Visual Attention Using 2D & 3D Displays

by

Zbigniew Zdziarski, BA, BCompSc (Hons)

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Doctor of Philosophy

University of Dublin, Trinity College

July 2015

Declaration

I, the undersigned, declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

Zbigniew Zdziarski

July 23, 2015

Permission to Lend and/or Copy

I, the undersigned, agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Zbigniew Zdziarski

July 23, 2015

Acknowledgments

Firstly I would like to thank Dr Rozenn Dahyot without whom I would never have submitted this thesis. I was very lucky to have her as a supervisor and will undoubtedly be recommending her to anyone wishing to do any research in computer vision.

Secondly I would like to thank everyone in the GV2 group for the laughs and good times. I would especially like to thank Dr Claudia Arellano next to whom I sat in the lab for over three years. We went through thick and thin together!

Next I would like to thank Kieran O'Reilly, frontman of the band White McKenzie that I was fortunate enough to join on my arrival to Ireland. Kieran's always been there for me - a true mate. I will never forget you and hope that our friendship will continue to develop even if greater distances will separate us.

Lastly I would like to thank all my other friends that made living away from Australia (the place I will always call home) that little bit more bearable. People like Ryan Connolly, Maria Salisbury, Br Conor McDonough, Br Michael O Dubhghaill, and Dr Alvaro Paul deserve a special mention here. Cheers, guys.

Zbigniew Zdziarski

University of Dublin, Trinity College
July 2015

Abstract

In the past three decades, robotists and computer vision scientists, inspired by psychological and neurophysiological studies, have developed many computational models of attentions (CMAs) that mimic the behaviour of the human visual system in order to predict where humans will focus their attention. Most of CMA research has been focussing on the visual perception of images and videos displayed on 2D screens. There has recently, however, been a surge in devices that can display media in 3D and CMAs in this domain are becoming increasingly important. Research in this context is minimal, however. This thesis attempts to alleviate this problem. We explore the Graph-Based Visual Saliency algorithm [68] and extend it into 3D by developing a new depth incorporation method. We also propose a new online eye tracker calibration procedure that is more accurate and faster than standard processes and is also able to give confidence values associated with each eye position reading. Eye tracking data is used to evaluate CMAs. We use our novel eye tracking method to create a 2D/3D video eye tracking dataset obtained from 50 people. A statistical analysis is performed to locate where perception differs in 2D and 3D in videos. Taking advantage of the uncertainties associated with our eye tracking data, we also propose a novel Gaussian mixture model for computing eye tracking heat maps.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

We never see and analyse an entire scene completely in one go [182]. Every second approximately 10^8-10^9 bits of visual stimuli enters our eyes [76, 95, 20]. This is too much information for the human visual system to handle in real-time so it optimises the analysis of scenes by picking out important information in regions. When given a scene, we move our gaze from one important region to another - and it is during this time that the brain builds its representation.

This optimising action of ours, called *selective attention*, is the subject of research in visual attention (VA). Inspired by psychological and neurophysiological studies, robotists and computer vision scientists have attempted to build systems that mimic the behaviour of the human visual system in order to predict where humans will focus their attention. As a consequence, in the past three decades numerous computational models of attention (CMAs) that attempt to locate salient regions in images and videos have been proposed.

1.1 Applications of Visual Saliency

There is a clear need for the study of VA and CMAs. Indeed, CMAs have proven useful in a number of applications. These would include:

- **Compression and recognition** (E.g. [152], [140], [19]). Content-based retrieval of images and videos is generally based around the concept of first automatically describing a media file (through the use of feature points, for instance) and then

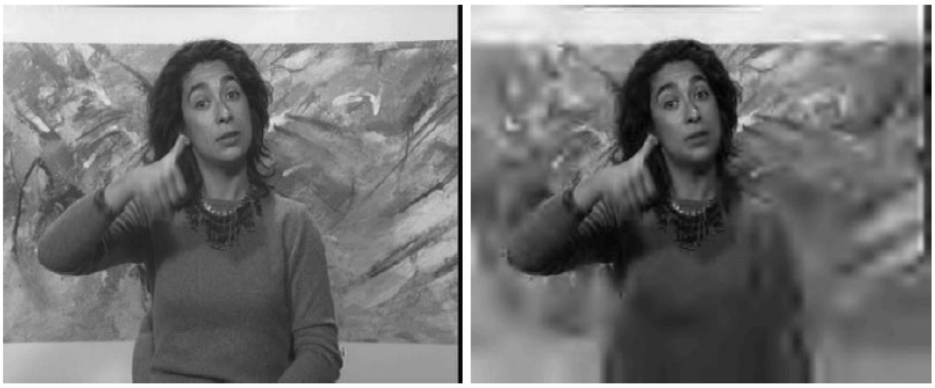


Figure 1.1: Foveated imaging example of a woman doing sign language. Image on right has its resolution adapted around the woman's hand and face (taken from [175]).

searching over these descriptions when a user queries for an image/video. Visual saliency can be used here to describe only the most important sections or regions of a media file, hence minimising the search space and retrieval time.

- **Foveated imaging and thumbnail creation**. Foveated imaging is when one varies the resolution in an image around certain fixation points. An application of this is progressive transmission for images: more important regions are transmitted with higher resolution to begin with and the remaining regions 'catch-up' later [156] [186]. Progressive transmission gives the end-user a quicker view of the most important parts of an image. Another example pertains to the transmission of videos. One can give a higher resolution to a more important part of a video hence saving transmission data on the lower resolution areas [175] [76]. An example of this is shown in Figure 1.1 where the most important area of the image (the lady doing sign-language with her hands) is given a higher resolution. Another application is thumbnail creation. Thumbnail creation is traditionally done by shrinking an original image to thumbnail size. Visual saliency can, for example, be used to create thumbnails by showing only a salient region as a summary of the image [161] [121]. This concept of thumbnail creation can also be migrated to the idea of adapting images for small displays like those found on devices such as mobile phones. It may sometimes be more beneficial for the user to see the salient parts of an image rather than the entire image itself [36] [108].

- **Image and video quality assessment**. Image and video quality can decrease as a result of compression. It is often useful to calculate the quality of media to see, for example, if the compression level is too large or to calculate the efficiency of a compression algorithm with respect to visible change. Visual saliency has been used in this task. Ma and Zhang [112], You et al. [185] and Feng et al. [51] assessed such media quality by giving more weight to quality measurements in salient regions. The justification for this is that it should be more important to measure the quality of salient regions rather than regions that a user will ignore.

- **Control of vehicles and robots, navigation, and surveillance**. Quickly calculating the state of a situation or analysing an environment by sifting out irrelevant data is significant in automating vehicular [149] and robotic control [53], [2], [155] and in tasks such as wilderness search and rescue [148]. CMAs can also be used in surveillance where important regions can be a factor in determining the position/viewpoint of a camera [8].

- **Automatic evaluation of advertisements, web pages, interface designs, etc.** [77], [29], [113]. Eye tracking is frequently utilised by advertising, web page and magazine design companies to assess the effectiveness of an advertisement, interface design, etc. An example of a heat map obtained from eye tracking of a search results page is shown in Figure 1.2. Eye tracking is, however, uncomfortable for the user and expensive, and automating this task would be very financially beneficial to these companies and others working with visual attention [43] [42].

1.2 Eye Tracking and Visual Attention

The most common way of measuring the performance of CMAs is by comparing their output to eye movements from humans that are captured by eye trackers. There is an assumption made when using eye trackers called the *Strong Eye-Mind Hypothesis*. It states that “there is no appreciable lag between what is fixated and what is processed” [90]. That is, what is being fixated on is what the subject is thinking about for the duration of the fixation. Hoffman et al. [70], however, reports that there is an approximate 100-200 milli-second lag between visual attention and eye position due to

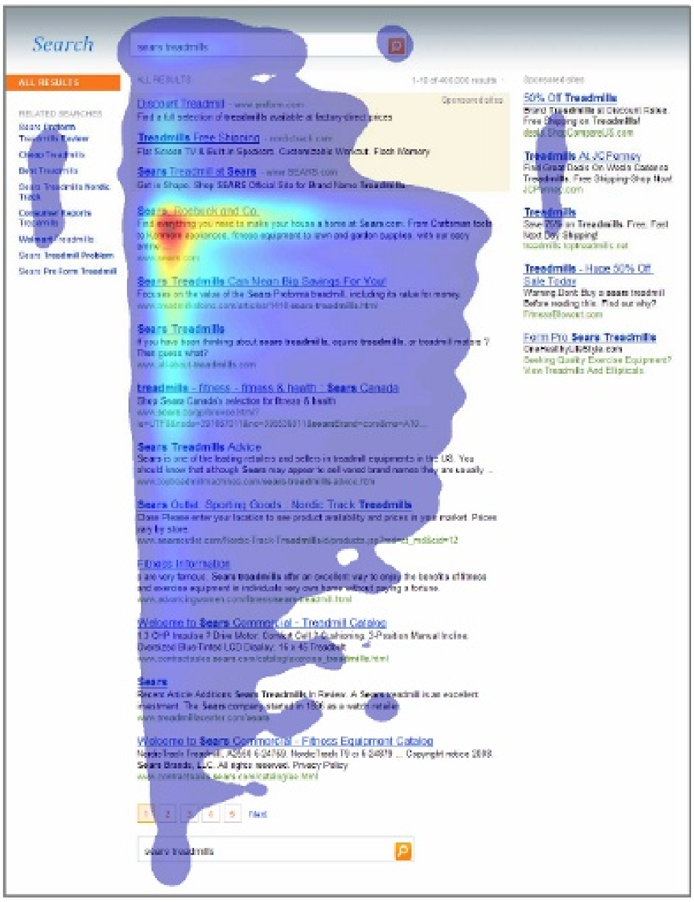
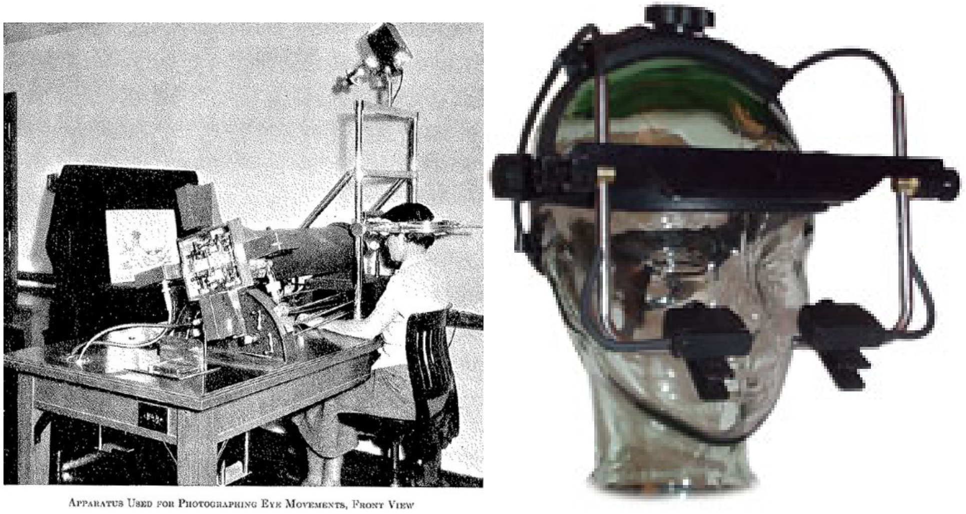


Figure 1.2: An example of a heat map of a search results page. The hotter the colour, the more attention was devoted to that area of the image for a given time period. Heat maps such as these can typically change as different time periods are analysed for users' attention (image taken from [30]).

what is called *covert attention* (discussed further in Section 2.1.1). Regardless of these findings, the eye-mind hypothesis is still a useful tool for measuring the correlation between visual attention and eye positions. Indeed, most research on VA uses this hypothesis [87].

Eye trackers have been around since the 1920s when Guy Thomas Buswell captured on film reflected beams of light that were shone onto people's eyes [31]. Since then eye trackers have become less cumbersome and invasive. Figure 1.3 shows the original Buswell eye tracker and a commonly-used-today head-mounted eye tracker. However, eye trackers still lack accuracy. For example, McDonnell et al. reported that



(a) The Buswell Eye Tracker [31] (b) The Eyelink II Eye Tracker[1]

Figure 1.3: The original Buswell eye tracker and the well-known head-mounted "Eyelink II" (SR Research Ltd, v. 2.0) eye tracker.

they accounted for approximately 100 pixels of error on a 1920 x 1200 LCD monitor when attempting to determine which object was being viewed [125]. Moreover, top-of-the-range eye trackers are very expensive - prices range from €4000-€30,000. It is necessary, therefore, to attempt to improve the accuracy of eye trackers - at least to avoid the infeasible (for most) notion of constantly upgrading eye trackers with newer, more accurate models when they become available.

1.3 Visual Attention in 2D & 3D

Most of VA research has been focussing on the visual perception of images and videos displayed on 2D screens. More recently, however, it has been shown that humans look differently at images displayed in 3D compared to 2D [172]. For example Jansen et al. [82] found that for images shown in 3D, people fixate more, have shorter and faster saccades (the movement of the eyes between fixations) and tend to explore more with their eyes.

Due to the differences in viewing behaviour CMAs, to be as accurate as possible,

[1] Image adapted from `http://www.sr-research.com/eyelinkII.html` (viewed 31/09/2014)

need to be specifically tuned or rebuilt for 3D media. Since 3D media is becoming ubiquitous (e.g. 3D films in cinemas or at home, 3D games, 3D hand-held devices such as mobile phones and the Nintedo 3DS) this is becoming an increasingly important area of research. Ultimately, all the applications listed above will need to be ported to the 3D domain.

Another problem is that the research that has been performed to compare 2D and 3D VA on videos is minimal. For example, no ground-truth eye tracking dataset exists for 3D videos. Without such a dataset, it is impossible to accurately verify the preciseness of 3D CMAs.

This thesis specifically focusses on 3D VA and CMAs and presents work that furthers this scientific field and helps with the problems mentioned above. It also presents a method to improve current state-of-the-art eye tracking measurement.

1.4 Contributions of this Ph.D.

The contributions of this thesis can be summarised as follows:

- In Chapter 3 we show that CMAs can be used to optimise content-based image retrieval and other such systems. By describing images containing only one dominant object using feature points solely collected from salient regions (as opposed to feature points obtained from the entire image), we show that an improvement in classification results can be obtained. We use the Graph-Based Visual Saliency Algorithm (described in detail in Section 4.1.1) to show that this algorithm can be applied to real applications. This algorithm is then used on 3D images in the following chapter.

- In Chapter 4 we extend the 2D Graph-Based Visual Saliency Algorithm into 3D by developing a new depth incorporation method. This new 3D CMA outperforms all other state-of-the-art 3D algorithms on 3D media.

- In Chapter 5, to more accurately measure visual saliency, we propose a new online eye tracker calibration procedure. This procedure is shown to be more accurate and faster than standard processes. It is also able to give confidence values associated with each eye position reading - something that current eye tracking procedures are unable to do.

- To better understand the differences in 2D and 3D viewing behaviour in videos we ran a full-scale eye tracking experiment with 50 participants who looked at professionally made videos. This experiment was done using our eyetracking procedure presented in Chapter 5. Chapter 6 presents this experiment and an analysis of the difference in viewing behaviour between the 2D and 3D videos. In this chapter we also present a more scientific way of presenting eye tracking heat maps. We hope that the analysis presented here will enable any future work in 3D VA to be improved. The resulting eye tracking dataset from this chapter will also be made available in the near future to the public. As was mentioned above, no such eye tracking datasets currently exist.

1.5 Publications

The work presented in this thesis has been published in the papers listed below:

- Z. Zdziarski and R. Dahyot. "Feature selection using visual saliency for content-based image retrieval". IET Irish Signals and Systems Conference, pages 1-6, 2012.

- Z. Zdziarski and R. Dahyot. "On creating a 2D & 3D visual saliency dataset". Proc. of the ACM Symposium on Applied Perception, page 132, 2013.

- Z. Zdziarski and R. Dahyot. "Extension of GBVS to 3D media". In IEEE Signal Processing and Communications Applications Conference (SIU), pages 2296-2300, 2014.

Chapter 2

Background

This background review is based on three seminal Ph.D.'s on the topic (Bruce 2008 [23], Gao 2008 [55], and Judd 2011 [87]), a recent state-of-the-art review published in 2013 by Borji and Itti [20] and is supplemented by new material published since.

Before any work, however, on visual attention (VA) and computational models of attention (CMAs) is presented, a definition of *attention* needs to be given. Perhaps the best such definition was provided by William James in 1890 [81]: "[Attention] is the taking possession by the mind, in clear and vivid form, of one out of what seems several simultaneously possible objects or trains of thought. Focalization, concentration, of consciousness are of its essence. It implies withdrawal from some things in order to deal effectively with others". This thesis, of course, focusses on visual attention, which deals with attention on visual stimuli.

2.1 Attention Mechanisms

The selective attention mechanism has already been mentioned in Chapter 1. This mechanism is commonly compared to the spotlight model [143, 49], which suggests that the fovea (the centre of the retina that has the highest resolution) is like a spotlight being directed to different areas of a dark room. Moving around (a.k.a. saccading) our fovea and fixating (i.e. focussing) on specific areas of a scene allows us to get an impression of it. There are two more groups of attention mechanisms important to VA that need to be presented: covert/overt and bottom-up/top-down.

2.1.1 Covert Versus Overt Attention

Attention can be distinguished by being either covert or overt. The overt attention mechanism involves specifically moving one's fovea to a region of interest and paying attention to it [59, 154]. For example, if you are listening and looking at someone talking to you, you are engaging the overt mechanism of attention. Covert attention does not explicitly involve movement of the eyes and head [59, 154]. An example of covert attention is concentrating on an object in the corner of one's eye. Another example is one of driving: when driving a driver focusses on the road while simultaneously keeping track of signs and traffic lights.

Covert and overt attention normally operate together [52]. It is widely believed that covert attention is used to locate interesting regions for overt attention to fixate on [20, 142]. This process was described in Chapter 1. Nonetheless, it is still possible to pay attention to things not directly encompassed by the fovea [87]. This is an ongoing piece of research and currently the standard course of action in CMA research is to not deal explicitly with covert attention. This thesis will follow suit.

2.1.2 Bottom-Up and Top-Down Attention

When building a representation of a scene, two main attentional mechanisms interact with each other: *bottom-up* and *top-down.* The bottom-up attentional mechanism (a.k.a. *automatic*, *reflexive*, *exogenous* or *stimulus driven*) pertains to the preattentive (pre-conscious) phase of attention. It is fast and involuntary. Psychophysicists, however, are still debating on which attributes exactly pertain to bottom-up attention. For example, object shape is said to be a "probable attribute" in preattentive attention guidance but no certainty exists on this. Faces, on the other hand, are thought to be "probable non-attributes" [182]. In computer vision, especially recently [20], this debate is abstracted over and bottom-up factors are understood to be ones that have been derived solely from the scene [159, 20, 21]. Top-down saliency uses top-down information that is based on conscious, higher-order cognitive processes to influence saliency analysis. It is voluntary and slower than bottom-up. Top-down information includes all task-related and personal factors such as the purposes of performing a task or things like a user's memory, emotions or likes and dislikes [76]. For example, if we need to locate a red dragon in a box of toys, we would naturally scan the scene first for red objects and from these possibilities locate the desired toy - we have prior (top-down) information so we know which salient features need to be given more weight. CMAs are distinguished by whether they rely on bottom-up influences, top-down influences or a combination of both.

Top-down saliency is a difficult problem in itself and many questions remain unanswered. E.g. are bottom-up and top-down calculations performed independently or does one influence the other? If the latter, how is this performed? How do you predict top-down influences? Because of such questions, and since bottom-up CMAs are simpler and easier to understand and implement many applications choose to remain with bottom-up CMAs. In fact, bottom-up CMAs are considered to be ‘general purpose’ attention algorithms and most research in VA is conducted on bottom-up models [20]. This thesis also focusses on bottom-up saliency.

2.2 Bottom-Up Visual Saliency Algorithms

When discussing bottom-up CMAs we limit ourselves to models that compute saliency maps for videos or images. Saliency can be understood as something that characterises parts of a scene that stand out from its surroundings. Parts of a scene can be regions or objects, for example. A saliency map, therefore, is a scalar, two-dimensional map that indicates the saliency value (conspicuity) for every pixel in an image [87]. Models that do not compute saliency maps are outside of the scope of this thesis and computer vision in general [20]. The term *visual saliency* refers to visual attention that focusses on bottom-up processes [21]. Visual saliency algorithms (VSAs) are, therefore, bottom-up CMAs.

2.2.1 Origins

Many VSAs that have been proposed to the scientific community are based on Treisman and Gelade’s [165] pioneering psychological work published in 1980. They formulated the “Feature Integration Theory” that explains which visual features are important and how they are combined to draw human attention. Christof Koch, a professor of biology and engineering at the California Institute of Technology, and Shimon Ullman, a professor of computer science at the Weizmann Institute of Science in Israel, then

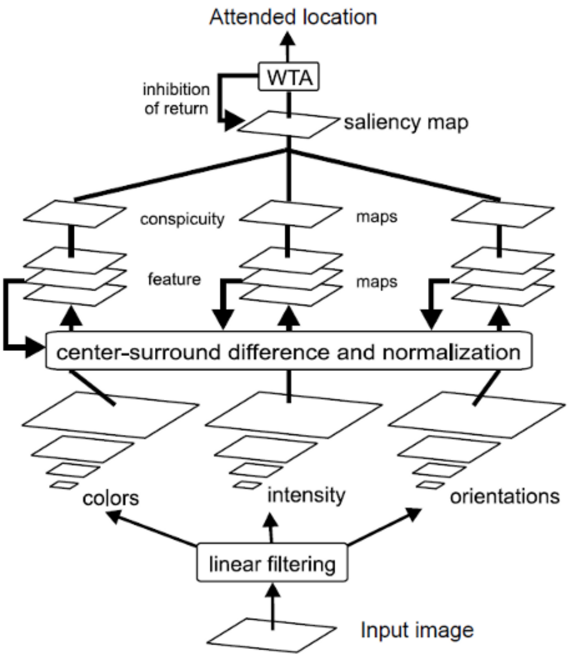


Figure 2.1: Architecture of the Koch and Ullman's model (adapted from [169])

proposed a model of this theory in 1985. Saliency in the Koch and Ullman model is first calculated by extracting the visual cues of colour, orientation and intensity from an image. An activation map (an initial saliency map) is then calculated in parallel for each of these channels. Next, the activation maps are combined into a master saliency map. Finally, a ranking of salient regions is computed through the use of a Winner-Takes-All (WTA) network. The architecture of the Koch and Ullman model is shown in Figure 2.1.

The process of obtaining a master saliency map (without the WTA step) can be summarised in three stages as suggested by Harel et al. in [68]:

1. extraction
2. activation
3. normalisation/combination

The extraction step is responsible for extracting different feature channels that will be used to calculate activation maps. An activation map (a.k.a. a conspicuity map

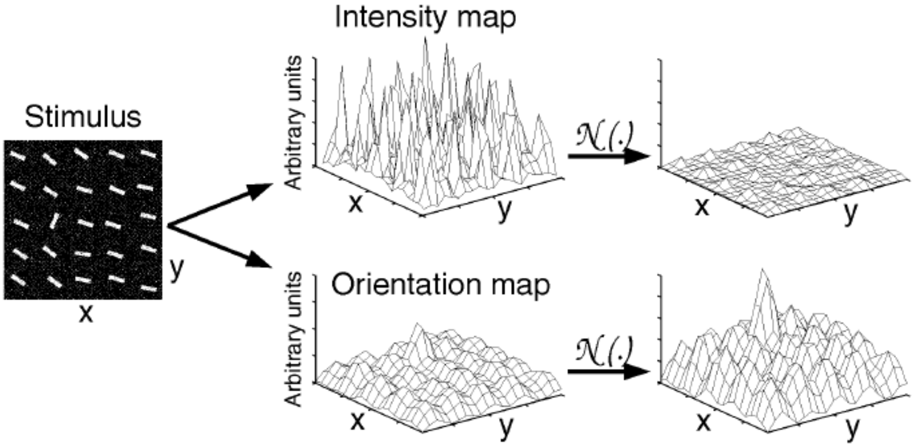


Figure 2.2: Activation and normalisation example on two feature channels - intensity and orientation. *N(.)* is a normalisation operator (adapted from [78]).

[124]) shows the salient areas of an image for a given feature channel (e.g. intensity, colour, etc.). So, for example, in the colour feature channel a yellow blob will be given higher saliency values if this blob is located on a completely black background. Or a vertical bar will be given a high saliency value in the orientation channel if the only other bars in the image are horizontal. Difference of Gaussian (DoG) is a common way of calculating saliency for each feature channel. This approach involves constructing a Gaussian pyramid for each channel and comparing the differences between scales.

The normalisation/combination stage involves the normalisation of activation maps and then amalgamation of them into one master saliency map. Normalisation is necessary since the feature cues are not all in the same range. Normalisation also takes into consideration relative differences in an activation map.

The first step shown in Figure 2.2 demonstrates the creation of an activation map for the intensity and orientation channels of an example image. This activation map is then normalised (second step in the figure) to be ready for global amalgamation into a master saliency map.

The first complete implementation and verification of the Koch and Ullman model was proposed by Itti et al. [78]. Further work to the Koch and Ullman model has generally contributed to or altered one or more of the three stages listed above [68]. Examples of further work on this model include: Milanese et al. [127], Leavers [102], and Maki et. al [117].

Since the Koch and Ullman model many more models have been proposed and implemented. The most important of these will be presented in the next section.

2.2.2 Other Important VSAs

In their review, Borji and Itti [20] divided VSAs into eight classes: cognitive, bayesian, decision theoretic, information theoretic, graphical, spectral analysis, pattern classification, and others. This thesis will follow suit. The VSAs described here have been either influential/pioneering in the study of visual saliency and/or performed well in the exhaustive comparison of 35 state-of-the-art VSAs that was made by Borji, Sihite and Itti [21] in 2013.

Cognitive models

Cognitive models are strongly based on psychological and neurophysiological findings such as, for example, Treisman and Gelade's [165] "Feature Integration Theory" described above. Itti's [78] implementation of the Koch and Ullman model is, therefore, the first obvious VSA that belongs in this section. His implementation subsamples an image into a Gaussian pyramid [63] of nine levels. Each pyramid level is then decomposed into channels for Red (R - calculated as $r-(g+b)/2$), Green ($G = g-(r+b)/2$), Blue ($B = b-(r+g)/2$), Yellow ($Y = (r+g)/2 - |r-g|/2 - b$), Intensity (the greyscale version of the image), and local Orientations (O - calculated using the Gabor filter at angles 0, 45, 90, and 135 degrees). From these, feature maps f_l are calculated by centre-surround (cs) operations. cs operations are defined as the difference between fine and coarse scales. For example, if the centre is a pixel at scale $c \in \{2,3,4\}$, the surround is the corresponding pixel at scale $s = c+d$, where $d \in \{3,4\}$. After the cs operations, the feature maps are then normalised:

$$f_l = N(\sum_{c=2}^{4} \sum_{s=c+3}^{c+4} f_{l,c,s}), \forall l \in L_I \cup L_C \cup L_O \tag{2.1}$$

where

$$L_I = \{I\}, L_C = \{RG, BY\}, L_O = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \tag{2.2}$$

and N is the normalisation operator.

In total, the *cs* operation is performed on 42 maps (6 for intensity, 12 for colour, and 24 for orientation). The calculated feature maps are finally linearly summed and normalised once more into a master saliency map. Variations and extensions of this implementation have been proposed by Frintrop [52], Walther in the Saliency Toolbox [170], and Walther et al. [171].

Itti in [76] used the concept of adding an additional channel for motion to the Koch and Ullman model for the video domain. This addition is depicted in Figure 2.3 with the red boundary showing the standard Koch and Ullman model. Along with standard colour, intensity and orientation information, Itti added two temporal feature channels: temporal flicker (flickering of light intensity) and four oriented motion energies: up, down, left, and right. Just like in the Koch and Ullman model, all these feature channels were normalised and then merged into a single saliency map. Interestingly, Itti applied this model to foveation. He reports a 50% average increase in video compression rate performance for MPEG-1 and MPEG-4 video clips.

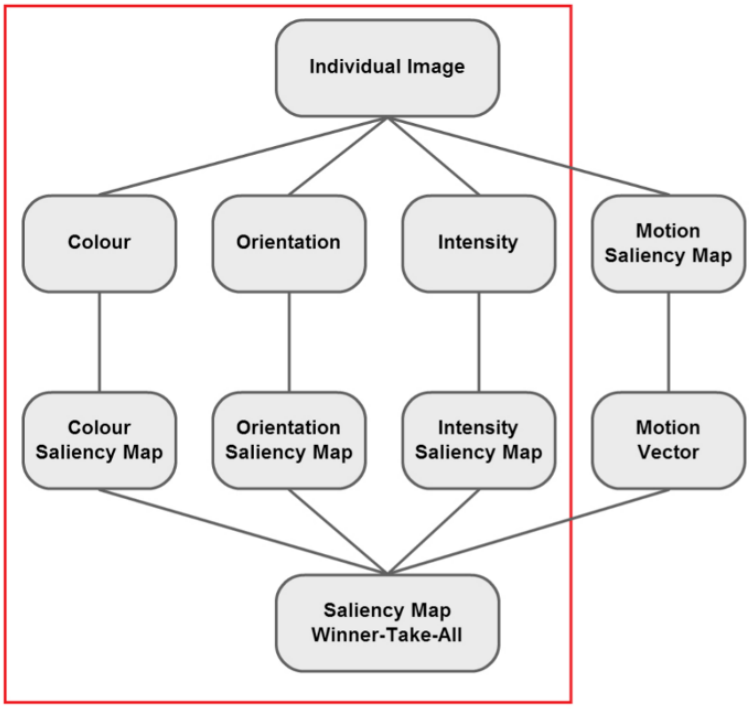


Figure 2.3: The Koch and Ullman model (in red) with a motion channel attached as explored by Itti in [76].

Since the Koch and Ullman model was proposed, research and understanding of the human visual system (HVS) has advanced. Le Meur et al. [100] proposed a VSA that modelled the HVS in a more complex way. Contrast sensitivity functions, visual masking and perceptual grouping are some of the additional functions they implemented. This algorithm was shown to outperform Itti's [78]. Le Meur et al. [101] later extended their model to the spatio-temporal domain by proposing a fusing algorithm for achromatic, chromatic and temporal information.

Marat et al. [120] proposed a spatio-temporal VSA that was inspired by the first steps of the HVS. Modelling the output of the retina, two signals are extracted from a video stream: parvocellular and magnocellular. Static and dynamic information (signal orientation, spatial frequencies and optical flow calculated using spatial Gabor-like filters [27]) is then extracted from these streams from which two saliency maps are calculated. These are finally fused into a spatio-temporal map. Marat et al. showed that they were able to accurately predict the eye movements of subjects for the first few frames of each short clip they analysed.

Bayesian models

In bayesian models, prior knowledge about the scene (e.g. scene context or gist) and sensory information (e.g. target features) are combined probabilistically in Bayes' rule to detect salient regions or objects of interest. Only one notable bottom-up algorithm from this class has been proposed.

Itti and Baldi [79] introduced a Bayesian definition of surprise as being something that changes the beliefs of an observer. Prior beliefs of an observer are captured and new data being observed is said to be surprising if the posterior resulting from new observations significantly differs (according to the KL divergence measure) from the prior. For images, prior information is taken as neighbouring locations. In the time domain, prior information at one point is captured from previous observations.

Decision theoretic models

Decision/Discriminant theoretic models are derived explicitly from a minimum Bayes error definition. They are based on the premise that perceptual systems evolve to make decisions that are optimal in a decision theoretic sense. The main idea is that

CMAs should be optimised with respect to the end task. Most models in this class are top-down in nature but some have been implemented to remain bottom-up while maintaining a decision theoretic framework.

Gao and Vasconcelos [56] define a decision theoretic formulation of saliency of visual features at a given location as the power (expected classification accuracy) of those features to discriminate (distinguish) between them and a null hypothesis. In bottom-up visual saliency, the null hypothesis for Gao and Vasconcelos is the neighbourhood (surroundings) of the given location. Optimality is defined in the minimum probability of error sense.

The binary classification problem (i.e. discriminating between stimuli of interest against the null hypothesis) in decision theoretic models is extended to the temporal domain by Mahadevan and Vasconcelos [115, 116]. They included motion features in a bottom-up saliency mode. Similarly to Gao and Vasconcelos [56], the neighbourhood is defined as the null hypothesis. Their spatio-temporal model was shown to robustly identify salient moving objects for complex backgrounds by using dynamic texture models as a substitute for optical flow (that can be used for more static clips).

Information theoretic models

Information theoretic models aim to detect regions that maximise the information sampled from an environment. That is, they aim to detect the most informative regions while discarding the rest.

Bruce and Tsotsos [24] developed the well-known AIM model (Attention based on Information Maximisation) that uses Shannon's self-information measure. Saliency of a region is calculated as the information it conveys relative to its surroundings. This is calculated as $I(X) = -log(p(X))$, where X is a visual feature and $p(X)$ is the probability of observing X based on its surround. To calculate $p(X)$, independent component analysis (ICA) is used to reduce the dimensionality of the problem. The bases for ICA are learned by sampling from a large number of random patches from natural images (since a single image will not have enough data for this). The final saliency value for a given image region is the product of the probabilities of observing this region for each ICA basis coefficient.

Mancas [118] developed a rarity and contrast-based measure of saliency. Contrast he calculates on a global and local level. Global contrast is measured by analysing histograms and local contrast by a centre-surround operation similar to that of Itti et al. [78]. Rarity is first calculated by computing the mean and variance of the neighbourhood and using other features such as size and orientation (e.g. smaller areas get higher saliency values). Finally, higher-level methods (e.g. Gestalt laws of grouping) are used to locate the salient regions.

Wang et al. [174] proposed a computational model to simulate human saccadic scanpaths on natural images. They integrated three factors to guide eye movements sequentially: 1) reference sensory responses (to provide a representatinon of the raw input signal); 2) fovea-periphery resolution discrepancy (to provide detailed information around a location and coarse details from the periphery); and 3) visual working memory (to prevent fixations from returning to previously fixated regions too early). For each eye movement, three multi-band filter response maps are calculated for the three factors. These filter response maps are then combined into multi-band residual filter response maps on which residual perceptual information (site entropy rate) is calculated at each location. Fixations are ordered by residual perceptual information scores.

Graphical models

Graphical models have a probabilistic graph model denoting the conditional independence between different variables. Graph-based approaches such as Hidden Markov Models, Dynamic Bayesian Networks and Conditional Random Fields can be employed in calculations.

Harel et al. [68] developed the Graph-Based Visual Saliency (GBVS) algorithm. The GBVS algorithm defines a Markov chain over each image channel. States in the chain represent nodes (pixels) and transition probabilities represent edges that denote similarity between two nodes. Each node is connected with every other node. Random walks are then performed on these nodes - the more frequently visited nodes are deemed more salient. This solution claimed to be biologically plausible (nodes here are claimed to be representations of neurons, the graph structure is a retinotopically organised network and communication between nodes is similar to synaptic firing) and strongly parallelisable. The GBVS algorithm is presented in more detail in Section 4.1.1.

Liu et al. [110] tackle the visual saliency problem for images with a single salient object. They propose new features to be used in their calculations such as multiscale contrast, center-surround histogram, and color spatial distribution. These are used to describe a salient object locally, regionally and globally. They use images from a large annotated dataset of 20,000+ images to train a conditional random field to effectively combine these features. Their model is extended to the spatio-temporal domain (Liu et al. [109]) by adding motion features (using SIFT flow [107]) to their calculations.

Spectral analysis models

Spectral analysis models perform their saliency calculations in the frequency domain as opposed to the spatial domain. These models are generally simple to explain and implement and can generate saliency maps in real-time.

Hou et al. [74] explored the properties of backgrounds by investigating the log spectrum of images to then extract the spectral residue. Finally, the spectral residue was transformed to the spatial domain to create a saliency map. This spectral residue technique has been used previously in, for example, sensory input to detect unexpected signals or anything that varies from the norm [12, 93].

The first step in Achanta et al.'s [4] model is to transform the colour image I to the CIELAB colour space. The saliency map is then calculated as:

$$S(x,y) = ||I_{\mu} - I_{w_{hc}}|| \tag{2.3}$$

where x, y are pixel coordinates, I_{μ} is the arithmetic mean image feature vector, $I_{w_{hc}}$ is a Gaussian blurred (with a 5 x 5 separable binomial kernel) version of I and $||.||$ is the euclidean distance. Achanta et al. is an example of a global contrast VSA meaning that it measures the saliency of a pixel by calculating its contrast to every other pixel in the image. These methods are good at detecting single salient objects on simple backgrounds but their performance deteriorates the more complex a scene becomes [184].

Bian and Zhang [17] proposed the Spectral Whitening (SW) model where they used spectral whitening (a.k.a. balancing or broadening). The first step in the calculations is to scale the image to a fixed size (a length of 64 pixels is used by Bian and Zhang with the image ratio being retained). A windowed Fourier transform F is then calculated:

$$f(u,v) = F[w(I(x,y))] \tag{2.4}$$

where I(x,y) is the resized grey-scale image and w is a windowing function. A whitened (or normalised/flattened) spectral response n is obtained by:

$$n(u,v) = f(u,v)/||f(u,v)|| \tag{2.5}$$

This response is then transformed into the spatial domain (reverse Fourier transform), squared to further enhance the salient regions and convolved with a Gaussian low-pass filter:

$$S(x,y) = g(u,v) * ||F^{-1}[n(u,v)]||^2 \tag{2.6}$$

where g is a Gaussian filter. The model is further extended to the spatio-temporal domain by attempting to separate background motion from localised (salient) motion. Background motion is detected by utilising phase correlation [39] of two frames. The resulting motion vector for panning movement is used to shift the global motion of the two frames to extract the local motion.

Pattern classification models

In these models machine learning techniques are used to model visual attention. The techniques use pre-recorded eye fixations or manually labelled salient regions as training data.

Kienzle et al. [92] used eye tracking data to train a support vector machine (SVM) to model attention. Their aim was to find the functional relationship between image patches and their corresponding visual saliency. The SVM was trained on a non-linear mapping between these patches and their saliency scores (obtained from 200 grey-scale images that were viewed by 14 people). Positive values were given to patches that had been fixated on and negative values to randomly selected patches. Only the intensity channel was used.

Judd et al. [88] proposed a similar approach but they trained a linear SVM using a set of low (e.g. colour, intensity and orientation), mid (horizon line detector), and high-level (people and face detectors) image features. Saliency values for these features were gathered from eye tracking data obtained from 15 people who viewed 1,003 images.

Other models

Other models that do not conform to the above classes are described in this section.

Goferman et al. [60] propose a more context-aware VSA. Four principles of human attention govern their salient region detection mechanism: low-level feature (colour, contrast, etc.) considerations, the suppressing of frequently occurring features on the global scale, the idea that salient patches tend to group together, and finally high-level feature considerations such as faces. Goferman et al. applied their VSA to image retargetting and summarisation.

Garcia-Diaz et al. [58] presented the Adaptive Whitening Saliency (AWS) model that is based on the adaptive whitening of colour and scale features. This is achieved through decorrelation and contrast normalisation in several steps in a hierarchical approach. The first step is to transform an (r, g, b) image into a whitened (z_1, z_2, z_3) representation. This representation is acquired through decorrelation by employing principal component analysis over multi-scale low-level features. A bank of log-Gabor filters is used (for orientations of 0, 45, 90, and 135 degrees) to create feature maps over (z_1, z_2, z_3). Seven scales are calculated for z_1 and five each for z_2 and z_3. Each feature map from the chromatic component is then whitened and contrast normalised in a hierarchical manner. The square of the vector norm in the resulting representation is the final saliency computation.

Table 2.1 shows the VSAs presented in this section accordingly classified into one of the eight VSA classes.

2.2.3 Summary of VSAs

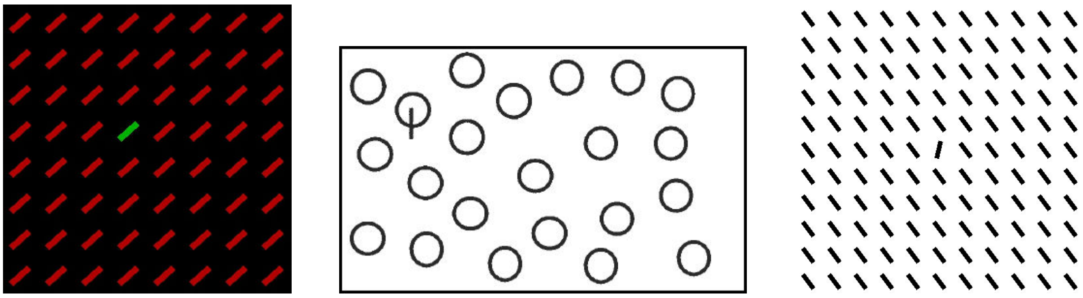
In 2013, as an accompaniment to the state of the art review of Borji and Itti [20], Borji, Sihite and Itti [21] performed an exhaustive comparison of 35 state-of-the-art VSAs against 54 classic synthetic patterns, three natural image datasets (Bruce and Tsotsos [26], Kootstra et al. [98], and Judd et al. [88]), and two video datasets (Mital et al. [128] and selected clips from Itti and Baldi [79]). Three metrics were used for evaluation: correlation coefficient (CC), normalised scanpath saliency (NSS) and shuffled area under curve (AUC) (these are described in Section 2.4 below). They

VSA Classes	VSAs
Cognitive	Itti (1998) [78], Itti* (2004) [76], Walther (2006) [170], Walther et al. (2002) [171], Frintrop (2006) [52], Le Meur et al. (2006) [100], Le Meur et al.* (2007) [101], and Marat et al.* (2009) [120]
Bayesian	Itti and Baldi* (2004) [79]
Decision Theoretic	Gao and Vasconcelos (2009) [56] and Mahadevan and Vasconcelos* (2009) [115, 116]
Information Theoretic	Bruce and Tsotsos (2009) [24], Mancas (2007) [118] and Wang et al. (2011) [174]
Graphical	Harell et al. (2007) [68], Liu et al. (2007) [110], and Liu et al.* (2008) [109]
Spectral Analysis	Hou and Zhang (2006) [74], Achanta et al. (2009) [4], and Bian and Zhang* (2009) [17]
Pattern Classification	Kienzle et al.* (2009) [92] and Judd et al. (2009) [88]
Others	Goferman et al. (2012) [60] and Garcia-Diaz et al. (2009) [58]

Table 2.1: VSAs presented in this section in their respective classes. * indicates a spatio-temporal algorithm.

report a number of findings. First, for synthetic images, models based on the feature integration theory performed well (e.g. Itti et al. [78] and Frintrop et al. [52]). The best results were obtained by Harel et al.'s GBVS algorithm [68]. Garcia-Diaz et al. [58] and Bian and Zhang [17] performed highly as well. Synthetic images were classic patterns that have been frequently used for psychophysical experiments and evaluation of attention models. These patterns contain only one item (target position) in each image that differs from all other (distractor) items. Example synthetic patterns used in the comparison experiments can be seen in Figure 2.4.

Overall, the best model for natural scenes using the CC and NSS metrics was the GBVS model [68]. With the shuffled AUC metric, the Garcia-Diaz et al. [58] model is significantly better than all the other models over the three datasets. The GBVS model performed well with this metric, too. For the two video datasets, it is interesting to note that models explicitly incorporating motion did not perform better than static models. The ranking of VSAs here is similar to the one with natural images. That



(a) Difference in colour

(b) Assymetry in one item

(c) Difference in orientation

Figure 2.4: Examples of classic synthetic patterns used in the comparison of VSAs by Borji et al. [21].

is, on average, GBVS out-scores all other models when using the NSS and CC metric. Garcia-Diaz et al. [58] is the best performer for both datasets with the shuffled AUC metric. Bian et al. [17] performed second-best. In light of these results for videos, Borji, Sihite and Itti call for the best performing static models to be extended into the spatio-temproal domain to possibly improve their results even more. With respect to computation time, Hou et al. [74] and Bian and Zhang [17] are the two fastest VSAs (0.30 secs and 1.1 secs on average, respectively) implemented in Matlab. These provide an option for trade-off between accuracy and speed sometimes necessary for real-life applications.

2.2.4 Saliency and Motion

Since this thesis deals with 3D/2D videos, it is appropriate that a separate section be dedicated to the discussion of motion saliency. Of course, standard visual saliency models can be used on a per frame basis on videos but the additional dimension of time provides supplementary information that can be employed to more accurately find salient regions. This temporal information, however, does add significant complexity to the equation because we can, if we choose, deal with things like occlusion and reappearance, changing lighting conditions when switching from indoor to outdoor locations and vice versa, constant camera angle changes of the same scene, etc. A number of papers have tackled these issues and a few of the more important ones will be discussed here.

It was already mentioned in Section 2.2.2 that Itti in [76] used the simple concept of adding an additional channel for motion to the classic Koch and Ullman model (cf. Figure 2.3). Itti compared his motion saliency results against human behaviour (with an eye tracker) to show that "subjects preferentially fixated locations which the model also determined to be of high priority, in a highly significant manner".

Williams and Draper published an interesting paper in which they concluded that "adding motion channels does not improve the performance of saliency-based selective attention" [180]. They used the same approach as Itti in [76] by adding a separate motion channel to the Koch and Ullman model. Their conclusion is surprising but a criticism of their report is that they used an overly simplistic motion analysis method - the Lucas-Kanade algorithm. This algorithm only looks at two adjacent frames for detecting potential salient regions. If the motion estimation algorithm is not accurate (i.e. not robust), then this may explain why it does not provide good information for saliency.

A complex solution was proposed by Jeong et al. in [83]. Their system is an amalgamation of bottom-up and top-down saliency techniques that combine motion, symmetry, as well as depth information. Motion information is analysed in the penultimate stage of saliency calculation: after standard low-level feature analysis and before incorporating depth data. Their motion analysis uses the model proposed by Fukushima in [54] and is partly biologically based. Motion analysis is only conducted on salient regions as calculated in step 1 and the rotation, expansion, contraction, and planar characteristics of motion as well as temporal contrast changes (used to retrieve local and then later relative velocity) are then examined. Motion changes in each of these characteristics were given different weight values. Results from this were merged with depth information to produce a final saliency map.

A very interesting analysis on motion in visual saliency was performed by Arpa et al. [28, 10]. They propose (after considering earlier psychological studies, e.g. Anderson [9], Abrams and Christ [3] and Koffka [96]) that objects in a spatio-temporal scene are in any one of the following six states at a given time: Static (no change of location), Object Appearance (object has just appeared in the scene), Motion Onset (motion has started), Motion Offset (motion has ended), Continuous Motion (motion is kept with the same velocity), Motion Change (motion has changed in direction or speed). Objects are then given a saliency value according the state they are in and the inhibition of return principle (if an object has already been viewed, its saliency value will be decreased). Each of these states have different saliency values, which were quantified by eye tracking experiments. Furthermore, a global saliency value is attributed to each object according to all relationships with other objects. Figure 2.5 depicts the motion cycle of an object, i.e. its 6 possible states and the relationships between them.

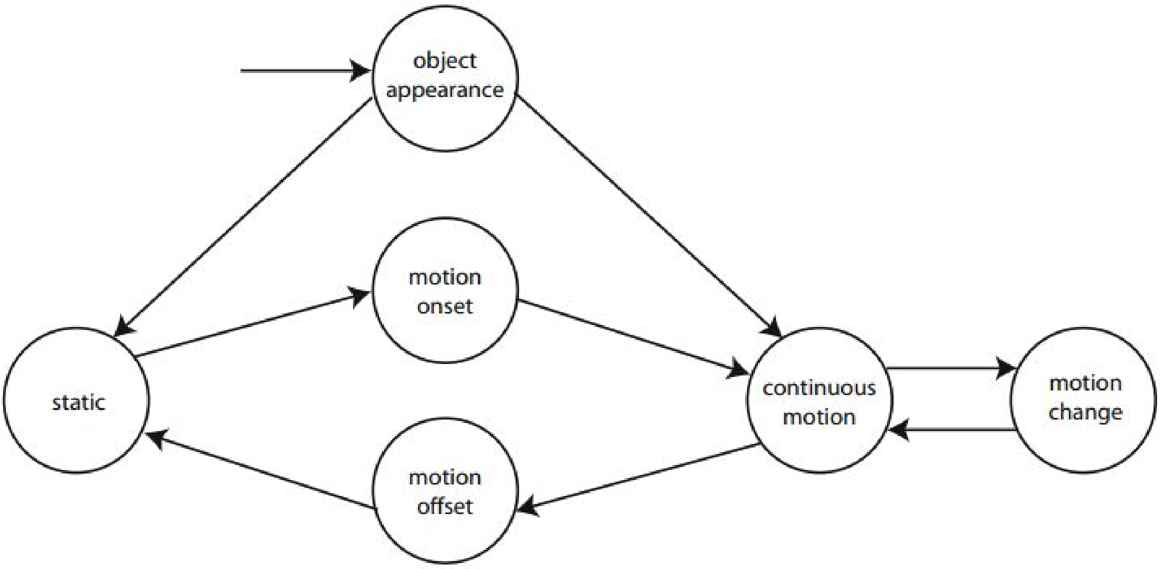


Figure 2.5: The motion cycle of an object in the spatio-temporal domain according to work performed by Bulbul et al. [28].

Bastan et al. in [13] in 2010 developed a content-based video retrieval (CBVR) system called BilVideo-7. This MPEG-7 compatible system has automatic content generation features due to which one can search over a video database with quite complex search terms such as “Golfer above golf cart” or “Clinton left Blair”. Motion analysis such as object trajectory and camera motion, among other techniques, is employed here to find salient objects that are then inserted into a video’s description.

Recently, Gao et al. [57] proposed a new background subtraction technique for videos to segment well-defined foreground regions. This technique deals with challenges such as illumination change, background motions (trees, waves, etc.) whose magnitude can be greater than those in the foreground, poor image quality, camouflage, etc. The motion saliency algorithm used by Gao et al. is based on the tracking of foreground candidate blocks detected via dense optical flow. Only those blocks whose trajectories are consistent for a certain minimum amount of duration are deemed to be salient and flagged as belonging to the foreground. Their background subtraction technique was shown to significantly outperform many state-of-the-art approaches.

Also recently, Li et al. [104] proposed a saliency-based unsupervised video object extraction framework. First, to detect each moving part, motion saliency was calculated using dense optical flow forward and backward propagation. Then, shape information was learned through motion cues for characterising each detected object. Next, standard saliency calculations (using colour and contrast information) were used to supplement the motion-induced shape information. Finally, conditional random fields were employed to combine the salient features to automatically detect objects. This technique was shown to deal well with unknown pose and scale variations of objects.

2.3 Eye Tracking Datasets

There are a number of eye tracking datasets available in the public domain for both images and videos. We limit ourselves here to presenting only those datasets that are predominantly used to evaluate and compare VSAs. Since we are dealing with only bottom-up saliency in this thesis, we also only present datasets in which participants were free-viewing (i.e. were not given a task to perform during the experiment). All participants in these datasets had normal or corrected-to-normal vision.

2.3.1 Image Datasets

Table 2.2 shows the datasets available in the public domain for images. Wang et al. [173] is the only dataset with 3D images.

2.3.2 Video Datasets

Table 2.3 shows the datasets available in the public domain for video clips. Mathe et al. had 16 subjects perform their experiment but only 4 of them free-viewed the videos. Marat et al. [119] had a large number of clips w.r.t. a small total length of videos because each clip was only 1-3 seconds long. Le Meur et al. [101] had a different number of people viewing each of the 7 clips. The average number of observers per clip was 22. No eye tracking datasets exist for 3D videos.

Study	Subjects	Dataset Size	Resolution
Kienzie et al. [92]	14	200	1024 x 768
Einhauser et al. [44]	7	54	640 x 480
Bruce and Tsotsos [26]	20	120	681 x 511
Stark and Choi [158]	7	15	15 x 20cm
Judd et al. [88]	15	1003	Various
Cerf et al. [33]	7	250	1024 x 768
Peters et al. [141]	4	100	Various
Kootstra et al. [98]	31	99	1024 x 768
Tatler [164]	14	48	800 x 600
Engmann et al. [48]	8	90	1280 x 1024
Engelke et al. [47]	30	7	512 x 512
Le Meur et al. [100]	40	46	800 x 600
Rajachekar et al. [146]	29	101	1042 x 768
Wang et al. [173]	35	18*	1920 x 1080

Table 2.2: Free-viewing eye tracking datasets on images available in the public domain for CMA evaluation and comparison. * indicates that images were displayed in 3D.

Study	Subjects	Dataset Size	Length	Resolution
Mital et al. [128]	42	26	42mins	1280 x 960
Marat et al. [119]	15	324	68secs	720 x 576
Le Meur et al. [101]	17-27	7	98secs	800 x 600
Mathe and Sminchisescu [122]	4	3869	21hrs	Various
Dorr et al. [41]	54	18	6mins	1280 x 720
Hadizadeh et al. [65]	15	12	105secs	352 x 288
Alers et al. [6]	12	25	8mins	1280 x 720
Li et al. [105]	14	50	8mins	1920 x 1080

Table 2.3: Free-viewing eye tracking datasets on videos available in the public domain for CMA evaluation and comparison. Dataset size is measured in number of video clips.

2.4 Evaluation Measures

The most common way of measuring the performance of VSAs is by comparing their output to eye movements from humans that are captured by eye trackers. There are a number of popular ways of doing this and the next few sections will present them.

There is no consensus as to which or how many of these methods should be used to assess a VSA. In practice one, sometimes two, methods are used and these are chosen at the VSA authors' discretion [20]. To assess VSAs these metrics can be used on the publically available eye tracking datasets (for images and videos) described in the previous section.

2.4.1 Kullback-Leibler Divergence (KLD)

The KLD is usually used to measure the distance between two probability distributions. It is similarly used in the context of visual saliency where it calculates the dissimilarity between the saliency maps obtained from saliency algorithms with fixation density maps (maps created from eye tracking experiments). Fixation density maps are obtained by convolving a Gaussian filter across fixation locations of all observers [88]. These maps are treated as two probability density functions (PDFs), defined as H and P respectively in the following equation:

$$\mathrm{KLD}(H, P) = \sum_{x} h_x \ln\left(\frac{h_x}{p_x}\right) \tag{2.7}$$

where h_x and p_x denote the values of the normalised maps of H and P respectively at pixel location x [100, 126]. The lower the KLD value obtained, the better the result (with a value of 0 denoting equality). This metric was used to assess a VSA, for example, in the studies of Wang et al. [173] and Itti and Baldi [79].

2.4.2 Pearson Linear Correlation Coefficient (PLCC)

The PLCC measures the linear correlation between the saliency and fixation density maps H and P:

$$\mathrm{PLCC}(H, P) = \frac{\mathrm{Cov}\,(H, P)}{\sigma_H \sigma_P} \tag{2.8}$$

where $\mathrm{Cov}(H, P)$ is the covariance and σ_H and σ_P denote the standard deviations of H and P respectively. The PLCC is a simple calculation that is also invariant to linear transformation [126]. The higher the PLCC value, the better the result. Studies that have used this metric include Le Meur et al. [100] and Guo et al. [64].

2.4.3 Area Under Curve (AUC)

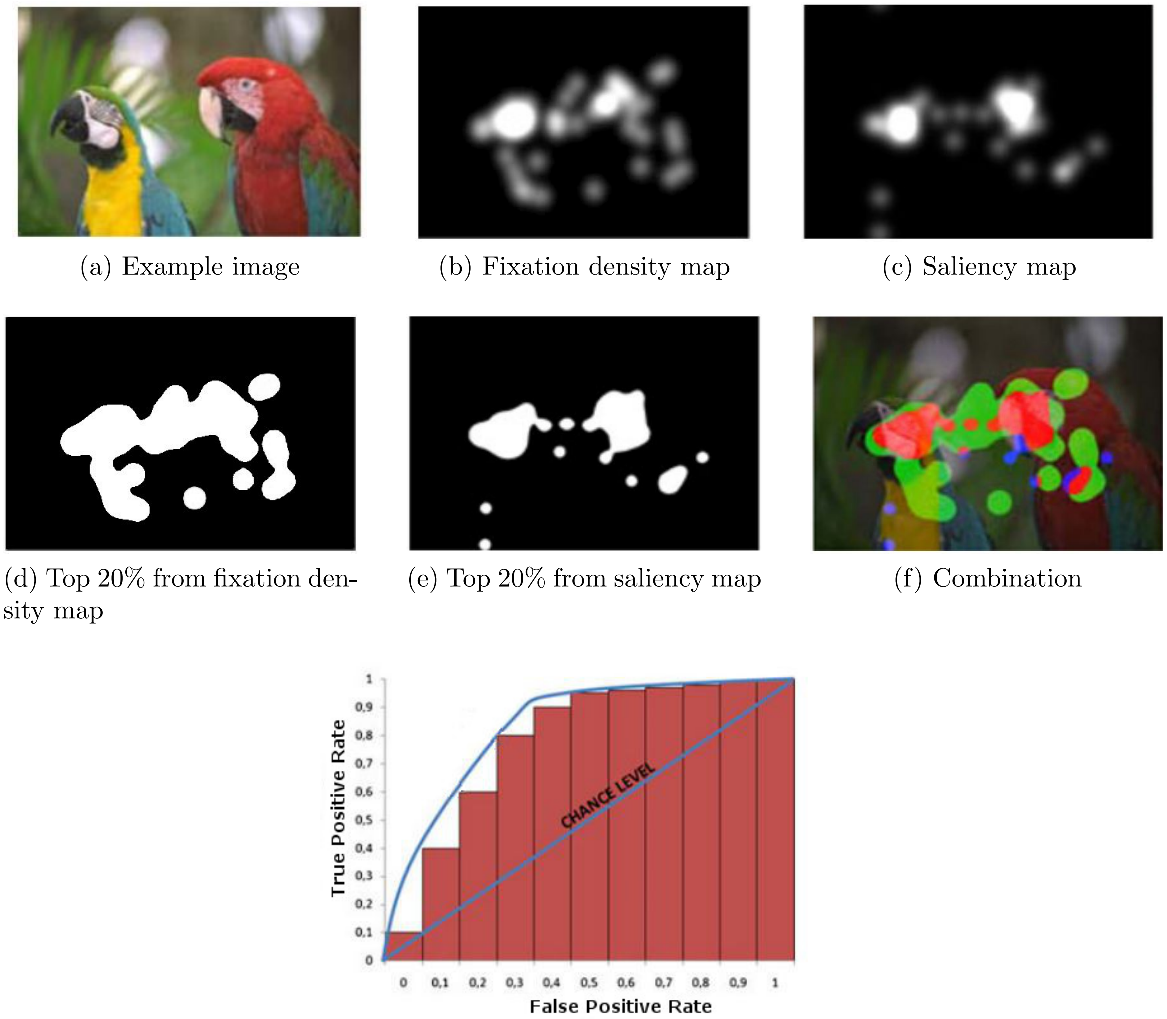
The AUC metric is the area under the Receiver Operating Characteristic (ROC) curve [62] on the saliency and fixation density maps. Each map is processed by a binary classifier (a variable threshold) applied to every pixel. If a value in the saliency map is larger than a threshold it is said to be fixated on with the rest of the pixels said to be nonfixated [26, 33]. The same is done to the fixation density map and a comparison between fixated and nonfixated pixels is made between the two maps. The threshold of the binary classifier is varied and an ROC curve is drawn as the *false positive rate* versus the *true positive rate*. The area under this curve is said to indicate how well the saliency map aligns with the ground truth (with the value of 1 indicating a perfect alignment) [20, 126].

Figure 2.6 shows (a) an example image, (b) corresponding fixation density map, (c) saliency (prediction) map and then (d) & (e) the top 20% of the salient areas from the ground truth and saliency map with (f) the classification result and (g) the final ROC curve. In this example the AUC is approximated by a left Riemann sum (represented by the red rectangles).

The AUC metric was used, for example, by Harel et al. [68] and Bruce and Tsotos [26] to assess their VSAs.

2.4.4 Normalised Scanpath Saliency (NSS)

The normalised scanpath saliency [137] measures the saliency values at fixation locations along a person's scanpath [126]. Borji and Itti [20] define it as a response value at given human eye positions from a saliency map. The first step in the calculation is to normalise the saliency map to have zero mean and unit standard deviation. Then the NSS is calculated as the mean of the saliency values at each fixation location along a participant's scanpath. NSS values greater than 0 indicate a better correspondence than random fixation locations [141]. The NSS was used in the studies of Hou and Zhang [74] and Pang et al. [136].



(a) Example image

(b) Fixation density map

(c) Saliency map

(d) Top 20% from fixation density map

(e) Top 20% from saliency map

(f) Combination

(g) AUC Graph

Figure 2.6: (a) Example image; (b) Fixation density map (ground truth); (c) Saliency map; (d) & (e) top 20% regions of the fixation and saliency maps respectively; (f) Classification results: red - true positives, green - false negatives, blue - false positives, other areas are true negatives; (g) ROC curve (adapted from [126]).

2.4.5 String Edit Metric

The string editing metric/distance is calculated by first grouping fixations into regions of interest (ROIs) using, for example, k-means. ROIs are then translated into a sequence of numbers or letters and ordered by the temporal ordering of fixations. For example, $string_{observer1}$ = "*abdffcd*" would indicate a sequence of fixations by an observer from region ‘a’ to ‘b’ then to ‘d’, etc. The prerequisite here from the CMA would be that its output is also composed of an ordering of ROIs (e.g. regions with higher saliency values can be given a higher temporal order). A string edit distance (also called the *Levenshtein distance* from Levenshtein [103]) is then calculated to measure the similarity between the string obtained from the ground truth and the string obtained from the CMA. The distance is calculated as the number of edits (deletions, insertions and substitutions) needed before the two strings are identical [145, 22]. Since this metric requires a segmentation and subsequent ordering of ROIs from the CMA, it is rarely used in practice. It receives, however, constant reference in the literature and on this merit is included in this review [20].

2.4.6 Challenges and Problems

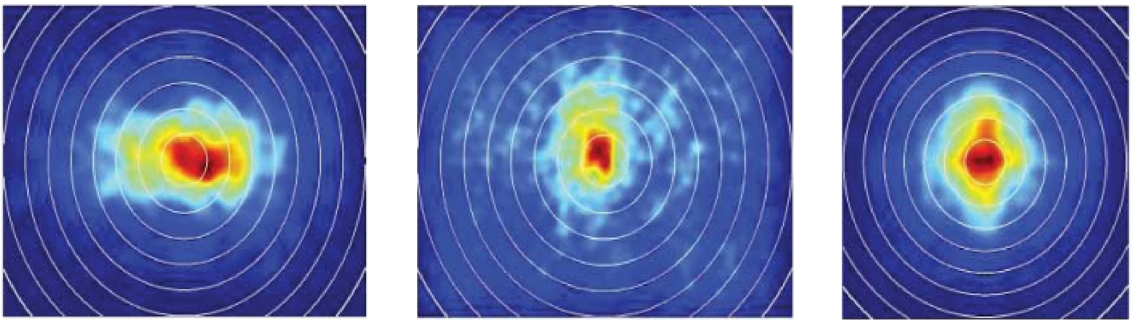
There are two important challenges/problems that need to be described here that deal with the evalution of CMAs: centre-bias and the edge effect.

Centre-Bias

A challenge in visual saliency is linked to centre-bias, which is the tendency of observers to fixate more on centre regions of images. There are two main reasons why people do this: 1) photographers tend to place salient objects in the middle of their images; and 2) observers, by fixating in the middle of an image, can use this as a strategy to acquire a quick global view of the scene [154, 164].

A number of studies have been performed to analyse centre-bias. For example, Judd et al. [88] in 2009 found that a simple Gaussian blob model in the centre of an image outperformed most saliency models at the time. Borji et al. [21] in 2013 created heat maps of fixations over all images from three datasets. Figure 2.7 depicts these heat maps in which clear centre-biases can be observed. For example, in the Bruce and Tsotsos dataset, 80% of fixations are found within the 40% circle radius.

A few CMAs attempt to deal with centre-bias explicitly by increasing saliency values closer to the centre of images (e.g. Judd et al. [88], Parkhurst and Niebur [138] and Zhang and Koch [191]). However, dealing with centre-bias explictly makes fair comparison of models challenging. In fact, Borji et al. [21] regard centre-bias as the biggest challenge in visual saliency model comparison. Three possible solutions are



(a) Bruce dataset [26] (b) Kootstra dataset [98] (c) Judd dataset [88]

Figure 2.7: Fixation heat maps for all images from the: (a) Bruce dataset; (b) Kootstra dataset; and (c) Judd dataset. White rings depict 10% increase in distance from centre of the image (adapted from [21])

given by them: 1) all CMAs add a centre-Gaussian blob to their saliency maps; 2) create a dataset that avoids the centre-bias; and 3) design better CMA evaluation metrics. The first solution would be very hard to police and the second solution will still not stop people from looking at the centre to acquire a global view of a scene. The third solution suggests using a shuffled version of the AUC metric.

The shuffled AUC metric was designed by Zhang et al. [190] in 2008. It is calculated by, for a given image, associating all fixations from one observer with the positive class. The negative class, however, is composed of a union of fixations from other subjects from other images [21]. Assuming that a centre-bias exists in the dataset, by including all other fixations in the negative set less credit is, therefore, given to central fixations. The idea, then, is that by placing a stronger emphasis on the assessment of fixations away from the centre (which are harder to predict) one should possess a better CMA evaluation technique. In fact, Borji et al. [21] regard the shuffled AUC metric as the most suitable metric for evaluating CMAs.

Edge Effect

The edge effect is intrinsically linked to the centre-bias problem. It is defined as adding a border of zeros (no saliency) around a saliency map. One can monitor the results from CMA evalution metrics and vary the size of this border accordingly. Zhang et al. [190] analysed the edge effect by creating a dummy white saliency map (of size 120 x

160 pixels) and varying the size of the black border. They found that ROC and KL scores increased when the black border was incremented from 0 through to 8. Since the edge effect is linked to centre-bias (people tend to look at the centre of the image so we remove any CMA output situated along the border), solutions proposed for centre-bias are applicable here also.

Chapter 3

Feature Selection Using Visual Saliency for Content-Based Image Retrieval

Content-based image retrieval (CBIR) is an area of research that aims at defining relevant visual features for efficient content retrieval in image databases. In this chapter we investigate if visual saliency can help in selecting visual features for retrieval and consequently reduce the computation time and memory consumption needed for visual feature storage. With one parameter (threshold), we control the number of selected features in images used for retrieval. We show that even with a small number of features, the classifiers trained to find the objects of interest still perform well. This result can be used to scale CBIR systems depending on the computational resources available on the device being used (e.g. tablets, mobiles, etc.).

3.1 Previous Work

An important question that needs addressing for this chapter is whether bottom-up saliency can be used for object detection and whether this object detection can be reliable. Indeed, this question was tackled by Elazary and Itti [46] who used Itti et al.'s saliency algorithm [78] on the *LabelMe* dataset [151]. *LabelMe* is an image dataset that contains over 24,000 natural images with over 74,000 objects that have been annotated manually by the computer science community. Annotations are traces of the objects themselves rather than bounding boxes. The purpose of this dataset (and online annotating tools) was initially to collect information about objects in natural scenes for object recognition algorithms. Elazary and Itti used the bottom-up saliency algorithm on these images and found that in 75% of images one or more of the top three salient locations fell on an outlined object. If bottom-up saliency is able to locate salient regions with such accuracy on images with multiple salient objects, this should be true more so for images with only a single salient object (as is the case for images used in the experiments performed for this chapter). This is because, generally speaking, bottom-up saliency will detect salient objects as long as these objects truly are 'salient', i.e. prominent and not camouflaged. These objects will naturally 'pop out' from the background and this effect is what bottom-up saliency attempts to detect.

When using saliency in CBIR, a common approach among researchers is to find salient regions and then utilise feature point locating techniques on these regions only. This was, for example, the strategy used by Rutishauser et al. [152] and Walther et al. [170]. They chose to employ an additional segmentation step in their calculations by extracting objects from salient regions. Features were then located from these regions, i.e. from the extracted objects, rather than the regions defined by visual saliency algorithms. This segmentation step, however, has not always performed well in practice and still requires further work [91].

No one has yet shown that bottom-up saliency can be used to optimise feature point extraction in CBIR without the need for any other processing steps (e.g. segmentation or object detection). Nor has anyone analysed what the effect is of changing the saliency measure to decrease the size of extracted regions on classification results.

In this chapter, we chose to use the GBVS algorithm [68] for saliency calculations because it has been shown to perform consistently well over numerous datasets [21]. The Speeded Up Robust Features (SURF) feature point algorithm developed by Bay in [14] is used to locate and describe feature points in this chapter. This feature detection method was selected due to its robustness, memory efficiency in describing feature points and superior speed when compared to other popular feature detection methods such as the Scale Invariant Feature Transform (SIFT) method [111, 86]. The standard classification technique Support Vector Machines (SVM) is used to perform the classification on the selected SURF features using saliency information.

		Correctly classified	Incorrectly classified
Horses	$\mathbf{C}$	84.5%	15.5%
	$\bar{\mathbf{C}}$	98.7%	1.3%
Flowers	$\mathbf{C}$	80.1%	19.9%
	$\bar{\mathbf{C}}$	97.8%	2.2%
Faces	$\mathbf{C}$	90.3%	9.7%
	$\bar{\mathbf{C}}$	99.9%	0.1%

Table 3.1: Percentage of training features correctly and incorrectly classified by the SVM.

We present next in more detail our basic CBIR system and the experiments done to assess the usefulness of saliency for this application.

3.2 CBIR Design

A dataset of three image classes was collected: horses, yellow flowers, and faces. The horses dataset was obtained from the INRIA Horses V1.03 dataset [89]. The yellow flowers dataset was compiled from images from the Visual Geometry Group's (The University of Oxford) set [130]. The faces dataset was a combination of the 'face94' Essex face database [1], the MIT CBCL Face dataset [177] and the Caltech101 dataset [50]. All faces chosen from the dataset were of head-on shots and each dataset was used separately. The three image classes were selected because they are a good representation of objects with differing colour, texture and context.

The first step in the experiment was to train a two-class SVM. For each dataset, 100 positive images were cropped to only include the object of interest: horse, yellow flower or face. These images and 100 negative images (taken from the negative images folder of the INRIA Horse V1.03 dataset) were then scanned for SURF feature points. All the calculated feature points were then fed into the SVM. Classification results of the SVM for the training features are shown in Table 3.1, with $\mathbf{C}$ representing feature points correctly classified in the positive group and $\bar{\mathbf{C}}$ feature points correctly classified in the negative group.

The GBVS algorithm gives a saliency value between 0 and 1 for each pixel from an image, where 0 signifies no saliency, and 1 indicates the highest saliency value possible. Entire regions can have saliency values of 1 and the distribution of saliency values can vary depending on each image.

A different subset of images to the training set was used for testing. For each object of interest (horse, flower, face), 40 positive (**P**) and 40 negative (**N**) images were tested. The objects in the positive image group cover approximately 25% to 35% of the total size of the image (contrary to the images used to train the SVM classifiers, the test images of the objects of interest were not cropped). Some examples of the images in **P** can be seen in Figure 3.3.

SURF features are computed for these test images; however, only the features associated with a saliency above a threshold **T** are fed into the classifier to try to find the images with the object of interest.

The following saliency thresholds were used in our experiment: **T** = 0 (no saliency information used), 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9. The higher the value of **T**, the fewer the number of feature points fed into the SVM. Figure 3.1 shows the training and testing flowchart of the experiment.

3.3 Performance Using Visual Saliency

The test set of positive images for each object of interest was treated as one large image by collecting all the feature points found in all images into one pool. The pool of features from the positive images were run separately in the SVM to the pool of features from the negative images. For each threshold we compute the following proportion:

$$p_p = \frac{a_p}{n_p}\ , \qquad p_n = \frac{a_n}{n_n} \tag{3.1}$$

where a_p is the number of feature points in **P** classified as **C**, a_n is number of feature points in **N** classified as $\bar{\mathbf{C}}$, n_p is the total number of feature points detected in **P** and n_n is the total number of feature points detected in **N**. The confusion matrix corresponds to:

	C	$\bar{\mathbf{C}}$
P	p_p	1-p_p
N	1-p_n	p_n

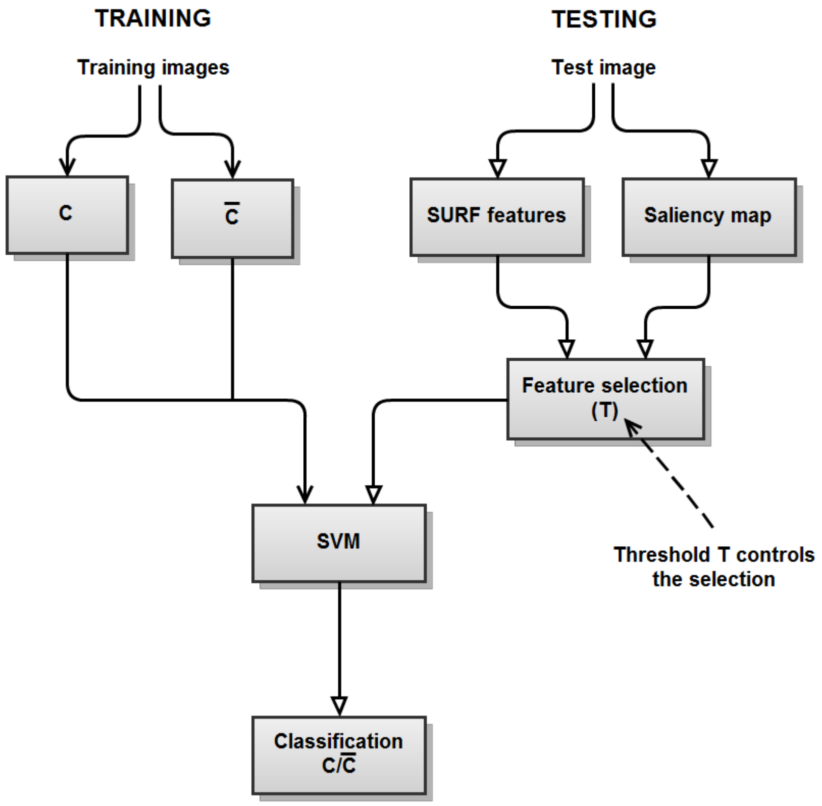


Figure 3.1: Flowchart of the training and testing phases of the experiment

A standard error can be computed to measure the uncertainty associated with the proportions p_p and p_n such that:

$$SE(p) = \sqrt{p(1-p)/n} \tag{3.2}$$

where p is p_p (resp. p_n) and n is n_p (resp. n_n) and the assumption that the data has a normal distribution. The proportion p_p and $1 - p_n$ for each dataset was computed for each threshold value **T** with its corresponding standard error appearing as error bars. If selecting features using saliency can reduce computation time, we want to check that the retrieval result will not deteriorate as fewer features are considered for classification.

3.3.1 Classification Results

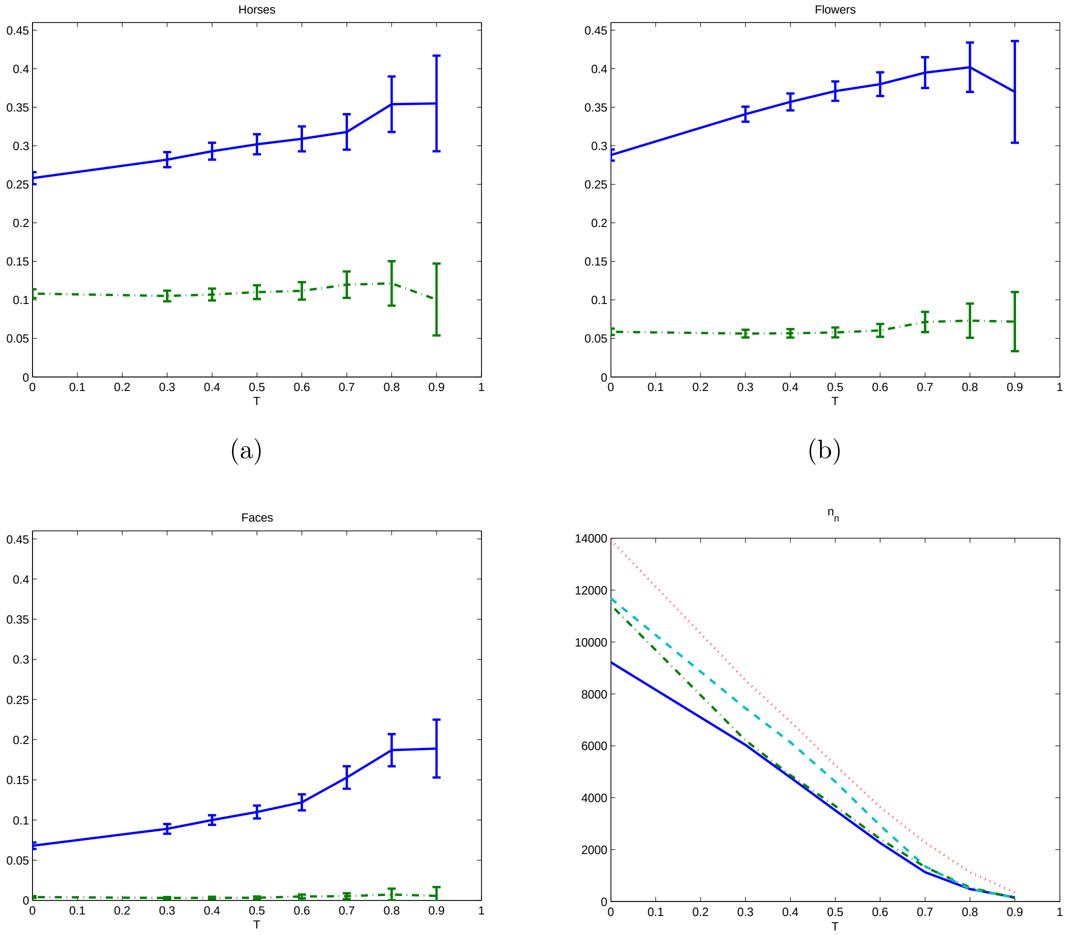
Figure 3.2 (a), (b) and (c) show these plots for the horses, yellow flowers and faces datasets respectively. The positive classifications p_p improves as the saliency threshold increases for all objects (with $0 \leq \mathbf{T} \leq 0.8$). The improvement with saliency ($\mathbf{T}=0.8$) versus without saliency ($\mathbf{T}$=0) is of 10% for Horses, 12% for Flowers, and 12% for Faces. This seems to indicate that the feature points that are left out for classification as $\mathbf{T}$ increases are more often from $\bar{\mathbf{C}}$ (background) than the class of interest $\mathbf{C}$ (horse, flower, face). On the other hand, the (false positive) proportion $1 - p_n$ measuring the proportion of points classified in $\mathbf{C}$ in the negative set of images (where no object of interest appears) remains more or less constant whatever the saliency level chosen.

This first result indicates that selecting features using saliency will not deteriorate classification performances but can in fact improve them.

As the saliency threshold increases, the standard errors for the proportions also increase. This is due to the fact that the number of selected features, n_p and n_n, decrease while the proportions, p_p and p_n, increase (and hence so does $p(1-p)$) as $\mathbf{T}$ increases (see Equation 3.2). Therefore, although p_p generally improves as the threshold increases, the certainty about how much the proportion actually improves is reduced. It can also be noticed that the standard errors for faces are smaller at each value of $\mathbf{T}$ compared to the other two classes. The reason for this is that more feature points were being detected in this class of images as opposed to flower and horse images (cf. Figure 3.2 (d)). Most of the face images were taken with relatively 'busy' backgrounds that foster bountiful amounts of feature point detection. The difference in the number of feature points detected in the three classes can also explain the low p_p values obtained for the faces dataset. More background points will bring down the value of p_p.

It should also be noted that variations in numbers of feature points can occur in images in the same class and not just across classes. This can be noticed in Figure 3.3. The yellow flower, for example, on the left has significantly less feature points than the flower on the right.

Figure 3.3 shows some results of the feature classification on images of the positive classes for two saliency thresholds. Although the classifier (SVM) used is not optimal (some points on the object are misclassified as $\bar{\mathbf{C}}$ while some points on the background are classifed as $\mathbf{C}$), the objects of interest are well found by many feature points even



(a)

(b)

(c)

(d)

Figure 3.2: (a): proportion of correct classification for the horses dataset p_p (solid blue line) and $1 - p_n$ (green dashdot line) w.r.t. $\mathbf{T}$; (b) proportion of correct classification for the flowers dataset p_p (solid blue line) and $1 - p_n$ (green dashdot line) w.r.t. $\mathbf{T}$; (c) proportion of correct classification for the faces dataset p_p (solid blue line) and $1 - p_n$ (green dashdot line) w.r.t. $\mathbf{T}$; (d) total no. of feature points n_p for horses (solid blue line), flowers (green dashdot line) and faces (dotted red line) and n_n (aqua dashed line).

as the saliency threshold increases.

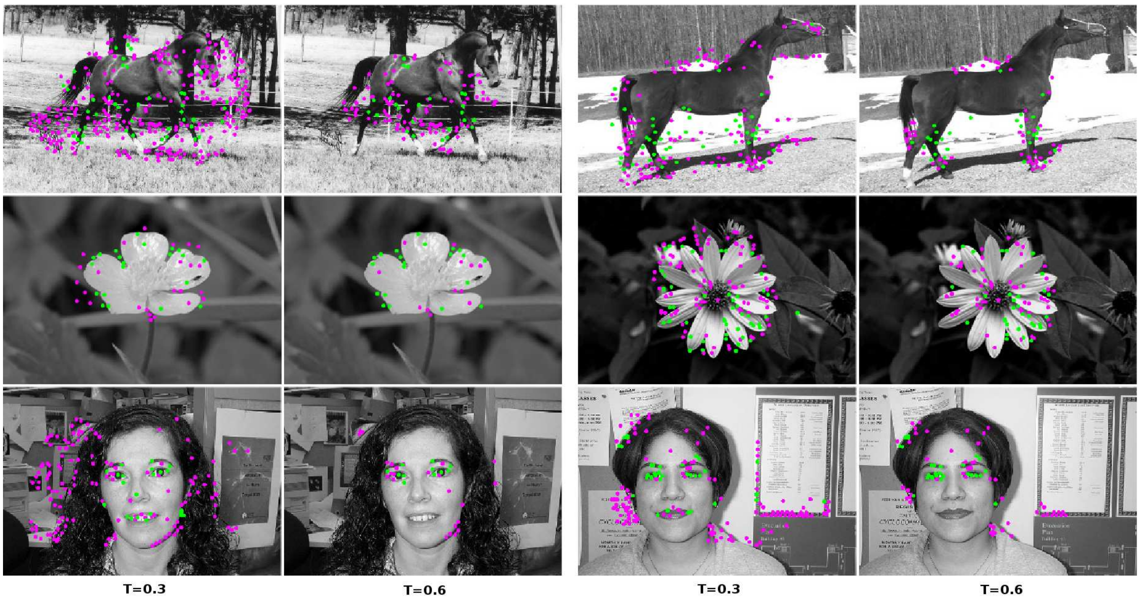


Figure 3.3: Example results for $\mathbf{T} = 0.3$ and 0.6. From top to bottom: horses, flowers and faces. Green dots indicate feature points classified in the image class ($\mathbf{C}$), purple dots indicate feature points classified outside of it ($\bar{\mathbf{C}}$).

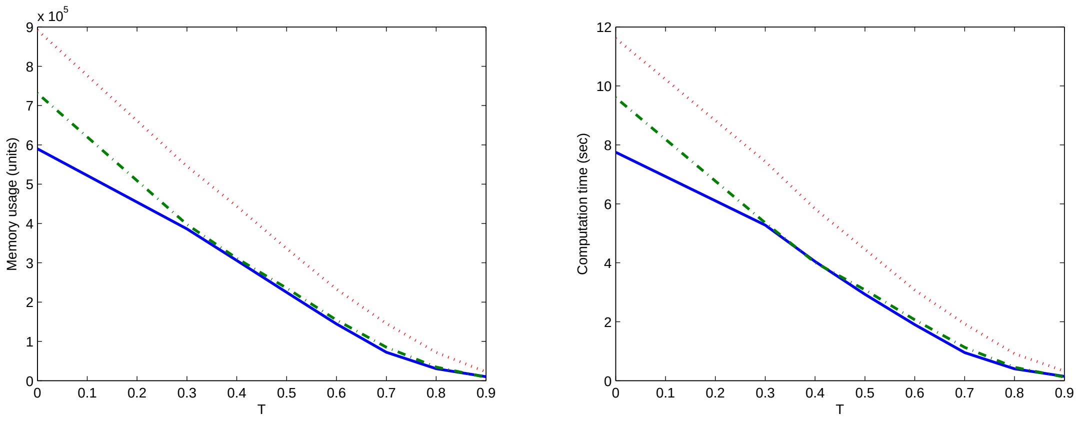
3.3.2 Disk Space Usage Results and Discussion

Another aspect of this experiment was to see what effect changing the value of $\mathbf{T}$ would have on disk space usage and classification time. Figure 3.4 (a) shows a plot of the memory usage required for storing feature points in memory for the three classes of images. A SURF feature point is a 64 element vector of floating point numbers. Since the size of doubles stored in memory is machine dependent, a single unit of memory usage was used for a double to abstract over this machine dependence (1 double = 1 unit, 1 SURF feature point = 64 units).

Figure 3.4 (a) depicts a clear downward trend for memory usage for all three classes. This trend is linear until $\mathbf{T} = 0.7$ when the rate of change decreases. At $\mathbf{T} = 0.5$, for example, a 60% decrease in required memory usage is obtained, while a 75% result is obtained at threshold value 0.6.

Figure 3.4 (b) shows the computation time that was clocked for classifying the images in the three image classes. These results were obtained on a 2.67 GHz Intel

Core2 Quad CPU running Windows 7 with 4 GB of RAM. The classifying application (libsvm [35]) was used in Matlab.



(a)

(b)

Figure 3.4: (a) Memory usage for positive images w.r.t. **T** for horses (solid blue line), flowers (green dashdot line) and faces (red dotted line); (b) Classification computation time for positive images w.r.t. **T** for horses (solid blue line), flowers (green dashdot line) and faces (red dotted line).

The computation time graph is very similar to the memory usage graph of Figure 3.4 (a). One would expect this as the classification computation time is directly proportional to the number of features stored in memory, assuming all the feature points can be easily stored in RAM. An approximate speed-up of 60% and 75% was recorded for the **T** values of 0.5 and 0.6 respectively, which is exactly the same level of improvement as recorded for memory usage.

3.4 Conclusion

This chapter showed that visual saliency can be used to help in efficiently selecting visual features in a retrieval system. Selection of features using visual saliency reduces computation time and memory consumption needed for visual feature storage. The saliency threshold can be tuned efficiently to get the best retrieval performance for both accuracy and speed, and it can be used to scale the system to different devices.

Chapter 4

Extension of GBVS to 3D Media

Most of visual saliency (VS) research has been focussing on the visual perception of images and videos displayed on 2D screens. More recently, however, it has been shown that humans look differently at images displayed in 3D compared to 2D [82, 172] and subsequently a few algorithms for 3D saliency have been proposed [24, 173]. This chapter investigates the extension of visual saliency algorithms (VSAs) to media displayed in 3D. The Graph-Based Visual Saliency (GBVS) [68] algorithm is assessed for 3D images and then a new 3D GBVS algorithm is proposed. Finally, experimental results are presented that show that the 2D and proposed 3D GBVS algorithms outperform other common 2D algorithms as well as their state-of-the-art 3D extensions on 3D media.

4.1 State of the Art

The following sections will present the Graph-Based Visual Saliency algorithm and discuss the current state of the art for 3D VSAs and depth incorporation in VSAs.

4.1.1 The Graph-Based Visual Saliency Algorithm

Proposed by Harel et al. [68], the GBVS algorithm is a 2D VSA that uses a Markovian approach to calculate its saliency maps. The first step in this algorithm is to break up the input image into the following feature channels: colour, intensity and orientation (similarly to Itti et al. [78]). Orientation is calculated by using the Gabor filter at 0 and 90 degrees. Intensity is the grey-scale version of the image calculated by eliminating the hue and saturation information while retaining the luminance with the following equation: $0.2989 * R + 0.5870 * G + 0.1140 * B$, where R, G and B and the red, green and blue colour channels respectively. Colour is treated as two channels: Blue-Yellow (calculated as $abs(B - min(R, G))$, and Red-Green (calculated as $abs(R - G)$). Salient regions are then located in each of these channels by computing:

$$d((i,j)||(p,q)) = \left|\log \frac{M(i,j)}{M(p,q)}\right| \tag{4.1}$$

where $M(i, j)$ is the value of the pixel (i, j) in the feature map M (i.e. in the feature channels of colour, intensity or orientation). Following this, a fully connected graph G_A is created by connecting every node (a.k.a. points or vertices - these represent pixels in this case) with all other nodes in each M. The edge (a.k.a. arc or line) of each node connection from (i, j) to (p, q) is assigned the following weight:

$$w((i,j),(p,q)) = d((i,j)||(p,q)) \times F(i-p, j-q) \tag{4.2}$$

where

$$F(a,b) = \exp\left(-\frac{a^2+b^2}{2\sigma^2}\right) \tag{4.3}$$

and σ is a free parameter set to approximatively one tenth to one fifth of the image width. All the weights, w, are, hence, proportional to their dissimilarity and distance in M. A Markov chain is then created over G_A. A Markov chain is defined as a system with a set of states $S = \{s_1, s_2, ..., s_n\}$ that undergoes transitions from one state to another. If the chain is currently in state s_i then it moves to state s_j according to probability p_{ij}. The system is memoryless meaning that the next state depends only on the current state and not on any previous sequences of states. Probabilities p_{ij} are called *transition probabilities.* The equilibrium (stationary) distribution, π, is the probability of being in the various states after n steps and is the same on all steps. It is defined as:

$$\sum_{i \in S} \pi(i)P(i,j) = \pi(j), j \in S, \tag{4.4}$$

where P is the transition function and $\pi(i)$ is a probability distribution. This condition is usually written as $\pi P^n = \pi$. Any ergodic Markov chain (i.e. any chain in which it is possible to eventually get from one state to any other state with positive probability) has a unique π that is strictly positive.

In the Markov ergodic chains defined by the GBVS algorithm nodes are treated as states and edge weights as transition probabilities. The equilibrium distribution π of the Markov chain calculated for each G_A reflects the time a random walker would spend at each node. Higher values are obtained for nodes with higher dissimilarities with their surrounding nodes. This is because it is more likely to transition into subgraphs with lower similarity measures. The final saliency map is calculated by linearly pooling all equilibrium distributions.

This algorithm has been stated as being biologically plausible, meaning that its method for calculating saliency is similar to the way our human visual system works. In our brain neurons are connected in a retinotopically organised network called the visual cortex. The neurons communicate with each other via synaptic firing in a way which produces emergent behaviour including the flagging of areas of a scene that require additional processing. The individual nodes in the GBVS algorithm act in a similar way to neurons in the visual cortex, i.e. nodes also communicate in a connected network to detect regions of interest. Moreover, computations at each node or region can also be done completely in parallel [68].

4.1.2 3D Visual Saliency Algorithms

Stereoscopic images displayed on 3D screens allow us to immediately perceive depth information [129]. Incorporating depth or disparity information in the calculation of saliency is therefore a natural extension to VSAs for automatically analysing 3D content. Three strategies have been proposed to use depth information [172]: the depth-weighting model, depth-saliency model and the stereo-vision model. The following sections will describe each strategy and all 3D VSAs pertaining to it. The depth maps used in these models can be obtained from either depth detecting apparatuses (e.g. the Microsoft Kinect) or calculated with depth estimating algorithms.

The depth-weighting model

The depth-weighting model weights 2D saliency computations with a corresponding depth map. That is, every pixel (or target, region, etc.) is directly related to its depth and no features are searched for or extracted from the depth map.

Examples of VSAs belonging to this strategy of depth incorporation are Chamaret et al. [34] and Maki et al. [117]. Chamaret et al. have proposed to multiply the saliency map computed with any 2D VSAs by the inverse of the depth map. Regions appearing closer to the viewer are then made more salient. Indeed, close areas have been shown to be viewed more often than regions further away [172]. These regions, hence, should be deemed more salient. Chamaret et al. then used this saliency calculation to refine a single region-of-interest that was selected earlier through a nearest-neighbour filtering and thresholding approach.

Maki et al. propose to assign the closest target region with highest priority. They locate this region by histogramming the depth map and creating a target mask by back propagating a selected depth range. The target mask is then used as a depth cue and incorporated into saliency calculations.

The depth-saliency model

The second strategy is the depth-saliency model that creates a depth saliency map (DSM) by first looking for features in the depth map and then linearly pooling this with 2D VSA computations.

A number of depth incorporating strategies belonging to this group include Ouerhani and Hugli [135], Potapova et al. [144] and Wang et al. [173]. Ouerhani and Hugli [135] propose a standard depth cue integration saliency algorithm based on the Koch and Ullman model [94] (see Section 2.2.1). Alongside the colour, intensity and orientation cues, depth information is also analysed in a similar fashion. Saliency output generated by all cues is combined equally or with different weights into a master saliency map. This solution, as it is based on the Koch and Ullman model, is biologically plausible.

In 2011, Potapova et al. [144] published a paper on saliency in robotics. In it they examined three different 3D cues for the task of detecting grasp points on objects for robots. These cues were: surface height (SH), relative surface orientation (RSO) and occluded edges (OE). They combined these cues with standard 2D visual cues (colour, orientation and intensity) to build a probability map. A probability map is a map that provides probabilities of detecting salient features.

Wang et al.'s [173] DSM calculations include a Difference of Gaussian (DoG) filtering step over the depth map and then correlation of the contrast map with the degree of depth saliency through the use of results obtained from a psychophysical experiment of theirs. The DSM is then either added or multiplied to the corresponding 2D saliency values (2D+DSM and 2DxDSM respectively).

The stereo-vision model

The third depth incorporating VSA strategy is the stereo-vision model. This strategy attempts to directly model the stereoscopic mechanisms in the human visual system. Bruce and Tsotsos [25] present the only VSA that belongs to this group. They extend their 2D VSA [166] that uses a visual pyramid processing architecture by adding additional neurons to model stereo vision [25].

Most 3D VSAs belong to the first and second categories. Figure 4.1 shows a diagram that summarises how they operate. The first model is simple to incorporate to existing 2D VSAs and is computationally less demanding than the second model. However, salient regions in more complex 3D scenes may be missed by the first model that the second model could detect due to it directly searching for depth features.

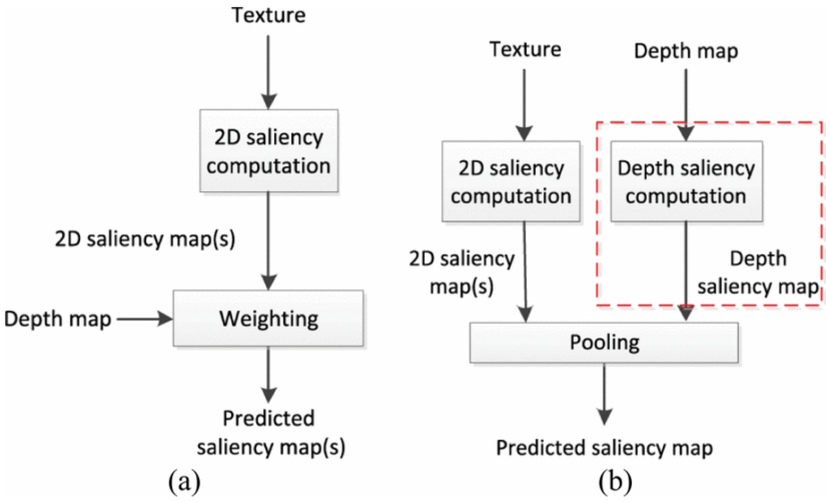


Figure 4.1: Two depth incorporating strategies: (a) the depth-weighting model; (b) the depth-saliency model [172].

4.1.3 Assessment of Depth Incorporating Algorithms

Apart from Wang et al. [173] and Potapova et al. [144], none of the depth incorporating strategies above have had any quantitative validation performed with eye tracking data. Moreover, none of these strategies have been compared to each other. Wang et al. performed quantitative experiments to remedy this. They showed that the best algorithms for incorporing depth information belong in the second strategy and that optimal results were obtained when DSMs were added to 2D VSA saliency maps. They also confirmed quantitatively that for 3D images 2D Bruce outperforms 2D Hou (and that 2D Hou outperforms 2D Itti) and this ordering remains true in 3D when adding the DSM. These algorithms proposed by Wang et al. are denoted 3D Itti, 3D Hou and 3D Bruce. Adding a DSM provided the best results but was the most computationaly demanding. Alternatively, Chamaret's method to include depth information performed poorly but was the most computationally efficient.

4.2 GBVS Extensions to 3D Media

The GBVS algorithm has never been extended to 3D and we propose to incorporate depth information in three ways: using Chamaret et al.'s approach [34] (denoted 3D GBVS (Chamaret)), using the DSM as proposed by Wang et al. [173] (denoted 3D GBVS (Wang)), and using our own approach (denoted 3D GBVS (our approach)).

This last method (3D GBVS (our approach)) involves a three-step process: selecting a low or high scaling factor corresponding to depth values, restriction of the depth-range that saliency values are affected by this scaling factor, and then the subsequent scaling of these saliency values. The first step entails calculating the median and mean values of the depth map. If the median is smaller than the mean, a higher scaling factor is chosen (SF in eqs 4.6 and 4.7) and vice versa otherwise. The idea behind this is to detect images that have unique objects in the foreground and hence will stand out on their own in 3D. A smaller median value compared to the mean would provide an indication of this. This method is a fast way of detecting features in the depth map. The second step is performed by retaining only the depth values in the depth map that have a 2D saliency value over a chosen saliency threshold ST (saliency values are scaled between 0 and 1, where 1 indicates very salient and 0 indicates no saliency).

We denote these selected depth and 2D GBVS saliency values $\{(d_i, s_i)\}$ (i is the pixel index in the map) and find the range of these values by computing:

$$\begin{cases} l = \min_i\{d_i\} & \text{(lower bound)} \\ h = \max_i\{d_i\} & \text{(upper bound)} \end{cases} \tag{4.5}$$

and the middle of the interval is then defined by $dm = \frac{h+l}{2}$. For the pixel i, we compute

- If $d_i < dm$ then
$$S_i = \frac{s_i}{1 + SF \times \frac{dm-l}{dm-d_i}} \tag{4.6}$$

- If $d_i > dm$ then
$$S_i = s_i \left(1 + SF \times \frac{h - dm}{d_i - dm}\right) \tag{4.7}$$

where SF is the chosen scaling factor (high or low from step 1), S_i is the new 3D GBVS value, and s_i is the 2D GBVS value. We have set $SF = 0.35$ (high SF) or $SF = 0.1$ (low SF) and $ST = 0.4$ in our experiments. These threshold values were chosen after an exhaustive search. The justification behind the last two steps is that depth information should only be used on areas with high-enough saliency and should not be 'wasted' elsewhere. If salient regions are only present in the foreground, the competition for scaling should only take place there.

4.3 Experimental Results

We used the eye tracking database supplied by Wang et al. [173] as ground truth. This database contains 18 stereoscopic images, eye tracking data obtained from 35 human subjects, corresponding depth and disparity maps and eye fixation density maps. This database serves as a ground truth and is noted 3D VS because the stereoscopic images were displayed on a 3D screen (as opposed to 2D VS ground truth that collects eye tracking data with the images displayed on a 2D screen). In our experiments, we used the Pearson Linear Correlation Coefficient (PLCC) [126, 114] and Kullback-Leibler divergence (KLD) [100, 126] to measure the performance of VSAs. These were also used in Wang et al.'s study [173].

4.3.1 Evaluation of 2D VSAs Against 3D VS

In Table 4.1 we compare the 2D GBVS algorithm on this dataset with Itti's, Hou's and Bruce's algorithms using the PLCC (defined in Eq. (2.8)) and KLD (Eq. (2.7)) metrics. Note that the 2D GBVS algorithm was not analysed by Wang et al. [173] but they did compute the PLCC and KLD for 2D Itti, 2D Hou and 2D Bruce. Our numerical results differ slightly from theirs due to rounding errors and different image scaling algorithms. Our results in Table 4.1 confirm Wang et al.'s assessment for 2D Itti, 2D Hou and 2D Bruce and our contribution in Table 4.1 is to show that the 2D GBVS algorithm outperforms all other 2D VSAs by far.

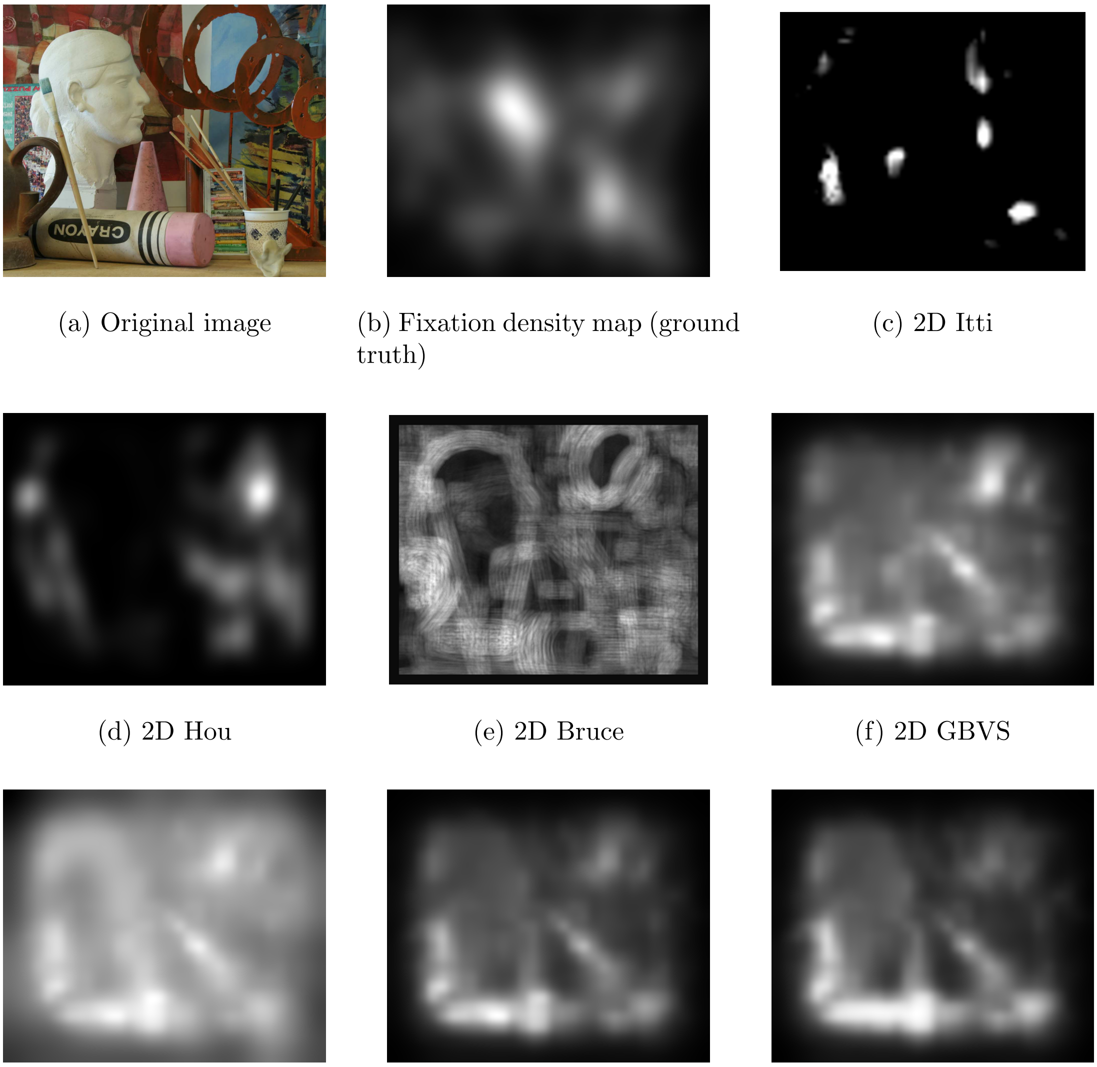
	PLCC	KLD	Yr of publication
2D Itti	0.154	2.781	1998 [78]
2D Hou	0.299	0.877	2007 [74]
2D Bruce	0.346	0.704	2009 [24]
2D GBVS	**0.589**	**0.314**	**2007** [68]
UTPL [173]	0.897	0.127	

Table 4.1: Performances of 2D VSAs. Higher PLCC and lower KLD values indicate better performances.

The Upper Theoretical Performance Limit (UTPL) [157] computed by Wang et al. [173] is also reported for both PLCC and KLD. The UTPL is commonly used as a benchmark for 2D visual saliency models, and 2D GBVS is halfway in performance between this theoretical limit and the best 2D VSA previously reported (2D Bruce). Figures 4.5 and 4.3 (c)-(f) shows some saliency maps for these 2D VSAs.

4.3.2 Evaluation of 3D VSAs Against 3D VS

Our 3D VSAs are also evaluated on the same dataset. Wang et al. found their proposed 2D+DSM approach to be the best way to incorporate depth in saliency calculations, and this method is referred to as (Wang) in Table 4.2. We implemented their approach for all the four 2D VSAs as well as Chamaret's and our own proposed method with the GBVS algorithm. Table 4.2 confirms the results for Itti (Wang), Bruce (Wang) and Hou (Wang) as reported by Wang et al. [173].



(a) Original image

(b) Fixation density map (ground truth)

(c) 2D Itti

(d) 2D Hou

(e) 2D Bruce

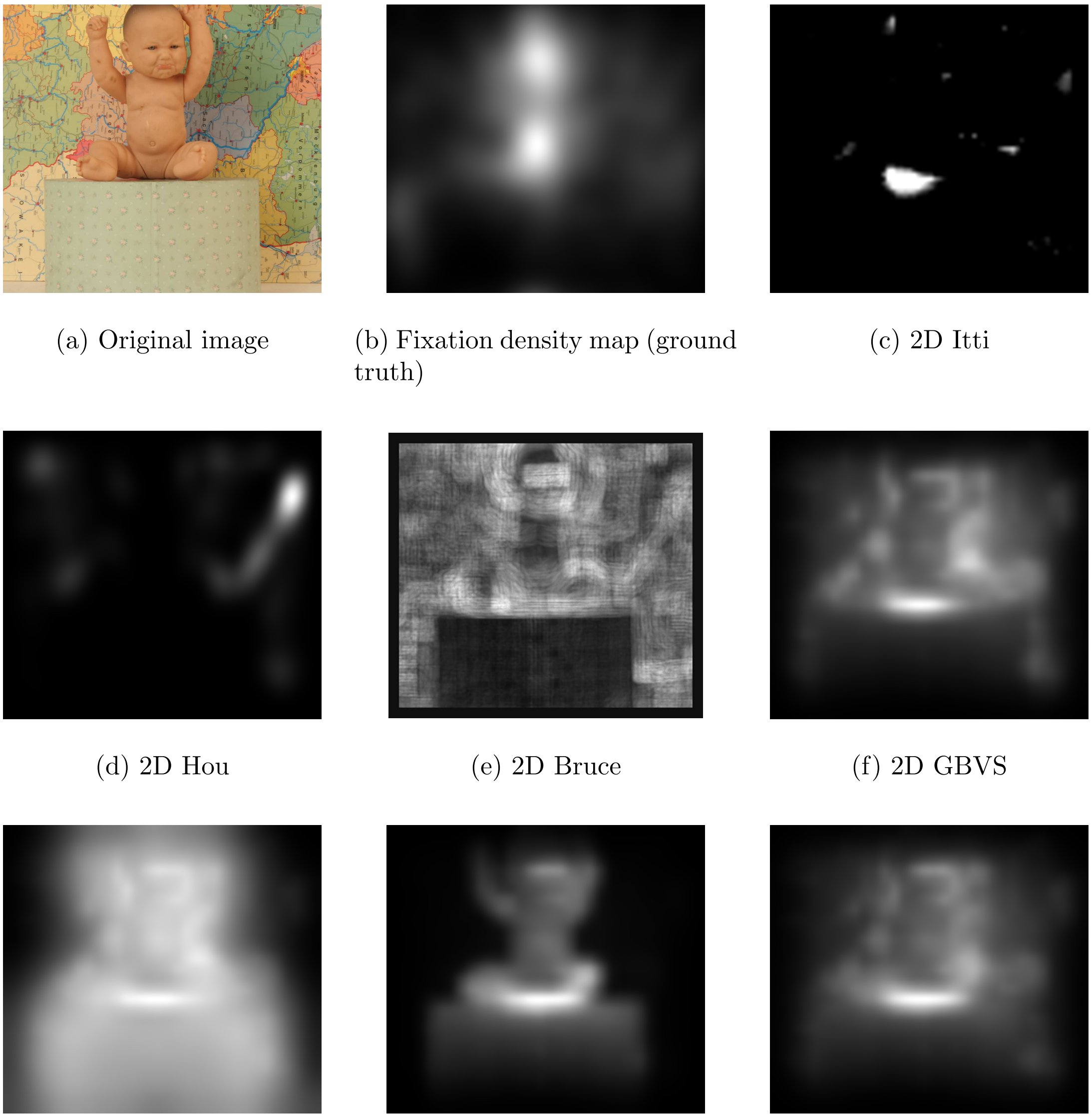
(f) 2D GBVS

(g) 3D GBVS (Wang)

(h) 3D GBVS (Chamaret)

(i) 3D GBVS (Our method)

Figure 4.2: Saliency maps with VSAs. (a) Original image #1 [173], (b) Corresponding fixation density map, (c)-(f) Saliency predictions from the four 2D VSAs: Itti, Hou, Bruce and GBVS, (g)-(i) Saliency predictions from the three 3D VSAs: 3D GBVS (Wang), 3D GBVS (Chamaret) and 3D GBVS (Our method)


(a) Original image

(b) Fixation density map (ground truth)

(c) 2D Itti

(d) 2D Hou

(e) 2D Bruce

(f) 2D GBVS

(g) 3D GBVS (Wang)

(h) 3D GBVS (Chamaret)

(i) 3D GBVS (Our method)

Figure 4.3: Saliency maps with VSAs. (a) Original image #2 [173], (b) Corresponding fixation density map, (c)-(f) Saliency predictions from the four 2D VSAs: Itti, Hou, Bruce and GBVS, (g)-(i) Saliency predictions from the three 3D VSAs: 3D GBVS (Wang), 3D GBVS (Chamaret) and 3D GBVS (Our method)

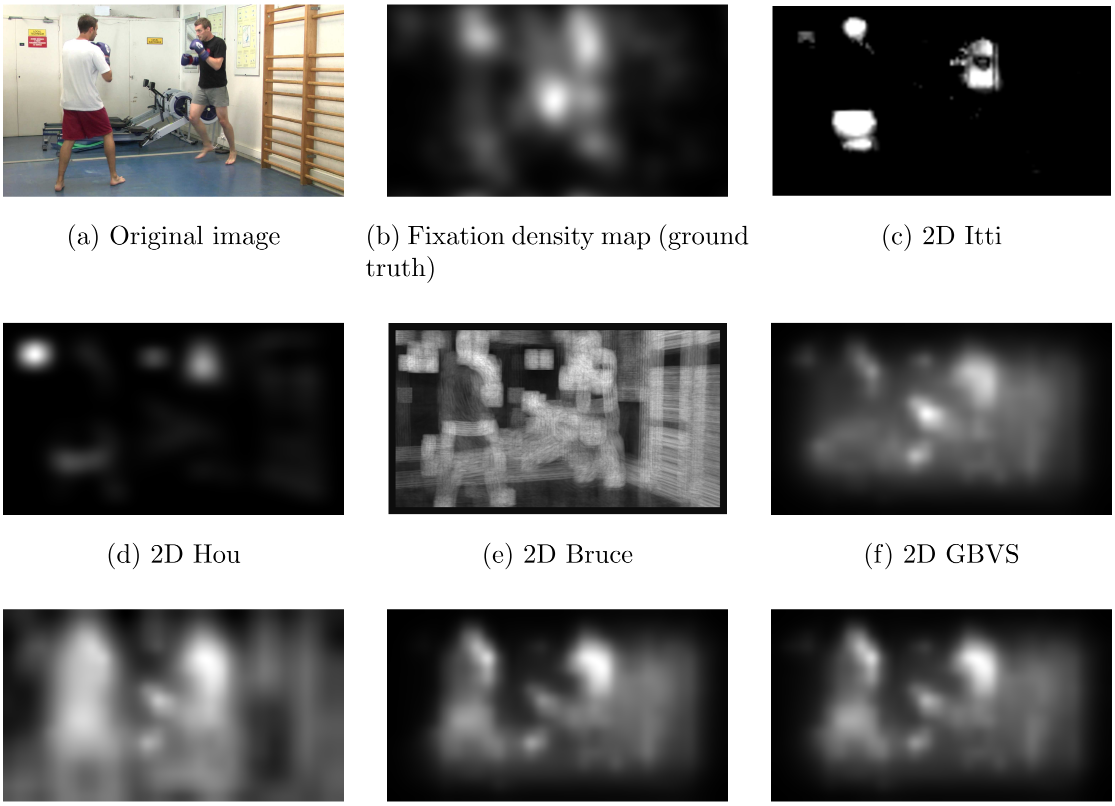
Our contribution in Table 4.2 is to show that the three 3D VSAs we proposed outperform all 3D VSAs proposed by Wang et al [173]. Surprisingly, Chamaret's method in the GBVS fared better than Wang's algorithm (that is also the most computationally demanding method). Note however that 3D GBVS (Chamaret) and 3D GBVS (Wang) are outperformed by 2D GBVS (cf. Table 4.1). Our simple and fast proposed method for incorporating depth obtained the best results. Figure 4.5 (f)-(i) shows example saliency map results for 2D GBVS, 3D GBVS (Wang), 3D GBVS (Chamaret) and 3D GBVS (Our method). An improvement on 2D GBVS can be seen in 3D GBVS (Chamaret) and 3D GBVS (Our method) - both are closer to the ground truth.

3D Saliency Algorithms	PLCC	KLD	Yr of pub.
3D GBVS (Chamaret)	0.573	0.379	
3D GBVS (Our method)	**0.602**	**0.306**	
3D GBVS (Wang)	0.561	0.484	
3D Itti (Wang)	0.364	0.627	2013 [173]
3D Bruce (Wang)	0.419	0.657	2013 [173]
3D Hou (Wang)	0.436	0.558	2013 [173]

Table 4.2: Performances of 3D-VSAs. Higher PLCC and lower KLD values indicate better performance.

Table 4.3 shows the results for each image in the database for 2D GBVS, 3D GBVS (Chamaret) and 3D GBVS (Our method). On average, 3D GBVS (Our method) improves 2D GBVS (significant improvement for the PLCC and KLD as shown by a paired t-test with $p<0.1$).

There were, however, some images in the dataset for which our depth incorporation method was detrimental to the final saliency calculations. Because our method is partly based on the depth-weighting model it operates under the assumption that closer objects are more salient. There are scenarios where this may not be the case, e.g. a long bare hallway with a person standing at the end of it. Two examples from the dataset fitting into this category can be seen in Figures 4.2 and 4.4. For these two images the 3D GBVS obtained worse results than 2D GBVS (cf. Table 4.3). In Figure 4.4 the rowing machines in the background have saliency values in the ground truth similar to the values of the two boxers. Our depth incorporation method chooses to increase the saliency values of the people at the expense of the rowing machines.



(a) Original image

(b) Fixation density map (ground truth)

(c) 2D Itti

(d) 2D Hou

(e) 2D Bruce

(f) 2D GBVS

(g) 3D GBVS (Wang)

(h) 3D GBVS (Chamaret)

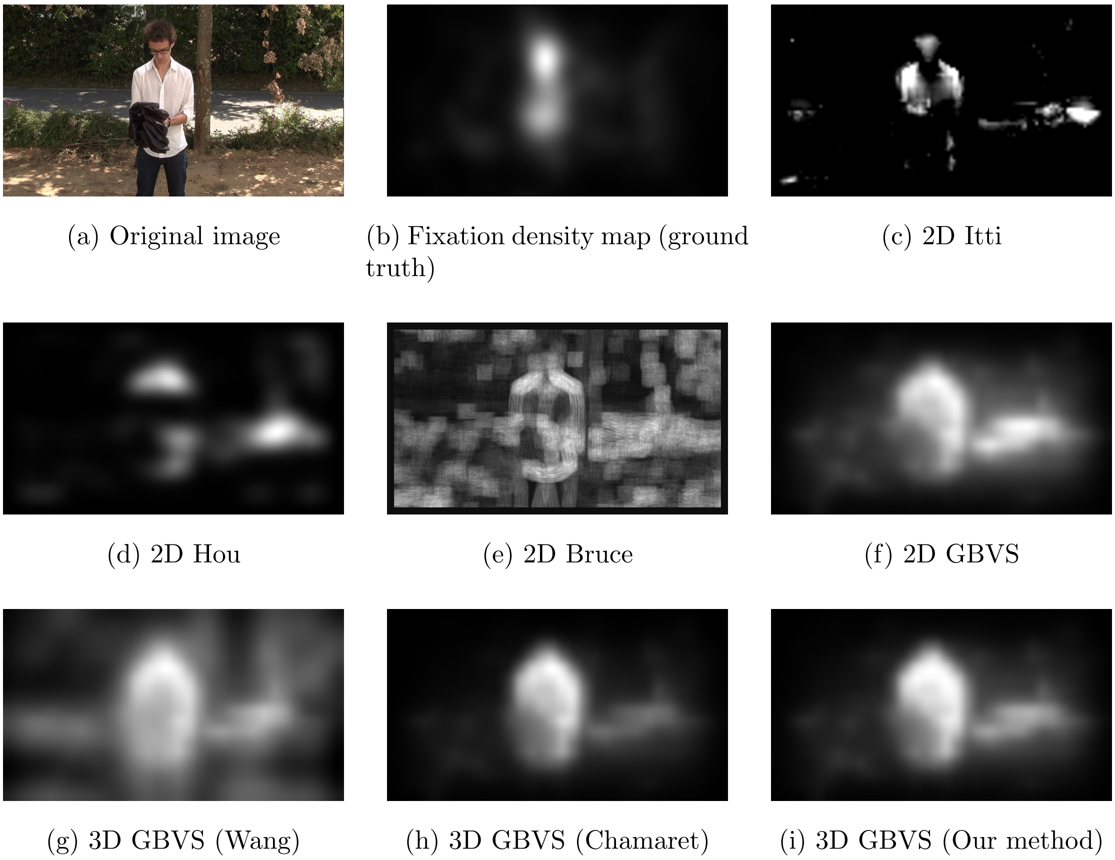
(i) 3D GBVS (Our method)

Figure 4.4: Saliency maps with VSAs. (a) Original image #11 [173], (b) Corresponding fixation density map, (c)-(f) Saliency predictions from the four 2D VSAs: Itti, Hou, Bruce and GBVS, (g)-(i) Saliency predictions from the three 3D VSAs: 3D GBVS (Wang), 3D GBVS (Chamaret) and 3D GBVS (Our method)

Similarly in Figure 4.2 the model of the head is the most salient object. Our 3D GBVS calculation emphasised the pink crayon more (which is closer) than the head model.

4.4 Conclusion

In this chapter we demonstrated that the GBVS algorithm is greatly superior in predicting fixations in 3D compared to other state-of-the-art algorithms. We also showed that our simple and fast extension to Chamaret's method of depth incorporation outperforms all other depth incorporation methods analysed by Wang et al. We showed



(a) Original image

(b) Fixation density map (ground truth)

(c) 2D Itti

(d) 2D Hou

(e) 2D Bruce

(f) 2D GBVS

(g) 3D GBVS (Wang)

(h) 3D GBVS (Chamaret)

(i) 3D GBVS (Our method)

Figure 4.5: Saliency maps with VSAs. (a) Original image #18 [173], (b) Corresponding fixation density map, (c)-(f) Saliency predictions from the four 2D VSAs: Itti, Hou, Bruce and GBVS, (g)-(i) Saliency predictions from the three 3D VSAs: 3D GBVS (Wang), 3D GBVS (Chamaret) and 3D GBVS (Our method)

	GBVS		GBVS+Chamaret		GBVS+Proposed	
Img. #	PLCC	KLD	PLCC	KLD	PLCC	KLD
1	0.540	0.188	0.488	0.217	0.489	0.216
2	0.636	0.206	0.623	0.454	0.678	0.203
3	0.490	0.300	0.496	0.401	0.520	0.295
4	0.477	0.226	0.496	0.284	0.503	0.226
5	0.612	0.212	0.665	0.199	0.643	0.200
6	0.703	0.236	0.669	0.314	0.688	0.262
7	0.608	0.235	0.650	0.246	0.642	0.221
8	0.516	0.693	0.454	0.820	0.497	0.741
9	0.564	0.263	0.463	0.286	0.560	0.237
10	0.552	0.190	0.577	0.172	0.573	0.175
11	0.599	0.399	0.467	0.506	0.565	0.417
12	0.490	0.406	0.542	0.456	0.519	0.404
13	0.688	0.263	0.434	0.624	0.684	0.280
14	0.609	0.388	0.559	0.442	0.590	0.395
15	0.796	0.315	0.801	0.362	0.808	0.297
16	0.524	0.509	0.523	0.540	0.541	0.494
17	0.544	0.249	0.606	0.233	0.574	0.236
18	0.647	0.380	0.805	0.270	0.763	0.292
Avg.:	0.589	0.314	0.573	0.379	**0.602#**	**0.306#**

Table 4.3: Results for each image for GBVS, GBVS-Chamaret and GBVS-Our method. Higher PLCC and lower KLD values indicate a better performance. # indicates that it is significantly different from the 2D model (paired t-test, $p<0.1$).

this with a significant result for the PLCC and KLD metrics.

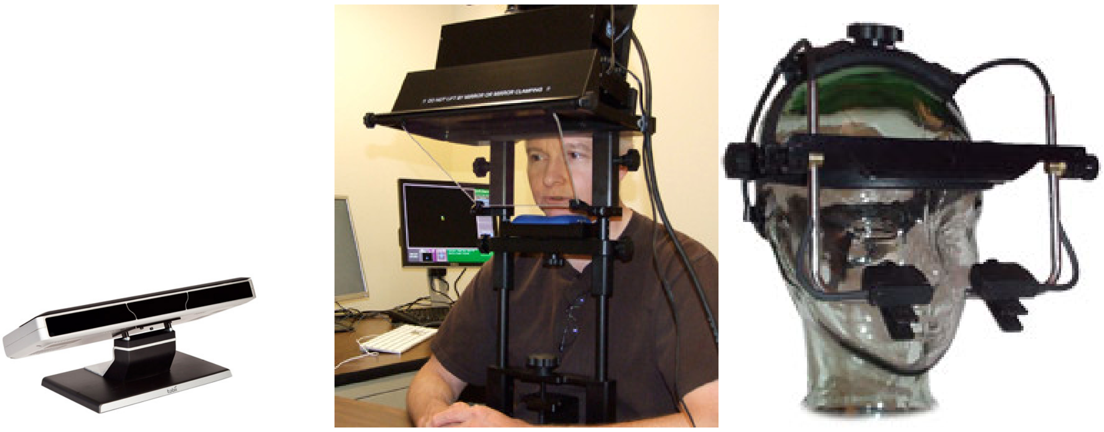
Chapter 5

Eye Tracking and Kriging

Eye tracking is used to validate eye movement behaviour for various applications such as image and video quality measurement [112, 185], interface design assessment [29, 30], to verify the accuracy of CMAs [188, 173] or simply to observe and assess the eye movement behaviour of people to whom different media are shown [66, 75]. This chapter presents the various eye tracking tools available and how they operate. It then presents an assessment of the accuracy of the "Eyelink II" (SR Research Ltd, v. 2.0) head-mounted eye tracker. Finally, a novel method is presented for capturing eye tracking data that improves the accuracy of eye trackers, provides a measure of uncertainty for all captured data and also significantly eases the eye tracking recording process.

5.1 Eye Tracking Equipment

Eye tracking equipment can be separated into two groups: remote and head mounted/tower-based. Remote sensors (a.k.a. non-intrusive systems) attempt to extract users' eye movements from a distance by using an external camera. These sensors provide a non-intrusive way of eye tracking to the point where the user may not even know that their eye movements are being recorded. Head-mounted eye trackers, as the name suggests, sit atop of the user's head and typically use infra red (IR) lighting to illuminate the pupil and extract its orientation. When IR light is impractical (e.g. outdoors), image-based methods have also been proposed [163]. Tower-based systems are similar



(a) Remote[1] (b) Tower-based[2] (c) Head mounted[3]

Figure 5.1: Three types of eye tracking devices

to head-mounted ones. The only difference is that the tracker is mounted onto a tower and the user places his head inside the tower. Figure 5.1 shows setups for remote, head-mounted and tower-based trackers. Since head movement, light, etc. need to be taken into consideration in the capturing process of remote trackers, head mounted and tower-based trackers have been shown to be more accurate [67, 133].

Eye tracking apparatuses can support binocular or monocular tracking of the eyes. Remote sensors typically record eye movements from 60Hz (60 readings per second) up to 1000Hz. Tower-based and head mounted trackers can have recording frequencies of up to 2000Hz.

The eye tracker that we used in our experiments was the "Eyelink II" (SR Research Ltd, v. 2.0) head-mounted eye tracker (shown in Figure 5.1 (c)). This eye tracker was connected to a machine running Windows 98 and the Eyelink software and was responsible for directly recording eye data from the tracker. The machine in turn was connected to a Windows 7 computer with 8GB Ram and 3.1GHz processor that ran the perception experiments. The eye tracker is accurate to within 0.5 degrees of visual angle [150]. Binocular data was collected at 500Hz. Participants viewed the

[1] Image adapted from `http://www.tobii.com/en/eye-tracking-research/global/products/hardware/tobii-tx300-eye-tracker/` (viewed 5/10/2014)

[2] Image adapted from `https://web.uvic.ca/psyc/masson/eyetracking.html` (viewed 5/10/2014)

[3] Image adapted from `http://www.sr-research.com/eyelinkII.html` (viewed 31/09/2014)

experiments on a 32" LCD monitor set at 1920x1080 resolution. Each participant's head was stabilised with a chin rest that was placed 80 cm from the screen. All participants in the experiments had normal or corrected-to-normal vision.

5.2 Calibration

To be accurate, eye trackers need a calibration phase prior to any tracking taking place. This calibration phase estimates physical characteristics of the human (e.g. cornea curvature) and determines the relationship between the eye position and position on the screen [73]. Calibration takes place by showing points of known position on the screen and requesting that the user focus on them. The more points displayed to the user, the more accurate the calibration can be. The calibration phase needs to be repeated after a certain amount of time because the accuracy of the eye tracker degrades over time. Due to this and the cost of good eye trackers, using eye tracking technologies to capture the visual behaviour of a set of candidates is a long, expensive and tedious process to set [153, 85].

Usually, eye tracking calibration is performed by the operator through the use of the eye trackers built-in calibration tool. Such tools, however, are black boxed meaning that it is not known exactly how this calibration takes place and what decisions are being made on behalf of the user. Moreover, even directly after calibration, error is still present in the data [18]. The default way to use eye trackers, therefore, is to allow accuracy to degrade until the next calibration stage.

5.2.1 State of the Art on Accuracy Analysis and Improvement

As mentioned above, eye tracking equipment lack in accuracy [67]. For example, McDonnell et al. reported that they accounted for approximately 100 pixels of error on a 1920 x 1200 LCD monitor when attempting to determine which object was being viewed [125]. A number of studies have been performed to analyse and improve the error produced by eye trackers. For instance, Holmqvist et al. [72] found that accuracy in many eye trackers tends to be the best in the middle of the screen and that bright backgrounds will cause the pupil size to decrease, which can lead to poorer results. Hornof and Halverson [73] found that the error can be different at various locations on the screen at a given time. They also suggest using implicit required fixation locations (locations that you know that everyone will look at while performing a task) to detect in real-time when calibration is required. Not all tasks, however, have such locations. Interestingly, Sugano et al. [160] established that saliency maps can assist in gaze prediction. They propose a calibration-free eye tracking method by mapping eye images to gaze points by kriging (described in the next section). Saliency maps, in this case, are treated as probability distributions of gaze points. Although an interesting concept, the accuracy reported was only 6 degrees (the Eyelink II eye tracker reports an accuracy of 0.5 degrees). Also, according to the in-depth survey on eye trackers by Hansen and Qiang [67] "new eye models and theories need to be developed to achieve calibration-free gaze tracking".

5.2.2 State of the Art on Kriging

This section will introduce the topic of interpolation and then the interpolation technique of kriging.

Interpolation is the method of obtaining (estimating) values within the range of a set of known points that have been obtained by, for example, sampling or experimentation. The intuitive way of calculating these values is to take a weighted average of the N surrounding known values:

$$\hat{X}(s_0) = \sum_{i=1}^{N} \lambda_i X(s_i) \tag{5.1}$$

where

$$\lambda_i \propto ||s_0 - s_i|| \tag{5.2}$$

and $\hat{X}(s_0)$ is the value being estimated at s_0, $X(s_i)$ are the known values at s_i ($i \in 1...N$). This interpolation method, although intuitive (its usage dates back to Ancient Babylonian astronomers and Greek mathematicians) does not provide results with a measure of certainty for interpolated values. To combat this disadvantage kriging was developed.

Kriging

Kriging interpolation has been used extensively in geographical and geostatistical applications. The mathematics behind it were developed by Andrey Kolmogorov (1903-1987) [97] and Herman Wold (1908-1992) [181] in the 1930s and by Norbert Wiener (1894-1964) in 1949 [178]. The word ‘kriging’ was coined by Pierre Carlier (French: *krigeage*) but brought into Anglo-Saxon mining terminology in 1963 by the French mathematician Georges Matheron (1930-2000) [123]. The technique is named after the South African mining engineer Danie G. Krige (1919-2013) who wrote a Master’s thesis on using kriging to evaluate concentrations of gold and other metals in blocks of rock from samples collected from drill holes [99, 38].

Kriging is based on the assumption that data being interpolated has a continuous nature and that nearby points have a higher degree of spatial correlation than points found further away. Interpolation is hence performed by giving larger weights to closer observed points. An advantage of kriging interpolation over standard linear interpolation methods includes the fact that it provides an estimation of the uncertainty along with interpolation results [5].

There has recently been a resurgence of kriging and today it may be better known as Gaussian Process Regression (GPR) in the spatial statistics field. Kriging is in fact a subcase of GPR with slightly differing terminology. For example, the terms ‘sill’ and ‘range’ (see Figure 5.2) would be referred to as the ‘variance paramater’ and ‘scale parameter’ in GPR literature [32, 80].

Ordinary kriging

Ordinary kriging is by far the most popular form of kriging in practice so it will be explained here first. It uses the standard interpolation technique with weights (Eq. 5.1) but to ensure an unbiased estimate, the weights are made to sum to 1:

$$\sum_{j=i}^{N} \lambda_i = 1 \tag{5.3}$$

The expected error is assumed to be $E[\hat{X}(s_0) - X(s_0)] = 0$ and the estimation variance is given by:

$$\begin{aligned} var[\hat{X}(s_0)] &= E[\{\hat{X}(s_0) - X(s_0)\}^2] \\ &= 2\sum_{i=1}^{N} \lambda_i \gamma(s_i, s_0) - \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_j \lambda_i \gamma(s_i, s_j) \end{aligned} \tag{5.4}$$

where $\gamma(s_i, s_j)$ is the semivariance of X between the data points s_i and s_j.

The goal in kriging is to find the weights λ that minimise the variance in 5.4. To impose the constraint in 5.3 a Lagrange multiplier (ψ) is added such that the auxiliary function $f(\lambda_i, \psi)$ containing the variance we wish to minimise is defined as:

$$f(\lambda_i, \psi) = \arg\min var[\hat{X}(s_0) - X(s_0)] - 2\psi \left\{ \sum_{i=i}^{N} \lambda_i - 1 \right\} \tag{5.5}$$

With the two partial derivatives of the auxiliary function set to 0 for $i = 1, 2, ..., N$ this gives a set of $N + 1$ equations with $N + 1$ unknowns:

$$\sum_{i=i}^{N} \lambda_i \gamma(s_i, s_j) + \psi(s_0) = \gamma(s_j, s_0), \forall j \tag{5.6}$$

This is the kriging system for points that provides the values for the weights λ in Equation 5.1. The variance of the estimated results can be obtained by:

$$\sigma^2(s_0) = \sum_{i=i}^{N} \lambda_i \gamma(s_i, s_0) + \psi(s_0) \tag{5.7}$$

Variogram

To solve Equation 5.5, a variogram function γ, which is directly linked to the covariance of the stochastic processes, is chosen to model the spatio-temporal dependencies. In practice, this variogram is chosen such that it best describes a computed empirical variogram. Depending on the empirical variogram, a different model can be chosen. One of the most common models is the exponential model, which can be described as:

$$\gamma(x) = c\left\{1 - \exp(\frac{-x}{r})\right\} \tag{5.8}$$

where c is the sill and r the range as shown in Figure 5.2. The sill and range will ultimately dictate the size of the weights λ. A nugget can also be added to the model to more closely control the covariance close to the origin.

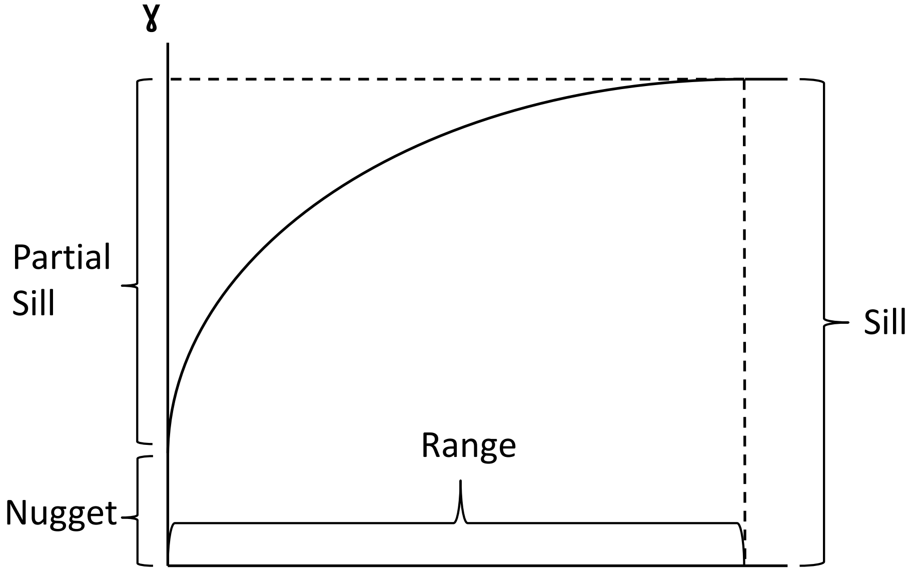


Figure 5.2: Exponential variogram model.

Some other common variogram models include the bounded linear, circular, spherical and pentaspherical.

Simple and universal kriging

As was mentioned earlier, the most common form of kriging is ordinary kriging, which assumes that the mean is unknown. Two other well known forms of kriging are simple and universal kriging. Simple kriging is used when the mean of a random variable is known (from past experience, for example). This additional knowledge can be used in calculations. With a known mean, μ, Equation 5.1 becomes

$$\hat{X}(s_0) = \sum_{i=1}^{N} \lambda_i X(s_i) + \{1 - \sum_{i=1}^{N} \lambda_i\}\mu \tag{5.9}$$

Universal kriging is used when the spatial processes are comprised of stochastic and deterministic components. It assumes that we have a model telling us how the mean evolves in the spatio-temporal domain. The general formula then becomes

$$\hat{X}(s_0) = \sum_{k=1}^{K} \sum_{i=1}^{N} a_k \lambda_i f_k X(s_i) \tag{5.10}$$

where f is a set of polynomials representing the non-stationary components with unknown coefficients a_k.

5.2.3 Use of Kriging/GPR

An example of kriging being used for interpolation is Oliver and Webster [134] who used it to map and control soil salinity in the Jordan Valley of Israel. Recently, Umer et al. [168, 167] used kriging in wireless sensor networks to make monitoring systems more resilient to coverage holes.

In the spatial statistics domain, GPR was used for trajectory analysis by Cox et al. [37] by using it to model trajectories of participants' hand movements during psychology experiments. They found it a suitable method that provided them with fine-grained glimpses into cognitive processes of participants' responses.

In the context of gaze estimation Noris et al. [132] used GPR and SVM to estimate and analyze gaze information for children aged 6-18 months. They created a wireless head-mounted camera (see Figure 5.3) for this purpose and trained the GPR using raw pixel intensities of the eye regions to known image coordinates that test participants were requested to fixate on. The same was done to train an SVM and results were compared between the two methods. Accuracy was, however, not of primary importance to them. They report results of up to 2.34 degrees (using GPR) but on adult subjects only.

Williams et al. [179] propose to use GPR for gaze estimation by also learning a mapping of images of the eyes from a camera to known gaze positions on the screen. They crop camera images to contain only images of the eyes and then perform a feature extraction step (greyscale intensity) of the eye images. Features are then converted into feature vectors that are used in the mapping from inputs to outputs. They use GPR with calibration data to estimate gaze positions and report accuracy results of up to

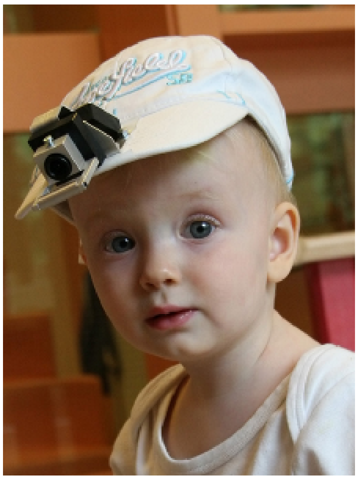


Figure 5.3: A baby wearing the WearCam from [132].

0.83 degrees. However, the test data to assess their method was captured in the middle of the calibration phase (between 16 calibration points shown for 1 second each). This is an unrealistic scenario for usual eye tracking experiments that require longer periods (a few minutes) between calibration phases. Hence, it is impossible to determine whether this method is robust for real-world scenarios. It also questions the accuracies reported for their method because the further you move from the calibration phase, the less accurate your readings will become.

Recently, Liang et al. [106] reported eye gaze estimation results equal to the Eyelink II eye tracker (0.5 degrees). They have a similar technique to Williams et al. except that they propose an improvement to their feature extraction step. However, they only had 5 people perform their assessment experiment and, therefore, their results are statistically insignificant especially since they did not provide any in-depth statistical analysis in their results. Similarly to Williams et al., they also gathered their test data close to their calibration phase meaning that in a real eye tracking scenario their accuracy results may prove to be worse than reported.

No one has used kriging (or GPR) to improve already existing commercial eye trackers that are more accurate than anything that has been so far proposed in the latest state of the art in computer vision. Nor has eye tracking with kriging been shown to be a viable option for real-world eye tracking scenarios.

5.3 Automatic Kriging Calibration for Eye Tracking

This section will propose a method to record eye tracking data more accurately by defining a new calibration process. By using interpolation, information obtained from all calibration steps can be used to accurately interpolate pixel shifts on eye tracking data. This method can also reduce costs. Good eye trackers are too expensive for general public use. Improving the accuracy of an older eye tracker can remove the need to invest in a newer more accurrate one [67].

Section 5.3.1 will describe the way kriging can be used to model drift in eye tracking and Section 5.3.2 will describe how the calibration phases were set up to implement the kriging technique.

5.3.1 Ordinary Kriging in Eye Tracking

We define two spatio-temporal stochastic processes, Δ_x and Δ_y, representing the eye tracking data shifts from recorded to real locations in horizontal and vertical directions respectively. These processes vary in space and time during the recording and we note $s = (x, y, t)$ the location or site in the space-time domain where (x, y) refers to the spatial position in pixels, and t is the time measured in seconds. At several known sites $\{s_i\}_{i=1,\cdots,N}$, calibration measurements are recorded $\{\Delta_x(s_i), \Delta_y(s_i)\}_{i=1,\cdots,N}$ as part of the recording session.

To perform interpolation we propose to use ordinary kriging that assumes that the expectations of the spatio-temporal stochastic processes do not depend on location in space and time, i.e. $\mathbb{E}[\Delta_x(s)] = \mu_x$ and $\mathbb{E}[\Delta_y(s)] = \mu_y$, $\forall s$. Both expectations μ_x and μ_y are, however, unknown. Considering a new site $s_0 = (x_0, y_0, t_0)$, the shifts can be estimated by interpolation as follows [176]:

$$\Delta_x(s_0) = \sum_{i=1}^{N} \lambda_i^x \ \Delta_x(s_i) + \epsilon_x(s_0) \tag{5.11}$$

and

$$\Delta_y(s_0) = \sum_{i=1}^{N} \lambda_i^y \ \Delta_y(s_i) + \epsilon_y(s_0) \tag{5.12}$$

where λ_i^x and λ_i^y are the weights assigned to the calculated (horizontal and vertical) shifts of the known sites. We assume the expectation of the errors are zeros, i.e. $\mathbb{E}[\epsilon_x(s_0)] = 0$ and $\mathbb{E}[\epsilon_y(s_0)] = 0$.

The parameters λs are then estimated by minimising the variance of the errors:

$$\begin{cases} \left(\{\hat{\lambda}_i^x\}_{i=1,\cdots,N}, \hat{\mu}_x\right) = \arg\min \mathbb{E}[(\epsilon_x(s_0))^2] \\ \left(\{\hat{\lambda}_i^y\}_{i=1,\cdots,N}, \hat{\mu}_y\right) = \arg\min \mathbb{E}[(\epsilon_y(s_0))^2] \end{cases} \tag{5.13}$$

with the following constraints on the weights (for unbiased estimation):

$$\sum_{i=1}^{N} \lambda_i^x = 1 \quad \text{and} \quad \sum_{i=1}^{N} \lambda_i^y = 1$$

This corresponds to

$$\begin{cases} \{\hat{\lambda}_i^x\}_{i=1,\cdots,N} = \arg\min \mathbb{E}[(\Delta_x(s_0) - \sum_{i=1}^{N} \lambda_i^x \ \Delta_x(s_i))^2] \\ \\ \{\hat{\lambda}_i^y\}_{i=1,\cdots,N} = \arg\min \mathbb{E}[(\Delta_y(s_0) - \sum_{i=1}^{N} \lambda_i^y \ \Delta_y(s_i))^2] \end{cases} \tag{5.14}$$

To solve Equation (5.14), with Lagrange multipliers to enforce the constraints of the weights, a variogram function γ, which is directly linked to the covariance of the stochastic processes, is chosen to model spatio-temporal dependencies:

$$\mathbb{E}[(\Delta_x(s_0) - \sum_{i=1}^{N} \lambda_i^x \ \Delta_x(s_i))^2] \ = \ 2\sum_{i=1}^{N} \lambda_i^x \ \gamma(s_i, s_0) - \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i^x \ \lambda_j^x \ \gamma(s_i, s_j) \tag{5.15}$$

and similarly

$$\mathbb{E}[(\Delta_y(s_0) - \sum_{i=1}^{N} \lambda_i^y \ \Delta_y(s_i))^2] \ = \ 2\sum_{i=1}^{N} \lambda_i^y \ \gamma(s_i, s_0) - \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i^y \ \lambda_j^y \ \gamma(s_i, s_j) \tag{5.16}$$

The same exponential variogram function γ is used for both stochastic processes Δ_x and Δ_y, and it is defined as $\gamma(s_i, s_j) = 0$ when the distance between the two sites is null $\|s_i - s_j\| = 0$, otherwise when $\|s_i - s_j\| > 0$, the variogram is defined as:

$$\gamma(s_i, s_j) = \left(c_{\mathbf{x}} + \sigma_{\mathbf{x}}^2 \left(1 - \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{a_{\mathbf{x}}}\right)\right)\right) \times \\ \left(c_t + \sigma_t^2 \left(1 - \exp\left(-\frac{\|t_i - t_j\|}{a_t}\right)\right)\right) \quad (5.17)$$

with notation $s_i = (x_i, y_i, t_i) = (\mathbf{x}_i, t_i)$ to address both time and (isotropic) space domains separately. For all our spatio-temporal kriging calculations we used the gstat package in R[4] and spacetime data structures [139]. To estimate the parameters of the variogram (cf. Tab. 5.1 and Fig. 5.2), using our data an empirical variogram was first calculated onto which the exponential variogram model was fitted. The variogram determines how the weights λs are computed, dictating the influence of the observed data on the interpolated point; e.g. sites that are too far away (as controlled by the range) would have little or no impact on the interpolated values for Δ_x and Δ_y. Figure 5.4 shows an example empirical variogram of data from a participant (Δ_y) and the variogram fitted onto this data.

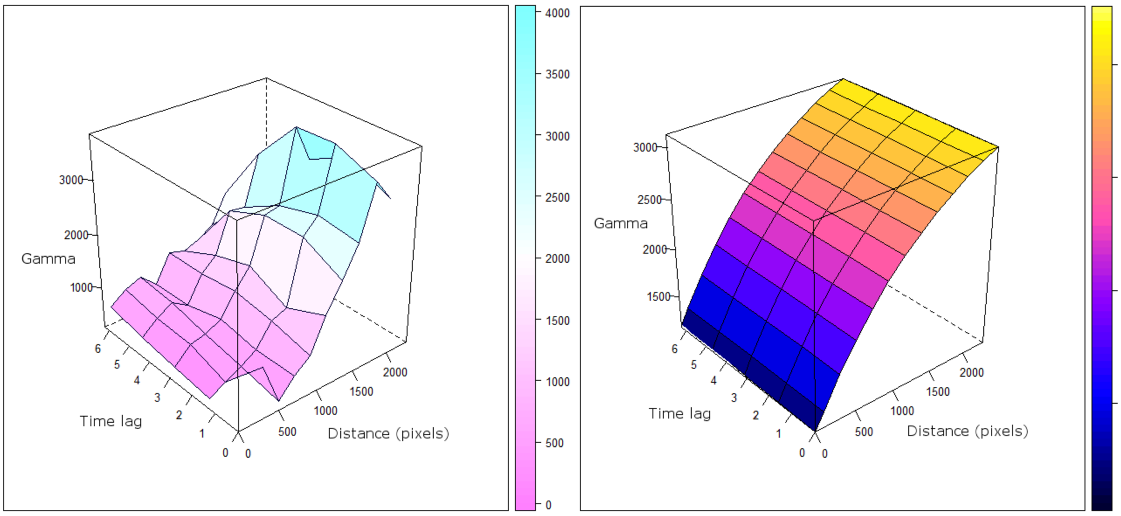
Function	**Sill**	**Range**	**Nugget**
Space	$\sigma_{\mathbf{x}}^2 + c_{\mathbf{x}} = 1000$	$a_{\mathbf{x}} = 750$ pixels	$c_{\mathbf{x}} = 0$
Time	$\sigma_t^2 + c_t = 1000$	$a_t = 150$ secs	$c_t = 0$

Table 5.1: Estimated parameters for the spatio-temporal exponential variogram (cf. Fig. 5.2).

5.3.2 Our Calibration Method

Our calibration method involves individually showing calibration points, of which true positions on the screen are known, distributed equally on the screen. The user focuses on each of the points and fixation data is recorded by the eye tracker. Calibration sites $\{s_i\}_{i=1,\cdots,13}$ in the spatio-temporal domain are where our shifts $\{\Delta_x(s_i), \Delta_y(s_i)\}_{i=1,\cdots,13}$ are recorded.

[4] http://www.r-project.org/



(a) Empirial variogram

(b) Fitted variogram

Figure 5.4: Example empirical and fitted variogram on real data from a particpant.

The next section will discuss the experiment that was set up to test and assess the new calibration method and the results obtained from this experiment.

5.4 Experimental Results

The experiment was divided into two stages. The first stage was performed using the built-in calibration tool that comes with eye trackers. The second stage was identical to the first except that it was performed using our kriging calibration apart from the initial calibration at the start of the recording that was performed with the built-in calibration tool. The order of these stages was randomised for each run of the experiment. For calibration we used 13 points for both stages. Fewer points can be used but accuracy is degraded as the number of calibration points decreases.

Each of the two stages was divided into parts consisting of the following: a calibration step, three test points shown of known position, a short film clip (25 secs), three test points shown again, a short film clip (25 secs), and the same three test points shown again. Each part was repeated to the participant four times with no breaks. The three test points were points chosen to assess the accuracy of kriging interpolation.

The adjustment performed via interpolation on the fixations on these points was compared to the real known positions of the points on the screen. One point was chosen to be in the middle of three calibration points, one point was chosen to be in the middle of two calibration points and one test point was chosen to be at exactly the same site as a calibration point. Figure 5.5 shows the positions of the calibration and test points used in the experiment. The built-in calibration tool was used only once for the second stage of the experiment - at the very beginning to align the eye tracker with the screen.

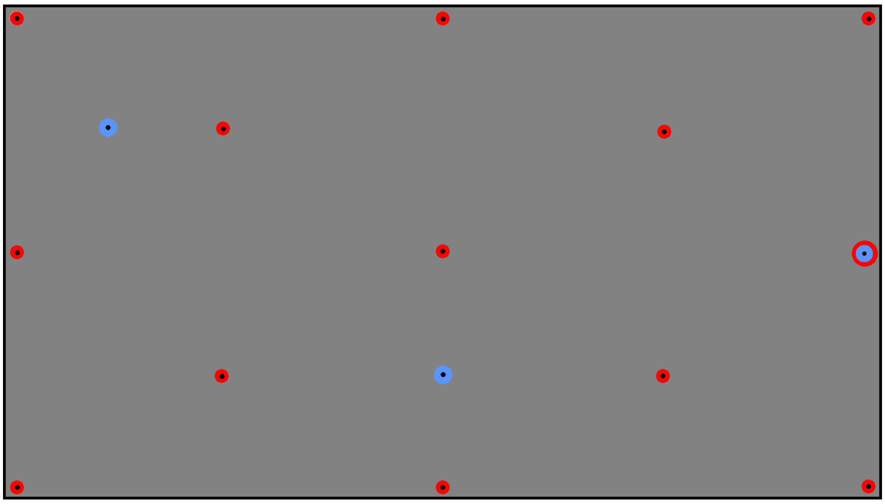


Figure 5.5: The 13 calibration points (red) and 3 test points (blue) used in the second stage of the experiment. The red-blue point was used as a calibration and test point.

Eye movements were recorded using the eye tracker and setup described in Section 5.1. The experiment was conducted 11 times on four different people. Since both eyes were tracked, this gave us de facto 22 separate runs of the experiment.

Firstly, we show how the accuracy of the eye tracker degrades over time. The data used for this was taken from the second stage of the experiment where we use our own calibration method. If we opt not to use interpolation, the raw fixation data can be used on the known positions of the three test points to see how this difference (error) from the real positions of the points changes over time. We use the Euclidean distance as a measure of the error. For each group of three test points shown, the average error was calculated for these and plotted over time. Figure 5.6 shows the average error with its standard deviation that is depicted as error bars. Next we compare our method

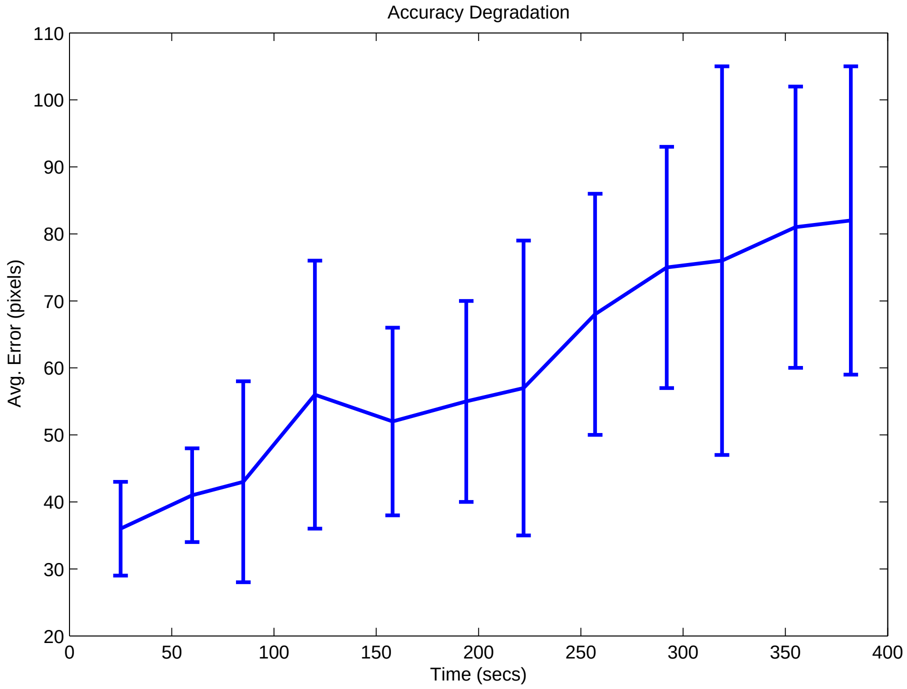


Figure 5.6: Average degradation of accuracy of the eye tracker over the 22 experiments. Standard deviation shown as error bars.

of calibration with the standard method. Figure 5.7 shows the error along with the standard deviation (error bars). There is a clear improvement of accuracy with the kriging method.

Figure 5.8 shows the interpolated shift $\Delta_x(x, y, t)$ at a chosen time t over the entire screen. We see that the shift varies significantly from approx. -10 to over 60 pixels.

Figure 5.9 shows the interpolated shifts $\Delta_x(\cdot, \cdot, t)$ and $\Delta_y(\cdot, \cdot, t)$ over time for one selected pixel (the left-most, middle calibration point from the start until the end of the experiment) for two people. It can be seen in this figure how the shift values generally increase as the experiment progresses. The shift values increase to compensate for the increase of error. Note that kriging interpolation also provides a measure of the uncertainty for each interpolated value and the confidence intervals are also shown: the confidence intervals are zero for calibration times (no uncertainty). The further away the point is from a calibration site, the larger the confidence interval.

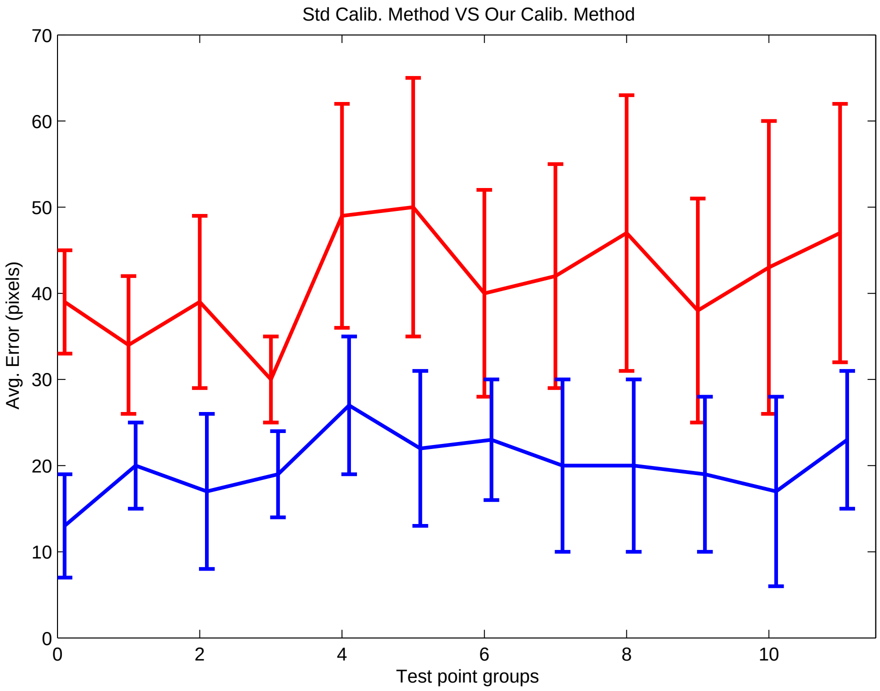


Figure 5.7: Average error for each showing of the three test points with standard deviations obtained using the standard calibration method (red) and our kriging method (blue) over the 22 experiments.

The kriging calibration approach presented here also allows for the time taken to conduct experiments to be reduced compared to standard calibration. This is because only one initial calibration step with the built-in tool is required with our kriging method. With the standard method, sometimes calibration with the built-in tool has to be repeated several times during a calibration stage. On average, the experiments using the standard method of calibration took 19% longer than the kriging approach. Shorter recording sessions both save time and also increase the comfort level of participants. The comfort level for the operator of the eye tracker is also improved since he/she is not tied to the computer controlling the eye tracker through the duration of the experiments.

Although the eye tracking recording sessions are shorter for the participants it should be mentioned that additional work is passed down to the experimentor in the

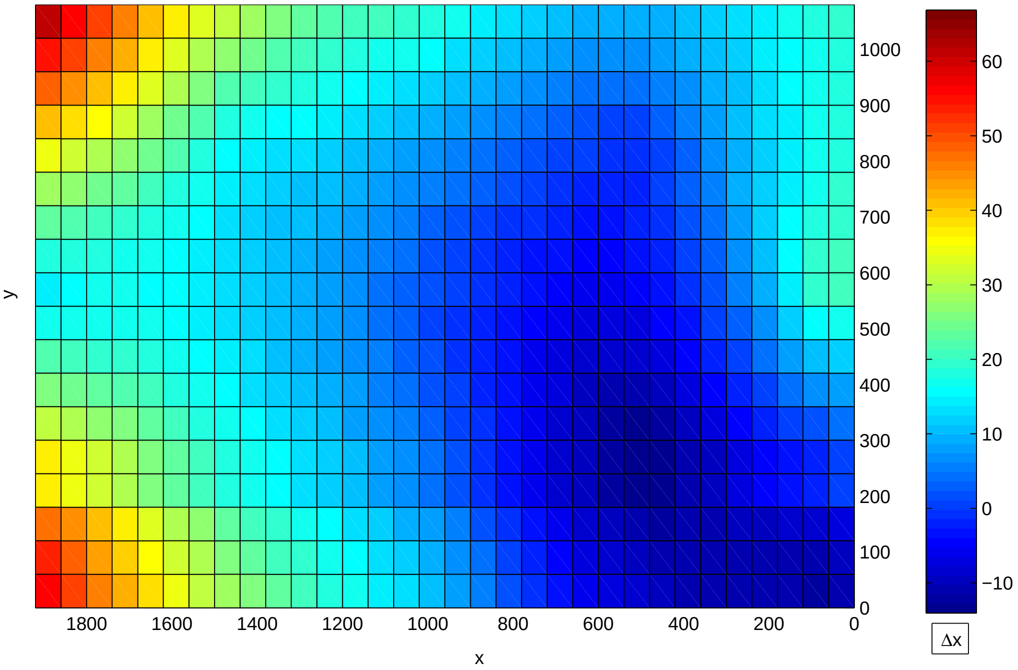
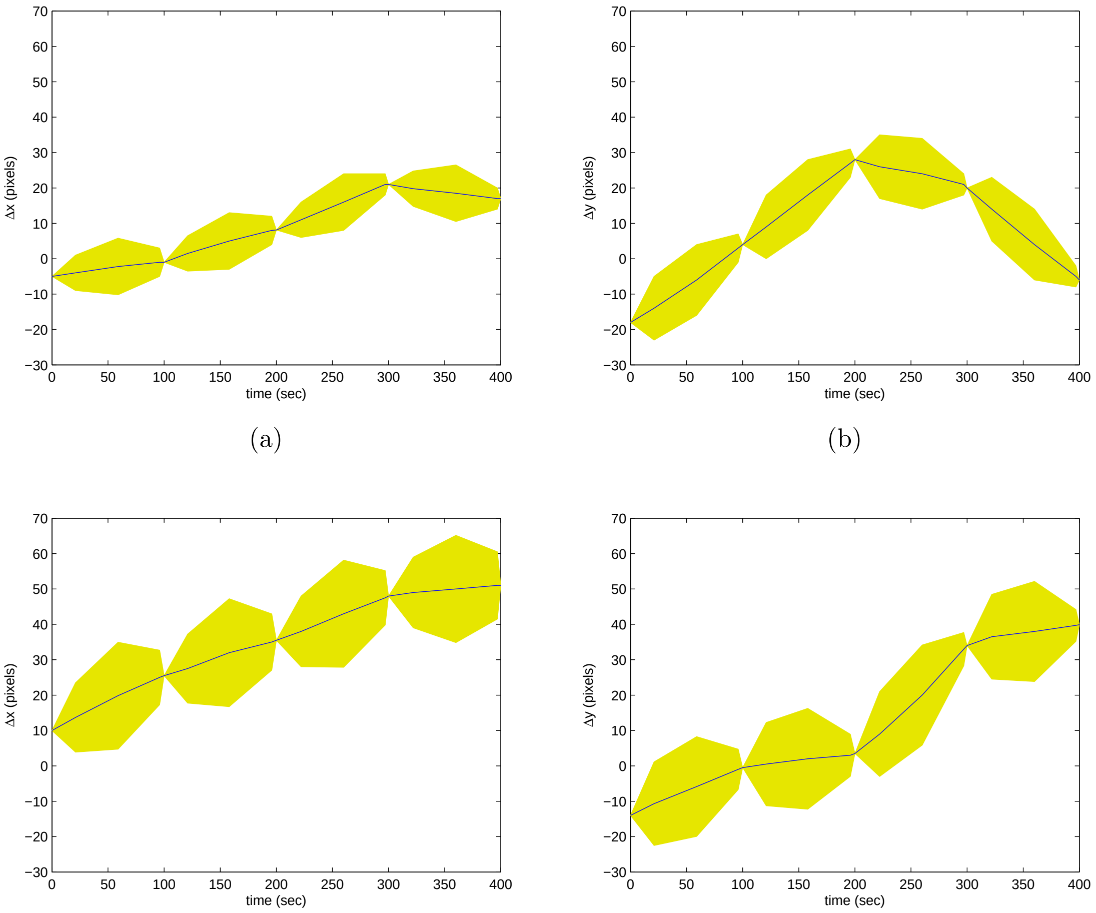


Figure 5.8: Example image representing $\Delta_x(x, y, t = 143)$ interpolated on the entire screen.

post-processing stage. Scripts need to be written to extract calibration point fixations of participants from eye tracking data. These fixations then need to be fed into the application that performs kriging. Finally, another script needs to be used to combine the original eye tracking data with shift calculations to yield the final interpolated eye position values. This additional effort can be deemed acceptable, however, considering the benefits of our method discussed above.

5.5 Conclusion

In this chapter we have proposed to use kriging interpolation to accurately correct eye tracking data. This new approach limits the need to use the black-boxed built-in calibration tool during the recordings. Kriging calibration has been shown to be more accurate than standard calibration, it provides a measure of uncertainty for all



(a)

(b)

(c)

(d)

Figure 5.9: (a) Δ_x for one pixel (a calibration point) for one person; (b) Δ_y for the same person and point; (c) Δ_x for one pixel (a calibration point) for another person; (d) Δ_y for the same person and point;

interpolated data and also significantly eases the eye tracking recording process. Since eye tracking is used to validata VSAs, improving eye tracking equipment accuracy with our kriging interpolation will have a positive knock-on effect on this field of research as well as on other fields that use eye tracking. Moreover, since eye trackers are expensive (prices range from €4000-€30,000), improving them will avoid the infeasible (for most) notion of constantly upgrading them with newer, more accurate models when they become available.

Chapter 6

2D & 3D Visual Perception

3D film viewing is becoming a major source of entertainment not only in cinemas but in private lounges as well. It is implicitly known that people view video content in 3D differently to 2D [7, 11]. Moreover, viewing content in 3D presents new challenges and paradigms (e.g. new artefacts and imperfections that are not present in 2D [16, 7]) that impact the viewing experience. 2D VSAs cannot, therefore, be directly ported into the 3D domain without the loss of accuracy. It is important, hence, that an analysis of the difference between 2D/3D viewing behaviour is performed so that work on VSAs in the 3D domain can advance.

This chapter presents the acquisition of a 2D/3D professionally-made video dataset, eye tracking data accompanying it and an analysis of the difference in 2D and 3D viewing behaviour of people on this dataset. The work presented here was performed over the space of 12 months: 2 months to ascertain the current level of research on this topic, 1 month to locate and extract appropriate videos to be used in the dataset, 2 months to implement the experimental setup (eye tracker interfacing with a C++ program using the GRUB library that connects to a Quadro video card to display 3D content on a 3D screen), 1 month of testing this setup, 1 month of running the experiment on 50 people, 1 month of performing kriging interpolation on the eye tracking data we obtained and, finally, 4 months of analysis of this data.

The first section of this chapter looks at the current state of the art in 2D/3D analysis of viewing behaviour. The next section presents our new method for creating fixation density maps and the following sections will present the results of our viewing behaviour analysis, which will also include advice to film makers and editors.

6.1 2D/3D Viewing Behaviour

Jansen et al. [82] performed a study to investigate the effect of disparity in viewing natural still images in 2D and 3D. They found that for images shown in 3D, people fixate more, have shorter and faster saccades (the movement of the eyes between fixations) and tend to explore more with their eyes. They also found that there was minimal difference in saliency of mean luminance, luminance contrast and texture contrast.

El-Nasr and Yan [45] conducted a study to analyse visual attention patterns of players within an interactive 3D game environment. The intention of the study was to provide data for game designers to improve gaming experiences. They found that since games are highly goal oriented, top-down cues dominate visual attention more than bottom-up ones (only motion and colour were analysed here) but in a few cases, bottom-up features overwrote any top-down influences. The specific example discussed where the latter took place was of a red object whose brightness increased over time. This object was located on the side of the screen and not on the path the player was supposed to be taking. The colour feature overwrote the top-down task and drew players' attention to the object.

Studies have been performed to analyse where people look when watching movies in 2D. For example, Goldstein et al. [61] performed experiments whose results showed that male and older people were more likely to look in the same place when watching movies than female and younger people. Their study was used to improve an automatic screen zooming device for the visually impaired.

Very little has been done, however, to analyse differences in viewing behaviour between 2D and 3D movies. Häkkinen et al. [66] in 2010 analysed six scenes from a short film. 20 students were asked to view these scenes in random order. The participants were given a task prior to viewing that asked them to decide which version (2D or 3D) was better. In their analysis, Häkkinen et al. divided each scene into areas of interest (e.g. water, boy, background) and counted the number of fixations in these. They found that the same content when viewed in 3D produced a more widely distributed gaze pattern from their viewers. Criticisms of their work include the fact that the comparison task introduced top-down factors that could have influenced their results, e.g. students may have specifically roamed with their eyes to assess quality in the foreground and background of scenes. Also, the six scenes analysed were chosen to not have large amounts of camera or object movements. The dataset analysed, therefore, was only a minimal representation of real movies.

Huynh-Thu and Schiatti [75] performed a similar experiment with 21 video sequences taken from a variety of sources such as advertisements, films and computer graphics animations. The clips were chosen to incorporate a wide variety of scene characteristics in terms of motion, contrast, colour and depth. 18 participants were told to view these clips freely. Half of the participants viewed the clips in 2D first, the other half in 3D. They found that a higher fixation frequency and longer average fixation duration were exhibited in 2D viewing. Faster saccades were reported for all but one of the 3D clips. Criticims of their work include the fact that the video clips did not include an audio track. Sound is inherent in films and, hence, it can be argued that the experiment was also not emulating a true film viewing experience. Sound has also been shown to affect eye movements [131].

Another important criticism of the work done by Huynh-Thu and Schiatti and Häkkinen et al. is that the same people were used to view the media in 2D & 3D (although sometimes 2D was shown before 3D and vice-versa). This completely disregards the effect of memory on attention guidance. If a person has already seen a scene in one dimension, when he looks at the same scene (with the same semantic information) in another dimension, he may look at it differently. Differences in viewing behaviour in these studies, therefore, cannot be just attributed to 2D-3D factors but also must be attributed to memory bias. Many studies have been conducted to show that memory bias plays a significant (if not prominent) role in attention guidance [71, 183, 15]. Most importantly, Hadizadeh et al. [65] recently performed an experiment where they showed the same 12 videos to participants twice. Significant differences were found in the viewing behaviour by observers between the first and second viewings.

Moreover, both Huynh-Thu and Schiatti and Häkkinen et al. used remote eye trackers, which, as was mentioned in the previous chapter, are less accurate than others.

Ramasamy et al. [147] also compared 2D and 3D viewing but specifically as a tool for filmmaking. They found that in one scene the gaze data in 3D was more focused when compared to the 2D version, which is a different result to that of Häkkinen et al. and Huynh-Thu and Schiatti as discussed above. The clips analysed, however, were taken from a first person stereoscopic film that simulated the human field of view. This viewing paradigm is dominant in games but not in films.

Finally, none of the analyses presented in this section have used in-depth statistical means to show significant differences between 2D and 3D viewing behaviour. With our kriging interpolation, we can accurately show when the difference between 2D and 3D viewing behaviour is statistically significant and we can also provide a measure of certainty.

6.2 Eye Position Maps

As an indication of participants' viewing patterns in our experiments for a given frame or number of frames eye position density maps were calculated. These were calculated using information obtained from the eye tracker and our kriging calculations. Currently, maps such as these are calculated by placing a Gaussian function for every fixation (for fixation density maps) or eye position (for eye position density maps) obtained from eye trackers to create a Gaussian mixture model (GMM). The standard deviation chosen, however, for each of these functions varies. Le Meur and Baccino [126] suggest that 1° is used as a convention. In practice, however, it is frequently chosen arbitrarily. For example, Wang et al. [173] chose 2° for each of their Gaussians. It can be argued that sometimes these numbers are chosen to make the maps look aesthetically pleasing because if the number of participants is small, a lower standard deviation will make the map look bare.

Our kriging calculations provide us with variance information associated with every eye tracker reading. This variance can be used on the Gaussians that make up a fixation/eye position density map and is a much more accurate, informative and scientific (because it is based on statistics) way of calculating these. For each eye position for each person we plot a Gaussian to create a GMM. Since the number of eye positions for a given frame can vary for each person (due to eye tracker variability, etc.) we average each person's GMM before summing them to create the final GMM (i.e. eye position density map):

$$f(x,y) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{J_i}\sum_{k=1}^{J_i}\frac{1}{2\pi\sigma_{ik}\sigma_{ik}}e^{-\left(\frac{(x-x_{ik})^2}{2\sigma_{ik}^{x2}}+\frac{(y-y_{ik})^2}{2\sigma_{ik}^{y2}}\right)} \tag{6.1}$$

where N is the number of participants, J_i the number of readings for a given frame for participant i, σ^x_{ik} is the variance for the k^{th} reading in the x-direction (σ^y_{ik} for the y-direction) for person i and x_{ik} the k^{th} x position of the eye (y_{ik} for the y position) for person i. Example eye position density maps can be seen in Figure B.1 (d) and (f).

6.3 2D & 3D Perception

This section will present a comparison between 2D and 3D viewing behaviour. Our novel kriging intepolation method was utilised to capture the viewing behaviour for this comparison.

6.3.1 The Dataset

The dataset, depicted in Table 6.1, is composed of 8 video clips. It is divided into three groups: strong 3D (i.e. videos with salient objects that protrude from the screen, scenes with a large depth plane, etc.), little camera and object movements, and fast camera and object movements. Häkkinen et al. [66] only analysed clips from the second group. We felt that it was necessary for there to be videos representing more than just one group. These three were chosen because they appear to have different effects on people when viewed in 3D compared to 2D. As will be shown, this proved to be true.

6.3.2 The Questionnaire

A short anonymous questionnaire was also given to every participant (see Appendix A). This was done to garner more information about the viewing population to discern cluster classes of viewing behaviour and to ascertain any consensus on 3D technology. This has not been done before for 3D and 2D viewing comparison experiments.

6.4 Experimental Results

This section will discuss results obtained from the perception experiment that was run. 50 people (25 males and 25 females) were asked to watch the videos listed in Section 6.3.1. The experimental setup with the eye tracker is described in Section 5.1. Half of

Movie Title	Segment location	Duration
Strong 3D		
1) The Ultimate Wave T.	0:02:57-0:03:18	0:21
2) Life of Pi	0:05:08-0:05:29	0:21
3) Megamind	0:21:09-0:21:28	0:19
Total		1:01
Little camera and object movements		
4) The Ultimate Wave T.	0:32:01-0:32:19	0:18
5) Avatar	0:05:19-0:05:47	0:28
6) To the Arctic	0:21:46-0:22:07	0:21
Total		1:07
Fast camera and object movements		
7) Alice in Wonderland	0:12:45-0:13:25	0:40
8) Life of Pi	0:40:44-0:41:05	0:21
Total		1:01
Total for all clips		3:05

Table 6.1: List of films used in the experiment

the participants watched the 2D versions of the videos and the other half just the 3D (with 3D goggles on) in random order. Only one version was watched by each person to avoid memory affecting the viewing of the same content in a different dimension. Interspersed between the videos in the dataset were 7 calibration phases as well as 16 2D and 3D pictures of real (from the BOLD 3D faces dataset [84]) and synthetic (from 3ds Max) faces - both male and female. Analysis of viewing behaviour of these 2D/3D faces is outside of the scope of this thesis and is left for future work.

6.4.1 Collection of Results

A t-test was conducted to measure the significance of the difference between mean positions computed in the 2D and 3D cases. The variances used for this were taken from our kriging uncertainties. Figure 6.1 summarises the results. Each bar represents a video sequence and the red regions show where there was significant difference in the viewing behaviour. Figure B.1 (e) & (g) shows the single Gaussian representation of eye positions for frame 55 from video #1. The difference between 2D and 3D was calculated to be significant by the t-test.

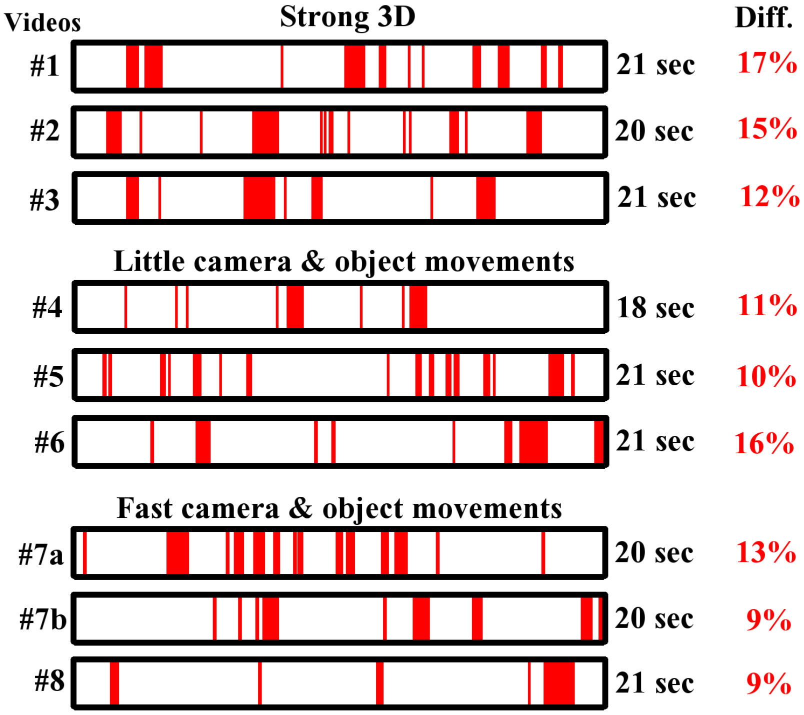


Figure 6.1: Summary of results. Video #7 was split in two for ease of comparison.

The calculations to ascertain significant differences in viewing behaviour were performed as a guide to help us notice when there were divergences in 2D/3D viewing patterns so that further analysis (for film makers, for example) could be done. This analysis is presented in the following sections. Due to the number of eye position maps referenced in these sections, they will be placed in the Appendix. Where optical flow diagrams and disparity maps are presented, they were created using Sun et al.'s algorithm [162]. This algorithm was inaccurate on two occasions. In these cases the disparity maps were created using OpenCV (version 2.4.9) and its semi-global block matching algorithm, which is a modified version of Hirschmuller's algorithm [69, 40].

6.4.2 Videos with Strong 3D

We observe a statistical difference between 2D and 3D visual saliency in the videos presenting strong 3D. In fact, these videos, on average, had the most amount of differences (see Table 6.1), which could indicate that the additional dimension plays a significant role in the viewing behaviour of viewers. Statistics from eye tracking data from this group of videos can be found in Table B.1.

Video #1

Video #1 is from the documentary 'Ultimate Wave Tahiti'. It shows a model of our solar system with planets orbiting the Sun. The camera pans from left to right during which the spinning Earth slowly comes into view with the Moon also featuring prominently. Other planets, especially Mercury and Venus, also have visible roles. The 3D in this video clip is very strong and the planets that feature and the Sun have distinct places in the 3D plane in front of a starry background. The audio for this clip is of a man talking generally about gravity and how it affects the tides on Earth and does not point out anything in particular in the video.

The major difference between 2D/3D perception in this video is that 3D viewers principally focussed on only 1-2 objects (mainly Earth or the Moon and Venus) at any given time. In 2D, viewers had a much more distributed viewing pattern with the Sun (the largest object in the clip), Earth, Moon, Venus and Saturn (which is behind the Sun) being focussed on a lot. This can be seen in Figure B.1 (d) & (f) and Figure B.2 (d) & (e) that present the eye position density maps for frames 55 & 59. This is the very beginning of the clip after a short amount of time has passed for the viewers to grasp the context of the visual and audio cues being presented. It is interesting to note, therefore, that distinct objects in the 3D plane grabbed the attention of participants from the very beginning of the clip. Figure B.3 and B.4 in Appendix B shows this to also be true for later parts of the clip. In fact, this held true throughout the video. Whereas in 2D, viewing was always more distributed until the end, when it appears to become more concentrated on the objects closest to the camera (see Figure B.5). This would indicate that the viewers' attention became more focussed when they had finished exploring the entire scene in 2D.

A number of things can be gathered from these results. Firstly, motion and depth can play a dominating role in attention grabbing in 3D when the audio is of a general nature (i.e. not originating from a source in the video or not pointing out anything in particular in the video). The Sun is the brightest and most intense object in the clip.

Moreover it also has a face on it. Despite this, it is rarely glanced at in 3D. The flying planets of Venus and Earth and its Moon dominate the saliency because they ‘pop out’ of the screen and move past the viewer - despite them being grey in colour. Moreover, the background planets (e.g. Saturn) are ignored, as is the starry background. This means that in scenarios such as this one, film makers need to devote a lot more time to detail for an entire scene in 2D whereas in 3D, just the prominent objects can be focussed on. Details such as a face on a sun may be missed entirely in 3D.

Video #2

Video #2 is from the movie ‘Life of Pi’. It begins with a close-up shot of water in a public swimming pool and a boy quickly surfacing from it. Other swimmers can be seen in the background. The camera then turns down to look at the boy from above. Shortly after, the camera changes as if to be looking through the eyes of the boy and the viewer sees a man at the side of the pool kneeling down and offering his hand to the boy. The scene then changes to show the man carrying the boy while walking beside the pool. The camera slowly zooms out and after a short time the boy is thrown back into the pool. The last part of the video shows the boy in the pool with the camera being situated directly underneath the boy in the water. The audio is first of a man talking about his past when he was a boy and then changes to the man in the video when he is shown carrying the boy beside the pool. He is telling the boy how he should learn to swim (before he throws him into the pool). The 3D is strong in this clip especially when the entire pool can be seen (with other swimmers or onlookers).

It can be noticed that the 3D in this video also plays an important role in the attention grabbing of viewers. A significant difference is seen when the man offers his hand to the boy, which in 3D protrudes from the screen. In 2D everybody focusses on the man’s face while in 3D most people turn their gaze to the hand. This can be seen in Figure B.6 (c) and (d). Once again it can also be noticed that in 3D, viewers’ gaze is more focussed while in 2D it tends to be more dispersed. Figure B.6 (g) and (h) shows how at the beginning of the clip before the boy surfaces, in 3D people tended to look at one point in the scene while in 2D the lack of the depth dimension had viewers spread out their viewing. Similarly, when the man is carrying the boy beside the swimming pool, in 3D viewers tended to remain focussed on the two characters while in 2D they switched from viewing the man and boy and exploring the scene especially when the camera zoomed out. Figure B.7 (c) and (d) shows a significant difference in viewing behaviour where 2D viewing is more dispersed than 3D and Figure B.7 (g) and (h) shows an example where this no significant difference in viewing behaviour showing that in 2D viewers focussed on the two characters as well.

To summarise, we see in this video how 3D, when used effectively, can focus the viewer's attention on salient objects or how it can control it by moving objects out of the screen.

Video #3

Video #3 is from the animated movie 'Megamind'. It shows how Megamind (the villain of the movie) appears in a cloud of smoke and a laser show in the middle of a city square. Around the city square are citizens and policemen behind barriers. As Megamind appears the policemen draw their guns. Megamind skips into the crowd joyfully and the policemen out of fright drop their weapons to the ground. The audio is simply of a song playing in the background. All scene cuts have deep 3D in them.

Once again 3D plays an important role here in the attention grabbing of viewers. There is a similar scene to one in the previous video with the man's hand in which there is a close-up shot of a policeman who has his gun drawn. In 3D this gun protrudes from the screen. Just like in the previous video, in 3D most people turned their gaze to the gun while in 2D gazes were focussed on the man's face. This can be seen in Figure B.8 (c) & (d). However, viewers' gazes in 3D in this video were not quite as focussed as they were in the previous two. When Megamind skips towards the crowd, the camera is behind him and you can see the entire crowd in front of this character. In 2D everybody (without exception) focussed on the moving character while in 3D, participants explored the scene for a short time. This can be seen in Figure B.8 (g) & (h). The quirky look of Megamind managed to hold the attention in 2D but the additional dimension in 3D coupled with vibrant colours and fine details of the background dragged the attention away in 3D for a short time.

We learnt from these three videos, therefore, that it is possible to control gaze a lot more in 3D. For example, motion and depth can override the urge to explore (as we saw in video 1). The only exception (noticed in our set of videos) to this rule is when a scene's background is vibrant in colour and 'interesting'. In this case, the viewer may choose to divert their attention away from the main object in the scene. Also, if a sole object significantly protrudes from the screen, this will drag attention away in 3D - as was noticed in videos 2 & 3.

6.4.3 Videos with Little Movement

Video #4

Video #4 is from the documentary 'Ultimate Wave Tahiti'. The clip shows two men sitting on the beach talking about surfing. Behind the two people is a small beach shack, surf boards and a beach forest. The audio is of the two men talking.

By far the two most salient regions in this video are the faces of the two men - both in 2D and 3D. Our test for significant viewing difference found that 11% of the video was different in 2D compared to 3D. This difference was always due to a few members of one viewing group choosing to look more at one man rather than the other at a particular moment in time that appears to be random - the audio does not explain this either. This can be seen Figure B.9 (c) & (d) where more spread is present in 2D while in Figure B.9 (g) & (h) more spread is present in 3D. Apart from that no other reasons for differences were noticed so we can conclude that for videos such as this one (a simple conversation between two people with no camera movements) minimal differences between viewing behaviour will be exhibited.

It was mentioned in Section 6.1 that Häkkinen et al. [66] performed their 2D/3D analysis on clips with little camera movement. In fact, one of their clips was very similar to video #4 - of two people having a conversation together. They concluded that people explore a lot more with their eyes in 3D. As just stated, we found this to not be the case for videos of this nature. Table B.2 shows that the total number of fixations, average fixation count and total distance travelled by the eyes is very similar. This shows that Häkkinen et al.'s pre-screening request to their participants to analyse which version of the clip is better (2D or 3D) played the defining role in their results and, therefore, contaminated them by adding an artificial (i.e. not present in real viewing conditions) top-down effect.

Video #5

Video #5 is from the film 'Avatar'. The first part of the clip is from the view of a camera looking down at a man's face while he is lying down. The camera slowly zooms into his eyes. The scene then changes to show two views of one of his eyes before going back to the initial downward looking shot. In this last shot a few drops of water slowly appear out of the camera and drop down. The audio is of atmospheric music and the man's breathing.

In this clip the face of the character was still the most salient feature of the video. However, since the audio was not of an engaging nature, a little bit of exploring took place also. The exploring was very similar in 2D and 3D. Nearly the same amount of fixations, fixation duration and distance travelled was exhibited in 2D and 3D (cf. Table B.2). There was one scene where the viewing behaviour was noticeably different in 3D, however. When the drops of water appear in the second half of the clip, in 3D they protrude from the scene and become salient and fixated upon. First the two drops in the middle of screen (that combine to make one drop) are noticed both in 2D and 3D (Figure B.10 (a)-(d)). Then the drop on the bottom right of the screen slowly appears. It is fixated on strongly in 3D (despite never completely coming into focus) but never in 2D (Figure B.10 (e)-(h)). Finally the drop on the left appears. It is fixated on strongly in 3D and a little less in 2D (Figure B.11 (a)-(d)). This drop comes into focus as the one in the middle and right drop out. You can notice, therefore, that 3D plays a role here as the 3 protroduing objects (the droplets) are heavily fixated on. In 2D only the objects in focus are looked at and even then, not so strongly. This shows that also in scenes with little camera/object movement, depth can be used to control viewers' attention in 3D.

Video #6

Video #6 is from the documentary entitled 'Into the Arctic'. It shows two deers lying on the ground amongst bushes. The audio is of a general nature - a soothing song playing in the background.

This clip did not have any faces or humans present nor a single source of audio. Because of this significantly more exploring took place in 3D. In the two previous videos, the two conversing people and the face (and eye) of the man managed to reign in the ‘urge’ to look around. The deers were not considered as interesting, even the young one. Table B.2 shows that there was 10% more distance travelled by the eye as well as more fixations and a shorter average fixation length. This shows that in 3D when there are no (or very little) camera movements, if the director of a film wants to keep viewers focussed on 1 or 2 objects he/she needs to make sure that there is either an interesting conversation taking place between people or to make important objects stand out in 3D space.

6.4.4 Videos with Fast Movement and Changes

The following sections present an analysis of the viewing behaviour on the last two videos in the dataset.

Video #7

Video #7 is from the 2010 film ‘Alice in Wonderland’. The 40 second long clip is of Alice falling down a hole with objects constantly moving past her and the camera view changing every few seconds. Alice is also spinning around throughout the clip. The wide array of objects flashing past (like pianos, books, lamps, etc.) adds to an atmosphere of confusion. The audio is of Alice screaming for the duration of her fall and of an orchestra playing a rhythmic piece to suit the mood.

There was very little difference in viewing behaviour between the 2D and 3D versions of this clip. The fast changing scene gave participants little time to grasp anything concrete and most of the gazes were focussed on the middle of Alice while she was falling both in 2D and 3D. There is a significant difference (according to our t-test) when Alice is first shown to the viewers (frames 80-115). During this time nearly all of the 2D participants were focussed on the middle of the screen. The 3D participants, however, took the opportunity to examine the person that had just appeared to them - see Figure B.12 (a)-(d). Once the character is grasped as being a young girl, the differences between 2D and 3D become minimal. The viewing behaviour for most of the frames flagged as being significantly different by our t-test appears to be erratic and random in nature (sometimes there is a little more spread in 2D and sometimes in 3D, for example) - this would coincide with the fast-paced scene. None of the objects that fly past Alice are focussed on despite some of them being prominent in 3D - they move past too quickly. The scene does slow down a little, relatively speaking, two times. Once when a big piano comes into scene (frames 495-575). First the camera shows it moving close to Alice's face who subsequently tries to protect herself from it with her arm. Then there is a close-up of its keys being invisibly pressed. During this time no significant differences in 2D and 3D were recorded. The last 'slow' scene is at the very end of the clip. Alice falls onto a bed, bounces off it and then is seen falling deeper into the hole with the camera not following. There is a small difference in viewing behaviour when Alice falls on the bed. In 2D the middle of the bed is the main area of interest. In 3D some of the participants followed Alice down onto it. Figure B.12 (e)-(h) shows this.

Video #8

Video #8 is from the film 'Life of Pi'. The clip shows Pi (the protagonist of the film) inside a lifeboat while it is being tossed around at sea by big waves at night. There is a large ship in close proximity that can be seen at the beginning and end of the scene. The camera follows Pi and the lifeboat erratically and changes angles frequently. The audio is of Pi struggling to remain in the boat and the waves crashing around him.

As with the previous video, there is little here to distinguish the viewing behaviour between the groups who saw the 2D and 3D versions of the clip. The salient object was Pi throughout the scene and viewers faithfully followed his struggles with the waves. There were only a few groups of frames where the viewing was flagged as being significantly different - it is difficult, however, to determine the reasons for this and, as with the previous video, appear to be random (to suit the erratic nature of the video, perhaps). The first is when the boat goes up a wave and all we see is the side and bottom of it - Pi disappears for a short time (frames 160-180). The 2D viewing pattern was a little more dispersed than the 3D one (Figure B.13 (a)-(d)). The second scene is when the boat is seen from a distance riding a wave (frames 265-300). Once again, viewing behaviour in 3D is more focussed (Figure B.13 (e)-(h)). The last two groups of frames flagged for being significantly different belong to the one scene: the boat is riding on top of a wave with the camera positioned at the back of it (frames 430-493). This scene is relatively less 'frantic' than the other parts of the clip. This time, however, the 3D viewing pattern is the one that is more dispersed (Figure B.14

(a)-(d)). As with the other two scenes flagged as having significantly different frames, the dispersion is minimal. No major exploring took place, i.e. the focus still remained on the salient object.

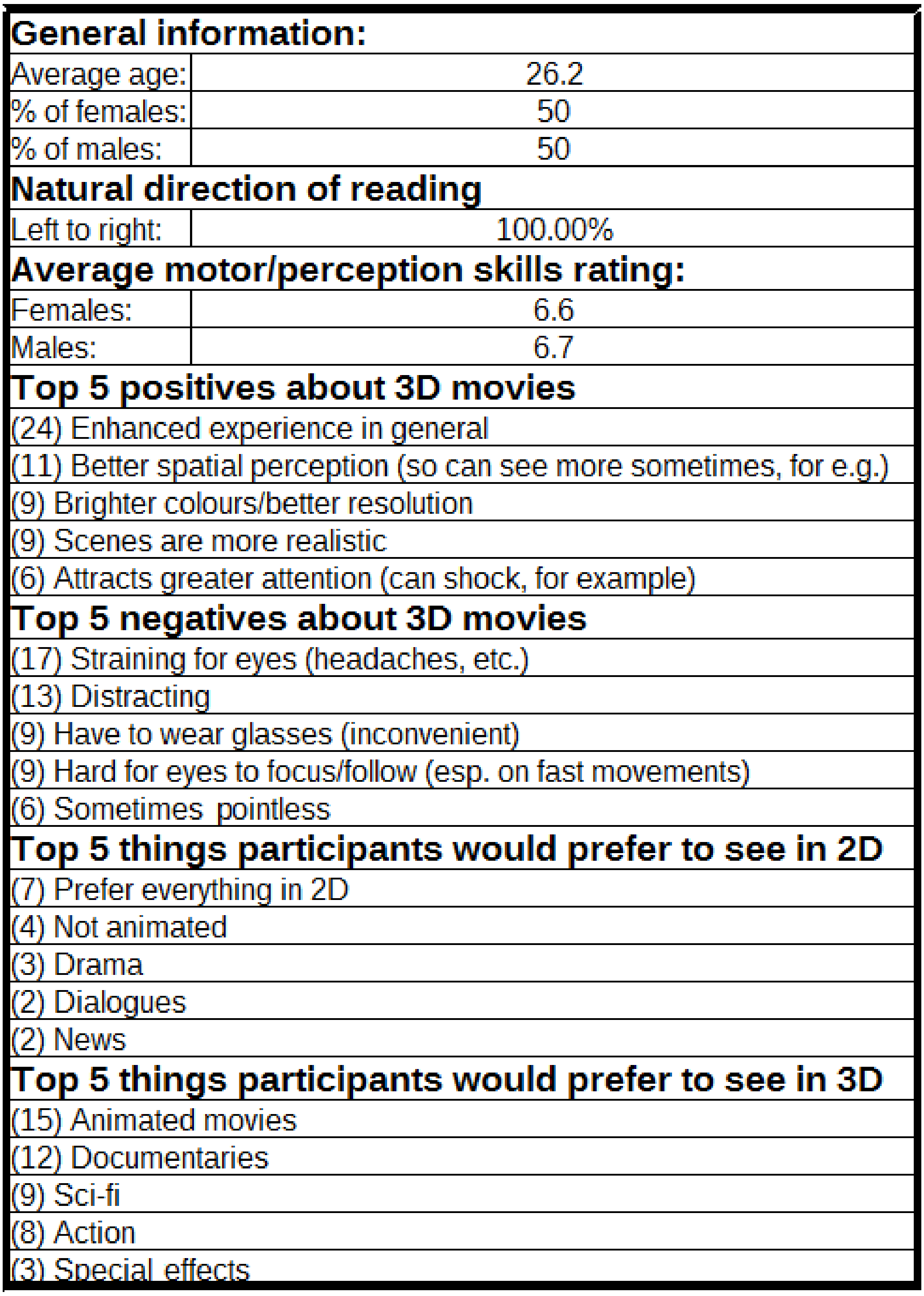
We saw in these last two videos, therefore, that there is very little difference between 2D and 3D viewing behaviour when a scene has fast changes throughout it. Although the general ambience can be altered with the addition of an extra viewing dimension (e.g. objects can be caught by peripheral vision flying at the user through the screen) it is a lot easier for film makers to predict where people are going to look - if they can predict this in 2D then chances are it will be the same in 3D. Even when a scene, relatively speaking, slows down for a short amount of time, viewers will not explore the scene too much because they know (subconsciously) that they have to remain focussed as things may change quickly any second.

6.4.5 Questionnaire

A summary of results from the questionnaire can be seen in Figure 6.2. There are a number of positive elements for 3D films that can be extracted from this questionnaire. For example, nearly half of the participants believe that the additional dimension enhances the viewing experience and over 20% of participants believe that better spatial perception is possible in 3D. Also, just under 20% of participants believe that films in 3D have brighter colours and/or better resolution and that scenes are more realistic in this domain. There were, however, a number of criticisms of 3D films, the main one (36%) being that 3D viewing is straining on the eyes or inconvenient (18%) due to the need to wear glasses. Interestingly, 18% of people stated that objects in 3D are hard to focus on or follow especially when fast camera or object movement is occurring. In Section 6.4.4 we came to the conclusion that the difference in 2D/3D viewing behaviour is minimal. Coupled with what participants said here, this could be an argument to make 3D effects minimal or non-existent in such fast-paced scenes (especially if it is going to cause strain). Also, a major criticism of 3D films is that they can be distracting (26%). We saw, for example, in our analysis of videos #2, #3 and #5 that attention can shift to salient objects protruding from the screen. Such 3D effects can be deemed to be distracting if they can break a person's immersion in the story of a film at the expense of thinking about 3D 'gimmicks'. These effects should, perhaps, be left to films like animations.

6.5 Conclusion

This chapter presented the large-scale acquisition of a 2D/3D video dataset along with accompanying eye tracking data from 50 people. Our method of kriging interpolation was used on the eye tracking data and subsequently an analysis of participants' viewing behaviour was performed. This analysis was supported by a novel Gaussian mixture model for computing eye tracking heat maps that takes advantage of the uncertainties associated with our eye tracking data. The eye tracking dataset will be made public and will be the first such dataset in the public domain. Finally, results from the questionnaire were also presented. This chapter represents a total of 12 months of work and results from it (e.g. that motion and depth combined can be the main drawing factor of attention or that there is minimal difference in viewing behaviour on fast-paced scenes) can be used to advance the field of visual saliency in 3D. Future work presented in the next chapter will delineate this further.



General information:	
Average age:	26.2
% of females:	50
% of males:	50
Natural direction of reading	
Left to right:	100.00%
Average motor/perception skills rating:	
Females:	6.6
Males:	6.7
Top 5 positives about 3D movies	
(24) Enhanced experience in general	
(11) Better spatial perception (so can see more sometimes, for e.g.)	
(9) Brighter colours/better resolution	
(9) Scenes are more realistic	
(6) Attracts greater attention (can shock, for example)	
Top 5 negatives about 3D movies	
(17) Straining for eyes (headaches, etc.)	
(13) Distracting	
(9) Have to wear glasses (inconvenient)	
(9) Hard for eyes to focus/follow (esp. on fast movements)	
(6) Sometimes pointless	
Top 5 things participants would prefer to see in 2D	
(7) Prefer everything in 2D	
(4) Not animated	
(3) Drama	
(2) Dialogues	
(2) News	
Top 5 things participants would prefer to see in 3D	
(15) Animated movies	
(12) Documentaries	
(9) Sci-fi	
(8) Action	
(3) Special effects	

Figure 6.2: Summary of results from questionnaire.

Chapter 7

Conclusion

7.1 Summary

In this thesis we showed that CMAs can be used to assist and improve content-based image retrieval systems. With a single threshold we controlled the number of features, taken from salient regions, that are used to classify an image. We show that even with a small number of features, the classifiers trained to find the objects of interest still perform well.

Our second contribution is a new way to incorporate depth into visual saliency calculations. Using this method we extend a state-of-the-art CMA and show that the new 3D CMA outperforms other common 2D algorithms as well as their state-of-the-art 3D extensions on 3D media.

The most common way to evaluate CMAs is to compare their output to eye movements from humans that are captured by eye trackers. Eye trackers, however, lack in accuracy. Our third contribution is a novel method for capturing eye tracking data that improves the accuracy of eye trackers, provides a measure of uncertainty for all captured data and also significantly eases the eye tracking recording process. Improving the accuracy of eye trackers has financial benefits also since top-of-the-range eye trackers are very expensive. It is frequently infeasible, therefore, to constantly upgrade eye trackers to newer more accurate models when they become available.

As a final contribution, we conducted a large-scale acquisition of a 2D/3D video dataset along with accompanying eye tracking data from 50 people. This dataset will be made public and will be the first such dataset in the public domain. It will now be possible to evaluate 3D CMAs on videos. We also present an analysis of the difference in viewing behaviour between the 2D and 3D videos. We find, for example, that motion and depth combined can be the main drawing factor of attention or that there is minimal difference in viewing behaviour on fast-paced scenes. Taking advantage of the uncertainties associated with our eye tracking data, we also propose a novel Gaussian mixture model for computing eye tracking heat maps. The contributions mentioned here have formed a foundation for the future improvement of 3D visual attention and CMAs.

7.2 Future Work

The fledgling world of CMA in 3D is an exciting one with vast opportunities for new and pioneering research. Even the work presented in our thesis can be extended in many ways. Our new depth incorporating method could be tested on many more 2D state-of-the-art CMAs. We improved on Wang et al.'s [173] journal paper but the ultimate goal with research in our field is to improve 3D visual saliency. Using and assessing these new 3D CMAs in real applications, such as thumbnail creation or vehicle/robot navigation, is one such sure way of reaching this goal.

More eye tracking datasets could also be created for 3D images and videos. Only one of each now exist compared to 22 in total for 2D. The more datasets with varying images and videos that exist the better that CMAs can be evaluated and fine-tuned before they are used in real-life applications. The large number of 2D eye tracking datasets have allowed 2D visual saliency to steadily improve. The same should be true for 3D visual saliency in the near future. Any new image datasets created should also keep in mind the centre-bias discussed in Section 2.4.6. This was not explicitly considered in the dataset of Wang et al.'s [173] but is an important challenge in image eye tracking dataset creation.

It would be useful to also perform some more accuracy analysis with respect to kriging and eye tracking. For example, one could analyse the trade-off between the number of calibration points in a calibration phase and the accuracy obtained. Fewer calibration points would speed up the eye tracking process even more. One could also analyse the effect of decreasing the number of calibration phases in an experiment or even increase the time between calibration phases. More calibration phases would produce a better variogram but, once again, at the expense of inflating the experiment time and perhaps participant discomfort.

There is potential for a lot more future work to be performed with the new eye tracking film dataset that we created as part of this thesis. One obvious extension is the creation of a spatio-temporal 3D CMA. No such algorithm currently exists. To assist in this, even more statistical analysis could be performed on the difference in viewing behaviour between the 2D and 3D videos.

More research on covert and overt attention in 3D is also necessary. It was shown that, in general, there is little difference in viewing behaviour in 2D/3D with fast-paced scenes. If people do not look differently in these cases, how necessary is it, then, to embellish scenes with 3D outside of salient regions? How does covert attention work in 3D in this respect?

Another possibility is to see if there is a difference in where people look based on culture, gender or age in 3D. There is preliminary material for this available with the answers from our questionnaire. We were hoping, for example, to analyse the viewing patterns of people who have different native directions of reading. Unfortunately, all of our observers naturally read from left to right but analysis like this has never been performed whether in 2D or 3D. With respect to gender, we conveniently managed to obtain an equal number of females and males to perform our experiment. Analysis of this kind, therefore, is ready to be performed. Clearly there are mountains of potential future work possible with our new 3D video eye tracking dataset.

Our dataset comprises of material that was professionally made for 3D viewing devices. We can assume that directors and cinematographers have been specially trained for this medium. Further research, then, could be performed on amateur 3D videos that could contain artefacts specific to the 3D domain in their clips. A recent satelitte project of ours has been to analyse and summarise social media videos (e.g. Zdziarski et al. [187] and Zdziarski et al. [189]). Glasses-free 3D mobile devices are slowly entering the market (e.g. the LG Optimus 3D P920) so it is only a matter of time before social media is flooded with 3D videos. Research in visual attention in this area is becoming a necessity.

Appendix A

Questionnaire for Experiment

Each question is optional. Feel free to omit a response to any question; however we would be grateful if all questions are responded to.

Questionnaire for experiment

Age: ……..

Are you male or female? □ Male □ Female

What is your natural (native) direction of reading?

□ Left to right
□ Other: ___________

On a scale of 1 to 10 (1 = poor, 10 = excellent), how would you rate your motor/perception skills in sports/activities such as juggling, tennis, basketball, football, etc.? ____________

Are you aware of any impairments that you may have related to visual perception? (e.g. colour blindness, problems with depth perception)

□ Yes
□ No

Do you wear glasses/contacts? □ Yes □ No

Can you give us 3 positive things about 3D movies and 3 negative things?

●______________________________ ●______________________________
●______________________________ ●______________________________
●______________________________ ●_______________________________
____
____

What kind of movies/information would you prefer to see in 2D or 3D?

____
____
____
____
____

Would you like to make any comments about the experiment you just participated in?

____
____
____

Appendix B

Eye Position Density Maps and Statistic Tables for 2D/3D Perception Experiment

Video	Avg. Fix. Length	Avg. # Fix.	Avg. # Sacc.	Avg. Sacc. Ampl.	Avg. Sacc. Vel.	Avg. Sacc. Peak Vel.	Avg. Dist
2D							
1	378	44	50	6.39	0.0473	257.1	$3.3\times10e^6$
2	448	37	41	4.93	0.0423	241.2	$1.5\times10e^6$
3	391	48	54	4.71	0.0395	230.0	$2.3\times10e^6$
3D							
1	402	41	50	6.55	0.0443	250.8	$3.5\times10e^6$
2	406	41	45	5.47	0.0424	259.1	$1.7\times10e^6$
3	390	48	53	4.80	0.0391	237.4	$2.4\times10e^6$

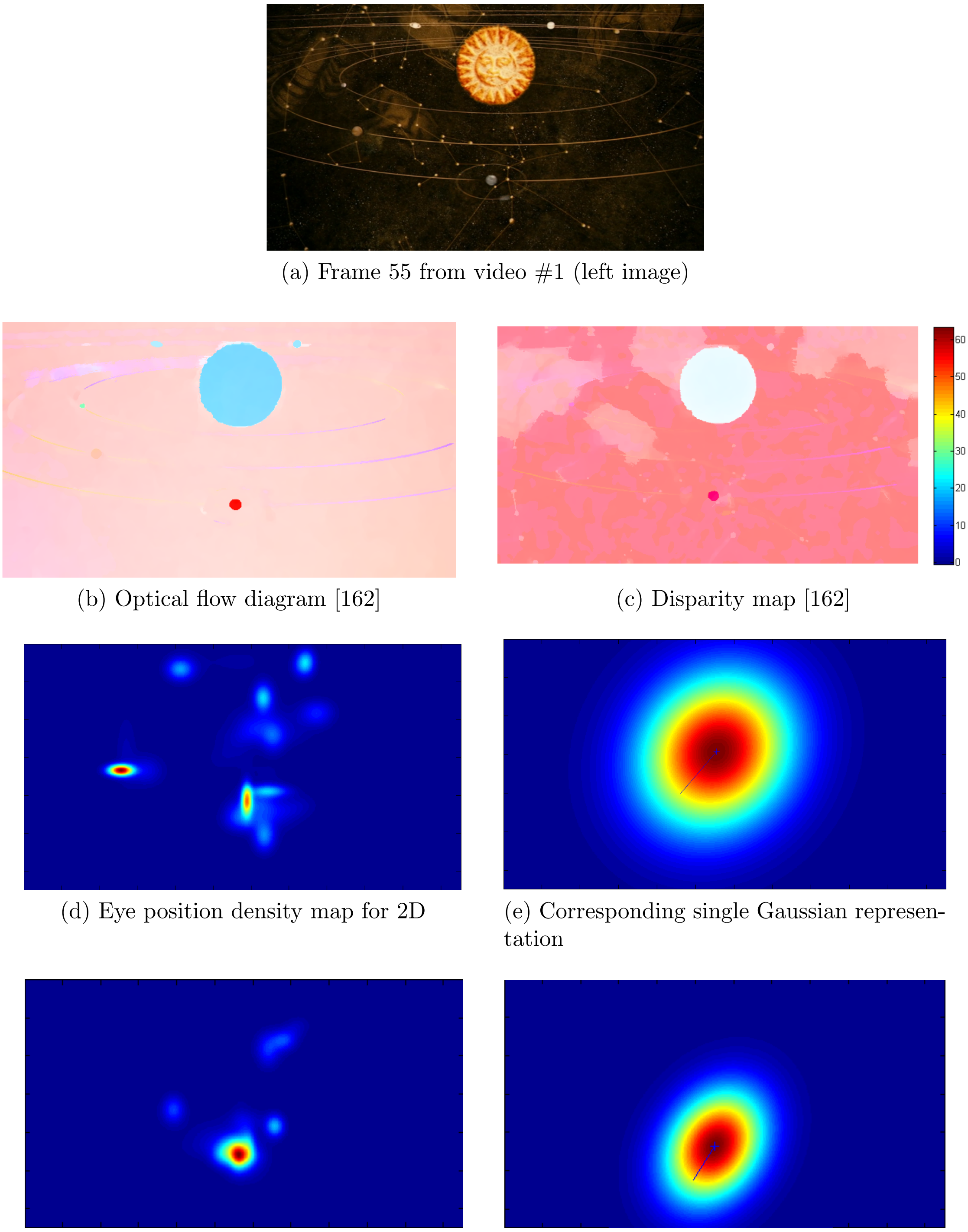
Table B.1: Eye tracking statistics from the left eye summarising 2D and 3D eye movements from the videos in group 1 (strong 3D). Euclidean distance is in pixels.

Video	Avg. Fix. Length	Avg. # Fix.	Avg. # Sacc.	Avg. Sacc. Ampl.	Avg. Sacc. Vel.	Avg. Sacc. Peak Vel.	Avg. Dist
2D							
4	407	32	37	5.61	0.0427	239.2	$2.1\times10e^6$
5	458	52	58	5.65	0.0477	245.2	$2.7\times10e^6$
6	402	36	40	5.34	0.0428	249.0	$2.9\times10e^6$
3D							
4	415	32	36	5.71	0.0431	233.8	$2.1\times10e^6$
5	441	54	59	5.78	0.0449	255.4	$2.6\times10e^6$
6	388	40	43	5.61	0.0443	256.4	$3.2\times10e^6$

Table B.2: Eye tracking statistics from the left eye summarising 2D and 3D eye movements from the videos in group 2 (little or no camera movement). Euclidean distance is in pixels.

Video	Avg. Fix. Length	Avg. # Fix.	Avg. # Sacc.	Avg. Sacc. Ampl.	Avg. Sacc. Vel.	Avg. Sacc. Peak Vel.	Avg. Dist
2D							
7	505	73	81	4.41	0.0433	242.8	$3.3\times10e^6$
8	533	35	38	4.33	0.0427	243.1	$2.1\times10e^6$
3D							
7	487	76	86	4.35	0.0435	244.8	$3.4\times10e^6$
8	514	36	39	4.69	0.0443	242.4	$2.3\times10e^6$

Table B.3: Eye tracking statistics from the left eye summarising 2D and 3D eye movements from the videos in group 3 (videos with fast movement and changes). Euclidean distance is in pixels.



(a) Frame 55 from video #1 (left image)

(b) Optical flow diagram [162]

(c) Disparity map [162]

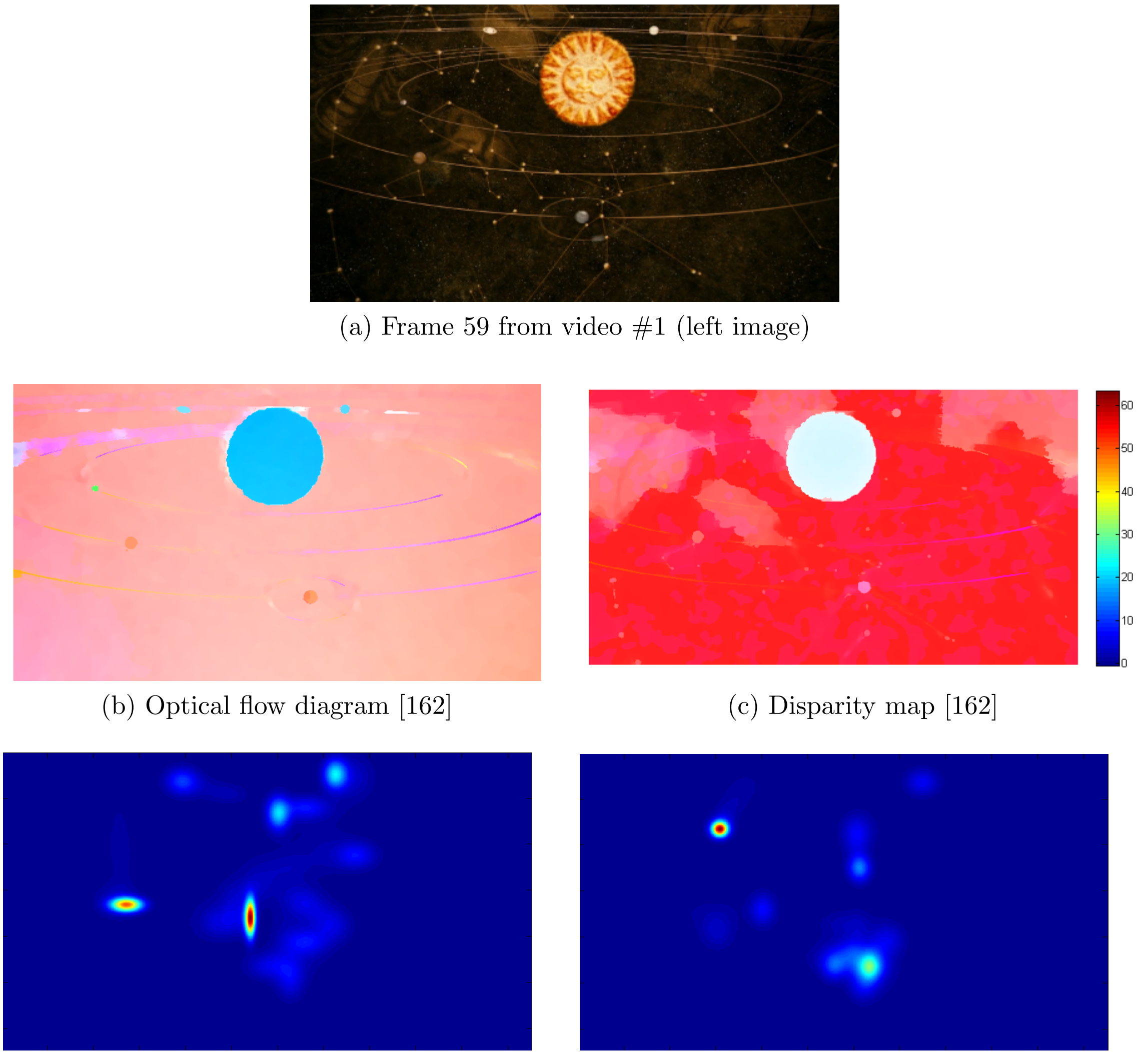
(d) Eye position density map for 2D

(e) Corresponding single Gaussian representation

(f) Eye position density map for 3D

(g) Corresponding single Gaussian representation

Figure B.1: (a) Frame 55 from video #1; (b) optical flow for this and next frame; (c) disparity map; (d) & (f) eye position density maps in 2D and 3D; (e) & (g) corresponding single Gaussian representations. The difference between the Gaussians is significant.



(a) Frame 59 from video #1 (left image)

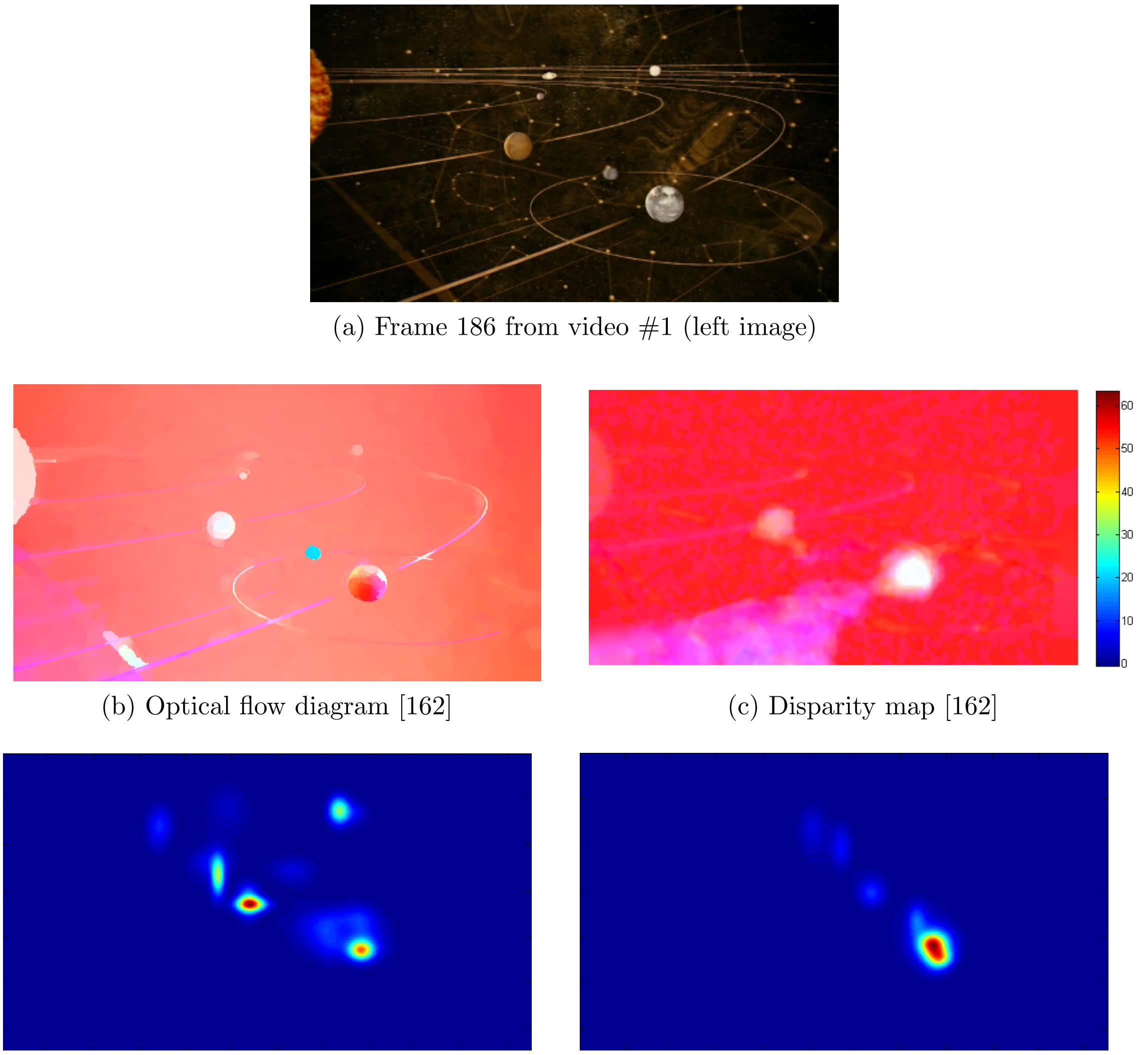
(b) Optical flow diagram [162]

(c) Disparity map [162]

(d) Eye position density map for 2D

(e) Eye position density map for 3D

Figure B.2: (a) Frame 59 from video #1; (b) optical flow for this and next frame; (c) disparity map; (d) & (e) eye position density maps in 2D and 3D.



(a) Frame 186 from video #1 (left image)

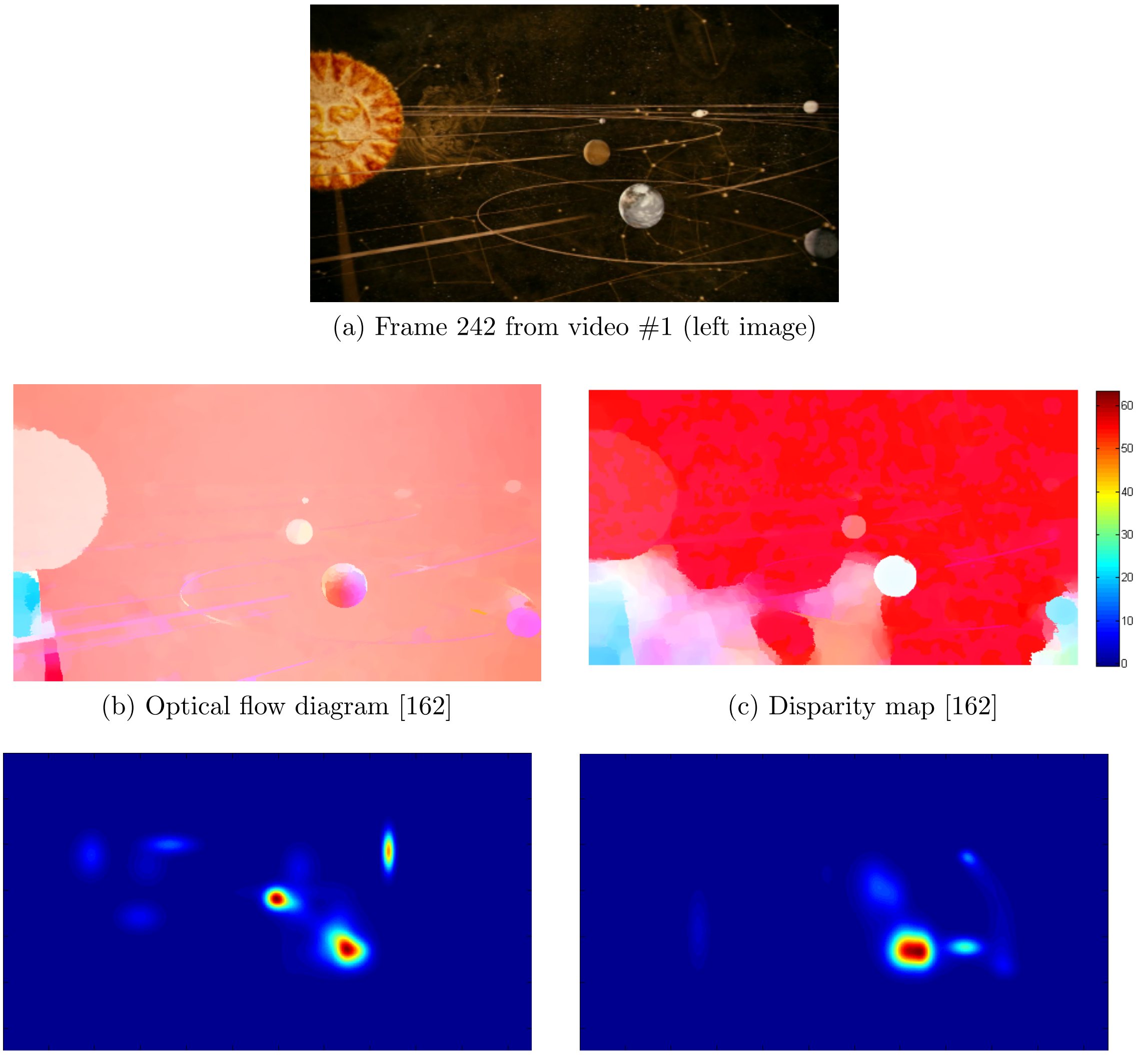
(b) Optical flow diagram [162]

(c) Disparity map [162]

(d) Eye position density map for 2D

(e) Eye position density map for 3D

Figure B.3: (a) Frame 186 from video #1; (b) optical flow calculation for this frame and the next; (c) disparity map for this frame; (d) & (e) eye position density maps in 2D and 3D for this frame.



(a) Frame 242 from video #1 (left image)

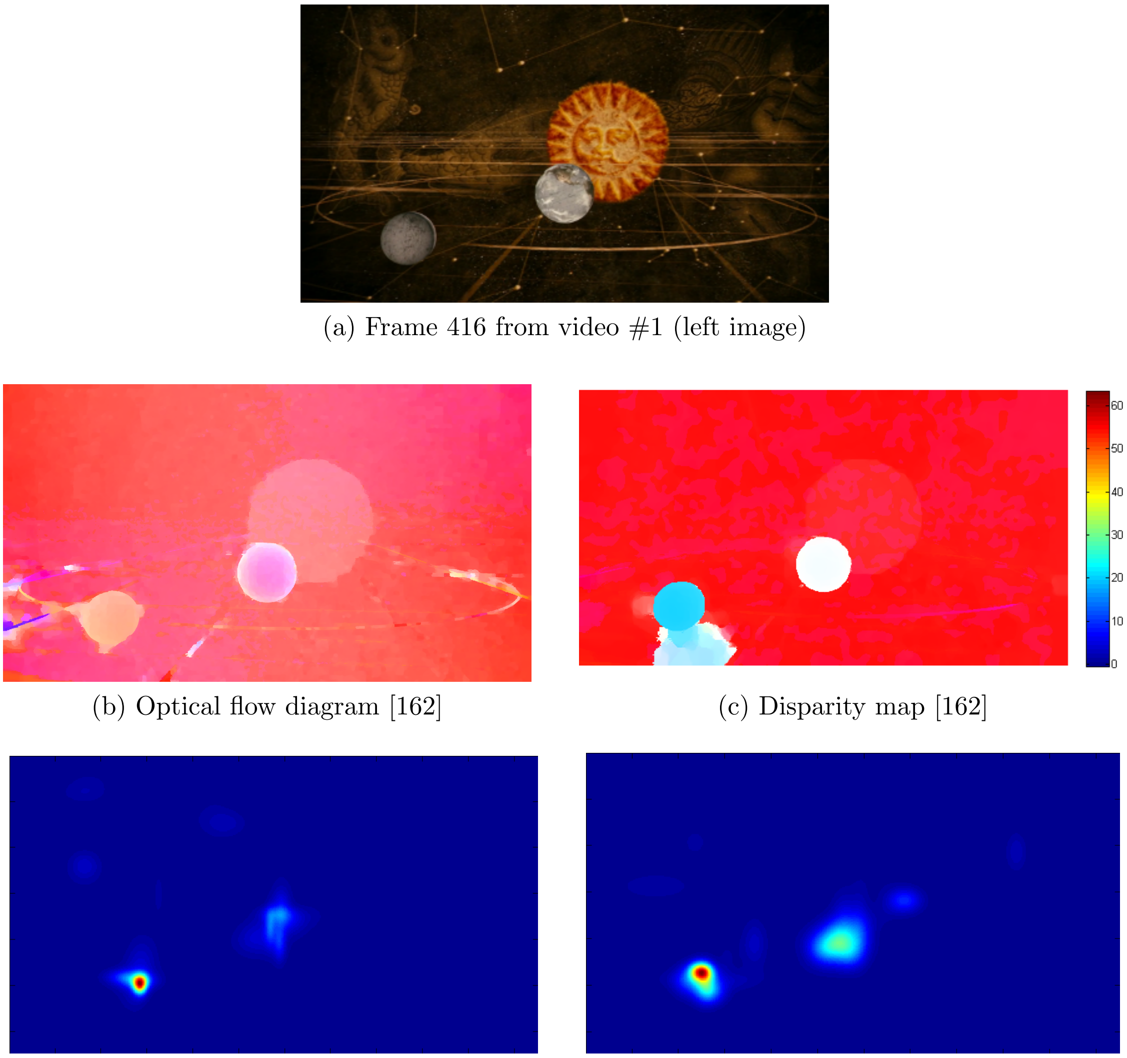
(b) Optical flow diagram [162]

(c) Disparity map [162]

(d) Eye position density map for 2D

(e) Eye position density map for 3D

Figure B.4: (a) Frame 242 from video #1; (b) optical flow calculation for this frame and the next; (c) disparity map for this frame; (d) & (e) eye position density maps in 2D and 3D for this frame.



(a) Frame 416 from video #1 (left image)

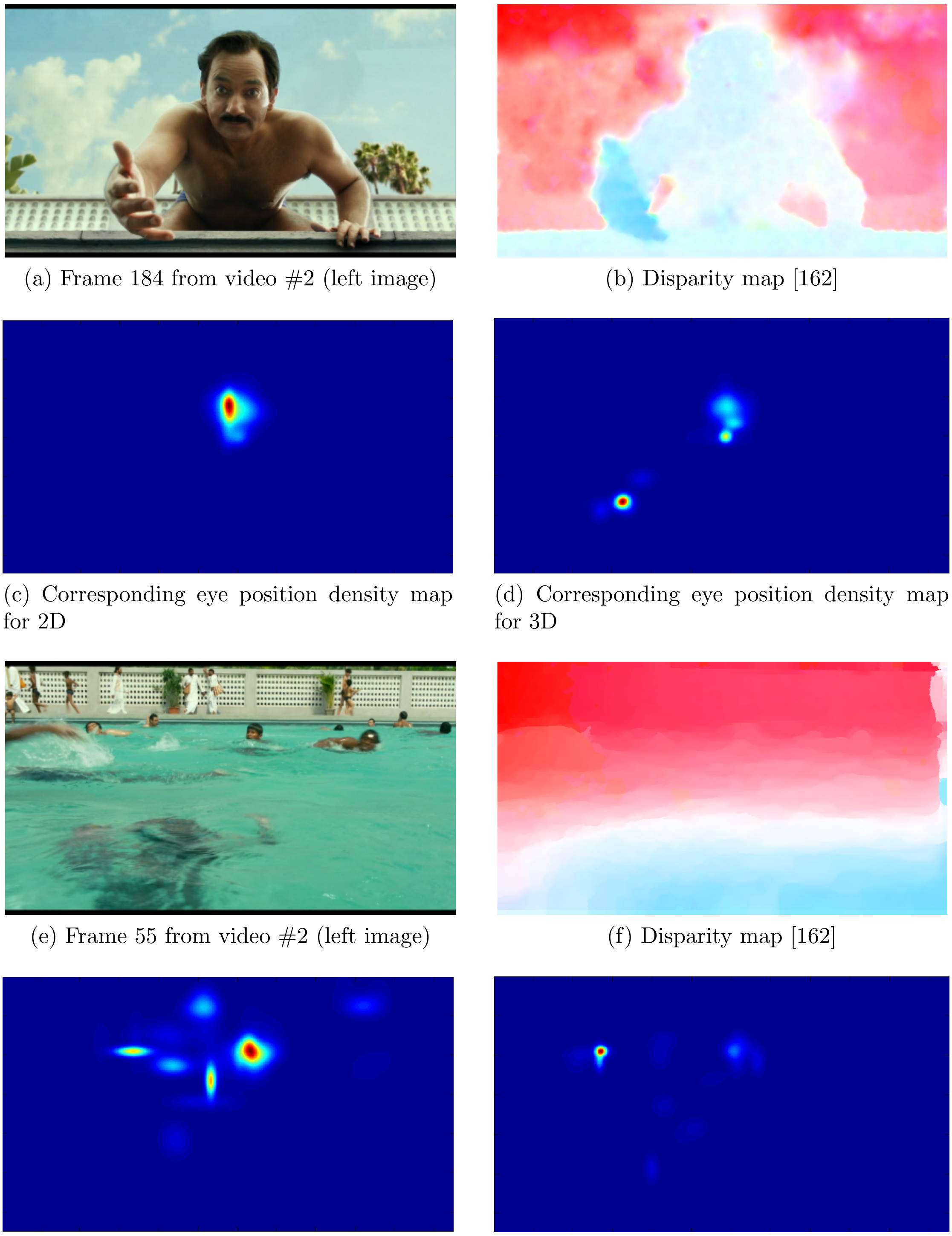
(b) Optical flow diagram [162]

(c) Disparity map [162]

(d) Corresponding eye position density map for 2D

(e) Corresponding eye position density map for 3D

Figure B.5: (a) Frame 416 from video #1; (b) optical flow calculation for this and next frame; (c) disparity map for this frame; (d) & (e) eye position density maps in 2D and 3D for this frame.



(a) Frame 184 from video #2 (left image)

(b) Disparity map [162]

(c) Corresponding eye position density map for 2D

(d) Corresponding eye position density map for 3D

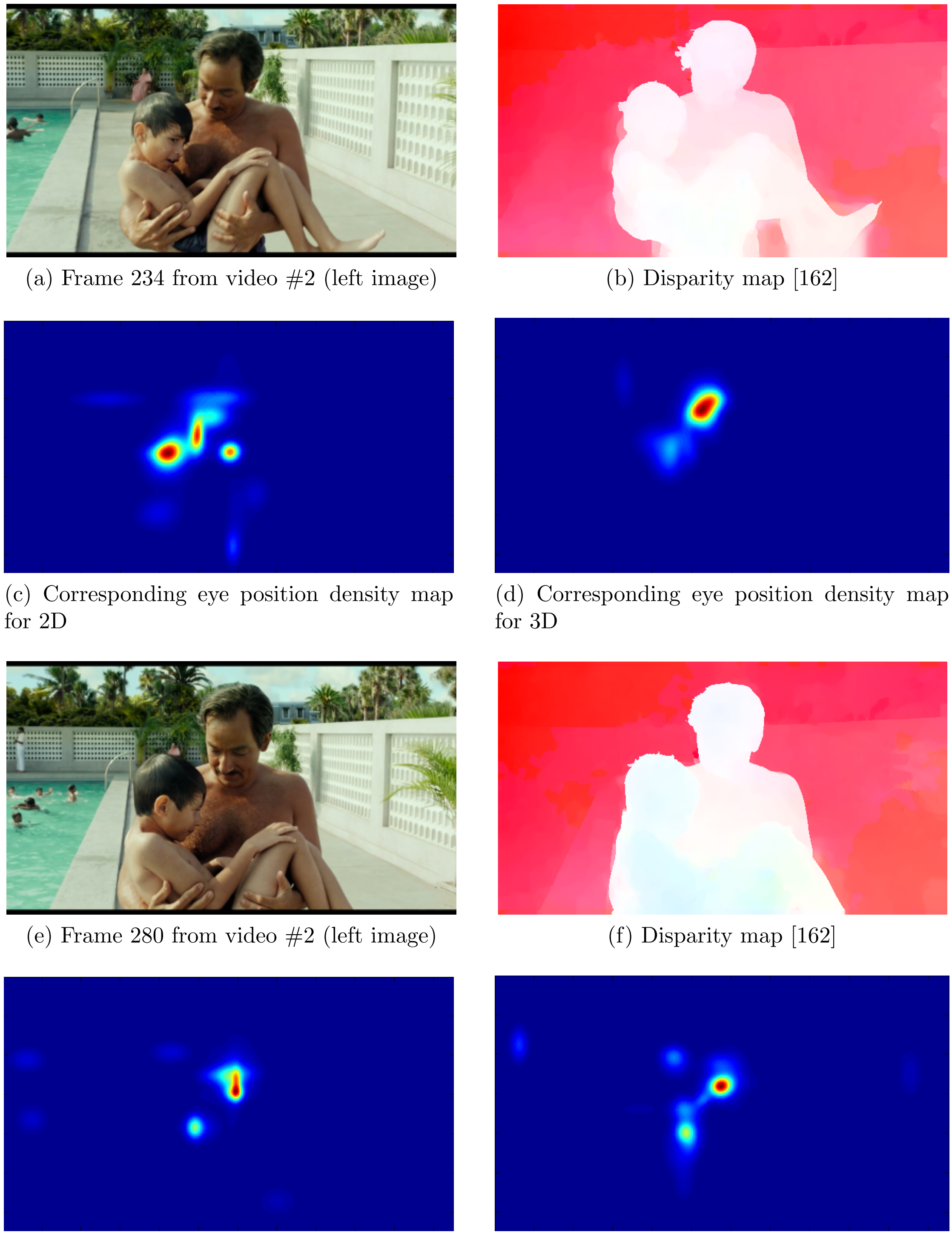
(e) Frame 55 from video #2 (left image)

(f) Disparity map [162]

(g) Corresponding eye position density map for 2D

(h) Corresponding eye position density map for 3D

Figure B.6: (a) Frame 184 from video #2; (b) disparity map; (c) & (d) eye position density maps in 2D and 3D. (e) Frame 55 from video #2; (b) disparity map; (g) & (h) eye position density maps in 2D and 3D.



(a) Frame 234 from video #2 (left image)

(b) Disparity map [162]

(c) Corresponding eye position density map for 2D

(d) Corresponding eye position density map for 3D

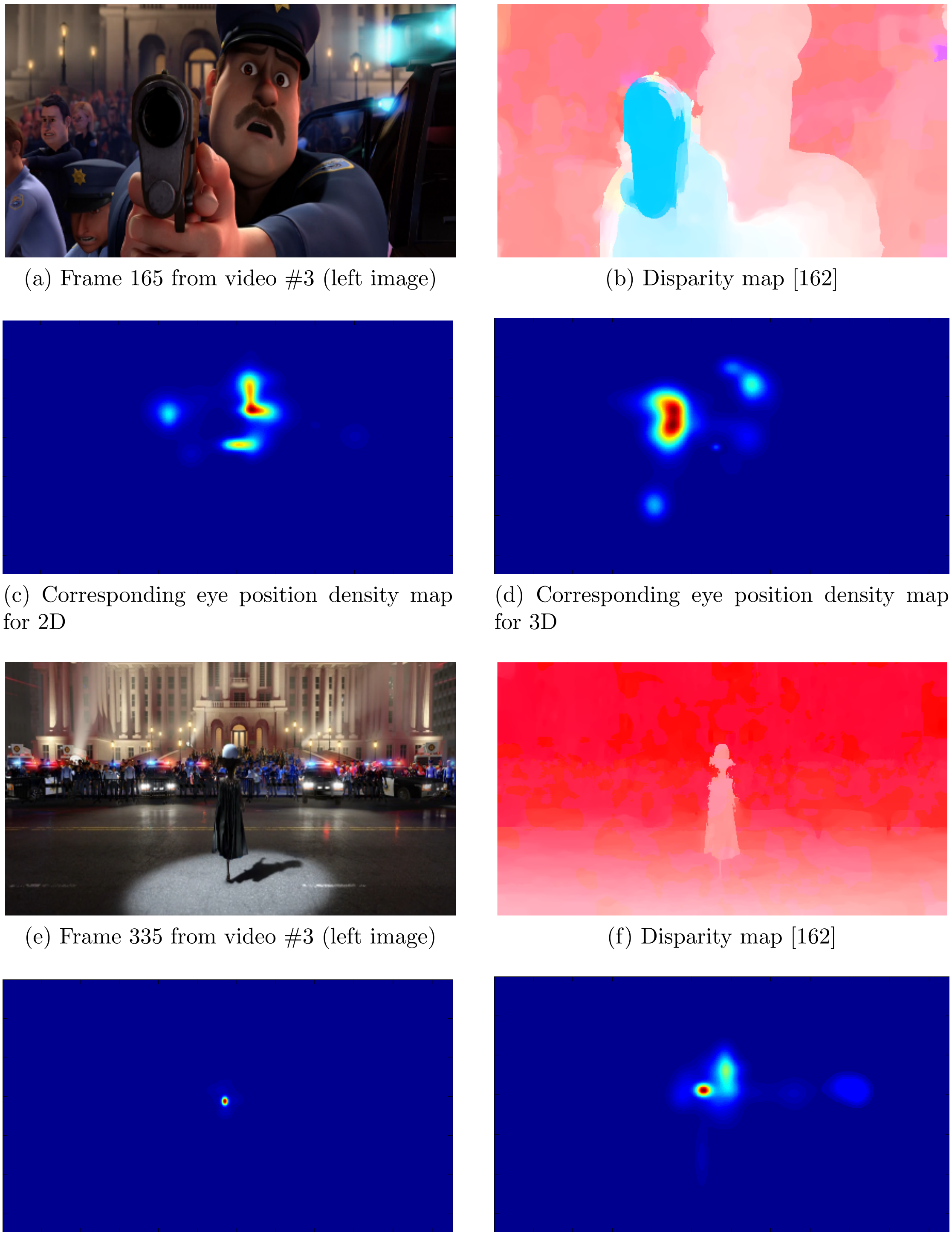
(e) Frame 280 from video #2 (left image)

(f) Disparity map [162]

(g) Corresponding eye position density map for 2D

(h) Corresponding eye position density map for 3D

Figure B.7: (a) Frame 234 from video #2; (b) disparity map; (c)&(d) eye position density maps in 2D and 3D. (e) Frame 280 from video #2; (b) disparity map; (g)&(h) eye position density maps in 2D and 3D.



(a) Frame 165 from video #3 (left image)

(b) Disparity map [162]

(c) Corresponding eye position density map for 2D

(d) Corresponding eye position density map for 3D

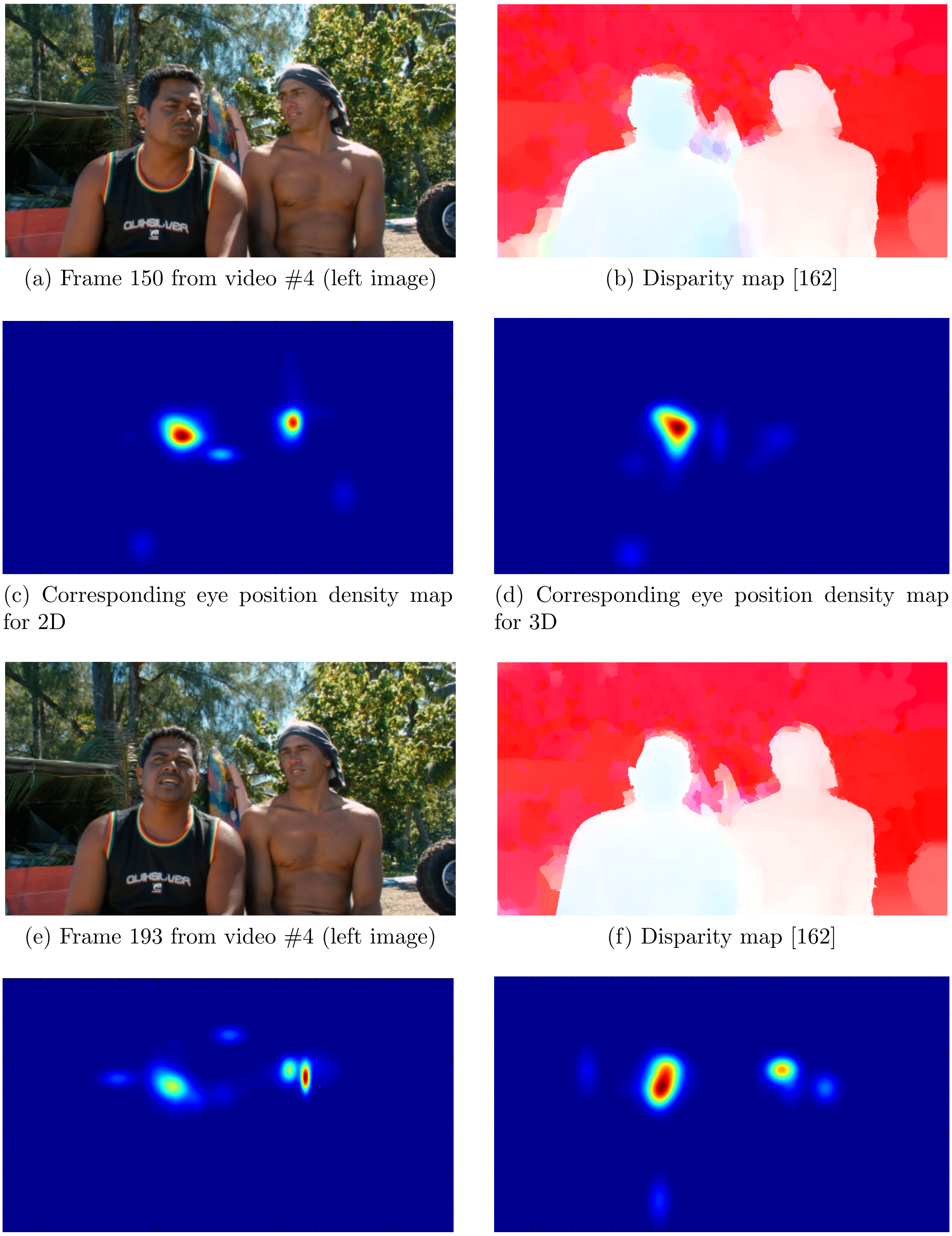
(e) Frame 335 from video #3 (left image)

(f) Disparity map [162]

(g) Corresponding eye position density map for 2D

(h) Corresponding eye position density map for 3D

Figure B.8: (a) Frame 165 from video #3; (b) disparity map; (c)&(d) eye position density maps in 2D and 3D. (e) Frame 335 from video #3; (b) disparity map; (g)&(h) eye position density maps in 2D and 3D.



(a) Frame 150 from video #4 (left image)

(b) Disparity map [162]

(c) Corresponding eye position density map for 2D

(d) Corresponding eye position density map for 3D

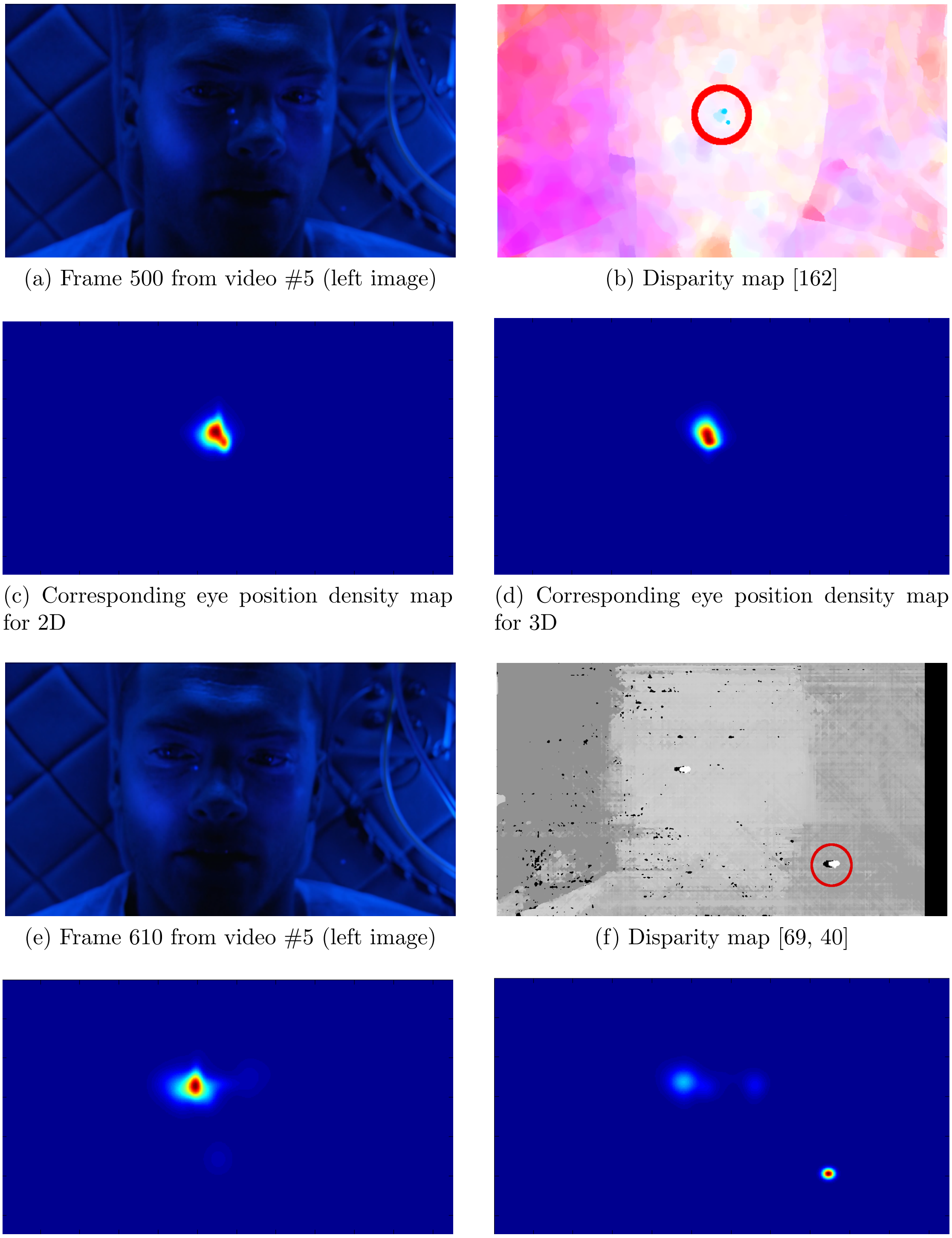
(e) Frame 193 from video #4 (left image)

(f) Disparity map [162]

(g) Corresponding eye position density map for 2D

(h) Corresponding eye position density map for 3D

Figure B.9: (a) Frame 150 from video #4; (b) disparity map; (c)&(d) eye position density maps in 2D and 3D. (e) Frame 193 from video #4; (b) disparity map; (g)&(h) eye position density maps in 2D and 3D.



(a) Frame 500 from video #5 (left image)

(b) Disparity map [162]

(c) Corresponding eye position density map for 2D

(d) Corresponding eye position density map for 3D

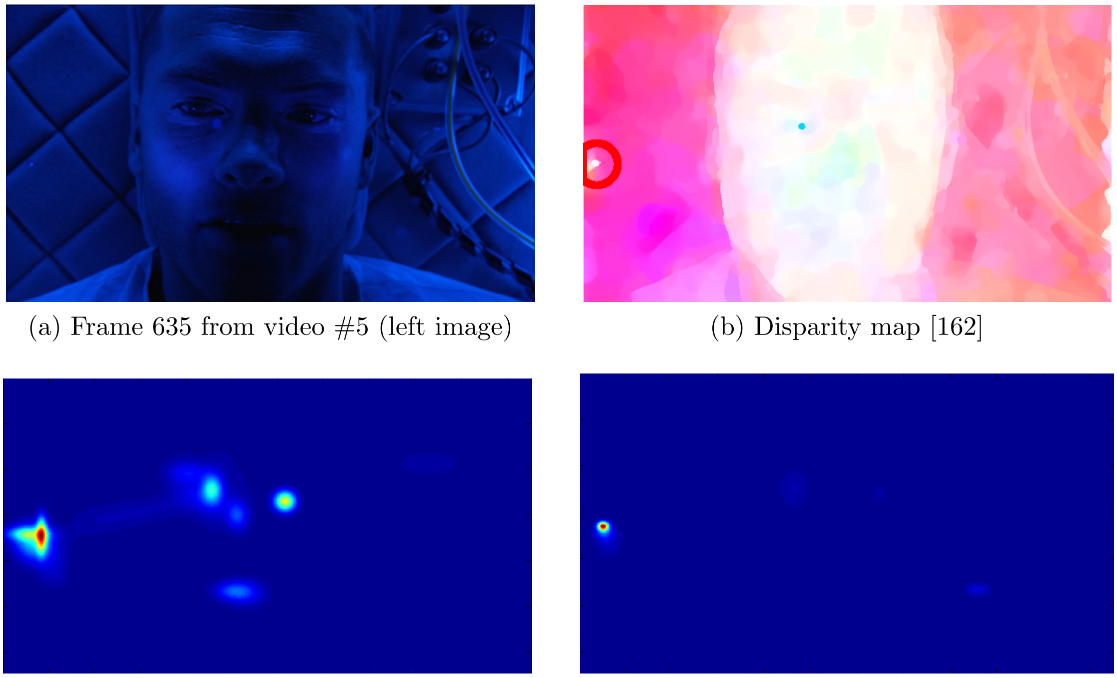
(e) Frame 610 from video #5 (left image)

(f) Disparity map [69, 40]

(g) Corresponding eye position density map for 2D

(h) Corresponding eye position density map for 3D

Figure B.10: (a) Frame 150 from video #5; (b) disparity map with new drop highlighted; (c)&(d) eye position density maps in 2D and 3D. (e) Frame 610 from video #5; (b) disparity map with new drop highlighted; (g)&(h) eye position density maps in 2D and 3D. A different disparity map extraction algorithm is used in (f) because the one from Sun et al. [162] would not detect the small droplet.



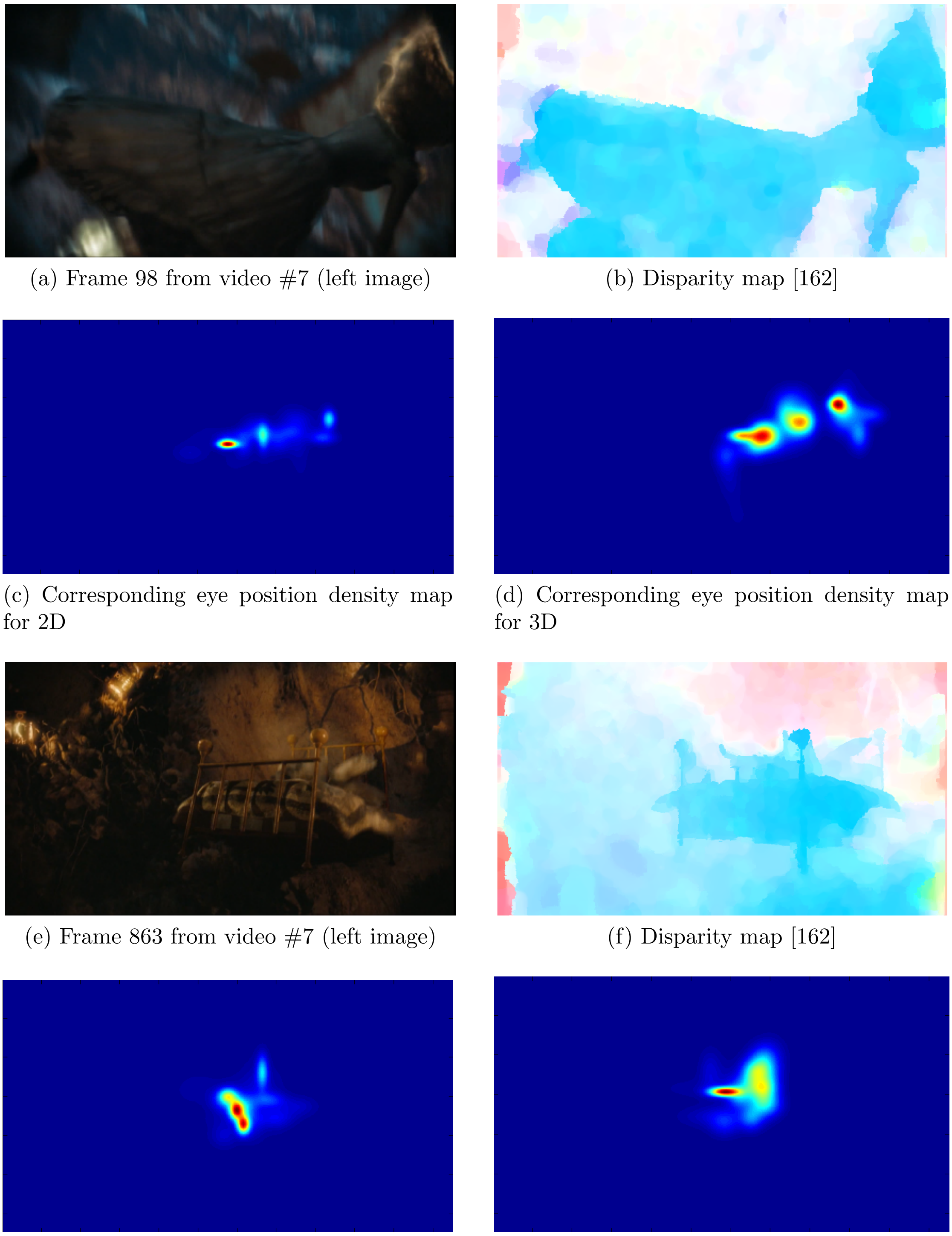
(a) Frame 635 from video #5 (left image)

(b) Disparity map [162]

(c) Corresponding eye position density map for 2D

(d) Corresponding eye position density map for 3D

Figure B.11: (a) Frame 635 from video #5; (b) disparity map for this frame with new drop highlighted; (c) & (d) eye position density maps in 2D and 3D for this frame.



(a) Frame 98 from video #7 (left image)

(b) Disparity map [162]

(c) Corresponding eye position density map for 2D

(d) Corresponding eye position density map for 3D

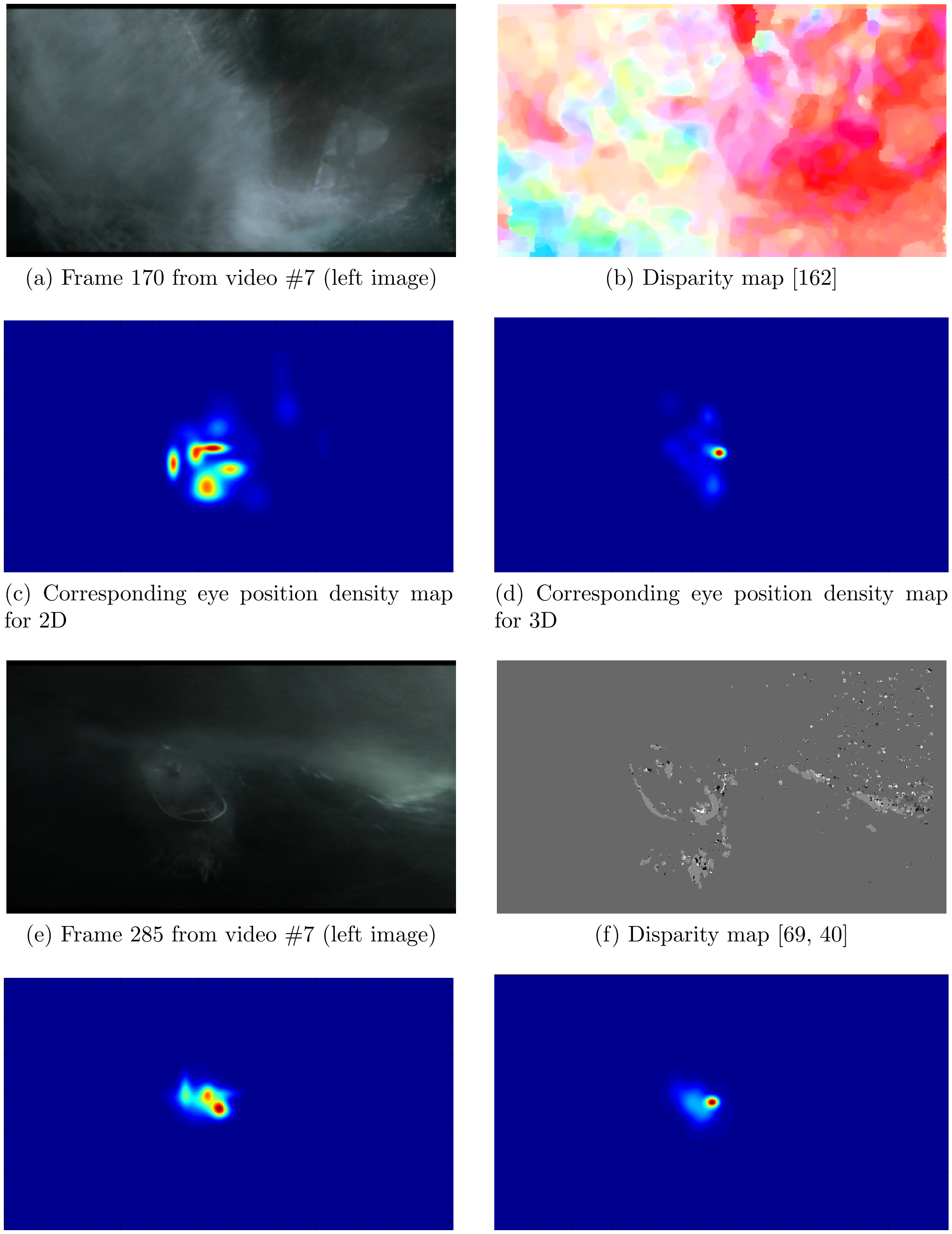
(e) Frame 863 from video #7 (left image)

(f) Disparity map [162]

(g) Corresponding eye position density map for 2D

(h) Corresponding eye position density map for 3D

Figure B.12: (a) Frame 98 from video #7; (b) disparity map; (c)&(d) eye position density maps in 2D and 3D. (e) Frame 863 from video #7; (b) disparity map; (g)&(h) eye position density maps in 2D and 3D.



(a) Frame 170 from video #7 (left image)

(b) Disparity map [162]

(c) Corresponding eye position density map for 2D

(d) Corresponding eye position density map for 3D

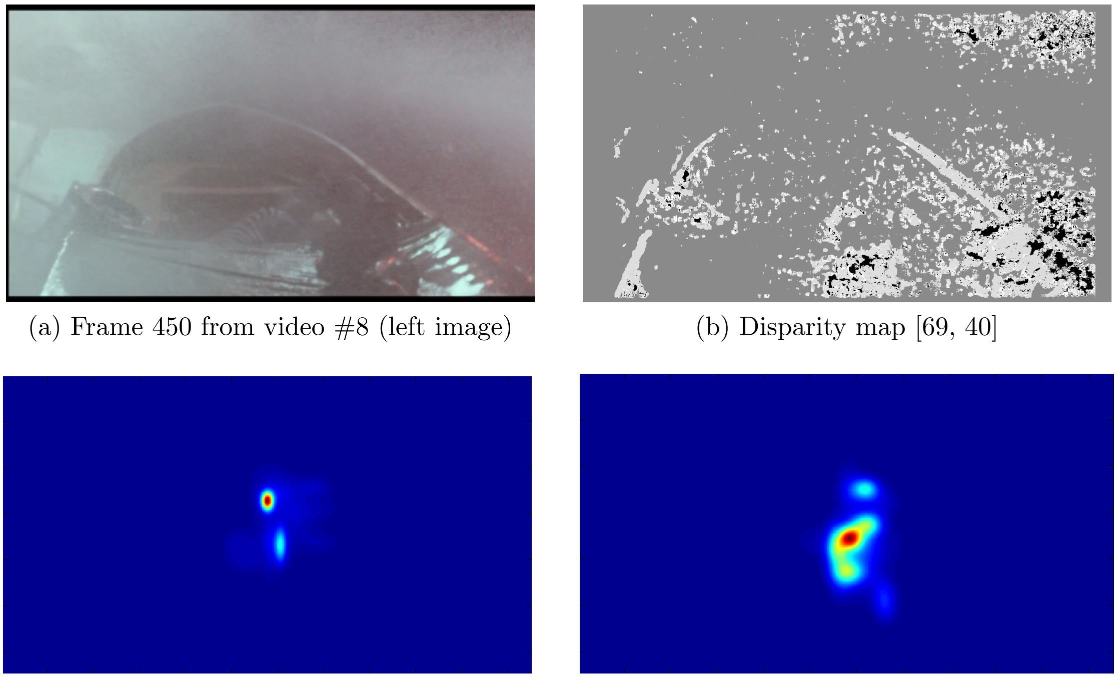
(e) Frame 285 from video #7 (left image)

(f) Disparity map [69, 40]

(g) Corresponding eye position density map for 2D

(h) Corresponding eye position density map for 3D

Figure B.13: (a) Frame 170 from video #8; (b) disparity map; (c) & (d) eye position density maps in 2D and 3D. (e) Frame 285 from video #8; (b) disparity map; (g) & (h) eye position density maps in 2D and 3D. A different disparity map extraction algorithm is used in (f) because the one from Sun et al. [162] would not detect the boat.



(a) Frame 450 from video #8 (left image)

(b) Disparity map [69, 40]

(c) Corresponding eye position density map for 2D

(d) Corresponding eye position density map for 3D

Figure B.14: (a) Frame 450 from video #8; (b) disparity map for this frame; (c) & (d) eye position density maps in 2D and 3D for this frame. A different disparity map extraction algorithm is used in (b) because the one from Sun et al. [162] would not detect the boat.

Bibliography

[1] Available at: `http://cswww.essex.ac.uk/mv/allfaces/faces94.zip`, retrieved 15/3/2012 at 10:40.

[2] A. Abbott. A survey of selective fixation control for machine vision. *IEEE Control Systems*, 12(4):25–31, 1992.

[3] Richard A. Abrams and Shawn E. Christ. Motion onset captures attention. *Psychological Science*, 14(5):427–432, 2003.

[4] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1597–1604. IEEE, 2009.

[5] A. Agarwal. *A New Approach to Spatio-Temporal Kriging and Its Applications.* PhD thesis, Ohio State Universtity, 2011.

[6] Hani Alers, Judith A Redi, and Ingrid Heynderickx. Examining the effect of task on viewing behavior in videos using saliency maps. In *IS&T/SPIE Electronic Imaging*, pages 82910X–82910X. International Society for Optics and Photonics, 2012.

[7] Robert S. Allison, Laurie M. Wilcox, and Ali Kazimi. Perceptual artefacts, suspension of disbelief and realism in stereoscopic 3d film. *Public*, 24(47):149–160, 2013-07-01T00:00:00.

[8] Y. Aloimonos, I. Weiss, and A. Bandopadhay. Active vision. *Int'l Journal of Computer Vision*, 1(4):333–356, 1988.

[9] Richard A Andersen. Neural mechanisms of visual motion perception in primates. *Neuron*, 18(6):865–872, 1997.

[10] Sami Arpa, Abdullah Bulbul, and Tolga Capin. A decision theoretic approach to motion saliency in computer animations. In JanM. Allbeck and Petros Faloutsos, editors, *Motion in Games*, volume 7060 of *Lecture Notes in Computer Science*, pages 168–179. Springer Berlin Heidelberg, 2011.

[11] M. S. Banks, J. C. A. Read, R. S. Allison, and S. J. Watt. Sterestereo and the human visual system. *SMPTE Motion Imaging Journal*, 121(4):24–43, 2012.

[12] H. Barlow. Possible principles underlying the transformation of sensory messages. In *Sensory Communication*, pages 217–234, 1961.

[13] M. Bastan, H. Cam, U. Gudukbay, and O. Ulusoy. Bilvideo-7: An mpeg-7 compatible video indexing and retrieval system. *Multimedia*, 17(3):62–73, 2007.

[14] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *European Conf. on Computer Vision*, 1:404–417, 2006.

[15] Mark W. Becker and Ian P. Rasmussen. The guidance of attention to objects and locations by long term memory of natural scenes. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 34(6):1325 – 1338, November 2008.

[16] K. Benzeroual, R.S. Allison, and L.M. Wilcox. 3d display size matters: Compensating for the perceptual effects of s3d display scaling. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 45–52, June 2012.

[17] Peng Bian and Liming Zhang. Biological plausibility of spectral domain approach for spatiotemporal visual saliency. In *Advances in Neuro-Information Processing*, pages 251–258. Springer, 2009.

[18] P. Blignaut and D. Wium. Eye-tracking data quality as affected by ethnicity and experimental design. *Behavior research methods*, 2013.

[19] G. B. Borba, H. R. Gamba, O. Marques, and L. M. Mayron. An unsupervised method for clustering images based on their salient regions of interest. *Proc. of ACM Int'l Conf. on Multimedia*, pages 145–148, 2006.

[20] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):185–207, Jan 2013.

[21] Ali Borji, Dicky N Sihite, and Laurent Itti. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *Image Processing, IEEE Transactions on*, 22(1):55–69, 2013.

[22] Stephan A Brandt and Lawrence W Stark. Spontaneous eye movements during visual imagery reflect the content of the visual scene. *Journal of cognitive neuroscience*, 9(1):27–38, 1997.

[23] N. Bruce. *Saliency, Attention and visual search: an information theoretic approach.* PhD thesis, York University, 2008.

[24] N. Bruce and J. Tsotsos. Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision*, 9(3):1–24, 2009.

[25] N.D.B. Bruce and J.K. Tsotsos. An attentional framework for stereo vision. In *Computer and Robot Vision, 2005. Proceedings. The 2nd Canadian Conference on*, pages 88–95, May 2005.

[26] N.D.B. Bruce and J.K. Tsotsos. Saliency based on information maximization. In *Advances in Neural Information Processing Systems*, 2005.

[27] Eric Bruno and Denis Pellerin. Robust motion estimation using spatial gabor-like filters. *Signal Processing*, 82(2):297–309, 2002.

[28] Abdullah Bulbul, Sami Arpa, and Tolga Capin. A clustering-based method to estimate saliency in 3d animated meshes. *Computers & Graphics*, 43(0):11 – 20, 2014.

[29] G. Buscher, E. Cutrell, and M. R. Morris. What do you see when youre surfing? using eye tracking to predict salient regions of web pages. *Proc. of the Int'l Conf. on Human Factors in Computing Systems*, pages 21–30, 2009.

[30] G. Buscher, S. Dumais, and E. Cutrell. The good, the bad, and the random: an eye-tracking study of ad quality in web search. *Proc. of Special Interest Group on Info. Retrieval (SIGIR)*, pages 42–49, 2010.

[31] Guy Thomas Buswell. *How people look at pictures.* University of Chicago Press Chicago, 1935.

[32] C. K. I. Williams C. E. Rasmussen. *Gaussian processes for machine learning.* MIT Press, 2006.

[33] Moran Cerf, Jonathan Harel, Wolfgang Einhaeuser, and Christof Koch. Predicting human gaze using low-level saliency combined with face detection. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 241–248. Curran Associates, Inc., 2008.

[34] S. Chamaret, S. Godeffroy, P. Lopez, and O. Le Meur. Adaptive 3d rendering based on region-of-interest. *SPIE*, 7524:75240V, 2010.

[35] C.-C. Chang and C.-J. Lin. Libsvm – a library for support vector machines. `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`, retrieved 23/3/2012 at 11:00.

[36] L.-Q. Chen, X. Xie, X. Fan, W.-Y. Ma, H.-J. Zhang, and H.-Q. Zhou. A visual attention model for adapting images on small displays. *ACM Multimedia Systems Joumal*, 9(4):353–364, 2003.

[37] Gregory E. Cox, George Kachergis, and Richard M. Shiffrin. Gaussian process regression for trajectory analysis. In *Cognitive Science Society*, pages 1440–1445, 2012.

[38] N. Cressie. The origins of kriging. *Mathematical Geology*, 22(3):239–252, 1990.

[39] E De Castro and C Morandi. Registration of translated and rotated images using finite fourier transforms. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (5):700–703, 1987.

[40] OpenCV dev team. Opencv 2.4.9 documentation. Technical report, `http://docs.opencv.org/`, 2014.

[41] Michael Dorr, Thomas Martinetz, Karl R Gegenfurtner, and Erhardt Barth. Variability of eye movements when viewing dynamic natural scenes. *Journal of vision*, 10(10):28, 2010.

[42] A. T. Duchowski and B. H. McCormick. Modeling visual attention for gaze-contingent video processing. *Proc. of Image and Multidimensional Signal Proc. Workshop*, pages 130–131, 1995.

[43] A. T. Duchowski and B. H. McCormick. Pre-attentive considerations for gaze-contingent image processing. *Human Vision, Visual Processing, and Digital Display*, 2411:128–139, 1995.

[44] Wolfgang Einhäuser, Merrielle Spain, and Pietro Perona. Objects predict fixations better than early saliency. *Journal of Vision*, 8(14):18, 2008.

[45] M. S. El-Nasr and S. Yan. Visual attention in 3d games. *Advances in Computer Entertainment Technology*, 22, 2006.

[46] L. Elazary and L. Itti. Interesting objects are visually salient. *Journal of Vision*, 8(3):1–15, 2008.

[47] Ulrich Engelke, H Zepernick, and Anthony Maeder. Visual attention modeling: region-of-interest versus fixation patterns. In *Picture Coding Symposium, 2009. PCS 2009*, pages 1–4. IEEE, 2009.

[48] Sonja Engmann, M Bernard, Thomas Sieren, Selim Onat, Peter König, and Wolfgang Einhäuser. Saliency on a natural scene background: Effects of color and luminance contrast add linearly. *Attention, Perception, & Psychophysics*, 71(6):1337–1352, 2009.

[49] Charles W Eriksen and James D St James. Visual attention within and around the field of focal attention: A zoom lens model. *Perception & Psychophysics*, 40(4):225–240, 1986.

[50] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *CVPR Workshop on Generative-Model Based Vision*, 12:178, 2004.

[51] X. Feng, T. Liu, D. Yang, and Y.Wang. Saliency based objective quality assessment of decoded video affected by packet losses. *Proc. of IEEE Int'l. Conf. on Image Processing (ICIP)*, pages 2560–2563, 2008.

[52] S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search.* PhD thesis, University of Bonn, 2006.

[53] S. Frintrop, E. Rome, A. Nuchter, and H. Surmann. A bimodal laser-based attention system. *Computer Vision and Image Understanding (CVIU)*, 10:124–151, 2005.

[54] K. Fukushima. Extraction of visual motion and optic flow. *Neural Networks*, 21(5):774–785, 2008.

[55] D. Gao. *A discriminant hypothesis for visual saliency: computational principles, biological plausibility and applications in computer vision.* PhD thesis, University of California, 2008.

[56] Dashan Gao and Nuno Vasconcelos. Decision-theoretic saliency: computational principles, biological plausibility, and implications for neurophysiology and psychophysics. *Neural computation*, 21(1):239–271, 2009.

[57] Zhi Gao, Loong-Fah Cheong, and Yu-Xiang Wang. Block-sparse rpca for salient motion detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(10):1975–1987, Oct 2014.

[58] Antón Garcia-Diaz, Xosé R Fdez-Vidal, Xosé M Pardo, and Raquel Dosil. Saliency from hierarchical adaptation through decorrelation and variance normalization. *Image and Vision Computing*, 30(1):51–64, 2012.

[59] Wilson S Geisler and Lawrence Cormack. Models of overt attention. *The Oxford Handbook of Eye Movements*, page 439, 2011.

[60] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1915–1926, 2012.

[61] R. B. Goldstein, R. L. Woods, and E. Peli. Where people look when watching movies: Do all viewers look at the same place? *Vision and Movement in Man and Machines*, 27(7):957–964, 2007.

[62] David Marvin Green, John A Swets, et al. *Signal detection theory and psychophysics*, volume 1. Wiley New York, 1966.

[63] Hayit Greenspan, Serge Belongie, Rodney Goodman, Pietro Perona, Subrata Rakshit, and Charles H Anderson. Overcomplete steerable pyramid filters and rotation invariance. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 222–228. IEEE, 1994.

[64] Chenlei Guo, Qi Ma, and Liming Zhang. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[65] H. Hadizadeh, M. J. Enriquez, and I. V. Bajic. Eye-tracking database for a set of standard video sequences. *Image Processing*, 21(2):898–903, 2012.

[66] J. Hakkinen, T. Kawai, J. Takatalo, R. Mitsuya, and G. Nyman. What do people look at when they watch stereoscopic movies? *SPIE Conf. Stereoscopic Displays and Applications*, 7524, 2010.

[67] D.W. Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(3):478–500, March 2010.

[68] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 19:545–552, 2007.

[69] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):328–341, Feb 2008.

[70] James E Hoffman. Visual attention and eye movements. *Attention*, 31:119–153, 1998.

[71] Andrew Hollingworth. Visual memory for natural scenes: Evidence from change detection and visual search. *Visual Cognition*, 14(4-8):781–807, 2006.

[72] K. Holmqvist, M. Nystrom, and F. Mulvey. Eye tracker data quality: What it is and how to measure it. *Symposium on Eye Tracking Research and Applications*, pages 45–52, 2012.

[73] A. J. Hornof and T. Halverson. Cleaning up systematic error in eye-tracking data by using required fixation locations. *Behavior Research Methods, Instruments & Computers*, 34(4):592–604, 2002.

[74] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[75] Q. Huynh-Thu and L. Schiatti. Examination of 3d visual attention in stereoscopic video content. *IS&T/SPIE Electronic Imaging*, 7865, 2011.

[76] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, 13(10):1304–1318, 2004.

[77] L. Itti. Models of bottom-up attention and saliency. *Neurobiology of Attention*, pages 576–582, 2005.

[78] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[79] Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. *Vision Research*, 49(10):1295 – 1306, 2009. Visual Attention: Psychophysics, electrophysiology and neuroimaging.

[80] T. E. Smith J. Dearmon. Gaussian process regression and bayesian model averaging: An alternative approach to modeling spatial phenomena. Submitted, 2014.

[81] William James. The principles of psychology, 1890.

[82] L. Jansen, S. Onat, and P. Konig. Influence of disparity on fixation and saccades in free viewing of natural scenes. *Journal of Vision*, 9(1):1–19, 2009.

[83] S. Jeong, S.-W. Ban, and M. Lee. Stereo saliency map considering affective factors and selective motion analysis in a dynamic environment. *Neural Networks*, 21:1420–1430, 2008.

[84] Xiaoyue Jiang, Andrew J. Schofield, and Jeremy L. Wyatt. Correlation-based intrinsic image extraction from a single image. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision - ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 58–71. Springer Berlin Heidelberg, 2010.

[85] S. A. Johansen and J. P. Hansen. Do we need eye trackers to tell where people look? In *Human Factors in Computing Systems*, pages 923–928, 2006.

[86] L. Juan and O. Gwun. A comparison of sift, pca-sift and surf. *Int'l Journal of Image Processing*, 3(4):143–152, 2009.

[87] Tilke Judd. *Understanding and predicting where people look in images.* PhD thesis, Massachusetts Institute of Technology, 2011.

[88] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *Computer Vision, 2009 IEEE 12th international conference on*, pages 2106–2113. IEEE, 2009.

[89] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. *Computer Vision and Pattern Recognition*, 2:90–96, 2004.

[90] Marcel A Just and Patricia A Carpenter. A theory of reading: from eye fixations to comprehension. *Psychological review*, 87(4):329, 1980.

[91] T. Kadir and M. Brady. Saliency, scale and image description. *Int'l Journal of Comp. Vision*, 45(2):83–105, 2000.

[92] Wolf Kienzle, Matthias O Franz, Bernhard Schölkopf, and Felix A Wichmann. Center-surround patterns emerge as optimal predictors for human saccade targets. *Journal of Vision*, 9(5):7, 2009.

[93] C. Koch and T. Poggio. Predicting the visual world: Silence is golden. *Nature Neuroscience*, 2(1):9–10, 1999.

[94] C. Koch and S. Ullman. Shifts in selective visual attention: Towards the underlying neural circuitry. *Human Neurology*, 4:219–227, 1985.

[95] K. Koch, J. McLean, R. Segev, M.A. Freed, M. J. Berry, V. Balasubramanian, and P. Sterling. How much the eye ttell the brain. *Current Biology*, 25(16–14):1428–34, 2006.

[96] K Koflka. Principles of gestalt psychology. *New York: Har*, 1935.

[97] A. N. Kolmogorov. Sur l'interpolation et extrapolation des suites stationnaires. *Comptes Rendus de Comptes Rendus de l'Aca de Paris*, 208:2043–2045, 1939.

[98] Gert Kootstra, Arco Nederveen, and Bart De Boer. Paying attention to symmetry. In *Proceedings of the British Machine Vision Conference (BMVC2008)*, pages 1115–1125. The British Machine Vision Association and Society for Pattern Recognition, 2008.

[99] D. G. Krige. Two-dimensional weighted moving average trend surfaces for ore-evaluation. *Journal of the South African Institute of Mining and Metallurgy*, 66:13–38, 1966.

[100] O. Le Meur, P. Le Callet, D. Barba, and D. Thoreau. A coherent computational approach to model bottom-up visual attention. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(5):802–817, May 2006.

[101] Olivier Le Meur, Patrick Le Callet, and Dominique Barba. Predicting visual fixations on video based on low-level visual features. *Vision research*, 47(19):2483–2498, 2007.

[102] V. F. Leavers. Preattentive computer vision towards a two-stage computer vision system for the extraction of qualitative descriptors and the cues for focus of attention. *Image and Vision Computing*, 12(9):583–599, 1994.

[103] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.

[104] Wei-Te Li, Haw-Shiuan Chang, Kuo-Chin Lien, Hui-Tang Chang, and Y.F. Wang. Exploring visual and motion saliency for automatic video object extraction. *Image Processing, IEEE Transactions on*, 22(7):2600–2610, July 2013.

[105] Zhicheng Li, Shiyin Qin, and Laurent Itti. Visual attention guided bit allocation in video compression. *Image and Vision Computing*, 29(1):1–14, 2011.

[106] Ke Liang, Youssef Chahir, Michèle Molina, Charles Tijus, and François Jouen. Appearance-based gaze tracking with spectral clustering and semi-supervised gaussian process regression. In *Proc. of the 2013 Conf. on Eye Tracking South Africa*, ETSA '13, pages 17–23, New York, NY, USA, 2013. ACM.

[107] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T Freeman. Sift flow: Dense correspondence across different scenes. In *Computer Vision–ECCV 2008*, pages 28–42. Springer, 2008.

[108] H. Liu, X. Xie, W. Ma, and H. Zhang. Automatic browsing of large pictures on mobile devices. *Proc. of the ACM Int'l Conf. on Multimedia*, pages 148–155, 2003.

[109] Huiying Liu, Shuqiang Jiang, Qingming Huang, and Changsheng Xu. A generic virtual content insertion system based on visual attention analysis. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 379–388. ACM, 2008.

[110] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(2):353–367, 2011.

[111] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l Journal of Computer Vision*, 60(2):91–110, 2004.

[112] Q. Ma and L. Zhang. Image quality assessment with visual saliency. *Proc. of the Int'l Conf. on Pattern Recognition*, pages 1–4, 2008.

[113] Z. Ma, L. Qing, J. Miao, and X. Chen. Advertisement evaluation using visual saliency based on foveated image. *Int'l Conf. on Multimedia and Expo (ICME)*, pages 914–917, 2009.

[114] A. Maeder and H. Zapernick. Analysing inter-observer saliency variations in task-free viewing of natural images. *Image Processing (ICIP)*, pages 1085–1088, 2010.

[115] Vijay Mahadevan and Nuno Vasconcelos. Saliency-based discriminant tracking. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1007–1013. IEEE, 2009.

[116] Vijay Mahadevan and Nuno Vasconcelos. Spatiotemporal saliency in dynamic scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):171–177, 2010.

[117] A. Maki, P. Nordlund, and J.-O. Eklundh. A computational model of depth-based attention. *Int'l Conf. on Pattern Recognition*, 4:734–738, 1996.

[118] M. Mancas. *Computational attention: Modelisation and application to audio and image processing*. PhD thesis, Faculte Polytechnique de Mons, 2007.

[119] Sophie Marat, Mickaël Guironnet, Denis Pellerin, et al. Video summarization using a visual attention model. In *Proceedings of the 15th European Signal Processing Conference, EUSIPCO-2007*, 2007.

[120] Sophie Marat, Tien Ho Phuoc, Lionel Granjon, Nathalie Guyader, Denis Pellerin, and Anne Guérin-Dugué. Modelling spatio-temporal saliency to predict gaze direction for short videos. *International Journal of Computer Vision*, 82(3):231–243, 2009.

[121] L. Marchesotti, C. Cifarelli, and G. Csurka. A framework for visual saliency detection with applications to image thumbnailing. *Proc. of the Int'l Conference on Computer Vision (ICCV)*, pages 2232–2239, 2009.

[122] Stefan Mathe and Cristian Sminchisescu. Dynamic eye movement datasets and learnt saliency models for visual action recognition. In *Computer Vision–ECCV 2012*, pages 842–856. Springer, 2012.

[123] M. Matheron. Principles of geostatistics. *Economic Geology*, 58:1246–1266, 1963.

[124] L. M. Mayron. *Image retrieval using visual attention*. PhD thesis, Florida Atlantic University, USA, 2008.

[125] R. McDonnell, M. Larkin, B. Hernndez, I. Rudomin, and C. O'Sullivan. Eye-catching crowds: saliency based selective variation. *ACM Transactions on Graphics*, 28(3), 2009.

[126] O. Le Meur and T. Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods*, 45(1):251–266, 2013.

[127] R. Milanese, H. Wechsler, S. Gil, J. Bost, and T. Pun. Integration of bottom-up and top-down cues for visual attention using non-linear relaxation. *Computer Vision and Pattern Recognition*, pages 781–785, 1994.

[128] Parag K Mital, Tim J Smith, Robin L Hill, and John M Henderson. Clustering of gaze during dynamic scene viewing is predicted by motion. *Cognitive Computation*, 3(1):5–24, 2011.

[129] K. Nakayama and G. H. Silverman. Serial and parallel processing of visual feature conjunctions. *Nature*, 320:264–265, 1986.

[130] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. *Int'l Conf. on Comp. Vision*, 2:1447–1454, 2006.

[131] T. Noesselt, D. Bergmann, M. Hake, H.-J. Heinze, and R. Fendrich. Sound increases the saliency of visual events. *Brain research*, 1220:157–163, 2008.

[132] Billard A. Noris B., Benmachiche K. Calibration-free eye gaze direction detection with gaussian processes. In *International Conference on Computer Vision Theory and Application*, pages 611–616, 2008.

[133] M. Nystrom, R. Andersson, K. Holmqvist, and J. van der Weijer. The influence of calibration method and eye physiology on eyetracking data quality. *Behavior Research Methods*, 45:272–288, 2013.

[134] M. A. Oliver and R. Webster. Kriging: a method of interpolation for geographical information systems. *International journal of geographical information systems*, 4(3):313–332, 1990.

[135] N. Ouerhani and H. Hugli. Computing visual attention from scene depth. *Proc. of the Int'l Conf. on Pattern Recognition (ICPR)*, 1:375–378, 2000.

[136] D. Pang, A. Kimura, T. Takeuchi, J. Yamato, and K. Kashino. A stochastic model of selective visual attention with a dynamic bayesian network. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 1073–1076, June 2008.

[137] Derrick Parkhurst, Klinton Law, and Ernst Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision Research*, 42(1):107 – 123, 2002.

[138] Derrick J Parkhurst and Ernst Niebur. Scene content selected by active vision. *Spatial vision*, 16(2):125–154, 2003.

[139] E. Pebesma. spacetime: Spatio-temporal data in r. *Journal of Statistical Software*, 51(7), November 2012.

[140] L. Pessoa and S. Exel. Attentional strategies for object recognition. *Proc. of the Int'l Work-Conference on Artificial Neural Networks (IWANN)*, 1606:850–859, 1999.

[141] Robert J. Peters, Asha Iyer, Laurent Itti, and Christof Koch. Components of bottom-up gaze allocation in natural images. *Vision Research*, 45(18):2397 – 2416, 2005.

[142] Matthew S Peterson, Arthur F Kramer, and David E Irwin. Covert shifts of attention precede involuntary eye movements. *Perception & psychophysics*, 66(3):398–405, 2004.

[143] Michael I Posner, Charles R Snyder, and Brian J Davidson. Attention and the detection of signals. *Journal of experimental psychology: General*, 109(2):160, 1980.

[144] E. Potapova, M. Zillich, and M. Vincze. Learning what matters: combining probabilistic models of 2d and 3d saliency cues. *Proc. of Computer Vision Systems (ICVS)*, pages 132–142, 2011.

[145] Claudio M. Privitera and Lawrence W Stark. Algorithms for defining visual regions-of-interest: Comparison with eye fixations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(9):970–982, 2000.

[146] Umesh Rajashekar, Ian Van Der Linde, Alan C Bovik, and Lawrence K Cormack. Gaffe: A gaze-attentive fixation finding engine. *Image Processing, IEEE Transactions on*, 17(4):564–573, 2008.

[147] C. Ramasamy, D. H. House, A. T. Duchowski, and B. Daugherty. Using eye tracking to analyze stereoscopic filmmaking. *SIGGRAPH*, Article No. 28 (Poster), 2009.

[148] N. Rasmussen, D. Thornton, and B. Morse. Enhancement of unusual color in aerial video sequences for assisting wilderness search and rescue. *Proc. of the IEEE Int'l Conf. on Image Processing (ICIP)*, pages 1356–1359, 2008.

[149] D. Reece and S. Shafer. Control of perceptual attention in robot driving. *Artificial Intelligence*, 78:397–430, 1995.

[150] SR Research. *Eyelink II Technical Specifications.* http://www.sr-research.com/pdf/elII_table.pdf.

[151] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *Computer Vision*, 77(1–3):157–173, May 2008.

[152] U. Rutishauser, D. Walther, C. Koch, and P. Perona. Is bottom-up attention useful for object recognition? *Proc. of Computer Vision and Pattern Recognition*, 2:37–44, 2004.

[153] S. K. Schnipke and M. W. Todd. Trials and tribulations of using an eye-tracking system. In *Human Factors in Computing Systems*, pages 273–274, 2000.

[154] P. Sharma. *Towards three-dimensional visual saliency.* PhD thesis, Norwegian University of Science and Technology, 2014.

[155] C. Siagian and L. Itti. Biologically-inspired robotics vision monte-carlo localization in the outdoor environment. *Proc. of the Int'l Conf. on Intelligent Robots and Systems (IROS)*, pages 1723–1730, 2007.

[156] T. E. Slowe and I. Marsic. Saliency-based visual representation for compression. *IEEE Int'l Conference on Image Processing (ICIP)*, pages 554–557, 1997.

[157] B. Stankiewicz, N. Anderson, and R. Moore. Using performance efficiency for testing and optimization of visual attention models. *Proc. of SPIE*, 7867:78670Y, 2011.

[158] Lawrence W Stark and Yun S Choi. Experimental metaphysics: the scanpath as an epistemological mechanism. *Advances in Psychology*, 116:3–69, 1996.

[159] Katrin Suder and Florentin Wörgötter. The control of low-level information flow in the visual system. *Reviews in the Neurosciences*, 11(2-3):127–146, 2000.

[160] Y. Sugano, Y. Matsushita, and Y. Sato. Calibration-free gaze sensing using saliency maps. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2667–2674, June 2010.

[161] B. Suh, H. Ling, B. B. Bederson, and D. W. Jacobs. Automatic thumbnail cropping and its effectiveness. *Proc. of the ACM symposium on User Interface Software and Technology*, pages 95–104, 2003.

[162] Deqing Sun, S. Roth, and M.J. Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439, June 2010.

[163] Kar-Han Tan, D. Kriegman, and N. Ahuja. Appearance-based eye gaze estimation. In *Applications of Computer Vision, 2002. (WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 191–195, 2002.

[164] Benjamin W Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision*, 7(14):4, 2007.

[165] A. Treisman and G. Gelade. A feature integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[166] John K Tsotsos, Scan M Culhane, Winky Yan Kei Wai, Yuzhong Lai, Neal Davis, and Fernando Nuflo. Modeling visual attention via selective tuning. *Artificial intelligence*, 78(1):507–545, 1995.

[167] Muhammad Umer, Lars Kulik, and Egemen Tanin. Kriging for localized spatial interpolation in sensor networks. In Bertram Ludscher and Nikos Mamoulis, editors, *Scientific and Statistical Database Management*, volume 5069 of *Lecture Notes in Computer Science*, pages 525–532. Springer Berlin Heidelberg, 2008.

[168] Muhammad Umer, Lars Kulik, and Egemen Tanin. Spatial interpolation in wireless sensor networks: localized algorithms for variogram modeling and kriging. *GeoInformatica*, 14(1):101–134, 2010.

[169] D. Walther. *Interactions of visual attention and object recognition: computational modeling, algorithms, and psychophysics.* PhD thesis, California Institute of Technology, Pasadena, CA, February 2006.

[170] D. Walther. Modeling attention to salient proto-objects. *Neural Networks*, 19(9):1395–1407, 2006.

[171] Dirk Walther, Laurent Itti, Maximilian Riesenhuber, Tomaso Poggio, and Christof Koch. Attentional selection for object recognition a gentle way. In HeinrichH. Blthoff, Christian Wallraven, Seong-Whan Lee, and TomasoA. Poggio, editors, *Biologically Motivated Computer Vision*, volume 2525 of *Lecture Notes in Computer Science*, pages 472–479. Springer Berlin Heidelberg, 2002.

[172] J. Wang, P. Le Callet, S. Tourancheau, V. Ricordel, and M. P. Da Silva. Study of depth bias of observers in free viewing of still stereoscopic synthetic stimuli. *Journal of Eye Movement Research*, 5(5):1–11, 2012.

[173] J. Wang, M. P. Da Silva, P. Le Callet, and V. Ricordel. A computational model of stereoscopic 3d visual saliency. *IEEE Trans. on Image Processing*, 22(6):2151–2161, 2013.

[174] Wei Wang, Cheng Chen, Yizhou Wang, Tingting Jiang, Fang Fang, and Yuan Yao. Simulating human saccadic scanpaths on natural images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 441–448. IEEE, 2011.

[175] Z. Wang, L. Lu, and A. C. Bovik. Foveation scalable video coding with automatic fixation selection. *IEEE Trans. on Image Processing*, 12:243–254, 2003.

[176] R. Webster and M. Oliver. *Geostatistics for Environmental Scientists.* John Wiley & Sons, Ltd, 2001.

[177] B. Weyrauch, J. Huang, B. Heisele, and V. Blanz. Component-based face recognition with 3d morphable models. *Proc. of Conf. on Computer Vision and Pattern Recognition Workshop*, 5(5):85, 2004.

[178] N. Wiener. *Extrapolation, interpolation, and smoothing of stationary time series.* MIT Press, Cambridge, MA, 1949.

[179] O. Williams, A Blake, and R. Cipolla. Sparse and semi-supervised visual mapping with the s3gp. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 230–237, June 2006.

[180] T. J. Williams and B. A. Draper. An evaluation of motion in artificial selective attention. *Computer Vision and Pattern Recognition Workshop (CVPRW)*, 3:85–93, 2005.

[181] Herman Wold. *A study in the analysis of stationary time series.* PhD thesis, , Stockholm College, 1938.

[182] J. M. Wolfe and T. S. Horowitz. What attributes guide the deployment of visual attention and how do they do it. *Nature Reviews Neuroscience*, 5:495–501, 2004.

[183] Geoffrey F. Woodman and Marvin M. Chun. The role of working memory and long-term memory in visual search. *Visual Cognition*, 14(4-8):808–830, 2006.

[184] Linfeng Xu, Liaoyuan Zeng, Huiping Duan, and Nii Longdon Sowah. Saliency detection in complex scenes. *EURASIP Journal on Image and Video Processing*, 2014(1):1–13, 2014.

[185] J. You, J. Korhonen, and A. Perkis. Attention modeling for video quality assessment: Balancing global quality and local quality. *Proc. of IEEE Int'l Conf. on Multimedia and Expo*, pages 914–919, 2010.

[186] H. Zabrodsky and S. Peleg. Attentive transmission. *Journal of Visual Comm. and Image Representation*, 1(2):189–198, 1990.

[187] Z. Zdziarski, C. Bourges, J. Mitchell, P. Houdyer, D. Johnson, and R. Dahyot. On summarising the 'here and now' of social videos for smart mobile browsing. In *Int'l Workshop on Computational Intelligence for Multimedia Understanding (submitted)*, 2014.

[188] Z. Zdziarski and R. Dahyot. Extension of gbvs to 3d media. In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pages 2296–2300, April 2014.

[189] Zbigniew Zdziarski, Joe Mitchell, Pierre Houdyer, Dave Johnson, Cyril Bourges, and Rozenn Dahyot. An architecture for social media summarisation. In *Irish Machine Vision and Image Processing Conference, Derry-Londonderry, Northern Ireland*, pages 27–29, 2014.

[190] Lingyun Zhang, Matthew H Tong, Tim K Marks, Honghao Shan, and Garrison W Cottrell. Sun: A bayesian framework for saliency using natural statistics. *Journal of vision*, 8(7):32, 2008.

[191] Qi Zhao and Christof Koch. Learning a saliency map using fixated locations in natural scenes. *Journal of vision*, 11(3):9, 2011.