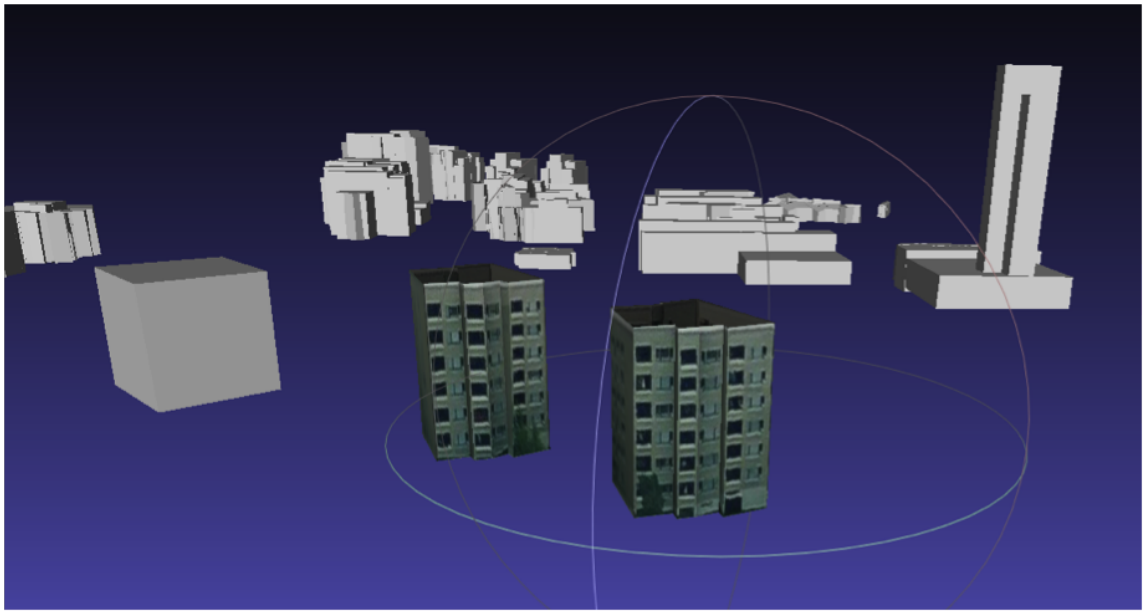

Figure 24: Two different buildings with textures on the façades and background buildings without any textures.

4.7 Other software.

Figure 25: The 3D model made with TerraPhoto, TerraScan and TerraModeler ©Tomi Rosnell, FGI 2014.

Figure 25 is made from the same Blom Pictometry images which were used in this thesis. This 3D model is made with a commercial software from Terrasolid.

TerraPhoto is used to process images and point clouds. "The software enables the production of rectified images and orthomosaics based on a ground model that has been extracted from the laser data. The positioning of the source images can be refined by using tie points for image to image adjustment while the ground control point can be involved for improving the absolute accuracy of the image block" (Terrasolid, 2014).

"TerraModeler creates surface models such as triangular irregular nets (TINs) from various sources, such as Lidar points stored in binary files or loaded in TerraScan, XYZ ascii files and graphical design elements. The software offers versatile visualization options including coloured shaded surfaces, contour lines, grids, coloured triangle nets, elevation texts, slope directions and textured surfaces (in combination with TerraPhoto)" (Terrasolid, 2014).

With the 3D model made with Terrasolid software, there are problems from the fully automatic program. Textures are somewhat as from right buildings but the size and position of textures are wrong. There can also be seen some removal of pixels which contain trees from a building texture.

Figure 26: Image capture from Nokia HERE maps 3D service ©HERE, DigitalGlobe (HERE, 2014).

The image capture from HERE maps 3D service in figure 26 is made from some other images and unknown techniques. Because there are no sharp edges, this 3D model looks like it would have been done only from images into TIN without any use of a point cloud made with Lidar.

Most of the preliminary work for gaining EO parameters manually was done with iWitness. The reason for this operation was that the image data and numerical data for the images were not completely reliable as it is described in section Results. iWitness needs IO for the cameras used. For the bundle adjustment, one has to measure tens of (the same object's) points from each image in the project.

After the points have measured and we are happy to have small enough RMS error, one has to bring some control points to the block. Control points put the camera's EO parameters in the same coordinate system as the buildings are in the point cloud. It is important to put the control points very carefully to the right position because those coordinates demand part of external orientation parameters for the camera 3D position. The purpose to use iWitness was to generate all external orientation parameters of rotation and position for each image separately.

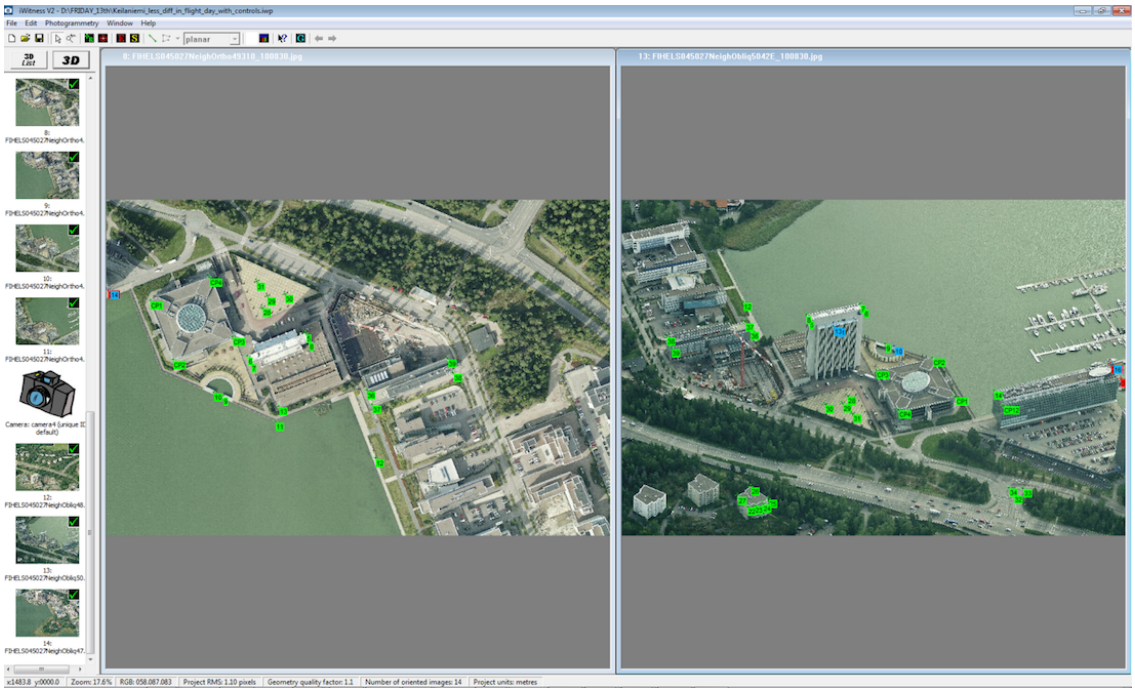

Figure 27: Control point measurements in iWitness project.

As it is seen in figure 27, there are not so many possibilities to find good control points which can be clearly separated from each other from the project images.

Other programs which were used for the comparison of the gained results were VisualSFM and SURE. The program name "VisualSFM" comes from a visual structure from motion system and SURE comes from a photogrammetric surface reconstruction from imagery. Since my computer is a normal desk machine and it was not possible to get more power or calculation time, I had to modify the size of the original images to only 15 per cent. This should still give some results for comparison with iWitness results since in VisualSFM it is possible to use much more images than iWitness because there

is much more automation than in iWitness.

When it was possible to use a somewhat more powerful computer to calculate points to the sift files and to compare automatically calculated points between all images, I was able to use original size jpeg images. The use of better quality images gave another problem since the VisualSFM program made several different point clouds instead of just one. This is not a big problem because those different point clouds can be put together if it is possible to find the corresponding points between the separate point clouds. Another way is to measure some control points for the images. Since this was not going to be fast and easily done, the VisualSFM program was left only at the earlier stage just for the curiosity of mind to try how it would work with the Pictometry images used.

After the results from the original data or from the parameters generated from iWitness were observed to not to be good enough, control points from all used data were measured in the VisualSFM project.

In figure 28, there is one point cloud generated by VisualSFM. In the same image there are also red squares which are control points, camera positions in the air and a table of transformation and its results.

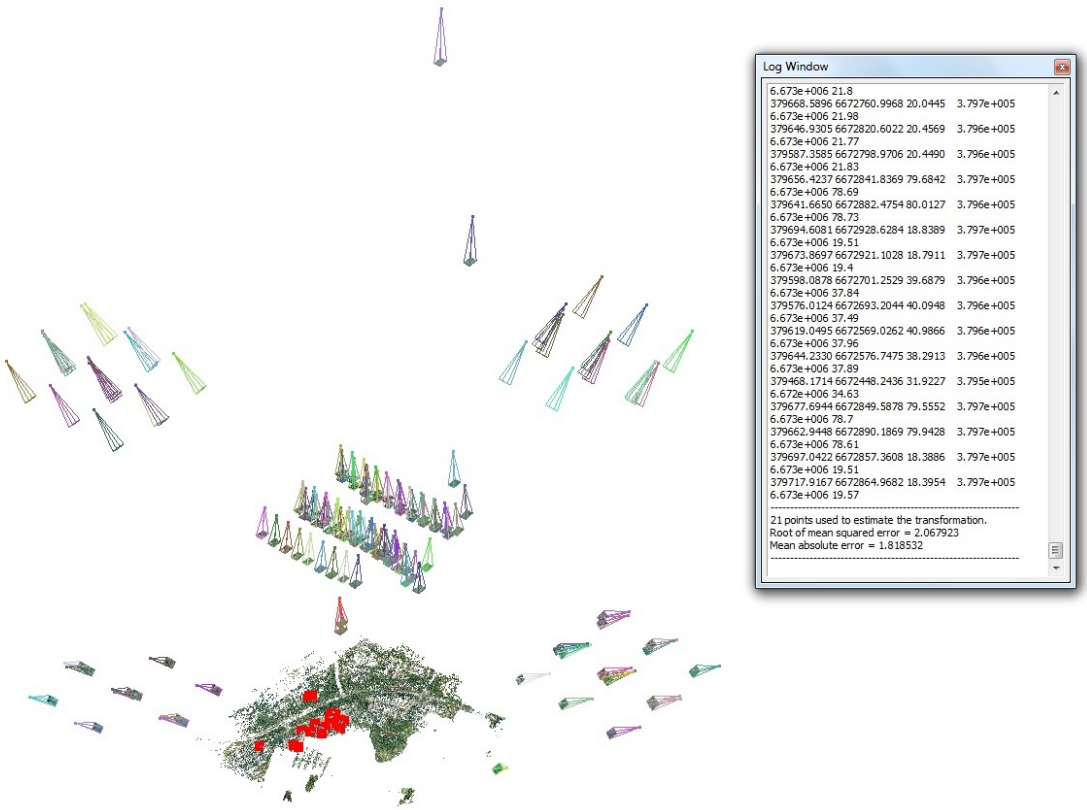

Figure 28: One point cloud of research area from VisualSFM.

The transformation was done with GCP-based Transform after the control points were

measured. As represented in the table in figure 28, there were 21 points used to estimate the transformation. The root of mean squared error (RMSE) is 2,067923m and mean absolute error (MAE) is 1,818532m.

Most of the positions of the cameras seem to be in a right place compared with other camera positions in four oblique and nadir direction. There are a couple of cameras in figure 28 which do not hold a logical position in the flight lines where the other cameras are situated. Some of the internal and all of the external orientation parameters can be found in the *.nvm file. Rotations are in quaternions but those can be transformed to normal rotation matrices. Instead using the saved *.nvm project it was run with SURE.

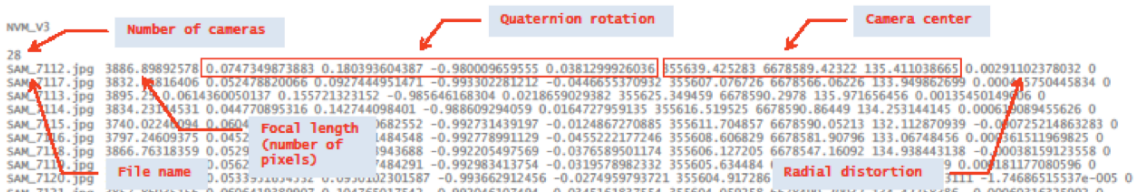

Figure 29: Specification of *.nvm file (Sandberg, 2013).

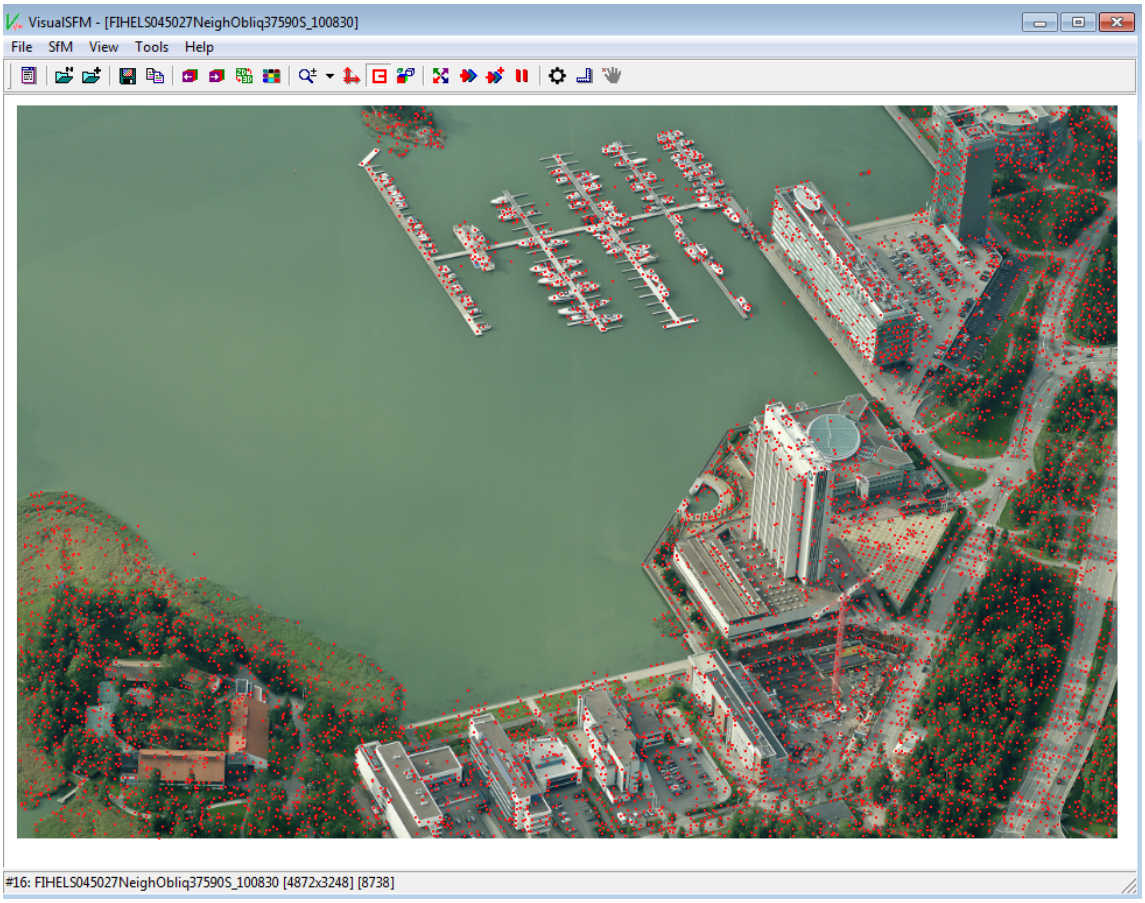

Figure 30: Example of point measurements with VisualSFM.

There are quite a lot of points measured from the images when VisualSFM software is used, as it can be seen in figure 30. The problem with a lot of sea points in the images is again very easy to see. About 50 per cent of the pixels in figure 30 are sea. The water level looks quite calm but still it would be wise not to use boats in the dock as equal measured points as the buildings are.

SURE is used for a making denser point cloud of the research area from the gained VisualSFM results. This point cloud is compared between ALS point cloud which is downloaded from NLS. The differences between two point clouds are investigated in section Results. In figure 31, there is a part of the RGB coloured point cloud processed by SURE and visualized by CloudCompare.

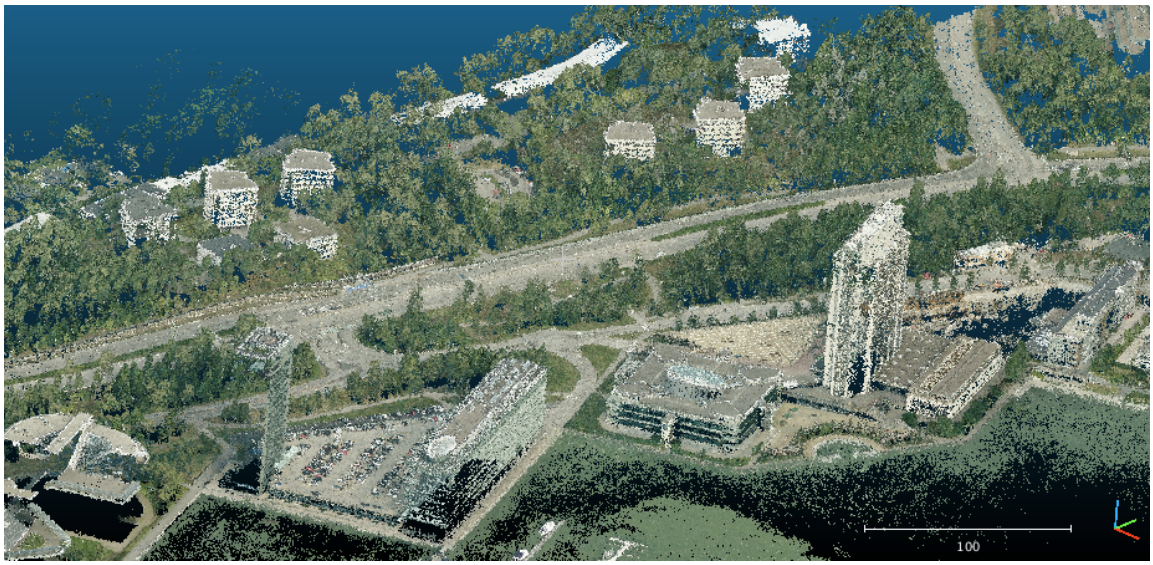

Figure 31: Point cloud produced with SURE.

SURE also makes *.ori files for each image used. In figure 32 there is an example of *.ori file for image: FIHELS045027NeighObliq5042E_100830.JPG. From the *.ori file can be found IO parameters (focal length, pixel size, sensor size, principal point, distortion) and EO parameters (rotation matrix and camera position). The pixel size is marked as one because any parameters were not given in the VisualSFM. No parameters were given in the VisualSFM since the idea was to generate as much parameters as possible without any prior knowledge of them. The used pixel size is 0,0074mm.

```
$ImageID_____________________________________________________(ORI_Ver_1.0)
          FIHELS045027NeighObliq5042E_100830
$IntOri_FocalLength__________________________________________________[mm]
         22908.41796880
$IntOri_PixelSize______(x|y)_________________________________________[mm]
              1.000000                  1.000000
$IntOri_SensorSize_____(x|y)_____________________________________[pixel]
                  4872                      3248
$IntOri_PrincipalPoint_(x|y)_____________________________________[pixel]
          2435.50000000           1623.50000000
$IntOri_CameraMatrix_____________________________(ImageCoordinateSystem)
         22908.41796880              0.00000000           2435.50000000
             0.00000000          22908.41796880           1623.50000000
             0.00000000              0.00000000              1.00000000
$ExtOri_RotationMatrix_____________________(World->ImageCoordinateSystem)
        -0.018729827610         -0.998852257495          0.044083571268
        -0.672886550374         -0.020018616145         -0.739474776672
         0.739508542130         -0.043513477288         -0.671739304650
$ExtOri_TranslationVector__________________________(WorldCoordinateSystem)
          377817.411496          6672930.635940             1684.698399
$IntOri_Distortion_____(Model|ParameterCount|(Parameters))_______________
           VSFM          1
        -0.019360428342
```

Figure 32: Example of *.ori file from SURE.

Also the softwares FugroViewer and CloudCompare have been used to investigate the quality of the SURE point cloud from the image data. The control points are chosen from NLS point cloud and tested with NLS internet software Karttapaikka (National Land Survey, 2013a). There is more discussion of the quality of the generated SURE point cloud in the next section Results.

The FGI's computer used in this thesis was Fujitsu Siemens, with Windows 7 Professional 64-bit operating system and 8GB RAM. The processor is Intel® Core™2 Duo CPU E8500 3.17GHz. MacBook OS X 10.9.5 with 4Gt RAM and 2 GHz Intel Core 2 Duo processor was also used for minor testing.

5 Results

In table 4 and 5, the difference in pixels in the images is shown when 3D coordinates from ALS are used. These tables are made from the images used in the building Harjuviita 22 (roof A) and Harjuviita 20 (roof B). Only the vertices from the roof are measured, because all vertices from the base are not as easily seen for measuring as the vertices are from the roof. The origin is in the left upper corner. The measurements of table 4 come from the original data. The measurements of Table 5 come from the data generated with SURE. The difference in the measurements is in the west direction.

Image	vertical [pix]	horizontal [pix]	dist. [pix]	angle [deg]	vertices	position
13	-25,75	-81,75	85,71	72,51	8	roof A
15	+10,00	+40,50	41,73	76,13	6	roof A
12	-27,00	-92,50	96,40	73,73	2	roof A
13	-27,13	-76,38	81,10	70,42	8	roof B
15	+7,17	+32,33	33,12	77,51	6	roof B
12	-23,50	-89,00	92,05	75,20	2	roof B

Table 4: Average values of the difference, when the used resolution value is 0.01 metre and the original data is used.

Image	vertical [pix]	horizontal [pix]	dist. [pix]	angle [deg]	vertices	position
13	-4,75	+10,88	11,89	66,27	8	roof A
15	+14,83	+14,33	20,70	43,93	6	roof A
12	+7,50	+1,00	7,65	9,22	2	roof A
13	-6,25	+12,00	13,71	62,64	8	roof B
15	+13,00	+21,50	25,18	59,03	6	roof B
12	+15,50	+4,50	16,15	16,24	2	roof B

Table 5: Average values of the difference, when the used resolution value is 0.01 metre and SURE data is used.

With the original data, this means the following. In the image number 13, the measurements are too much up and on the left; in the image number 15, the measurements are too much down and on the right, and in the image number 12, the measurements are too much up and on the left.

In figure 33, there is an example where red and yellow circles represent the measured image points from ALS data and light blue circles represent the vertices where the measured ones should be. In figure 34, there are the same circles as earlier but the used data set comes from SURE instead of using the original data set as in figure 33.

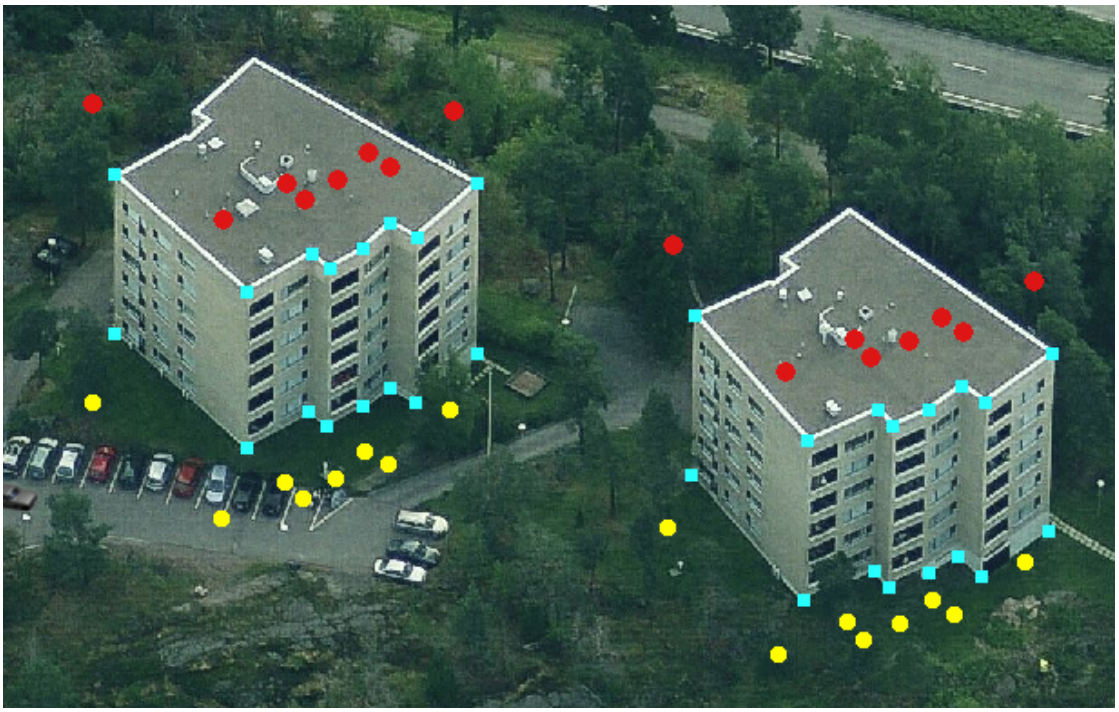

Figure 33: How the difference affects ALS to image with original data.

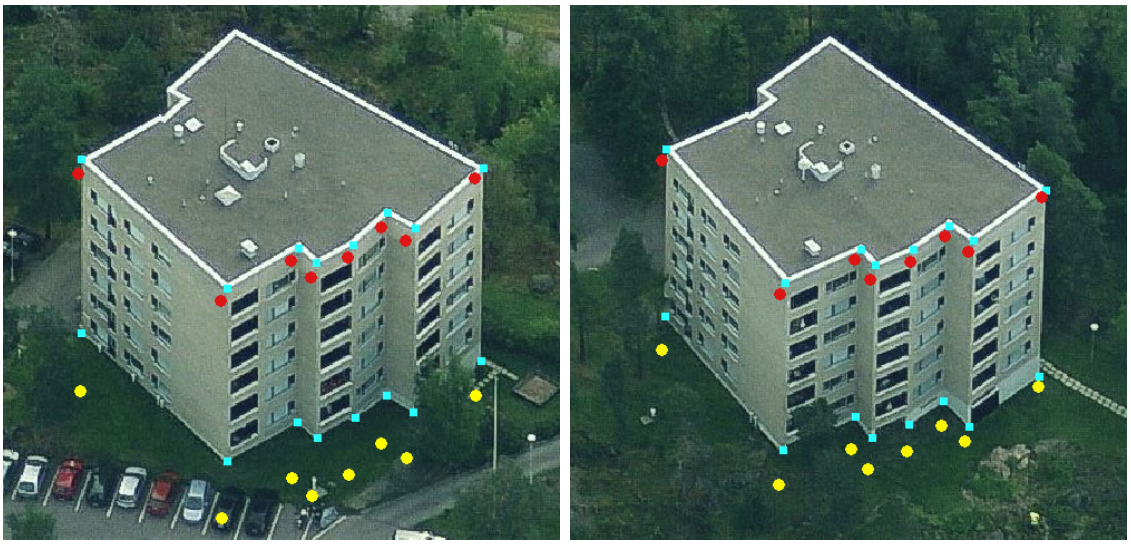

Figure 34: How the difference affects ALS to image with SURE data.

In table 6 and table 7, there are difference in metres when 3D coordinates are measured from two different images with the collinearity equation (27). These tables are made out from the figures used in the building Harjuviita 20 but the same images are also used in the building Harjuviita 22.

East [m]	North [m]	height [m]	images used	vertices
47,90	2,54	-48,83	5 and 13	8
21,86	230,67	184,31	3 and 15	2
-68,05	3,13	-56,62	12 and 14	2

Table 6: Average values of the difference in the 3D coordinates.

East [m]	North [m]	height [m]	images used	vertices
1,06	1,31	-2,21	5 and 13	8
-0,14	2,04	0,61	3 and 15	6
-0,89	0,94	-1,00	12 and 14	2

Table 7: Average values of the difference in the 3D coordinates, when SURE orientation is used.

In the next figure 35, it can be seen how much these differences affect the wanted façade orthophoto. In the leftmost image there is the original orthophoto of building façade which cannot be used to reconstruct a textured 3D model of the building. In the middle, there is a manually made version of the original orthophoto with corrections to the vertical displacement. On the right hand side there is an orthophoto made with the orientation parameters from SURE.

Figure 35: Illustration how the difference affects the orthophoto of a certain façade.

Textured 3D models of two buildings with fixed parameters and with SURE parameters were made. In figure 24, the textured 3D models of two buildings with fixed parameters are presented. In figure 36, the textured 3D models of two buildings with SURE parameters are presented.

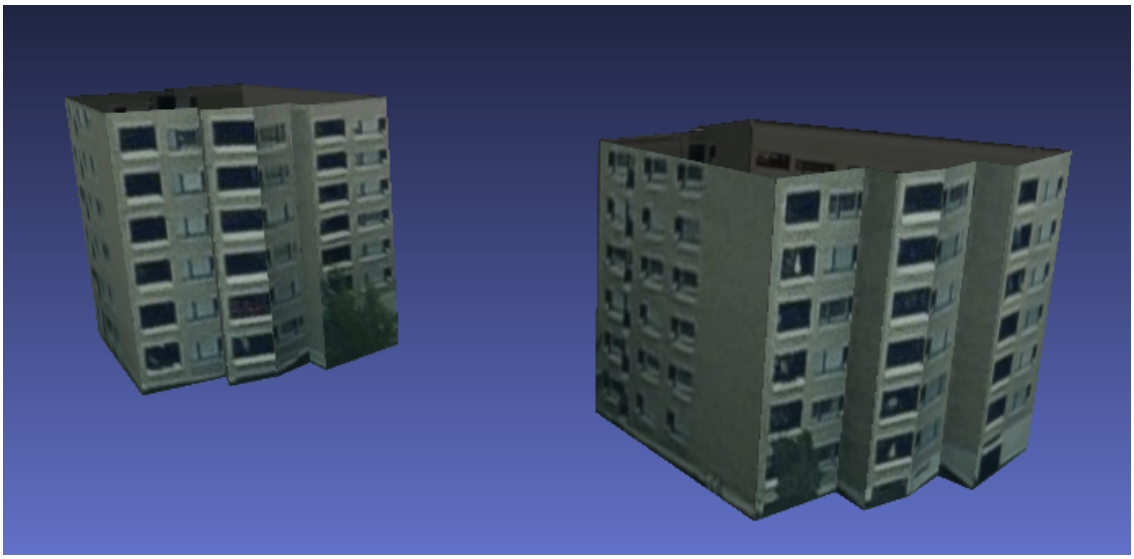

Figure 36: Two different buildings with textures on the façades.

In the basic Blom data sets parameters for both internal and external orientation were given. The use of the given data was not enough for producing the desired automation for texturing 3D models of the buildings with Matlab. The texturing of the 3D models using georeferenced images was nonetheless accomplished with a couple of simple buildings in the research area. To achieve this from the original data, one had to perform some manual transformations for the coordinates used, to measure the vertices of a certain façade of a certain building. The basic principle of the methods used in such kind of a process can be derived from very basic knowledge in photogrammetry.

Two different programs have also been used to calculate the external orientation of the images: iWitness and VisualSFM with SURE. There was not enough time and images of good quality to get a strong bundle with iWitness and therefore good orientations. VisualSFM with SURE make much better results even if the point cloud generated by SURE has some differences compared with the NLS point cloud of the area.

In figure 37, colourized difference in distance between the NLS point cloud and the SURE point cloud from blue to red can be seen. The most accurate area is in the centre of the amount of used images. Also most of the used control points are in this area. The points which are red come from the construction area which was not build in 2010 when the images were taken, but it was ready when NLS had a flight with their ALS device over the area. Approximately 95 per cent of the whole SURE point cloud is less than 4.6 metres from the NLS data. This means that the orientation files (*.ori) can be used to produce texturing for 3D models with minor errors. The used buildings Harjuviita 20 and Harjuviita 22 are in the blue centre area.

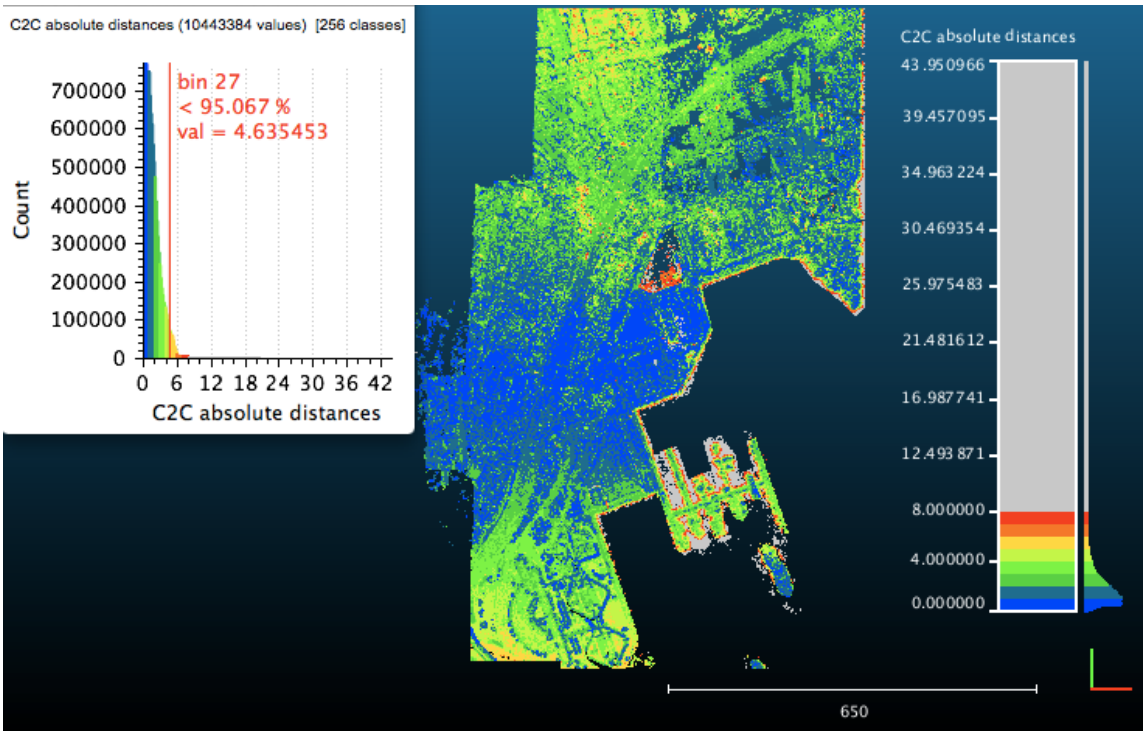

Figure 37: Absolute difference of the NLS point cloud and the SURE result.

In figure 38, there is a three colourized point cloud from the building Harjuviita 22. Most of the points which come from the trees are taken away and only points from the building and ground are left. There are 33788 points in this subset point cloud and most of them are on the ground. The vertices of the building's roof are also drawn as well as a possible base at 12 metres from the sea level.

The points which are less accurate come from the building's walls because there are less images for the walls than roofs. In the east direction, 52 per cent of the points are less than 0.26 metres, in the north direction, 51 per cent are less than 0.02 metres and in the height direction, 51 per cent are less than 0.56 metres from the NLS reference point cloud.

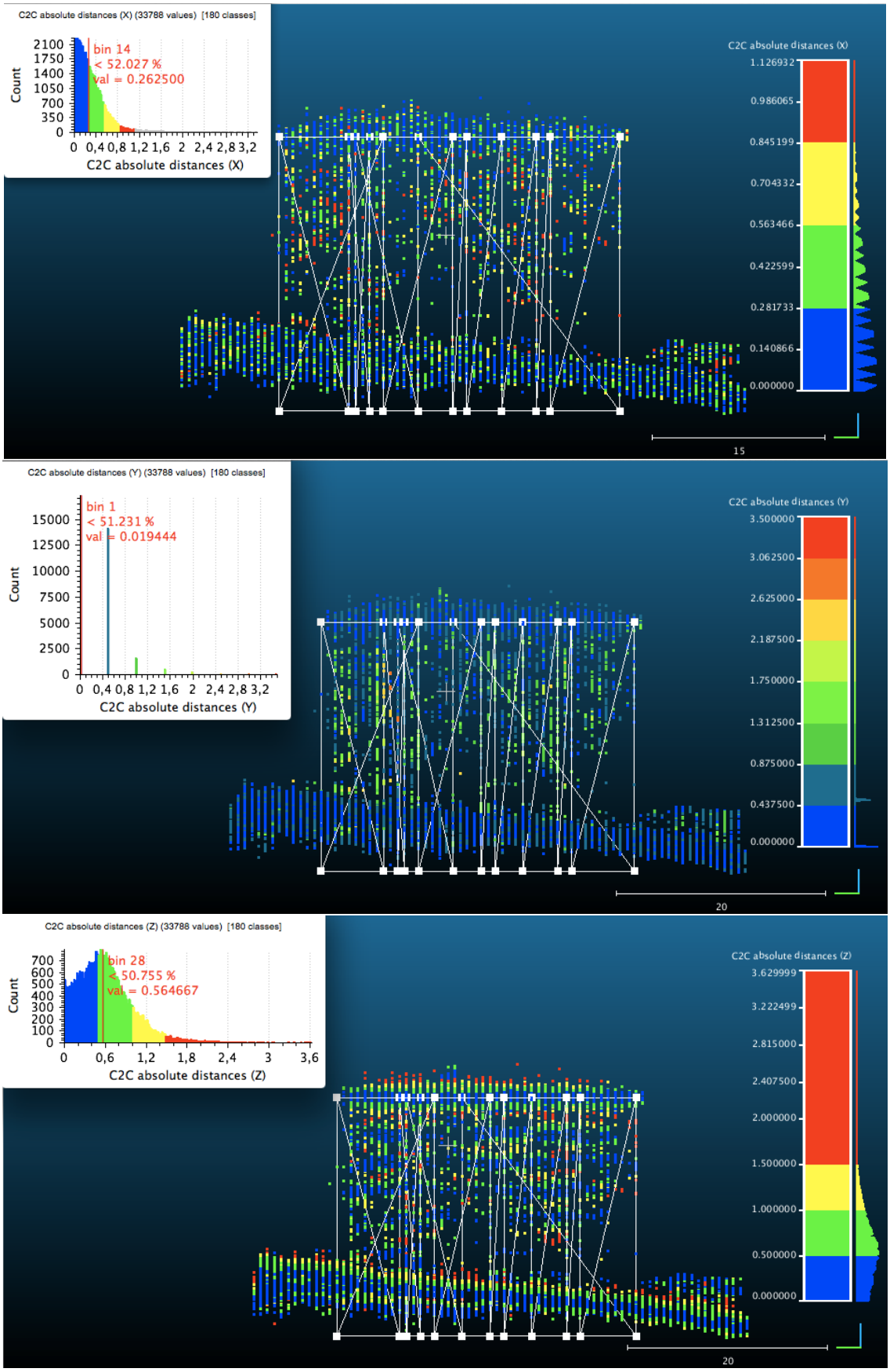

Figure 38: Absolute differences of the NLS point cloud and the SURE results.

6 Conclusion

The purpose of this thesis was to consider automated texturing of 3D models by using georeferenced airborne images made with the Pictometry technique. As an exemplary application, a 3D building model was generated. The aim of the study was to texture the building models based on photogrammetry knowledge, to be realized using Matlab software.

In the original data sets, parameters for both internal and external orientation were given. The use of the given data was not enough for producing the desired automation for texturing of 3D models of the buildings with Matlab. The texturing using georeferenced images was nonetheless accomplished with a couple of simple buildings in the research area. To achieve this from the original data, one had to perform some manual transformations for the coordinates used, to measure the vertices of a certain façade of a certain building. The basic principle of the methods used in such kind of a process can be derived from very basic knowledge in photogrammetry.

In addition to Pictometry technique, two different programs were also used to calculate the external orientation of the images: iWitness and VisualSFM with SURE. There were not enough images of good quality to get a strong bundle with iWitness and therefore good orientations. VisualSFM with SURE made much better results even if the point cloud generated by SURE had some differences compared with the NLS point cloud of the area.

Approximately 95 per cent of the whole SURE point cloud was less than 4.6 metres from the NLS data. This means that the orientation files (*.ori) could be used to produce texturing for 3D models with minor errors. When the image points were measured from 2D to 3D with the inverse collinearity equation from the SURE data, the absolute difference between measurements and the real position was from 0.14 to 1.06 metres in the east coordinates, from 0.94 to 2.04 metres in the north coordinates and from 0.61 to 2.21 metres in the height. The average distance was 2.18 metres. When the object coordinates were measured from 3D to 2D with the collinearity equation from the SURE data, the absolute difference between measurements and the real position was from 4.75 to 15.50 pixels in the vertical direction and from 1.00 to 21.50 pixels in a horizontal direction. The average distance was 15.81 pixels.

It was possible to make 3D models of buildings with texture from aerial oblique images. When the camera systems become more developed the texturing quality will be better. In the future, there will be developed for example better camera systems and more competition for Pictometry in the oblique imaging technology. For instance, Multi-cameras Integrated Digital Acquisition System MIDAS is one possible camera system for the oblique image tasks. The study of the texture map and optimization was not consummate in this thesis. In the future, I should investigate more of the optimization and how to make the most efficient texturing for the task.

References

Brown, D. C. (1971). Close-range camera calibration. *PHOTOGRAMMETRIC ENGINEERING*, 37(8):855–866. Reference date 08.10.2014. Available from: http://www.vision.caltech.edu/bouguetj/calib_doc/papers/Brown71.pdf.

City of Espoo (2011). Espoon muunnokset N60->N2000 ja VVJ->ETRS-GK25. Technical report, City of Espoo, Espoon kaupunki, kaupunkimittausyksikkö, PL 41 02070 Espoon kaupunki. Reference date 09.09.2014. Available from: http://www.espoo.fi/download/noname/%7B6D1E1066-4702-4C06-AB27-6F2C267AC847%7D/31745.

DeChant Consulting Services - DCS Inc. (2014). Frequently Asked Questions. Reference date 09.09.2014. Available from: http://www.iwitnessphoto.com/iwitness/faqs.html#iwitness.

Floréen, P. (2013). 58131 Tietorakenteet ja algoritmit. Reference date 16.09.2014. Available from: www.cs.helsinki.fi/u/floreen/tira2013/tira.pdf.

Fotogrammetrian ja Kaukokartoituksen Seura ry (1995). Suositukset Suomessa tehtävälle mittaus- ja kartoitusilmakuvaukselle. Technical report, Fotogrammetrian ja Kaukokartoituksen Seura ry. Reference date 08.10.2014. Available from: http://foto.hut.fi/seura/julkaisut/erillisjulkaisu1_1995/teksti.html#4.

Fraser, C. S. (1997). Digital camera self-calibration. *ISPRS Journal of Photogrammetry and Remote Sensing*, 52(8):149–159. Reference date 08.10.2014. Available from: http://ac.els-cdn.com.libproxy.aalto.fi/S0924271697000051/1-s2.0-S0924271697000051-main.pdf?_tid=53651708-c546-11e3-92f6-00000aacb35d&acdnat=1397639338_0913223f3b5473e2f7e5b1c68698f06e.

Fraser, C. S. and Edmundson, K. L. (2000). Design and Implementation of a Computational Processing System for Off-Line Digital Close-Range Photogrammetry. *ISPRS Journal of Photogrammetry & Remote Sensing*, 55(2):94–104. Reference date 04.02.2015. Available from: http://www.photometrix.com.au/downloads/australis/Australis-ISPRSJ2000.pdf.

Haala, N. and Kada, M. (2010). An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing 65.*, 65. Available from: http://testcis.cis.rit.edu/~cnspci/references/dip/urban_extraction/haala2010.pdf.

Hamruni, A. M. (2010). *The use of oblique and vertical images for 3D urban modelling.* PhD thesis, The University of Nottingham, Institute of Engineering Surveying and Space Geodesy. Reference date 18.09.2014. Available from: http://eprints.nottingham.ac.uk/11551/1/Ahmed_thesis.pdf.

HERE (2014). HERE maps 3D service. Reference date 16.09.2014. Available from: http://here.com/60.175799,24.828026,18.9,76,64,3d.day.

Honkavaara, E. and Haggrén, H. (2002). Maa-57.220 fotogrammetrinen kartoitus: Ortokuvien tuottaminen. Reference date 08.10.2014. Available from: http://foto.hut.fi/opetus/220/luennot/9/L9_2004.pdf.

Höhle, J. (2008). Photogrammetric Measuriments in Oblique Aerial Images. *Photogrammetrie, Fernerkundung, Geoinformation*, 1(3):8–9. Available from: http://vbn.aau.dk/ws/files/17874670/Photogr_Meas.pdf.

Jokinen, O. (2013). Maa-57.3120 analytical photogrammetry 1: Space intersection. Reference date 12.09.2014. Available from: http://foto.hut.fi/opetus/220/luennot/9/L9_2004.pdf.

Kaartinen, H., Hyyppä, J., Gülch, E., Vosselman, G., Hyyppä, H., Matikainen, L., Hofmann, A., Mäder, U., Persson, Å., Söderman, U., Elmqvist, M., Ruiz, A., Dragoja, M., Flamanc, D., Maillet, G., Kersten, T., Carl, J., Hau, R., Wild, E., Frederiksen, L., Holmgaard, J., and Vester, K. (2005a). Accuracy of 3D city models: EuroSDR comparison. *ISPRS WG III/3, III/4, V/3 Workshop "Laser scanning 2005" Enschede, the Netherlands.* Reference date 10.09.2014. Available from: http://www.icc.cat/layout/set/print/content/download/3751/12310/file/accuracy_3d_city_models.pdf.

Kaartinen, H., Hyyppä, J., Gülch, E., Vosselman, G., Hyyppä, H., Matikainen, L., Hofmann, A., Mäder, U., Persson, Å., Söderman, U., Elmqvist, M., Ruiz, A., Dragoja, M., Flamanc, D., Maillet, G., Kersten, T., Carl, J., Hau, R., Wild, E., Frederiksen, L., Holmgaard, J., and Vester, K. (2005b). EuroSDR building extraction comparison. *ISPRS Workshop "High-resolution Earth Imaging Geospatial Information". Hannover.* Reference date 10.09.2014. Available from: http://www.cccartografica.cat/espl/content/download/3754/12323/file/eurosdr_building_extraction_comparison.pdf.

Kalinski, A. (2013). Oblique Imagery: The New Kids on the Block. Reference date 05.02.2015. Available from: http://geospatial-solutions.com/oblique-imagery-the-new-kids-on-the-block/.

Kirby, B. J. (2009). Micro- and Nanoscale Fluid Mechanics: Transport in Microfluidic Devices. Cornell University. Reference date 06.02.2015. Available from: http://www.kirbyresearch.com/index.cfm/wrap/textbook/microfluidicsnanofluidicsse127.html.

Kraus, K. (1993). *Photogrammentry: Fundamentals and Standard Processes.*, volume 1. UMMLER/BONN.

Kraus, K. (1997). *Photogrammentry: Advanced Methods and Applications.*, volume 2. UMMLER/BONN. ISBN 3-427-78694-3.

Lemmens, M., Lemmen, C., and Wubbe, M. (2007). Pictometry: Potentials for Land Administration. 6th FIG Regional Conference in San José, Costa Rica 12-15 November 2007. Available from: https://www.fig.net/pub/monthly_articles/may_2008/may_2008_lemmens_lemmen_wubbe.pdf.

Longley, P. A., Goodchild, M. F., Maguire, D. J., and Rhind., D. W. (2005). *Geographical information systems and science.* John Wiley & Sons, Ltd. ISBN 0-470-87000-1. Available from: https://teemu-tomcat.linneanet.fi/vwebv/holdingsInfo?bibId=394891.

Mikhail, E. M., Bethel, J. S., and McGlone, J. C. (2001). *Introduction to Modern Photogrammetry*. John Wiley & Sons, Inc. ISBN 0-471-30924-9.

National Land Survey (2003). Kaavoitusmittausohjeet. Technical report, National Land Survey. Reference date 08.10.2014. Available from: `http://www.maanmittauslaitos.fi/sites/default/files/kaavoitusmittausohjeet_2003_0.pdf`.

National Land Survey (2005). UMT -lehtijako. Reference date 09.09.2014. Available from: `http://www.maanmittauslaitos.fi/sites/default/files/UTM_lehtijakopdf.pdf`.

National Land Survey (2010). Open source map window. Reference date 09.09.2014. Available from: `http://www.paikkatietoikkuna.fi/web/fi/kartta`.

National Land Survey (2013a). NLS karttapaikka. Reference date 09.09.2014. Available from: `http://kansalaisen.karttapaikka.fi`.

National Land Survey (2013b). NLS peruskartta. Reference date 09.09.2014. Available from: `http://www.maanmittauslaitos.fi/en/maps/map-products/printed-maps-and-prints/basic-map`.

Petrie, G. (2008). Systematic Oblique Aerial Photography Using Multiple Digital Cameras. VIII International Scientific & Technical Conference "From Imagery to Map: Digital Photogrammetric Technologies" September 15-18, 2008 – Porec, Croatia. Reference date 05.02.2015. Available from: `http://www.petriefied.info/Petrie_Croatia_Multiple_Oblique_Camera_Systems2.pdf`.

Petrie, G. (2009). Modular Cameras; Multiple Configurations The IGI DigiCAM Range. Technical Report 6. Reference date 08.10.2014. Available from: `http://web2.ges.gla.ac.uk/~gpetrie/Petrie_IGI_DigiCAM_Range_GEO_6_2009.pdf`.

Pihlajamäki, T. and Pekonen, J. (2011). aaltologo.sty, Package for drawing Aalto University logos. Reference date 09.08.2014. Available from: `https://wiki.aalto.fi/display/aaltolatex/`.

Ramesh, J., Rangachar, K., and Schunk., B. G. (1995). *Machine vision*. MIT Press and McGraw-Hill, Inc. ISBN 0-07-032018-7.

Rothermel, M., Wenzel, K., Fritsch, D., and Haala, N. (2012). SURE: Photogrammetric Surface Reconstruction from Imagery. *Proceedings LC3D Workshop Berlin*. Reference date 09.10.2014. Available from: `http://www.ifp.uni-stuttgart.de/publications/2012/Rothermel_etal_lc3d.pdf`.

Rönnholm, P. (2002). Special Assignments: Kiertomatriisi. Technical report, Institute of Photogrammetry and Remote Sensing. Reference date 15.09.2014. Available from: `http://foto.hut.fi/opetus/290/julkaisut/pronnhol/kiertomatriisi.pdf`.

Rönnholm, P. (2008). Information about iwitness rotation matrix and rotations. Reference date 15.09.2014.

Sandberg, M. (2013). VisualSFM-Socet set-SURE. Reference date 08.10.2014.

Schults, S. L., Giuffrida, F. D., Gray, R. L., and Mondello., C. (2008). Method and apparatus for capturing, geolocating and measuring oblique images. Reference date 08.10.2014. Available from: http://www.freepatentsonline.com/7424133.pdf.

Suoranta, S., Kanerva, P., Karppinen, L., Staffans, L., Takkinen, L., Larja, J., Sorvakko, S., and numerous unknown authors (2011). aalto-thesis.sty, one latex template for master's thesis. Reference date 09.08.2014. Available from: https://into.aalto.fi/download/attachments/6301921/thesistemplate42.zip?version=3&modificationDate=1386680946000&api=v2.

Terrasolid (2014). Terrasolid products. Reference date 16.09.2014. Available from: http://www.terrasolid.com/products.php.

The MathWorks, Inc (1994). Graphical input from mouse or cursor. Reference date 09.09.2014. Available from: http://www.mathworks.se/help/matlab/ref/ginput.html.

Varshosaz, M. (2003). True realistic 3D models of buildings in urban areas. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIV-5/W10. Reference date 17.09.2014. Available from: http://www.isprs.org/proceedings/XXXIV/5-W10/papers/varshosa.pdf.

Wang, Y., Shultz, S., and Giuffrida, F. (2008). Pictometry's Proprietary airborne digital imaging system and its application in 3D city modelling. *The international Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.*, XXXVII part B1. Reference date 16.09.2014. Available from: http://www.isprs.org/proceedings/XXXVII/congress/1_pdf/182.pdf.

Wu, C. (2011). VisualSFM: A Visual Structure from Motion System. Technical report. Reference date 18.11.2014. Available from: http://ccwu.me/vsfm/.

Wu, C., Agarwal, S., Curless, B., and Seitz, S. M. (2011). Multicore Bundle Adjustment. Technical report, University of Washington. Reference date 18.11.2014. Available from: CVPR2011,http://ccwu.me/vsfm/.

Figure 39: Images from Keilaniemi.

Australis Bundle Adjustment Results: Camera Parameters

11 September, 2009 11:30:13

Project: \\Armor\calibrations\Penta Cameras\16MP Cameras Ready for Installation\
7655_101763_120mm_6356\7655_101763_120mm_6356.aus

Adjustment: Free-Network
Number of Points: 137
Number of Images: 15
RMS of Image coords: 0.33 (um)

Results for Camera 1 Pictometry16 120mm Lens

Sensor Size Pixel Size (mm)
H 4872 0.007
V 3248 0.007

Camera Variable	Initial Value	Total Adjustment	Final Value	Initial Std. Error	Final Std. Error
C	119.4718	-0.00395	119.4679	1.0e+003	1.021e-002 (mm)
XP	0.2338	-0.00407	0.2297	1.0e+003	2.783e-003 (mm)
YP	0.0877	0.04222	0.1299	1.0e+003	3.333e-003 (mm)
K1	4.52353e-007	-5.561e-007	-1.03752e-007	1.0e+003	1.394e-007
K2	-4.83994e-011	3.035e-009	2.98656e-009	1.0e+003	7.251e-010
K3	2.82635e-012	-4.486e-012	-1.65950e-012	1.0e+003	1.140e-012
P1	3.67866e-026	5.501e-026	9.17926e-026	1.0e-016	7.508e-017
P2	2.48581e-025	-6.610e-027	2.41970e-025	1.0e-016	7.508e-017
B1	3.58424e-026	1.525e-026	5.10954e-026	1.0e-016	7.508e-017
B2	3.25042e-026	9.620e-027	4.21239e-026	1.0e-016	7.508e-017

Maximum Observational Radial Distance Encountered: 20.9 mm

Exterior Orientation Summary (Xc, Yc, Zc are in project units, rotations are in decimal degrees)

Station	Image	Xc	Yc	Zc	Alpha	Elev.	Roll
1	Image001	-138.13125	-35.16287	380.28113	-75.592235	-67.337737	141.214226
2	Image002	39.46267	31.83133	420.55236	125.900917	-79.419050	-34.789713
3	Image003	39.69284	4.70874	420.42524	103.892100	-81.456954	-13.143401
4	Image004	53.40972	-28.23720	418.74328	80.677697	-80.144185	-81.294486
5	Image005	159.80807	-33.18545	382.11861	83.229566	-64.516704	-134.529281
6	Image006	138.28970	2.01868	391.55612	95.256909	-65.296236	-5.370362
7	Image007	144.07409	21.05867	388.81358	95.254133	-67.214793	56.137794
8	Image008	-137.40738	5.23356	379.35132	-98.619406	-65.741735	-172.558096
9	Image009	-131.36430	41.20348	381.44476	-107.185367	-65.873196	130.979587
10	Image010	-66.97337	31.95729	403.81103	-125.429746	-78.523356	-145.205170
11	Image011	-66.47060	4.79509	403.51337	-107.475712	-78.617651	-162.963081
12	Image012	-52.94122	-28.04621	406.40704	-72.533596	-83.886156	71.347053
13	Image013	-11.46670	31.85900	411.24243	-173.844805	-83.647508	-95.927447
14	Image014	-11.86192	5.50997	411.53343	174.498269	-86.115997	-84.261082
15	Image015	3.31203	-28.30345	412.14506	38.750959	-87.955003	-40.041557

Australis Bundle Adjustment Results: Camera Parameters

20 August, 2009 14:05:03

Project: \\Armor\calibrations\Penta Cameras\16MP Cameras Ready for Installation
\8822_103019_170mm_7972\8822_103019_170mm_7972.aus

Adjustment: Free-Network
Number of Points: 83
Number of Images: 15
RMS of Image coords: 0.42 (um)

Results for Camera 1 Pictometry16 170mm Lens

Sensor Size Pixel Size (mm)
H 4872 0.007
V 3248 0.007

Camera Variable	Initial Value	Total Adjustment	Final Value	Initial Std. Error	Final Std. Error
C	170.0000	-0.45464	169.5454	1.0e+003	4.420e-002 (mm)
XP	0.0000	0.02050	0.0205	1.0e+003	1.099e-002 (mm)
YP	0.0000	-0.15833	-0.1583	1.0e+003	1.293e-002 (mm)
K1	0.00000e+000	-1.577e-007	-1.57653e-007	1.0e+003	2.775e-007
K2	0.00000e+000	-1.153e-010	-1.15322e-010	1.0e+003	1.502e-009
K3	0.00000e+000	8.475e-013	8.47538e-013	1.0e+003	2.427e-012
P1	0.00000e+000	-6.429e-026	-6.42889e-026	1.0e-016	1.036e-016
P2	0.00000e+000	-1.737e-025	-1.73707e-025	1.0e-016	1.036e-016
B1	0.00000e+000	1.253e-026	1.25337e-026	1.0e-016	1.036e-016
B2	0.00000e+000	5.543e-027	5.54289e-027	1.0e-016	1.036e-016

Maximum Observational Radial Distance Encountered: 20.8 mm

Exterior Orientation Summary (Xc, Yc, Zc are in project units, rotations are in decimal degrees)

Station	Image	Xc	Yc	Zc	Alpha	Elev.	Roll
1	Image001	-137.48177	-32.85462	372.36846	-76.160142	-65.398254	130.477300
2	Image002	53.31190	31.19971	410.18119	109.518501	-77.700109	-19.447708
3	Image003	53.39731	5.72104	410.25130	89.105459	-78.499675	0.418740
4	Image004	65.57282	-27.88445	409.08091	70.280877	-76.569407	-72.348728
5	Image005	169.30650	-32.85506	369.71262	84.545480	-59.188295	-136.654107
6	Image006	150.87244	1.91568	379.45799	89.817935	-61.360779	7.342607
7	Image007	156.12820	18.49517	377.08472	97.640063	-62.761880	46.800631
8	Image008	-137.79612	1.66854	370.95752	-90.830455	-64.769739	179.626896
9	Image009	-134.00823	38.92104	372.52037	-102.606834	-64.159890	129.055837
10	Image010	-57.28568	30.68891	401.42677	-113.027297	-80.125992	-158.075506
11	Image011	-57.55842	4.83052	401.13399	-113.291656	-80.318680	-157.830614
12	Image012	-45.69412	-27.68699	404.04207	-55.090654	-81.596391	57.109197
13	Image013	-3.75009	31.14622	412.39505	168.322749	-85.668666	-79.008734
14	Image014	-3.97348	6.11816	412.57941	165.544816	-85.613403	-76.239835
15	Image015	8.78811	-28.03375	413.61670	17.385187	-85.447676	-15.708379

Australis Bundle Adjustment Results: Camera Parameters

05 June, 2008 16:19:50

Project: \\Armor\Calibrations\Penta Cameras\Simmons\16mp update 4\7933_102242_170mm_6360\7933_102242_170mm_6360.aus

Adjustment: Free-Network
Number of Points: 99
Number of Images: 15
RMS of Image coords: 0.48 (um)

Results for Camera 1 Pictometry16 170mm Lens

Sensor Size	Pixel Size (mm)
H 4872	0.007
V 3248	0.007

Camera Variable	Initial Value	Total Adjustment	Final Value	Initial Std. Error	Final Std. Error
C	170.0000	0.18493	170.1849	1.0e+003	4.330e-002 (mm)
XP	0.0000	-0.20540	-0.2054	1.0e+003	1.202e-002 (mm)
YP	0.0000	0.00847	0.0085	1.0e+003	1.385e-002 (mm)
K1	0.00000e+000	-5.274e-007	-5.27382e-007	1.0e+003	3.026e-007
K2	0.00000e+000	2.303e-009	2.30308e-009	1.0e+003	1.623e-009
K3	0.00000e+000	-2.562e-012	-2.56172e-012	1.0e+003	2.572e-012
P1	0.00000e+000	-8.304e-026	-8.30434e-026	1.0e-016	1.176e-016
P2	0.00000e+000	1.302e-026	1.30237e-026	1.0e-016	1.176e-016
B1	0.00000e+000	1.614e-026	1.61439e-026	1.0e-016	1.176e-016
B2	0.00000e+000	-1.694e-027	-1.69407e-027	1.0e-016	1.176e-016

Maximum Observational Radial Distance Encountered: 20.5 mm

Exterior Orientation Summary (Xc, Yc, Zc are in project units, rotations are in decimal degrees)

Station	Image	Xc	Yc	Zc	Alpha	Elev.	Roll
1	Image001	-149.76746	-35.95155	385.95735	-77.246666	-64.514704	149.986295
2	Image002	49.04537	31.77497	383.17026	106.166796	-78.054793	-17.526412
3	Image003	49.51398	2.99354	383.42156	88.704719	-78.324782	-2.824288
4	Image004	61.36276	-28.06295	382.56307	67.880154	-76.835385	-66.441876
5	Image005	149.89182	-34.09696	378.13031	84.100848	-62.618219	-140.825136
6	Image006	130.70759	2.11191	387.03105	89.443565	-64.688741	-0.069211
7	Image007	135.94559	19.43315	385.03049	100.174526	-66.161693	44.951252
8	Image008	-149.39715	0.46680	385.37831	-97.283779	-64.732386	-172.846985
9	Image009	-144.55234	36.51077	387.27741	-106.441213	-65.349716	128.015357
10	Image010	-62.15665	31.26266	404.97799	-106.131502	-80.032168	-163.941466
11	Image011	-62.12627	2.15844	404.93488	-106.789418	-80.255564	-163.271509
12	Image012	-50.18481	-27.61737	408.18831	-55.094317	-80.427689	55.942440
13	Image013	-8.24985	31.87709	411.57426	171.499598	-86.812506	-81.392016
14	Image014	-8.35125	1.46822	411.70943	174.791879	-86.854598	-84.560891
15	Image015	3.74219	-27.94482	413.10130	17.388171	-84.727213	-14.167314

Australis Bundle Adjustment Results: Camera Parameters

05 June, 2008 16:03:13

Project: \\Armor\Calibrations\Penta Cameras\Simmons\16mp update 4\7887_102139_170mm_6441\7887_102139_170mm_6441.aus

Adjustment: Free-Network
Number of Points: 92
Number of Images: 15
RMS of Image coords: 0.45 (um)

Results for Camera 1 Pictometry16 170mm Lens

Sensor Size	Pixel Size (mm)
H 4872	0.007
V 3248	0.007

Camera Variable	Initial Value	Total Adjustment	Final Value	Initial Std. Error	Final Std. Error
C	170.0000	-0.12983	169.8702	1.0e+003	4.398e-002 (mm)
XP	0.0000	-0.05731	-0.0573	1.0e+003	1.289e-002 (mm)
YP	0.0000	0.06099	0.0610	1.0e+003	1.328e-002 (mm)
K1	0.00000e+000	4.893e-007	4.89324e-007	1.0e+003	3.091e-007
K2	0.00000e+000	-3.337e-009	-3.33680e-009	1.0e+003	1.591e-009
K3	0.00000e+000	6.276e-012	6.27571e-012	1.0e+003	2.439e-012
P1	0.00000e+000	-7.893e-026	-7.89333e-026	1.0e-016	1.124e-016
P2	0.00000e+000	-3.847e-026	-3.84710e-026	1.0e-016	1.124e-016
B1	0.00000e+000	-3.427e-028	-3.42655e-028	1.0e-016	1.124e-016
B2	0.00000e+000	3.779e-027	3.77851e-027	1.0e-016	1.124e-016

Maximum Observational Radial Distance Encountered: 21.0 mm

Exterior Orientation Summary (Xc, Yc, Zc are in project units, rotations are in decimal degrees)

Station	Image	Xc	Yc	Zc	Alpha	Elev.	Roll
1	Image001	-146.84524	-36.17692	386.80741	-79.680603	-65.734668	151.281095
2	Image002	49.39522	29.91847	384.80493	97.492507	-78.041801	-5.714911
3	Image003	49.75817	0.20109	384.92398	97.107425	-78.189110	-5.467678
4	Image004	61.93519	-28.18870	383.60423	74.413751	-76.145515	-73.156638
5	Image005	154.18220	-35.39294	372.85924	84.566287	-60.805183	-144.885760
6	Image006	134.15481	1.59727	382.94329	91.793404	-63.899454	-0.989353
7	Image007	138.97598	18.95817	381.16730	99.035678	-64.621372	46.346447
8	Image008	-146.52948	-1.12272	386.24081	-93.141585	-65.586223	-176.563148
9	Image009	-139.38581	38.88970	388.58315	-108.292389	-64.933838	123.714299
10	Image010	-64.58041	29.45512	404.29864	-101.835788	-79.738590	-165.781057
11	Image011	-64.76568	-0.84499	404.44207	-101.183083	-79.966343	-167.201285
12	Image012	-53.67933	-27.75804	407.53312	-61.933374	-81.067082	65.223789
13	Image013	-2.60709	30.16472	393.09249	127.628125	-87.956005	-36.340855
14	Image014	-2.73085	1.15826	393.07176	128.129505	-88.119763	-36.802288
15	Image015	9.12469	-28.03641	393.76256	26.431323	-85.312540	-23.575803

Australis Bundle Adjustment Results: Camera Parameters

05 June, 2008 15:44:12

Project: \\Armor\Calibrations\Penta Cameras\Simmons\16mp update 4\7883_102136_170mm_4911\7883_102136_170mm_4911.aus

Adjustment: Free-Network
Number of Points: 97
Number of Images: 15
RMS of Image coords: 0.42 (um)

Results for Camera 1 Pictometry16 170mm Lens

Sensor Size	Pixel Size (mm)
H 4872	0.007
V 3248	0.007

Camera Variable	Initial Value	Total Adjustment	Final Value	Initial Std. Error	Final Std. Error
C	169.6026	-0.04488	169.5577	1.0e+003	3.932e-002 (mm)
XP	-0.0886	-0.04701	-0.1356	1.0e+003	9.898e-003 (mm)
YP	0.5153	-0.02981	0.4855	1.0e+003	1.196e-002 (mm)
K1	-5.90277e-007	5.651e-007	-2.52221e-008	1.0e+003	2.600e-007
K2	2.61399e-009	-4.250e-009	-1.63596e-009	1.0e+003	1.339e-009
K3	-3.65800e-012	7.479e-012	3.82063e-012	1.0e+003	2.053e-012
P1	-1.49912e-025	-3.129e-026	-1.81204e-025	1.0e-016	1.033e-016
P2	4.44725e-025	1.469e-025	5.91658e-025	1.0e-016	1.033e-016
B1	2.94274e-026	7.438e-027	3.68658e-026	1.0e-016	1.033e-016
B2	-9.24928e-027	-7.587e-027	-1.68365e-026	1.0e-016	1.033e-016

Maximum Observational Radial Distance Encountered: 20.7 mm

Exterior Orientation Summary (Xc, Yc, Zc are in project units, rotations are in decimal degrees)

Station	Image	Xc	Yc	Zc	Alpha	Elev.	Roll
1	Image001	-147.91560	-36.76150	386.71734	-79.383475	-64.943970	153.561460
2	Image002	49.15641	29.89393	386.79655	104.452279	-77.899344	-11.865780
3	Image003	49.42466	-0.59525	386.97201	105.115731	-78.206927	-11.945285
4	Image004	62.23777	-28.14429	385.81126	69.242337	-76.830262	-68.565815
5	Image005	150.08716	-33.72540	378.26893	80.808845	-62.264686	-136.511951
6	Image006	130.56809	1.06877	387.15890	95.445600	-64.266297	-4.653836
7	Image007	136.40016	18.10496	385.51069	104.027483	-64.674153	48.844064
8	Image008	-147.38433	2.13212	386.25002	-87.459270	-64.859698	177.624597
9	Image009	-141.71611	38.56838	388.09874	-107.596141	-65.204253	133.402043
10	Image010	-61.28331	30.32773	403.71536	-116.159299	-79.960911	-153.276417
11	Image011	-61.35396	0.92386	403.64017	-95.760957	-80.730392	-173.522770
12	Image012	-48.07216	-27.81984	406.52983	-54.207433	-82.777778	53.962178
13	Image013	-16.33751	30.71157	420.30617	-158.608359	-87.345296	-112.031421
14	Image014	-15.92323	2.49198	420.67580	-160.139746	-87.370852	-110.554601
15	Image015	-2.51050	-28.03348	422.06089	6.475307	-86.023758	-6.841605

```
# From Masters Thesis function made by Mika Kekäläinen 27.8.2014, FGI.

mtllib Harjuviita22_from_matlab.mtl
v 379438.734 6672931.001 36.630
v 379432.522 6672936.797 36.630
v 379434.242 6672938.674 36.630
v 379431.386 6672941.355 36.630
v 379429.370 6672944.458 36.630
v 379431.457 6672945.771 36.630
v 379427.523 6672951.533 36.630
v 379442.078 6672961.117 36.630
v 379446.210 6672954.896 36.630
v 379447.988 6672953.192 36.630
v 379449.022 6672954.068 36.630
v 379454.960 6672948.671 36.630
v 379438.734 6672931.001 2.850
v 379432.522 6672936.797 2.850
v 379434.242 6672938.674 2.850
v 379431.386 6672941.355 2.850
v 379429.370 6672944.458 2.850
v 379431.457 6672945.771 2.850
v 379427.523 6672951.533 2.850
v 379442.078 6672961.117 2.850
v 379446.210 6672954.896 2.850
v 379447.988 6672953.192 2.850
v 379449.022 6672954.068 2.850
v 379454.960 6672948.671 2.850
#24

vt 0.0 1.0
vt 0.0 0.0
vt 1.0 0.0
vt 1.0 1.0
# "#f..." is a mirror image version.
# change "#" for the upper "f" to get mirror texture in use.

usemtl Harjuviita22_1
#f   1/4  2/3 14/2 13/1
f   1/3  2/4 14/1 13/2
usemtl Harjuviita22_2
f   2/4  3/3 15/2 14/1
#f   2/3  3/4 15/1 14/2
usemtl Harjuviita22_3
```

```
f  3/4  4/3 16/2 15/1
#f  3/3  4/4 16/1 15/2
usemtl Harjuviita22_4
f  4/4  5/3 17/2 16/1
#f  4/3  5/4 17/1 16/2
usemtl Harjuviita22_5
f  5/4  6/3 18/2 17/1
#f  5/3  6/4 18/1 17/2
usemtl Harjuviita22_6
f  6/4  7/3 19/2 18/1
#f  6/3  7/4 19/1 18/2
usemtl Harjuviita22_7
f  7/4  8/3 20/2 19/1
#f  7/3  8/4 20/1 19/2
usemtl Harjuviita22_8
#f  8/4  9/3 21/2 20/1
f  8/3  9/4 21/1 20/2
usemtl Harjuviita22_9
f  9/4 10/3 22/2 21/1
#f  9/3 10/4 22/1 21/2
usemtl Harjuviita22_10
f 10/4 11/3 23/2 22/1
#f 10/3 11/4 23/1 22/2
usemtl Harjuviita22_11
f 11/4 12/3 24/2 23/1
#f 11/3 12/4 24/1 23/2
usemtl Harjuviita22_12
#f 12/4  1/3 13/2 24/1
f 12/3  1/4 13/1 24/2
#12
```

```
newmtl Harjuviita22_1
Ka 0.900000 0.900000 0.900000
Kd 0.905882 0.905882 0.905882
Ks 0.000000 0.000000 0.000000
Tr 0.200000
illum 2
Ns 0.000000
map_Kd D:\Week_35\final_harjuviitat\build1\Harjuviita22_1_rez.jpg

newmtl Harjuviita22_2
Ka 0.900000 0.900000 0.900000
Kd 0.905882 0.905882 0.905882
Ks 0.000000 0.000000 0.000000
Tr 0.200000
illum 2
Ns 0.000000
map_Kd D:\Week_35\final_harjuviitat\build1\Harjuviita22_2_rez.jpg

newmtl Harjuviita22_3
Ka 0.900000 0.900000 0.900000
Kd 0.905882 0.905882 0.905882
Ks 0.000000 0.000000 0.000000
Tr 0.200000
illum 2
Ns 0.000000
map_Kd D:\Week_35\final_harjuviitat\build1\Harjuviita22_3_rez.jpg

newmtl Harjuviita22_4
Ka 0.900000 0.900000 0.900000
Kd 0.905882 0.905882 0.905882
Ks 0.000000 0.000000 0.000000
Tr 0.200000
illum 2
Ns 0.000000
map_Kd D:\Week_35\final_harjuviitat\build1\Harjuviita22_4_rez.jpg

newmtl Harjuviita22_5
Ka 0.900000 0.900000 0.900000
Kd 0.905882 0.905882 0.905882
Ks 0.000000 0.000000 0.000000
Tr 0.200000
illum 2
Ns 0.000000
map_Kd D:\Week_35\final_harjuviitat\build1\Harjuviita22_5_rez.jpg

newmtl Harjuviita22_6
Ka 0.900000 0.900000 0.900000
Kd 0.905882 0.905882 0.905882
Ks 0.000000 0.000000 0.000000
Tr 0.200000
illum 2
Ns 0.000000
map_Kd D:\Week_35\final_harjuviitat\build1\Harjuviita22_6_rez.jpg
```

```
newmtl Harjuviita22_7
Ka 0.900000 0.900000 0.900000
Kd 0.905882 0.905882 0.905882
Ks 0.000000 0.000000 0.000000
Tr 0.200000
illum 2
Ns 0.000000
map_Kd D:\Week_35\final_harjuviitat\build1\Harjuviita22_7_rez.jpg

newmtl Harjuviita22_8
Ka 0.900000 0.900000 0.900000
Kd 0.905882 0.905882 0.905882
Ks 0.000000 0.000000 0.000000
Tr 0.200000
illum 2
Ns 0.000000
map_Kd D:\Week_35\final_harjuviitat\build1\Harjuviita22_8_rez.jpg

newmtl Harjuviita22_9
Ka 0.900000 0.900000 0.900000
Kd 0.905882 0.905882 0.905882
Ks 0.000000 0.000000 0.000000
Tr 0.200000
illum 2
Ns 0.000000
map_Kd D:\Week_35\final_harjuviitat\build1\Harjuviita22_9_rez.jpg

newmtl Harjuviita22_10
Ka 0.900000 0.900000 0.900000
Kd 0.905882 0.905882 0.905882
Ks 0.000000 0.000000 0.000000
Tr 0.200000
illum 2
Ns 0.000000
map_Kd D:\Week_35\final_harjuviitat\build1\Harjuviita22_10_rez.jpg

newmtl Harjuviita22_11
Ka 0.900000 0.900000 0.900000
Kd 0.905882 0.905882 0.905882
Ks 0.000000 0.000000 0.000000
Tr 0.200000
illum 2
Ns 0.000000
map_Kd D:\Week_35\final_harjuviitat\build1\Harjuviita22_11_rez.jpg

newmtl Harjuviita22_12
Ka 0.900000 0.900000 0.900000
Kd 0.905882 0.905882 0.905882
Ks 0.000000 0.000000 0.000000
Tr 0.200000
illum 2
Ns 0.000000
map_Kd D:\Week_35\final_harjuviitat\build1\Harjuviita22_12_rez.jpg
```