Space-variant picture coding

Popkin, Timothy John

Space-Variant Picture Coding

A thesis presented to the University of London
for the degree of Doctor of Philosophy
in Electronic Engineering

by

Timothy John Popkin

School of Electronic Engineering and Computer Science,
Queen Mary University of London,
Mile End Road, London, E1 4NS.

May 10th, 2010

Abstract

Space-variant picture coding techniques exploit the strong spatial non-uniformity of the human visual system in order to increase coding efficiency in terms of perceived quality per bit. This thesis extends space-variant coding research in two directions. The first of these directions is in *foveated* coding. Past foveated coding research has been dominated by the single-viewer, gaze-contingent scenario. However, for research into the multi-viewer and probability-based scenarios, this thesis presents a missing piece: an algorithm for computing an additive multi-viewer sensitivity function based on an established eye resolution model, and, from this, a blur map that is optimal in the sense of discarding frequencies in least-noticeable-first order. Furthermore, for the application of a blur map, a novel algorithm is presented for the efficient computation of high-accuracy smoothly space-variant Gaussian blurring, using a specialised filter bank which approximates perfect space-variant Gaussian blurring to arbitrarily high accuracy and at greatly reduced cost compared to the brute force approach of employing a separate low-pass filter at each image location.

The second direction is that of artificially increasing the depth-of-field of an image, an idea borrowed from photography with the advantage of allowing an image to be reduced in bitrate while retaining or increasing overall aesthetic quality. Two synthetic depth of field algorithms are presented herein, with the desirable properties of aiming to mimic occlusion effects as occur in natural blurring, and of handling any number of blurring and occlusion levels with the same level of computational complexity. The merits of this coding approach have been investigated by subjective experiments to compare it with single-viewer foveated image coding. The results found the depth-based preblurring to generally be significantly preferable to the same level of foveation blurring.

Acknowledgements

This work was funded by a grant jointly provided by the Engineering and Physical Sciences Research Council and BT Innovate & Design. It was performed under the supervision of Dr Andrea Cavallaro, whose guidance was fundamental throughout this work and the value of whose vigilant direction, support and help was enormous. Valuable support and advice was provided by Dr David Hands, particularly regarding subjective experimentation. Great help, especially in proof-reading, was provided by other supervisees of Dr Cavallaro. Valuable advice on parts of the work was provided by anonymous reviewers of the IEEE Transactions on Image Processing. Conference presentations of parts of the work were enabled by grant awards from the British Machine Vision Association, the Institute of Electrical and Electronics Engineers and the Royal Academy of Engineering.

To G., P., T., M. and D.

Contents

List of Tables **8**

List of Figures **9**

Associated Publications **10**

1 Introduction **12**

1.1 Motivation 12

1.2 Contribution 13

1.3 Structure of the Thesis 14

2 Background **15**

2.1 Introduction 15

2.2 Determining Saliency/Fixation 18

2.2.1 Human Interaction Techniques 18

2.2.2 Attention Models 19

2.3 Eye Models and How They Are Applied 23

2.3.1 Log Polar Model 23

2.3.2 Geisler & Perry Model 25

2.3.3 Other Models 26

2.3.4 Combining Saliency/Fixation Information with Eye Models 26

2.4 Exploiting Spatial Priority in Encoding 27

2.4.1 Using Established Block-Based Formats 28

2.4.2 Using Formats with Intrinsic Space-Variance 32

2.5 Selective Blurring Methods 35

2.5.1 Depth Blurring 37

2.6 Evaluation of Techniques 39

2.6.1 Objective Evaluation 39
2.6.2 Subjective Evaluation 43
2.7 Discussion 44
2.7.1 Foveation 45
2.7.2 Foveated Versus ROI Coding 46
2.7.3 Evaluation 47
2.7.4 Summary 48

3 Foveated Coding 50

3.1 Introduction 50
3.2 Optimal Blur Maps for Multi-Foveation Filtering 51
3.2.1 Multi-Viewer Sensitivity Model 52
3.2.2 Computing Multi-Viewer Sensitivity 54
3.2.3 Computing A Cut-Off Frequency Map 57
3.2.4 Example Output 60
3.3 Smoothly Space-Variant Gaussian Blurring 65
3.3.1 Proposed Approach 65
3.3.2 Evaluation and Comparisons 75
3.4 Conclusion 82

4 Depth-Blurred Coding 85

4.1 Introduction 85
4.2 Challenges in Synthetic Depth Blurring 87
4.3 The Blurring Algorithms 89
4.4 Implementation of Algorithm 1 94
4.5 Implementation of Algorithm 2 100
4.6 Computational Complexity 103
4.7 Conclusion 104

5 Subjective Evaluation 105

5.1 Introduction 105
5.2 Method 106
5.2.1 General Test Framework 106
5.2.2 Method-of-Adjustment Tests 106

5.2.3 Preblurring and Encoding 107
5.2.4 Details of the Experiment 109
5.3 Results 112
5.3.1 Discussion of the Results 114
5.4 Conclusion 120
6 Conclusion 121
6.1 Summary of Achievements 121
6.2 Future Work 123
Bibliography 125

List of Tables

2.1 Prior work in modelling human attention 20

2.2 Prior work in preprocess-only space-variant encoding 30

2.3 Prior work in spatially-non-uniform block-based encoding 31

2.4 Prior work using space-variant formats 33

2.5 Prior work in evaluating of space-variant coding 40

3.1 Gaussian blurring accuracy results 79

3.2 Gaussian blurring computational cost results 79

5.1 SSCQS and method-of-adjustment test scores 113

5.2 Mean scores across images 116

List of Figures

2.1 Generic space-variant encoder 17

2.2 Example output of spatial priority detection 21

2.3 Eye cell density of log-polar model 24

2.4 Geisler & Perry cut-off frequency curve 26

2.5 Example selective preblurring output 35

3.1 Foveated encoder block diagram 51

3.2 Eye model sensitivity curves 52

3.3 Eye sensitivity curve basis functions 54

3.4 Compute single sensitivity $\gamma' = S_{\mu,\mathbf{x}}(f)$ 56

3.5 Compute cut-off map $\phi_{\mu,\gamma}$ 58

3.6 Compute λ-percent cut-off map $\phi_{\mu,\sigma_\mu(\lambda)}$ 59

3.7 Example multifoveation cut-off maps 60

3.8 Frames used to demonstrate multifoveation 61

3.9 Multifoveation cut-off map comparison (1) 61

3.10 Multifoveation cut-off map comparison (2) 62

3.11 Multifoveation cut-off map comparison (3) 63

3.12 Block diagram of the proposed Gaussian blurring approach 66

3.13 Basis functions of the family of Gaussians 69

3.14 Cross-sections of the basis functions of the Gaussians 69

3.15 First 20 eigenvalues of the Gaussian blurring basis functions 71

3.16 Examples of applying Gaussian blurring on a synthetic image 71

3.17 Mandrill example of Gaussian blurring 74

3.18 Kodim08 example of Gaussian blurring 74

3.19 Cost-versus-accuracy plots of Gaussian blurring 76
3.20 Examples of foveal Gaussian blurring of Lena 77
3.21 Examples of foveal Gaussian blurring of Kodim18 78
3.22 Impulse response of blended Gaussian Pyramid 81
3.23 JPEG bitrate comparison of approximated Gaussian blurring 82

4.1 Depth/disparity map examples 86
4.2 Examples of occlusive effects in depth blurring 87
4.3 Examples of occlusive blurring effects on a synthetic image 88
4.4 Depth blurred coding block diagram 89
4.5 Blurring geometry diagram 91
4.6 Depth blurring block diagram 92
4.7 Pixel spreading conceptual diagram 93
4.8 Occlusive Pixel Spreader block diagram for Algorithm 1 94
4.9 Algorithm 1 pixel-spreading pseudocode 95
4.10 Treeadd and Treeget pseudocode 96
4.11 Treeadd illustrative example 97
4.12 Depth blurring example with various focal points 99
4.13 Synthetic-versus-camera depth blurring comparison 100
4.14 Algorithm 2 pixel-spreading pseudocode 101
4.15 Algorithm 2 summation structure illustration 102
4.16 Illustration of Algorithm 2's *get* operation 102
4.17 Tsukuba head & lamp example of depth blurring 103

5.1 Geisler & Perry cut-off frequency curve 108
5.2 Middlebury test set images 110
5.3 Two-level-disparity-map test images 111
5.4 Full-size example test images 114
5.5 Zoomed-in portions of favourable test images (1) 115
5.6 Zoomed-in portions of *Cones* test image 117
5.7 Zoomed-in portions of *Dolls* test image 117
5.8 Zoomed-in portions of favourable test images (2) 118

Associated Publications

The following publications have been produced in association with this thesis:

[1] T. Popkin, A. Cavallaro and D. Hands, “Multi-Foveation Filtering”, in *Proc. IEEE ICASSP*, Taipei, Taiwan, Apr. 2009, pp. 669-672.

[2] T. Popkin, A. Cavallaro and D. Hands, “Accurate and Efficient Method for Smoothly Space-Variant Gaussian Blurring”, in *IEEE Trans. Image Process.*, vol. 19, no. 5, May 2010, pp. 1362-1370.

[3] T. Popkin, A. Cavallaro and D. Hands, “Distance Blurring for Space-Variant Image Coding”, in *Proc. IEEE ICASSP*, Taipei, Taiwan, Apr. 2009, pp. 665-668.

Chapter 1

Introduction

1.1 Motivation

The dominant aim in lossy image and video coding is to obtain the highest coding efficiency in terms of quality per bit. The ultimate measure of quality is that as perceived by human viewers. For maximum efficiency, lossy systems aim to allow, for a given bitrate constraint, a distortion that is maximally acceptable to the viewer. The usual approach in aiming for maximally acceptable distortion is to aim for minimally visible distortion. Distortion can be measured in a number of ways, but most image and video encoders aim to satisfy measures which are spatially uniform; that is, which show no priority to different parts of the scene. However, the human visual system has a strong spatial non-uniformity which can be exploited to increase coding efficiency by applying a selective resolution reduction to images or video frames, as has been done by a number of published image coding techniques [4–19] and video coding techniques [4–7, 20–45].

This thesis specifically addresses the problem of how to spatially vary image resolution in order to exploit the human visual system for the purposes of lossy coding, given certain supplementary information such as eye traces. Towards solving this problem, this thesis extends space-variant coding research in two directions: (1) by pursuing the popular approach of aiming for minimally visible distortion according to a space-variant model of human retina resolution (*foveated*[†] coding); (2) by more directly addressing the ulti-

[†] Note: "fovea" is a medical term for "pit"; the dictionary defines "foveation" as "a pitted condition". However, for consistency with the literature, "foveated coding" and "foveation" herein denote the coding and processing of images or video while matching spatially-variant resolution and sensitivity of the retina.

mate aim of maximally acceptable distortion by aiming for a style of distortion that is aesthetically more acceptable when noticed; specifically, by using an approach borrowed from photography, of reducing the *depth of field* of an image, and using this for selective resolution reduction for lossy coding purposes (referred to herein as *depth-blurred coding*).

The impact of any image or video coding technique relates to applications of lossy image and video coding such as digital cameras, internet browsing, IPTV, digital video broadcast, HDTV, 3G phones, video content repositories, surveillance, DVD, Blu-Ray Disc and digital cinema. In such applications, there has always been a desire to store or transmit the maximum visual content using the smallest possible space, and this desire has applied to varying extents across all bitrates. Small compression improvements of a few percent might not have great impact on their own, but even in applications which are only affected by large leaps in technology, there is merit even with a small improvement of 5%, since, according to the "rule of 70" [46], this could be combined with 13 other non-interfering methods that each have the same small improvement in order to achieve a 50% bitrate reduction overall.

In lossy encoding, compression improvements can be obtained either by improving the efficiency of how selected information is stored or by improving the initial selection of the information which is most perceivable to the human visual system. The work of this thesis is confined within the latter case, investigating on still images certain principles which also have some applicability to video.

1.2 Contribution

The main points of contribution of this thesis (with reference to associated publications [1–3] which share these contributions) are as follows:

1. A novel algorithm (in section 3.2) for computing a multi-viewer or infinite-viewer eye sensitivity measure for use in foveation filtering, and the associated cut-off frequency map, in a manner which takes account of local fixation point density, which existing alternatives tend to disregard [1].

2. A novel algorithm (in section 3.3) for computing smoothly space-variant Gaussian blurring to high accuracy where previously only a discretely-varying blur level or

approximated Gaussian blurring were considered practical [2].

3. A novel approach (in chapter 4) for image or video coding by increasing the depth-of-field of an image or video frame using a novel algorithm for computing synthetic depth-of-field effects, along with subjective evidence (in chapter 5.3) of the advantage of this approach over foveated coding [3]. This takes the unusual approach of aiming for aesthetically acceptable distortion rather than the more conventional approach of aiming for minimally-visible distortion.

1.3 Structure of the Thesis

This thesis is structured as follows: Chapter 2 provides a review of space-variant coding techniques and relevant related background and highlights a number of open challenges that exist in this research. Chapter 3 presents proposals in the realms of foveated coding and space-variant image filtering. Chapter 4 presents proposals in the novel realm of depth-blurred coding. Chapter 5 presents the method and results of an experiment to subjectively compare depth-blurred coding with a simple foveated coding approach. Chapter 6 concludes the thesis.

Chapter 2

Background

This chapter provides a wide-ranging literature review of the current state of research into space-variant coding, encompassing the modelling of human visual attention, the modelling of eye sensitivity, space-variant encoder optimisation and perceptual quality evaluation, and highlights open challenges that exist in this research.

2.1 Introduction

Image and video coding techniques which employ a variation of bitrate or quality across an image or video frame according to spatial variation in visual importance (*saliency*) have received an increasing amount of interest in recent years [4–45]. This is generally due to the potentially large savings in bitrate that they are considered to have, because of the highly spatially-variant nature of the human visual system: at any instant in time, a human eye will only see a narrow visual region in sharp focus, due to having an increasing density of photoreceptor cells towards a highest density at the focal centre of the retina (the *fovea centralis*) [47, p. 236].

This thesis refers to such techniques collectively as *space-variant* image and video coding techniques, for consistency with the more common term *space-variant image processing* [48, 49]. Such coding techniques can be classed either based on whether or not they vary picture resolution according to a space-variant eye resolution model or based on whether they input external information about where observers are looking at each time. In the former case, *region-of-interest* (ROI) coding refers to techniques which apply a discrete number of differing levels of quality or bitrate to a discrete number of image regions of different saliency; whereas *foveated* coding refers to techniques which aim to match the eccentricity-dependent resolution and sensitivity of the human eye (aiming for minimally noticeable distortion) in conjunction with knowledge about where viewers look (their *fixation points*).

In the latter case, techniques can be classed as either gaze-contingent coding or coding based on spatially-varying estimated saliency. *Gaze-contingent* coding is a foveated coding approach in which video is encoded according to real-time fixation point knowledge (e.g. from eye tracking) for the single viewer. This approach has direct applications such as the teleoperation of remotely controlled vehicles. In this thesis, encoding techniques that encode video according to spatially-varying saliency are referred to as *saliency-based* encoders. A saliency map can be pre-determined either automatically from image or video content alone or using human input, such as an aggregation of measured eye traces. In this thesis, any such automatic saliency detection technique is regarded as an *attention model.* Although the automatic approach is more applicable to general use, it is held back by the difficulty of automatic saliency estimation.

The concept of a probability density map of viewer fixation at a given moment in time is referred to herein as a *saliency map*, although "saliency" is often treated as being a measurable quantity in terms of certain low-level features [38].

Space-variant encoders are generally oriented towards two space-variant properties of the human visual system: *optical* focus (a human viewer will only see a narrow part of his view in sharp focus at any point in time) and *mental* focus (the human brain is only likely to be mentally concentrating on a small number of things at any given time). However, it has generally been assumed that, at any point in time, the focus of attention and the fixation point of the eye will be at the same point [50]. So the problem of generating spatial priority information for use in image and video coding generally reduces to the problem of detecting or predicting where the eyes of viewers look. In all these encoders, the aim is the same: to maximise the overall perceptual quality (taking into account human visual space variance) for any given bitrate and to make the best possible use of the spatial priority information in doing so.

The ideal *foveated* encoding system depends on three pieces of information: (1) fixation point / saliency knowledge; (2) an eccentricity-dependent eye resolution / sensitiv-

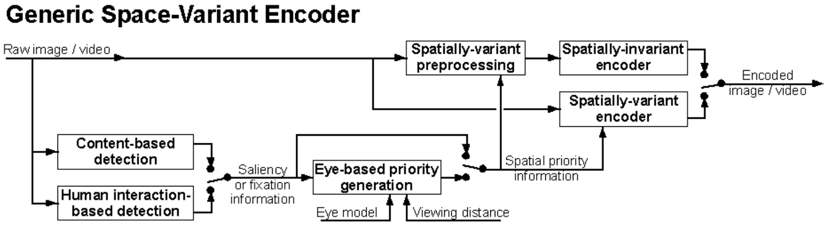

Generic Space-Variant Encoder
Raw image / video
Spatially-variant preprocessing
Spatially-invariant encoder
Encoded image / video
Spatially-variant encoder
Content-based detection
Human interaction-based detection
Saliency or fixation information
Eye-based priority generation
Spatial priority information
Eye model
Viewing distance

Figure 2.1: The generic simplified structure of a space-variant encoding system. Most space-variant encoders can be represented by a subset of the parts shown here. In this diagram, switches indicate the typical alternative sources of data at various stages. For a given switch arrangement, the diagram represents an encoder which consists of only the parts whose output is being used.

ity model; (3) knowledge about viewing distances and directions relative to the display (however, all foveated coding techniques reviewed herein have assumed head-on viewing direction, so this is not discussed further). These dependencies are shown in an illustration of a generic space-variant encoder in Fig. 2.1.

A useful review of the space-variant nature of the eye and models thereof, along with a review of foveated coding techniques as a whole, is provided by Wang & Bovik [4]. In contrast, this chapter additionally addresses the various sources of spatial priority information that have been applied to space-variant coding.

This chapter provides a wide-ranging overview of past work in the area of space-variant image and video coding, creating a structured summary of what has become a fragmented field of research. This review encompasses the modelling of human visual attention, the modelling of eye sensitivity, space-variant encoder optimisation and space-variant perceptual quality evaluation, and the open challenges that exist in this research are highlighted. This chapter discusses the wide range of approaches that have been taken in these areas and provides a literature-based intercomparison of existing techniques.

The chapter is organized as follows. Section 2.2 introduces the various approaches that have been taken in generating fixation points or saliency information for use in space-variant coding, focussing firstly on techniques involving human interaction and then on attention models. Section 2.3 discusses the spatially-variant resolution of the eye and introduces the work that has been done to model this for foveated coding purposes and how it has been combined with fixation points or saliency information in order to obtain the spatial priority that is applied in encoding. Section 2.4 introduces the various approaches that have been taken in exploiting spatial priority information in encoding, together with other associated prior work. Section 2.6 covers the background of the evaluation of space-variant coding techniques and related evaluation methods used in more general perceptual coding approaches. Finally, section 2.7 highlights open challenges that exist in this research and concludes the chapter.

2.2 Determining Saliency/Fixation

Any space-variant coding system needs a source of spatial priority information, whether as part of the system or as a prerequisite input sourced from elsewhere. Ultimately, this information tends to come in the form of a saliency map or a list of one or more estimated or measured fixation points. This section reviews the sources of such information, classifying them according to whether they use human interaction, as illustrated in Fig. 2.1.

2.2.1 Human Interaction Techniques

Because of the difficulty in reliably predicting human fixation, techniques which involve human interaction have played a significant rôle in space-variant coding. However, human interaction techniques are only useful in a limited number of scenarios. In *gaze-contingent* scenarios the video is encoded according to the exactly measured fixation point of a single viewer, thus allowing encoding to be directly optimised against a spatial priority map based on a model of the spatially-varying resolution and sensitivity of the human eye [10, 51]. This scenario can be extended to deal with a number of viewers, and, as this number increases, the problem converges to that of optimising against a priority map based on an appropriate combination of an eye resolution / sensitivity model with a probability density map of viewer fixation; that is, it converges to a saliency-based foveated coding scenario.

Exact knowledge of fixation points is available for certain scenarios such as flight simulation, teleoperation of a remote vehicle, teleconferencing, telemedicine and infrared and indirect vision devices [52, 53].

Eye tracking is the most obvious source of the human fixation information, providing a direct and reliable indication of fixation points. The greatest claimed benefits of space-variant encoding have involved a human viewer who is fixed in place and monitored by an eye tracker which feeds into a real-time image or video coding system. For example, Kortum's & Geisler's 18.8-to-1 bandwidth reduction [10] provides a view of the maximum bitrate savings that space-variant coding can provide (using selective resolution reduction of an otherwise uncompressed image) in the gaze-contingent scenario, in which knowledge is available of where a viewer's eyes will be looking at any point in time.

Instead of eye tracking, a mouse or other hand-operated pointing device can be controlled by the viewer (e.g., Geisler & Perry [28], who make a case for a number of practical applications of this approach), but this relies on his skill in moving the mouse to where his eyes are looking.

Semi-automatic techniques also exist. For example, for video, real-time eye tracking will invariably have some processing time lag, which may result in an increased noticeability of distortion during rapid eye movements; to address this, Komogortsev & Khan [27] propose a scheme combining eye tracking with automated short-term prediction of eye movements.

Human interaction can also be used as a substitute for an attention model. For example, a number of eye traces can be pre-collected from a number of viewers [11, 18, 19] for a given image or video sequence, and then aggregated into what may be considered to be the most reliable saliency map for that image or video sequence.

Fig. 2.2 shows the example output of a number of eye traces recorded from human viewers on a video sequence.

2.2.2 Attention Models

This section addresses the range models of human attention that have been employed in space-variant encoding. The full body of work in modelling human attention forms a very large research area that falls outside the scope of this thesis. Attention models are used for automatically estimating where viewers are likely to look within a scene and are generally classified into two groups: bottom-up (data driven, originating from computationally simple quantities that can be considered to be relevant to human perception) and top-down (task driven, originating from theories of human knowledge). Sometimes attention models can involve a combination of both (e.g., Navalpakkam & Itti [65]). In the words of Tang *et al.* [45], "visual attention can be guided by stimulus-driven (bottom-up) and goal-directed

	Top-down / Bottom-up?	Motion?	Colour / Intensity / Contrast?	Edges?	Faces?	Trained?
Agrafiotis *et al.* [38]	Bot.-up	1: √ 2: ×	1: √ 2: ×	1: × 2: √	×	×
Bradley & Stentiford [14, 54, 55]	Bot.-up	×	×	×	×	×
Cavallaro *et al.* [56]	Bot.-up	√	×	×	×	×
Chen *et al.* [57]	Bot.-up	√	√	×	×	×
Doulamis *et al.* [37]	As trained	×	×	×	As trained	√
Ho *et al.* [58]	Both	√	√	×	√	×
Itti & Koch [59, 60]	Bot.-up	×	√	×	×	×
Itti & Koch [29, 61]	Bot.-up	√	√	×	×	×
Tang [45]	Bot.-up	√	×	×	×	×
Tsapatsoulis [31]	Both	×	√ [59, 62]	×	√ [63]	×
Wang *et al.* [4, 21, 22]	Top-down	×	√	×	√	√
You, Liu & Li [64]	Bot.-up	√	√ (local st. dev.)	×	×	×

Table 2.1: Prior work in the modelling of human attention for use in image or video coding.

(top-down) mechanisms".

Attention models can be used to produce, for each video frame, a *saliency map*, which is considered herein to be a probability density map of predicted fixation points.

Bottom-up techniques for saliency detection generally involve some combination of one or more of motion, intensity/contrast, edges or other low-level features. Low-level image features such as intensity, contrast and edge density have been shown to correlate with genuine human attention (e.g. see Parkhurst *et al.* [66]). Examples of bottom-up techniques include those of Cucchiara *et al.* [67] and Cavallaro *et al.* [41, 56], which all make use of motion detection, as do the techniques of Koch & Ullman [68] and Tang [45]. Wang & Bovik [9], Le Meur *et al.* [69], Wolf & Deng [70] and Sun & Fisher [71] use the detection of intensity, contrast, motion or other low-level features. You, Liu & Li [64] combine motion detection with the use of the local standard deviation of intensity, and also use a prior spatial priority such that more central positions are favoured. Agrafiotis

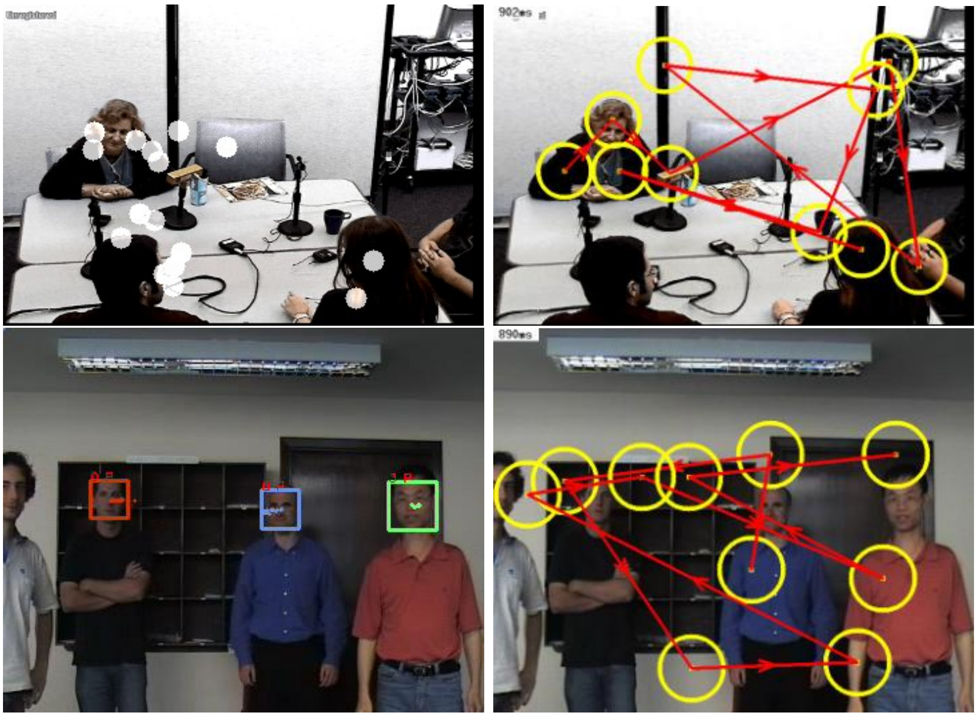


Figure 2.2: Example output of spatial priority detection. Top left: fixation points recorded on the given video sequence using an eye tracker. Top right: example predicted eye fixations created on the given still frame using the attention model of Itti [29]. Bottom left: example output of a face tracking technique [72]. Bottom right: example predicted eye fixations created using the same attention model as in Fig. 2.2, shown for comparison. Top video sequence taken from the CLEAR dataset [73].

et al. [38] propose two context-specific approaches, one for sign language, based on motion and contrast and the other for football matches, based on the detection of edges. The Itti & Koch model [29,59–61] uses a combination of motion detection and contrast, specifically in terms of local orientation of features. Tsapatsoulis *et al.* [31] extend the Itti & Koch model, combining it with a face detection technique [63]; Chen *et al.* [57] combine the Itti & Koch model with video objects from an unspecified source. Bradley & Stentiford [14,54,55] employ an evolutionary programming approach looking for the novelty of local structure.

Bottom-up approaches tend to be easier to implement than top-down approaches, and produce reasonably effective results in most scenarios. However, the capacity of *bottom-up* techniques to predict human fixation is considered by some to be limited. Henderson *et al.* [74] found in psychophysical experiments that "intensity, contrast, and edge density differed at fixated scene regions compared to regions that were not fixated, but these fixated regions also differ in rated semantic informativeness", suggesting that the success of such techniques may be due to a general co-location, in real-world images, between low-level features and regions with high-level meaning.

Top-down attention approaches (defined as goal-directed by Tang *et al.* [45]) arise from the fact that the things a person looks at may depend in any way on very high-level mental effects. So, it is not considered feasible to perfectly predict where a person or group of people are going to look. However, certain things have a distinct tendency to attract attention, such as faces or text. The tendency to look at faces or face-like depictions is instinctive, as demonstrated by Fantz by his experimental observation of babies [75].

Face-based top-down techniques include those of the skin-colour-based face detection technique of Wang *et al.* [4, 21, 22], which performs face detection using "binary template matching" [76]. Ho *et al.* [58] use skin colour segmentation. Fig. 2.2 shows some example results of a face-tracking technique.

Another approach for top-down techniques is that of trained mechanisms. Examples of this include the neural network-based face detection technique of Rowley, Baluja & Kanade [77] and the technique of Navalpakkam & Itti [65], which uses accumulated statistical knowledge of low-level features. The DCT-domain neural network technique of Doulamis *et al.* [37] uses a neural network classifier, producing a block-by-block, two-level saliency.

Some of the top-down attention model literature may be considered to be partially, if not dominantly, bottom-up, due to combining low-level features with a top-down aspect such as a decision scheme (e.g., Sun & Fisher [71]).

An existing field which holds a large number of ready-made candidates for top-down attention models, or alternative bottom-up attention models, which have mostly been unexploited in video coding, is that of object detection and tracking.

Fig. 2.2 shows example output obtained by running Itti's [29] bottom-up attention model, shown for comparison against recorded eye traces and against a face tracking (top-down) technique.

Table 2.1 summarises the key attributes of techniques for the human attention modelling for use in image or video coding. The predominance of bottom-up techniques over top-down techniques can clearly be seen.

2.3 Eye Models and How They Are Applied

The defining factor of *foveated* coding techniques is their employment of a of model of space-variant nature of an eye. As illustrated in Fig. 2.1 the saliency or fixation information is combined (in the "Eye-based priority generation" block) with the eye model to provide a spatial priority map of some form, such as a map of local cut-off frequencies [21].

The spatially-varying resolution and sensitivity of the human eye has been measured through psychophysical experiments (e.g., Robson & Graham [78], Banks *et al.* [79] and Arnow & Geisler [80]). Such experiments generally involve trials whereby a human subject looks into a controlled display at a specified point while a trial is carried out in which a large number of visual stimuli of differing characteristics are displayed at differing locations relative to this point of focus, and the subject indicates which of these he does or does not detect. The recorded data can be used to construct a generic model of the spatially-variant nature of the human eye, which can subsequently be employed in foveated coding. This has been done in the past, generally by fitting a much-simplified model to the empirical data. Foveated coding techniques tend to assume a radially-symmetric model of the spatial variation of acuity (i.e., eye resolution). This subsection introduces these examples of such models that have been used in foveated coding, and also touches on other eye models that were not derived from empirical data.

Tables 2.2, 2.3 and 2.4 show which eye models have been used by a number of foveated coding techniques. (For non-foveated techniques, the eye model is marked as "None".)

2.3.1 Log Polar Model

The most established eye model used in foveated image processing and representation is known as the logmap, or log polar, model [48], which directly provides a spatial map of cut-off frequencies. It proposes that the inter-spacing between eye cells in any location of the human retina is directly proportional to the distance from the fovea centralis. Therefore, local resolution is inversely proportional to eccentricity. In practice, an upper limit on resolution has to be imposed, for example by assuming a level cell density in a region around the fovea centralis.

The log polar model is a simplified, radial model of the eye which, for example, takes no account of the retina's blind spot.

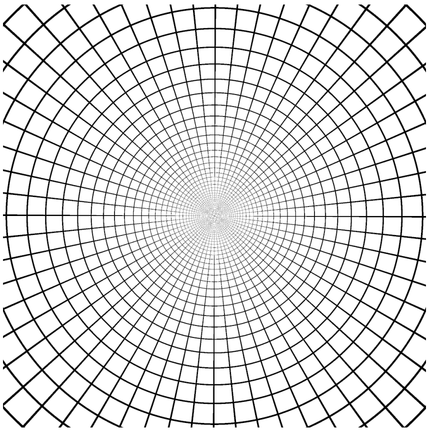


Figure 2.3: An illustration of the variation in eye cell density assumed by the log-polar model.

The log polar model is a practical model for use in foveated coding because foveated image processing such as *foveation filtering* (that is, eye-resolution-based blurring) can be performed by transforming the image into log polar co-ordinates around the fixation point, then performing uniform blurring on the resulting transformed image and finally performing the inverse transformation.

Another interesting property of the log polar model is its invariance to scaling about its central point, as illustrated by Fig. 2.3. When the reader focuses on the central point, every square shown will typically represent roughly 1000 ganglion cells[†], irrespective of viewing distance. This assumption falls down in the narrow central, foveal region, in which true eye cell density levels out, whereas a perfect log polar model assumes the cell density to increase asymptotically towards the centre. In general, if a log-polar model of eye resolution is assumed and employed and if the viewer focusses on the expected point, the relative local resolution at any location in the foveated image will be the same in relation to the corresponding local eye resolution at that location, irrespective of what distance the viewer is looking from. In other words, under these assumptions, the viewing distance doesn't matter. This property is also demonstrated by Anstis' eye chart [82], which comprises an arrangement of characters around a central point, with the sizes of the characters increasing in direct proportion to the distance from that point. On this chart, "each character is about five times the smallest perceivable size when the center is

[†] Calculated assuming 7.5 arcminutes between ganglion cells at 40° eccentricity (as per Davson [81], Fig. 14.15, lower line), and assuming hexagonal packing of cells.

fixated", irrespective of viewing distance [83]).

2.3.2 Geisler & Perry Model

A popular model of the eccentricity-dependent variation of eye sensitivity is that of Geisler and Perry [28]. This model gives more than just the spatial resolution of the eye (in terms of a maximum observable frequency for each eccentricity); it actually yields a frequency-dependent sensitivity curve for each eccentricity. This model was introduced in the form of their *contrast threshold formula*, which provides, for each eccentricity e (in degrees) and spatial frequency f (in cycles per degree), a contrast threshold $CT(f, e)$, below which components of that frequency are considered to be invisible, as follows:

$$CT(f, e) = CT_0 \exp\Big(\alpha f \tfrac{e+e_2}{e_2}\Big), \tag{2.1}$$

where constants $e_2 = 2.3$, $\alpha = 0.106$ and $CT_0 = 1/64$ were chosen to provide the best-fitting model to the empirical data of Robson & Graham [78] and subsequently verified against other empirical data [79,80]. Contrast threshold can be defined as the contrast (i.e., luminosity relative to the overall display) for which 90% of Gaussian-windowed sinusoidal patches of given frequencies at given locations are not noticed by subjects [79]. Alternative contrast threshold formulae exist (e.g. Kelly [84] and Reddy [85]), but eq. (2.1) has been repeatedly re-used in subsequent research (e.g., Wang et al. [4, 5, 8, 9, 21, 22, 86], Sheikh et al. [23] and Ho et al. [35]).

Note that the contrast threshold formula only makes implications about what will and will not be visible; it says nothing about *how* noticeable it is if visible, but it can be used as the basis for a sensitivity function or a cut-off-frequency map for a given critical threshold. For example, Wang and Bovik [9, 21] define *contrast sensitivity* as $CS(f, e) = 1/CT(f, e)$; that is,

$$CS(f, e) = \tfrac{1}{CT_0} \exp\Big({}^{-}\alpha f \tfrac{e+e_2}{e_2}\Big); \tag{2.2}$$

and also, for a critical threshold $CT(f, e) = 1$, they obtain a cut-off frequency $f_c(e)$ (in cycles per degree), defined

$$f_c(e) = \frac{e_2 ln(1/CT_0)}{(|e| + e_2)\alpha} \tag{2.3}$$

for each eccentricity e (again in degrees). This formula is illustrated in Fig. 2.4. Others (e.g. Sheikh et al. [23, 24] and Ho & Wu [33, 35]) have taken similar approaches. Sheikh *et*

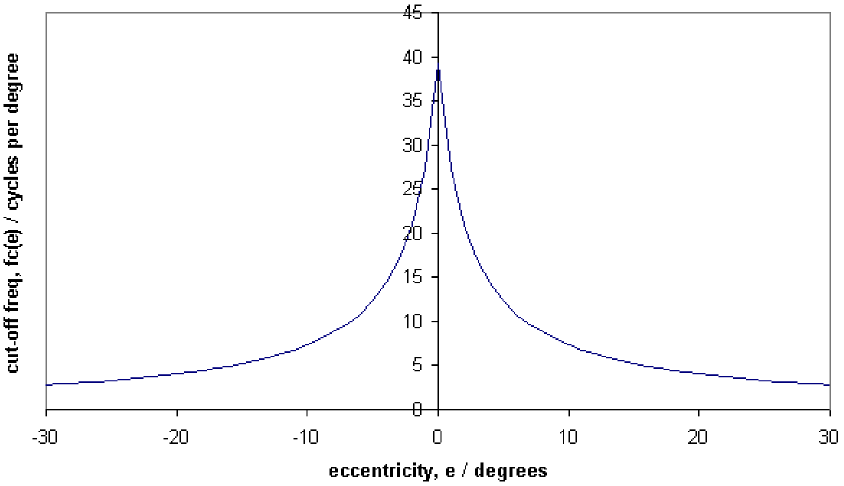


Figure 2.4: A model of the spatial cut-off frequency curve of the human eye: the cut-off frequency curve interpretation [21] of the Geisler & Perry model.

al. [24] allow the critical threshold to be adjustable, so that the cut-off frequency map can be varied by the rate control mechanism of the foveated encoder. Note that taking this approach of solving equations (2.1) or (2.2) for a fixed contrast threshold or sensitivity value will yield a cut-off frequency map that is optimal in the sense of discarding least-noticeable local frequency components, but only for the single-viewer scenario.

2.3.3 Other Models

Other models of eye resolution that have been employed (with possible adaptation) in foveation include those of Daly *et al.* [36] and Peli *et al.* [87], which both take the approach of fitting a simple curve to empirical data.

Rather than using a simplified model, it is possible to use the measured eye sensitivity data more directly, as done by Duchowski [88] using the data of Foster *et al.* [89].

2.3.4 Combining Saliency/Fixation Information with Eye Models

The application of an eye model effectively converts a saliency map or a finite list of fixation points into a spatial priority map. The majority of foveated coding techniques are oriented to the single-viewer scenario, in which it is assumed that the viewer is looking at a single, known fixation point. In the single-viewer scenario, this can be done by simply combining the single fixation point with the projection of the eye resolution model onto the image plain, for example to derive a corresponding space-variant cut-off frequency to use in encoding (e.g. Wang *et al.* [21]). However, in typical coding scenarios there is no eye tracking and there may be any number of viewers, each gazing at a different point. In this multi-viewer scenario, the conversion from a list of fixation points or saliency map into some sort of spatial priority map that can be exploited by the encoder is non-trivial.

While the larger proportion of foveated coding techniques have assumed a single viewer, a number of attempts have been made (e.g., Dhavale & Itti [30], Sheikh *et al.* [23] and Wang & Bovik [4] to extend foveated coding to the multi-viewer scenario, or even to the probability-based (infinite-viewer) scenario, in which only a fixation probability map (a *saliency map*) is available, such as from a visual attention model.

In the case of foveation filtering and similar approaches which aim to work by cutting off the least visible frequencies, the majority of existing approaches to producing a cut-off map to satisfy a number of viewers have tended to take an approach that can be regarded as aiming for distortion that lies below the contrast threshold of every viewer, rather than aiming for distortion that minimises some measure of collective noticeability. Because of the radial symmetry and decreasing nature of the sensitivity function, these approaches [4, 21, 23–26] effectively take the overall sensitivity value of each location as that of the nearest fovea, thus allowing the inverse (i.e. the cut-off map) to be computed in a straightforward manner.

Regarding the issue of viewing distance and orientation, in gaze-contingent coding, the exact viewer location relative to the display tends to be known a priori. However, in the more general saliency-based foveated coding scenario, viewers may be located at any distance, in any direction from the display. Most foveated coding research has simply assumed a fixed viewing distance. The only exception to this is Wang & Bovik [9], who assumed a log-normal probability distribution of viewing distance. Furthermore, all foveated coding literature has assumed head-on viewing.

2.4 Exploiting Spatial Priority in Encoding

As illustrated in Fig. 2.1, the spatial priority information used by a space-variant encoder may be in different forms, such as a list of fixation points, a segmentation of each image or video frame into a finite number of regions each with an associated importance value, or a spatial map giving some measure of the importance of each pixel. An example of this map is, in the case of foveated coding, a map of local cut-off frequencies [21].

Techniques for the exploitation of this information in image and video coding are grouped herein into: (a) techniques which use classic block-based coding approaches such as JPEG for images, and MPEG-1 and H.264 for video (in section 2.4.1); (b) techniques which use formats that are intrinsically suited to encoding with spatially-variant resolution, such as wavelet-based and object-based techniques (in section 2.4.2). Under both of these groups, prior work has included techniques which adhere to established coding formats (i.e., the encoder is modified but the corresponding decoder is untouched), and techniques which introduce bespoke formats (i.e., encoder and decoder both changed or created afresh), specially devised for space-variant video coding, usually by modifying established formats.

Note that not all space-variant coding approaches follow the structure presented in 2.1. For example, an approach employed by Van Der Linde [12] and subsequently Çöltekin [51], for use with dual video streams feeding into stereoscopic displays, takes the foveated coding a step further, by not only aiming to exploit the eccentricity-dependent resolution of the retina of the eye, but also the distance-varying resolution of the lens of the eye. That is, they employ differing levels of resolution so as to exploit the fact that the eye will, at any moment in time, be focussed on one particular distance, such that anything located at other distances will appear blurred in the eye and hence can be encoded with reduced resolution (and hence reduced bitrate). This relies on real-time eye tracking data, working on two eyes instead of the usual single eye. A notable consequence of this approach is that it simulates “the limited depth of field phenomenon” [12].

2.4.1 Using Established Block-Based Formats

The realm of image and video coding is dominated by block-based formats such as JPEG and MPEG-1, MPEG-2, MPEG-4, H.261, H.263 and H.264. Because of the abundance of encoders and decoders for such formats, much of the work in space-variant coding has aimed to use these formats, whether by preprocessing the input image or video in some way which allows an existing encoder to attain a reduced bit rate, or by modifying such an encoder in some way to allow a spatial non-uniformity across the image or video frame, possibly also making modifications to the format itself so that the corresponding decoder also needs to be modified.

Using Space-Variant Preprocessing

Many space-variant coding techniques involve preprocessing of the image or video so as to remove perceptually redundant information (generally by selective blurring) in a manner which will result in an improved rate-distortion efficiency when passing the output into an ordinary encoder ("Spatially-invariant encoder" in Fig. 2.1) that employs a spatially-uniform cost function. Such techniques have invariably used block-based coding formats.

This filtering approach aims to selectively cut off the less noticeable local spatial frequency components of the image. For example, in the case of foveation filtering, the noticeability of local spatial frequency components is determined according to a sensitivity map based on the eye. The result can then be encoded with an ordinary encoder that does not prioritise any part of the scene. Because the eye is generally more sensitive to lower spatial frequencies, this amounts to selective preblurring. Preblurring allows bitrate reduction because the more blurring has been incurred by a region of the image or frame, in formats such as MPEG-2 which encode near-zero DCT coefficients and sequences of zeros efficiently, the lower the bitrate is likely be for that region, for a given distortion requirement. Selective preblurring has also been used successfully in ROI coding [31,41,44].

In any type of lossy coding, the aim is to remove the least-noticeable or least-important information. Because the eye is generally more sensitive to lower spatial frequencies, selectively cutting off the less noticeable local spatial frequency components of an image or video frame amounts to *selective preblurring*. This is a popular, simple approach for exploiting spatial priority information in image and coding. It involves blurring each image or video frame so that different parts of the scene are blurred by different amounts depending on their estimated (or measured) levels of interest or in a way which aims to be minimally noticeable to observers. The output is typically then passed into an ordinary, off-the-shelf image or video encoder of a chosen output format. The encoder does not prioritise any part of the scene in terms of quality. Instead, it reduces the bitrate devoted to the more blurred regions, as mentioned before. Apart from being straightforward to implement, selective blurring has the added advantage that, if noticed, the style of

	Fixation point assumptions	**Eye resolution model assumed**	**Source of spatial priority information**	**Coding format**
Cavallaro ***et al.*** [56]	n/a	None	Automatic	MPEG-1
Dhavale & Itti [30]	Multiple	Cauchy distribution [85]	Automatic [62]	MPEG-1
Dikici ***et al.*** [43]	Single	Gaussian-like	Automatic	None suggested
Duchowski [88]	Multiple	From MAR data [89]	Unspecified	Unspecified
Itti [60]	n/a	None	Automatic [60]	Modified JPEG
Itti [29]	Small number	Unspecified	Automatic [29]	MPEG-1 & MPEG-4
Karlsson ***et al.*** [44]	n/a	None	Assumed prior knowledge	H.264 High Profile
Tsapatsoulis [31]	n/a	None	Automatic [31]	MPEG-1

Table 2.2: Prior work in space-variant encoding using preprocess-only techniques.

the distortion it creates is more likely to occur naturally, and may be considered more acceptable to a human than alternative artifacts such as blocking that occurs with block-based formats as the bitrate is reduced.

The selective preblurring approach has been demonstrated with foveated coding (e.g., Itti [29], Dhavale & Itti [30] and Dikici *et al.* [43]) and with non-foveated coding (e.g., Cavallaro *et al.* [41, 42]). Duchowski also uses non-eye-based resolution models.

Karlsson *et al.* [44] extend the notion of blurring to the time domain, performing temporal as well as spatial selective low-pass filtering.

In spite of the abundance of newer video formats, MPEG-1 [90] is the most popular format used for investigating the selective preblurring approach [29–31, 56], although the approach has been shown to work with later formats such as MPEG-4 [29] and H.264 [44].

Table 2.2 gives an overview of some notable aspects of a number of existing techniques which work by selective preprocessing approaches with ordinary encoders.

Example bitrate reductions reported for these approaches include a typical bitrate reduction of 1.8-to-1 with minimal perceptual loss, as reported by Dhavale & Itti [30], and equivalently 2.8-to-1 by Itti [29].

	Fixation point assumptions	Eye resolution model assumed	Source of spatial priority information	Coding format
Agrafiotis *et al.* [38]	1: n/a; 2: saliency map	1: None; 2: Geisler & Perry [28].	Context-specific, automatic [38]	H.264
Doulamis *et al.* [37]	n/a	None	Automatic [37]	MPEG-1 or any MCP-DCT encoder
Ho *et al.* [33–35]	Single (but extendable)	Geisler & Perry [28]	Unspecified	MPEG-1
Khan & Komogortsev [7, 27]	Single	Variation of Daly et al. [36]	Real-time eye tracking + short-term prediction	MPEG-2
Liu & Bovik [20]	Single	Geisler & Perry [28]	Assumed prior knowledge	H.263 or appropriate DCT-based format
Sheikh *et al.* [23, 24]	Multiple	Geisler & Perry [28]	Unspecified	H.263 & MPEG-4
Tang [45]	n/a	None	Automatic [45]	H.264

Table 2.3: Prior work in space-variant encoding using block-based encoding with internal spatial non-uniformity.

Block-based Encoding with Internal Spatial Non-uniformity

Many techniques bring the exploitation of spatial priority information into the encoder itself, encoding with a spatial non-uniformity in terms of quality or bitrate, rather than applying uniform priority across each image or frame. In contrast with subsection 2.4.1, this subsection discusses the techniques for which the encoding itself has a spatial non-uniformity in terms of quality or bitrate, rather than selectively removing information prior to an ordinary, spatially uniform encoder. Table 2.3 gives an overview of some notable aspects of a number of such techniques that have been applied to established block-based formats. These techniques sometimes exploit features of formats that were not designed for exploiting the space-variance of the human visual system. A particularly common approach, particularly with MPEG-1 [90] or MPEG-2 [91] or other transformed-block-based video coding formats, is to apply the spatial priority block-by-block, reduce the bitrate of the blocks classed as “less interesting” by increasing the quantisation granularity (e.g., Chai *et al.* [40]). Similarly, Agrafiotis *et al.* [38] employ a macroblock-level variation of quantization with H.264, Doulamis *et al.* [37] perform blockwise prioritised bit allocation with MPEG-1, and Ho *et al.* [33–35] perform DCT-domain optimisation (specifically coefficient elimination) using Lagrange multipliers, also with MPEG-1. Liu & Bovik [20] perform a similar approach but focus on H.263, as do Sheikh *et al.* [23, 24], also focussing on MPEG-4, with suitability for any DCT-based encoder that uses motion compensated prediction.

Other interesting approaches include those of Khan & Komogortsev [7, 27], which combines real-time eye tracking with short-term prediction of eye movements, and that of Tang [45], which takes account of the phenomenon whereby the presence of signals may reduce the visibility of other signals.

Itti’s work [60], aiming to demonstrate the application of an attention model in image coding, used a modified JPEG to demonstrate image coding capability, working by adjusting DCT coefficient quantization and including an encoded saliency map.

Example reported bitrate reductions of these approaches include up to 15% reduction without perceptual degradation by the ROI coding technique of Tang [45], and a reduction of 35% with little quality loss by the foveated coding technique of Liu & Bovik [20]. Agrafiotis *et al.* reported up to 30% bitrate reduction with negligible loss of quality for both their foveated approach and their ROI coding approach.

2.4.2 Using Formats with Intrinsic Space-Variance

Certain image and video coding techniques such as wavelet-based and object-based techniques are, by their nature, more suited to encoding with spatially-variant resolution than the more common approaches such as the DCT block-based techniques.

Most of the earlier research into space-variant coding, as well as much of the more recent work has involved the proposal of a novel coding format or a modification of an existing format. Table 2.4 gives an overview of some notable aspects of a number of such techniques.

	Fixation point assumptions	**Eye resolution model assumed**	**Source of spatial priority information**	**Coding format**
Bradley & Stentiford [15]	n/a	None	Unspecified	JPEG 2000
Bradley & Stentiford [13, 14]	n/a	None	Automatic [54, 55]	JPEG 2000
Cavallaro ***et al.*** [56]	n/a	None	Automatic	MPEG-4 object-based
Chang ***et al.*** [92]	Single	Log polar	Assumed prior knowledge	Bespoke (wavelet coefficient quantization)
Chen ***et al.*** [57]	n/a	None	Automatic + manually-defined video objects.	MPEG-4 object-based
Ebrahimi-Moghadam & Shirani [17]	n/a	None	Assumed prior knowledge	Bespoke
Farid ***et al.*** [32]	Single	Concentric squares	Real-time eye tracking	Bespoke (wavelet-based)
Kortum & Geisler [10]	Single	Log polar	Real-time eye tracking	Raw
Nguyen ***et al.*** [18, 19]	Multiple	None	Multiple pre-recorded eye traces	JPEG 2000
Nystrom ***et al.*** [11]	Multiple	None	Multiple pre-recorded eye traces	Bespoke
Sanchez ***et al.*** [16]	n/a	None	Assumed prior knowledge	JPEG 2000
Silsbee ***et al.*** [39]	Single	Two concentric circles	Assumed prior knowledge	Bespoke
Wang ***et al.*** [8, 9]	Multiple	Geisler & Perry [28]	Assumed prior knowledge	Bespoke (wavelet based, modified SPIHT [93])
Wang ***et al.*** [4, 5, 21, 22, 86]	Multiple	Geisler & Perry [28]	Either assumed prior knowledge or automatic [21]	Bespoke

Table 2.4: Prior work in space-variant encoding using formats with intrinsic space-variance.

Wavelet & Hierarchical Tree Techniques

Wavelet and hierarchical tree techniques lend themselves easily to space-variant encoding. For example, space-and-resolution-oriented coefficient trees can be selectively pruned so as to control the level of detail in each region of an image or video frame [39]. The JPEG 2000 format, designed for image coding but also usable on video, has special provision for applying region-of-interest bias to parts of an image. Bradley & Stentiford [15] present three spatially-variant coding mechanisms associated with the JPEG 2000 format: tiling, code-block selection and coefficient scaling. A more general approach that can be taken with wavelet-based techniques is to assigned priorities to blocks of wavelet coefficients, as done by Nguyen *et al.* [18, 19] based on eye trace clustering & image statistics. Another approach used in wavelet-based techniques is the "Maxshift" algorithm, as used by Bradley & Stentiford [13, 14] in conjunction with a binary attention map comprising one or two elliptical foreground regions, or by Sanchez *et al.* [16] in a progressive ROI coding approach. The choice of regions and quality levels with such techniques is tightly restricted to a fixed hierarchy of discrete levels and locations, so they are not perfectly suited to foveated coding, for example, where the aim is to encode with a close-fitting match to the spatial variation of the human eye.

A popular approach here is the use of wavelet-based compression in non-standard formats, as done by Nystrom *et al.* [11], Chang *et al.* [92], Farid *et al.* [32], Ebrahimi-Moghadam & Shirani [17] using the "matching pursuit" [94] technique for *progressive* ROI compression (that is, for scenarios when it is desired to decode parts of an image before others), Wang & Bovik [8, 9] with their modified SPIHT algorithm [93] optimized by coefficient weighting against an eye-based distortion measure, and Wang, Lu & Bovik [4, 5, 21, 22, 86], who have developed a full wavelet-based video compression technique involving "intra" frames and "predictive" frames.

Object-based Techniques

Some image and video coding formats are well-suited for space-variant coding. Object-based video coding formats (e.g., parts of MPEG-4 Part 2 [95]) are ideal for space-variant coding. Examples of the application of automatically detected saliency to object-based MPEG-4 include the work of Cavallaro *et al.* [56] and Chen *et al.* [57].

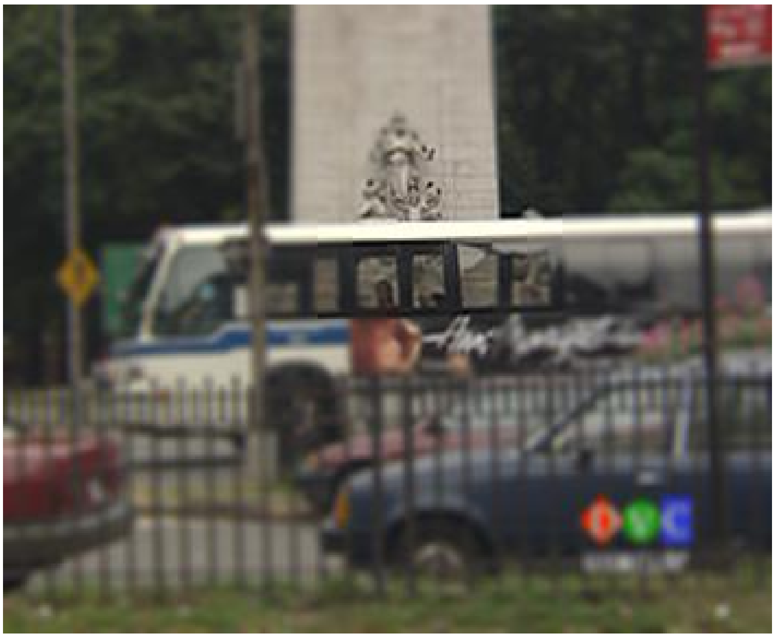


Figure 2.5: Example output of an existing selective preblurring technique [23]. The sharper focus area can be seen around the centre of the image.

Other Techniques with Space-Variant Formats

While the majority of published space-variant encoding techniques tend to use a block-based, wavelet-based or object-based coding format, some of the earliest examples did not follow this trend. The technique of Silsbee *et al.* [39], which is the earliest known example of an image coding technique that exploits a model of the spatial variation of the eye, is 3-D block-based, using a look-up table into a predefined set of patterns, with spatial priority exploited in a resolution hierarchy tree. Kortum & Geisler [10] employed a simple variable-sampling scheme according to the assumed variation in eye resolution, with no other compression employed.

2.5 Selective Blurring Methods

Selective preblurring, or selective prefiltering, whereby different parts of a video frame are blurred to different extents, is a key part of a number of space-variant coding techniques (see section 2.4.1).

Existing approaches to space-variant blurring include filter banks, in which the image is typically filtered using a number of parallel band-pass or low-pass filters, whose outputs undergo a space-variant combination [4]. For example, Sheikh *et al.* [23] quantize the blur levels into a finite number of discrete levels, for each of which a separate filtration is computed. Fig. 2.5 shows some example output of their technique. These result in a discretisation of blur levels (frequency bands), which precludes *smooth* space-variance. However, in foveated coding, because the spatial acuity models tend to be continuously varying, it is generally ideal to employ a continuously varying blur level.

Another approach is to apply a spatial co-ordinate transformation, such as log polar mapping [48], and then apply uniform blurring, before applying the reverse transformation [26]. However, the log polar mapping approach can only generate foveal blur maps, and not general blur maps as with the other techniques. The simplest approach is the *summed area table* [96] (or *integral image* [97]) approach, but its square blurring gives it poor frequency domain characteristics which are detrimental to compression performance.

The linear filters in a filter bank can be implemented by number of approaches, the simplest being a non-recursive convolution, implemented as a direct finite-impulse response (FIR) filter. Alternatively, a recursive, infinite impulse response (IIR) filter can be employed [98]. Recursive filter approximations of Gaussian blurring have also been adapted to be space-variant [99]. However, high-precision FIR and IIR filters require a large number of taps, and there will always be a point at which a fast convolution technique [100, p. 538] will be more efficient (e.g., data lengths in the range of 20 to 50 points in the case of 1-D filtering, dependent on implementation [101, p. 8-2]).

Gaussian blurring is a style of blurring that is particularly common. Gaussian blurring is an effect that may occur naturally, and it has a very natural appearance to the human eye. In particular, a Gaussian has a rapid fall-off both in the spatial domain and in the frequency domain and does not suffer from the ringing effect. Moreover, due to the Central Limit Theorem, repeated blurring of any type will converge to Gaussian blurring.

In the case of 2-D Gaussian blurring, the filtering at each level can be performed by separate vertical and horizontal convolution by a 1-D Gaussian, but to do this in a general space-variant manner is too expensive for a real-time system [99].

Hierarchical techniques, which construct and employ a pyramid of blurred versions of the original image at differing levels of resolution, are highly efficient. Of these, the blended *Gaussian Pyramid* approach [102, 103] performs a smoothly space-variant approximation of Gaussian blurring. In this, the blurring is initially done at discrete levels, but the final blurred image is computed by interpolating, for each pixel, between the blurred images of blur levels either side of the desired blur level for that pixel. Gaussian Pyramid works by computing a logarithmic hierarchy of Gaussian-filtered, subsampled versions of the image or video frame, and obtaining the blurred image by interpolating between pixels of the subsampled images at the appropriate levels as well as the aforementioned interpolation between the levels of the pyramid. However, the approximation of Gaussian blurring provided by blended Gaussian Pyramid has limited accuracy.

Smoothly space-variant blurring with a non-trivial filter and a general blur map is considered prohibitively expensive in terms of computational cost, potentially requiring a different low-pass filter for each image location. In the words of Wang and Bovik [4, p. 435],

> *... an ideal implementation of foveation filtering would require using a different low-pass filter at each location in the image. Although such a method delivers very high quality foveated images, it is extremely expensive in terms of computational cost when the local bandwidth is low.*

However, this does not consider the possibility of the existence of a fast computation technique that achieves the same result with greatly reduced cost. In the specific case of selective Gaussian blurring, section 3.3 presents such a technique.

Selective blurring has other applications outside the realm of space-variant coding. For example, Geisler & Perry [102] extend the notion of foveation filtering to use arbitrary eye resolution maps (deviating from the usual, radially-symmetric models), and suggest the simulation of visual impairments of patients of glaucoma (a condition in which pressure in the eyeball damages the retina [81, p. 79] to help non-glaucoma sufferers to obtain an understanding of the problem.

2.5.1 Depth Blurring

Another interesting area of where selective blurring is applicable is in computer graphics, where realistic rendering of synthetic scenes requires the simulation of depth blurring. Realistic depth blurring necessitates the simulation of occlusion effects that are not present in ordinary selective blurring.

Depth blurring and space-variant coding are areas that have met in recent publications. As mentioned in section 2.4, foveation was extended into the 3rd (depth) dimension by [12], and continued by [51]. This work employs Gaussian pyramid blurring [102] for resolution-reduction purposes, aiming for minimally perceivable distortion rather than photorealistic blurring that is aesthetically acceptable on close inspection as proposed herein. This work relies on eye tracking and does not test the plausibility of removing the eccentricity-dependent foveation aspect altogether along with any assumptions about where the viewer will look.

A recent review of techniques for the computation of depth blurring (*depth of field rendering*) in a computer graphics scenario is given by [104]. Existing techniques are classed as either *multipass approaches*, in which high-accuracy techniques such as ray tracing are repeated a number of times from slightly different directions and averaged [105], or *postfiltering*, in which the rendering output itself is subjected retrospectively to synthetic depth blurring. Multipass approaches [105] produce highly realistic depth blurring, including the occlusive blurring effects that naturally occur at object edges in blurring, but they are generally inappropriate for real-time applications due to heavy computational cost. The postfiltering approaches are dominated by the use of a fast resolution pyramid approach such as a mipmap or Gaussian Pyramid [12, 51, 106–108]. These employ the *gather* method; that is, they approximate depth blurring by taking the local average of pixel values around the desired location, which inherently leads to *intensity leaks* [104] as the intensity from sharp source pixels is spread over surrounding background that they should not influence. The alternative approaches employ the *scatter* method (such as the depth-related blurring of splats [109]), whereby the intensity of each source pixel is spread over an area (its *circle of confusion*, in the case of circular blurring). However, due to speed, blending and image energy conservation issues scatter methods are not the choice for real-time depth blurring [110]. The favoured compromise between cost and quality is to use multiple depth layers, whereby a separate computation is performed on depth-segmented sub-images, at a discrete number of depths [104, 111–113]. However, the computational cost of this approach increases with the number of layers to filter, which places a practical limit on the blurring quality, since with most techniques a low number of layers results in the *discretization artifact* [114].

2.6 Evaluation of Techniques

For ordinary encoding, measures such as the PSNR are often used due to their convenience and simplicity. Taking a step beyond that, measures can be used that are more perceptually-oriented, such as the Structural Similarity Index Measure (SSIM) [115]. Any such measure, requiring no human input, is an *objective* measure. However, the evaluation of space-variant codecs must take a further step again, by using measures that are themselves space-variant in accordance with the human visual system if they are to demonstrate any advantages over encoding techniques with no spatial variation. Ultimately, the measure that any space-variant codec should aim to satisfy is that of the quality as perceived by human viewers, and the only widely recognized method of providing this information is subjective experimentation [116].

This section provides an overview of the realm of image and video quality evaluation as relevant to space-variant coding and the broader area of perceptual coding in general. Note that to consider the space-variant aspects of quality evaluation alone would not be appropriate, since any codec or quality measures that go to the effort of addressing the space-variant characteristics of the human visual system will generally also address other, easier-to-model aspects of the human visual system, such as its lower sensitivity to higher spatial frequencies.

A wide range of information about image and video quality assessment is contained within the book *Digital Video, Image Quality and Perceptual Coding* [117]. This section provides a more condensed view, with a bias towards space-variant coding.

Table 2.5 summarises the key attributes of approaches that have been used for evaluating space-variant coding techniques. Of these techniques, five out of eleven involved a subjective element, six out of eleven either tested a gaze-contingent system or made the assumption that the viewer would fixate on an author-defined point, four out of eleven involved a space-variant distortion measure and five out of eleven were informal assessments which relied on the judgement of the authors themselves as a critical step.

2.6.1 Objective Evaluation

Although the ideal measure of the quality of any lossy coding technique, space-variant or otherwise, is subjective quality, this is complex and time-consuming [116], and it is invari-

	Subjective / objective	**Assumes known fixation**	**Subjective methodology used**	**Space-variant distortion measure?**	**Type of result**	**Based on author opinion?**
Bradley & Stentiford [13, 14]	Subj.	×	2-Alt. Forced Choice	×	PSNR improvement	×
Cavallaro *et al.* [56]	Subj.	×	Abs. Category Rating [118]	√	R-D curves	×
Kortum & Geisler [10]	Subj.	√	Author-defined	×	Compr. ratio	×
Tang [45]	Subj.	×	DSCQS [119]	×	Compr. ratio	×
Tsapatsoulis [31]	Subj.	×	2-Alt. Forced Choice	×	Compr. ratio	√
Dhavale & Itti [30]	Obj.	√	n/a	×	Compr. ratio	√
Dikici *et al.* [43]	Obj.	√	n/a	√	PSNR improvement	√
Itti [29]	Obj.	√	n/a	√	Compr. ratio	×
Karlsson *et al.* [44]	Obj.	×	n/a	×	Local PSNR improvement	√
Liu & Bovik [20]	Obj.	√	n/a	×	Compr. ratio	√
Wang *et al.* [8, 9]	Obj.	√	n/a	√	PSNR improvement	×

Table 2.5: Prior work in the evaluation of space-variant coding techniques.

ably quicker and less complicated to use an objective quality measure, without the need for human input. Furthermore, subjective evaluation severely restricts the range of test parameters that can reasonably be investigated, thereby precluding styles of assessment such as detailed rate-distortion curves that are common in objective video quality assessment. The advantages of objective evaluation outweigh the disadvantages more often than not and, as a result, examples of objective evaluation occur far more frequently than those of subjective evaluation and are often performed alongside subjective tests.

PSNR A commonly used measure of image or video quality is the Peak Signal To Noise Ratio (PSNR), which, for an 8-bit-per-pixel monochrome image, is defined as [116, 120]

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\mathrm{MSE}}$$

where MSE is the *mean squared error* of the image or video sequence; that is, the mean of the squared differences between pixels of the reference image or video and pixels of the image or video being assessed. Here, the reference image or sequences is ideally the raw version as it was before any encoding took place.

There is no general agreement on the computation of the equivalent measure which takes colour into account [116]. For colour images or video, it is common to consider the "Y" (luminance) component alone, using a measure known as *Y-PSNR*.

The problem with the PSNR is that can deviate from the evaluations of a human subject, for example due to the low sensitivity of the human visual system to high spatial frequencies [116], whereas the PSNR treats all spatial frequencies with the same weighting.

Structural Similarity Index The Structural Similarity (SSIM) index [115] is intended as a measure of the local structural similarity between images, and the mean SSIM (MSSIM) index is a measure of the overall similarity between images [121]. The principal behind the SSIM index is that it moves away from the notion that the distorted image is the result of applying additive noise to the reference image and instead aims to disregard less important information, specifically the variation of luminance and contrast [121]. The MSSIM has been demonstrated by subjective tests [122] to give better prediction capabilities of subjective opinion scores based on the ITU five point quality scale (see subsection 2.6.2) than the PSNR [121]. SSIM and MSSIM values range between 0 and 1.

No-Reference Measures Quality measures which work by comparing a distorted image or video sequence with a reference image or sequence (nominally the original raw image or sequence as it was before encoding) are known as *full-reference* measures. In contrast, *no-reference* (NR) measures aim to estimate quality based on the distorted content alone. NR measures have the disadvantage that without a reference, there is no way to be certain that what may appear to be distortion was not, in fact, part of the original raw content.

Full-reference measures provide a more reliable indication of quality, and are better for providing an assessment of the general performance of a codec. However, scenarios exist, such as in flexible control systems for video delivery [123], in which it may be desirable to continuously monitor the receiver side, where the original content will not be available.

Winkler [116] points out that most NR quality measures are oriented towards *blockiness*, which is a compression artifact of all DCT block-based encoding formats, and provides a review of such measures, of which this subsection mentions a brief selection.

Approaches to detecting blockiness include frequency-domain methods which look for distinct spectral peaks at spatial frequencies relating to the (known) block size, as done by Wang *et al.* [124], and the examination of pixel differences at the (known) locations of block boundaries [125]. Other compression artifacts used as the target of NR measures include blurriness and ringing [126, 127]. NR quality measures tend to be oriented to distortion of a known and recognisable nature. However, Gastaldo *et al.* [128] use a neural network approach which has no prior assumptions about human perception.

Predicting Subjective Measures Because of the clear superiority of the output of subjective evaluation and the tendency of standard subjective testing methods to follow their own scales (see subsection 2.6.2), there is a desire for objective measures which aim to accurately predict these subjective quality ratings. Work to assess of such measures in recent years has mostly been undertaken by the ITU's Video Quality Experts Group (VQEG) [123]. The group have completed two phases (I and II) of subjective assessment full-reference objective quality measures. In the Phase II tests, which were oriented to two picture resolutions of 525 and 625 lines, six objective human visual models were compared with subjective quality data, and one only landed in the top-performing group for both picture resolutions: that of NTIA [129, p. 38]. This model uses a *reduced-reference* approach, using certain features extracted from spatio-temporal regions of the video sequence [129, p. 44]. It achieved an average Pearson correlation of 0.91 with the subjective rating [129, p. 52]. This compared with an average of 0.77 for ordinary PSNR [129, pp. 21-22].

Space-Variant Measures As mentioned above, a quality measure needs itself to be space-variant if it is to demonstrate any advantages of a space-variant technique. For this, there is no widespread standard.

An approach sometimes taken for space-variant objective assessment is to perform a stand-alone assessment of the encoding end of the system against the spatial priority information which it takes as an input. This can be done using a space-variant quality measure which itself inputs the same spatial priority information and which, in the case of foveated coding, assumes the same eye model as applied in the encoding process [8, 9, 43]. This may provide an indication of how well the encoder part performs its immediate task of exploiting the spatial priorities, but whether this is a reliable indication of the true perceptual quality depends on the reliability of the spatial priority information which it uses. In the case of gaze-contingent coding, the original eye movements are available and can be used to assess the visual quality using a space-variant distortion measure based on the eye model. However, for space-variant coding approaches that rely on estimated saliency or fixation points, this issue can be addressed by separately, subjectively assessing the attention model itself [29], or by using recorded eye traces of one or more human viewers as the source of spatial priority. Methods which use human input in this way can be regarded as partially subjective.

2.6.2 Subjective Evaluation

Standards For the evaluation of a complete coding system including its source of spatial priority information, the ideal aim is to measure the perceived quality of its decoded output as judged by human subjects. ITU recommendations ITU-R Rec. BT.500 [119] and ITU-T Rec. P.910 [118] define standard procedures for performing subjective evaluation under controlled, repeatable conditions, such as the Double-Stimulus Continuous Quality Scale (DSCQS) method, in which each subject gives each image or video clip under test a quality rating in the range 0 to 100, guided loosely by the ITU five-point quality scale (*Excellent/ Good/Fair/Poor/Bad*) and judged in comparison with a reference image. However, of the space-variant coding publications reviewed herein, only three make use of one or other of these ITU recommendations [45, 56, 69].

Just-Noticeable Distortion In space-variant coding literature, the bulk of the quantitative indications of the advantages of space-variant approaches over ordinary encoding have come in the form of relative improvements in compression rate whilst roughly preserving perceptual quality. These are produced by tests in which encoding parameters are chosen so as to impose minimal or no perceptual loss as judged independent subjects [10, 45] or, in informal tests, as judged by the authors themselves [20, 29, 30, 38]. A common approach in psychophysics is to define the notion of *just noticeable distortion* as the point at which 50% of subjects cannot perceive any loss of quality [130, p. 27], but the majority of space-variant coding literature has not restricted itself to this sort of established formality. For example, Kortum & Geisler chose fixed encoding parameters themselves and used informal subjective testing to support a claim of “minimal perceptual artifacts”. A step further than this is a *method of adjustment* test [130, p. 27], in which the subject himself is able to tune the parameters to a point where the quality of the image or video under test matches that of a reference. This approach would have the benefit of allowing precise quantitative subjective evaluation, rather than just the pass/fail outcome of a hypothesis test when the encoding parameters are fixed. Also, by using reference images to which a well-known type of distortion has been applied, method-of-adjustment tests would allow subjective testing against a rigid, well-defined scale rather than against an arbitrary scale that may vary between subjects as is the case with ITU-R BT.500 [119] and ITU-T P.910 [118]. In spite of this, the use of method-of-adjustment tests is completely absent from the space-variant coding literature reviewed herein and has not been incorporated into any ITU subjective quality standard to date.

Alternatives As mentioned earlier, a problem with direct subjective evaluation is that the range of test parameters that can reasonably be investigated is severely restricted, thereby precluding styles of assessment such as detailed rate-distortion curves. However, Cavallaro *et al.* [56] performed a post-processing on their subjective quality results, combining them with certain assumptions so as to re-interpret the data in the form of a “semantic PSNR”, and created rate-distortion curves accordingly.

2.7 Discussion

This section highlights a number of limitations of the current state of research into space-variant coding and the other closely related areas covered by this chapter.

2.7.1 Foveation

Estimating Human Fixation In spite of the assumed potential of space-variant coding, space-variant techniques have had limited employment outside the gaze-contingent viewing scenario due to the general unavailability or difficulty in estimation of knowledge of which parts of a scene viewers are likely to be looking at or thinking about. As the non-gaze-contingent scenario is the only area of foveated coding that could be applied in everyday image or video coding, the source of estimated human fixation is very important.

Bypassing the difficulty of providing a reliable source of estimated or measured human fixation, chapter 4 will present an alternative approach which does not need precise knowledge of human fixation points.

Blurring As discussed in section 2.5, Gaussian blurring is a particularly common and sought-after style of blurring, but past techniques proposed for doing this have either been prohibitively expensive or only provided crude approximations. A method for accurate smoothly space-variant Gaussian blurring is therefore desirable. Section 3.3 will answer this directly.

Multi-viewer Foveation Even when saliency or fixation information is fully available, gaze-contingent coding is simpler than saliency-based foveated coding. In gaze-contingent coding, the “Eye-based priority generation” block of Fig. 2.1 can generate a spatial priority map by simple translation and scaling of the eye sensitivity function (taken as a 2-D map of sensitivity) according to the viewing distance. For multiple fixation points or for a saliency map, the combination is more complicated. As mentioned in section 2.3.4, the majority of existing approaches have tended to solve the multi-viewer foveation problem by segmenting the image or video frame into a number of single-viewer foveation problems, whereby each pixel’s sensitivity-frequency curve is taken from the single-viewer eye sensitivity model assuming the nearest fovea, rather than aiming for distortion that minimises some measure of collective noticeability. However, this is not ideal because, for example, any number of co-fixated viewers are treated exactly as a single viewer with the given fixation point. Therefore, within regions which attract fixation, as the number of viewers becomes large and the problem converges to the probability-based (saliency map) scenario (and the inter-fixation-point distance becomes small), the solution locally converges within these regions to that of ordinary, uniform-priority encoding, therefore neglecting local variations in fixation point density and losing some, or possibly all, of the coding advantages of the knowledge of human fixation. An alternative ideal would be that whatever combination method is employed should instead be derivable from some measure of average viewer satisfaction. Section 3.2 will answer this directly.

2.7.2 Foveated Versus ROI Coding

A further question in saliency-based coding aims right at the heart of foveated coding itself: whether it is worthwhile using a space-variant eye model at all. That is, whether foveated coding is any better than simpler ROI coding. In the gaze-contingent scenario, the answer is arguably yes, given that the aim is to match scene detail as closely as possible to eye sensitivity on a local level, aiming to optimally prioritise what is encoded according to what is observable. However, in saliency-based coding, when exact fixation is unknown, it is questionable whether the use of the eye model provides an overall advantage over applying a saliency map directly as a spatial priority map in the encoding. Firstly, it is conceivable that, in an image that has been notably distorted by foveation, the attention of the viewer may be distracted by this and hence any estimated or pre-collected distribution of fixation points may become invalid; furthermore, it is conceivable that a viewer's mere awareness of any such distortion might affect his judgement of the overall visual quality even if his main focus of attention is as predicted. Secondly, a limitation of employing an eye model is that the more spread out the saliency map, the lower is the potential for bitrate savings. The same can be said of any saliency-based coding approach, but it can be argued to be more of a problem for foveated coding because the foveation effects themselves apply a spread on the saliency map. ROI coding techniques, which apply the saliency map more directly therefore generally have a greater contrast in local bitrates, and hence a lower overall bitrate for a given range of quality levels across an image. Furthermore, in object-based coding approaches, whereby the sharp boundaries in quality level will tend to correspond with the natural boundaries of objects, this semi-natural effect may cause the resultant image or video to be visually more attractive for a given overall bitrate than if the degradation is more gradual. It is interesting to note that the sorts of bitrate savings that have been reported for non-gaze-contingent foveated coding approaches (e.g., 35% [20]) are not greatly different from the sorts of savings reported for ROI coding approaches (e.g., up to 15% [45]).

Section 5.3 will take steps to answer the question of whether foveated coding is any better at exploiting estimated saliency or fixation points than alternative space-variant techniques, including by comparing a simple foveated coding approach with a two-level image segmentation.

2.7.3 Evaluation

One of the biggest problems facing researchers in any form of perceptual coding is how to measure the performances of techniques (that is, performance in terms of perceptual quality for a given bitrate) while such techniques, by definition, aim to directly satisfy the human visual system, rather than indirectly, by satisfying an established quality measure such as the PSNR. The ideal answer is subjective testing, but this is expensive and laborious, and in practice, it is more convenient to use an objective performance measure which itself has been separately justified by subjective evaluation or has an arguable grounding in empirical evidence or established theories. Subjective evaluation standards (such as ITU-R BT.500 [119] and ITU-T P.910 [118]) have the added limitation that they rely on the human subjects devising their own ways of judging, and so the numerical scales are vulnerable to differences between subjects. Also, the values they produce do not relate to any established quality scale in objective coding.

To address the difficulty of evaluation of techniques subjectively, given that standard scales are vulnerable to differing human interpretation and do not produce values that can be compared with objective evaluations, section 5.2.2 will present a subjective evaluation method which uses no word-based scale and produces output in the form of compression ratios.

In space-variant coding, there is the difficulty that perceived quality depends in general on where people look within an image or video frame, so the use of an automatic measure is unsound unless the objective measure itself takes as input a subjective element such as human eye traces, or unless a solid argument can be provided to justify assumptions about areas of interest, such as subjective tests to directly assess the correctness of the underlying attention model used (e.g., Itti [29]). In practice, much of the space-variant coding literature has failed to provide a sound evaluation of proposed techniques in terms of benefits in overall perceptual quality, and much of the evaluation work that has been performed relies heavily on assumptions about the areas of interest, which may have been manually chosen [8, 9, 44] or use the same attention model and eye sensitivity as were proposed as part of the technique itself [43]. Where subjective tests have been performed, they have often not followed an established formal procedure. For example, as mentioned in subsection 2.6.2, a common approach in psychophysics is to define the notion of *just noticeable distortion* as the point at which 50% of subjects cannot perceive any loss of quality [130, p. 27]; however, the greater proportion of gaze-contingent coding literature has tended not to follow any sort of standard in this type of test.

To add one further quantitative evaluation of space-variant coding techniques, section 5.3 will provide a quantitative subjective evaluation of the approach in chapter 4.

2.7.4 Summary

This chapter has presented a wide-ranging overview of past work in the area of space-variant image and video coding, creating a structured summary of what has become a fragmented field of research. Such research has been driven by the highly space-variant nature of the human visual system, which can be exploited by encoding different parts of an image or video frame with a variation of bitrate or quality.

This chapter has qualitatively intercompared the space-variant encoders themselves, but a universal quantitative comparison is unfeasible because of the wide range of scenarios, encoding formats and evaluation methods used. The levels of improvement reported for space-variant encoding schemes in comparison with their spatially-uniform counterparts vary widely. Typical improvements reported for gaze-contingent approaches range as high as an 18.8-to-1 reduction in bitrate, whereas the typical improvements for saliency-based techniques tend to be much more modest, at between 15% and 2.8-to-1 reduction with minimal perceived degradation.

The general unavailability of exact fixation point knowledge and the difficulty in predicting human fixation pose a great challenge against the drive to exploit the space-variant nature of the human visual system in image and video coding. However, the drive continues due to the expected benefits. To facilitate this drive, it is desirable to address a number of holes in the field. This chapter has highlighted a number of such holes, to which the remaining chapters of this thesis will take a number of steps. The next chapter addresses the first of these, in the realm of *foveated coding*.

Chapter 3

Foveated Coding

This chapter addresses two issues in foveated coding. Firstly, a method is presented for computing an additive multi-viewer sensitivity function based on the Geisler & Perry contrast threshold formula, and, from this, a cut-off frequency map (as used in foveation filtering*) that is optimal in the sense of discarding frequencies in least-noticeable-first order. Secondly, a method is presented for performing smoothly space-variant accurate Gaussian blurring where previously only a discretely-varying blur level or roughly approximated Gaussian blurring was considered practical.*

3.1 Introduction

As mentioned in section 2.1, the most common approach to space-variant coding is foveated coding, which aims to spatially vary image or video frame resolution in order to match the spatially-varying resolution of the human retina. As mentioned in sections 2.4.1 and 2.5, a simple approach for foveated coding is the simple approach of selective preblurring of an image or video sequence as a prior stage to an ordinary encoder, thereby exploiting the fact that most lossy encoders use fewer bits to encode image regions that have a lower frequency band. Fig. 3.1 gives an example of a simple hypothetical architecture for such an encoder in the case of video, here assuming that the source of spatial priority is an object detection technique.

This chapter extends foveated coding research by two further steps, corresponding respectively to the "Blur Map Generation" and "Selective Preblurring" blocks of the architecture in Fig. 3.1. In section 3.2, a novel algorithm is proposed for computing the optimal blur map in the sense of discarding least-noticeable frequencies first, given a saliency map

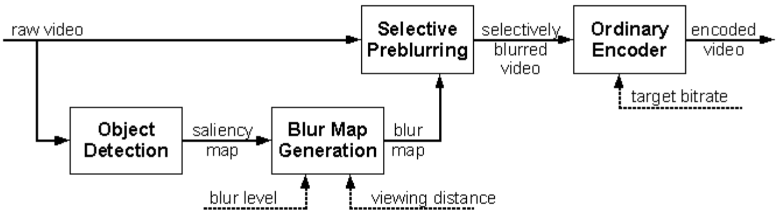


Figure 3.1: Block diagram of a possible foveated video encoder. A saliency map is generated based on video content, in this case using an object detection technique. From this, a blur map is generated (according to an assumed model of eye resolution and a given viewing distance), which is applied to each frame of the video prior to ordinary encoding. The blur level and target bitrate are controlled as part of the overall encoding process.

or a number of known fixation points (multi-foveation). In section 3.3, an algorithm is for applying any blur map in the form of arbitrarily accurate Gaussian blurring. Section 3.4 concludes the chapter.

3.2 Optimal Blur Maps for Multi-Foveation Filtering

As mentioned in section 2.2.1, the greatest benefits of space-variant coding have been demonstrated in the *gaze-contingent* scenario, working with a single known fixation point. However, as mentioned in section 2.3.4, typical coding scenarios may have any number of viewers, each gazing at a different point, and a number of attempts have been made to extend foveated coding to the multi-viewer or probability-based (infinite-viewer) scenarios.

The problem which this section addresses is that of how a frequency-dependent sensitivity function, and the corresponding cut-off frequency map, should be defined and computed in the multi-viewer or infinite-viewer scenarios.

This section is organised as follows: a multi-viewer or saliency-based sensitivity model based on the Geisler & Perry model is proposed in subsection 3.2.1, and a novel algorithm is proposed in subsection 3.2.2 for the computation of this sensitivity; accordingly a novel algorithm for computing a cut-off frequency map that is optimal for foveation filtering in the sense of discarding least-noticeable local frequency components is proposed in subsection 3.2.3, along with an extension which enables an up-front choice of the percentage of spatio-frequency components that are desired to be discarded. Subsection 3.2.4 presents the example output of the technique and compares it with the output of an existing ap-

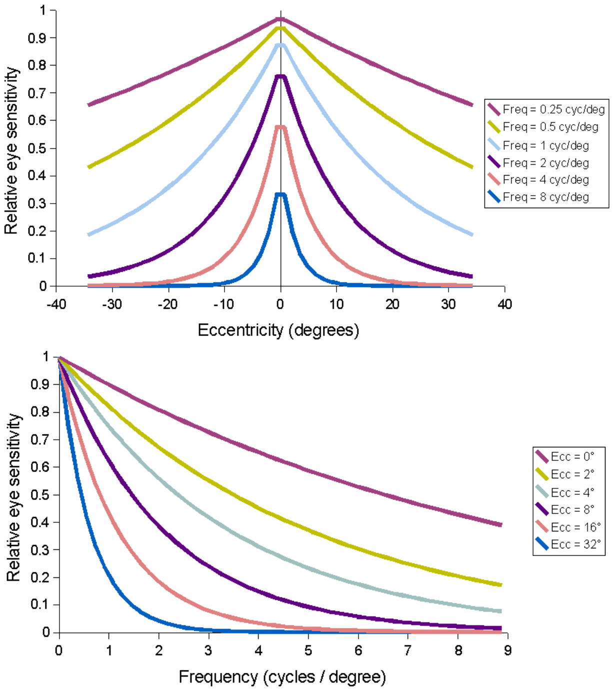


Figure 3.2: Illustration of the assumed eye sensitivity model. Left: sensitivity-versus-eccentricity curves at fixed frequencies. Right: sensitivity-versus-frequency curves at fixed eccentricities.

proach to multi-foveation.

3.2.1 Multi-Viewer Sensitivity Model

Recall from section 2.3.4 that existing multi-foveation approaches have tended to aim for distortion that lies below the contrast threshold of every viewer, rather than aiming for distortion that minimises some measure of collective noticeability, and effectively take the overall sensitivity value of each location as that of the nearest fovea. Recall that any number of co-fixated viewers are therefore treated exactly as a single viewer with the given fixation point and that, within regions which attract fixation, as the number of viewers becomes large and the problem converges to the probability-based (saliency map) scenario (and the inter-fixation-point distance becomes small), the solution locally converges, within these regions, to that of ordinary, uniform-priority encoding, therefore neglecting local variations in fixation point density, and losing some, or possibly all, of the coding advantages of the knowledge of human fixation.

This subsection introduces an alternative sensitivity function, which takes the approach of adding the sensitivities at each point, rather than effectively taking the maximum sensitivity.

Recall equation 2.2 (from section 2.3.2):

$$CS(f,e) = \tfrac{1}{CT_0}\exp\left({}^{-}\alpha f \tfrac{e+e_2}{e_2}\right),$$

which gives a measure of the sensitivity $CS(f,e)$ of a human eye to a given frequency component f (in cycles per degree) in a given direction e (in degrees), where $e_2 = 2.3$, $\alpha = 0.106$ and $CT_0 = 1/64$. From this, a normalised sensitivity function s, can be defined as follows:

$$s(f,e) = \exp({}^{-}(|e|+e_2)\,\alpha f/e_2)\,, \tag{3.1}$$

for all $e \in \mathbb{R}$ and $f \in [0,\infty)$, with α and e_2 as before. This function is illustrated in Fig. 3.2.

To convert this to the image domain, assume that the viewer is positioned so that he has head-on viewing of the fixation point. Therefore, given that the fixation point is located at $\mathbf{y}$ and the viewing distance is d (both in units of one pixel width), as an approximation (neglecting trigonometry), the eccentricity of image location $\mathbf{x}$ will be $360\|\mathbf{x}-\mathbf{y}\|/2\pi d$. Now, define function $a\colon[0,\infty)\to[0,\infty)$ as follows:

$$a(r) = (360r/2\pi d \;+\; e_2)\,\alpha/e_2 \tag{3.2}$$

for all r. Let the set $D = \{0,...,W{-}1\}\times\{0,...,H{-}1\}$ represent the domain of any $W\times H$ image. Then, a sensitivity function $s_{\mathbf{y}}\colon D{\times}[0{,}\infty)\to[0{,}1]$, for a given fixation point $\mathbf{y}$, can be defined as

$$s_{\mathbf{y}}(\mathbf{x},f) = \exp({}^{-}a(\|\mathbf{x}-\mathbf{y}\|)\,f) \tag{3.3}$$

for all $\mathbf{x}$ and f.

This can be extended to the multi-viewer and infinite-viewer scenarios by summation, so that, in the infinite-viewer scenario, in which a fixation probability density map (saliency map) $\mu\colon D\to[0,1]$, is available, the infinite-viewer sensitivity level $S_{\mu,\mathbf{x}}(f)$, for location $\mathbf{x}$ and frequency f, becomes

$$\begin{aligned} S_{\mu,\mathbf{x}}(f) &= \sum_{\mathbf{y}\in D}\mu(\mathbf{y})s_{\mathbf{y}}(\mathbf{x},f) \\ &= \sum_{\mathbf{y}\in D}\mu(\mathbf{y})\exp({}^{-}a(\|\mathbf{x}-\mathbf{y}\|)f). \end{aligned} \tag{3.4}$$

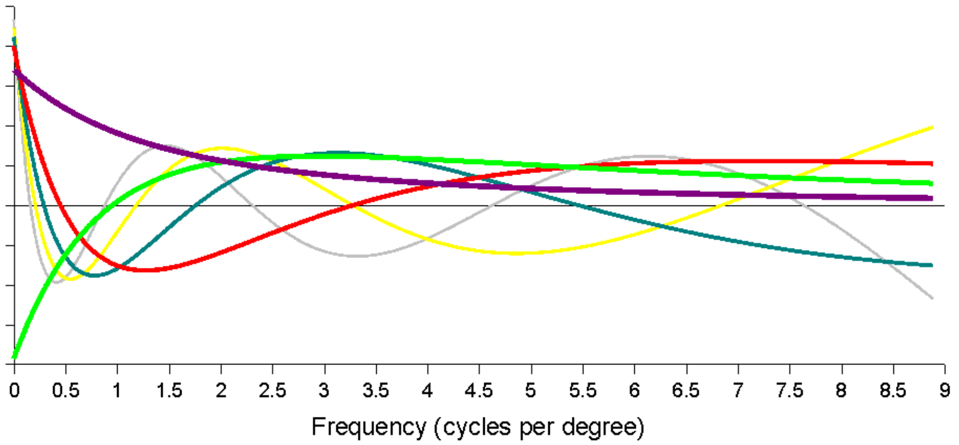


Figure 3.3: The basis functions used in the computation of the sensitivity-v-frequency curves.

Note that the interpretation of a finite-viewer sensitivity function from this infinite-viewer function can be performed by trivially setting map μ to be a sum of 2-D Dirac delta functions.

3.2.2 Computing Multi-Viewer Sensitivity

To compute each $S_{\mu,\mathbf{x}}(f)$ value by interpreting eq. (3.4) verbatim would be prohibitive, as each sensitivity value would involve sum with $H \times W$ terms. However, a close approximation of this computation can be performed using a faster approach which will now be described.

Define a family $E = \{e_\beta : \beta \in [0, A]\}$ of functions e_β , where $A = a(\sqrt{(H-1)^2 + (W-1)^2})$ is the maximum $a(\|\mathbf{x}–\mathbf{y}\|)$ value that can occur in eq. (3.4) and each $e_\beta : [0, F] \to \mathbb{R}$ is a limited-domain function defined as follows:

$$e_\beta(f) = \exp({}^{-}\beta f) \tag{3.5}$$

for all $f \in [0, F]$. Here, $F \in (0, \infty)$ is the maximum-representable frequency, which, neglecting the possibility of non-head-on viewing (for simplicity), equates to the pixel-diagonal Nyquist frequency; that is, the maximum-representable number of cycles per $\sqrt{2}$ pixel widths, which is $\sqrt{1/2}$ cycles per pixel, which is $\sqrt{1/2} \times 2\pi d/360$ cycles per degree, where d is the viewing distance, as in eq. (3.2). Consider a function $z : [0,F] \times [0,F] \to \mathbb{R}$, defined such that

$$z(f_1, f_2) = \int_0^A e_\beta(f_1) e_\beta(f_2) d\beta \tag{3.6}$$

for all $f_1, f_2 \in [0, F]$. Now, when z is approximated by a discrete-domain matrix Z, this matrix is symmetric, and hence can be diagonalised by an orthonormal basis of eigenvectors [131, p. 379]. This process can be used to compute close approximations to the orthonormal eigenfunctions $b_1, b_2, b_3, ...,$ of z. It happens that all except a small number of the eigenvalues of z are very close to zero, the result being that each function e_β can be approximated closely by a linear combination of the first N principal eigenfunctions, $b_1, ..., b_N$, for a suitably chosen N. That is for every β and f,

$$e_\beta(f) \approx \sum_{n=1}^{N} c_n(\beta) b_n(f), \tag{3.7}$$

where each function $c_n : [0, A] \to \mathbb{R}$ is defined as follows:

$$c_n(\beta) = \int_0^F b_n(f) e_\beta(f) df \tag{3.8}$$

for all $\beta \in [0, A]$; that is, thinking of functions as vectors, each scalar value $c_n(\beta)$ is the component of vector e_β in the direction of vector b_n. For practicality, each $c_n(\beta)$ can be closely approximated using a discrete-domain summation, and stored in a lookup table for use thereafter in all e_β approximations.

These linear combinations of the N chosen eigenfunctions can be used to approximate the $\exp(-a(\|\mathbf{x}-\mathbf{y}\|)f)$ part of eq. (3.4), and hence they form an approximate basis for the space of possible sensitivity-versus-frequency curves. A test of a very large number of random values of β has shown empirically that, with $N=6$, $H=240$, $W=360$ and $d=3H$, the worst root-mean-squared error of any of the approximated e_β functions was roughly 0.0001. These first six eigenfunctions are depicted in Fig. 3.3.

Substituting eq. (3.7), combined with eq. (3.5), with $\beta = a(\|\mathbf{x}-\mathbf{y}\|)$, into eq. (3.4), gives the following:

$$S_{\mu,\mathbf{x}}(f) \approx \sum_{\mathbf{y}\in D} \mu(y) \sum_{n=1}^{N} c_n(a(\|\mathbf{x}-\mathbf{y}\|))\, b_n(f),$$

for all f, μ and $\mathbf{x}$, which can be written as follows:

$$\begin{aligned} S_{\mu,\mathbf{x}}(f) &\approx \sum_{\mathbf{y}\in D} \mu(\mathbf{y}) \sum_{n=1}^{N} C_n(\mathbf{x}-\mathbf{y})\, b_n(f) \\ &= \sum_{n=1}^{N} b_n(f) \sum_{\mathbf{y}\in D} \mu(\mathbf{y})\, C_n(\mathbf{x}-\mathbf{y}) \\ &= \sum_{n=1}^{N} b_n(f)\, (\mu * C_n)(\mathbf{x}), \end{aligned}$$

Algorithm 3.1: Compute single sensitivity $\gamma' = S_{\mu,\mathbf{x}}(f)$

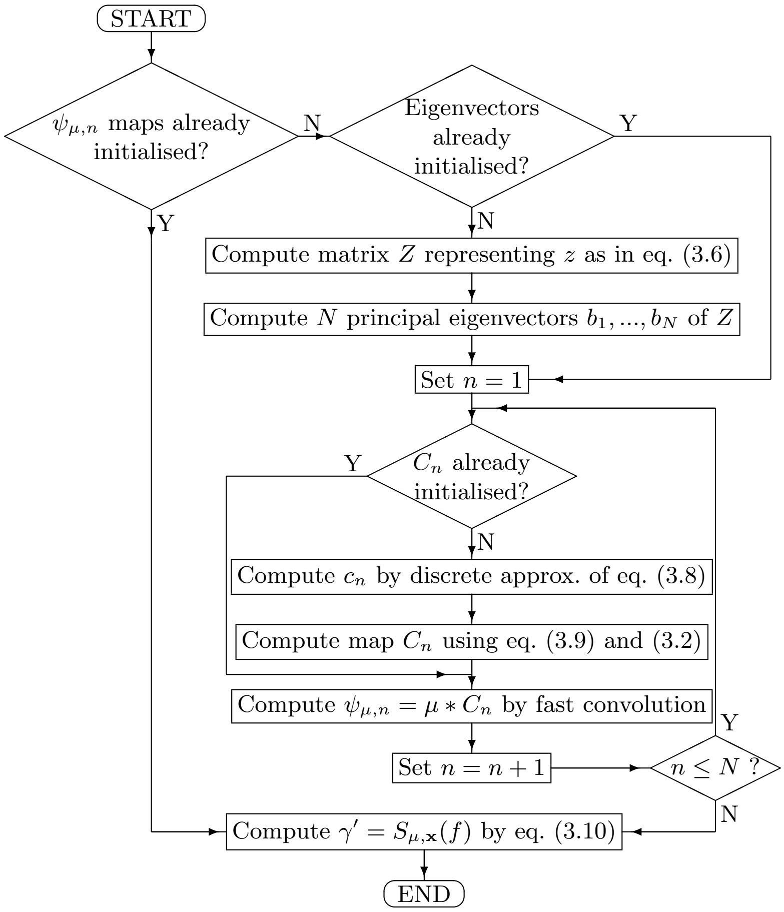

START
$\psi_{\mu,n}$ maps already initialised?
N
Y
Eigenvectors already initialised?
Y
N
Compute matrix Z representing z as in eq. (3.6)
Compute N principal eigenvectors $b_1, ..., b_N$ of Z
Set $n = 1$
Y
C_n already initialised?
N
Compute c_n by discrete approx. of eq. (3.8)
Compute map C_n using eq. (3.9) and (3.2)
Compute $\psi_{\mu,n} = \mu * C_n$ by fast convolution
Y
Set $n = n + 1$
$n \leq N$?
N
Compute $\gamma' = S_{\mu,\mathbf{x}}(f)$ by eq. (3.10)
END

Figure 3.4: Computing a single sensitivity value γ' for a given frequency f, location $\mathbf{x}$ and saliency map μ. Here, N is the (fixed) number of eigenvectors to be used. Note that once the coefficient maps, $\psi_{\mu,n}$ (for all $n \in \{1, ..., N\}$) have all been created for a given saliency map μ, the procedure always jumps straight to eq. (3.10).

where " * " denotes convolution and each 2-D-domained function $C_n : \breve{D} \rightarrow \mathbb{R}$ is defined in terms of the corresponding 1-D-domained function c_n as

$$C_n(\mathbf{w}) = c_n(a(\|\mathbf{w}\|)), \tag{3.9}$$

for all $\mathbf{w} \in \breve{D}$, where $\breve{D} = \{\mathbf{x}–\mathbf{y} : \mathbf{x}, \mathbf{y} \in D\}$ is an extended version of the image domain D. Therefore,

$$S_{\mu,\mathbf{x}}(f) \approx \sum_{n=1}^{N} b_n(f)\, \psi_{\mu,n}(\mathbf{x}), \tag{3.10}$$

for all f, μ and x, where each coefficient map $\psi_{\mu,n} : D \rightarrow \mathbb{R}$ is defined as a convolution

$$\psi_{\mu,n} = \mu * C_n. \tag{3.11}$$

Each of these N convolutions, which can be performed using a fast convolution technique [131, p. 449], needs only to be done once for each saliency map μ and thereafter stored in a look-up table, after which the same $\psi_{\mu,n}$ maps will be looked up and used for the computation of each sensitivity value $S_{\mu,\mathbf{x}}(f)$ for a given location $\mathbf{x}$ and frequency f. Note also that each eigenfunction b and map C_n only need to be computed once for given values of W, H and d, and thereafter re-used for all saliency maps of size $W \times H$ without any need for recomputation. The overall algorithm for computing a single $S_{\mu,\mathbf{x}}(f)$ value is summarised in Fig. 3.4.

Note that applying eq. (3.10) for a single sensitivity f and location $\mathbf{x}$ will cost only N look-up operations (one for each function b_n), N multiplications and $N-1$ additions, so is an order $\mathcal{O}(N)$ operation. However, the dominant part of the computation will be the computation of the maps $\psi_{\mu,1}, ..., \psi_{\mu,N}$, each of which will be of order $\mathcal{O}(HW \log_2(HW))$, but which only need to be done once for each saliency map μ. Therefore, the cost per pixel is of order $\mathcal{O}(N \log_2(HW))$, which can be regarded as $\mathcal{O}(\log_2(HW))$ because N is fixed ($N=6$ was used for the work herein.) This compares with $\mathcal{O}(HW)$ per pixel for a verbatim implementation of eq. (3.4).

3.2.3 Computing A Cut-Off Frequency Map

Consider the solution f to the equation $\gamma = S_{\mu,\mathbf{x}}(f)$ for a given location $\mathbf{x}$, satisfying some given overall sensitivity level γ. If this were to be solved for every $\mathbf{x} \in D$, the result would be a spatial map of frequencies of this given sensitivity. Combining this with the knowledge that each sensitivity function $S_{\mu,\mathbf{x}} : [0, \infty) \to [0, 1]$ is strictly decreasing, this map can be interpreted as a spatial map $\phi_{\mu,\gamma}$ of the lowest frequencies that have lower sensitivity than γ, defined as follows: $\phi_{\mu,\gamma}(\mathbf{x}) = S^{-1}_{\mu,\mathbf{x}}(\gamma)$ for all $\mathbf{x} \in D$. Thus, for lossy coding purposes, each map $\phi_{\mu,\gamma}$ is optimal in the sense of discarding least-noticeable information first. Given a forward computation for function $S_{\mu,\mathbf{x}}$, the inverse function $S^{-1}_{\mu,\mathbf{x}}$ can be computed by an inversion technique such as a binary search (also known as *bisection* [131, p. 277]). Fig. 3.5 summarises the algorithm for performing this computation, to a desired accuracy ϵ.

Calculating $\phi_{\mu,\gamma}$ given μ and γ will only be useful in conjunction with a mechanism for choosing a sensitivity value γ. From an image or video coding perspective, an appropriate approach would be to aim for a given percentage, λ, of the spatio-frequency components

Algorithm 3.2: Compute cut-off map $\phi_{\mu,\gamma}$

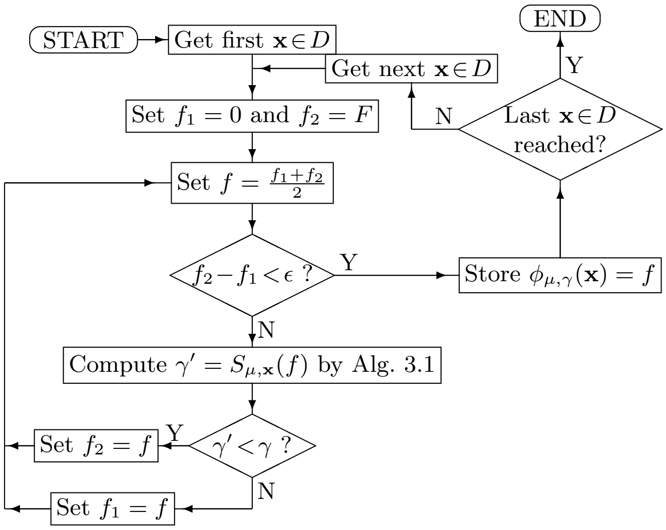

START
Get first $\mathbf{x}\in D$
END
Get next $\mathbf{x}\in D$
Y
Set $f_1 = 0$ and $f_2 = F$
N
Last $\mathbf{x}\in D$ reached?
Set $f = \frac{f_1+f_2}{2}$
$f_2 - f_1 < \epsilon$?
Y
Store $\phi_{\mu,\gamma}(\mathbf{x}) = f$
N
Compute $\gamma' = S_{\mu,\mathbf{x}}(f)$ by Alg. 3.1
Set $f_2 = f$
Y
$\gamma' < \gamma$?
N
Set $f_1 = f$

Figure 3.5: Computing cut-off frequency map $\phi_{\mu,\gamma}$ given saliency map μ and threshold sensitivity γ, incorporating a binary search which inverts sensitivity function $S_{\mu,\mathbf{x}}$ to compute each $S_{\mu,\mathbf{x}}(f)$ value to a given target accuracy ϵ. Here, F is the maximum representable frequency (see eq. (3.5)) and each $\mathbf{x}$ is an image location, from the set D (see eq. (3.4)).

to be discarded. That is, to compute $\phi_{\mu,\sigma_\mu(\lambda)}$ where $\sigma_\mu(\lambda)$ is the sensitivity value that gives cut-off percentage λ of hypothetically infinitessimally-small uniformly-distributed spatio-frequency bins. The proposed approach does this by converting each cut-off frequency $\phi_{\mu,\gamma}(\mathbf{x})$ into a percentage, $p(\phi_{\mu,\gamma}(\mathbf{x}))$ of the possible frequency bins at location $\mathbf{x}$, which is summed to obtain the total cut-off percentage $P_\mu(\gamma)$, and this is repeated for a number of different values of γ in a binary search to home in on the γ that gives the desired percentage λ. Here, function $P_\mu : [0,1] \to [0,100]$ is defined as

$$P_\mu(\gamma) = \frac{1}{|D|}\sum_{\mathbf{x}\in D} p(\phi_{\mu,\gamma}(\mathbf{x})), \tag{3.12}$$

for all $\gamma \in [0,1]$, where $|D|$ denotes the number of pixels in image domain D, and $p : [0,\infty) \to [0,100]$ is a function which converts from a cut-off frequency into a discarded-frequency percentage, taking into account the fact that the frequency space is two-dimensional and that each scalar cut-off frequency $f \in [0,F]$ (where F is as defined for equation 3.5) defines a two-dimensional locus (circular in most cases) of frequency bins of higher frequency. Consider the hypothetical situation whereby the localised frequency space at each image location has the same resolution as the non-localised frequency space, $\tilde{D}$, of

Algorithm 3.3: Compute λ-percent cut-off map $\phi_{\mu,\sigma_\mu(\lambda)}$

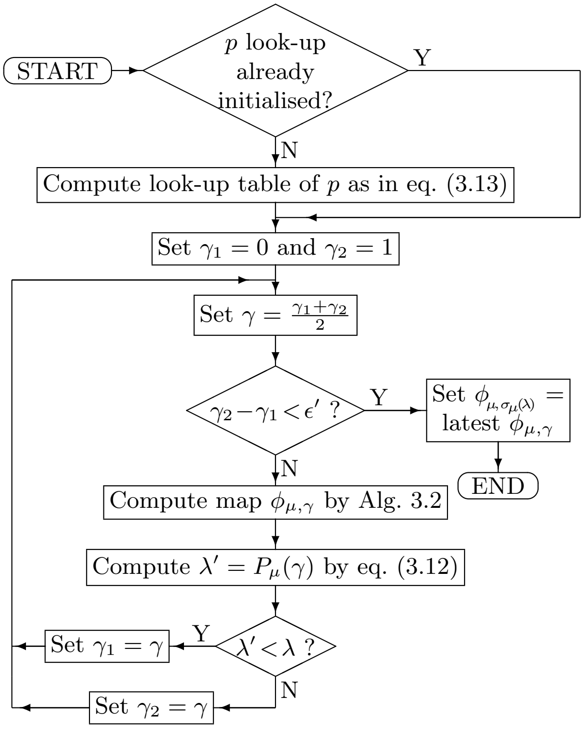

START
p look-up already initialised?
Y
N
Compute look-up table of p as in eq. (3.13)
Set $\gamma_1 = 0$ and $\gamma_2 = 1$
Set $\gamma = \frac{\gamma_1+\gamma_2}{2}$
$\gamma_2 - \gamma_1 < \epsilon'$?
Y
Set $\phi_{\mu,\sigma_\mu(\lambda)} =$ latest $\phi_{\mu,\gamma}$
END
N
Compute map $\phi_{\mu,\gamma}$ by Alg. 3.2
Compute $\lambda' = P_\mu(\gamma)$ by eq. (3.12)
$\lambda' < \lambda$?
Y
Set $\gamma_1 = \gamma$
N
Set $\gamma_2 = \gamma$

Figure 3.6: Computing a spatial map, $\phi_{\mu,\sigma_\mu(\lambda)}$, of cut-off frequencies, $\phi_{\mu,\sigma_\mu(\lambda)}(\mathbf{x})$ (for each $\mathbf{x} \in D$), given saliency map μ and target cut-off percentage λ, by using a binary search to invert function P_μ (see eq. (3.12)), to a given target accuracy ϵ'.

the image as a whole, and thereby define

$$p(f) = 100 \frac{\left|\left\{\xi \in \tilde{D} : \|\xi\| > f/F\right\}\right|}{|\tilde{D}|}, \tag{3.13}$$

for all $f \in [0, F]$, where $\tilde{D}$ represents the set of discrete-frequency space bins (each represented by a pair of numbers of cycles per pixel, horizontally and vertically) associated with image domain D; for example, if $D = \{0, ..., 100\} \times \{0, ..., 50\}$ then $\tilde{D} = \frac{\{^-50,...,^+50\}}{101} \times \frac{\{^-25,...,^+25\}}{51}$. As the function p will be fixed with respect to the size and shape of image domain D, it can be computed as a look-up table, which only needs to be computed once, which can be done using a simple verbatim interpretation of eq. (3.13). Each sensitivity value $\sigma_\mu(\lambda)$ can then be regarded simply as $P_\mu^{-1}(\lambda)$, which can be computed by the binary search as described, but in practice it makes more sense to directly compute $\phi_{\mu,\sigma_\mu(\lambda)}(\mathbf{x})$ (that is, $S_{\mu,x}^{-1}(P_\mu^{-1}(\lambda), \mathbf{x})$) for the given saliency map μ and cut-off percentage λ. The overall algorithm for performing this computation, to a given target accuracy ϵ', is

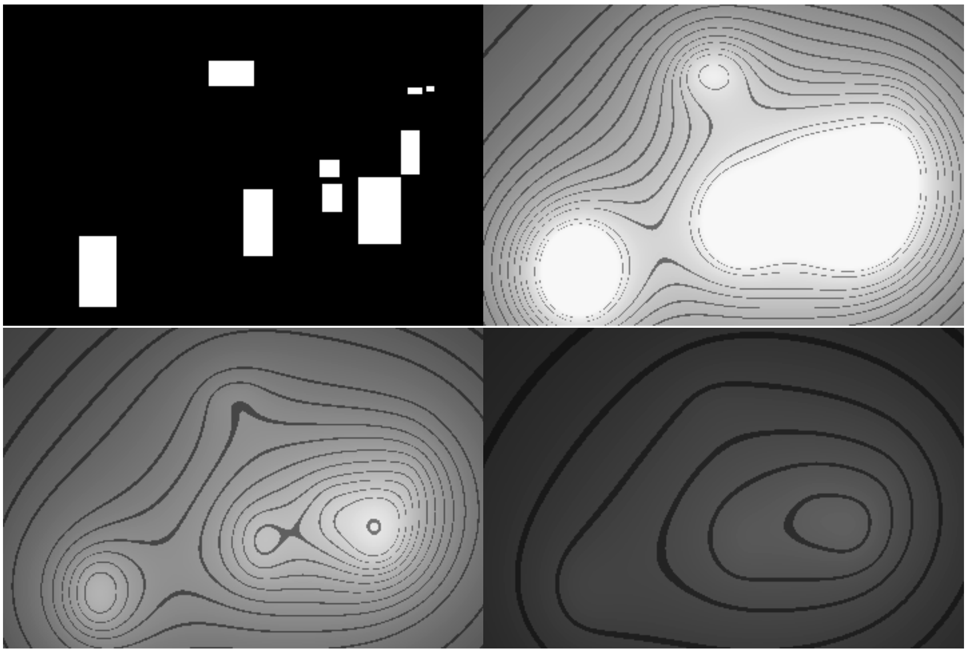


Figure 3.7: Example cut-off frequency maps. Top left: the saliency map (output of a multi-target visual tracker [132]). Top right: blur level 20.9% ($s = 0.037$). Bottom left: blur level 55.7% ($s = 0.096$). Bottom right: blur level 90.6% ($s = 0.296$). In each cut-off frequency map, the cut-off frequency is represented by the grey level of each pixel (maximum grey level = 247). Contour lines have been overlaid for illustrative purposes.

summarised in Fig. 3.6. Example cut-off frequency maps, as produced by this algorithm when applied to an object detection mask (in lieu of a saliency map) are shown in Fig. 3.7. These give a clear indication of how the proposed approach will provide a smooth transisition of blur level.

3.2.4 Example Output

In this subsection, example cut-off frequency maps generated by the proposed technique are presented, alongside the equivalent maps generated by that of Sheikh *et al.* [24], which, in simple terms, works by taking, for each image location, the highest assumed cut-off frequency of any of the viewers, with the cut-off frequency of each viewer controlled by a tuning parameter. To produce these cut-off maps, fixation points were used, as collected using an eye tracker from 16 subjects independently viewing three video sequences. Of these sequences, three video frames were used as shown in Fig. 3.8 with the corresponding fixation points highlighted.

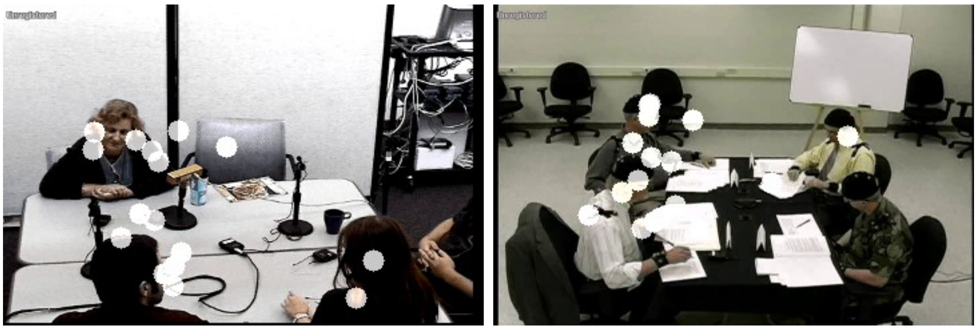


Figure 3.8: Test frames, from CLEAR 2006 video dataset [73] with fixation points depicted by large white circles. Left: frame of sequence "CMU_2_cam3_1". Right: frame of sequence "VT_2_cam2_1". All frames were 720 × 480 when viewed by the subjects but averaged and subsampled to 360 × 240 for convenience for these experiments.

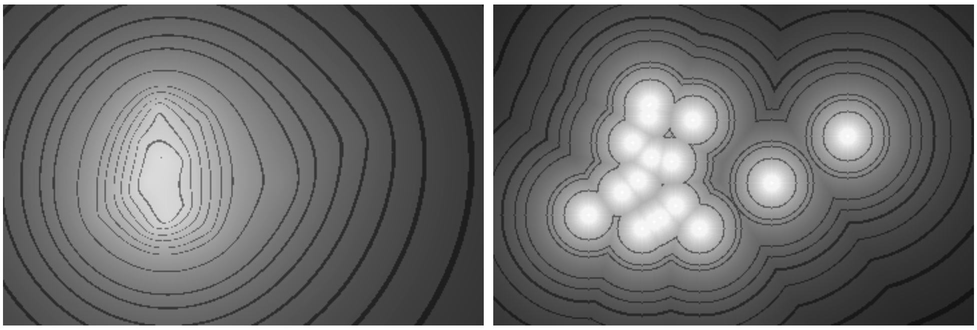


Figure 3.9: Comparison of a cut-off frequency map from the method presented herein with the equivalent from the continuous variant of the Sheikh *et al.* method. Maximum white represents the maximum displayable frequency; black represents zero. Contour lines have been added for clarity. The fixation points are as shown in Fig. 3.8 (right). Both maps were tuned to cut-off 70% of the frequency space.

The frame dimensions of each original sequence were 720 × 480, which were scaled down to a quarter frame of 360 × 240, to allow the test sequence to be comparable with a more common frame size of 352 × 288 (CIF). For all techniques, a viewing distance of three times the frame height (i.e., a distance of 720 pixel widths) was assumed. The technique of Sheikh *et al.* was implemented exactly as published, with the original suggested parameters of block width 16 (intended for use in a DCT-block-based coding scheme) and 8 quantisation levels (intended to allow computational speed). Also, to enable closer comparison with a technique closer to the approach proposed herein, a continuous version of their technique was tried, by reducing their block width to 1 and increasing the number of quantisation levels to a large number, effectively allowing a continuous range of blurring levels. Also for consistency with the approach proposed herein, their hard-limited maximum frequency

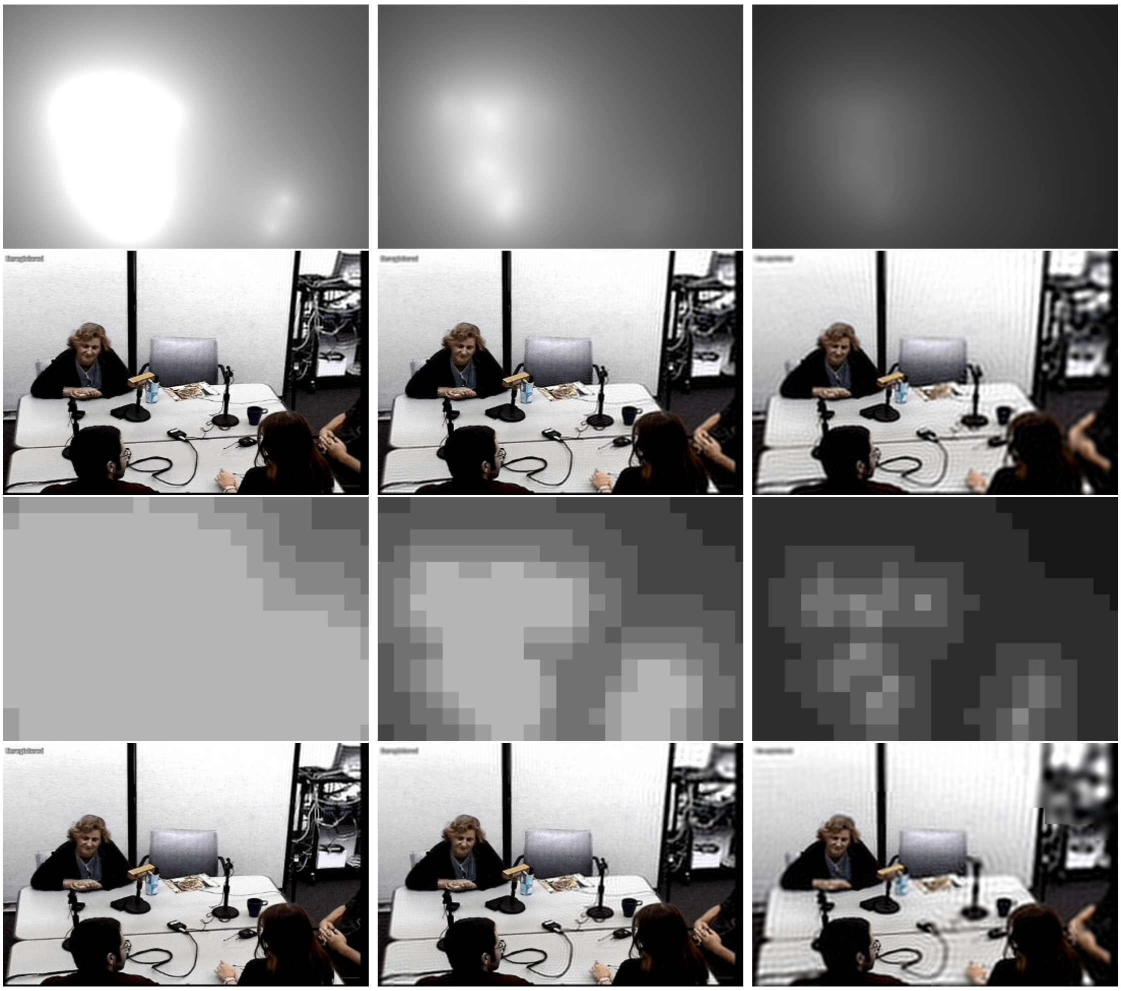


Figure 3.10: Example cut-off frequency maps for a frame of sequence "CMU_2_cam3_1" [73], with correspondingly blurred frames underneath. In each map, the cut-off frequency is represented by the grey level of each pixel (maximum white represents the maximum displayable frequency; black represents zero). Left to right: 30%, 60% and 90% blurring. Rows 1 & 2: generated by the approach proposed herein; rows 3 & 4: by Sheikh *et al.*

was increased from 0.5 cycles per pixel-width to $\sqrt{0.5}$ cycles per pixel-width, thus allowing the full range of possible 2-D spatial frequency vectors. To obtain a desired overall cut-off percentage from both versions of their technique, the same binary search as presented in Algorithm 3.3 (Fig. 3.6) was employed, by substituting their technique in place of Algorithm 3.2.

In conjunction with the cut-off maps, the corresponding blurred frames are shown for demonstration purposes. These were computed by separately blurring each frame with a range of different cut-off frequencies, and then taking, at each location, the pixel from

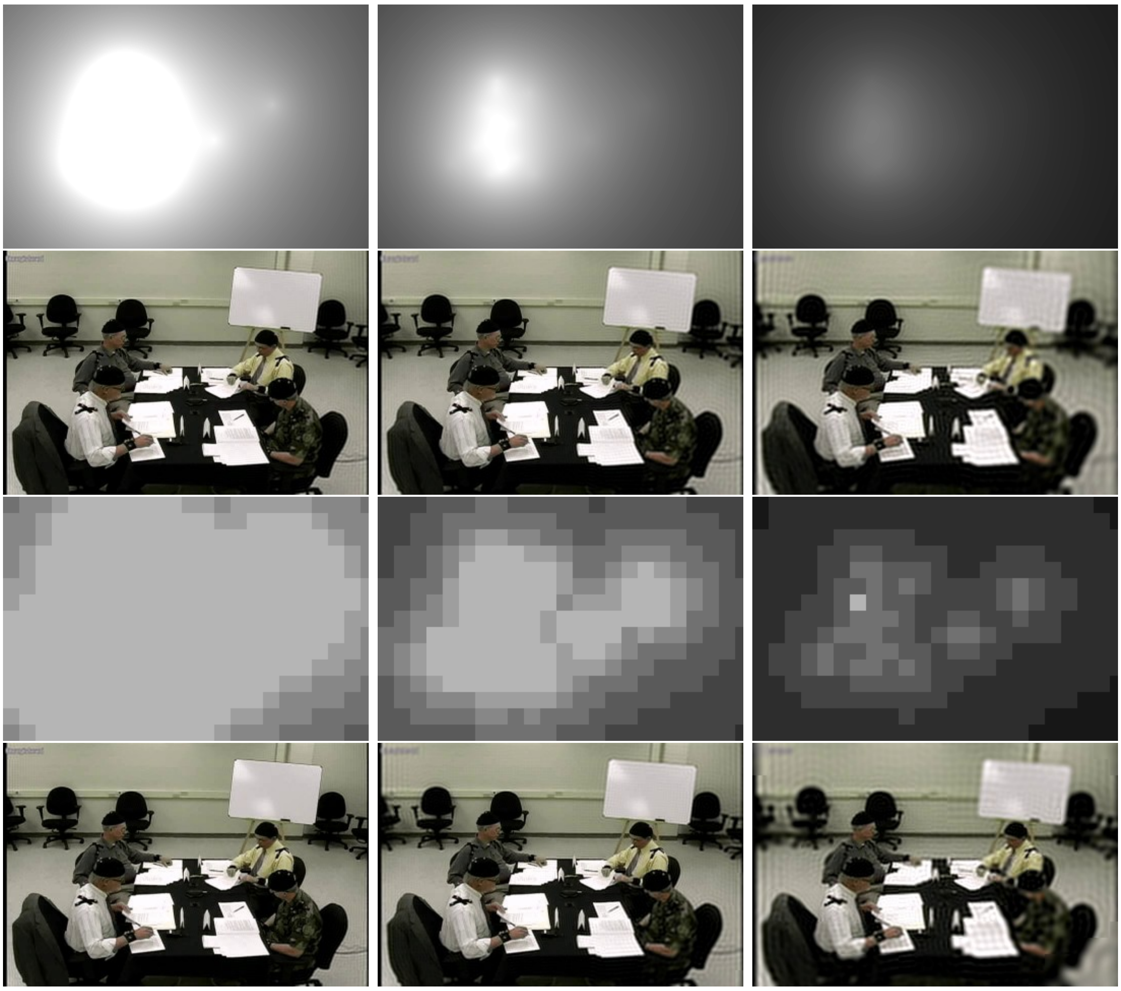


Figure 3.11: Example cut-off frequency maps for a frame of sequence "VT_2_cam2_1" [73], with correspondingly blurred frames underneath. In each map, the cut-off frequency is represented by the grey level of each pixel (maximum white represents the maximum displayable frequency; black represents zero). Left to right: 30%, 60% and 90% blurring. Rows 1 & 2: generated by the approach proposed herein; rows 3 & 4: by Sheikh *et al.*

the blurred frame of the cut-off frequency of that location in the cut-off map. The cut-off frequencies were rounded to the nearest whole number of cycles per 361 pixel-widths, such that the range of possible frequencies were represented by the quantized range $[0, 255]$ (note that $255 \times \sqrt{2} \approx 360.62$). The blurring of each frame was performed by applying a hard cut-off in frequency space to remove all frequency vectors greater than the given frequency magnitude while leaving all others unchanged. This type of blurring incurs a heavy ringing effect, as is visible in the results, but it is an appropriate demonstration of the perfect application of a variable cut-off frequency.

Figs 3.10 and 3.11 compare the proposed technique with the published Sheikh *et al.* method, on all three of the video frames used. The blocky and quantised nature of the Sheikh *et al.* technique as published can clearly be seen.

Fig. 3.9 provides a comparison of a cut-off frequency map from the proposed method with the equivalent from the continuous variant of the Sheikh *et al.* method highlighting the nature of the latter method as effectively partitioning video frame according to the nearest fixation point to each location, and separately computing a single-viewer cut-off map for each location. This is also illustrated by the cut-off maps in Figs 3.10 and 3.11.

A key issue in deciding which technique provides the more appropriate cut-off map is that of how they handle outlying fixation points. Figs 3.10 and 3.11 both show the effect of a scenario in which one or two fixation points outlying towards the right of the image. The proposed approach appears almost to neglect such points, with the main body of preserved frequencies surrounding the dominant cluster of fixations, whereas the Sheikh *et al.* technique effectively boosts the significance of the outliers, providing a more widely-spread region of higher resolution.

Because the proposed approach concentrates the frequencies into a narrower area, the average resolution over this area will be higher. Also, the proposed approach has a smoother variation in frequency, thus reducing the chances that local variation in blur level may itself be observed as an undesirable compression artifact.

Another possible advantage of the proposed technique is that, due to its robustness to outliers, the subsequent blur map itself is likely to incur a lower level of temporal variation. This would have the benefit not only of removing what may itself be an observable artifact but also of alleviating a known problem in foveated video coding, whereby prefiltering with a time-varying blur map may have a diverse effect on the motion compensated prediction.

The question of which approach is the more correct depends on the aim of the foveation filtering. In simplistic terms, if the aim is to cater for the worst case scenario at the expense of the majority, then the Sheikh *et al.* technique is more appropriate, whereas if the aim is to cater for the majority at the expense of a small number of individuals, then the technique proposed herein is more appropriate.

This section has focussed on the generation of blur maps for use in a preprocessing stage in a foveated image or video encoder. The next section focusses on the computation of blurring using a given blur map.

3.3 Smoothly Space-Variant Gaussian Blurring

As mentioned in section 2.5, space-variant, or *selective*, blurring is an image processing effect whereby different parts of an image or video frame are blurred to different extents. This section addresses the issue of how to perform such blurring using Gaussian blurring. It presents an algorithm which bridges the cost-versus-accuracy performance gap between faster, less accurate approaches such as blended Gaussian Pyramid [102, 103], and the prohibitively slow, high-quality approach of a different low-pass filter at each location [4].

This section is organised into two subsections: subsection 3.3.1 presents the algorithm, and subsection 3.3.2 evaluates and compares the accuracy and cost of the algorithm.

3.3.1 Proposed Approach

Core algorithm

Let $\mathbf{I}$ be a W-by-H colour image and b a blur map of the same size. Let $\mathbf{S}_b$ be a space-variant Gaussian-blurred version of $\mathbf{I}$. Fig. 3.12 gives a block diagram of the proposed approach. The image and blur map are functions such that $\mathbf{I}, \mathbf{S}_b : D \to \mathbb{R}^3$ and $b : D \to \mathbb{R}$, where domain $D = \{(x_1, x_2) \in \mathbb{Z}^2 : 0 \leq x_1 < W, 0 \leq x_2 < H\}$ is the set of possible pixel locations and $\mathbb{Z}$ is the set of all integers. Consider also a domain $\breve{D}$, of size L-by-L, centred around zero, and defined as $\breve{D} = \{(x_1, x_2) \in \mathbb{Z}^2 : \lceil \frac{-L}{2} \rceil \leq x_1, x_2 < \lceil \frac{L}{2} \rceil\}$, where $\lceil$ and $\rceil$ denote upward integer rounding. Consider a Gaussian point spread function (PSF), $G_\sigma : \breve{D} \to \mathbb{R}$, defined for all $\mathbf{x} = (x_1, x_2) \in \breve{D}$ as follows:

$$G_\sigma(x_1, x_2) = \begin{cases} k_\sigma \exp\left(\frac{x_1^2 + x_2^2}{-2\sigma^2}\right) & \text{if } \sigma \neq 0 \\ \delta(x_1^2 + x_2^2) & \text{if } \sigma = 0, \end{cases} \tag{3.14}$$

where δ is the delta function and each normalised constant k_σ is defined as

$$k_\sigma \;=\; 1 \Bigg/ \sum_{(x_1, x_2) \in \breve{D}} \exp(-(x_1^2 + x_2^2)/2\sigma^2)\;. \tag{3.15}$$

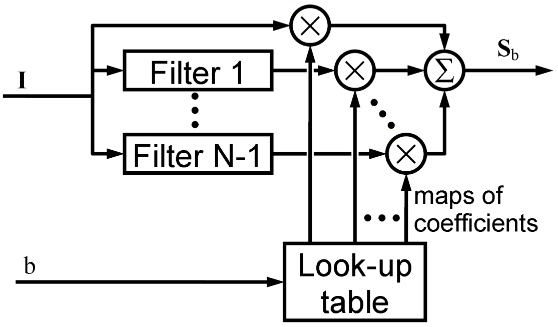


Figure 3.12: Block diagram of the proposed Gaussian blurring approach. The input image, **I** is separately convolved with a number of predefined filters and each convolved image is multiplied pixel-by-pixel by a spatial map of coefficients and the results are summed, giving the output image, $\mathbf{S}_b$. Each coefficient, at each pixel location, is given by a predefined look-up table according to the value in the blur map, b, at that location. The proposed approach is a filter bank method, except that normal filter banks use bandpass or lowpass filters, whereas the proposed approach uses specially-derived basis functions for the space of Gaussian functions.

Consider image $\mathbf{U}_\sigma : D \to \mathbb{R}^3$, defined as a uniformly Gaussian-blurred version of **I**, as follows:

$$\mathbf{U}_\sigma(\mathbf{x}) = (\mathbf{I} * G_\sigma)(\mathbf{x}) = \sum_{\mathbf{y}\in\breve{D}} \tilde{\mathbf{I}}(\mathbf{x}-\mathbf{y})G_\sigma(\mathbf{y}), \tag{3.16}$$

for all $\mathbf{x}\in D$, where $\mathbf{I}*G_\sigma$ is the symmetric convolution of **I** with G_σ and $\tilde{\mathbf{I}}$ is the symmetric extension of **I** over the whole of $\mathbb{Z}^2$. In precise terms, $\tilde{\mathbf{I}}(2pW-\frac{1}{2}\pm(x_1+\frac{1}{2}),\ 2qH-\frac{1}{2}\pm(x_2+\frac{1}{2})) = \mathbf{I}(x_1, x_2)$ for all $(x_1, x_2)\in D$ and all $p, q\in\mathbb{Z}$.

Now, given the blur map $b : D \to \mathbb{R}$, define image $\mathbf{S}_b : D \to \mathbb{R}^3$, a space-variant Gaussian-blurred version of **I**, as follows:

$$\begin{aligned}\mathbf{S}_b(\mathbf{x}) &= \mathbf{U}_{b(\mathbf{x})}(\mathbf{x}) && (3.17)\\ &= \sum_{\mathbf{y}\in\breve{D}} \tilde{\mathbf{I}}(\mathbf{x}-\mathbf{y})G_{b(\mathbf{x})}(\mathbf{y}). && (3.18)\end{aligned}$$

Here, $\mathbf{U}_{b(\mathbf{x})}$ is the uniformly-blurred image as given by Eq. (3.16) and $G_{b(\mathbf{x})}$ is the PSF as given by Eq. (3.14), but with $b(\mathbf{x})$ substituted for σ in both cases.

To compute each $\mathbf{S}_b(\mathbf{x})$ value by interpreting Eq. (3.18) verbatim would be prohibitive, as this would require a sum of L^2 terms for every pixel. However, a close approximation of this computation can be performed using a faster approach which will now be described.

Given minimum and maximum blur levels, $m \in \mathbb{R}$ and $M \in \mathbb{R}$, consider a family, $\Gamma = \{G_\sigma : \sigma\in[m, M]\}$, of Gaussian PSFs. Consider the equivalent family, $\breve{\Gamma} = \{\breve{G}_\sigma : \sigma \in$

$[m, M]\}$ of PSFs that have been modified to be orthogonal to the delta function, δ. That is, for all $\mathbf{x} \in \breve{D}$,

$$\breve{G}_\sigma(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} = \mathbf{0} \\ G_\sigma(\mathbf{x}) & \text{if } \mathbf{x} \neq \mathbf{0}. \end{cases} \tag{3.19}$$

Consider a tensor $Z : \breve{D} \times \breve{D} \to \mathbb{R}$, defined as follows:

$$Z(\mathbf{x}, \mathbf{y}) = \int_m^M \frac{\breve{G}_\sigma(\mathbf{x})\breve{G}_\sigma(\mathbf{y})}{\sigma} d\sigma, \tag{3.20}$$

for all $\mathbf{x}, \mathbf{y} \in \breve{D}$. Note that Z is symmetric with respect to its arguments; that is, $Z(\mathbf{x}, \mathbf{y}) = Z(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}, \mathbf{y} \in \breve{D}$. Since the set $\breve{D}$ is finite (with L^2 members), Z can be handled numerically as a matrix. This matrix, being symmetric, can be diagonalised and has an orthonormal basis of eigenvectors [100, p. 459]; that is

$$Z(\mathbf{x}, \mathbf{y}) = \sum_{n=1}^{L^2} \beta_n(\mathbf{x}) \lambda_n \beta_n(\mathbf{y}), \tag{3.21}$$

for some eigenvalues $\lambda_1, ..., \lambda_{L^2} \in \mathbb{R}$ and some orthonormal set of eigenfunctions, $\beta_1, ..., \beta_{L^2} : \breve{D} \to \mathbb{R}$. It happens that all except a small number of the eigenvalues of Z are very close to zero, the result being that each $\breve{G}_\sigma \in \breve{\Gamma}$ can be approximated closely by a linear combination of the first few eigenfunctions (assuming the eigenfunctions to be arranged in descending order of eigenvalue). Furthermore, suppose that this basis is extended by adding an extra function β_0, defined to be the delta function; that is

$$\beta_0(\mathbf{x}) = \delta(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{0} \\ 0 & \text{if } \mathbf{x} \neq \mathbf{0} \end{cases} \tag{3.22}$$

for all $\mathbf{x} \in \breve{D}$. Recall that each $\breve{G}_\sigma$ is orthogonal to β_0, and hence so are the $\beta_1, \beta_2, ...$ which span the space of $\breve{G}_\sigma$ functions. Now, each Gaussian PSF $G_\sigma \in \Gamma$ can be approximated by a linear combination of the first N basis functions, $\beta_0, ..., \beta_{N-1}$, for a suitably chosen N. That is, for every $\sigma \in [m, M]$ and every $\mathbf{x} \in \breve{D}$,

$$G_\sigma(\mathbf{x}) \approx \sum_{n=0}^{N-1} c_n(\sigma) \beta_n(\mathbf{x}), \tag{3.23}$$

where each coefficient function $c_n : [m, M] \to \mathbb{R}$ is defined as follows:

$$c_n(\sigma) = \sum_{\mathbf{x} \in \breve{D}} \beta_n(\mathbf{x}) G_\sigma(\mathbf{x}) \tag{3.24}$$

for all $\sigma\in[m, M]$. That is, thinking of the functions as vectors, each scalar value $c_n(\sigma)$ is the component of vector G_σ in the direction of basis vector β_n. For each n and N, consider also a normalised coefficient function $\hat{c}_{n,N} : [m, M] \to \mathbb{R}$ defined as follows:

$$\hat{c}_{n,N}(\sigma) = c_n(\sigma) \bigg/ \sum_{\mathbf{x}\in\breve{D}} \sum_{n=0}^{N-1} c_n(\sigma)\beta_n(\mathbf{x}) \tag{3.25}$$

for all $\sigma \in [m, M]$. Note that because $\sum_{n=0}^{N-1} c_n(\sigma)\beta_n(\mathbf{x})$ converges to $G_\sigma(\mathbf{x})$ as $N \to \infty$, coupled with the fact that $\sum_{\mathbf{x}\in\breve{D}} G_\sigma(\mathbf{x})=1$, it is a fact that $\sum_{n=0}^{N-1} \hat{c}_{n,N}(\sigma)\beta_n(\mathbf{x})$ converges to $G_\sigma(\mathbf{x})$. Therefore,

$$G_\sigma(\mathbf{x}) \approx \sum_{n=0}^{N-1} \hat{c}_{n,N}(\sigma)\beta_n(\mathbf{x}), \tag{3.26}$$

for all $\mathbf{x} \in \breve{D}$. This is important because the use of $\hat{c}_{n,N}$ rather than c_n will ensure that the approximated Gaussian will always sum to unity.

Substituting Eq. (3.26), with $\sigma = b(\mathbf{x})$, into Eq. (3.18) gives the following:

$$\begin{aligned}
\mathbf{S}_b(\mathbf{x}) &\approx \sum_{\mathbf{y}\in\breve{D}} \tilde{\mathbf{I}}(\mathbf{x}-\mathbf{y}) \sum_{n=0}^{N-1} \hat{c}_{n,N}(b(\mathbf{x}))\beta_n(\mathbf{y}) \\
&= \sum_{n=0}^{N-1} \hat{c}_{n,N}(b(\mathbf{x})) \sum_{\mathbf{y}\in\breve{D}} \tilde{\mathbf{I}}(\mathbf{x}-\mathbf{y})\beta_n(\mathbf{y}) \\
&= \sum_{n=0}^{N-1} \hat{c}_{n,N}(b(\mathbf{x}))\,(\mathbf{I} * \beta_n)(\mathbf{x}),
\end{aligned}$$

where $\mathbf{I} * \beta_n$ is the symmetric convolution of $\mathbf{I}$ with β_n. Therefore,

$$\mathbf{S}_b(\mathbf{x}) \approx \sum_{n=0}^{N-1} \hat{c}_{n,N}(b(\mathbf{x}))\,\psi_n(\mathbf{x}), \tag{3.27}$$

where each filtered image $\psi_n : D \to \mathbb{R}$ is defined as a convolution, $\psi_n = \mathbf{I} * \beta_n$. That is,

$$\psi_n(\mathbf{x}) = \sum_{\mathbf{y}\in\breve{D}} \tilde{\mathbf{I}}(\mathbf{x}-\mathbf{y})\beta_n(\mathbf{y}). \tag{3.28}$$

The reason Z and its eigenfunctions are relevant is because these linear combinations of the N chosen eigenfunctions can be used in approximating the Gaussian PSFs, as stated in equations (3.23) and (3.26), and hence they form an approximate basis for the space, Γ, of possible PSFs. Effectively, this derivation of eigenfunctions is a form of principal component analysis on the set $\breve{\Gamma}$. The eigenvectors are the optimal basis of $\breve{\Gamma}$ in terms of providing the lowest expected value of the sum-of-squared-errors of the approximated

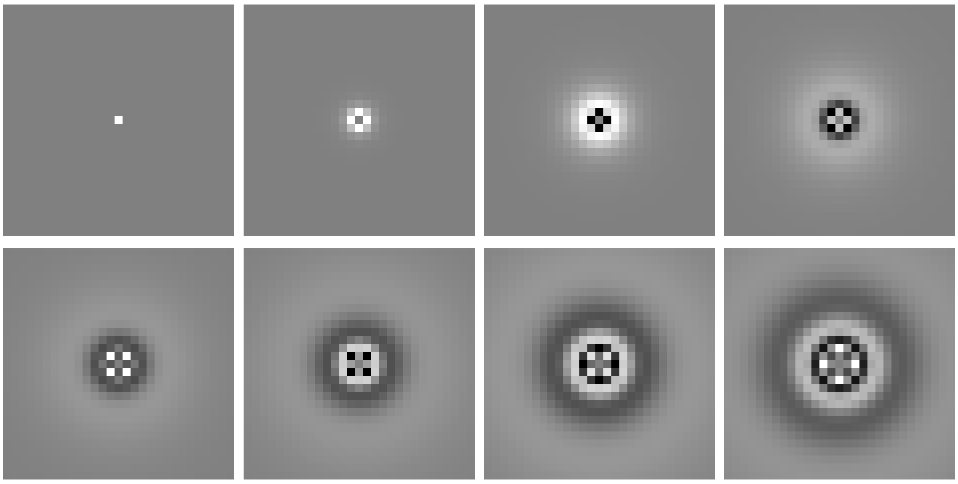


Figure 3.13: Top (left to right): first four basis functions of the family of Gaussian PSFs, when $L = 81$, $m = \frac{1}{3}$ and $M = 10$ (see Eq. (3.20)); bottom: next four basis functions. Mid-grey represents zero; dark shades represent negative; light shades represent positive.

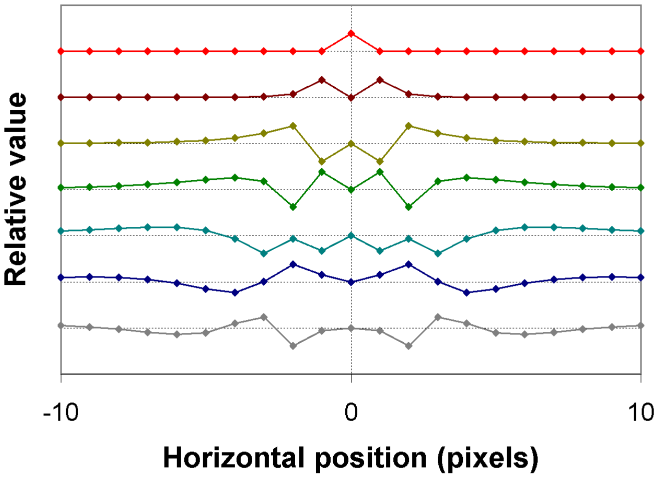


Figure 3.14: Horizontal cross-sections of the first seven basis functions (as shown in Fig. 3.13). To aid visibility, all functions have here been scaled to have the same maximum absolute value.

Gaussians (i.e., the lowest statistical mean squared error across a representative range of test images). This statistical perspective intrinsically involves an *a priori* probability assumption of the distribution of σ values. The assumption associated with the $1/\sigma$ factor in Eq. (3.20) is a log-uniform distribution of σ; that is, a uniform *a priori* probability of $\log(\sigma)$ values. This is desirable because over the very low sigma values, the Gaussian bell curves vary greatly between close σ values, whereas over the higher σ values, there is far less variation. The $1/\sigma$ factor in Eq. (3.20) is effectively a weighting factor which reduces the significance of higher σ values which would dominate the integral in Eq. (3.20) at the expense of the lower σ values. This reduces the worst-case error of approximated Gaussian curves. For example, applying the proposed approach with $N=8$ to a random 256×256 black & white image with a uniform blur map, with σ at different multiples of 0.1, gives a worst-case error of 49.4dB at $\sigma=0.2$. However, the worst-case error if the $1/\sigma$ factor is removed from Eq. (3.20) is 42.8dB, also at $\sigma=0.2$.

It has been found empirically that with $m = \frac{1}{3}$, $M = 10$, $L = 81$ and just 7 basis functions (i.e., $N=7$), the worst root-mean-squared error of any of the approximated G_σ functions is roughly 0.00005. This implies, in the trivial example of a greyscale image with all pixels zero except for one pixel with grey level 255, that the sum of square errors will be $0.00005^2 \times 81^2$, so the worst possible grey level error of any pixel in the blurred image cannot be greater than $255 \times 0.00005 \times 81 = 1.03275$. However, for general images with most pixels non-zero, the overall errors will be greater and will be dependent on the image itself, as shown by the results in section 3.3.2.

The first eight basis functions, when $m = \frac{1}{3}$ and $M = 10$, are represented in Figs 3.13 and 3.14. A graph showing the exponential nature of the descent of the eigenvalues is given in Fig. 3.15. This illustrates the nature of the space of eigenvector as being of low approximate dimensionality. In rough terms, each eigenvalue can be seen as the square of the width of the space when measured in the direction of the given eigenvector. The square error induced by discarding the least significant eigenvectors will, in general, be roughly proportional to the sum of the eigenvalues of these discarded eigenvectors. The mean ratio between any adjacent pair within the first 20 eigenvalues as shown is 4.39; i.e., the width of the space roughly halves in the direction of each new eigenvector.

Implementation details

Eq. (3.27) is the top-level stage of the proposed algorithm. Note that being a weighted sum of convolutions (i.e., filters) makes it a special instance of the filter bank method [4]. Because β_0 is the delta function, the first of these $\psi_n = \mathbf{I} * \beta_n$ convolutions, with $n = 0$, is the identity operation. That is, $\psi_0 = \mathbf{I}$, so no work needs to be done here. However, each of the remaining $N-1$ convolutions can be performed using any fast convolution technique [100, p. 538], using fast Fourier transforms, fast *number theoretic transforms*, or

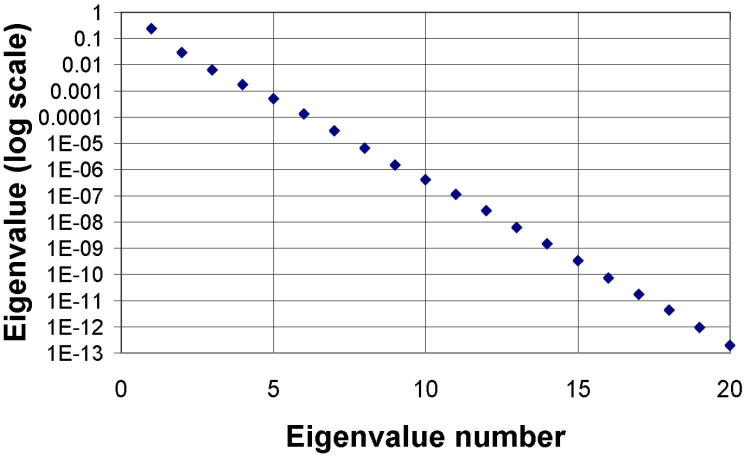


Figure 3.15: Plot of the first 20 eigenvalues of the computed basis functions. The vertical scale is logarithmic, clearly showing the exponential decrease of the eigenvalues, which corresponds with an exponential convergence of any approximated Gaussian generated by different numbers of basis functions.

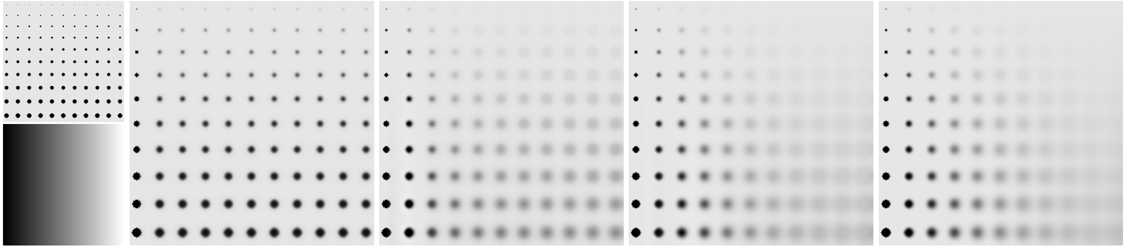


Figure 3.16: Examples of applying the proposed algorithm to a 256×256 synthetic image, with varying number, N, of basis functions employed. Top left: original image. Bottom left: blur map (σ increasing left to right from 0 to 10). Remaining images, left to right: output of the proposed approach with $N = 2, 4, 6, 8$. The most significant basis functions (with the highest eigenvalues) tend to correspond to the higher frequencies, which can be seen in the fact that the lower-blur regions in the above change less than the higher-blur regions as the number of basis functions increases.

fast discrete cosine transforms (DCTs). All these approaches in their simplest form restrict the input image to array dimensions which are exact powers of two, and therefore, in its simplest form as presented herein, the proposed approach follows the same restriction. The approach can be extended to other image sizes by zero padding of the image.

In the specific implementation reported in this paper, the convolution performed was a symmetric convolution, performed using *convolution form* DCTs [133]. Using Martucci's terminology [133], the aim was *half-sample symmetry* around image boundaries, and *whole-sample symmetry* around the point spread function's origin point. This requires involved applying a Type I DCT to a quadrant of the basis function and a Type II DCT to the image, followed by per-element multiplication then an inverse Type II DCT to obtain the convolved image. This involves converting each basis function β_n to a $(W+1)$-by-$(H+1)$ domain size by zero padding up to $x_1 = W$ and $x_2 = H$ and discarding $\beta_n(x_1, x_2)$ values for negative x_1 and x_2. This loses no information, since $\beta_n(x_1, x_2) = \beta_n(\pm x_1, \pm x_2)$ for all n, x_1 and x_2, due to every Gaussian PSF G_σ having the same symmetry. Each 2-D DCT was computed by applying the corresponding type of 1-D DCT firstly replacing each row of the image or PSF with its DCT then replacing each column with its DCT. Every 1-D DCT was performed using a verbatim implementation of the sparse matrix decompositions described in [134]. These classic DCT operations were converted into *convolution form* DCTs by appropriately weighting the array elements before and after the classic DCT, as prescribed in [133].

The proposed algorithm assumes that all the eigenfunctions have been precomputed and their Type I discrete cosine transforms have been stored (for the fast convolution stage), along with a look-up table approximation of each coefficient function, $\hat{c}_{n,N}$. The width L of the PSF domain $\breve{D}$ should be chosen to be sufficiently large that every G_σ function is sufficiently close to zero for the desired accuracy. In practice, $L \geq 6M$ is sufficient due to the fact that a Gaussian is near zero beyond three standard deviations from its mean, but in section 3.3.2 of this paper, $L = 81$ was used with $M = 10$. Furthermore, L should be an odd number so that every G_σ function has symmetry about its central point.

The eigenfunctions $\beta_1, \beta_2, \beta_2, ...$ were computed by diagonalisation of the matrix representation of Z, which was numerically approximated by a discrete summation equivalent of the integral in Eq. (3.20). Because of the $1/\sigma$ factor, this integral was approximated by summing $\breve{G}_\sigma(\mathbf{x})\breve{G}_\sigma(\mathbf{y})$ samples over a discrete range of σ values with density decreasing in proportion to $1/\sigma$. Specifically,

$$\int_m^M \frac{\breve{G}_\sigma(\mathbf{x})\,\breve{G}_\sigma(\mathbf{y})}{\sigma}\,d\sigma \approx \frac{\log_e(\frac{M}{m})}{Q+1}\sum_{q=0}^{Q}\breve{G}_{\sigma_q}(\mathbf{x})\,\breve{G}_{\sigma_q}(\mathbf{y}), \tag{3.29}$$

where each σ_q is defined as $\sigma_q = ((M/m)^{q/Q})m$. In the specific implementation of this paper, $Q = 99$ was used, so that each $Z(\mathbf{x}, \mathbf{y})$ computation was approximated by a sum of 100 terms.

The computation of Z and its eigenvectors are computationally expensive processes in which some savings are possible. As these are precomputed, this expense does not effect the cost of the core algorithm. However, a computational saving has been made for the purpose of this paper by reducing the $\breve{G}_\sigma$ arrays in size, by restricting each $\breve{G}_\sigma$ function to a one-eighth segment of the domain $\breve{D}$, so as to exploit the $\breve{G}_\sigma(x_1,x_2)=\breve{G}_\sigma(\pm x_1,\pm x_2)=\breve{G}_\sigma(\pm x_2,\pm x_1)$ symmetries of the Gaussian functions. This restriction was done before eigenvector decomposition. After this, the reverse process was applied to reconstruct the basis functions over the whole of $\breve{D}$, followed by normalisation of the basis. In order to produce exactly the same resulting eigenfunctions, it is necessary to apply appropriate weightings to each boundary point in proportion to the square root of the number of identical-shaped eighth-part segments which share the boundary point (and to apply the inverse weightings to the corresponding locations afterwards). That is, each function used was represented by a triangle of points as shown below:

$$\begin{pmatrix} \frac{1}{\sqrt{8}}g_{0,0} & & & & \\ \frac{1}{\sqrt{2}}g_{1,0} & \frac{1}{\sqrt{2}}g_{1,1} & & & \\ \frac{1}{\sqrt{2}}g_{2,0} & g_{2,1} & \frac{1}{\sqrt{2}}g_{2,2} & & \\ \frac{1}{\sqrt{2}}g_{3,0} & g_{3,1} & g_{3,2} & \frac{1}{\sqrt{2}}g_{3,3} & \\ \vdots & \vdots & \vdots & & \ddots \\ \frac{1}{\sqrt{2}}g_{h,0} & g_{h,1} & g_{h,2} & \cdots & \frac{1}{\sqrt{2}}g_{h,h} \end{pmatrix} \tag{3.30}$$

where each $g_{y,x}$ represents $\breve{G}_\sigma(x,y)$, and $h=(L-1)/2$, with L an odd number.

As an example application of the proposed algorithm, consider foveation filtering. In this scenario, the blur map would be a spatial map of cut-off frequencies, using knowledge of the point of human fixation combined with a model of the human visual system, such as a contrast threshold formula [28]. Each cut-off frequency f is then converted into a σ value by employing the common convention of treating the cut-off frequency of a filter as its 3dB point. This would give $\sigma = \sqrt{(}2\log_e(\sqrt{2}))/f$.

Fig. 3.16 shows example output of the algorithm on a synthetic image, for a range of numbers of basis functions employed. Fig. 3.17 shows example output of the algorithm on a real image in a vision research application. This allows normally-sighted people to visualise the effects of sight problems such as glaucoma. Fig. 3.18 shows example output in variable resolution rendering. With all these examples, the same set of basis functions

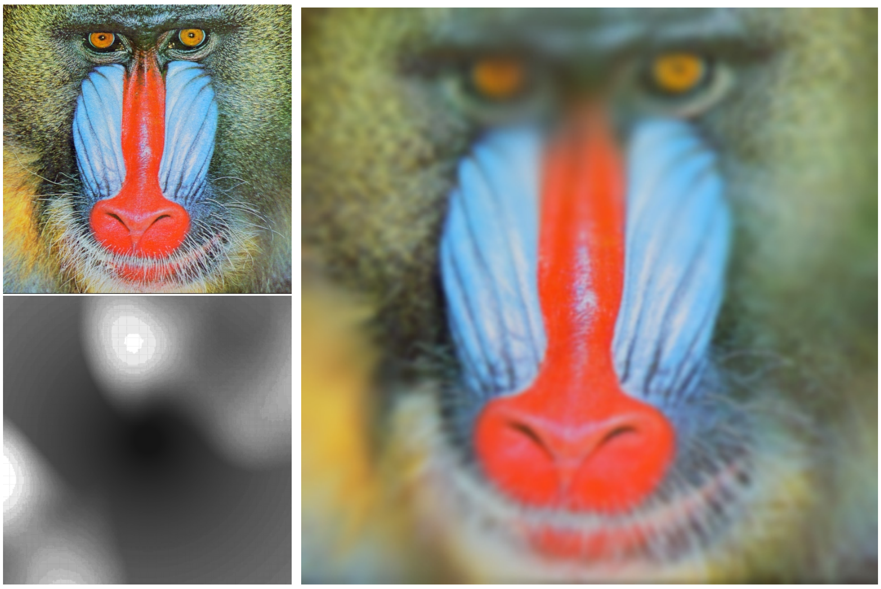


Figure 3.17: Top left: raw image (*Mandrill*). Bottom left: blur map based on the visual field of a glaucoma patient [102]; white = maximum ($\sigma = 10$); black = zero. Right: space-variant Gaussian-blurred image according to the blur map, using the proposed algorithm with $N=8$.

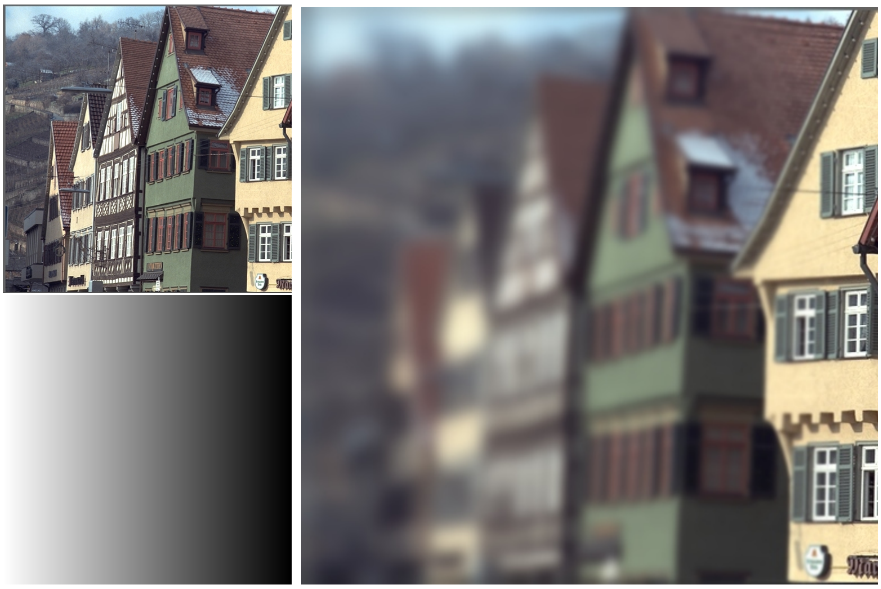


Figure 3.18: Top left: raw image (leftmost 512×512 portion of 512×768 image *Kodim08*). Bottom left: blur map, with blur increasing steadily from right to left; white = maximum ($\sigma = 10$); black = zero. Right: space-variant Gaussian-blurred image according to the blur map, using the proposed algorithm with $N=8$.

were used, as generated for a range $[m, M] = [\frac{1}{3}, 10]$ of σ values.

Computational cost

Applying Eq. (3.27) for a single location $\mathbf{x}$ is an order $\mathcal{O}(N)$ operation as it costs only N look-up operations (one for each $\hat{c}_{n,N}$), N multiplications and $N-1$ additions. The dominant cost in the algorithm is the fast DCTs in computation of the filtered images $\psi_1, ..., \psi_{N-1}$, each of which will be of order $\mathcal{O}(HW \log(HW))$, which only need to be done once for each image $\mathbf{I}$. Therefore, the overall cost of these convolutions is of order $\mathcal{O}(NHW \log(HW))$, which can be regarded as $\mathcal{O}(HW \log(HW))$ since N is fixed. This compares with $\mathcal{O}(H^2W^2)$ for the approach of applying an independent filter for every pixel, as is necessary with the reference method in the extreme case of a different blur level for every pixel. As an additional comparison, the fastest space-variant blurring approach is the integral image approach, whose cost is of order $\mathcal{O}(HW)$ for an W-by-H image. The blended Gaussian Pyramid approach is also $\mathcal{O}(HW)$, assuming a fixed maximum blur level (and hence a fixed number of hierarchy levels) as H and W increase. The blended Gaussian Pyramid cost therefore increases at roughly the same relative rate as the integral image approach as H and W become large.

3.3.2 Evaluation and Comparisons

Evaluation of the proposed approach

The proposed approach approximates space-variant Gaussian blurring to arbitrarily high accuracy, permitting any number N of basis functions to be employed. This allows a trade-off between computational cost and blurring accuracy, which is evaluated in this section in terms of PSNR. To this end, I have applied the algorithm with a varying number of basis functions, from one to fifteen. In order to provide an implementation-dependent measure of computational cost, a count of the total number of arithmetic operations (floating point additions, subtractions and multiplications) was employed. Average computational times per pixel were also recorded when running the (non-optimised) Java code, on a 3.19 GHz Pentium D machine running Microsoft Windows XP and the Java Runtime Environment 1.6.0. All computations in the evaluations were performed using 64-bit floating point arithmetic, to ensure a high accuracy ceiling for the experiments.

For simplicity, the test images [135–137] were selected to be a power of two in width and height (512×512): (1) *5.2.08*; (2) top-left 512×512 portion of *Barbera*; (3) *F-16*; (4)

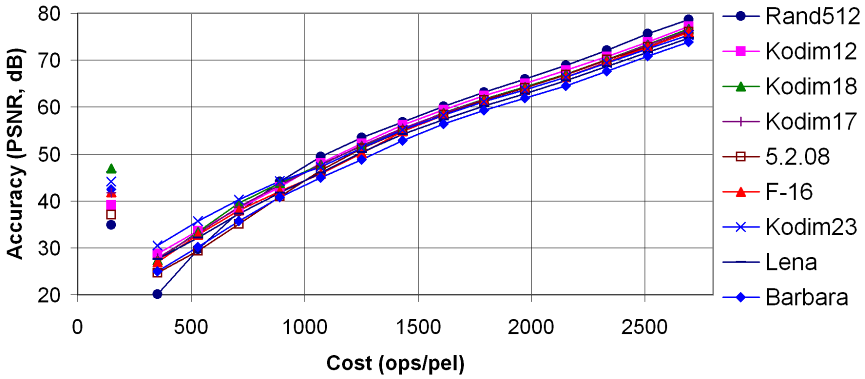


Figure 3.19: Cost-versus-accuracy plot (PSNR) for the proposed algorithm using varying numbers (2 to 15) of basis functions, when applied to the test images according to the corresponding foveal blur map shown in Figs 3.20 and 3.21. The isolated cluster of points to the left give the distribution of cost and accuracy for the blended Gaussian Pyramid approach for comparison purposes. Cost is measured in terms of the count of arithmetic operations per pixel. Accuracy is measured as PSNR relative to the perfectly Gaussian blurred image.

leftmost 512 × 512 portion of *Kodim12*; (5) topmost 512 × 512 portion of *Kodim17*; (6) topmost 512 × 512 portion of *Kodim18*; (7) leftmost 512 × 512 portion of *Kodim23*; (8) *Lena*; (9) a synthetic white noise image, *Rand512*, with each RGB sample randomly taken as either (0,0,0) or (255,255,255) with equal probability. For simplicity, all images were treated as full-colour RGB images, including the greyscale images *5.2.08* and *Rand512*, which were treated as colour images during experiments, for consistency with the other images. Therefore each measured cost of the implementation was precisely three times what it would have been for a greyscale image of the same size.

For all test images, a simplified foveal blur map was employed, defined for all pixel locations $(x_1, x_2) \in D$ as

$$b(x_1, x_2) = 2M\sqrt{\frac{(x_1 - c_1)^2 + (x_2 - c_2)^2}{H^2 + W^2}},$$

where $(c_1, c_2) \in D$ is the centre of the W-by-H image, M is the maximum blur level, and $\lfloor$ and $\rfloor$ denote integer downward rounding. Note that this map has zero blurring at the image centre and max blur level of 10 at each image corner. This blur map is depicted in Figs 3.20 and 3.21.

Each blurred image under test was computed using the proposed algorithm, according to the approximation in Eq. (3.27), for the given number, N, of basis functions. The basis functions as described in Section 3.3.1 were generated with a maximum blur level $M = 10$

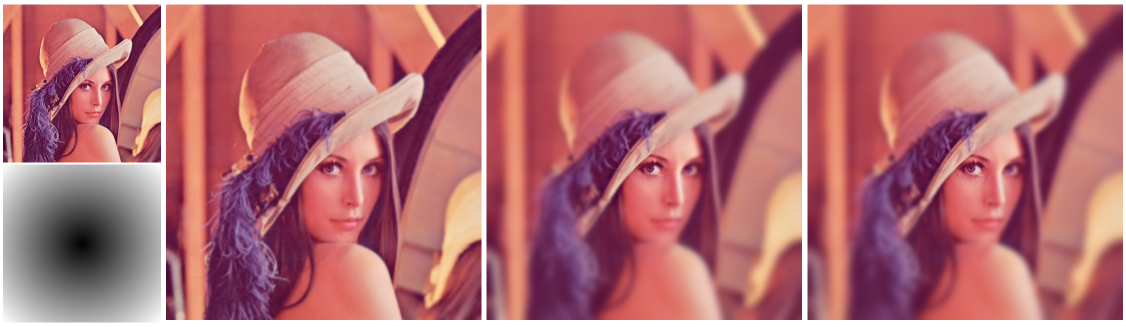


Figure 3.20: Examples of foveal blurring of *Lena*. Top left: raw image. Bottom left: blur map (black: $\sigma = 0$; white: $\sigma = 10$). The remaining images give the output of the proposed approach as it converges towards perfect Gaussian blurring (left to right: $N = 2, 5, 8$).

and a minimum $m = \frac{1}{3}$, and they were generated on a reduced-size domain of size 81×81, then padded with zeros to the full 512×512 size.

The reference method for perfect Gaussian blurring worked by computing each target image using Eq. (3.17), after separately computing every uniformly Gaussian-blurred image $\mathbf{U}_\sigma$, for 101 discrete σ values, $\sigma \in \{\frac{0}{10}, \frac{1}{10}, \frac{2}{10}, ..., \frac{99}{10}, \frac{100}{10}\}$. Each computation was performed using the same fast convolution technique as for the proposed approach except for the trivial case of $\sigma = 0$, which was dealt with by simply assigning $\mathbf{U}_0 = \mathbf{I}$, the input image. To ensure consistency across techniques, the same restriction to this discrete set of σ values was applied to the proposed approach.

The PSNR accuracy figures of the proposed approach using an increasing number of basis functions are shown in Table 3.1. The convergence of the proposed approach to perfect Gaussian blurring can be seen as the number of basis functions increases. At $N = 15$, for all test images, the PSNR exceeds 70 dB.

The cost figures in terms of floating point operation counts and average computational times per pixel are shown in Table 3.2. Fig. 3.19 visualizes the relationship between cost and PSNR. Each floating point count given in the graph and table is k/HW, where k is the total number of arithmetic operations used by the proposed algorithm to blur the given image, and HW is the number of pixels in the image. The PSNR improves at a steady rate on the logarithmic scale. The average improvement is 3.8 dB per basis function for the mean PSNR across images. The increase in cost is a fixed 180.1 ops/pel, and the average cost increase for each decibel of improvement to the mean accuracy is 50.4 ops/pel/dB.

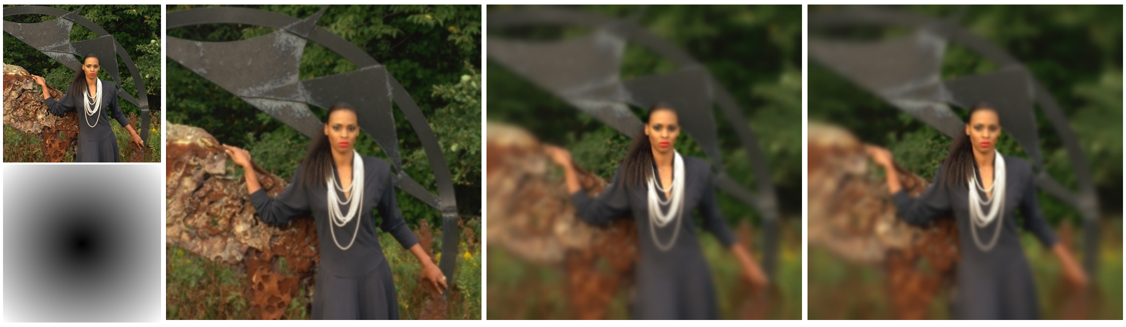


Figure 3.21: Examples of foveal blurring of *Kodim18*. Top left: raw image. Bottom left: blur map (as for Fig. 3.20). The remaining images give the output of the proposed approach as it converges towards perfect Gaussian blurring (left to right: $N = 2, 5, 8$).

Figs 3.20 and 3.21 show the output of the proposed technique for *Lena* and *Kodim18*, respectively, when using a differing number of basis functions ($N = 2, 5, 8$). In the $N = 2$ images, the image is sharp inside a narrow region around the foveation point (at the image centre), whereas outside this region, the level of blur is roughly uniform. This reflects the fact that only two basis functions are employed here, one of which is the delta function (no blurring) and the other of which cannot provide blurring greater than roughly $\sigma = 2$ (see the second basis function cross-section in Fig. 3.14). The remaining images show the result converging to the true space-variant Gaussian blurring as the number of basis functions employed increases, with the visible peripheral blurring increasing.

Comparison with other approaches

This section compares the results of the proposed approach with those of a blended Gaussian Pyramid technique [102] and the *integral image* technique [96, 97], in the same tests as described in the previous subsection. The *integral image* approach employed a square window width computed as 3.3σ then rounded to the nearest odd number (the ratio of 3.3 minimises the mean squared difference between a square window and a 2-D Gaussian window).

The details of the blended Gaussian Pyramid approach are as follows. The main part of the blended Gaussian Pyramid approach is the Gaussian Pyramid technique itself [103], which works as follows. Firstly, the RGB image is subsampled by factor of two vertically and horizontally, each time preceded by a 5-tap filter (i.e., weighted average)

Table 3.1: Blurring accuracy results (PSNR, decibels)

Image	**Integral Image**	**Gaussian Pyramid**	**Proposed approach (using 2 to 15 basis functions)**													
			2	3	4	5	6	7	8	9	10	11	12	13	14	15
Random512x512	32.3	34.9	20.1	29.7	37.7	44.1	49.4	53.5	56.8	60.1	63.2	66.0	68.8	72.1	75.7	78.7
Lena	41.2	46.1	27.6	32.2	37.2	41.7	46.0	50.4	54.2	57.4	60.2	62.8	65.7	68.7	71.7	74.7
5.2.08	37.2	37.1	24.7	29.3	35.1	41.0	46.5	51.2	54.9	58.3	61.5	64.1	66.9	70.2	72.9	76.0
Barbara	38.1	42.5	25.0	30.2	35.7	40.8	44.9	48.8	52.8	56.3	59.3	61.9	64.5	67.7	70.9	73.9
F-16	40.4	41.8	27.0	32.8	38.0	42.0	45.8	50.1	54.8	58.2	61.2	63.7	66.4	69.6	72.7	75.9
Kodim12	41.3	39.1	28.7	33.6	38.4	43.0	48.1	52.3	56.2	59.5	62.5	65.1	67.8	70.8	73.9	77.2
Kodim17	41.9	41.8	27.9	33.1	38.7	43.4	47.6	51.8	55.4	58.7	61.6	64.1	66.9	70.0	73.2	76.3
Kodim18	41.1	46.9	27.4	33.5	39.5	43.8	47.8	51.7	55.3	58.7	61.7	64.3	67.0	70.2	73.3	76.7
Kodim23	43.6	44.1	30.5	35.7	40.3	44.2	47.2	51.3	55.1	58.3	61.1	63.7	66.3	69.4	72.4	75.3
Mean	39.7	41.6	26.5	32.2	37.9	42.7	47.0	51.2	55.1	58.4	61.4	64.0	66.7	69.8	73.0	76.1

Table 3.2: Computational cost results

	Ref method	**Integr Image**	**Gauss Pyram**	**Proposed approach (using 2 to 15 basis functions)**													
				2	3	4	5	6	7	8	9	10	11	12	13	14	15
Ops per pel	17576.3	22.0	148.9	351.2	531.2	711.3	891.4	1071.5	1251.6	1431.7	1611.7	1791.8	1971.9	2152.0	2332.1	2512.1	2692.2
Time (μs/pel)	609.3	0.45	0.22	11.7	17.4	23.1	28.7	34.6	40.3	46.0	51.8	57.3	63.0	68.8	74.2	80.3	86.1

of $(\frac{1}{20}, \frac{1}{4}, \frac{2}{5}, \frac{1}{4}, \frac{1}{20})$ in each direction. This gives a multi-level pyramid of low-resolution representations of the original, each with 1/4 as many pixels as the previous level. Then, each level is upsampled back up to the original size, by repeated upsampling by a factor of two in each direction. Each upsampling is done as if by zero padding followed by averaging, in each direction, by $(0, \frac{1}{2}, 0, \frac{1}{2}, 0)$ at inter-pixel (odd) locations, and $(\frac{1}{10}, 0, \frac{4}{5}, 0, \frac{1}{10})$ at even locations. For consistency with the other approaches employed, the image is assumed to extend symmetrically beyond its boundaries. The result is a sequence of increasingly blurred images, the first of which is unblurred and each successive one is twice as blurred as the previous. To use the Gaussian Pyramid approach for smoothly space-variant blurring, inter-level blending is necessary. For this, the scheme of Perry & Geisler [102] is used, which works by taking, for each pixel, a linear combination between the two corresponding pixels of the images blurred to levels above and below the desired blur level for the given pixel.

This combination of pixels $\mathbf{p}_{i_r}, \mathbf{p}_{i_r-1} \in \mathbb{R}^3$ of the blurred images at levels i_r and $i_r - 1$ of the pyramid is computed as $B(r)\mathbf{p}_{i_r-1} + (1 - B(r))\mathbf{p}_{i_r}$, where $B(r) \in \mathbb{R}$ is the blending factor, computed as

$$B(r) = (0.5 - T_{i_r}(r))/(T_{i_r-1}(r) - T_{i_r}(r)),$$

T_{i_r} and T_{i_r-1} are the transfer functions (i.e., frequency responses) of the blurring at levels i_r and $i_r - 1$, and r is the half-amplitude frequency of the desired Gaussian curve, which was computed as $r = \sqrt{\log_e 4}/2\pi b(\mathbf{x})$ at a given location $\mathbf{x}$. Each level i_r is chosen as the value for which level $i_r - 1$ gives too little blurring and i_r gives too much. That is, for each r,

$$i_r = 1 + \max\{j : j \in \mathbb{Z} \text{ and } T_j(r) \geq 0.5\}.$$

The transfer functions were computed by applying the Gaussian Pyramid method to a simple impulse function and computing the magnitude of its Fourier Transform, restricted to a straight line horizontally through the zero frequency point.

Table 3.1 compares the accuracy of the proposed approach with that of the blended Gaussian Pyramid approach. The proposed approach outperforms blended Gaussian Pyramid in terms of PSNR at $N = 4$, for *Rand512*, at $N = 5$ for the mean across images and at $N = 7$ for the worst-case image (*Lena*). At $N = 8$, the PSNR improvement is typically 10 to 15 dB, with a 22.0 dB improvement in the case of *Rand512*. I choose therefore $N = 8$ as the number of basis functions to employ for high-quality Gaussian blurring. It should also be noted that the blended Gaussian Pyramid approach, while having on average only 1.9 dB better approximation of true Gaussian blurring than the *integral image* approach, has a cost more than 7 times greater in terms of the primitive operation count.

Discussion

The main reason for the limited accuracy of the blended Gaussian Pyramid approach is its slower-decaying tail in the frequency and spatial domains, when compared to perfect Gaussian blurring and the proposed technique. This is due to the fact that each effective approximated Gaussian PSF is a weighted sum of two Gaussian PSFs, one twice the width of the other. This slower decay can be expected to act as a disadvantage in foveated coding, given that the aim is to remove high frequency components. An example of a cross section

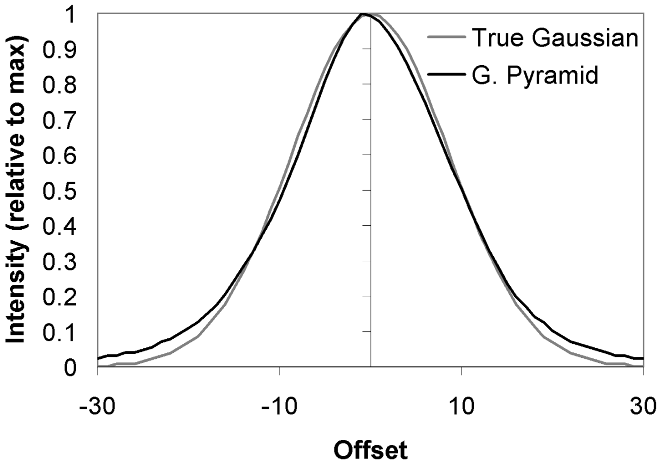


Figure 3.22: An example of an impulse response of the blended Gaussian Pyramid approach and true Gaussian blurring, when $\sigma = 8$. Note the wider tails of blended Gaussian Pyramid. These were generated by applying the blurring to a synthetic image consisting of a narrow vertical bar, which had a deliberate offset from zero to demonstrate the fact that the effective impulse response of blended Gaussian Pyramid will not generally be perfectly central.

of the impulse response of the blended Gaussian Pyramid approach, when applied with a fixed blur level, is given in Fig. 3.22. The plot shows how the impulse response deviates notably from that of true Gaussian blurring in the outer tails of the curve.

Table 3.2 compares the costs of the proposed method, the reference method (i.e., independent filter for each blur level) and the blended Gaussian Pyramid method. In all cases, the counts of arithmetic operations were the same for all images. For $N = 5$, at 891.4 ops/pel, the proposed approach costs 6 times as much as Gaussian Pyramid, while typically providing only marginally better blurring accuracy. However, with $N = 8$, for which the proposed approach gave on average 13.5 dB better accuracy, the cost is 1431.7 ops/pel, which is less than 9% of the cost of the reference method and less than 10 times the cost of the blended Gaussian Pyramid approach.

Fig. 3.23 provides a σ-dependent comparison between the proposed technique, blended Gaussian Pyramid and the *integral image* technique [96,97] in the context of space-variant preblurring for coding. Each curve gives the relative increase in JPEG bitrate of the given technique when compared with the equivalent using perfect Gaussian blurring. Each bitrate was taken by applying a uniform blur map of the given σ value, to a 256-by-256 white noise image and encoding to a fixed quality level, measured using PSNR (fixed at 35dB in all cases). The graph demonstrates the advantage of a direct substitution of

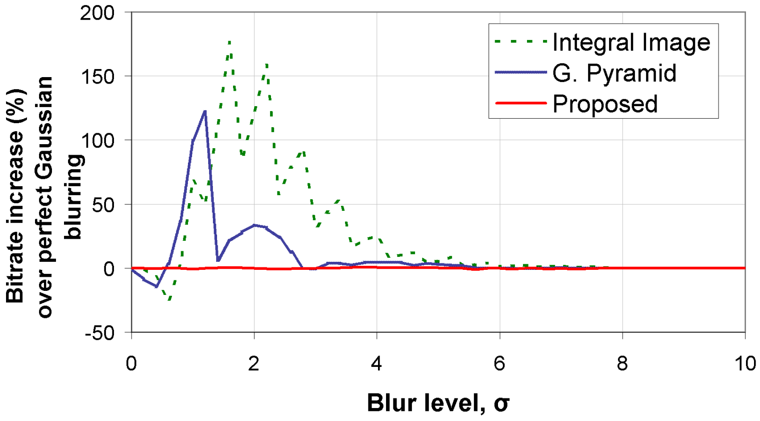


Figure 3.23: A JPEG bitrate comparison between different blurring techniques as a function of blur level, σ. Each line gives the relative increase in JPEG bitrate of the given technique when compared with the equivalent using perfect Gaussian blurring.

blended Gaussian Pyramid with the proposed approach in a foveated coding scenario, showing how the bitrate advantages in a given region of an image will be dependent on the blur level in that region. The poorer general performance of *integral image* blurring demonstrates the advantage of instead using a blurring technique with a smooth point spread function. The poorer general performance of *integral image* blurring demonstrates the advantage of instead using a blurring technique with a smooth point spread function. The drop into negative percentages for *integral image* blurring below $\sigma = 1$ is a consequence of the round-to-nearest interpretation of blur levels that was applied, which causes this approach to be effectively given a higher blur level than the others, resulting in a lower bitrate. The average bitrate improvement with the proposed approach compared with blended Gaussian Pyramid, is 5.4%.

3.4 Conclusion

This chapter has provided two pieces that were missing in the realm of foveated coding. Firstly, an algorithm has been presented which allows blur maps to be computed that are optimal for lossy coding purposes in the sense of discarding visually least-noticeable frequencies first. This algorithm assumes a multi-viewer or saliency-based spatio-frequency eye sensitivity model, which is based on the Geisler & Perry contrast threshold formula but which differs from prior approaches in that it combines individual viewer sensitivities additively rather than by always taking the sensitivity of the viewer with the nearest fixation point. The key part of the algorithm is the efficient computation of collective sensitivity values, and the key step in this is the choice of a best-approximating basis for the set of possible Geisler & Perry frequency-sensitivity curves. By employing this basis, what would have been an $\mathcal{O}(n)$-per-pixel operation (where n is the number of pixels in the image) is enabled to be performed in $\mathcal{O}(\log n)$ operations per pixel, thus allowing the use of an additive sensitivity function that would otherwise have been infeasible. An additional benefit of the proposed approach is that it handles the infinite-viewer (saliency-based) scenario with equal cost to the multi-viewer scenario. A comparison of the output of this algorithm with an alternative multi-foveation technique (Sheikh *et al.*) has been shown and discussed but not evaluated subjectively. The proposed approach appears almost to neglect outlying fixation points, with the main body of preserved frequencies surrounding the dominant cluster of fixations, whereas the Sheikh *et al.* technique effectively boosts the significance of the outliers, providing a more widely-spread region of higher resolution. Also, the proposed approach has a smoother variation in frequency, thus reducing the chances that local variation in blur level may itself be observed as an artifact.

Secondly, an algorithm has been proposed which makes high-accuracy space-variant Gaussian blurring a practicality. True Gaussian blurring has a more rapid frequency fall-off than blurring performed by the nearest practical alternative – that of blended Gaussian Pyramid – and this would allow the higher frequencies to be more effectively discarded by an encoder. The proposed algorithm comprises a specialized filter bank whose filters are implemented using a fast DCT approach. These filters are an optimal basis from the perspective of spanning a given range of Gaussian point spread functions, and are computed using principal component analysis. As the number of basis functions employed is increased, the resultant blurring converges rapidly to true space-variant Gaussian blurring. Arbitrarily-perfect Gaussian blurring can be obtained depending on the number of basis functions used. Experimental results show that the proposed algorithm provides typically 10 to 15 dB better approximation of perfect Gaussian blurring than the blended Gaussian Pyramid blurring approach when using a bank of just eight filters. The computational cost of the algorithm is the same regardless of the number of desired blurring levels or the complexity of the blur map.

As discussed in Chapter 2, foveated coding is a widely-adopted approach to space-variant coding, taking the lossy coding conventional approach of aiming for perceptually minimal or invisible distortion. The next chapter introduces an alternative approach which more directly addresses the ultimate lossy coding aim of maximally acceptable distortion.

Chapter 4

Depth-Blurred Coding

This chapter introduces the concept of depth-based blurring to achieve an aesthetically acceptable distortion when reducing the bitrate in image coding. The proposed depth-based blurring is a prefiltering that reduces high frequency components by mimicking the limited depth of field effect that occurs in cameras. Two selective blurring algorithms are presented that simulate occlusion effects as occur in natural blurring. These algorithms can handle any number of blurring and occlusion levels.

4.1 Introduction

For lossy picture coding, as mentioned in section 2.4.1, the approach of selective preblurring before an ordinary encoder has the advantages, over internal encoder space-variance, of simplicity and the fact that the distortion it creates is more likely to occur naturally, and therefore may be considered more visually acceptable. Taking this notion a step further, consider the practice often employed in photography, whereby shots are taken with low depth of field; that is, one part of the scene is sharply focussed, while the rest of the scene becomes increasingly blurred as it fades into the distance or gets nearer to the camera. There is an established notion that this effect makes a positive contribution to the aesthetic quality of the result. A convenient side effect of this from the perspective of picture coding is that with most coding formats, performing what amounts to a selective preblurring of an image will reduce the average bitrate (for example, as mentioned in section 2.4.1, with DCT-based coding formats, the prefiltering increases the abundance of zero or near-zero DCT coefficients). Therefore, if a reduction of depth of field is synthetically applied to an image or video sequence, it will not only allow bitrate reduction due to the selective

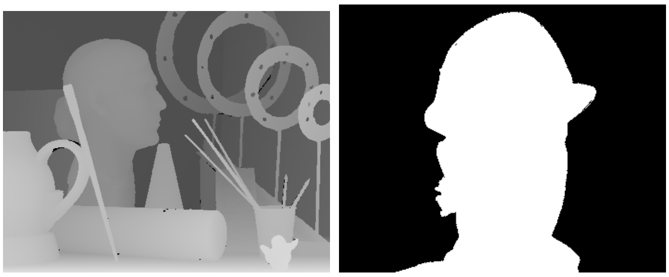


Figure 4.1: Depth/disparity map examples. Left: *Art* disparity map, from Middlebury dataset [138, 139]; right: two-level manual segmentation of *Foreman*.

blurring, but possibly even an increase in the overall perceived quality. This approach is referred to herein as *depth-blurred coding*.

As foveated coding requires knowledge of points or regions of interest, whether from eye tracking (gaze-contingent) [10] or estimated (e.g., saliency detection [29]), this might be problematic, as the selection of these priority regions remains an open problem [29]. If the estimated points of interest are wrong, the distortion becomes noticeable. In contrast, the depth blurred coding approach is not so susceptible to this problem. An important idea here is that if viewers look away from the predicted point of interest (as when eye tracking is not employed), they might judge the degradation more favourably than with foveation filtering. Moreover, in contrast with the difficulty of estimating saliency information (for foveation), the proposed approach uses depth information, which is becoming increasingly available, for example from time-of-flight cameras [140], as disparity (reciprocal of depth) information from stereo camera pairs in conjunction with dense stereo correspondence techniques [141] or from scene structure estimation techniques [142, 143]. Furthermore, in the simplest case, depth blurring can take a region-of-interest approach, by applying a two-level depth map (Fig. 4.1) generated for example by background–foreground segmentation [56] or by an object detector such as face detector [97].

This chapter proposes two similar algorithms for applying realistic depth of field effects to images. The proposed algorithms simulate occlusion effects as occur at the boundaries of objects and can handle a depth map which is continuous (up to blur level quantization granularity), with a cost of order $\mathcal{O}((\log N)^2 N)$ for an N-pixel image. The argument made herein is that the relative ease of obtaining a depth map, when compared with the difficulty of predicting human fixation to a sufficient level of certainty for foveation, makes depth-blurring a preferable space-variant coding approach for the non-gaze-contingent

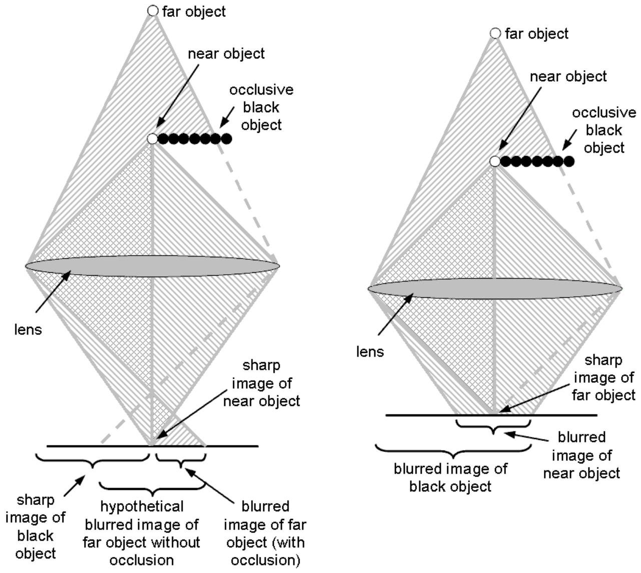

far object
near object
occlusive black object
lens
sharp image of near object
sharp image of black object
hypothetical blurred image of far object without occlusion
blurred image of far object (with occlusion)
far object
near object
occlusive black object
lens
sharp image of far object
blurred image of near object
blurred image of black object

Figure 4.2: Examples of occlusive effects in depth blurring. Left: near object in focus; right: far object in focus. In the left diagram, the light from the far (blurred) object is spread over a region which is occluded sharply at the image of the near object. In the right diagram, the light from the near (blurred) object is unaffected by anything beyond it.

scenario.

The remainder of this chapter is organised as follows: Section 4.2 discusses the challenges addressed by the two algorithms. Section 4.3 introduces the common aspects of the two algorithms. Sections 4.4 and 4.5 give the specific details of each algorithm respectively. Section 4.7 concludes the chapter.

4.2 Challenges in Synthetic Depth Blurring

This subsection discusses the two main challenges in depth blurring, namely the proper treatment of occlusion effects and the computational complexity.

As mentioned in section 2.5.1, techniques for synthesizing depth of field can be classed as either multipass approaches or postfiltering. In multipass approaches, high-accuracy techniques such as ray tracing are repeated from slightly different directions and averaged [105]. Although high quality, multipass approaches generally involve heavy computational cost. In postfiltering, the rendering output itself is subjected retrospectively to synthetic depth blurring [104]. Postfiltering approaches can in turn be grouped into *gather* or *scatter* methods. Techniques which employ the *gather* method approximate depth blurring by

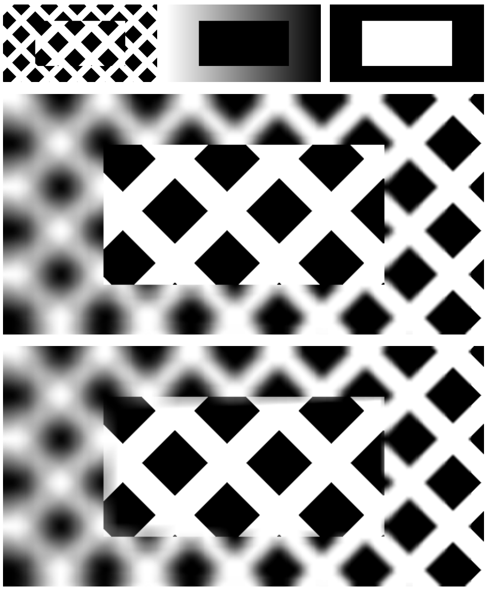


Figure 4.3: Examples of occlusion effects using Algorithm 1 on a synthetic image. Top, left to right: raw image, blur map (black: unblurred) and occlusion map (white: more occlusive); the blur map and occlusion map may come directly from a depth map. Middle: blurred with occlusion. Bottom: the same blurring except with occlusion effects switched off. The spread of the background blur over the foreground boundary can be seen in the non-occlusive case, but not in the occlusive case.

taking the local average of pixel values around the desired location, which inherently leads to *intensity leaks* [104] as the intensity from sharp source pixels is spread over surrounding background that they should not influence. Approaches that employ the *scatter* method spreadthe intensity of each source pixel over an area. However, due to speed, scatter methods are not regarded as the choice for real-time depth blurring [110].

Although foveation was first extended into the 3rd (depth) dimension by Van Der Linde [12], and continued by Çöltekin [51], as mentioned in sections 2.1 and 2.5, these rely on eye tracking and do not test the plausibility of removing the eccentricity-dependent foveation aspect altogether along with any assumptions about where the viewer will look. It employs simple Gaussian pyramid blurring [102] for resolution-reduction purposes, aiming for minimally perceivable distortion rather than photorealistic blurring that is aesthetically acceptable on close inspection. These techniques rely not only on knowledge of 3-D structure of the scene but also on precise real-time knowledge of the fixation point and focal depth of the eyes. In contrast, the depth-blurred coding approach proposed herein makes no precise assumptions about viewer fixation points or focal distances. Addition-

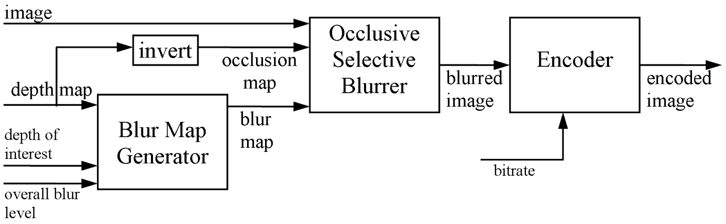


Figure 4.4: Block diagram of the overall approach that receives as input a depth map (or equivalent information) and a chosen depth of interest. The Occlusive Selective Blurring block is the core algorithm, for which two variants are described in sections 4.3 to 4.5.

ally, the selective blurring techniques used by Van Der Linde and Çöltekin take no account of the occlusion effects present in natural depth blurring.

One key aspect of the style of depth blurring that a human viewer is accustomed to seeing in photographs is the effect that occurs around the boundaries of objects that occlude further-away objects in the scene. This *occlusive* aspect of the blurring occurs, for example, when a sharply-focussed object is in front of a blurred distant object, in which case the blur of the distant object stops abruptly at the edge of the nearer object, with no part of the blur overlapping any part of the nearer object. However, when a blurred object is in front of some sharply-focussed background, the edges of the blur of the nearer object spread over the background, because the blur goes in all directions, some of which will overlap the background. See Fig. 4.2 for an illustration of these points. The proposed approach caters for this occlusive effect, taking occlusion information from an occlusion map in addition to the blur map. An ordinary, unocclusive selective blurring technique would cause every blurred pixel to be spread over its neighbours regardless of whether they are considered nearer to or further from the camera. Examples of occlusive and unocclusive blurring are shown in Fig. 4.3, where the spread of the background blur over the foreground boundary can be seen in the non-occlusive case, but not in the occlusive case.

4.3 The Blurring Algorithms

This section introduces the top level of the two algorithms, referred to as Algorithm 1 and Algorithm 2, for computing occlusive depth blurring. Both algorithms aim to produce the same output, but compute this output differently. The proposed depth blurring algorithms assume a context as illustrated in Fig. 4.4. The bitrate and overall blur level control the overall quality of the encoding. The blur map is computed from the depth map so that the chosen depth has zero blur level.

Given a depth map (generated for example by a time-of-flight or stereo camera, or by a segmentation algorithm), a desired bitrate and overall blur level (e.g. from the rate-control mechanism of an encoder) and a depth of interest chosen to be in sharp focus (such as by taking the nearest-to-camera depth or the depth at a point of interest selected using saliency detection), the proposed depth blurring algorithm takes as inputs a colour image $\mathbf{C}$, a continuously-varying blur map B and an *occlusion map* Ω, all defined over a $W \times H$ image domain $D = \{(x, y) : x \in \{1, ..., W\}, y \in \{1, ..., H\}\}$. Note that the advantages of the proposed approach, as with any other prefiltering approach, apply to the scenario where there is scope for selective resolution reduction of the input image; therefore any existing blurring will lessen the need for further blurring for bitrate reduction purposes. Regarding the choice depth of interest, note also that if the level of blur already present is sufficient for sourcing the depth map from a depth-from-defocus technique [144], the in-focus depth should be chosen as the existing focal plane, so that the synthetic blurring will enhance the photographer's original choice of depth of interest.

The occlusion map gives, in arbitrary units, the occlusion level of each point, thereby providing a ranking of which pixels should or should not be overlapped by the blur regions of which other pixels. Therefore it may be taken directly as the negative of the depth map, so that more occlusive (nearer-to-camera) points have a higher occlusion level. The blur map is taken from the depth map such that the chosen depth will be in sharp focus (zero blur) and the other depths will have gradually increasing blur away from this depth; e.g., each blur level $b(\mathbf{x})$ (at location $\mathbf{x} \in D$) may be taken as

$$b(\mathbf{x}) = k \left| \frac{1}{d(\mathbf{x})} - \frac{1}{d_0} \right| \tag{4.1}$$

for depth $d(\mathbf{x})$, sharp-focus depth d_0 and constant k chosen to obtain a desired overall blur level. Equation (4.1) is explained in fig 4.5 for the case when $d(\mathbf{x}) > d_0$.

The proposed algorithm has an $\mathcal{O}((\log N)^2 N)$ cost and it can handle a depth map which is continuous, up to blur level quantization granularity. Both algorithms spread the intensities of each pixel uniformly over a square area of variable size, subject to sharp occlusions by any nearer pixel. Then an adjustment factor is applied, to compensate for

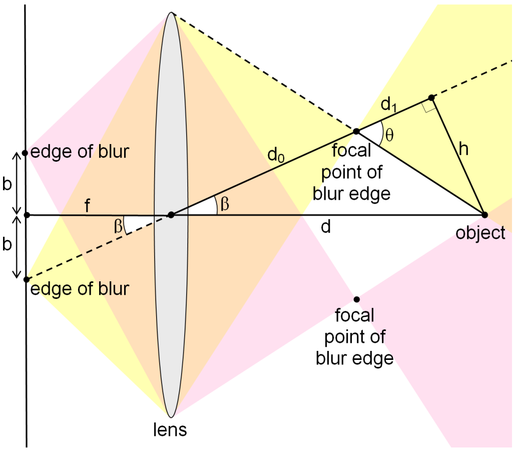


Figure 4.5: Diagram of the geometry of blurring. The extremities of the blur of the point object (i.e. the edges of the *circle of confusion*) are shown, and the areas from which light travels into these edge points are shown in yellow and pink respectively. The relationships $b = f\tan\beta$, $h = d\sin\beta$, $h = d_1\tan\theta$, $\tan\beta \approx \sin\beta$ and $d \approx d_0 + d_1$ collectively imply that $d\tan\beta \approx (d - d_0)\tan\theta$ and therefore that $b \approx (1/d_0 - 1/d)fd_0\tan\theta$. For a given fixed focal distance d_0, θ can be regarded as fixed for a sufficiently small range of β; therefore, if constant k is defined as $k = fd_0\tan\theta$, then $b \approx (1/d_0 - 1/d)k$. The error in this approximation rapidly approaches zero as the lens height becomes small relative to d_0 and d.

the fact that this pixel spreading will unnaturally darken or brighten the image in regions where the blur level is not constant. For n=1,2,3, each occlusively selectively blurred colour plane C'_n is computed (using adjustment factor $1/U'(\mathbf{x})$) as follows:

$$C'_n(\mathbf{x}) \quad = \quad P_n(\mathbf{x})/U'(\mathbf{x}), \tag{4.2}$$

where P_n is the occlusively pixel-spreaded version of original colour plane C_n under the given blur map and occlusion map, and U' is the equivalent when applied to a pure white image, U. The terms *pixel-spreading* and *blurring* will be used herein to describe the creation of the unadjusted result and the final, adjusted result, respectively. The top level (the “blurring” stage) of both proposed algorithms is illustrated in Fig. 4.6.

A key aspect of the proposed algorithms is the concept of a *corner* of the spread of a given pixel under the blurring. Given a colour plane G, and a blur map B, the intensity $G(\mathbf{x})$ at location $\mathbf{x}$ will, neglecting image boundary issues, be spread over a square area of width $2B(\mathbf{x}) + 1$, with centre $\mathbf{x}$.

Considering the pixel at $\mathbf{x} = (x, y)$, the image, $\mathcal{P}_{\mathbf{x}}$, of the spread of this sole pixel will

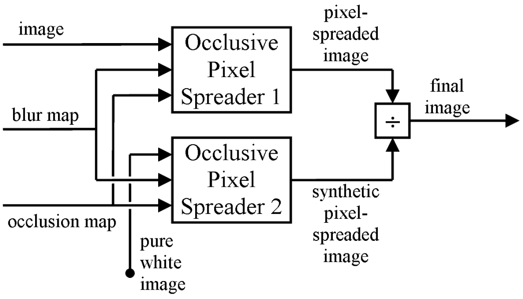


image
blur map
occlusion map
Occlusive Pixel Spreader 1
Occlusive Pixel Spreader 2
pixel-spreaded image
÷
final image
synthetic pixel-spreaded image
pure white image

```
$\mathbf{C}' = \textbf{\textit{blur}}(\mathbf{C}, B, \Omega)$
Inputs: 3-colour image $\mathbf{C}$, blur map $B$, occlusion map $\Omega$.
Create blank monochrome image $U$.
Create blank 3-colour images $\mathbf{P}$ and $\mathbf{C}'$, with colour planes $P_1, P_2, P_3$ and $C'_1, C'_2, C'_3$.
for each pixel location $\mathbf{x} \in D$ do
  Set $U(\mathbf{x}) = 1$.
end for
Compute $U' = \textbf{\textit{spread}}(U, B, \Omega)$.
for each colour plane, $C_n$ $(n = 1, 2, 3)$, of $\mathbf{C}$, do
  Compute $P_n = \textbf{\textit{spread}}(C_n, B, \Omega)$.
  for each pixel location $\mathbf{x} \in D$ do
    Set $C'_n(\mathbf{x}) = P_n(\mathbf{x})/U'(\mathbf{x})$.
  end for
end for
Output: occlusively selectively blurred-image $\mathbf{C}'$.
```

Figure 4.6: Left: Block diagram of the top level of the proposed occlusive selective blurring algorithm. For simplicity, the "pixel spreading" is shown as being applied to the whole colour image, whereas in reality it is applied separately to each of the three colour planes (R,G,B). Right: the step-by-step top-level instructions, also common to both algorithms. Here, D represents the set of location in the image, blur map and occlusion map. For Algorithm 1, *spread1* (as defined in Fig. 4.9) should be used in place of *spread.* For Algorithm 2, *spread2* (as defined in Fig. 4.14) should be used.

be given, for every image location $\mathbf{x}' = (x', y')$, by

$$\mathcal{P}_{\mathbf{x}}(x', y') = \begin{cases} v & \begin{array}{c} \text{if } |x' - x| \leq B(\mathbf{x}) \\ \text{and } |y' - y| \leq B(\mathbf{x}) \end{array} \\ & \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

where $v = \frac{G(\mathbf{x})}{(2B(\mathbf{x})+1)^2}$. $\mathcal{P}_{\mathbf{x}}$ can also be expressed as a cumulative sum, as follows:

$$\mathcal{P}_{\mathbf{x}}(x', y') = \sum_{\substack{x'' \leq x' \\ y'' \leq y'}} \mathcal{P}'_{\mathbf{x}}(x'', y''), \tag{4.4}$$

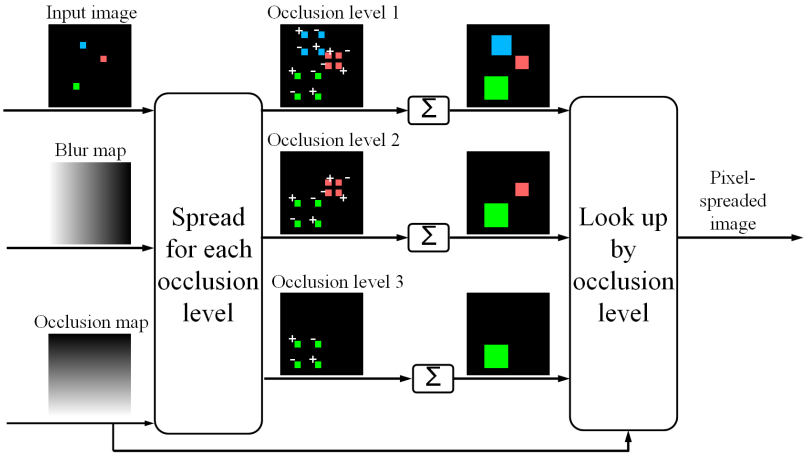


Figure 4.7: Block diagram illustrating the concept of corners in the pixel-spreading. For each occlusion level, there is conceptually a differential image (second column from left), whose cumulative sum gives a pixel-spreaded image (third column from left), and from these, the overall occlusively pixel-spreaded image can be formed by looking up each output pixel from the appropriate cumulative sum according to its individual occlusion level. However, in both algorithms, in practice, separate full-image cumulative summations (denoted above by "Σ") are not performed, as the occlusive sum look-up structure allows the selected output pixels to be computed on their own, without full cumulative sums.

for all $\mathbf{x}' = (x', y')$, where image $\mathcal{P}'_{\mathbf{x}}$ is defined as follows:

$$\mathcal{P}'_{\mathbf{x}}(\mathbf{x}'') = \begin{cases} v & \text{if } \mathbf{x}'' = \mathbf{c}^{11}(\mathbf{x}, B(\mathbf{x})) \\ -v & \text{if } \mathbf{x}'' = \mathbf{c}^{12}(\mathbf{x}, B(\mathbf{x})) \\ -v & \text{if } \mathbf{x}'' = \mathbf{c}^{21}(\mathbf{x}, B(\mathbf{x})) \\ v & \text{if } \mathbf{x}'' = \mathbf{c}^{22}(\mathbf{x}, B(\mathbf{x})) \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

for all possible $\mathbf{x}''$. Functions $\mathbf{c}^{11}$, $\mathbf{c}^{12}$, $\mathbf{c}^{21}$ and $\mathbf{c}^{22}$ can be thought of as giving the four corners of the spread of the pixel at $\mathbf{x}$. The horizontal components, c_1^{11}, c_1^{12}, c_1^{21} and c_1^{22}, and vertical components, c_2^{11}, c_2^{12}, c_2^{21} and c_2^{22}, of these functions are defined as

$$c_1^{11}(\mathbf{x}, b) = c_1^{12}(\mathbf{x}, b) = \begin{cases} x - b & \text{if } x > b \\ 1 & \text{otherwise} \end{cases} \tag{4.6}$$

$$c_2^{11}(\mathbf{x}, b) = c_2^{21}(\mathbf{x}, b) = \begin{cases} y - b & \text{if } y > b \\ 1 & \text{otherwise} \end{cases} \tag{4.7}$$

$$c_1^{22}(\mathbf{x}, b) = c_1^{21}(\mathbf{x}, b) = x + b + 1 \tag{4.8}$$

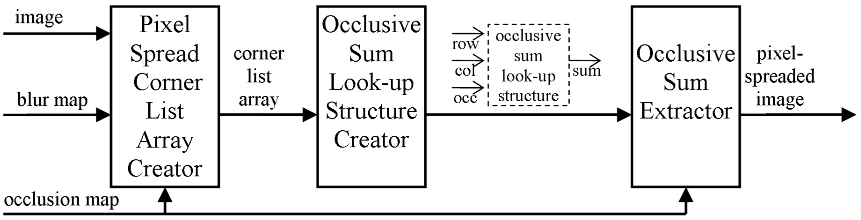


image
blur map
occlusion map
Pixel Spread Corner List Array Creator
corner list array
Occlusive Sum Look-up Structure Creator
row
col
occ
occlusive sum look-up structure
sum
Occlusive Sum Extractor
pixel-spreaded image

Figure 4.8: Block diagram of the Occlusive Pixel Spreader (see Eq. 4.10) for Algorithm 1. The blocks correspond to the three main blocks of the *spread1* operation. The leftmost block creates a two-dimensional array of the corners of the spread of each pixel (see Eqs (4.6)-(4.9) and Fig. 4.7). The workings of the middle and rightmost blocks are given by the *createstruct* and *extractsum* operations of Fig. 4.9.

$$c_2^{22}(\mathbf{x}, b) = c_2^{12}(\mathbf{x}, b) \quad = \quad y + b + 1 \tag{4.9}$$

for every possible location $\mathbf{x} = (x, y)$ and blur level b.

The overall pixel-spreaded colour plane, P, as produced by the *Occlusive Pixel Spreader* (see Fig. 4.6), may be defined as

$$\begin{aligned} P(\mathbf{x}) &= \sum_{\substack{\mathbf{x}' \in D: \\ \Omega(\mathbf{x}') \geq \Omega(\mathbf{x})}} \mathcal{P}_{\mathbf{x}'}(\mathbf{x}) & (4.10) \\ &= \sum_{\substack{x'' \leq x \\ y'' \leq y}} \sum_{\substack{\mathbf{x}' \in D: \\ \Omega(\mathbf{x}') \geq \Omega(\mathbf{x})}} \mathcal{P}'_{\mathbf{x}'}(\mathbf{x}''), & (4.11) \end{aligned}$$

for all $\mathbf{x} = (x, y) \in D$, where D is the set of image locations, as before and $\mathcal{P}'_{\mathbf{x}'}(\mathbf{x}'')$ is as defined in Eq. 4.5, but with $\mathbf{x}'$ substituted for $\mathbf{x}$. This selective cumulative sum is conceptually represented in Fig. 4.7, which illustrates the meaning of the corners of a spread and how they are used.

The $\mathcal{O}(N^2)$ cost that would be required by the naïve approach of independently computing the spreading at each occlusion level is reduced to the order of $\mathcal{O}((\log N)^2 N)$ by using either of the methods described in the next two sections, 4.4 and 4.5.

4.4 Implementation of Algorithm 1

The method used by Algorithm 1 for computing the occlusive pixel-spreading of the proposed depth-blurring algorithm is illustrated in Fig. 4.8.

Firstly, the corner list array is created as follows. For each pixel location $\mathbf{x}$ in the original image, the intensity $g = G(\mathbf{x})$ and blur level $b = B(\mathbf{x})$ are read. The spreaded

```
$P$ = spread1$(G, B, \Omega)$
Inputs: colour plane $G$, blur map $B$, occlusion map $\Omega$.
 1: Set $L$ = listcorners$(G, B, \Omega)$.
 2: Set $S$ = createstruct$(L)$.
 3: Set $P$ = extractsum$(S, \Omega)$.
Output: pixel-spreaded colour plane $P$.
```

```
$S$ = createstruct$(L)$
Inputs: 2-D array $L$ of corner lists.
 1: for each $m \in \{0, ..., \lfloor \log_2(H) \rfloor\}$ do
 2:   Set $h = 2^m$.
 3:   for each $Y \in \{1, ..., \lfloor H/h \rfloor\}$ do
 4:     Set $T$ = null.
 5:     for each $x \in \{1, ..., W\}$ do
 6:       for each $y \in \{Yh-(h-1), ..., Yh\}$ do
 7:         for each pair $(v, \omega) \in L(x, y)$ do
 8:           Set $T$ = treeadd$(T, v, \omega)$.
 9:         end for
10:       end for
11:       Set $S(m, Y, x) = T$.
12:     end for
13:   end for
14: end for
Output: summation structure $S$.
```

```
$L$ = listcorners$(G, B, \Omega)$
Inputs: colour plane $G$, blur map $B$, occlusion map $\Omega$.
 1: for each pixel location $\mathbf{x} \in D$ do
 2:   Set $g = G(\mathbf{x})$, $b = B(\mathbf{x})$, $\omega = \Omega(\mathbf{x})$ and $v = g/(2b+1)^2$.
 3:   for each $\mathbf{z} \in \{\mathbf{c}^{11}(\mathbf{x}, b), \mathbf{c}^{12}(\mathbf{x}, b), \mathbf{c}^{21}(\mathbf{x}, b), \mathbf{c}^{22}(\mathbf{x}, b)\}$ do
 4:     if $\mathbf{z} \in D$ then
 5:       Append pair $(v, \omega)$ to list $L(\mathbf{z})$.
 6:     end if
 7:   end for
 8: end for
Output: 2-D array $L$ of corner lists.
```

```
$L$ = extractsum$(S, \Omega)$
Inputs: summation structure $S$, occlusion map $\Omega$.
 1: for each pixel location $(x, y) \in D$ do
 2:   Set $s = 0$, $m = 0$ and $y' = y$.
 3:   repeat
 4:     if $y'$ is odd then
 5:       Set $s$ += treeget$(S(m, \lceil \frac{y'}{2} \rceil, x), \Omega(x, y))$.
 6:     end if
 7:     Set $y' = \lfloor \frac{y'}{2} \rfloor$ and $m$ += 1.
 8:   until $y' = 0$.
 9: end for
Output: pixel-spreaded colour plane $P$.
```

Figure 4.9: The step-by-step details of *spread1*, the approach used by Algorithm 1 for occlusive pixel-spreading of a single colour plane, and its three main parts (which correspond to the three blocks in the block diagram in Fig. 4.8). Here, $D = \{(x, y) : x \in \{1, ..., W\}, y \in \{1, ..., H\}\}$ is the set of location in the image, blur map and occlusion map (all width W and height H), $\lceil\ \rceil$ and $\lfloor\ \rfloor$ denote integer upward and downward rounding, and functions $\mathbf{c}^{11}$, $\mathbf{c}^{12}$, $\mathbf{c}^{21}$ and $\mathbf{c}^{22}$ yield corner points as defined in equations (4.6) to (4.9). The details of operations *treeadd* and *treeget* are given in Fig. 4.10. Each iteration of the outer loop of *extractsum* (i.e. each run of lines 2 to 8) corresponds to an "occlusive sum look-up structure" call for a single pixel (see the block diagram in Fig. 4.8).

intensity $v = \frac{g}{(2b+1)^2}$ is paired with occlusion level $\omega = \Omega(\mathbf{x})$. The pair (v, ω) is appended to four lists, associated with the four corner points $\mathbf{c}^{11}(\mathbf{x}, b)$, $\mathbf{c}^{12}(\mathbf{x}, b)$, $\mathbf{c}^{21}(\mathbf{x}, b)$ and $\mathbf{c}^{22}(\mathbf{x}, b)$ (see Eqs (4.6)-(4.9)).

The look-up structure is the key part of the algorithm, and is the part which reduces the complexity of the occlusive selective blurring from $\mathcal{O}(N^2)$ to $\mathcal{O}((\log N)^2 N)$. It takes

```
$T' = \textbf{\textit{treeadd}}(T, v, \omega)$
Inputs: tree root node $T$, colour component $v$, occlusion level $\omega$.
Set $T'$ = a new tree node.
if $T$ is null then
    Set $R_{T'} = \{\omega\}$ and $V_{T'} = v$.
else if $R_T = \{\omega\}$ then
    Set $R_{T'} = \{\omega\}$ and $V_{T'} = V_T + v$.
else
    Set $R_{T'} = [\min(\omega, \min(R_T)), \max(\omega, \max(R_T))]$ and $V_{T'} = V_T + v$.
    if $\omega \in$ lower half of $B(R_T)$ then
        Set $U_{T'} = U_T$ and $L_{T'} = \textbf{\textit{treeadd}}(L_T, v, \omega)$.
    else if $\omega \in$ upper half of $B(R_T)$ then
        Set $L_{T'} = L_T$ and $U_{T'} = \textbf{\textit{treeadd}}(U_T, v, \omega)$.
    else
        Set $T''$ = a new tree node, $R_{T''} = \{\omega\}$ and $V_{T''} = v$.
        if $\omega < \min(R_T)$ then
            Set $L_{T'} = T''$ and $U_{T'} = T$.
        else
            Set $U_{T'} = T''$ and $L_{T'} = T$.
        end if
    end if
end if
Output: new tree root node $T'$.
```

```
$s = \textbf{\textit{treeget}}(T, \omega)$
Inputs: tree root node $T$, occlusion level $\omega$.
if $\omega < \min(R_T)$ then
    Set $s = 0$.
else if $\omega \geq \max(R_T)$ then
    Set $s = V_T$.
else
    Set $s = \textbf{\textit{treeget}}(L_T, \omega) + \textbf{\textit{treeget}}(U_T, \omega)$.
end if
Output: summation result $s$.
```

Figure 4.10: The step-by-step details of operations *treeadd* and *treeget*, which are important parts of the occlusive pixel-spreading part of Algorithm 1 (see Fig. 4.9). Here, each tree node T, at the root of its own subtree, may be regarded as a pointer to a tuple, (V_T, R_T, L_T, U_T), where $V_T \in \mathbb{R}$ is the grey level (or colour component) total for the subtree, $R_T \subset \mathbb{R}$ is the smallest contiguous subset of $\mathbb{R}$ which spans all ω (occlusion) values covered by the subtree, and L_T and U_T are pointers to the lower and upper branch nodes which (optionally) sprout from node T. The set $B(R_T)$, a superset of R_T, is the smallest contiguous range of real numbers of the form $[2^b a,\ 2^b(a+1)-1]$ for integers a and b satisfying $R_T \subseteq B(R_T)$; therefore, the number of integers coincident with $B(R_T)$ will always be a power of two. Note that $\{\omega\}$ denotes a set containing only the value ω and note that square brackets $[,]$ are used to denote a contiguous range of real numbers with given range limits.

as inputs the location of a pixel (by row and column) and an occlusion level, and outputs the sum of all the entries with a higher occlusion level in the corner list array in the rectangle bounded by that pixel and the top-left pixel of the image. This is created firstly by partitioning the image domain into a hierarchy of groups of adjacent rows of pixel locations, with sets of 1 row at the bottom level of the hierarchy, then sets of 2 adjacent

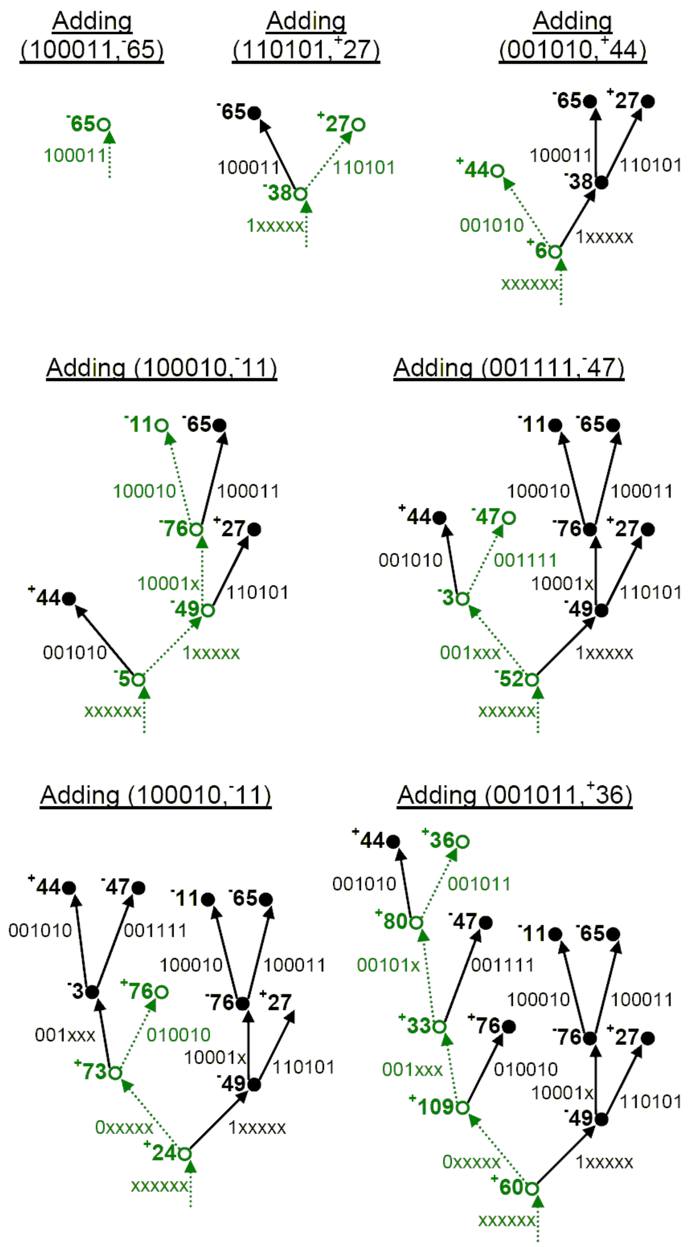


Figure 4.11: An illustrative example of using the *treeadd* operation (see Fig. 4.9). Each stage in the diagram illustrates the addition of a new list item to the tree. Hollow circles and dotted arrows represent newly added nodes and pointers. The occlusion levels are represented here in binary, whereas the values to be stored and summed are represented in decimal. The letter "x" may be digit 0 or 1; for example, "001xxx" represents the range 001000 to 001111. For the detailed workings of this tree structure, refer to Fig. 4.10.

rows at the next level, then sets of 4 adjacent rows, then sets of 8 adjacent rows, etc. For each level of the hierarchy, and each row group, a one-dimensional array (one location for each horizontal position) of trees is constructed (by *createstruct* in Fig. 4.9), each of which can be used to efficiently look up the sum of all values in that row group to the left of the given column. These trees are referred to herein as *occlusive sum look-up trees.* The look-up tree for each location in each of these one-dimensional arrays can be considered to hold an array of partial sums, one for each occlusion level.

The final block (*Occlusive Sum Extractor*) reads the occlusive sum look-up structure for every pixel in the image. To calculate the cumulative occlusive sum for a given row, column and occlusion level, the appropriate partial sums from the appropriate row groups are separately extracted then added together (as in *extractsum* in Fig. 4.9).

The workings of the occlusive sum look-up tree are given in detail in Fig. 4.10 and illustrated in Fig. 4.11. Each tree node T, at the root of its own subtree, may be regarded as a pointer to a tuple, (V_T, R_T, L_T, U_T), where $V_T \in \mathbb{R}$ is the grey level (or colour component) total for the subtree, $R_T \subset \mathbb{R}$ is the smallest contiguous subset of $\mathbb{R}$ which spans all ω (occlusion) values covered by the subtree, and L_T and U_T are pointers to the lower and upper branch nodes which (optionally) sprout from node T. The occlusive sum look-up tree has the following properties:

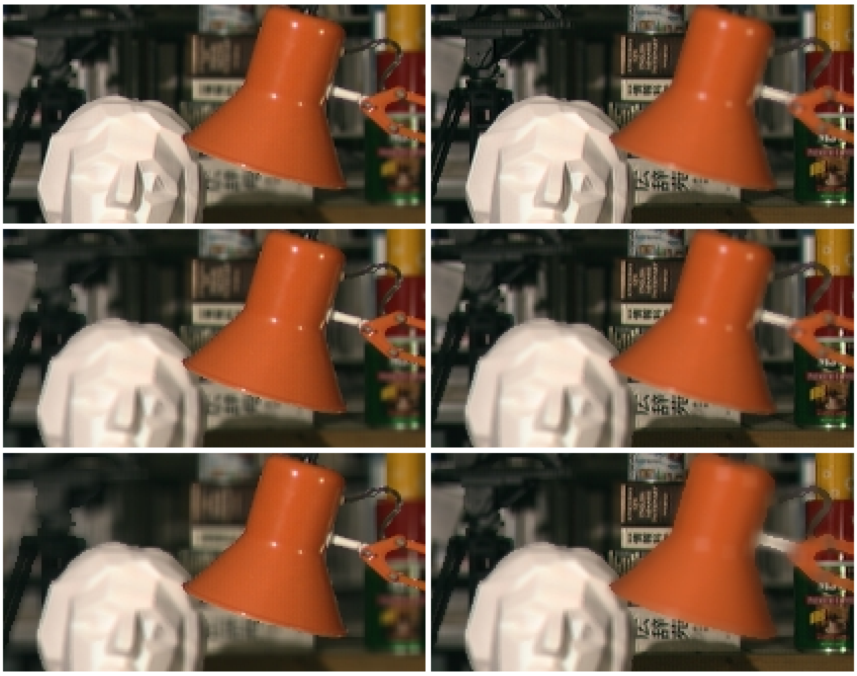
- When an addition or removal is made to the tree, an unaltered copy of how the tree was before may be retained at no extra cost, and at each stage of the algorithm, all non-new nodes are shared with the previously constructed trees.

- The number of operations required to add a new occlusion level-value pair to a tree is of order $\mathcal{O}(log(M))$, where M is the maximum absolute value of any integer occlusion level.

- The amount of additional storage space required each time a new occlusion level-value pair is added is $\mathcal{O}(log(M))$.

- The number of operations required to look up a sum value for a given occlusion level is $\mathcal{O}(log(M))$.

The asymptotic complexity of applying Algorithm 1 to an N-pixel image ($N{=}HW$), with potentially N occlusion levels, is $\mathcal{O}(HW \log(H) \log(HW))$. Therefore, assuming a fixed aspect ratio as N gets larger, the total cost is of order $\mathcal{O}((\log N)^2 N)$. This compares



(a) (b)

Figure 4.12: Examples of occlusive selective blurring applied to Tsukuba head and lamp image and disparity map [141]. The depth map was simply taken as the reciprocal of the disparity map; then, each blur level was computed according to Eq. (4.1), with k chosen each time to attain a desired maximum blur level. (a) Lamp in focus. (b) Cans in focus. Max blur levels (i.e., max spread, in pixels, in any direction): top: 3; middle: 5; bottom: 10.

with the $\mathcal{O}(N^2)$ cost that would be required by the naïve approach of independently computing the spreading at each occlusion level.

Fig. 4.12 shows the results of applying Algorithm 1 to a raw image and disparity map with a variety of blur levels and focal points. When the nearest object (lamp) is not in focus, for high blur levels, a sharp occlusion boundary caused by the blurring algorithm can be seen around the edge of the lamp, where ideally the occlusion would occur gradually, with partial translucency covering a narrow part of the background; however, the background behind the original object is completely unknown to the algorithm, and the normalisation method (see equation (4.2)) causes the blur of the foreground to be brightened so as to obscure what would otherwise simply be rendered as black. Fig. 4.13 compares the output of the proposed technique to real-world depth of field effects as caused by the lens of a camera. The synthetic blur level has been chosen manually to provide a result which is visually almost identical to the true depth-of-field effects.

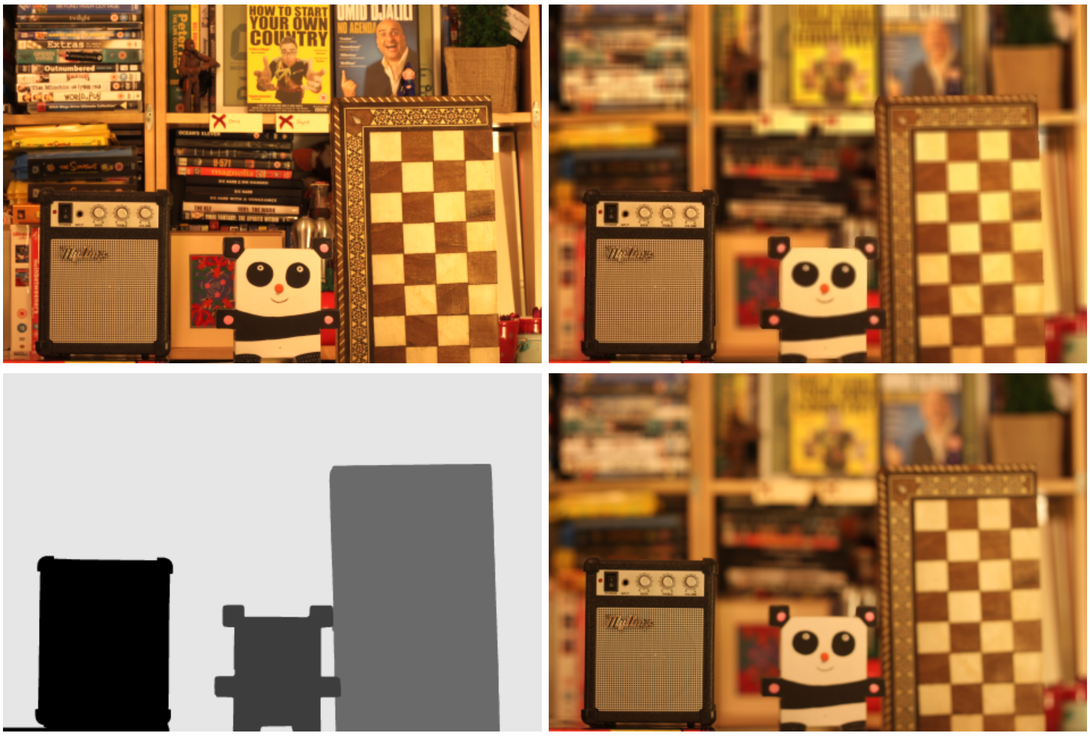


Figure 4.13: Visual comparison between the output of the proposed approach and true depth of field as cause by a camera lens. Top left: input image. Bottom left: manually-created input blur map (black: unblurred), the inverse of which was used as the occlusion map. Top right: emulated depth of field using the proposed approach, with average blur set to 3 (i.e., spread of 3 pixel widths in each direction). Bottom right: real depth of field created by the lens of a camera (a Canon EOS 500D with 50mm lens, set at F2.8).

However, close inspection of the blurred background reveals a slight presence of higher frequency components that is characteristic of the square blurring approach. Furthermore, the boundary of the blur on the edge of the chess board is sharp in the synthatic image but soft in the true depth-of-field image.

4.5 Implementation of Algorithm 2

The method used by Algorithm 2 for computing the occlusive pixel-spreading is shown in Fig. 4.14. The cumulative sum illustrated in Fig. 4.7 is performed by partitioning it into partial sums according to the location to sum up to. These partial sums are stored in a summation structure, $\mathcal{S}$, which is simpler than the structure used by Algorithm 1 (see section 4.4). This structure is a hierarchy of sums over $2^m \times 2^n$-sized rectangles, for each $m \in \{0, 1, 2, ..., \lfloor \log_2 W \rfloor\}$ and $n \in \{0, 1, 2, ..., \lfloor \log_2 H \rfloor\}$, where H and W are the image

$P = \boldsymbol{spread2}(G, B, \Omega)$

Inputs: colour plane G, blur map B, occlusion map Ω.

```
Create an empty summation structure $\mathcal{S}$.
Create a blank output colour plane $P$.
Create an empty list $L$ of 4-element tuples.
for each pixel location $\mathbf{x} \in D$ do
    Append tuple $(\mathbf{x}, g, b, \omega) = (\mathbf{x}, G(\mathbf{x}), B(\mathbf{x}), \Omega(\mathbf{x}))$ to $L$.
end for
Sort $L$ in descending order of $\omega$ values (occ. levels).
for each different $\omega$ value (highest first) in sorted list $L$, do
    for each tuple $(\mathbf{x}, g, b, \omega)$ in $L$, of the given $\omega$, do
        Set $v = g/(2b+1)^2$.
        Set $\mathcal{S} = \boldsymbol{store}(\mathbf{c}^{11}(\mathbf{x}, b), v, \mathcal{S})$,
        then $\mathcal{S} = \boldsymbol{store}(\mathbf{c}^{22}(\mathbf{x}, b), v, \mathcal{S})$,
        then $\mathcal{S} = \boldsymbol{store}(\mathbf{c}^{12}(\mathbf{x}, b), -v, \mathcal{S})$,
        then $\mathcal{S} = \boldsymbol{store}(\mathbf{c}^{21}(\mathbf{x}, b), -v, \mathcal{S})$.
    end for
    for each tuple $(\mathbf{x}, g, b, \omega)$ in $L$, of the given $\omega$, do
        Set $P(\mathbf{x}) = \boldsymbol{get}(\mathbf{x}, \mathcal{S})$.
    end for
end for
```

Output: pixel-spreaded colour plane P.

$\mathcal{S} = \boldsymbol{store}(\mathbf{x}, v, \mathcal{S})$

Inputs: location $\mathbf{x} = (x, y)$, pixel value v, summation structure $\mathcal{S}$.

```
if $1 \le x \le W$ and $1 \le y \le H$ then
    Set $m = 0$ and $x' = x$.
    repeat
        if $x'$ is odd then
            Set $n = 0$ and $y' = y$.
            repeat
                if $y'$ is odd then
                    Set $\mathcal{S}(m, n, \lceil \frac{x'}{2} \rceil, \lceil \frac{y'}{2} \rceil)$ += $v$.
                end if
                Set $y' = \lceil \frac{y'}{2} \rceil$ and $n$ += 1.
            until $2^n > H$.
        end if
        Set $x' = \lceil \frac{x'}{2} \rceil$ and $m$ += 1.
    until $2^m > W$.
end if
```

Output: modified summation structure $\mathcal{S}$.

$s = \boldsymbol{get}(\mathbf{x}, \mathcal{S})$

Inputs: location $\mathbf{x} = (x, y)$, summation structure $\mathcal{S}$.

```
Set $s = 0$, $m = 0$ and $x' = x$.
repeat
    if $x'$ is odd then
        Set $n = 0$ and $y' = y$.
        repeat
            if $y'$ is odd then
                Set $s$ += $\mathcal{S}(m, n, \lceil \frac{x'}{2} \rceil, \lceil \frac{y'}{2} \rceil)$.
            end if
            Set $y' = \lfloor \frac{y'}{2} \rfloor$ and $n$ += 1.
        until $y' = 0$.
    end if
    Set $x' = \lfloor \frac{x'}{2} \rfloor$ and $m$ += 1.
until $x' = 0$.
```

Output: sum s.

Figure 4.14: The step-by-step details of *spread2*, the approach used by Algorithm 2 for occlusive pixel-spreading of a single colour plane, and two operations it employs. The operations *get* and *store* are for storing and reading a value in/from the summation structure which Algorithm 2 employs. Here, W and H are the width and height of the image, and $\lceil\ \rceil$ and $\lfloor\ \rfloor$ denote integer upward and downward rounding, and functions $\mathbf{c}^{11}$, $\mathbf{c}^{12}$, $\mathbf{c}^{21}$ and $\mathbf{c}^{22}$ yield corner points as defined in equations (4.6) to (4.9). See Fig. 4.15 for an illustrative explanation of the format of the summation structure which Algorithm 2 employs and an illustrative example of a summation performed by a *get* operation.

height and width, and $\lfloor$ and $\rfloor$ denote the integer part with rounding downwards. For each pair (m, n), the corresponding $2^m \times 2^n$ rectangles form a non-overlapping covering of the image domain D, and beyond if 2^m and 2^n do not divide W and H exactly. The occlusive aspect is handled by progressively incorporating the effect of each pixel into the summation in descending order of occlusion level, so that no pixel of the blurred image is affected in any way by any further-away pixels. The operation *spread*, for occlusive pixel-spreading of a single colour plane, is described in terms of operations *get* and *store* (in Fig.

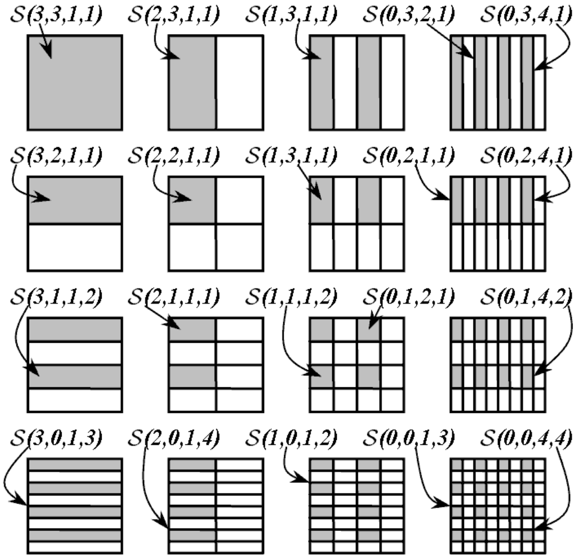


Figure 4.15: Illustration of the format of Algorithm 2's summation structure, $\mathcal{S}$, in the case of an 8×8 greyscale image or colour plane. Each $2^m \times 2^n$ region is shown in grey, with a selection of these regions labelled with the corresponding $\mathcal{S}(m, n, x, y)$ values that give their sums.

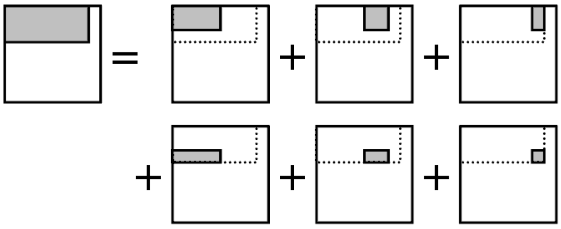


Figure 4.16: A conceptual illustration of a *get* operation (see Fig. 4.14) in the case of a cumulative sum to location (7,3) in an 8×8 image. The cumulative sum (summing over the image from top left) is separated into a number of partial sums as stored in the summation structure.

4.14). These handle the summation structure, as illustrated in Fig. 4.15. The summation structure facilitates dynamic cumulative summation; that is, it allows efficient computation of cumulative sums of an image (summing from top-left), while allowing the underlying image to be dynamically altered. Every possible top-left-justified rectangle can be uniquely represented as the union of one or more of the grey rectangles illustrated in Fig. 4.15. Therefore every cumulative sum can be represented as a sum of appropriate $\mathcal{S}(m, n, x, y)$ values (denoted by $s = \mathit{get}(\mathbf{x}, \mathcal{S})$, giving cumulative sum s for a given location $\mathbf{x}$ and summation structure $\mathcal{S}$). Fig. 4.16 shows an example of this. The dynamic alteration works as follows: whenever a pixel of the underlying image is adjusted, the same adjustment is

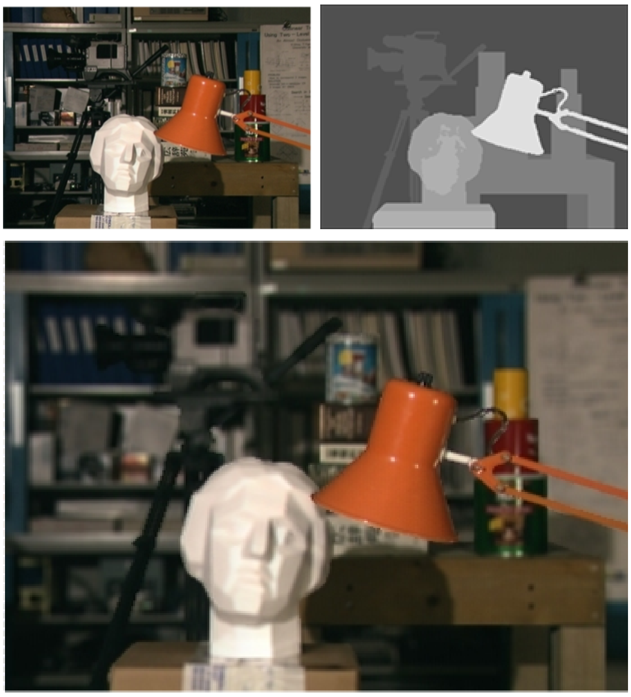


Figure 4.17: Top left: raw image (*Tsukuba head & lamp*). Top right: disparity map. Bottom: occlusively selectively blurred image. At each point the occlusion level was taken directly as the negative of the disparity level and the blur level was directly proportional to the difference in disparity between the local disparity level and that of the in-focus point (the head in the foreground).

made to the $\mathcal{S}(m, n, x, y)$ value of every grey region in Fig. 4.15 that overlaps the given pixel (denoted by $\mathcal{S}' = \mathit{store}(\mathbf{x}, v, \mathcal{S})$, for a given location $\mathbf{x}$, intensity value v, summation structure $\mathcal{S}$ and modified summation structure $\mathcal{S}'$).

Fig. 4.17 shows the results of applying Algorithm 2 to a raw image and disparity map. Each disparity $d(\mathbf{x})$ at pixel location $\mathbf{x}$ was converted into a blur level $b(\mathbf{x})$ according to $b(\mathbf{x}) = k|d(\mathbf{x}) - d_0|$, where d_0 represents the in-focus disparity and k was a constant chosen in order to obtain a given maximum blurring level. This is an example of the level of realism achievable with a sufficiently detailed disparity map. At the low blurring level employed here (max blur: 5 pixel widths), with the nearest object (the lamp) chosen to be in focus, there are virtually no visual indications that the blurring is synthetic.

4.6 Computational Complexity

The asymptotic complexity of applying Algorithm 2 to an N-pixel image ($N{=}HW$), with potentially N occlusion levels, can be broken down as follows: A call to either *store* or *get* involves $\mathcal{O}(\log(H)\log(W))$ primitive operations. In operation *spread2* (Fig. 4.14), the *for* loop in lines 8 to 16 involves $4HW$ calls to *store* and HW calls to *get*, so is therefore an $\mathcal{O}(HW\log(H)\log(W))$ operation. The sorting performed in line 7 can be regarded as an $\mathcal{O}(HW\log(HW))$ operation. The *for* loop in lines 4 to 6 involves HW primitive operations. Therefore, *spread2* is overall an $\mathcal{O}(HW\log(H)\log(W))$ operation, as the $\mathcal{O}(HW\log(H)\log(W))$ cost dominates. Therefore, this gives an overall cost of $\mathcal{O}(HW\log(H)\log(W))$ operations for a single call to the *blur* operation (Fig. 4.6). Therefore, assuming a fixed aspect ratio as N gets larger, the cost is of order $\mathcal{O}((\log N)^2N)$, which is the same as Algorithm 1. As for Algorithm 1, this compares with the $\mathcal{O}(N^2)$ cost that would be required by the naïve approach of independently computing the spreading at each occlusion level.

4.7 Conclusion

This chapter has made a case for the use of depth-blurred coding, which works by using selective preblurring of a style which a human viewer is intended to mistake for depth blur effects that naturally occur in cameras. This approach is particular appealing now given the increasing availability of cameras providing depth information (e.g., stereo or time-of-flight cameras or set-top boxes with software for 2D to 3D video conversion).

In the context of a general depth-blurred coding coding architecture, two algorithms have been proposed for the computation of depth blurring (including occlusion effects) which both have, for an N-pixel image, an $\mathcal{O}((\log N)^2N)$ cost regardless of the number of blurring levels and occlusion levels to be dealt with. Both algorithms aim to compute precisely the same output, and assume a sharp cut-off of the blurred light from far objects by the boundaries of nearer objects, even if those boundaries are themselves blurred.

This chapter has argued that that this novel approach to space-variant coding will generally be visually more acceptable than the equivalent level of foveation filtering (the more common space-variant approach) if the viewer looks away from predicted points or regions of fixation. The question of whether the depth-blurred coding concept performs better than foveation filtering will be addressed by the next chapter.

Chapter 5

Subjective Evaluation

This chapter presents subjective evaluation comparing the perceived quality of a foveated image coding technique with a depth-based preblurring technique under equivalent conditions, including presenting a method-of-adjustment approach for measuring the overall perceived image quality in terms of equivalent JPEG*. The results of a subjective comparison between depth-blurred coding and foveated coding are presented.*

5.1 Introduction

As mentioned in chapter 4, the question of whether the depth-blurred coding concept is any better than a more common space-variant approach is an open one. This chapter reports on experiments for demonstrating and quantifying the difference in perceived quality between depth-blurred coding and the equivalent foveated coding for still images. To evaluate the relative merits of two styles of space-variant blurring – depth and foveation blurring – the evaluations presented herein work by assessing them in an image coding context, by applying each of them as a preprocessing stage prior to an ordinary JPEG codec. The results provide empirical evidence as to whether depth-based preblurring is better in terms of the level of overall perceived quality than the equivalent foveated preblurring.

The chapter is organised as follows: section 5.2 describes the two evaluation methods employed; section 5.3 provides the results and discusses the outcomes; section 5.4 concludes the chapter.

5.2 Method

5.2.1 General Test Framework

Two types of test were performed, namely a single stimulus test and a *method-of-adjustment* [130, p. 27] test. The first type of test, referred to herein as *SSCQS* was a single-stimulus modified version of Variant I of the *Double-Stimulus Continuous Quality Scale (DSCQS)* method [119], producing a score for each image in the range 0 to 100 according to the ITU-R five-point adjective quality scale (*Excellent/Good/Fair/Poor/Bad* – see section 2.6.2). These tests deliberately avoided showing any image under test alongside its reference image, so as to address the possibility that if an observer were to see a processed image in conjunction with its unprocessed original, he may become aware of distortions that he might otherwise not notice.

The method-of-adjustment test was designed to address the anticipated problem of the SSCQS tests.

The test images were chosen to present a mixture of two types: (i) images for which high-fidelity depth maps or disparity maps are available; (ii) images for which a two-level foreground/background manual segmentation had been performed. The first of these image types allowed the depth-blurred coding to be evaluated in its ideal scenario, in which full depth map information was available. The second type allowed a cruder, simpler version of depth-blurred coding to be evaluated alongside. In both cases, each image had a predominant feature such as a face, which can be argued to be a strong attractor of attention (see section 2.2.2).

5.2.2 Method-of-Adjustment Tests

The method-of-adjustment test was intended to address the anticipated problem of the difference in interpretation, by different people, of the adjectives used to label the ITU-R five-point quality scale. This test used specially-written test software which performed JPEG coding of the unblurred test image in real time to produce, for each image, an *equivalent distortion* as judged by the subject. Each subject was instructed as follows: “The right-hand picture is associated with a vertically-sliding scale that will change the picture’s quality when you move it. You are asked to move this sliding scale up or down until it is your opinion that both pictures have the same overall quality." In order to represent this quality on a meaningful scale, the compression ratio of each image was recorded, in logarithmic form; specifically, $\log(b_2/b_1)$ was stored, where b_1 is the bitrate of the (preblurred) image under test and b_2 is the bitrate of the equivalent-quality unblurred JPEG-encoded image. The reason for the logarithm is that it reduces sensitivity to outliers when processing results, as opposed to using the compression ratio directly.

Unlike the first test type, this test was a double-stimulus test, in which the image under test was compared with its corresponding unencoded reference image. The two different blurring types were never shown for comparison directly with each other; whenever two versions of the same image were compared in the same presentation, they were always the preblurred image compared with the original unblurred image (the reference image).

5.2.3 Preblurring and Encoding

The depth-blurred coding scheme and the foveated coding scheme were both implemented as selective preblurring approaches, with the blurred images passed into an off-the-shelf JPEG encoder. The experiment used ordinary JPEG instead of, for example, JPEG 2000 because of the similarity shared between JPEG's DCT block encoding scheme and that of most MPEG and ITU-T video coding formats.

For the blur map used in the foveation filtering technique, the experiment used the model of the eccentricity-dependent variation of eye sensitivity as proposed by Geisler and Perry in their *contrast threshold formula* [28]. Specifically, the cutoff frequency interpretation of Wang, Lu & Bovik [21] was employed, whereby, for all e,

$$f_c(e) = \frac{e_2 ln(1/CT_0)}{(\|e\| + e_2)\alpha}, \tag{5.1}$$

where $f_c(e)$ is the spatial cutoff frequency (in cycles per unit angle) for a given retinal eccentricity, e (that is, the angle, relative to the observer's eye, between a given point and the point of focus), and e_2, α and CT_0 are constants defined as follows: $e_2 = 2.3°$, $\alpha = 0.106°/\text{cycle}$ and $CT_0 = 1/64$ (see Fig. 5.1). The viewing direction was assumed to be head-on and angles were taken directly from pixel co-ordinates using a fixed conversion factor based on a viewing angle of one pixel width at the nearest point of the image to the viewer. (this approximation becomes increasingly valid as the viewing distance increases).

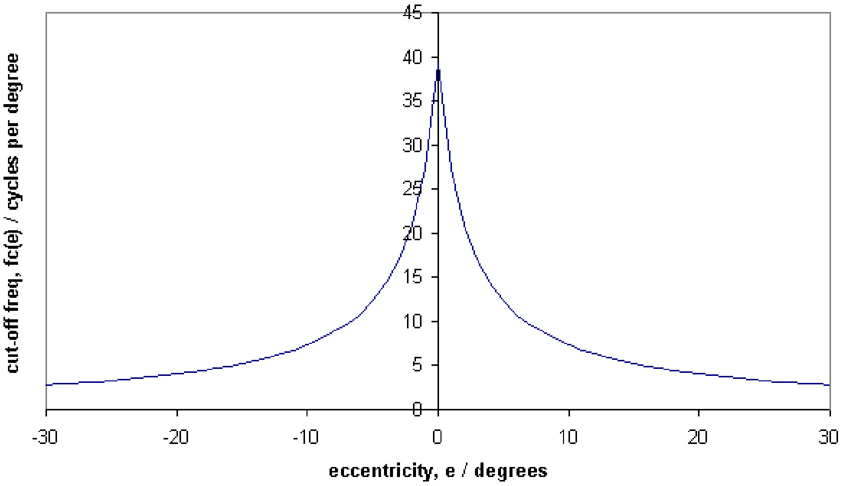


Figure 5.1: Model of the spatial cut-off frequency curve of the human eye.

Given any distance, d, in pixels, from the chosen point of interest, this was converted into an angle, e, as follows:

$$e = \frac{360^\circ\ d}{2\pi HR}, \tag{5.2}$$

where H represents the image height (in pixels) and R represents the distance÷height ratio (calculated assuming a chosen viewing distance and screen resolution). This angle was then passed into equation 2.3 to produce the cut-off frequency, $f_c(e)$. The conversion from $f_c(e)$ values into a blur map was done by calculating the corresponding $1/f_c(e)$ values and scaling them proportionally, such that the mean blur level becomes a predefined value. This scaling of blur levels was intended to reflect a desire that, assuming the viewer's focus is fixed on the given point, the imposed blurring would then be, in relative terms, the same at each location in comparison to the eye's resolution at that location. This is appropriate if it can be assumed that, across the retina, every local sensitivity-versus-spatial-frequency curve is a frequency-scaled version of the same curve.

In order to provide a fair comparison between depth-blurred coding and foveated coding, both blurring schemes were implemented using exactly the same occlusive selective blurring code (of depth blurring Algorithm 1, as defined in sections 4.3 and 4.4), with the only differences being the blur maps and occlusion maps used. All the foveation blurring was done using a uniformly-valued occlusion map, such that no occlusion effects would take place. The blur maps used for the depth blurring were created in a manner such that, firstly, the chosen point of interest is in focus and, secondly, the blur-level histogram (that is, the distribution of the number of pixels incurring each blur level) is exactly the same as for the foveation blurring. This is to make it so that it can only be the spatial distribution and nature of the blurring, rather than the amount of blurring, that determines the outcome of these experiments. In place of depth maps, disparity maps were used. Disparity maps give the disparity, for each pixel, between a stereo camera pair and thus have a direct mapping to distance values; however, only the relative ranking of disparity values was used here (relating directly to the relative ranking of distance values). Each image location in the disparity map was ranked in order of how close its disparity value was to that of the chosen point of interest of the image; the blur levels were then assigned in order of blurring, such that the point of interest was the sharpest in focus. When the disparity values alone were not sufficient to define the ordering of the pixels (that is, when the same disparity value is shared by more than one pixel), the order is resolved according to their image-plane distances from the point of interest (note that this therefore means that, if a one-level disparity or depth map is used, it will have exactly the same blur map as the foveation blurring, because of the fact that the foveation blur map has radial symmetry about the point of interest).

With the objective of seeing how the level of the different types blurring affects the perceived quality of the result, a number of different mean blurring levels were applied. The blurring level here represents roughly half the width of the square area over which the corresponding pixels were spread (the square window has width $2b + 1$ for blurring level b). Each blurring level was rounded to the nearest integer value before being used, so that it was not necessary to perform inter-pixel interpolation at any stage.

The selectively preblurred images were JPEG encoded using the same JPEG encoder, with the input quality parameter adjusted by binary search until a desired bitrate was obtained as closely as possible. For each different blur level, a number of bitrates were applied so as to obtain a broad view of the differences between depth-blurred coding and foveated coding at different bitrates.

5.2.4 Details of the Experiment

Forty-three non-expert subjects performed both types of tests for a selection of test images, blur levels and bitrates. Precautions were taken to ensure subjects were ignorant of any

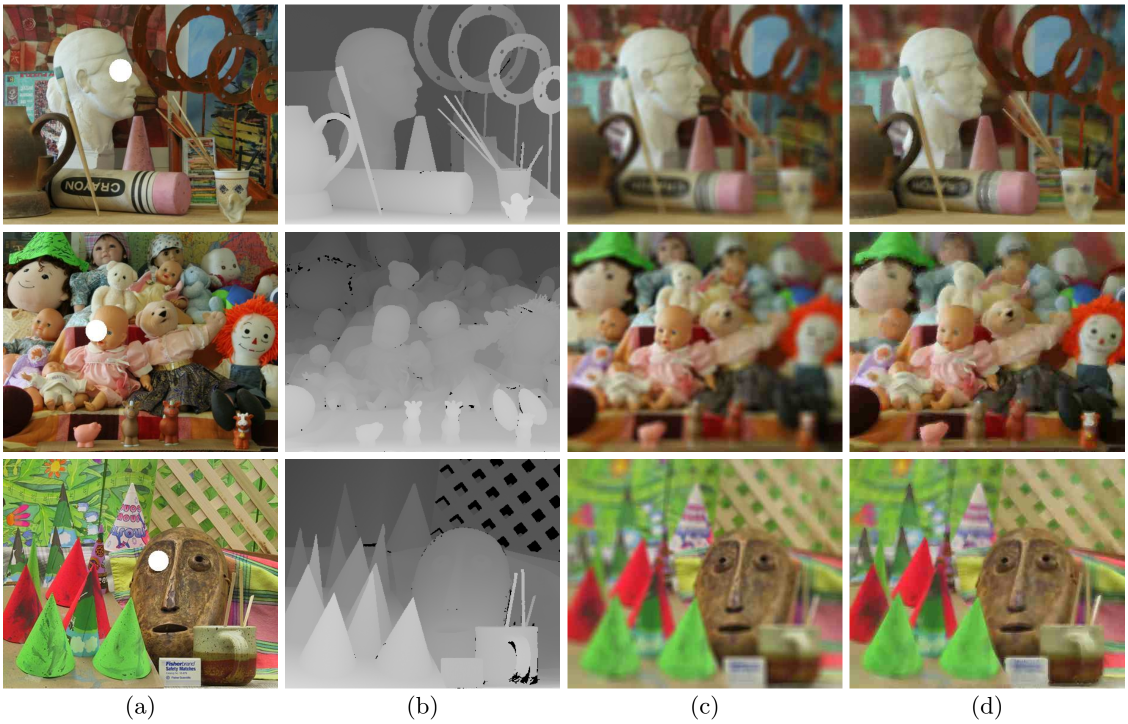


Figure 5.2: (a) The three Middlebury test set images used (top to bottom: Art, Dolls and Cones), with each one's manually-chosen point of interest highlighted by a white circle; (b) the associated disparity maps; (c) examples of foveation blurred test images; (d) the equivalent depth-based blurred test images. Test image bit rates: all 0.4 ± .004 bpp. Mean blur levels: all 10 pixel-widths. In the disparity maps, it is possible to notice the errors in the form of black patches.

processing that had been applied to the images. All subjects had normal or corrected to normal vision. In both types of test, each subject was asked to assess overall quality; when the meaning of this was not clear, the subject was asked to think in terms of his preference in choosing an image as the backdrop of his computer desktop.

Six publicly available images were chosen for the experiments. Three of them, *Cones* (450×375) *Dolls* (463×370) and *Art* (463×370), came from the Middlebury Stereovision test set [138, 139] along with their associated continuously-varying disparity maps. Each of the remaining three was the first frame of the well-known *Akiyo*, *Foreman* and *Silent* video sequences (all 352×288). For each of the latter three images, a two-level depth map was manually generated such that, in each case, the human in the scene was segmented as foreground, with the rest of the scene as background. Only test images with faces or face representations were selected, as these are known to instinctively attract human fixation [75]. This homogeneous choice of test class allows the strongest possible chance

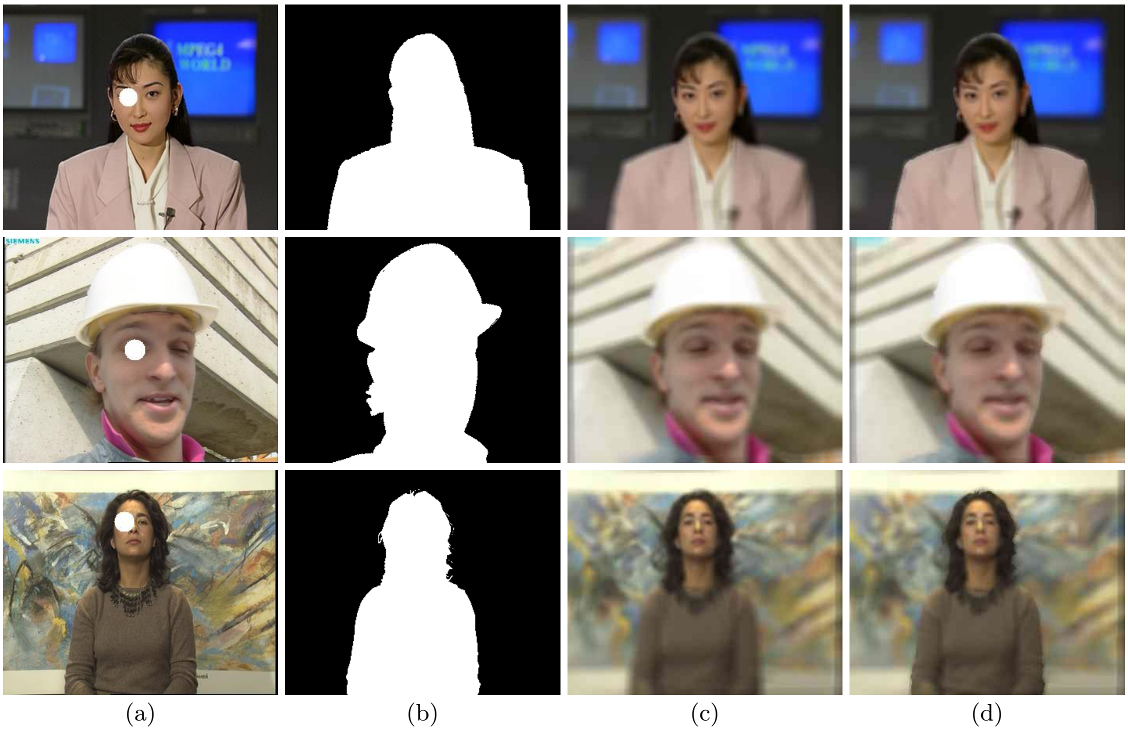


Figure 5.3: (a) The three two-level-disparity-map images used (top to bottom: Akiyo, Foreman and Silent, with each one's manually-chosen point of interest highlighted by a white circle; (b) the associated disparity maps; (c) examples of foveation blurred test images; (d) the equivalent depth-based blurred test images. Test image bit rates: all 0.4±.004 bpp. Mean blur levels: all 10 pixel-widths.

of an assumption of a single fixation point being satisfied, as non-face images may not have such strongly-attractive points of interest. An assumed fixation point was manually placed on a face in each case, reflecting a common approach in the assessment of foveation techniques (e.g., [9, 20, 145]). This is also reflective of the availability of face detection techniques [97] that may be used in a practical coding application. The single fixation point assumption reflects a number of foveated coding publications [10,20,32,43]. Figs 5.2 and 5.3 show the raw test images and disparity maps used, along with example blurred test images. Except in the case of *Dolls*, in all images it can be seen that a single face or face representation was present and that one of the eyes of this has been chosen as the assumed fixation point (marked with a large white spot). In the case of the blurred images in Fig. 5.3, for which two-level manual depth segmentation is employed, it is possible to observe on the depth-blurred images that the blurring increases towards the periphery, due to the closest-to-point-of-interest scheme employed to apply the foveal blur histogram at locations where the disparity values alone were not sufficient to define the ordering of the pixels.

All JPEG encoding was performed using the Sun Microsystems Standard JPEG Image Writer (v. 0.5) and focused on a range of low bitrates in the region of 0.5 bits per pixel (bpp) or less. Three blur levels were investigated: 3, 5 and 10 pixel widths (i.e. square window widths of 7, 11 and 21 pixels respectively). These levels were the average of the b values applied when spreading a pixel over an area of width $2b + 1$. Each b value was rounded to the nearest integer value before being used, so that inter-pixel interpolation would never be necessary. For the middle blur level, three fixed bitrates were applied: 0.3, 0.4 and 0.5 bpp; for the others, only 0.4 bpp was applied. For all the test images, the bitrate deviated by no more than 0.004 bpp from the given bitrate.

All images were shown at native screen resolution. The display resolution was 0.264 mm per pixel and assumed a viewing distance of 40 cm (±10 cm).

All the results of both tests types were passed through the recommended screening for DSCQS tests [119], but no observers needed to be rejected.

5.3 Results

Table 5.1 shows the mean scores of both test types for both styles of preblurring. In each case, a higher score is better than a lower score. The SSCQS scores were normalised so that each subject had the same mean and standard deviation as the overall mean and standard deviation across subjects; following this, all statistics were computed using the recommended DSCQS formulae [119]. For SSCQS, the statistics were computed on the scores directly. For the method-of-adjustment tests, the statistics were computed on the log compression ratios; however, to add meaning to these figures, their antilogarithms are displayed (i.e., compression ratios instead of log compression ratios); therefore *mean* here is the *geometric mean* for the method-of-adjustment results (but the normal, *arithmetic mean* for the SSCQS results). Comparison figures between the equivalent results for foveated and depth-based blurred images are also given; for the SSCQS results these are the mean score differences (positive = “depth-based is better”) whereas for the method-of-adjustment results they are relative ratios (greater than one = “depth-based is better”). For the SSCQS results, the half-widths of the 95% confidence intervals are given (i.e., true = estimate ± CI), while the multiplicative equivalents of these are given for the method-

Table 5.1: SSCQS and Method-of-Adjustment Test Scores

Raw image	Blur level	Rate (bpp)	Normalised SSCQS score (0-100)				Method-of-adjustment score (cmpr ratio)			
			Fovea-ted (mean)	Depth-blurred (mean)	Comparison		Fovea-ted (mean)	Depth-blurred (mean)	Comparison	
					Mean	C.I.			Mean	C.I.
Cones	3	0.4	**48.42**	**42.82**	**-5.61**	**5.10**	1.121	1.141	1.019	1.113
	5	0.3	25.13	22.91	-2.22	4.73	1.254	1.253	1.000	1.089
		0.4	43.11	46.07	2.96	5.45	**0.982**	**1.120**	**1.141**	**1.081**
		0.5	51.53	54.63	3.10	6.17	0.880	0.859	0.976	1.117
	10	0.4	**29.00**	**43.85**	**14.85**	**4.14**	**0.873**	**1.014**	**1.162**	**1.156**
Dolls	3	0.4	43.00	42.27	-0.73	3.91	1.159	1.207	1.042	1.107
	5	0.3	22.32	22.26	-0.06	4.11	1.153	1.191	1.033	1.065
		0.4	39.56	44.47	4.91	5.06	1.015	1.062	1.046	1.062
		0.5	**50.48**	**57.86**	**7.38**	**4.96**	0.899	1.018	1.133	1.138
	10	0.4	**24.83**	**36.87**	**12.03**	**5.16**	**0.862**	**0.963**	**1.118**	**1.083**
Art	3	0.4	53.03	56.71	3.68	5.28	1.036	1.074	1.036	1.100
	5	0.3	33.53	33.63	0.09	4.51	**1.087**	**1.211**	**1.114**	**1.104**
		0.4	**50.16**	**55.00**	**4.84**	**3.97**	1.009	1.032	1.023	1.075
		0.5	**56.07**	**66.71**	**10.64**	**5.05**	0.902	0.940	1.043	1.099
	10	0.4	**32.50**	**46.44**	**13.94**	**5.11**	**0.857**	**0.956**	**1.116**	**1.110**
Akiyo	3	0.4	62.97	62.41	-0.56	4.28	0.999	1.037	1.038	1.081
	5	0.3	34.72	36.38	1.66	5.29	1.094	1.122	1.026	1.041
		0.4	61.52	62.44	0.93	3.38	**1.019**	**0.936**	**0.918**	**1.078**
		0.5	68.26	69.32	1.06	3.80	0.824	0.824	1.000	1.090
	10	0.4	**24.90**	**32.29**	**7.39**	**4.88**	0.750	0.747	0.995	1.040
Foreman	3	0.4	40.03	38.60	-1.43	3.85	1.087	1.131	1.040	1.051
	5	0.3	19.18	21.21	2.03	2.85	1.235	1.247	1.010	1.056
		0.4	39.31	41.69	2.38	4.13	1.082	1.068	0.987	1.049
		0.5	45.66	50.55	4.90	5.58	0.911	0.912	1.001	1.067
	10	0.4	22.48	25.39	2.91	3.88	0.946	0.958	1.013	1.064
Silent	3	0.4	**32.86**	**38.66**	**5.79**	**4.84**	1.087	1.139	1.048	1.073
	5	0.3	17.22	17.70	0.49	4.05	1.242	1.277	1.028	1.043
		0.4	**33.14**	**40.59**	**7.45**	**4.45**	1.117	1.199	1.073	1.091
		0.5	52.83	57.67	4.83	4.94	1.068	1.156	1.082	1.084
	10	0.4	**21.68**	**28.97**	**7.29**	**4.74**	**0.926**	**1.006**	**1.086**	**1.085**

of-adjustment results (i.e., true = estimate $\overset{\times}{\div}$ CI). That is, by using the edges of these confidence as significance thresholds, there is considered to be less than a 5% chance of wrongfully rejecting the null hypothesis (that neither image is better than the other) in favour of the alternative hypothesis (that one of the images is better than the other). Significant results (where the *comparison* values lie outside the confidence intervals) are highlighted in bold font.

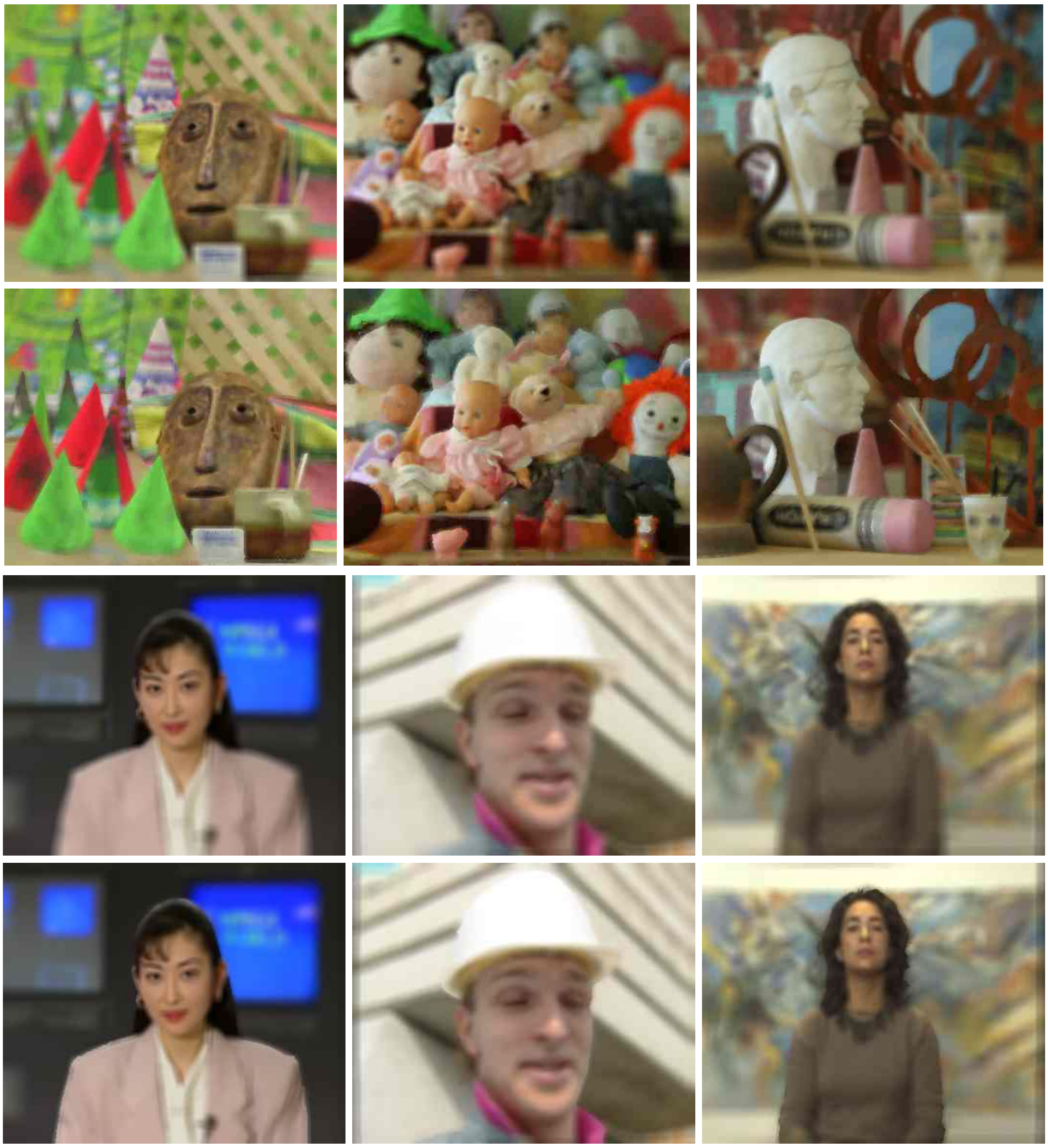


Figure 5.4: Test images in the case of mean blur level 10 and 0.4 bpp encoding. Rows 1 and 2, left to right: *Cones*, *Dolls* and *Art*. Rows 3 and 4, left to right: *Akiyo*, *Foreman* and *Silent*. Rows 1 and 3 employ foveation-preblurring, and rows 2 and 4 give the equivalent depth-preblurred images. The point of interest is the right eye of the face or face-like object in each image (the central doll in the case of *Dolls*). The depth-preblurred images in row 2 were generated using multi-level disparity maps were from the Middlebury test set (see Fig. 5.2). The depth-preblurred images in row 4 were generated using two-level depth maps which separate the foreground (person) from the background (see Fig. 5.3, right).

5.3.1 Discussion of the Results

The average of the compression ratios found by the method-of-adjustment tests across the test images was 1.016 for the foveated images, and 1.060 for the depth-blurred images.

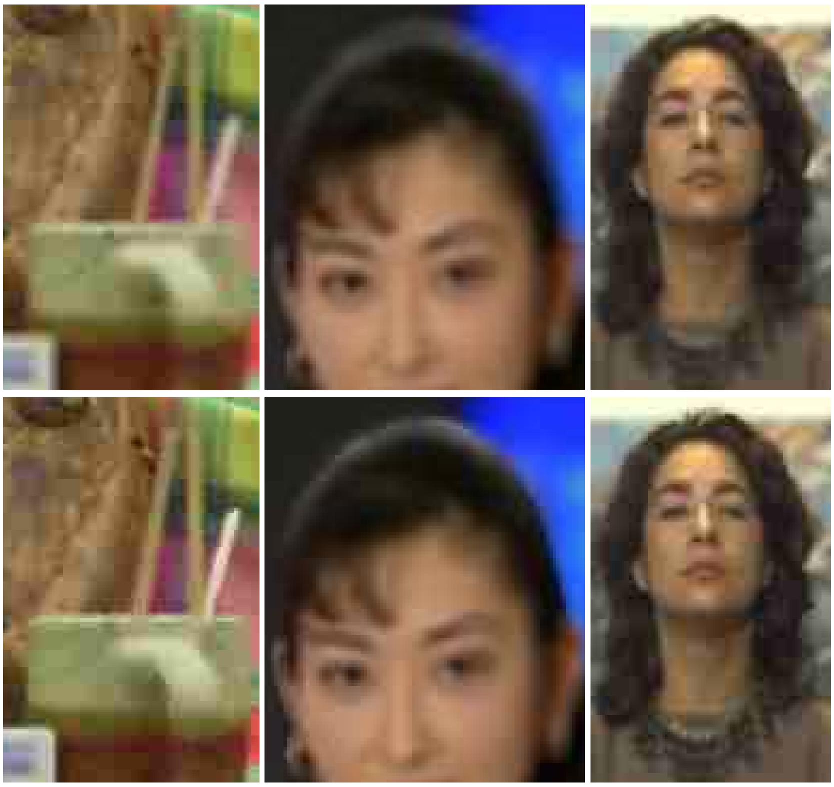


Figure 5.5: Zoomed-in portions of sample test images of (left to right) *Cones* (blur 5, 0.4 bpp), *Akiyo* (blur 10, 0.4 bpp) and *Silent* (blur 5, 0.4 bpp). Top row: foveation preblurring; bottom row: depth-based preblurring. All these images yielded significant results in at least one of the two test types, all in favour of depth-based blurring.

That is, the average foveated JPEG image was as good as the equivalent unblurred JPEG image with 1.6% more bits, and the average depth-preblurred JPEG image was as good as the equivalent unblurred JPEG image with 6.0% more bits. These figures are shown in Table 5.2, along with the average foveated-versus-depth-blurred comparison scores for the method-of-adjustment tests and also for the SSCQS tests.

All test images for the maximum blur level are shown in Fig. 5.4. Figs 5.5, 5.6 and 5.7 show sample extracts from test images of varying blurs and bitrates. In Fig. 5.5, the selected regions highlight distinctive differences between the foveated and depth blurred images which may have contributed to the subjective preference of one over the other: in *Cones*, the foveal blur of the paintbrushes obscures detail which the depth blurred image partly preseve; in *Akiyo*, the distinctive boundary of the head is far more blurred in the foveated image; in *Silent*, the relative sharpness of the necklace is a notable advantage of the depth blurring. In 5.6, a chance occurrence of notable JPEG artifacts around the eye

Table 5.2: Mean scores across images

Blur Level	Rate (bpp)	Normalised SSCQS		Method-of-adjustment					
		Comparison mean		Foveated mean		Depth blurred mean		Comparison mean	
		Cones, Dolls & Art	Akiyo, Foreman & Silent	Cones, Dolls & Art	Akiyo, Foreman & Silent	Cones, Dolls & Art	Akiyo, Foreman & Silent	Cones, Dolls & Art	Akiyo, Foreman & Silent
3	0.4	-0.89	1.27	1.105	1.058	1.141	1.102	1.032	1.042
5	0.3	-0.73	1.39	1.165	1.190	1.218	1.215	1.049	1.021
	0.4	4.24	3.58	1.002	1.073	1.071	1.068	1.070	0.993
	0.5	7.04	3.60	0.894	0.934	0.939	0.964	1.051	1.028
10	0.4	13.61	5.86	0.864	0.874	0.978	0.904	1.132	1.031
Column mean:		4.65	3.14	1.006	1.026	1.069	1.051	1.067	1.023
Overall mean:		3.90		1.016		1.060		1.045	

for blur level 3 image explains the significant negative preference the subjects had for this image. In Fig. 5.7, the tradeoff between blur level and bitrate can be seen, with a higher level of JPEG artifacts visible in the least blurred image than in the most blurred image at the same bitrate.

Overall, 14 out of 30 of the images gave statistically significant results in at least one of the two types of test. Of these, 12 results indicated an average preference for the depth blurring over the foveation blurring. The *Foreman* image yielded no statistically significant results whatsoever for any of its variants. This is understandable as, for this image, the visual differences between the two types of blurring are not easy to distinguish even when side by side (see Fig. 5.4). For this image, the area classed as foreground (the face) occupied a large portion (roughly 30%) of the image, and within this area, both types of blurring were identical, due to the preservation of blur-level histograms (see section 5.2.3). Two results only (the SSCQS test for the minimum-blur *Cones* image and the method-of-adjustment test for the 0.4 bpp mid-blur *Akiyo* image) gave a statistically significant results favouring the foveation blurring over the depth blurring. These can be explained by fact that at low blur levels and bit rates, the JPEG compression artifacts can have visual dominance over the blurring effects, and the question of which type of blurring has better perceived quality can become obscured by chance differences in the appearances of compression artifacts (see Fig. 5.6).

Of the statistically significant results, 9 were from the Middlebury test set images and 5 were from the images with two-level depth maps. This is reflected in higher cross-image

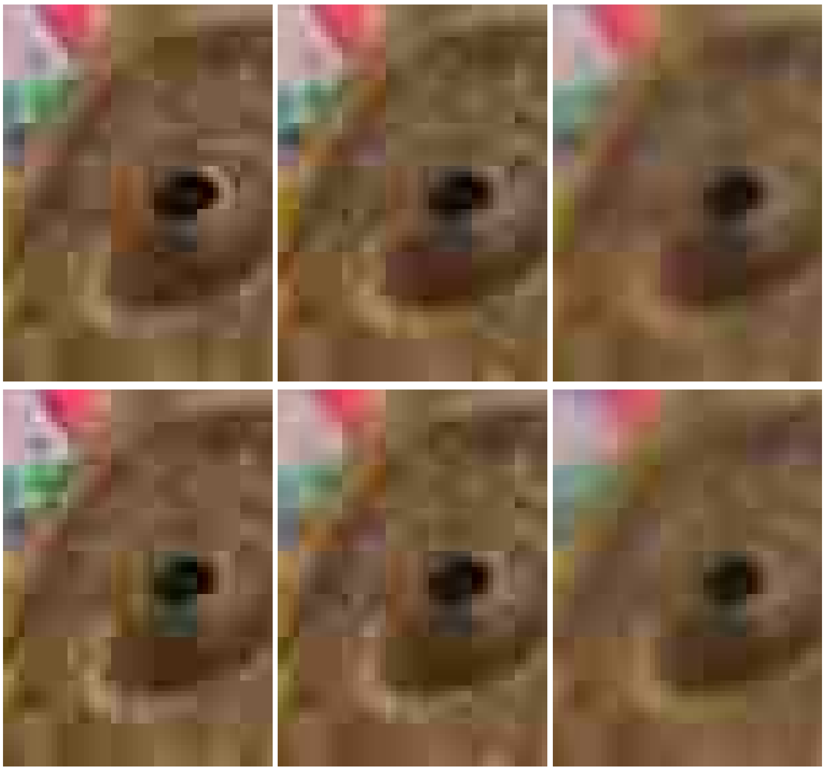


Figure 5.6: Sample zoomed-in extracts from different variants of the *Cones* image. Preblurring: top: foveated; bottom: depth-based. Blur levels, left to right: 3, 5 and 10. All encoded at 0.4 bpp. The pseudorandom JPEG artifacts around the eye can be seen to be worse for the depth-based blurring in the case of blur level 3 but worse for the foveation in the case of blur level 5, explaining the significant negative and positive score differences for these images.

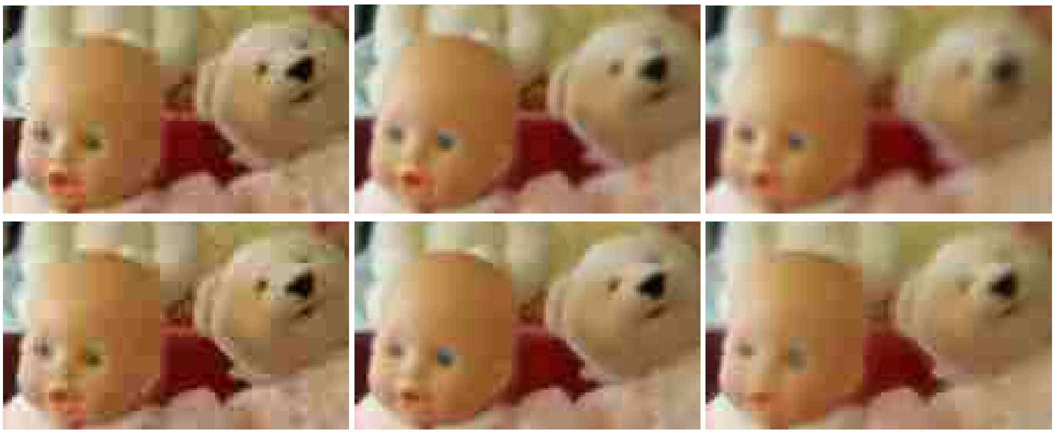


Figure 5.7: Zoomed-in portions of *Dolls*, with foveation preblurring (top) and depth-based preblurring (bottom), with (left to right) blur levels 3, 5 and 10 and respective JPEG bitrates 0.4 bpp, 0.5 bpp and 0.4 bpp.

mean score differences of 4.65 compared with 3.14 for the SSCQS tests and 1.067 compared with 1.023 for the method-of-adjustment tests (see Table 5.2). In terms of bitrate, this

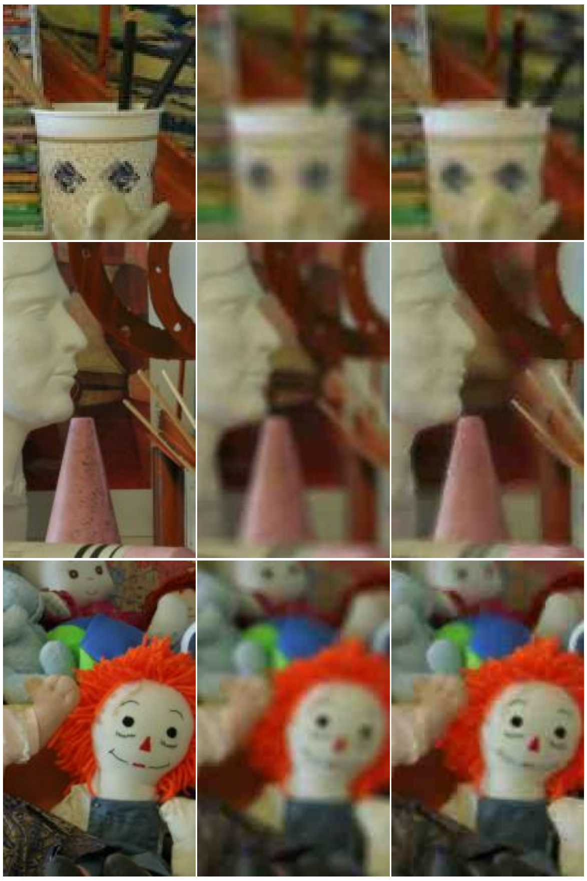


Figure 5.8: Zoomed-in portions of images. Top to bottom: "Art" 0.5 bpp, blur 5; "Art" 0.4 bpp, blur 10; "Dolls" 0.5 bpp, blur 5. Left: no blurring or encoding (i.e., the raw images). Middle: JPEG-encoded after foveation preblurring. Right: JPEG-encoded after depth-based preblurring.

means that, on average, the depth-preblurred images were as good as the 6.7%-higher bitrate foveated equivalents in the case of the Middlebury test set images, and as good as the 2.3%-higher bitrate foveated equivalents in the case of the other test images. It can also be seen from Table 5.2 that that, on average, the depth-preblurred images were as good as the 6.9%-higher bitrate unblurred JPEG equivalents in the case of the Middlebury test set images, and as good as the 5.1%-higher unblurred JPEG equivalents in the case of the other test images, whereas the corresponding figures for the foveated JPEG images were far more modest, at 0.6% and 2.6%. The very low figure of 0.6% may be due to the Middlebury test images having a greater amount of image detail in areas away from the chosen object of interest, which may have provided a greater level of distraction to cause the viewer's attention to deviate from the assumed fixation point, hence causing the viewer to be more aware of the peripheral blurring which had been applied to these images (which would not be such a problem with the more aesthetically acceptable depth blurring).

The results show a clear preference for the higher blurring level of ten pixel-widths, for which the method-of-adjustment tests gave statistically significant results for all three Middlebury test set images, and for which the SSCQS tests gave statistically significant results for all images other than *Foreman*. These test images are shown in Fig. 5.4.

Fig. 5.5 shows sample extracts from test images that exhibited significant positive results. In each case, the distribution of blurring as provided by the depth blurring achieves a more satisfying image than the foveation. For example, for *Silent*, the foveation causes an undesirable level of blurring of the necklace, whereas the depth blurring causes slightly greater blurring of the background, which is more acceptable to the viewer. Fig. 5.7 shows sample extracts from different variants of the *Dolls* image. Significant results were obtained for the images with blur levels 5 and 10, but not for blur level 3, for which the JPEG artifacts dominated over the visible differences in blurring.

A further point of note is the behaviour of the proposed algorithm in the case of small patches of missing information from the depth/disparity map. The occlusive effects of the algorithm causes all blurring in these regions to be completely contained within the regions, thus making them barely noticeable due to their small size. Examples of such missing information can be seen in Fig. 5.2, in the form of small visible patches of black (representing zero disparity, which is interpreted as maximum distance from the camera). For these patches, no artifacts are apparent in the corresponding depth-blurred images (see Fig. 5.4).

Fig. 5.8 shows further extracts from images comparing the depth-based and the foveation blurring approaches. In each case, a portion of the scene has been shown in which the same object has been clearly less blurred in the depth blurred image than in the foveated image. The existance of such regions reflect the major problem of non-gaze-contingent foveated coding in that the curiosity of each viewer is likely to distract his attention away from the assumed point (or points) of interest.

Although the compression ratios seem low, it should be noted that the nature of the blurring algorithm employed was of a square blurring technique (as for the *integral image* approach), and, having sharp boundaries in the spatial domain, its wide-tailed frequency domain characteristics would be poor for compression purposes in comparison to a smoother blurring technique (e.g. potentially taking up to 75% higher bitrate than Gaussian blurring, as suggested by the top peak in Fig. 3.23). However, the significant subjective preference for depth blurred coding over foveated coding suggests greater potential for the concept of depth blurred coding itself. If an alternative depth blurring algorithm with similar compression performance to Gaussian blurring could be employed in place of the existing algorithm, the result would be expected to produce better bitrate reduction than the levels found by past non-gaze-contingent foveated coding techniques (e.g. 35% [20]).

5.4 Conclusion

This chapter has presented an experiment which subjectively compared and quantified the difference between two space-variant still-image coding techniques: a simple depth-blurred coding technique and a simple foveated coding technique, under equivalent bitrates and blur level histograms.

The depth blurring was found to be significantly preferable to the foveation filtering for 12 out of 30 test images for at least one of two types of subjective tests; a converse preference was found for 2 out of 30 test images. The fact that the depth-blurred coding was found to be better in most of the results that were conclusive suggests that viewer awareness of blurring can be effectively offset by disguising blurring as effects which they are accustomed to in photographs. On an *equivalent JPEG quality* scale, the depth-preblurred images were as good as the 6.7%-higher bitrate foveated equivalents and as good as the 6.9%-higher bitrate unblurred JPEG equivalents in the case of test images for which a high-detail disparity map was used. The equivalent figures were 2.3% and 5.1% respectively in the case of test images with manually-generated two-level depth maps.

Chapter 6

Conclusion

6.1 Summary of Achievements

This thesis has addressed two areas of the problem of how two apply a space-variant resolution reduction for lossy coding purposes, employing the principle of local degradation for global improvement. Firstly, for foveated coding, which has in the past been dominated by the single-viewer, *gaze-contingent* scenario, this thesis has provided a missing piece for use in the multi-viewer and infinite-viewer (probability-based) scenarios which are more relevent to everyday coding scenarios. Namely, an algorithm has been proposed and demonstrated herein for computing an additive multi-viewer sensitivity function based on the Geisler & Perry contrast threshold formula, and, from this, a cut-off frequency (blur) map that is optimal in the sense of discarding local frequency components in least-noticeable-first order. The advantages of this approach have been argued from a purely logical perspective, with subjective justification as future work. Furthermore, a novel algorithm has been presented for applying the blur map with high-accuracy Gaussian blurring in a computationally efficient manner. Experimental results demonstrated the proposed Gaussian blurring algorithm provides typically 10 to 15 dB better approximation of perfect Gaussian blurring than blended Gaussian Pyramid blurring, which was only 2 dB better than *integral image* square blurring. Therefore, in scenarios where high-accuracy space-variant Gaussian blurring is desired, the proposed approach is the best choice; and, due to the broader spectral tails of Gaussian Pyramid blurring, the proposed approach is expected to have superior compression when used for preprocessing with an ordinary encoder.

Secondly, this thesis has investigated a relatively untapped field that has possibilities in the realm of image and video coding – namely, the reduction of the depth of field of an image, as is performed in photography – and thereby proposing a new research area of depth-blurred coding. For use in depth-blurred coding, two novel selective blurring algorithms have been presented that mimic the optical depth of field blurring that occur naturally in cameras. The proposed algorithms both provide a realistic simulation of depth blurring, with the desirable properties of aiming to mimic occlusion effects as occur in natural blurring, and of being able to handle any number of blurring and occlusion levels with the same order of computational complexity. Subjective experiments have been reported to compare the perceived quality of a rudimentary foveated image coding technique with a depth-based preblurring technique under equivalent conditions. Moreover, a method-of-adjustment approach has been presented for measuring overall perceived image quality in terms of *equivalent JPEG*. The depth-based blurring was found to be significantly preferable to the foveated blurring for 12 out of 30 test images for at least one of two types of subjective tests; a converse preference was found for only 2 out of 30 test images. Therefore, the results showed that depth-based preblurring is generally better in terms of the level of perceived quality than the equivalent foveated preblurring; the depth-preblurred images were as good as the 6.7%-higher bitrate foveated equivalents and as good as the 6.9%-higher bitrate unblurred JPEG equivalents in the case of test images for which a high-precision disparity map was used, and as good as the 2.3%-higher bitrate foveated equivalents and the 5.1%-higher bitrate unblurred JPEG equivalents in the case of images with manually-generated two-level depth maps.

The suggested compression improvement of 5-7% (when compared to ordinary unblurred encoding) is a small benefit on its own, but certain applications, such as the storage of images on a memory card or hard disk, can benefit directly from a 5-7% improvement; for instance, a 2GB memory card storing 5 megapixel, 1.5MB JPEG images would be able to store in the region of 30 to 50 extra images. If the same level of compression could be attained when applying the proposed depth blurring as preprocessing for an off-the-shelf MPEG-2 video encoder, this 5-7% improvement would enable 6-8 minutes of extra storage at the same quality on a 120 minute DVD. Moreover, this modest 5-7% figure can be attributed to the low-performance square blurring employed, and the key point is that depth blurred coding concept was found to be generally superior to non-gaze-contingent foveated coding in equivalent conditions; therefore, if an alternative depth blurring algorithm with similar compression performance to Gaussian blurring could be employed in place of the existing algorithm, the result may be expected to produce better bitrate reduction than the levels found by past non-gaze-contingent foveated coding techniques (e.g. 35% [20]).

The general unavailability of exact fixation point knowledge and the difficulty in predicting human fixation pose a great challenge against the drive to exploit the space-variant nature of the human visual system in image and video coding. However, the drive continues due to the expected benefits. With such a disparity between its expected potential and actual achievements, space-variant coding can, even after more than fifteen years of research, be seen as an area of research in its infancy.

6.2 Future Work

A number of questions arise which present avenues for the extension of this research:

In multifoveation, is *average* better than *maximum*? It is necessary to provide subjective evidence in favour of or against the hypothesis presented in section 3.2 that an additive (or mean average) model of collective viewer sensitivity provides a preferable space-variant resolution map compared to the established approach of addressing only the viewer of the nearest fixation point matters at each image location (i.e. taking the maximum of viewer sensitivities).

Extensions to Video *How well does depth blurred coding work for video?* The approach and experiment of chapters 4 and 5 can be adapted to video and investigated; a possible scenario for a complete depth-blurred video coding system to investigate could be a teleconferencing system which uses the disparity map input from a stereovision technique. Additionally, it would be beneficial to develop the proposed approach to work with circular point spread functions (rather than performing square blurring) and to cater for partial occlusion, by applying a gradual occlusion of the blur of far objects when the boundaries of nearer objects are themselves blurred, as in genuine blur effects.

How should an overall blur level be chosen for a given bit rate? The experiment in chapter 5 investigated a number of blur levels and bit rates; however, a practical encoder (whether for depth-blurred coding or foveated coding) would need to choose a single blur level automatically, such as from a fixed, pre-trained look-up table based on subjectively determined preferences. This applies both to depth-blurred coding and to foveated coding.

How should foveation be applied with a motion-compensated-prediction video format? Applying single- or multi-viewer foveation to a video sequence has an intrinsic problem with formats such as MPEG-2 and H.264 in that the dynamic resolution maps that are applied may result in a bitrate increase rather than a reduction, because, for example, in a scene dominated by a static background, the normally low-bitrate P frames will increase in bitrate if the dynamic resolution maps introduce changes to the background blocks. Therefore simple preblurring becomes unsuitable; instead, internal alterations to the encoder will be necessary, as well as the introduction of a temporal element to the space-variance such that the changes in spatial priority do not themselves cause changes in otherwise unchanged parts of the scene.

Bibliography

[1] T. Popkin, A. Cavallaro and D. Hands, "Multi-Foveation Filtering," in *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, Apr. 2009, pp. 669–672, ISBN 978-1-4244-2353-8, DOI 10.1109/ICASSP.2009.4959672.

[2] T. Popkin, A. Cavallaro and D. Hands, "Accurate and Efficient Method for Smoothly Space-Variant Gaussian Blurring," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1362 – 1370, May 2010, ISSN 1057-7149, DOI 10.1109/TIP.2010.2041400.

[3] T. Popkin, A. Cavallaro and D. Hands, "Distance Blurring for Space-Variant Image Coding," in *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, Apr. 2009, pp. 665–668, ISBN 978-1-4244-2353-8, DOI 10.1109/ICASSP .2009.4959671.

[4] Z. Wang and A. C. Bovik, "Foveated image and video coding," in *Digital Video, Image Quality and Perceptual Coding*, H. R. Wu and K. R. Rao, Eds. CRC Press, 2006, ch. 14, pp. 431–457, ISBN 0-8247-2777-0.

[5] L. Lu, Z. Wang and A. C. Bovik, "Scalable foveated visual information coding and communications," 2002, http://live.ece.utexas.edu/publications/2002/zw_icccas_2002_scalvideo.pdf (accessed May 20th, 2008).

[6] J. I. Khan and O. Komogortsev, "Dynamic gaze span window based foveation for perceptual media streaming," Kent State University, Ohio, USA, Medianet Lab Technical Report TR2002-11-01, Nov. 2002, http://www.medianet.kent.edu/techreports/TR-2002-11-01-focus-KK.pdf (accessed May 20th, 2008).

[7] J. I. Khan and O. Komogortsev, "A hybrid scheme for perceptual object window design with joint scene analysis and eye-gaze tracking for media encoding based on perceptual attention," *J. Electronic Imaging*, vol. 15, no. 2, Apr. 2006.

[8] Z. Wang, A. C. Bovik and L. Lu, "Wavelet-based foveated image quality measurement for region of interest image coding," in *Proc. International Conf. on Image Processing*, vol. 2, Thessaloniki, Greece, 2001, pp. 89–92.

[9] Z. Wang and A. C. Bovik, "Embedded foveation image coding," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1397–1410, Oct. 2001.

[10] P. Kortum and W. Geisler, "Implementation of a foveated image coding system for image bandwidth reduction," *Proc. SPIE, Vol. 2657*, pp. 350–360, Apr. 1996.

[11] M. Nyström, M. Novak and K. Holmqvist, "A novel approach to image coding using off-line foveation controlled by multiple eye-tracking measurement," in *24th Picture Coding Symposium*, San Francisco, CA, USA, Dec. 2004, http://www.ece.ucdavis.edu/PCS2004/pdf/ID5_offliineFoveation.pdf (accessed May 20th, 2008).

[12] I. van der Linde, "Multi-resolution image compression using image foveation and simulated depth of field for stereoscopic displays," in *Proc. SPIE*, vol. 5291, 2004, pp. 71–80.

[13] A. P. Bradley, "Can region of interest coding improve overall perceived image quality?" in *Proc. APRS Workshop on Digital Image Computing*, Brisbane, Australia, Feb. 2003, pp. 41–44.

[14] A. P. Bradley and F. W. M. Stentiford, "Visual attention for region of interest coding in JPEG 2000," *J. Visual Communication and Image Representation*, vol. 14, no. 3, pp. 232–250, Aug. 2003.

[15] A. P. Bradley and F. W. M. Stentiford, "JPEG 2000 and Region of Interest Coding," *Digital Image Computing Techniques and Applications (DICTA), Melbourne, Australia*, pp. 303–308, 2002.

[16] V. Sanchez, A. Basu and M. K. Mandal, “Prioritized region of interest coding in JPEG 2000,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 9, pp. 1149–1155, Sep. 2004, ISSN 1558-2205.

[17] A. Ebrahimi-Moghadam and S. Shirani, “Matching pursuit-based region-of-interest image coding,” *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 406–415, Feb. 2007, ISSN 1057-7149.

[18] A. Nguyen, V. Chandran and S. Sridharan, “Gaze tracking for region of interest coding in JPEG 2000,” *Signal Processing: Image Communication*, vol. 21, no. 5, pp. 359–377, Jun. 2006, ISSN 0923-5965.

[19] A. Nguyen, V. Chandran and S. Sridharan, “Gaze-J2K: gaze-influenced image coding using eye trackers and JPEG 2000,” *Journal of Telecommunications and Information Technology*, 2006, http://www.nit.eu/czasopisma/JTIT/2006/1/3.pdf (accessed May 5th, 2010).

[20] S. Liu and A. C. Bovik, “Foveation embedded DCT domain video transcoding,” *J. Visual Communication and Image Representation*, vol. 16, no. 6, pp. 643–667, Dec. 2005.

[21] Z. Wang, L. Lu and A. C. Bovik, “Foveation scalable video coding with automatic fixation selection,” *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 243–254, Feb. 2003.

[22] Z. Wang, L. Lu and A. C. Bovik, “Rate scalable video coding using a foveation-based human visual system model,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, Salt Lake City, USA, May 2001, pp. 1785–1788.

[23] H. R. Sheikh, S. Liu, B. L. Evans and A. C. Bovik, “Real-time foveation techniques for H.263 video encoding in software,” in *Proc. IEEE ICASSP ’01*, vol. 3, Salt Lake City, USA, May 2001, pp. 1781–1784.

[24] H. R. Sheikh, B. L. Evans and A. C. Bovik, “Real-time foveation techniques for low bit rate video coding,” *Real-Time Imaging*, vol. 9, no. 1, pp. 27–40, Feb. 2003.

[25] S. Lee, M. S. Pattichis and A. C. Bovik, “Foveated video compression with optimal rate control,” *IEEE Trans. Image Process.*, vol. 10, no. 7, pp. 911–992, Jul. 2001.

[26] S. Lee and A. C. Bovik, “Fast algorithms for foveated video processing,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 2, pp. 149–162, Feb. 2003.

[27] O. Komogortsev and J. I. Khan, “Predictive perceptual compression for real time video communication,” *Proc. 12th Annual ACM International Conference on Multimedia*, pp. 220–227, 2004.

[28] W. S. Geisler and J. S. Perry, “A real-time foveated multiresolution system for low-bandwidth video communication,” *Proc. SPIE, Vol. 3299*, pp. 294–305, Jul. 1998.

[29] L. Itti, “Automatic foveation for video compression using a neurobiological model of visual attention,” *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1304–1318, Oct. 2004.

[30] N. Dhavale and L. Itti, “Saliency-based multifoveated MPEG compression,” in *Proc. 7th International Symposium on Signal Processing and Its Applications*, vol. 1, Jul. 2003, pp. 229–232.

[31] N. Tsapatsoulis, K. Rapantzikos and Y. Avrithis, “Priority coding for video-telephony applications based on visual attention,” in *Proc. 2nd International Mobile Multimedia Communications Conference*, vol. 324, Alghero, Sardinia, Italy, Sep. 2006, article No. 31.

[32] M. M. Farid, F. Kurugollu and F. D. Murtagh, “Adaptive wavelet eye-gaze based video compression,” in *Proc. SPIE, Vol. 4877*, Mar. 2003, pp. 255–263.

[33] C.-C. Ho and J.-L. Wu, “A foveation-based rate shaping mechanism for MPEG videos,” in *Proc. 3rd IEEE Pacific Rim Conference on Multimedia (Vol. 2532)*. Springer-Verlag, 2002, pp. 485–492.

[34] C.-C. Ho and J.-L. Wu, “Toward User Oriented Scalable Video by Using Foveated FGS Bitstreams,” in *Proc. IEEE International Conference on Consumer Electronics (ICCE)*, Jun. 2003, pp. 46–47, ISBN 0-7803-7721-4.

[35] C.-C. Ho, J.-L. Wu and W.-H. Cheng, “A practical foveation-based rate-shaping mechanism for MPEG videos,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1365–1372, Nov. 2005.

[36] S. J. Daly, K. E. Matthews and J. Ribas-Corbera, “Visual eccentricity models in face-based video compression,” in *Proc. SPIE, Vol. 3644 (Human Vision and Electronic Imaging IV)*, May 1999, pp. 152–166.

[37] N. Doulamis, A. Doulamis, D. Kalogeras and S. Kollias, “Low bit-rate coding of image sequences using adaptive regions of interest,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 928–934, Dec. 1998.

[38] D. Agrafiotis, S. J. C. Davies, N. Canagarajah and D. R. Bull, “Towards efficient context-specific video coding based on gaze-tracking analysis,” *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 3, no. 4, Dec. 2007, article No. 4.

[39] P. L. Silsbee, A. C. Bovik and D. Chen, “Visual pattern image sequence coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 4, pp. 291–301, Aug. 1993.

[40] D. Chai, K. N. Ngan and A. Bouzerdoum, “Foreground/background bit allocation for region-of-interest coding,” in *Proc. International Conference on Image Processing*, vol. 2, 2000, pp. 923–926.

[41] A. Cavallaro, O. Steiger and T. Ebrahimi, “Semantic segmentation and description for video transcoding,” *Proc. 2003 International Conference on Multimedia and Expo*, vol. 3, no. 11, pp. III–597 to III–600, Jul. 2003, ISBN 0-7803-7965-9.

[42] A. Cavallaro, O. Steiger and T. Ebrahimi, “Perceptual pre-filtering for video coding,” *Proc. IEEE International Symposium on Intelligent Multimedia, Video and Speech Processing*, Oct. 2004.

[43] Ç. Dikici, H. Isil Bozma and M. R. Civanlar, “Fovea based coding for video streaming,” in *Proc. International Conf. Image Analysis and Recognition*, vol. 3211, Sep. 2004, pp. 285–294.

[44] L. S. Karlsson, M. Sjöström and R. Olsson, “Spatio-temporal filter for roi video coding,” in *Proc. 14th European Signal Processing Conference (EUSIPCO)*, Florence, Italy, Sep. 2006.

[45] C. W. Tang, “Spatiotemporal visual considerations for video coding,” *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 231–238, 2007.

[46] G. J. Sullivan, J.-R. Ohm, A. Ortega, E. Delp, A. Vetro and M. Barni, “Future of Video Coding and Transmission,” *IEEE Signal Processing Magazine*, vol. 23, no. 6, Nov. 2006.

[47] B. A. Wandell, *Foundations of Vision.* Sinauer Associates, 1995, ISBN 0-87893-853-2.

[48] R. S. Wallace, P.-W. Ong, B. B. Bederson and E. L. Schwartz, “Space-variant image processing,” *International J. Computer Vision*, vol. 13, no. 1, pp. 71–90, Sep. 1994.

[49] C. G. Ho, R. C. D. Young and C. R. Chatwin, “Sensor geometry and sampling methods for space-variant image processing,” *Pattern analysis and applications*, vol. 5, no. 4, pp. 369–384, 2002, ISSN 1433-7541.

[50] A. L. Yarbus, *Eye movements and vision.* New York, USA: Plenum Press, 1967.

[51] A. Çöltekin, “Foveation for 3D Visualization and Stereo Imaging,” Ph.D. dissertation, Helsinki University of Technology, Finland, 2006, doctoral thesis, ISBN 951-22-8016-7.

[52] L. C. Loschky and G. W. McConkie, “User performance with gaze contingent multiresolutional displays,” in *Proc. Eye Tracking Research & Applications Symposium*, Palm Beach Gardens, FL, USA, 2000, pp. 97–103, ISBN 1-58113-280-8.

[53] P. Baudisch, D. DeCarlo, A. Duchowski and W. Geisler, “Focusing on the essential: Considering attention in display design,” *Communications of the ACM*, vol. 46, no. 3, pp. 60–66, Mar. 2003.

[54] F. Stentiford, “An estimator for visual attention through competitive novelty with application to image compression,” in *Proc. Picture Coding Symposium*, Seoul, South Korea, Apr. 2001, pp. 25–27.

[55] F. W. M. Stentiford, “An evolutionary programming approach to the simulation of visual attention,” in *Proc. Congress on Evolutionary Computation*, vol. 2, Seoul, South Korea, May 2001, pp. 851–858, ISBN 0-7803-6657-3.

[56] A. Cavallaro, O. Steiger and T. Ebrahimi, “Semantic video analysis for adaptive content delivery and automatic description,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 1200–1209, Oct. 2005.

[57] Z. Chen, J. Han and K. N. Ngan, “Dynamic bit allocation for multiple video object coding,” *IEEE Trans. Multimedia*, vol. 8, no. 6, pp. 1117–1124, Dec. 2006.

[58] C.-C. Ho, W.-H. Cheng, T.-J. Pan and J.-L. Wu, “A user-attention based focus detection framework and its applications,” in *Proc. 4th Pacific Rim Conference on Multimedia*, vol. 3. Springer-Verlag, Dec. 2003, pp. 1315–1319.

[59] L. Itti, C. Koch and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[60] L. Itti, "Models of bottom-up and top-down visual attention," Ph.D. dissertation, California Institute of Technology, Pasadena, CA, USA, 2000, doctoral thesis, ISBN 0-599-77919-5.

[61] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews: Neuroscience*, vol. 2, no. 3, pp. 194–203, Mar. 2001, ISSN 1471-0048.

[62] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision Research*, vol. 40, pp. 1489–1506, 2000.

[63] N. Tsapatsoulis, Y. Avrithis and S. Kollias, "Facial image indexing in multimedia databases," *Pattern Analysis & Applications*, vol. 4, no. 2-3, pp. 93–107, Jun. 2001, ISSN 1433-7541.

[64] J. You, G. Liu and H. Li, "A novel attention model and its application in video analysis," *Applied mathematics and computation*, vol. 185, no. 2, pp. 963–975, 2007, ISSN 0096-3003.

[65] V. Navalpakkam and L. Itti, "An integrated model of top-down and bottom-up attention for optimizing detection speed," *Proc. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 06)*, 2006.

[66] D. Parkhurst, K. Law and E. Niebur, "Modeling the role of salience in the allocation of overt visual attention." *Vision Research*, vol. 42, pp. 107–123, 2002.

[67] R. Cucchiara, C. Grana and A. Prati, "A framework for semantic video transcoding," *Atti del Workshop "Percezione e Visione delle Macchine", Siena*, Sep. 2002.

[68] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human Neurobiology*, vol. 4, no. 4, pp. 219–227, 1985, ISSN 0721-9075.

[69] O. Le Meur, P. Le Callet, D. Barba and D. Thoreau, "A coherent computational approach to model bottom-up visual attention," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 802–817, May 2006.

[70] H. Wolf and D. Deng, "Image saliency mapping and ranking using an extensible visual attention model based on MPEG-7 feature descriptors," *The Information Science Discussion Paper Series*, no. 2005/10, Dec. 2005, ISSN 1172-6024.

[71] Y. Sun and R. Fisher, "Hierarchical selectivity for object-based visual attention," *Lecture Notes in Computer Science*, vol. 2525, pp. 427–438, 2002, ISSN 0302-9743.

[72] E. Maggio, E. Piccardo, C. Regazzoni and A. Cavallaro, "Particle PHD filtering for multi-target visual tracking," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 1, Honolulu, Hawaii, Apr. 2007, pp. I–1101 to I–1104, ISBN 1-4244-0728-1.

[73] R. Kasturi, *Performance evaluation protocol for face, person and vehicle detection & tracking in video analysis and content extraction (VACE-II)*, Computer Science & Engineering, University of South Florida, Tampa, Jan. 2006.

[74] J. M. Henderson, J. R. Brockmole, M. S. Castelhano and M. Mack, "Visual saliency does not account for eye movements during visual search in real-world scenes," in *Eye movements: A window on mind and brain*, R. V. Gompel, M. Fischer, W. Murray, and R. Hill, Eds. Elsevier, 2007, ch. 25, pp. 537–562, ISBN 0080449808.

[75] R. L. Fantz, "The origin of form perception," *Scientific American*, vol. 204, pp. 66–72, 1961.

[76] H. Wang and S.-F. Chang, "A highly efficient system for automatic face region detection in MPEG video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 4, pp. 615–628, Aug. 1997, ISSN 1051-8215.

[77] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23–38, Jan. 1998.

[78] J. G. Robson and N. Graham, "Probability summation and regional variation in contrast sensitivity across the visual field," *Vision Research*, vol. 21, pp. 409–418, 1981.

[79] M. S. Banks, A. B. Sekular and S. J. Anderson, "Peripheral spatial vision: limits imposed by optics, photoreceptors and receptor pooling," *J. Optical Society of America A*, vol. 8, no. 11, pp. 1775–1787, Nov. 1991.

[80] T. L. Arnow and W. S. Geisler, "Visual detection following retinal damage: Predictions of an inhomogeneous retino-cortical model," in *Proc. SPIE, Vol. 2674 (Laser-Inflicted Eye Injuries: Epidemiology, Prevention, and Treatment)*, 1996, pp. 119–130.

[81] H. Davson, *Physiology of the eye*, 5th ed. Basingstoke, UK: Macmillan, 1990, ISBN 0-333-45860-5.

[82] S. M. Anstis, "A chart demonstrating variations in acuity with retinal position," *Vision Research*, vol. 14, no. 7, pp. 589–592, 1974.

[83] C. Ware, *Information Visualization – Perception for Design*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann, 2004, ISBN 1-55860-819-2.

[84] D. H. Kelly, "Spatial Frequency Selectivity in the Retina," *Vision Research*, vol. 15, no. 6, pp. 665–672, Jun. 1975.

[85] M. Reddy, "Perceptually modulated level of detail for virtual environments," Ph.D. dissertation, Univ. Edinburgh, UK, 1997.

[86] L. Lu, Z. Wang and A. C. Bovik, "Adaptive frame prediction for foveation scalable video coding," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, Aug. 2001, pp. 705–708, ISBN 0-7695-1198-8.

[87] E. Peli, J. Yang and R. B. Goldstein, "Image invariance with changes in size: the rôle of peripheral contrast thresholds," *J. Optical Society of America A*, vol. 8, no. 11, pp. 1762–74, Nov. 1991.

[88] A. T. Duchowski, "Acuity-matching resolution degradation through wavelet coefficient scaling," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1437–1440, Aug. 2000.

[89] D. H. Foster, S. Gravano and A. Tomoszek, "Acuity for fine-grain motion and for two-dot spacing as a function of retinal eccentricity: differences in specialization of the central and peripheral retina," *Vision Research*, vol. 29, no. 8, pp. 1017–1031, 1989.

[90] BS EN ISO/IEC 11172-2:1995, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 2: Video*, ISO/IEC Std., the MPEG-1 video standard, ISBN 0-580-22593-3.

[91] BS ISO/IEC 13818-2:1996, *Information technology - Generic coding of moving pictures and associated audio information: Part 2: Video*, ISO/IEC Std., the MPEG-2 video standard, ISBN 0-580-27229-X.

[92] E.-C. Chang, S. Mallat and C. Yap, "Wavelet foveation," *Applied and Computational Harmonic Analysis*, vol. 9, no. 3, pp. 312–335, Oct. 2000, ISSN 1063-5203.

[93] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Jun. 1996, ISSN 1051-8215.

[94] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[95] ISO/IEC 14496-2:2004, *Information technology – Coding of audio-visual objects – Part 2: Visual*, ISO/IEC Std., the MPEG-4 video standard.

[96] F. C. Crow, "Summed-area tables for texture mapping," in *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques.* New York, USA: ACM Press, 1984, pp. 207–212.

[97] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.

[98] J. G. Proakis and D. G. Manolakis, *Digital signal processing : principles, algorithms and applications*, 2nd ed. Macmillan, 1992, ISBN 0-02-396815-X.

[99] S. Tan, J. L. Dale and A. Johnston, "Performance of three recursive algorithms for fast space-variant Gaussian filtering," *J. Real-Time Imaging*, vol. 9, pp. 215–228, 2003.

[100] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge University Press, 1992, ISBN 0-521-43108-5.

[101] V. K. Madisetti and D. B. Williams, *The Digital Signal Processing Handbook.* CRC Press, 1998, ISBN 0-8493-8572-5.

[102] J. S. Perry and W. S. Geisler, "Gaze-contingent real-time simulation of arbitrary visual fields," in *Proc. SPIE*, vol. 4662, Jun. 2002, pp. 57–69.

[103] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Communications*, vol. COM-31, no. 4, pp. 532–540, Apr. 1983.

[104] S. Lee, G. J. Kim and S. Choi, "Real-Time Depth-of-Field Rendering Using Anisotropically Filtered Mipmap Interpolation," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 3, pp. 453–464, 2009.

[105] R. L. Cook, T. Porter and L. Carpenter, "Distributed ray tracing," in *SIGGRAPH '84: Proc. of the 11th Computer Graphics and Interactive Techniques*, vol. 18, no. 3, Jul. 1984, pp. 137–145.

[106] P. Rokita, "Fast Generation of Depth of Field Effects in Computer Graphics," *Computers and Graphics*, vol. 17, no. 5, pp. 593–595, Sep. 1993, ISSN 0097-8493.

[107] J. D. Mulder and R. van Liere, "Fast perception-based depth of field rendering," in *VRST '00: Proc. of the ACM Symposium on Virtual Reality Software and Technology*, 2000, pp. 129–133.

[108] J. Hammon, "Practical post-process depth of field," in *GPU Gems 3*, H. Nguyen, Ed. Addison-Wesley, Aug. 2007, ch. 28, pp. 583–606, ISBN 0321515269.

[109] J. Krivanek, J. Zara and K. Bouatouch, "Fast depth of field rendering with surface splatting," in *Proc. Computer Graphics International*, Los Alamitos, 2003, pp. 196–201.

[110] M. Kass, A. Lefohn and J. Owens, "Interactive Depth of Field Using Simulated Diffusion on a GPU," 2006, http://graphics.pixar.com/library/DepthOfField/paper.pdf (accessed Jul 17th, 2009).

[111] B. A. Barsky, M. J. Tobias, D. P. Chu and D. R. Horn, "Elimination of artifacts due to occlusion and discretization problems in image space blurring techniques," *Graph. Models*, vol. 67, no. 6, pp. 584–599, 2005.

[112] L. Blonde, T. Viellard and D. Sahuc, "Method of generating blur," European Patent EP1 494 174, Jan., 2005.

[113] M. Kraus and M. Strengert, "Depth-of-field rendering by pyramidal image processing," *Eurographics*, vol. 26, no. 3, pp. 645–654, Sep. 2007.

[114] S. Lee, G. J. Kim and S. Choi, "Real-time depth-of-field rendering using point splatting on per-pixel layers," *Comput. Graph. Forum*, vol. 27, no. 7, pp. 1955–1962, 2008.

[115] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error measurement to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[116] S. Winkler, "Perceptual video quality metrics – a review," in *Digital Video, Image Quality and Perceptual Coding*, H. R. Wu and K. R. Rao, Eds. CRC Press, 2006, ch. 5, pp. 155–179, ISBN 0-8247-2777-0.

[117] H. R. Wu and K. R. Rao, Eds., *Digital Video, Image Quality and Perceptual Coding.* CRC Press, 2006, ISBN 0-8247-2777-0.

[118] ITU-T Rec. P.910, *Subjective video quality assessment methods for multimedia applications*, International Telecommunication Union Std., Rev. 09/99, 1999.

[119] ITU-R Rec. BT.500-11, *Methodology for the subjective assessment of the quality of television pictures*, International Telecommunication Union Std., 2002.

[120] J. J. Hwang, H. R. Wu and K. R. Rao, "Digital picture compression and coding structure," in *Digital Video, Image Quality and Perceptual Coding*, H. R. Wu and K. R. Rao, Eds. CRC Press, 2006, ch. 1, pp. 3–43, ISBN 0-8247-2777-0.

[121] Z. Wang, A. C. Bovik and H. R. Sheikh, "Structural similarity based image quality assessment," in *Digital Video, Image Quality and Perceptual Coding*, H. R. Wu and K. R. Rao, Eds. CRC Press, 2006, ch. 7, pp. 225–241, ISBN 0-8247-2777-0.

[122] H. R. Sheikh, Z. Wang, A. C. Bovik and L. K. Cormack, "Image and video quality assessment research at live," http://live.ece.utexas.edu/research/quality/ (accessed Sep 23rd, 2008).

[123] J. E. Caviedes and F. Oberti, "No-reference quality metric for degraded and enhanced video," in *Digital Video, Image Quality and Perceptual Coding*, H. R. Wu and K. R. Rao, Eds. CRC Press, 2006, ch. 10, pp. 305–324, ISBN 0-8247-2777-0.

[124] Z. Wang, A. C. Bovik and B. L. Evans, "Blind measurement of blocking artifacts in images," in *Proc. IEEE Int. Conf. Image Proc.*, Vancouver, Canada., 2000, pp. 981–984.

[125] H. R. Wu and M. Yuen, "A generalized block-edge impairment metric for video coding," *IEEE Signal Process. Lett.*, vol. 4, no. 11, pp. 317–320, Nov. 1997.

[126] P. Marziliano, F. Dufaux, S. Winkler and T. Ebrahimi, "A no-reference perceptual blur metric," in *Proc. ICIP*, vol. 3, Rochester, NY, USA, Sep. 2002, pp. 57–60.

[127] P. Marziliano, F. Dufaux, S. Winkler and T. Ebrahimi, "Perceptual blur and ringing metrics: application to JPEG 2000," *Signal Processing: Image Communication*, vol. 19, no. 1, Jan. 2004.

[128] P. Gastaldo, R. Zunino and S. Rovetta, "Objective assessment of MPEG-2 video quality," *J. Electronic Imaging*, vol. 11, no. 3, pp. 365–374, Jul. 2002.

[129] P. Corriveau and A. Webster, "Final Report From The Video Quality Experts Group On The Validation Of Objective Models Of Video Quality Assessment, Phase II," ITU VQEG, Tech. Rep., Aug. 2003.

[130] J. G. Snodgrass, "Psychophysics," in *Experimental Sensory Psychology*, B. Scharf, Ed. Scott, Foresman and Company, 1975, ch. 2, pp. 17–67, ISBN 0-673-05428-4.

[131] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in Pascal.* Cambridge University Press, 1989, ISBN 0-521-37516-9.

[132] E. Maggio, M. Taj and A. Cavallaro, "Efficient Multi-target Visual Tracking Using Random Finite Sets," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1016–1027, Aug. 2008, ISSN 1051-8215.

[133] S. A. Martucci, "Symmetric convolution and the discrete sine and cosine transforms," *IEEE Trans. Signal Process.*, vol. 42, no. 5, pp. 1038–1051, May 1994.

[134] Z. Wang, "Fast Algorithms for the Discrete W Transform and for the Discrete Fourier Transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 4, pp. 1038–1051, Aug. 1984.

[135] "The USC-SIPI Image Database," http://sipi.usc.edu/database/ (accessed Dec 18, 2008).

[136] Site http://www.hlevkin.com/TestImages/classic.htm (accessed Dec 18, 2008).

[137] "Kodak Lossless True Color Image Suite," http://r0k.us/graphics/kodak/ (accessed Dec 18, 2008).

[138] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, Madison, WI, USA, Jun. 2003, pp. 195–202.

[139] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8.

[140] S. Hsu, S. Acharya, A. Rafii and R. New, "Performance of a Time-of-Flight Range Camera for Intelligent Vehicle Safety Applications," in *Advanced Microsystems for Automotive Applications 2006*, J. Valldorf and W. Gessner, Eds. Springer Berlin Heidelberg, 2006, pp. 205–219, ISBN 978-3-540-33409-5.

[141] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International J. Computer Vision*, vol. 47, no. 1-3, pp. 7–42, Apr-Jun 2002.

[142] A. Saxena, M. Sun and A. Y. Ng, "Make3D: Learning 3D Scene Structure from a Single Still Image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.

[143] G. Zhang, J. Jia, T.-T. Wong and H. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 974–988, Jun. 2009.

[144] W. N. Klarquist, W. S. Geisler and A. C. Bovik, "Maximum-likelihood depth-from-defocus for active vision," in *IROS '95: Proc. International Conf. Intelligent Robots and Systems*, vol. 3. Washington, DC, USA: IEEE Computer Society, Aug. 1995, pp. 374–379.

[145] E.-C. Chang and C. K. Yap, "A wavelet approach to foveating images," in *Proc. of the 13th Symposium on Computational Geometry.* ACM, 1997, pp. 397–399.

ERRATA OF THESIS

Space-Variant Picture Coding by T. J. Popkin

May 20th, 2010

p. 95, fig. 4.9: **listcorners:**

line 3: for $\mathbf{z} \in \{\, \mathbf{c}^{11}(\mathbf{x}, b),\ \mathbf{c}^{12}(\mathbf{x}, b),\ \mathbf{c}^{21}(\mathbf{x}, b),\ \mathbf{c}^{22}(\mathbf{x}, b) \,\}$ *read* $(v', \mathbf{z}) \in \{\ (v, \mathbf{c}^{11}(\mathbf{x}, b)),\ (\text{-}v, \mathbf{c}^{12}(\mathbf{x}, b)),\ (\text{-}v, \mathbf{c}^{21}(\mathbf{x}, b)),\ (v, \mathbf{c}^{22}(\mathbf{x}, b))\ \}$

line 5: for (v, ω) *read* (v', ω)

extractsum:

line 5: for $\lceil \frac{y'}{2} \rceil$ *read* y'

lines 8-9: between these lines, insert new line: Set $P(x, y) = s$.

p. 96, fig. 4.10: **treeget:**

line 1: insert two preceding lines: **if** T is null **then** *and* Set $s = 0$.

line 1: for **if** $\omega{<}\mathrm{min}(R_T)$ *read* **else if** $\omega{>}\mathrm{max}(R_T)$

line 3: for $\omega{\geq}\mathrm{max}(R_T)$ *read* $\omega{\leq}\mathrm{min}(R_T)$

p. 97, fig. 4.11: *for bottom occurrence of* Adding $(100010,{}^{-}11)$ *read* Adding $(010010,{}^{+}76)$