UNIVERSITY OF CALIFORNIA, SAN DIEGO

Applications of visual saliency to video processing

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Signal and Image Processing)

by

Natan Haim Jacobson

Committee in charge:

Professor Truong Q. Nguyen, Chair
Professor Pamela Cosman
Professor Yoav Freund
Professor William Hodgkiss
Professor Nuno Vasconcelos

2011

The dissertation of Natan Haim Jacobson is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____
Chair

University of California, San Diego

2011

DEDICATION

In memory of Simone Ellen Jacobson, 1946-1998

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Thanks to the support of my advisor, family, friends, colleagues and co-authors.

Chapter 2, in part, is a reprint of the material as it appears in *IEEE Transactions on Image Processing*. Co-authors include: Yen-Lin Lee, Vijay Mahadevan, Prof. Nuno Vasconcelos and Prof. Truong Nguyen. The dissertation author was the primary investigator and author of this material.

Chapter 3, in part, has been accepted for publication in *IEEE Transactions on Image Processing*. Co-authors include: Prof. Truong Nguyen. The dissertation author was the primary investigator and author of this material.

Chapter 4, in part, is currently being prepared for submission to *Journal of Vision*. Co-authors include: Elizabeth Schotter, Yang Liu, Prof. Alan Bovik, Prof. Keith Rayner and Prof. Truong Nguyen. The dissertation author was the primary investigator and author of this material.

Chapter 5, in part, has been accepted for publication in *IEEE Transactions on Image Processing*. Co-authors include: Prof. Yoav Fruend and Prof. Truong Nguyen. The dissertation author was the primary investigator and author of this material.

VITA

2006	B. S. E. in Electrical Engineering *cum laude*, Arizona State University,
2008	M. S. in Electrical Engineering (Signal and Image Processing), University of California, San Diego
2011	Ph. D. in Electrical Engineering (Signal and Image Processing), University of California, San Diego

PUBLICATIONS

N. Jacobson, Y.-L. Lee, V. Mahadevan, N. Vasconcelos, T. Q. Nguyen, "A Novel Approach to FRUC using Discriminant Saliency and Frame Segmentation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2924-2934, November 2010

N. Jacobson, Y. Freund, T. Q. Nguyen, "An Online Learning Approach to Occlusion Detection," accepted to *IEEE Trans. Image Process.*, July 2011

N. Jacobson, T. Q. Nguyen, "Scale-Aware Saliency for Application to Frame Rate Up-Conversion," accepted to *IEEE Trans. Image Process.*, August 2011

N.H. Jacobson, E.R. Schotter, Y. Liu, A.C. Bovik, K. Rayner, T.Q. Nguyen, "Effect of Stereoscopic Depth on Saliency: Evidence from Eye Movements," in preparation for *J. Vis.*, August 2011

N. Jacobson, T.Q. Nguyen, F. Crosby, "Curvature Scale Space Application to Distorted Object Recognition and Classification," *2007 Asilomar Conf. Sig., Syst., Comp.*, pp. 2110-2114, 4-7 Nov. 2007

N. Jacobson, Y.-L. Lee, V. Mahadevan, N. Vasconcelos, T.Q. Nguyen, "Motion Vector Refinement for FRUC using Saliency and Segmentation," *2010 IEEE Int. Conf. Multimedia and Expo*, pp. 778-783, 19-23 Jul. 2010

N. Jacobson, Y. Freund, T.Q. Nguyen, "Occlusion Boundary Detection Using an Online Learning Framework," *2011 IEEE Int. Conf. Accoustics, Speech, Sig. Process.*, pp. 913-916, 22-27 May 2011

N. Jacobson, T.Q. Nguyen, "Video Processing with Scale-Aware Saliency: Applciation to Frame Rate Up-Conversion," *2011 IEEE Int. Conf. Accoustics, Speech, Sig. Process.*, pp. 1313-1316, 22-27 May 2011

ABSTRACT OF THE DISSERTATION

Applications of visual saliency to video processing

by

Natan Haim Jacobson

Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

University of California, San Diego, 2011

Professor Truong Q. Nguyen, Chair

Our understanding of the human visual system has advanced significantly over the past quarter-century. With the availability of modern computers and development of sophisticated algorithms, it is now possible to efficiently predict human attention patterns for images and video. A saliency map can easily be generated, which provides a measure of how *important* each portion of a scene is, with respect to the human visual system. A region with a high saliency value is more likely to be fixated upon by a human than a region with a low saliency value. In this work, we explore the application of saliency to video processing. In our first project, saliency is applied to Frame Rate Up-Conversion. By enforcing motion vector refinement only for salient regions, we reduce process-

ing time while maintaining a high level of visual quality for the up-converted video sequence. In our second project, we propose a new method for saliency detection which considers object scale using a scale-space model. Excellent results are demonstrated, including improved performance of our saliency-based Frame Rate Up-Conversion algorithm. Finally, an experiment is conducted on the salient power of the stereoscopic depth feature using two different datasets. While local contrasts in luminance, color, orientation and motion are known to be highly salient, less is understood about local contrasts in depth. Using a mirror stereoscope for 3D display to subjects and an eye-tracking system, we measure human fixations for 2D (no depth) and 3D scenes. We determine that contrast in stereoscopic depth repels human fixations for natural scenes, while attracting it for synthetic scenes. This conflict may arise from different stages of human attention (bottom-up vs. top-down), activated by the different scene content in the two datasets.

Chapter 1

Introduction

"The journey is part of the experience - an expression of the seriousness of one's intent. One doesn't take the A train to Mecca." - Anthony Bourdain

The research literature on saliency detection is vast and has been motivated by a wide range of research interests. The common goal of all saliency detection algorithms is to classify image content using a measure of *importance*. In general, this *importance* is measured with respect to the Human Visual System (HVS). The resultant saliency map is an invaluable tool for image/video processing based on the psychophysical characteristics of human vision. This tool immediately lends itself to such video processing applications as: enhancement, compression and detection. For video enhancement, techniques may be applied selectively to salient regions, thereby reducing the total computational expense of the operation. Alternately, saliency-based compression is applied with the goal of allocating bits based on their perceptual weighting, minimizing the perceived distortion. Finally, saliency may be applied to automatic object detection. Without any prior information, a detector may locate the most salient object in a scene for further investigation.

In this work, we focus on the enhancement method of Frame Rate Up-Conversion (FRUC), which increases the temporal sampling rate of a sequence. Used commonly for mobile video as well as home theater applications, this enhancement technique increases the smoothness of a video sequence while coun-

teracting motion blur caused by LCD panels. By including the saliency information in our process, we are able to improve the visual quality of up-converted content while minimizing the computational complexity. Our proposed FRUC architecture is analyzed both using previous saliency methods, as well as a proposed scale-aware method.

In addition to the FRUC application, we investigate saliency with respect to the stereoscopic depth feature. Here, stereoscopic depth refers to the depth sensation humans experience as a result of stereo vision. While saliency is well understood with respect to features such as luminance, color, orientation and motion; the connection between depth and saliency is unknown. Previous research is conflicting on whether the depth feature attracts or repels human attention. We extend the previous research by conducting a large eye-tracking experiment with 20 subjects and monocular/stereo data from two different datasets. Our results are mixed, confirming the previous results for one dataset, while obtaining conflicting results from the other. As it will be shown, this conflict may arise from differences in scene content between the two datasets.

Finally, we take a small detour and investigate occlusion for video. Occlusion is the phenomenon which occurs whenever one object in a video covers or uncovers other objects due to its proximity to the camera. This causes some difficulty with video processing algorithms, as content is introduced or removed from the scene, breaking the assumptions on which many algorithms are based. In this work, we use an online-learning method to detect occlusion boundaries in a video sequence. Our approach does not require any training data and demonstrates promising results.

This dissertation is organized as follows. In chapter 2, we present the application of saliency to Motion-Compensated Frame Interpolation (MCFI). This method is validated using both objective and subjective measures. In chapter 3, the scale-aware saliency method is introduced. Next, in chapter 4, We discuss the effect of stereoscopic depth on saliency. This is accomplished through a set of experiments using a mirror stereoscope and eye-tracker. A

brief departure occurs in chapter 5, where we discuss an online-learning based approach to occlusion boundary detection. Finally, we conclude in chapter 6.

Chapter 2

Saliency for Motion-Compensated Frame Interpolation

"The sky calls to us. If we do not destroy ourselves, we will one day venture to the stars." - Carl Sagan

Motion-compensated frame interpolation (MCFI) is a technique used extensively for increasing the temporal frequency of a video sequence. In order to obtain a high quality interpolation, the motion field between frames must be well-estimated. However, many current techniques for determining the motion are prone to errors in occlusion regions, as well as regions with repetitive structure. We propose an algorithm for improving both the objective and subjective quality of MCFI by refining the motion vector field. We first utilize a discriminant saliency classifier to determine which regions of the motion field are most important to a human observer. These regions are refined using a multistage motion vector refinement (MVR), which promotes motion vector candidates based upon their likelihood given a local neighborhood. For regions which fall below the saliency-threshold, a frame segmentation is used to locate regions of homogeneous color and texture via normalized cuts. Motion vectors are promoted such that each homogeneous region has a consistent motion. Experimental results

demonstrate an improvement over previous frame rate up-conversion (FRUC) methods in both objective and subjective picture quality.

2.1 Introduction

FRUC is an area of significant research with many important applications. In mobile video, bandwidth restrictions make it infeasible to transmit at high frame rates. Instead the focus is on increasing spatial video quality while reducing the number of frames transmitted. FRUC is then used on the receiver end to recreate a smooth video. A typical example would be transmission at 15Hz with the FRUC engine performing up-conversion by a factor f two to 30Hz. Another important application is motion blur reduction for Liquid Crystal Display (LCD) televisions. This is necessary because of the sample-and-hold nature of LCD displays, which causes noticeable motion blur at low frame rates. Newer LCD displays on the market are capable of displaying at 120 to 240Hz thus significantly reducing the noticeable effect of motion blur. In order to take advantage of these high frame rates, FRUC is required to up-convert source material to the required rate.

FRUC is composed of two portions: Motion Estimation (ME) and Motion Compensated Frame Interpolation (MCFI). A block-based ME algorithm operates by partitioning each frame into uniform blocks (generally 8x8 pixels) and determining the relative translation between each block in successive video frames. The result of the ME step is a motion field for the entire frame. Next, an intermediate frame is generated by the algorithm by interpolating along the motion field direction. Interpolation is performed bi-directionally to avoid any holes in the resultant frame. Given a motion vector (v_x, v_y) from the motion estimator, a block in the interpolated frame f_t is calculated as follows from the current frame f_{t+1} and reference frame f_{t-1}:

$$f_t(x,y) = 0.5 f_{t-1}\left(x + \frac{v_x}{2}, y + \frac{v_y}{2}\right) + 0.5 f_{t+1}\left(x - \frac{v_x}{2}, y - \frac{v_y}{2}\right) \tag{2.1}$$

Because FRUC is performed on a block basis, there are several issues which we aim to resolve. One limitation of a block-based method is that objects in the scene generally do not conform to block boundaries. Therefore,

a single block may contain multiple objects with conflicting motion. Another limitation is that the motion vector which minimizes predicted block error may in fact not be the best choice. This can occur because of changes in luminance between frames or due to repetitive structures. Finally, FRUC can suffer from a ghosting artifact which is caused by large motions being assigned outside of object boundaries. These shortcomings are addressed in this work.

There are also difficulties experienced during the MCFI stage. One primary concern is motion aliasing due to low temporal sampling rates [1]. This occurs when the video frame rate falls below the Nyquist rate describing an object's trajectory. Because the trajectory is incorrect, temporal upconversion is incapable of recovering the true motion. However, these issues are negligible in MCFI for most video processing applications. Most FRUC methods assume that trackable objects in a scene have momentum over small time scales. As a result, object motions are close to linear between frames, and are safely interpolated as such.

We propose a novel method for FRUC aimed at improving both objective and subjective quality compared with previous methods. Saliency detection is employed in order to determine which regions of the scene are visually important to a human observer, thereby requiring very accurate motion vectors. Conversely, motion-vector smoothness and consistency are enforced for non-salient regions using a fast frame segmentation.

The section is organized as follows. In Section 5.2, we present a review of previous research in FRUC and Motion Compensated Frame Interpolation (MCFI). A detailed overview of discriminant saliency is introduced in Section 2.3 and of frame segmentation in Section 2.4. The proposed algorithm is detailed in Section 5.4 along with a description of all parameters used. Objective and subjective experimental results for the proposed method are presented in Section 5.5. Finally, we conclude in Section 5.6.

2.2 Previous Work

Improvement of FRUC techniques has been the scope of many research projects over the past few decades. A very early contribution is the idea of 3D Recursive Search (3DRS) for ME [2], in which motion vectors are estimated based on spatial and temporal candidates from the same neighborhood. By considering candidates which have already been encountered, 3DRS is an efficient method which promotes a smooth motion field. Temporal Compensated ME with Simple Block-based Prediction (TC-SBP) is another method similar to 3DRS in that it exploits spatial and temporal candidates [3]. In this method, only three candidates are necessary for block prediction, while temporal update candidates aid in convergence of the global motion field. While these methods exploit spatial information for ME, frequency information is exploited by Phase Plane Correlation [4]. This method has the benefit of arbitrarily accurate motion vectors as well as good estimation of local motion. However, the assumptions behind PPC are only valid for blocks undergoing pure translation. More recent approaches such as [5] focus on improving FRUC for regions with high block error by merging neighboring regions with large error. Another approach is to iteratively refine the motion vector field while propagating motion vector candidates [6]. Gao et al. proposed the idea of Adaptive FRUC based on Motion Classification [7]. In this work, the scene is classified into global and local motion regions, and bidirectional or unidirectional ME is used based on the classification result.

We consider recent research into the field of saliency for determination of visually important regions. Saliency has previously been used for the task of video compression by Itti [8]. In this work, salient locations are determined for each frame of a video sequence and the frame is then blurred for regions sufficiently distant from the determined salient locations. This allows for the blurred regions to be compressed using fewer bits while the salient locations remain untouched. Research by Walther and Koch [9] models bottom-up saliency as a biologically plausible model of contrast between Gabor-filter orientation

and color oppency. Gabor filtering is consistent with spatial filtering in the primary visual cortex, while color oppency is consistent with processing by retinal ganglion cells. In [10], the link between human recognition and bottom-up saliency is explored. It is determined that Human observers require very little spatial attention in order to recognize animals in images of natural scenes, supporting bottom-up saliency for detection of important regions in a scene. A bottom-up discriminant saliency detector is proposed in [11], built upon a center-surround framework. This detector performs well for predicting human-eye fixation, and is also in agreement with related literature in psychophysics of the Human Visual System (HVS).

Segmentation has been used extensively for previous video processing applications. Among these, it is typical to use optical flow [12, 13] for motion estimation, rather than block-based methods. An early contribution to the field of motion segmentation is due to Thompson [14]. A combination of motion information and contrast is used in order to segment a scene into regions of consistent motion. This work makes use of a non-matching technique for motion estimation based on time-varying changes in intensity and image gradient. A contribution by Tian and Shah handles the problems of motion estimation and segmentation concurrently [15]. In this work, Markov Random Field (MRF) techniques are exploited in order to simultaneously compute the motion between adjacent frames and segment the scene into a collection of objects. Similar research is conducted in [16], where a Bayesian framework is introduced to model the motion field as the sum of a parametric motion model (based on the scene segmentation) and a residual motion vector. Khan and Shah propose a method for video segmentation using spatial location, color and motion for segmentation [17]. They show that the fusion of these features performs better than any single feature used independently. A separate approach is explored by Cremers and Soatto in [18]. Rather than segmenting each frame of a sequence into regions, this work aims to segment an entire sequence into disjoint phases of homogeneous motion, however demonstrated results depend heavily on the initialization of the segmentation parameters. Finally, a patent by Eastman

Kodak [19] introduces a method for FRUC based on segmentation of the image into foreground and background regions. Separate motion data is used for each region and the occlusion regions are handled gracefully. However, this method assumes that the occlusions are manually marked by a Human technician rather than determined algorithmically.

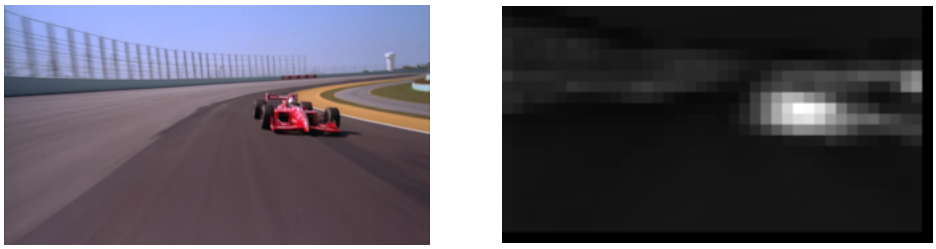
2.3 Discriminant Saliency

Human observers typically focus their visual attention on small regions of the video frame that appear interesting. By subjecting only these attended regions to post-processing such as motion vector refinement, the quality of FRUC can be improved while keeping computational complexity manageable. The automatic selection of the regions of interest as perceived by the human visual system (HVS) has been well studied in the context of bottom-up saliency, and has been applied to improve video compression [8]. However, these techniques have been developed for static images and are not suitable for motion based region of interest identification. Therefore, in this work, we use the recently proposed discriminant center-surround model for motion saliency [20] to automatically identify salient moving objects.

2.3.1 Discriminant Center-Surround Motion Saliency

Discriminant center-surround saliency is a biologically plausible algorithm that has been shown to replicate the psychophysics of saliency mechanisms in the HVS. It can directly be applied to motion saliency simply by using appropriate motion models such as optical flow or dynamic textures [21]. In this work, a dynamic texture model is used to determine the motion-based feature response.

Dynamic texture data is obtained by determining an Autoregressive Moving Average (ARMA) model for a small piece of spatiotemporal data. This data is a three-dimensional volume with two spatial dimensions and one time



(a) (b)

Figure 2.1: Discriminant Saliency map using dynamic texture model, (a) input frame from "Speedway" sequence, (b) saliency map. Larger pixel intensity (closer to white) represents higher saliency value.

dimension. The volume of data represents an observed sequence $\{y(t)\}$ seen as the output of a dynamic texture $\{I(t)\}$ with added noise $n(t)$. Using this notation, the dynamic texture coefficients can be determined using the following process:

$$\begin{cases} x(t) &= \sum_{i=1}^{k} A_i x(t-i) + Bv(t) \\ y(t) &= \phi(x(t)) + n(t) \end{cases} \tag{2.2}$$

where ϕ is a spatial filter, $I(t) = \phi(x(t))$, $v(t)$ is selected IID from an unknown distribution, and $n(t)$ is selected IID from a given distribution $p_n(\cdot)$.

Discriminant saliency is defined with respect to two classes of stimuli and a feature $\mathbf{Y}$: the class of visual stimuli in the center (with label $C = 1$), and class of visual stimuli in the *background* or surround (with label $C = 0$). The saliency of location l of the video, denoted $S(l)$, is the extent to which the feature $\mathbf{Y}$ can discriminate between *center* and *surround* at l. This is quantified by the mutual information between features, $\mathbf{Y}$, and class label, C,

$$S(l) = I_l(\mathbf{Y}; C) = \sum_{c=0}^{1} \int p_{Y,C(l)}(\mathbf{y}, c) \log \frac{p_{Y,C(l)}(\mathbf{y}, c)}{p_Y(\mathbf{y}) p_{C(l)}(c)} d\mathbf{y}. \tag{2.3}$$

A large $S(l)$ implies that center and surround have a large disparity of feature responses, i.e. large *local feature contrast* indicating that the location is salient. By selecting an appropriate feature $\mathbf{Y}$ that encodes both spatial

and temporal characteristics of the video (e.g. dynamic textures, optical flow) we can obtain regions that are spatiotemporally salient. Figure 2.1 shows the saliency map for the "Speedway" sequence obtained by using dynamic textures. The map shows that the regions predicted to have high saliency (e.g the car) are indeed the regions that appear visually salient to a human observer.

2.4 Segmentation

The goal of a segmentation algorithm is to partition each frame into distinct objects. Significant progress has been made on this research topic, although the problem itself is fundamentally ill-posed. For the scope of this work, the segmentation algorithm presented in [22] is employed. This algorithm is based on Normalized Cuts [23] as well as Probability of Boundary (pB) [24] for detection of edges using color and texture information.

The method of [22] is performed using several steps. First, a contour detector is engaged in order to detect changes in brightness. The contour detector may take texture information into account, as is the case in pB. Next, a texture map is computed of the frame using textons [25]. A weight matrix W_{ij} is formed between each pair of pixels using these two cues to measure pixel similarity. Finally, N-Cuts is used to partition the image using information from the weight matrix. Further details can be explored in [22].

As is common in the literature, this segmentation scheme is used to oversegment the image. Each frame is segmented into a predetermined number of regions based on the image size. These settings are presented in Table 2.1. The parameter n_{ev} specifies the number of eigenvectors which will be used for the Normalized Cuts algorithm. These are the eigenvectors with smallest eigenvalue and determine the initial segmentation of the frame. After Normalized Cuts, the image is further segmented into a coarse oversegmentation of n_{spc} regions by using K-means on the initial segmentation. The use of K-means is discussed in [26], and is chosen as it offers robust segmentation when initialized by N-cuts. This step is repeated once more to produce a fine oversegmentation with n_{spf}

Table 2.1: Normalized-Cuts parameters for varying frame size

Frame Size	n_{ev}	n_{spc}	n_{spf}
CIF (352x288)	50	100	200
HD720 (1280x720)	100	200	400
HD1080 (1920x1080)	200	400	800

regions.

One concern with frame segmentation is the presence of motion blur in video sequences [1]. Motion blur is caused by large object motions occurring during the open shutter period of the video camera. Motion blur of sufficient magnitude may cause object boundaries to be falsely determined, thereby decreasing the performance of the proposed algorithm. In practice, this has not proven to be a concern for two reasons. First, any object with extreme motion blur will be travelling rapidly and will be difficult for the HVS to track. In addition, blurred pixels have similar luminance to the associated moving object, because the blur is caused by object regions moving along multiple pixels in the camera sensor. Therefore the proposed algorithm is capable of including proper blurred pixels into the region segmentation.

2.4.1 Merging Oversegmentation

With the frame oversegmented, the next step is to merge regions with similar characteristics. Regions with similar color and texture are merged on the assumption that they belong to the same object. This process is repeated until a small number of regions exist. The merge operation terminates when no two nodes can be located with a sufficiently small dissimilarity.

Color information is obtained directly in the RGB color space. It is important to use color rather than simply relying on luminance information, since a boundary between two objects may be isoluminant while varying between color channels.

In order to compute the texture measure, the variance of the AC coeffi-

cients of the Discrete Cosine Transform (DCT) of each 8x8 block is computed. In Eqs. 2.4- 2.6, the matrix $B_{p,q}$ contains the DCT coefficients for the input block $A_{i,j}$ with parameters $M = 8, N = 8$. The matrix is then vectorized, and the DC coefficient (the first entry) is removed because the mean of block $A_{i,j}$ says nothing about its texture. This results in a vector of length 63 denoted as $\boldsymbol{a}$ The variance of the resulting vector is given in Eq. 2.7 where $N = 63$.

$$B_{p,q} = \alpha_p \alpha_q \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} A_{i,j} \cos\left(\frac{\pi\,(2i+1)\,p}{2M}\right) \cos\left(\frac{\pi\,(2j+1)\,q}{2N}\right) \tag{2.4}$$

$$\alpha_p = \begin{cases} \sqrt{\frac{1}{M}} & p = 0 \\ \sqrt{\frac{2}{M}} & 1 \leq p \leq M-1 \end{cases} \tag{2.5}$$

$$\alpha_q = \begin{cases} \sqrt{\frac{1}{N}} & q = 0 \\ \sqrt{\frac{2}{N}} & 1 \leq q \leq N-1 \end{cases} \tag{2.6}$$

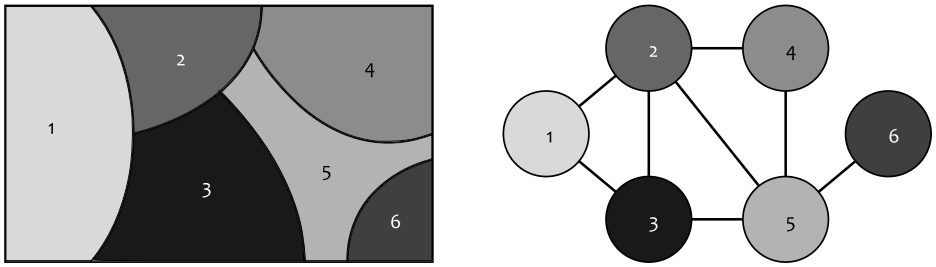
$$var\,(\boldsymbol{a}) = \frac{1}{N} \sum_i a_i^2 - \left(\frac{1}{N} \sum_i a_i\right)^2 \tag{2.7}$$

The superpixel merge procedure is posed as a problem over the graph $G = (V, E)$. Here, $\{v_1, \ldots, v_n\} \in V$ is the set of all superpixel regions, and the edges $\{e_{i,j}\} \in E$ for $i, j \in [1, n]$ contain a dissimilarity measure between each pair of nodes. $E_{ij} = 0$ if nodes $v_i, v_j \in V$ are non-adjacent. We use an indicator function $b_{i,j}$ to represent node adjacency. $b_{i,j} = 1$ if $v_i, v_j \in V$ are adjacent and $b_{i,j} = 0$ otherwise.

$$E_{i,j} = b_{i,j} \left(\lambda \max\left\{\boldsymbol{I}_i^{RGB} - \boldsymbol{I}_j^{RGB}\right\} + (1-\lambda)\,|T_i - T_j|\right) \tag{2.8}$$

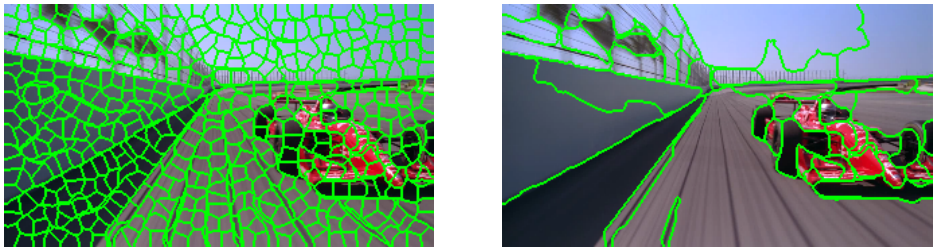
$$\boldsymbol{I}_i^{RGB} = \frac{1}{|\,\{v_i\}\,|} \left[\sum_{j \in v_i} R\,(j)\,, \sum_{j \in v_i} G\,(j)\,, \sum_{j \in v_i} B\,(j)\right]^T \tag{2.9}$$

where $\boldsymbol{I}_i^{RGB}$ is the average intensity over the RGB color planes and T_i is the average texture measure for superpixel region v_i. The tuning parameter λ allows the user to emphasize either color or texture for the merging process. For all experiments conducted in this work, the parameter is set to $\lambda = 0.5$. The



(a) (b)

Figure 2.2: Example graph for superpixel merge operation: (a) partition of region into six distinct superpixel regions, (b) superpixel neighbor graph $G = (V, E)$



(a) (b)

Figure 2.3: Superpixel merge process: (a) oversegmentation of frame from "Speedway" sequence into $n = 200$ regions, (b) merge process after 175 iterations ($n = 25$ regions)

merge procedure requires iteratively locating the pair of nodes $v_i, v_j \in V$ such that $E_{i,j}$ is minimized. These nodes are then merged, and the process continues. An example is demonstrated in Fig. 2.3 for the Speedway sequence.

2.5 Proposed Algorithm

The proposed FRUC architecture improves MV accuracy for salient regions while enforcing smoothness of the MV field for non-salient regions. In this way, both objective and subjective video quality will be increased. The

proposed architecture is detailed in Algorithm 1.

Algorithm 1 Proposed MV Consistency and Refinement

input: frame data, oversegmented and merged region map $R_1, \ldots, R_n$, saliency map S, saliency threshold τ

for region $R_i \in \{R_1, \ldots, R_n\}$ **do**

  if $\frac{1}{|j \in R_i|} \sum_{j \in R_i} S(j) < \tau$ **then**

    enforce region consistency for R_i as discussed in Section 2.5.2

  else

    for all blocks contained in region R_i **do**

      perform MVR as described in Section 2.5.3

    end for

  end if

end for

output: refined MV field

2.5.1 Saliency Map Generation

The saliency map is generated according to [20] with a dynamic texture model used for the feature $\mathbf{Y}$. A spatial window size of 8x8 pixels, and a temporal window size of 11 frames is employed for the spatiotemporal volume. The saliency map is normalized to have a maximum value of 1 pertaining to the most salient points, and a minimum value of 0 for non-salient points. The average saliency value for each region is calculated and compared with a threshold τ to determine whether region consistency or MVR is employed.

2.5.2 Region Consistency

The result of the frame oversegmentation and merging process is a segmentation with n distinct regions $\{R_1, \ldots, R_n\}$ where $R_1 \cup \ldots \cup R_n = I$. In order to promote *natural motion*, we restrict the candidate set of available motions to those which are statistically most likely. A MV histogram is computed

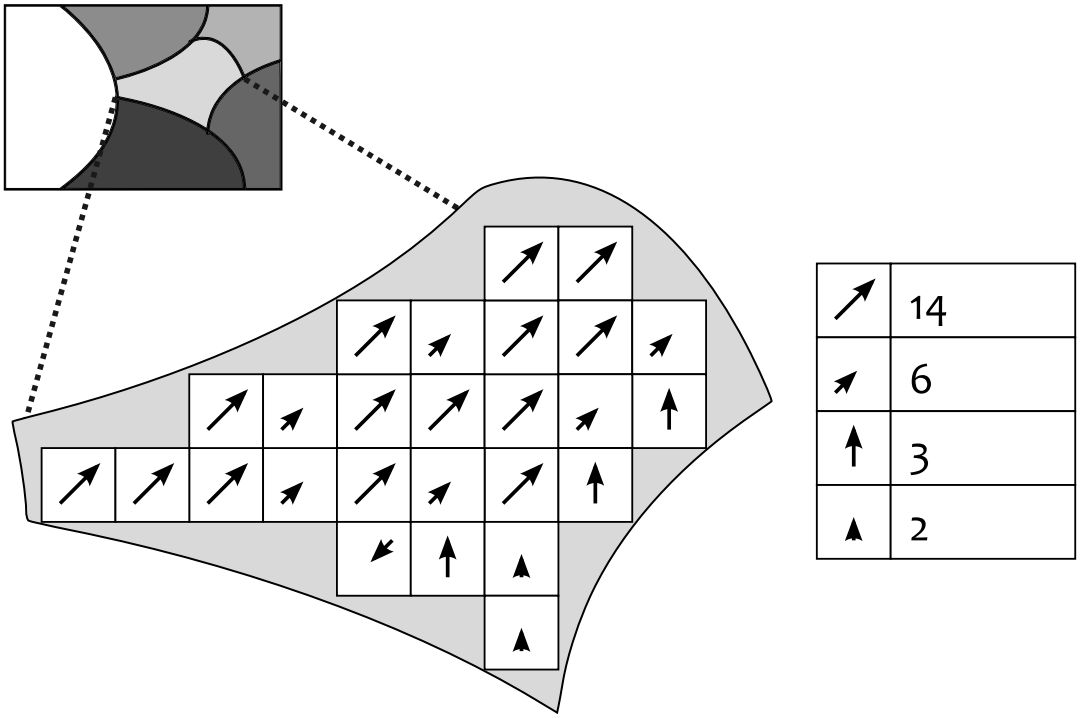


Figure 2.4: Toy example for region consistency algorithm. The upper left portion demonstrates a frame which has been segmented into $n = 6$ regions. We create a MV histogram for region R_3 and select the $m = 4$ most commonly occurring motions for the candidate set $CS(R_3)$

for each region R_i consisting of the motions assigned to all blocks $B \in R_i$. From this histogram, the m most commonly occurring motions are promoted as a candidate set. This process is demonstrated in Fig. 2.4. Selection of the parameter m is discussed in Section 2.5.5. Denote the candidate set for region R_i as $CS(R_i) = \{\boldsymbol{mv_1}, \ldots, \boldsymbol{mv_m}\}$.

For each candidate $\boldsymbol{mv_j}$ in the candidate set, the Total Error $TE(\boldsymbol{mv_j}, R_i)$ is calculated over region R_i to determine which candidate best explains the total motion of the region. Denote the x and y-components of candidate $\boldsymbol{mv_j}$ as v_{jx} and v_{jy}, respectively. For reference frame f_{t-1} and current frame f_t, TE is

computed as:

$$TE\left(\boldsymbol{mv_j}, R_i\right) = \sum_{M \in R_i} \sum_{x,y \in M} |f_{t-1}\left(x + \frac{v_{jx}}{2}, y + \frac{v_{jy}}{2}\right) - f_t\left(x - \frac{v_{jx}}{2}, y - \frac{v_{jy}}{2}\right)| \quad (2.10)$$

where M is a block contained in region R_i with upper-left pixel index (i, j). Block ownership is determined by which region owns a majority of the block's pixels. Ties are broken arbitrarily. Penalties are applied to these candidates based on the total distortion produced by the candidate for the region R_i. In case of non-integer offsets ($\frac{v_{jx}}{2}, \frac{v_{jy}}{2} \notin \mathbb{Z}$), bilinear interpolation is used to determine TE.For candidate $\boldsymbol{mv_j} \in CS\left(R_i\right)$:

$$p\left(\boldsymbol{mv_j}\right) = \frac{TE\left(\boldsymbol{mv_j}, R_i\right)}{\sum_{k \neq j} TE\left(\boldsymbol{mv_k}, R_i\right)} \quad (2.11)$$

With the penalties determined over the candidate set, we are now able to promote MV consistency for each superpixel region. The Region Consistent MV ($\boldsymbol{mv_{rc}}$) for a block $B \in R_i$ is computed as:

$$\boldsymbol{mv_{rc}} = \min_{j:\boldsymbol{mv_j} \in CS(R_i)} \sum_{x,y \in M} |f_{t-1}\left(x + \frac{v_{jx}}{2}, y + \frac{v_{jy}}{2}\right) - f_t\left(x - \frac{v_{jx}}{2}, y - \frac{v_{jy}}{2}\right)| p\left(\boldsymbol{mv_j}\right) \quad (2.12)$$

2.5.3 Motion Vector Refinement

For scene regions which exceed the saliency threshold τ, Motion Vector Refinement (MVR) is applied to increase the accuracy of the motion field. The refinement is computed without motion re-estimation in an approach similar to [27]. MVR is computed in three stages of decreasing local neighborhood, which is particularly important at object boundaries, where the MV field is difficult to determine. The method is based on the idea of *natural motion*, this is the assumption that, for any given area, there are a limited number of motions which need to be considered. The candidate selection process is demonstrated in Fig. 2.5. MVR is computed in multiple stages in order to improve the ac-

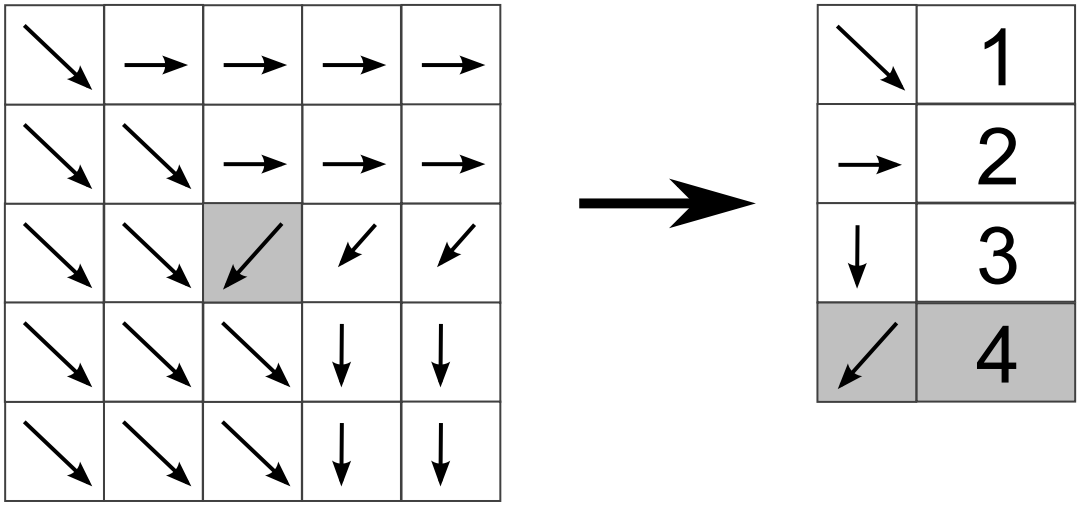


Figure 2.5: Proposed candidate selection method for the center block (gray) with parameter $m = 3$. Here, the top three most commonly occurring motions in the neighborhood are considered as the first three motion vector candidates. The original motion vector for the center block is the fourth candidate.

curacy of the motion field around object boundaries. At each stage, the local neighborhood of consideration is decreased in order to consider more relevant MV candidates. In the first stage, enlarged block matching is considered with a 24x24 pixel measurement window for each 8x8 block. A MV histogram is created containing the original block motion and all spatial neighbors within a neighborhood of ± 2 blocks. These 25 MVs are analyzed, and the $m = 3$ most commonly occurring motions, as well as the original block motion, are promoted as a candidate set. As before, the candidate which produces the smallest error is chosen as the MV. For stage one, the error is calculated as:

$$\begin{aligned} SAD_1\left(v_x, v_y\right) &= \sum_{x,y \in M_1} |f_{t-1}\left(x + \frac{v_x}{2}, y + \frac{v_y}{2}\right) \\ &\quad - f_t\left(x - \frac{v_x}{2}, y - \frac{v_y}{2}\right)| \end{aligned} \tag{2.13}$$

using the Sum of Absolute Differences (SAD) error measure where M_1 is defined as in Eq. (2.14) for a 24x24 pixel enlarged measurement window with upper-left pixel located at (i, j). The second stage proceeds in a similar fashion. The

candidate set is increased to four motion histogram candidates and the original block motion. An 8x8 block is selected with no enlarged matching to improve the motion accuracy around object boundaries. The error for stage 2 is computed using block M_2.

In the third stage, the resolution of the motion field is increased by a factor of two in each direction. Each block is partitioned into four 4x4 subblocks (quadrants), and refinement proceeds as in previous stages. The four subblocks are defined by $M_{3i}, i = 1, \ldots, 4$

$$\begin{aligned} M_1 &= \{x, y : x \in [i-8, i+15], y \in [j-8, j+15]\} \\ M_2 &= \{x, y : x \in [i, i+7], y \in [j, j+7]\} \\ M_{31} &= \{x, y : x \in [i, i+3], y \in [j, j+3]\} \\ M_{32} &= \{x, y : x \in [i, i+3], y \in [j+4, j+7]\} \\ M_{33} &= \{x, y : x \in [i+4, i+7], y \in [j, j+3]\} \\ M_{34} &= \{x, y : x \in [i+4, i+7], y \in [j+4, j+7]\} \end{aligned} \tag{2.14}$$

2.5.4 Inconsistency between boundaries

It is common for the object boundaries determined by the segmentation algorithm to disagree with motion-based saliency boundaries. This is no cause for concern as the decision to perform MVR is based on the average saliency value of a region as determined by the segmentation algorithm. Therefore the processing method will be consistent within each region. This is the case because the segmentation boundaries are based on color and texture information, while saliency boundaries are based on differences in motion models. A broken color or texture edge would be disruptive to the HVS, and therefore is avoided by the proposed algorithm.

2.5.5 Parameter Selection

The proposed algorithm depends on several parameters which must be tuned for optimal performance. These include the saliency threshold τ, the

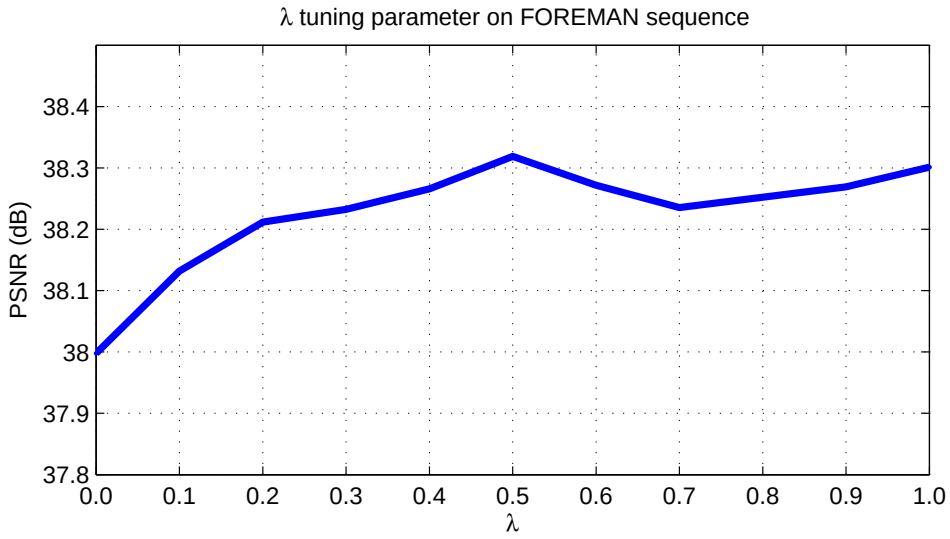


Figure 2.6: PSNR of the proposed algorithm as a function of the parameter λ. performance is very nearly constant for $\lambda > 0.2$, with a peak at $\lambda = 0.5$.

candidate set size m, the balance between intensity and texture (λ in Eq. (2.8)), and the temporal window size for dynamic texture computation. The saliency threshold is heuristically set to 0.75. Recall from algorithm 1 that each region is processed based on the average saliency value. Therefore, any region with an average saliency value in the top 25% of the frame will be subjected to MVR. τ is fixed across all sequences. Next, m for region consistency is determined through testing. In Fig. 2.7 we measure the objective algorithmic performance as a function of the candidate set size. The **planes** sequence is used, with error averaged over all frames. Notice that PSNR achieves its maximum at $m = 2$. This is because with $m = 1$ candidate, the algorithm must blindly accept the majority vote. On the other hand, when $m > 3$, the regularizing effect of m disappears, thereby decreasing performance. SSIM decreases almost monotonically with increasing m. SSIM achieves its maximum at $m = 1$ because this forces large uniform motion fields. A good compromise on PSNR and SSIM performance is attained by selecting $m = 2$. Similar methods are employed to determine suitable parameters for the MVR candidate sets. We arrive at $m = 3$ for the first stage, and $m = 4$ for stages two and three. The parameter λ is

chosen such that luminance and texture contribute equally to the region merging algorithm. This equal weighting was selected due to experimental results as demonstrated in Fig. 2.6. Here, the **foreman** sequence is interpolated with all parameters other than λ held fixed. The given PSNR results are an average over the entire frame sequence. It is observed that performance is degraded when only texture information is used for the merging algorithm. However, for $\lambda > 0.2$, performance is nearly constant with the highest value attributed to $\lambda = 0.5$. It is concluded that color information is the most important for the region merging procedure, however the addition of texture information improves performance. From these results, we conclude that λ can be fixed to 0.5 for all experiments.

Finally, the dimensions of the spatiotemporal volume for dynamic texture computation are selected as discussed in [28]. In this work, the authors investigate the sensitivity of Discriminant Saliency with respect to the temporal window size τ and the spatial window size. It is concluded that performance remains nearly constant over a large gamut of spatiotemporal sizes. Because the range $\tau \in [5, 21]$ provides for roughly uniform performance, $\tau = 11$ is selected. While the authors believe that performance can be fine-tuned using an optimization over τ, it would not boost performance commensurate with the complexity of finding the optimal τ for each frame.

2.6 Experimental Setup

Objective results are calculated using the following experimental procedure. Each 24 frame per second (fps) video sequence is temporally reduced by a factor of two to 12fps. The 12fps sequence is then up-converted using MCFI via one of the FRUC algorithms discussed previously. The resulting interpolated frames are compared with the originals to determine the error.

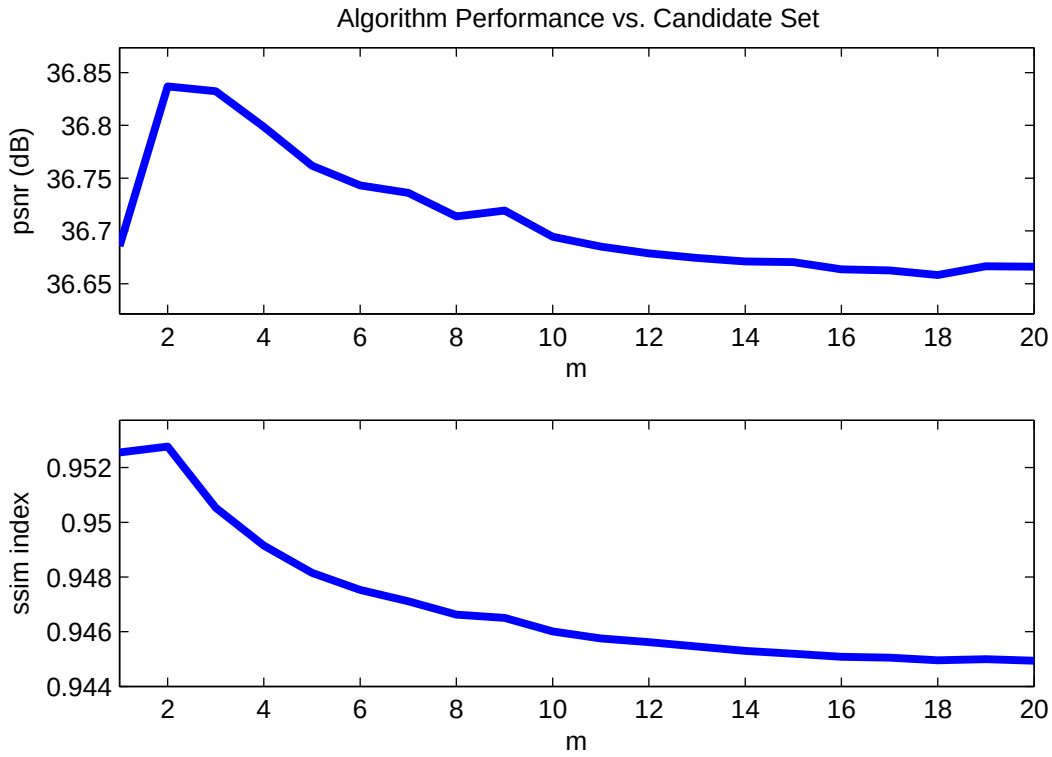


Figure 2.7: Algorithm performance decays as the region consistency candidate set size (m) increases.

2.6.1 Objective Results

The proposed algorithm is tested against several competing methods for FRUC. Among these are: Full Search (FS) with bidirectional MCFI [29], 3D Recursive Search (3DRS) [2], MSEA method with bidirectional MCFI [30] and a Multistage Motion Vector Processing method (MMVP) [31]. The metrics for comparison are Peak Signal to Noise Ratio (PSNR), which is calculated as the average Mean-Squared Error (MSE) of the predicted frame; and Structural Similarity Index (SSIM), which models error as perceived by a Human observer [32]. Eight sequences have been selected for comparison. Among these are four CIF sequences (352x288) and four HD720p sequences (1280x720). The CIF sequences are: **coastguard**, **football**, **foreman** and **tennis**. These sequences are prevalent in the video processing literature. The HD sequences

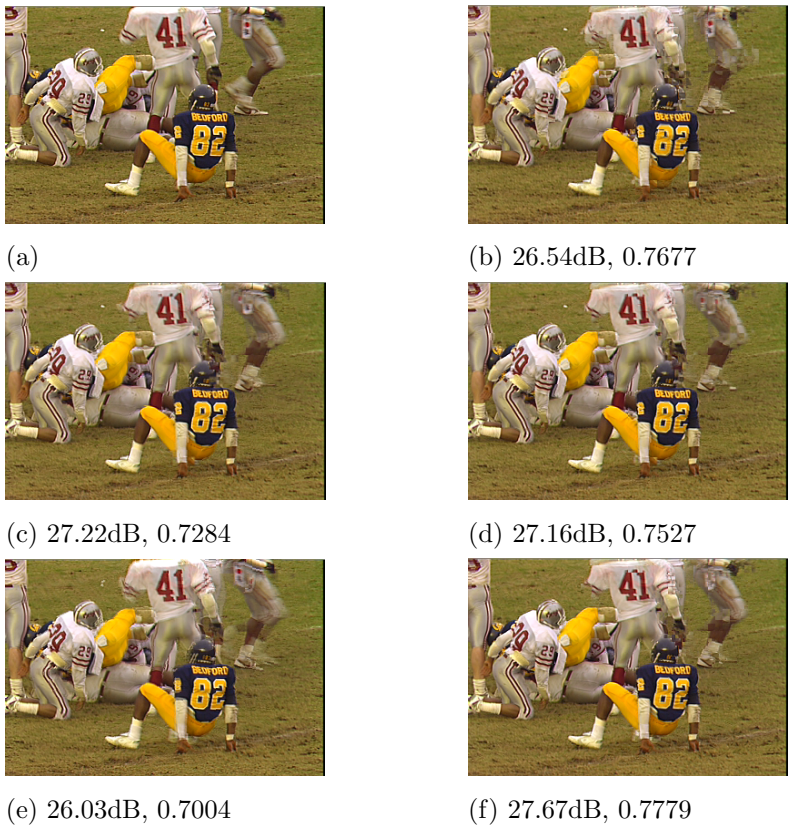
are: **dolphins**, **limit**, **planes** and **speedway**. All objective results for these sequences are tabulated in Table 2.2. Additional results including video sequences, saliency information, and all error plots are located on our web site [1]. In addition to objective results for full-frame comparison, performance results for the salient regions are included. We consider the top 25% of each saliency map as the mask for calculation of objective results in the salient region. This is consistent with our goal of improving the performance of FRUC most in visually salient regions.

First, the **football** sequence is examined. This is a difficult task for any FRUC engine, as there are many different motions, occlusions, and therefore numerous object boundaries. An example interpolated frame is demonstrated in Fig. 2.8 using the five FRUC methods. The most noticeable distortion occurs around the object boundary of player #41 in the middle of the frame. This is most evident in the 3DRS interpolation shown in Fig. 2.8(b). Here, significant blocking artifacts can be seen on the arms of player #41, as well as the leg of the player on the right side of the frame. Interpolation performance increases for the FS and MSEA methods in Figs. 2.8(c,d), which can be seen in the improved boundary of player #41. However, there are still errors in the leg of the player on the right of the frame. Because no constraints are imposed to promote consistent motion of objects, the previous methods all fail to properly assign motion in this region. The MMVP method in Fig. 2.8(e) combines block motions with high residuals, thus changing the appearance of the leg of the player on the right. The merging of motion vectors creates a consistent motion in this region, however the motion is too large. The result is duplication of the leg appearing as a ghosted copy. Finally, the proposed interpolation in Fig. 2.8(f) demonstrates consistent motion of the player on the right side of the frame. In addition, the saliency map determined for this frame sequence allows for motion vector refinement to player #41, resulting in further improvement over Full Search and MSEA. The error for **football** sequence is plotted as a function of frame number in Fig.

[1] Additional results including video files may be found online at `http://videoprocessing.ucsd.edu/~NatanHaim/TIP_2011a/`.

Table 2.2: Objective results for CIF and HD720p test sequences. Each cell provides results in PSNR (top row) and SSIM (bottom row)

Sequence	3DRS [2]	FS [29]	MSEA [30]	MMVP [31]	Proposed
Entire frame					
coastguard	34.4422 0.8973	36.9724 0.9444	37.0120 0.9448	36.0431 0.9401	**37.5361** **0.9505**
football	24.9455 0.7422	25.7013 0.7602	25.7035 0.7616	24.5524 0.6847	**26.0087** **0.7885**
foreman	37.6367 0.9413	38.5156 0.9499	**38.5159** 0.9502	34.6369 0.9416	38.4558 **0.9530**
tennis	31.3513 0.8689	31.6365 0.8559	31.5762 0.8575	28.7834 0.7393	**31.8027** **0.8737**
dolphins	34.0322 0.8585	35.1030 0.8790	35.0952 0.8814	**35.1120** **0.8835**	34.9936 0.8832
limit	39.3535 0.9151	39.2591 0.9156	39.2382 0.9150	39.4234 0.9159	**39.5608** **0.9209**
planes	34.2114 0.9258	36.3117 **0.9517**	36.2967 0.9510	36.3942 0.9469	**36.8768** 0.9516
speedway	28.9685 0.8517	29.3508 0.8673	29.3658 0.8670	**29.3960** 0.8638	29.3729 **0.8687**
Salient frame region					
coastguard	32.4071 0.9833	33.8265 0.9866	33.8948 0.9868	33.7229 0.9858	**34.8572** **0.9893**
football	23.0517 0.9441	23.8816 0.9511	23.9127 0.9513	22.5563 0.9334	**24.2448** **0.9571**
foreman	36.1712 0.9869	37.5657 0.9904	37.6198 0.9905	36.4874 0.9883	**37.6827** **0.9911**
tennis	29.7027 0.9667	30.7389 0.9664	30.6444 0.9666	27.3591 0.9430	**31.3106** **0.9733**
dolphins	30.6850 0.9417	31.8903 0.9504	31.8539 0.9511	31.6042 0.9497	**31.9006** **0.9537**
limit	37.5492 0.9855	38.2784 0.9866	38.5500 0.9871	38.2704 0.9866	**38.6604** **0.9876**
planes	36.6685 0.9940	37.1436 0.9950	37.2119 0.9950	37.1292 0.9944	**38.2912** **0.9952**
speedway	25.7847 0.9335	26.6485 0.9407	26.6092 0.9404	26.6632 0.9408	**26.6846** **0.9411**



(a) (b) 26.54dB, 0.7677

(c) 27.22dB, 0.7284 (d) 27.16dB, 0.7527

(e) 26.03dB, 0.7004 (f) 27.67dB, 0.7779

Figure 2.8: Objective FRUC results for **football** sequence frame 74: (a) Original CIF frame, (b) 3DRS, (c) FS, (d) MSEA, (e) MMVP, (f) Proposed. PSNR, SSIM results shown for each frame.

2.9(a,c). It becomes evident that the proposed method consistently improves interpolation quality for this scene. The average PSNR and SSIM values for this sequence are considerably higher than the four competing methods. The difference between the proposed method and the competing methods is plotted in Fig. 2.9(b,d) as a function of frame number. Here a positive value indicates that the proposed method outperforms the competing method. Notice that for all but a few frames, the PSNR and SSIM differences are positive when compared with 3DRS, FS, MSEA and MMVP.

The **foreman** sequence is examined in Fig. 2.10. In this sequence, 3DRS produces poor results. Notice distortion to the facial structure in Fig. 2.10(b).

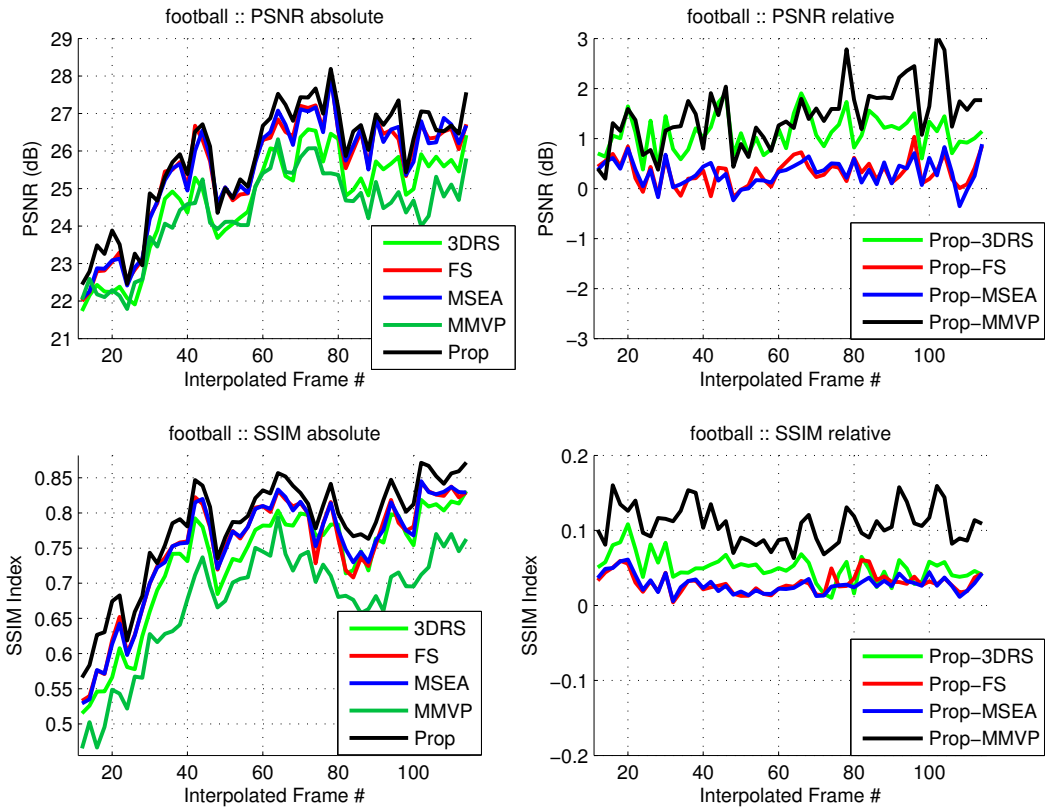
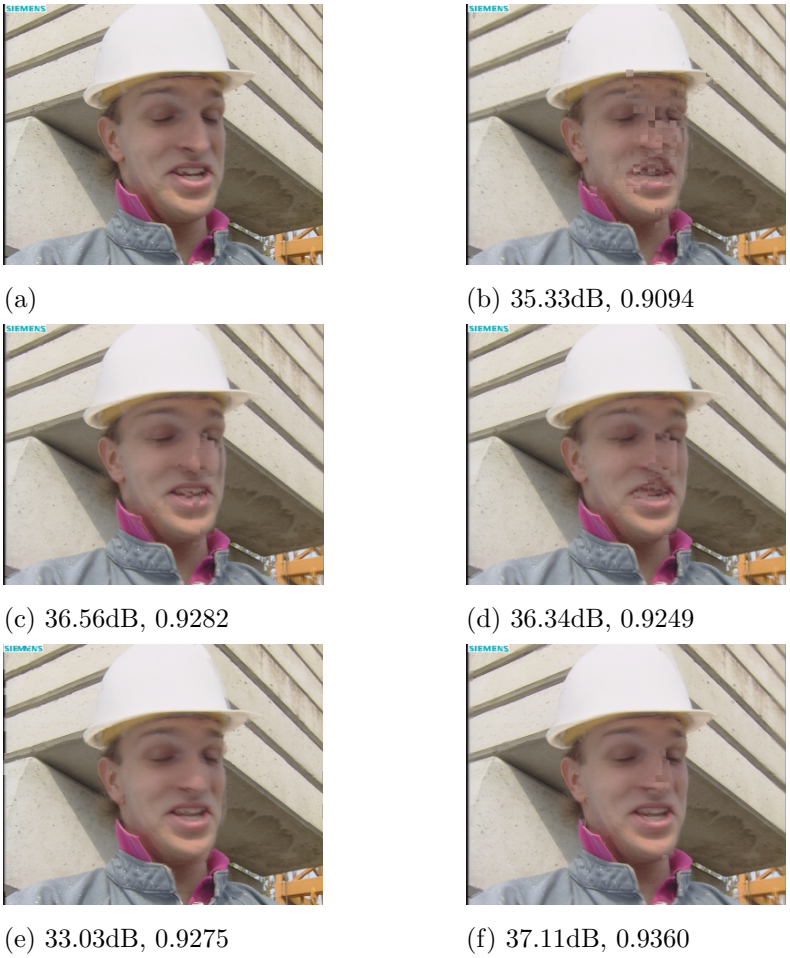


Figure 2.9: Objective performance of Proposed FRUC algorithm for interpolated frames 12 through 114 of **football** sequence: (a) PSNR, (b) relative PSNR, (c) SSIM, (d) relative SSIM. Relative graphs compare the performance of methods 1-4 to that of the proposed method. A positive score denotes higher performance from the proposed method.

This is caused by the way in which 3DRS chooses motion vector candidates spatially and temporally. The facial motion is complex and proper motion vectors may not exist in the above spatial and below temporal candidates. Fig. 2.10(c) demonstrates the advantage of FS over 3DRS. Here, the facial structure is mostly maintained, aside from an incorrect patch on the left side of the face. MSEA also results in noticeable distortion to the face in Fig. 2.10(d). The MMVP algorithm in Fig. 2.10(e) merges the face region and its immediate surround to a single motion. While this noticeably improves interpolation of the nose, it causes blurring at the edges of the face. The proposed algorithm



(a) (b) 35.33dB, 0.9094

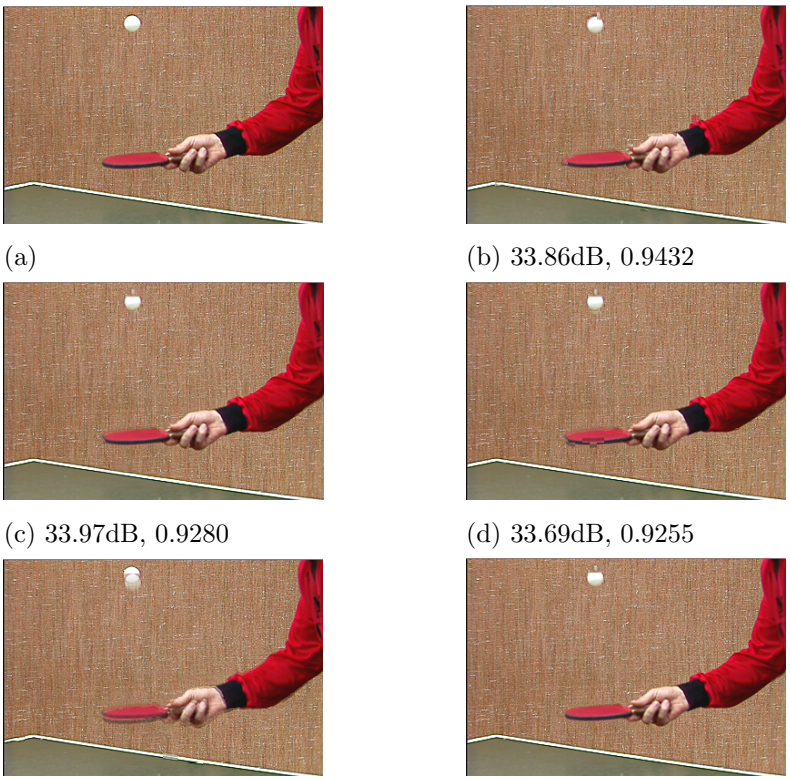
(c) 36.56dB, 0.9282 (d) 36.34dB, 0.9249

(e) 33.03dB, 0.9275 (f) 37.11dB, 0.9360

Figure 2.10: Objective FRUC results for **foreman** sequence frame 78: (a) Original CIF frame, (b) 3DRS, (c) FS, (d) MSEA, (e) MMVP, (f) Proposed. PSNR, SSIM results shown for each frame.

addresses the shortcomings of competing methods by refining the motion field for the salient nose region while enforcing consistency in the background. This can be observed by in the interpolated frame in Fig. 2.10(f).

Objective results for the **tennis** sequence are demonstrated in Fig. 2.11. This is a complicated scene involving a moving ping-pong ball and paddle against a textured background. There are significant errors at the boundary of both the arm and paddle caused by the 3DRS method. This can be seen in



(a) (b) 33.86dB, 0.9432

(c) 33.97dB, 0.9280 (d) 33.69dB, 0.9255

(e) 31.82dB, 0.8877 (f) 34.72dB, 0.9515

Figure 2.11: Objective FRUC results for **tennis** sequence frame 20: (a) Original CIF frame, (b) 3DRS, (c) FS, (d) MSEA, (e) MMVP, (f) Proposed. PSNR, SSIM results shown for each frame.

Fig. 2.11(b). Similar errors at the boundary of the paddle can be observed with methods 2.11(c,d). The MMVP method in Fig. 2.11(e) aims to resolve this problem, however is unable to do so properly without segmentation information. The result is a ghosted copy of the paddle. The advantage of the proposed method in this instance comes from the use of segmentation information. Because the paddle is determined to be a single object, the proposed method advances motion vectors which treat the paddle as a rigid object. The benefit of this feature can be observed in Fig. 2.11(f). In addition, the saliency detector ensures that motion vector candidates are refined for the region including the paddle and arm.

2.6.2 Subjective Results

In addition to objective results, it is crucial to determine the perceptual quality of the proposed algorithm. This is accomplished by performing double-blind subjective testing on a group of Human observers. Subjective results are obtained using the stimulus comparison non-categorical judgment method as described in [33]. A selected group of 20 observers were shown video clips which had been processed by the proposed method, in addition to 3DRS, FS and MSEA methods. In each instance, two video clips are shown side-by-side with each processed via a different method. The observer is presented with a rating scale on the range $[-3, 3]$, where a score of -3 corresponds with the left side appearing "much better" than the right side, and 3 corresponding with the right side "much better" than the left side. Any score between these two values is acceptable with 0 representing "no difference" between the two sequences. Findings are tabulated in Table 2.3 for the sequences: **football**, **planes**, **speedway** and **tennis** across all 20 observers. In this table, the mean (μ) and standard deviation (σ) are calculated for each sequence where a positive score on the mean corresponds to a perceptual improvement of the proposed method over the competing method. The rejection region (γ) is calculated using the Student's T-Test, where a decision is made between the null hypothesis (the proposed algorithm has no positive affect over the competing method) and the alternative hypothesis. Therefore, a mean score exceeding the calculated rejection region (τ) corresponds to a statistical improvement of the proposed method.

According to the subjective results, the proposed algorithm demonstrates a significant improvement over the competing methods for both HD sequences. However, no telling results are obtained for the CIF sequences. While the objective results are positive for the CIF sequences, the video size is too small for a significant perceptual improvement.

Table 2.3: Subjective results for two CIF sequences and two 720p sequences across 20 human observers. A positive score on the average corresponds with a perceptual improvement of the proposed method when compared with each competing method.

Sequence	Comp. Method	Std. Dev	Rej. Region	Average
Football	3DRS	0.50	0.19	**2.34**
	FULL	1.02	0.39	0.21
	MSEA	0.74	0.29	−0.15
Planes	3DRS	0.55	0.21	**2.24**
	FULL	1.26	0.49	**1.11**
	MSEA	0.77	0.30	**1.48**
Speedway	3DRS	0.30	0.12	**2.81**
	FULL	0.99	0.38	**0.78**
	MSEA	1.15	0.44	**0.85**
Tennis	3DRS	1.22	0.47	**1.51**
	FULL	0.51	0.20	**0.21**
	MSEA	0.86	0.33	0.26

2.6.3 Computational Complexity

Complexity of the proposed algorithm is dependent on the methods used for saliency and segmentation calculation. For saliency, the method of [20] is used which can calculate the saliency for a 720p frame in roughly 15 seconds. This assumes that the frame is downsampled by a factor of four to 320×180. Segmentation of the same downsampled frame size is computed using an implementation of [34] which completes in 60 seconds. The remainder of the algorithm, including the region merging procedure, requires 10 seconds. In order to realize real-time performance, certain tradeoffs may be considered. For example, the saliency and segmentation maps may be computed every n frames and propagated using the refined/consistent motion vector field.

2.7 Conclusion

Over the past decades, there has been significant research in methods to improve the performance of FRUC algorithms. This research has been fuelled by the high adoption rates of LCD televisions and the developing demand for mobile video. In this work, we have proposed a novel method for FRUC which incorporates the ideas of frame segmentation and discriminant saliency. By exploiting these methods, we are able to increase the performance of interpolated image quality, especially in visually salient regions.

The text of Chapter 2 is adapted from *A Novel Approach to FRUC using Discriminant Saliency and Frame Segmentation*, Natan Jacobson, Yen-Lin Lee, Vijay Mahadevan, Nuno Vasconcelos, Truong Nguyen, *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2924-2934, November 2010. The dissertation author is the primary author of this publication.

Chapter 3

Scale-Aware Saliency

"We can allow satellites, planets, suns, universe, nay whole systems of universe, to be governed by laws, but the smallest insect, we wish to be created at once by special act." - Charles Darwin

The application of saliency to Frame Rate Up-Conversion has proven beneficial. We have shown that both objective and subjective performance can be improved by restricting motion vector refinement to the salient scene regions. We show further evidence of this fact through the use of an improved saliency detector and revised FRUC procedure. By considering object scale, we are able to improve saliency detection, validated by a dataset of measured human fixations. This improved saliency map is then used to selectively perform Frame Rate Up-Conversion. Improved results are demonstrated with respect to the previous work, as well as a number of competing methods.

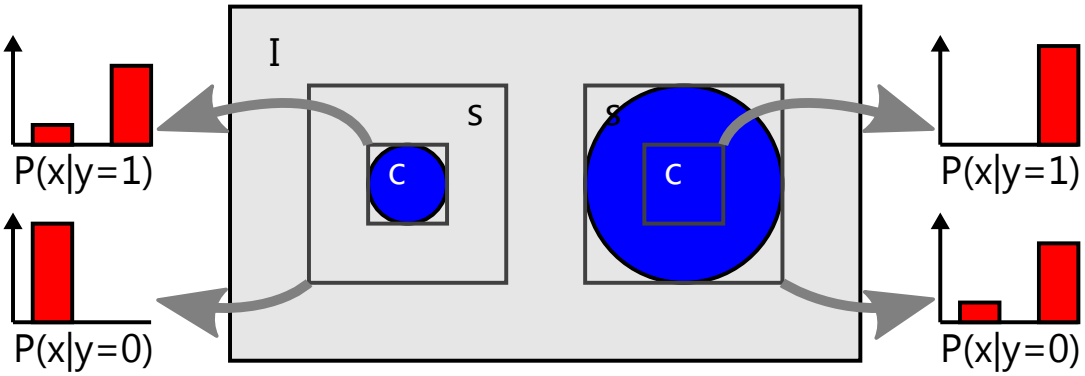


Figure 3.1: Illustration of the fixed-scale saliency detection problem. The selected scale is well tuned for the object on the left of the frame, but ill-suited for the object on the right. In the proposed work, both objects are detected through a scale-aware saliency framework.

3.1 Scale Problem

In the saliency literature, little attention has been paid to explicitly considering scale information in any of these stages when computing a saliency map. The importance of this consideration is demonstrated using an example.

The problem of scale selection in a center/surround saliency framework is illustrated in Fig. 3.1. For the sake of this example, consider color as the sole discriminating feature. Therefore the saliency measure is the ability of the color feature to discriminate between the center and surround windows. For the object on the left, the color feature can easily discriminate between the two regions, yielding a high saliency score. However, the object on the right is too large to be contained within the center window, and will therefore yield a much lower saliency value. This is unfortunate as a human observer will declare both objects to be equally salient.

In this work, a scale-aware algorithm is proposed to improve saliency detection performance for images and video. Performance of the proposed detector is assessed using a database of human fixations [36]. Application of the proposed method towards video processing is investigated in terms of video

enhancement for Frame Rate Up-Conversion (FRUC). This technique has recently gained popularity for motion-blur reduction in LCD HDTV as well as decoder-side mobile video reconstruction. This section is organized as follows. Previous work in the field of saliency detection is discussed in Section 3.2. Next, we introduce the biologically-plausible method of Discriminant Saliency in Section 3.3 for both static and motion-based applications. The proposed algorithm is described in detail in Section 5.4, and compared with results from the Human fixation database in 3.5. Next, the application of Frame Rate Up-Conversion is explored in 3.6. Finally, the proposed work is concluded in Section 3.7.

3.2 Previous Work

A number of approaches to saliency detection have been proposed in the literature. These methods range in biological plausibility as well as computational complexity. The earliest method explored in this work is the center/surround approach of [37]. Here, saliency is modelled using local feature contrast of color, brightness and orientation features. Scale is incorporated by computing the feature contrasts on four levels of a Gaussian pyramid, and combining these in a linear fashion. Next, the method of [36] is explored which utilizes an information theoretic approach to determining saliency. This method is based on a connection between sparsity of image statistics and simple-cell receptive fields in the primary visual cortex. Local image statistics are learned from a set of $3,600$ natural images and the self-information for each local image patch is used to measure the saliency value. A significant departure is explored in the spectral residual method [38]. In this approach, the information content of each image is modelled as the information owing to *i*nnovation, and that owing to *p*rior knowledge. Saliency is then seen as an information coding mechanism aimed at reducing visual redundancy. The *n*ovelty of each image is computed as the difference between the log-spectrum of the image, and the log-spectrum averaged over a large set of natural images. The method of spectral residual has been extended in [39] to include multiple feature responses, including the frame

difference of adjacent frames for video applications. This method is known as the phase spectrum of the quaternion Fourier transform (PQFT). In addition, we explore the discriminant saliency detector on which the proposed work is based [20]. Discriminant saliency is explored using all default parameters (center window of 24 pixels, center-surround ratio of 6.0). Next, the graph-based method of [35] is investigated. The entire image is treated as an undirected graph, with the weight between each pair of nodes measuring their dissimilarity. A Markov chain is defined for this graph by normalizing edge weights, and the equilibrium distribution provides information on attentive locations. A graph-based method is also applied to normalize the attention map for the final saliency result. Finally, two recent methods are explored which measure saliency using self-resemblance [40], and by including global context [41].

Many of these methods have been applied to video processing applications in the past. In the work of [28], discriminant saliency using a motion cue is applied to the task of background subtraction for video. Classification of a video sequence into foreground and background is accomplished by thresholding the saliency value for each frame. Performance of this algorithm is quite good compared with previous work. In another application, saliency is used as a pre-processing step to improve video compression [8]. The video is blurred for non-salient regions which increases the effectiveness of the video encoder's spatial prediction algorithm. The result is a reduced bitrate while maintaining high fidelity for salient regions.

3.3 Discriminant Saliency

Discriminant center-surround saliency is a biologically plausible algorithm that has been shown to replicate the psychophysics of saliency mechanisms in the human visual system (HVS). This method has been applied to static images [20] as well as video sequences [28, 42]. In the case of static saliency, the feature set is comprised of: color, brightness and Gabor filter orientation. For video sequences, a motion model such as optical flow or dynamic textures [21]

is employed.

The dynamic texture model for motion-based discriminant saliency is an Autoregressive Moving Average (ARMA) model for spatiotemporal data. This data is a three-dimensional volume with two spatial dimensions and one time dimension; representing an observed sequence $\{y(t)\}$ seen as the output of a dynamic texture $\{I(t)\}$ with added noise $n(t)$. Using this notation, the dynamic texture coefficients can be determined using the following process:

$$\begin{cases} x(t) &= \sum_{i=1}^{k} A_i x(t-i) + Bv(t) \\ y(t) &= \phi(x(t)) + n(t) \end{cases} \tag{3.1}$$

where ϕ is a spatial filter, $I(t) = \phi(x(t))$, $v(t)$ is selected Independently and Identically Distributed (IID) from an unknown distribution, and $n(t)$ is selected IID from a given distribution $p_n(\cdot)$.

Discriminant saliency is defined with respect to two classes of stimuli and a feature $\mathbf{Y}$: the class of visual stimuli in the center (with label $C = 1$), and class of visual stimuli in the *background* or surround (with label $C = 0$). The saliency of location l of the video, denoted $S(l)$, is the extent to which the feature $\mathbf{Y}$ can discriminate between *center* and *surround* at l. This is quantified by the mutual information between features, $\mathbf{Y}$, and class label, C,

$$S(l) = I_l(\mathbf{Y}; C) = \sum_{c=0}^{1} \int p_{Y,C(l)}(\mathbf{y}, c) \log \frac{p_{Y,C(l)}(\mathbf{y}, c)}{p_Y(\mathbf{y}) p_{C(l)}(c)} d\mathbf{y}. \tag{3.2}$$

A large $S(l)$ implies that center and surround have a large disparity of feature responses, i.e. large *local feature contrast* indicating that the location is salient. The saliency can be calculated efficiently using moment estimation via the simplifying assumption that natural image statistics follow a Generalized Gaussian Distribution (GGD), discussed in [43].

3.4 Proposed Method

A flowchart for the proposed method is shown in Fig. 3.2. In the left half of the flowchart, calculation of the scale map $\Theta(x, y)$ is demonstrated. In the

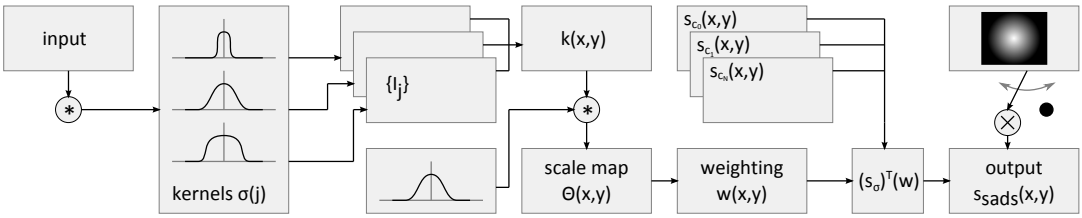


Figure 3.2: Flowchart for the proposed scale-aware saliency algorithm. The input is an image or a single frame from a video sequence. The array $s_\sigma(x, y)$ is comprised of discriminant saliency maps with varying center/surround parameters. The switch in the upper-right of the flow diagram represents element-wise multiplication with a Gaussian kernel to simulate center-bias for certain applications.

right half, the scale map is used as a soft-weighting over the set of discriminant saliency maps.

3.4.1 Scale Space

The efficient method due to [44] is exploited in which a map of N scales is produced. The scale-space representation is created by convolving each image with a set of Gaussian kernels of increasing standard deviation. The amount by which a pixel deviates from its original value as the blur increases is measured. A pixel which varies only slightly with increasing kernel width is denoted as a large-scale pixel. Similarly, a pixel for which dramatic deviation is computed will be denoted as a small-scale pixel. The scales are defined in Eq. (3.3), where $N = 5$ for all experiments conducted in this research. Distance is determined using a Euclidean norm over the CIELAB (L*a*b*) color space. This color space was selected because of its perceptual uniformity - a change in the color value corresponds with a commensurate change in visual importance.

$$\left\{\sigma(j) = 1.9^j | 0 \leq j \leq N\right\} \rightarrow \{I_j\} \tag{3.3}$$

In the first step of scale map generation, the input image is convolved with a bank of Gaussian kernels, distributed as $\mathcal{N}(0, \sigma(j))$ to produce the set of

filtered images $\{I_j\}$. Denote I_0 as the original image, and I_j the filtered image with kernel standard deviation $\sigma(j)$. The scale-space measure at pixel (x, y), denoted $k(x, y)$ is the maximum scale j such that the deviation between $I_j(x, y)$ and $I_0(x, y)$ is less than a given threshold β. For all subsequent work we fix $\beta = 10$.

$$\| I_j(x, y) - I_0(x, y) \|_2 < \beta \qquad \forall j \leq k(x, y) \tag{3.4}$$

The resultant scale map $k(x, y)$ is smoothed using a Gaussian kernel $h \sim \mathcal{N}(0, \sigma_s)$ where $\sigma_s = 5$ to produce $\hat{k} = k \star h$. The final scale-map is given by $\Theta(x, y) = \langle \hat{k}(x, y) \rangle$, where $\langle \cdot \rangle$ denotes rounding to the nearest integer.

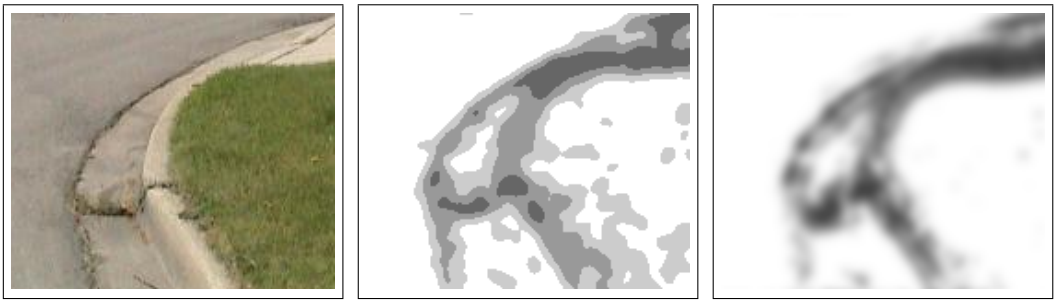
3.4.2 Texture Cue

A texture cue is included in the proposed scale-space model to supplement color information in the distance calculation. This is an important addition as regions with uniform texture will be considered as homogeneous regions by the human visual system, although changes in luminance and color may be present. An example of this phenomenon is evident in the grass field of Fig. 3.3(a). The standard scale map using the L*a*b* color space is shown in Fig. 3.3(b). Here, the grass region is classified both as large and small scale for different patches. This conflicts with the notion that the grass is a single homogeneous region. In Fig. 3.3(c), the texture cue is included, and the grass is now classified as a single large scale patch.

In this work, *t*exture is captured by the distribution of AC coefficients in the Discrete Cosine Transform (DCT) of image data. Only the luminance of the image is considered for texture, as the chrominance channels contain less information about image structure. For pixel location (i, j), a local patch is defined as:

$$B_{i,j} = \{ I(x, y) \, | x \in [i - \alpha, i + \alpha], y \in [j - \alpha, j + \alpha] \} \tag{3.5}$$

where the size of the local patch is controlled by the parameter α. For each



(a) (b) (c)

Figure 3.3: Example of texture contribution to scale-space image: (a) image portion containing flat pavement and textured grass, (b) scale-space image using HSV color space, (c) scale-space image including texture.

patch, the mean is subtracted off to remove the effect of local illumination changes on the texture measure. The forward DCT is then computed: $\hat{B}_{i,j} = \text{DCT}\left(B_{i,j} - \mu\right)$. As the mean is removed, the DC component of the resulting array will be zero. Finally, the texture measure for location (i, j) is the variance of the AC coefficients over the patch.

$$T\left(i,j\right) = E\left[\hat{B}_{i,j}^2\right] - E\left[\hat{B}_{i,j}\right]^2 \tag{3.6}$$

Finally, we use an image tensor to combine the three L*a*b* color channels with the additional texture channel, T. Denote the tensor as $I_t = [I_{L*}, I_{a*}, I_{b*}, T] \in \mathbb{R}^{m \times n \times 4}$, where $m \times n$ is the resolution of the input. The scale is computed using Eq. (3.4), with the Euclidean distance now measured as:

$$\|I_t\| = \sqrt{I_{L*}^2 + I_{a*}^2 + I_{b*}^2 + T^2} \tag{3.7}$$

3.4.3 Tuning Center-Surround Parameters

In the proposed method, scale of objects in the scene contributes to the saliency map. In particular, the scale map is used to weight a set of discriminant saliency maps, each of which has been tuned with different center/surround

parameters. For each image, a set of discriminant saliency maps is computed, denoted as: $\left\{s_{c_0}^{ds}, \ldots, s_{c_M}^{ds}\right\}$. The scale of the center window is indicated by the subscript c_i. That is, the smallest center window is denoted by c_0 and the largest by c_M. For all experiments conducted, the center window sizes are determined (in pixels) as:

$$c_m = 4 \times 2^m, \qquad m = 0, \ldots, M \tag{3.8}$$

The largest scale considered in the proposed method is set to half the smallest dimension of the image:

$$M = \lfloor \log_2 \left(\min\left(I_w, I_h\right)\right) \rfloor - 2 \tag{3.9}$$

In this way, objects can be detected ranging from a few pixels all the way to the majority of the image size. Finally, the scale map is used as a soft-weighting over the tuned discriminant saliency maps. For each pixel (x, y) in the image, the saliency values at (x, y) along the scale dimension are captured in the vector $\mathbf{s}_\sigma$.

$$\mathbf{s}_\sigma\left(x, y\right) = \left[s_{c_0}^{ds}\left(x, y\right), \ldots, s_{c_N}^{ds}\left(x, y\right)\right] \in \mathbb{R}^{1 \times M} \tag{3.10}$$

Next, the soft-weighting vector $\mathbf{w}\left(x, y\right)$ is formed by centering a small Gaussian kernel around the scale value, $\Theta\left(x, y\right)$.

$$\mathbf{w}\left(x, y\right) \sim \mathcal{N}\left(\Theta\left(x, y\right), 0.5\right) \in \mathbb{R}^{1 \times M} \tag{3.11}$$

The weighting vector is scaled and shifted such that it aligns the scale map with the number of saliency maps computed. Finally, the dot-product of the weighting vector and the saliency vector yields the scale-aware saliency value at pixel (x, y):

$$s_{sa}\left(x, y\right) = \mathbf{s}_\sigma\left(x, y\right) \mathbf{w}\left(x, y\right)^T \tag{3.12}$$

The result s_{sa} is the final scale-aware saliency map. One final option which can be enabled to model the center bias of the human fixation database is an element-wise multiplication with a large Gaussian kernel.

$$\hat{s}_{sa} = s_{sa} \otimes h \tag{3.13}$$

3.5 Human Fixation Database

Objective performance for the scale-aware saliency algorithm is assessed with respect to a database of measured human fixations [36]. This database is comprised of 120 natural images including indoor and outdoor scenes. Eye measurements are captured for a a set of 20 subjects using an Eye-gaze Response Interface Computer Aid (ERICA) workstation. Each image is presented in random order for four seconds with a gray mask presented between images. Participants are positioned 0.75 meters from a 21-inch CRT monitor. Multiple fixations are recorded for each observer to produce a *fixation map*. Three images from the database along with fixation maps are demonstrated in Fig. 3.4. The fixation map is filtered with a small Gaussian kernel to replicate foveal visual angle. Before results are discussed, the heavy center-bias of the human fixation database is introduced.

The phenomenon of center-bias in the fixation database is present for two distinct reasons. First, the construction of a natural image by a photographer is generally intended to center salient objects. This is intuitive as a photograph is intended to capture a specific subject. In addition, viewers taking part in the eye-tracking experiment have a physical tendency to fixate on the center of the computer monitor [45]. In the work of [46], no correlation was found between center bias and distribution of image content. Center bias was present in all experiments even when the image content was shifted far from the center of the screen. In addition, center bias was detected irrespective of the viewer's task.

Overall performance results are presented in Table 3.1. The performance metric used for these experiments is area under the ROC curve. For each image in the database, a saliency map is produced by the proposed method and competing methods. This saliency map is normalized to $[0, 1]$ and compared with the measured fixation data by thresholding the saliency map. An ROC curve is generated by this procedure for each image and for each method. Finally, the area under the ROC curve is integrated and this area is averaged over all 120 sequences in the database. This metric has been shown to correlate well with

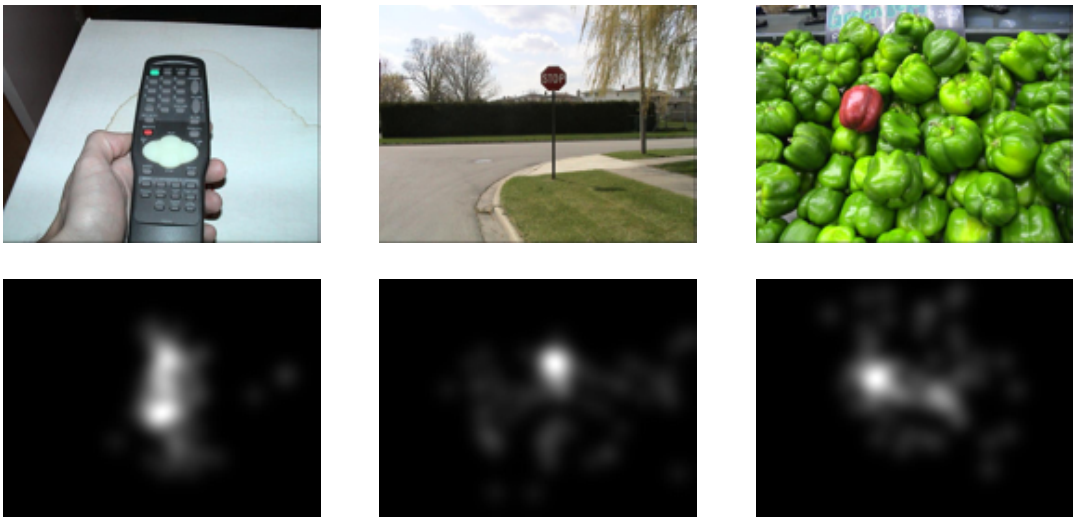
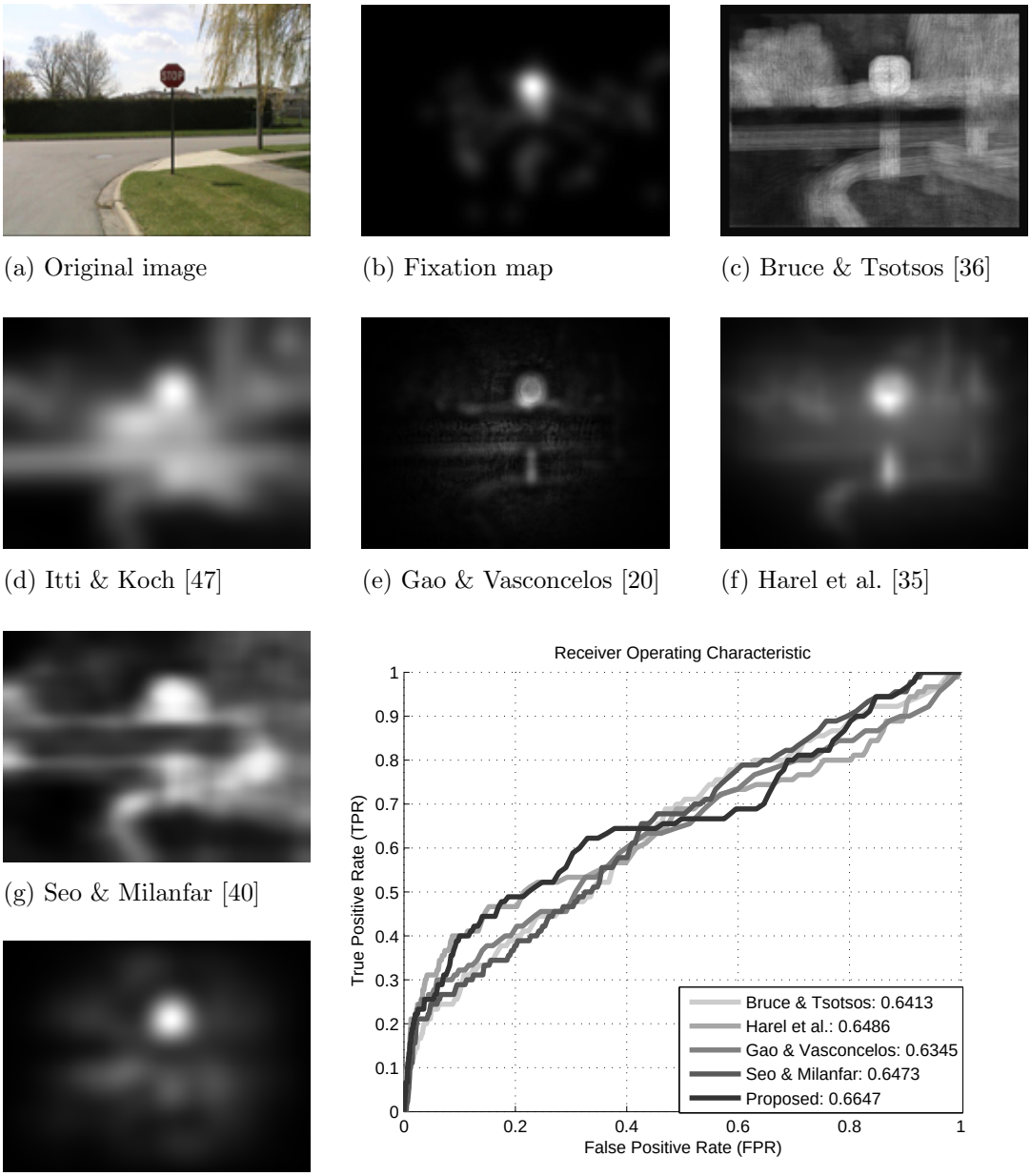


Figure 3.4: Natural images from the human fixation database in the left column with fixation maps in the right column. First row: **remote control**, second row: **stop sign**, third row: **peppers**.

the overall quality of a detector. As is demonstrated by the result, performance of the proposed scheme exceeds all other methods. It should be noted that no tuning has been used for any of the methods. The same parameters used for the proposed method in this experiment will be used in all experiments presented in this work. However, the element-wise multiplication with a Gaussian kernel is only applied to the human fixation database for a fair comparison with previous methods which take center-bias into account.

Results are demonstrated for two sample images from the human fixation database, **stop sign** and **peppers**. The former is an image of a salient stop sign in an outdoor scene. The latter contains a red pepper in a large bin of green peppers. In both cases the ground truth fixation maps confirm the saliency of these objects. Results for **stop sign** image are presented in Fig. 3.5. It is clear from the fixation map that the stop sign is the most salient object in the scene. Each of the saliency detection methods competently detects this object as the most salient in the scene. However, it is also important to suppress all regions other than the stop sign. In the competing methods (Fig. 3.5(c-g)), there is



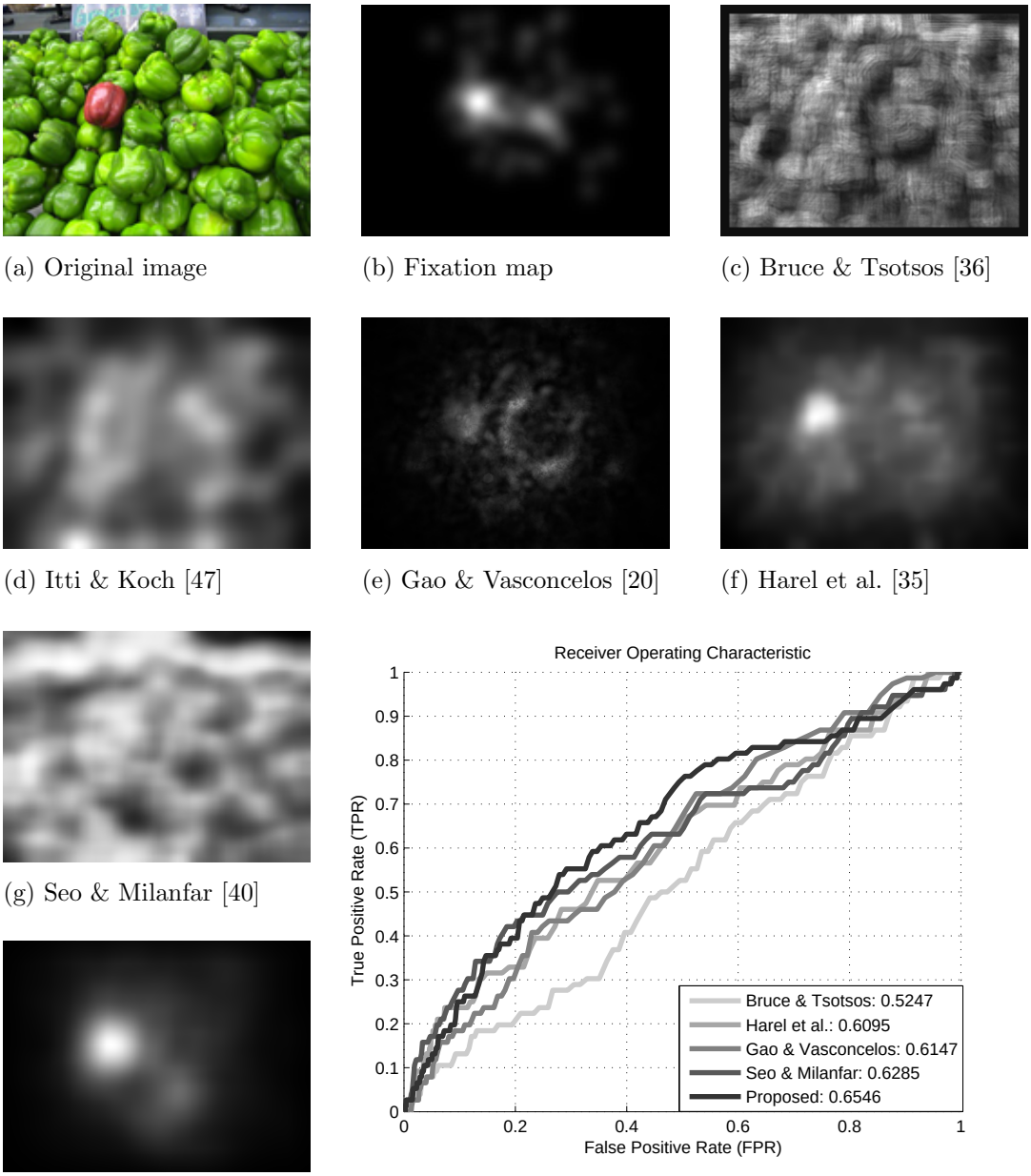
(a) Original image (b) Fixation map (c) Bruce & Tsotsos [36]

(d) Itti & Koch [47] (e) Gao & Vasconcelos [20] (f) Harel et al. [35]

(g) Seo & Milanfar [40]

(h) Proposed (i) ROC

Figure 3.5: Results for the **stop sign** image [36]. While competing methods detect the stop sign as a salient object, it is the suppression of surrounding regions which is important in obtaining good detector performance. ROC curves are shown for this image, with the area under the ROC curve provided in the legend.



(a) Original image (b) Fixation map (c) Bruce & Tsotsos [36]

(d) Itti & Koch [47] (e) Gao & Vasconcelos [20] (f) Harel et al. [35]

(g) Seo & Milanfar [40]

(h) Proposed (i) ROC

Figure 3.6: Results for the **peppers** image [36]. Only GBVS [35] and the proposed method properly detect the red pepper as a salient object. Again, the proposed method outperforms previous approaches due to suppression of the non-salient green peppers. ROC curves for the images are shown with the area under the ROC curve provided in the legend.

Table 3.1: Saliency detection performance compared with measured human fixations [36]. Results given as area under ROC curve, reported along with standard error.

Method	ROC (SE)
Itti & Koch [37]	0.6146 (0.0008)
Gao & Vasconcelos [20]	0.6395 (0.0007)
Harel et al. [35]	0.6444 (0.0007)
Bruce & Tsotsos [36]	0.6727 (0.0008)
Goferman et al. [41]	0.6808 (0.0007)
Seo & Milanfar [40]	0.6896 (0.0007)
Proposed	**0.6934** (**0.0007**)

a high level of false prediction – non-salient regions being classified as salient. However, this is not the case with the proposed algorithm. Here, the background is suppressed while the stop sign is detected as a single, highly salient object. The ROC curves in Fig. 3.5(i) demonstrate that the proposed method performs the best when a low false positive rate is required.

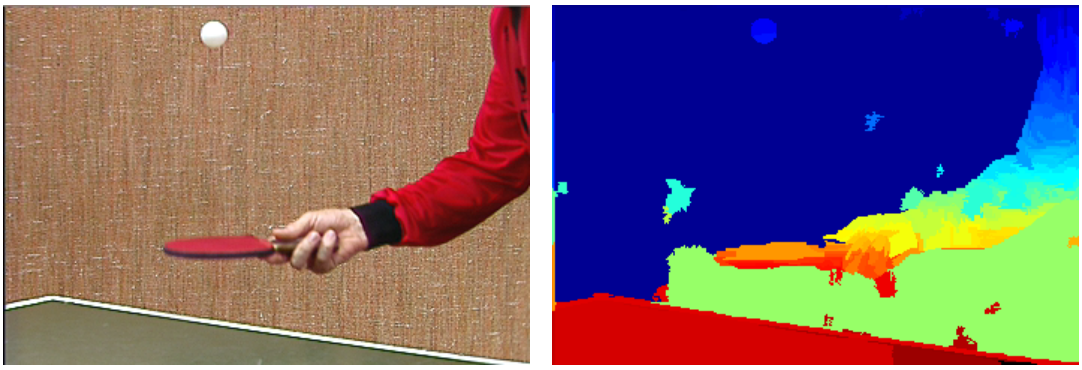
Results for the **peppers** image of the human fixation database are investigated in Fig. 3.6. The most salient object in this scene is the red pepper due to a high contrast in the $R - G$ color channel. The only methods which accurately classify the red pepper as a salient object are GBVS [35] and the proposed method. ROC curves are shown in Fig. 3.6(i) which demonstrate the performance of the proposed method at a wide range of false positive rates.

3.6 Frame Rate Up-Conversion

The proposed scale-aware saliency method has demonstrated improved results with respect to the Human fixation database. In addition, the algorithm is capable of detecting salient objects at multiple scales. This makes it an excellent candidate for any perceptually-based video processing algorithm. Here, Frame Rate Up-Conversion (FRUC), a video enhancement technique, will be investigated when paired with the proposed saliency detector.

FRUC is an area of significant research with applications both to large digital displays and mobile video playback. The massive adoption rate of Liquid Crystal Display (LCD) television sets has necessitated FRUC in order to reduce the visual artifact of motion blur. Unlike earlier Cathode Ray Tube (CRT) displays which are driven by impulse, LCD sets are of the sample-and-hold type. Motion blur caused by the sample-and-hold display is then reduced by decreasing the sample period, thus increasing the frame rate. It is common for LCD sets currently on the market to achieve frame rates of 120-240 Hz. The problem then is that no source material exists at this frame rate. 24 Hz is specified as the FILM standard for movies, and broadcast television is limited to 60 Hz. Therefore, FRUC is essential for these new LCD sets to reduce motion blur and deliver fluid motion to the viewer. At the other end of the spectrum, FRUC is employed by mobile video devices due to restrictive bandwidth constraints for video transmission. It is common for the mobile video encoder to decimate a video signal in time and allocate bits to spatial quality instead. FRUC is then employed at the mobile device in order to re-establish the original framerate of the sequence.

The proposed FRUC algorithm is an extension of the work presented in [48]. Saliency and segmentation are employed to increase the quality of interpolated frames. Consistency of the motion field is enforced for non-salient portions of the scene, while Motion Vector Refinement (MVR) is applied to salient scene regions. In this way, the more expensive MVR algorithm is applied only to regions which are being attended by Human viewers, thus reducing the total computational expense. A departure from the previous work of [48] has been the adoption of mean-shift [49] for frame segmentation in place of normalized-cuts. The EDISON implementation of mean-shift proposed in [50] is used owing a great deal to its computational efficiency. In addition, a robust single-directional interpolation scheme has been adopted in place of the previous bi-directional interpolation. This choice is intended to exploit the improvements allowed by mean-shift segmentation and the scale-aware saliency mask.



(a) (b)

Figure 3.7: Segmentation of frame from **tennis** sequence using mean-shift.

3.6.1 Mean-Shift Segmentation

An efficient mean-shift image segmentation [50] is utilized by the proposed algorithm in place of Normalized-Cuts [23]. This selection was made due to the increased computational efficiency of the mean-shift procedure. To achieve reasonable color segmentation, the L*a*b* color space is employed so that perceived color differences correspond with Euclidean distances in the color space. Essentially, mean-shift segmentation works by defining an affinity between each pixel and all other pixels. Boundaries between segments are then defined as locations where affinity force vectors diverge from one another. The default parameters for the EDISON interface are used for all sequences presented in this work. That is, the spatial bandwidth and range bandwidth are fixed to $h_s = 7$ and $h_r = 6.5$, respectively. A sample output of the mean-shift algorithm is demonstrated in Fig. 3.7. Each region is rendered with a uniform color to demonstrate the segmentation boundaries.

3.6.2 Updated FRUC Algorithm

In order to utilize the improved saliency maps and fast segmentation offered by mean-shift, a new FRUC algorithm has been developed. This algorithm is outlined in Fig. 3.8. Both Motion Estimation (ME) and Motion-Compensated Frame Interpolation (MCFI) are block-based, and rely on a single-direction

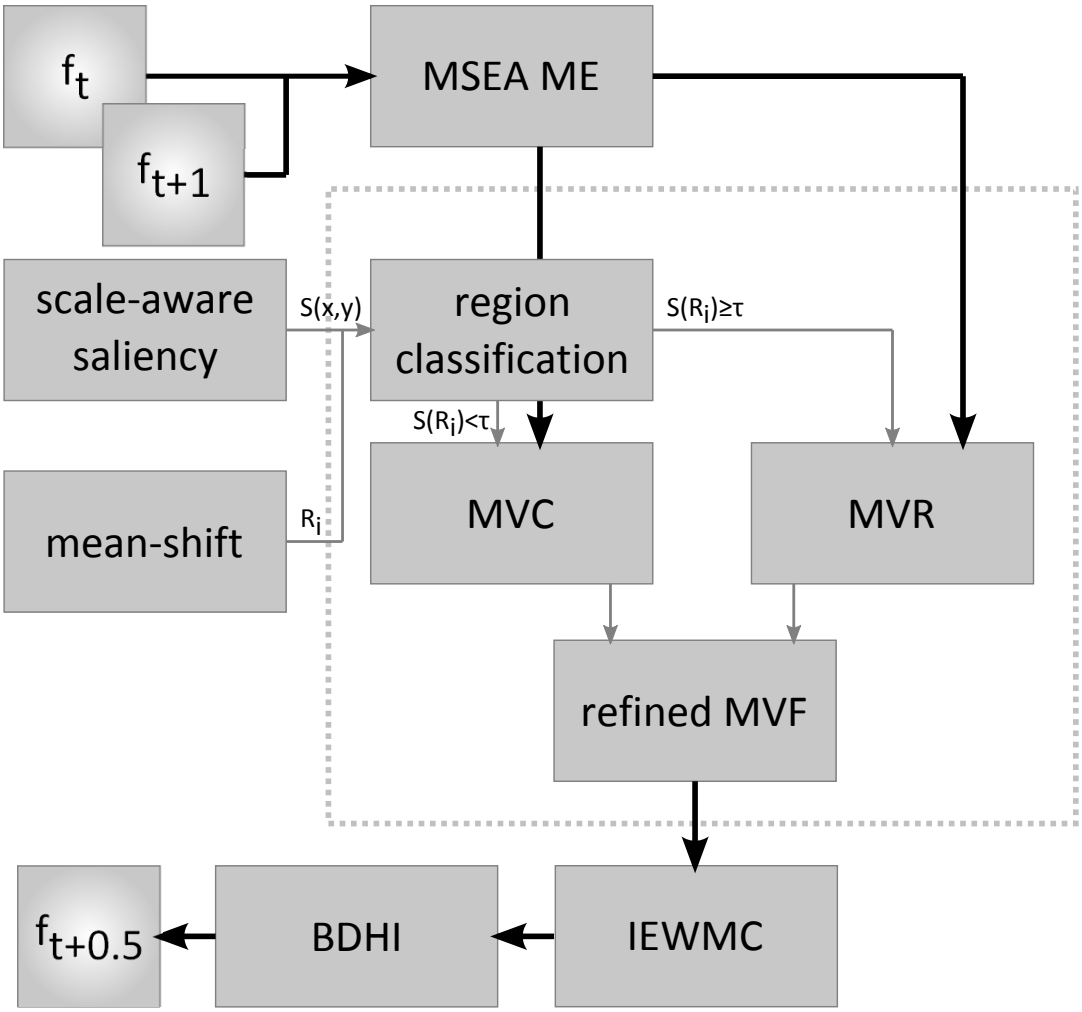


Figure 3.8: Updated framework for the saliency-based FRUC algorithm. The proposed saliency-based motion field improvement is demonstrated inside the dashed box. ME is performed prior, and MCFI is performed subsequently.

search. A bi-directional search has been considered previously, however this ultimately limits the performance as neither segment nor saliency information exist for the interpolated frames. In the proposed work, the ME algorithm due to [51] is used as the input motion field. This work is based on the Multi-level Successive Elimination Algorithm (MSEA) [30] for fast single-pass ME.

The input to the proposed FRUC algorithm is a block-based Motion Vector Field (MVF), where the block size is fixed to 8×8 pixels. If a block in the current frame is denoted as $f_t(\mathbf{x}, \mathbf{y})$, and its location in the reference frame is $f_{t+1}(\mathbf{x}+i, \mathbf{y}+j)$, then the translation (i, j) is the MV assigned to the current block. Next, the saliency map $S(x, y)$ for the current frame is computed using the Scale-Aware Saliency method introduced in this work. A frame segmentation is performed using the mean-shift procedure. The resultant regions are denoted as $R_1, R_2, \ldots, R_n$ where $R_1 \cup \ldots \cup R_n = f_t$. Finally, the input MVF is upsampled using nearest-neighbor interpolation to the same resolution as the input frame, f_t.

With all pre-processing steps completed, the MVF is then improved using saliency and scale information. An indicator vector is used to keep track of salient regions:

$$R_s(k) = \begin{cases} 1 & \sum_{(i,j) \in R_k} S(i,j) \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{3.14}$$

where the threshold τ is fixed to 10% for all experiments conducted in this work. Motion Vector Consistency (MVC) is performed as described in [48] for regions R_i where $R_s(i) = 0$. Similarly, Motion Vector Refinement (MVR) is performed when $R_s(i) = 1$. The resolution of the refined MVF is then reduced so that a block-based MCFI algorithm can be used to create the interpolated frame. The final MV for each 4×4 block is decided by majority vote. This procedure has produced superior results to operating entirely within the block-basis, since object boundaries may be refined for salient objects.

Finally, interpolated frames are generated using Motion-Compensated Frame Interpolation (MCFI). One disadvantage that single-directional inter-

Table 3.2: Objective Performance for FRUC methods using PSNR (first row) and SSIM (second row) metrics. Results presented are averaged over all frames of each sequence.

sequence	FS [29]	MSEA [51]	MMVP [31]	DyTex [48]	SAS
football	25.6782	25.6062	24.4704	25.668	**26.1026**
	0.7592	0.7594	0.6866	0.7606	**0.7834**
foreman	38.622	38.6219	37.0895	37.8687	**39.4502**
	0.9515	0.9519	0.9444	0.9442	**0.9595**
ice	34.8873	34.7735	33.5193	37.3869	**37.4275**
	0.9652	0.9651	0.9518	0.9768	**0.9778**
tennis	31.6748	31.6194	28.966	31.8824	**32.0893**
	0.8600	0.8622	0.7392	**0.8791**	0.8734

polation has is the presence of overlap and holes in the interpolated picture. To remedy this situation, two techniques have been adopted from the recent MCFI work due to [52]. These are Irregular-Grid Expanded-Block Weighted Motion-Compensation (IEWMC) to reduce the presence of holes and Block-Wise Directional Hole Interpolation (BDHI) to fill in all that remain.

3.6.3 Simulation Results

The proposed algorithm is compared with previous FRUC implementations using the following procedure. First, a sequence of rate 30 Hz is decimated in time by a factor of two. Motion Estimation is computed between each pair of frames in the decimated sequence. Next, one interpolated frame is created per source frame using MCFI. Bidirectional interpolation is employed for all non saliency-based methods. For the method of [48] and the proposed work, interpolation is performed in one direction. This is necessary as saliency maps can only be generated for source frames. Finally, the interpolated frame is compared with the original using the Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM). Results are investigated for four sequences common to the video processing literature. These are: **football**, **foreman**, **ice** and **ten-**

nis. Results are investigated using the procedure above for Full Search [29], Multi-level Successive Elimination Algorithm (MSEA) [51], Multi-stage Motion Vector Processing (MMVP) [31], as well as the previous saliency-based method described in [48]. Simulation results using the PSNR and SSIM metrics are presented in Table 3.2. PSNR is calculated as the average mean-squared error (MSE) of the predicted frame and presented in decibels. SSIM is a complementary error in which perceptual errors are modeled [32]. For all sequences, a significant improvement in PSNR is observed due to the increased accuracy of the saliency maps. A particularly obvious example of this is observed in Fig. 3.10. A significant error is present in the ping-pong ball for previous methods in Figs. 3.10(b-d). This is due to the relatively small size of the ball and the static motion surrounding it. The previous saliency-based method in Fig. 3.10(e) is unable to correct this error because the ball is not detected as salient in this frame, owing to the previously-discussed scale problem. In the proposed multi-scale saliency map, the ball is detected as a salient object and refinement is conducted. This refinement may also be observed in the hand of the table tennis player. For FRUC, the improper interpolation of the ping-pong ball results in a flickering artifact which is easily noticeable to the viewer. Video results are available on the author's website [1]. In addition to PSNR, performance is significantly improved for SSIM over all sequences other than **tennis**. Here, the previous saliency-based method performs slightly better due to some loss of structure in the background achieved by correcting foreground errors. Perceptually, the proposed method using scale-aware saliency is still much more pleasing. In addition to **tennis**, results are presented for the **football** sequence in Fig. **??**. This graph is intended to demonstrate the improvement of the proposed algorithm over previous methods. Results are presented as the difference between the proposed method and previous methods for each frame. An improvement is realized by any difference greater than zero for each frame. By inspection, nearly all frames indicate an improvement both in PSNR and SSIM.

[1] Video results and additional data available at `http://videoprocessing.ucsd.edu/~NatanHaim/TIP_2011a/`

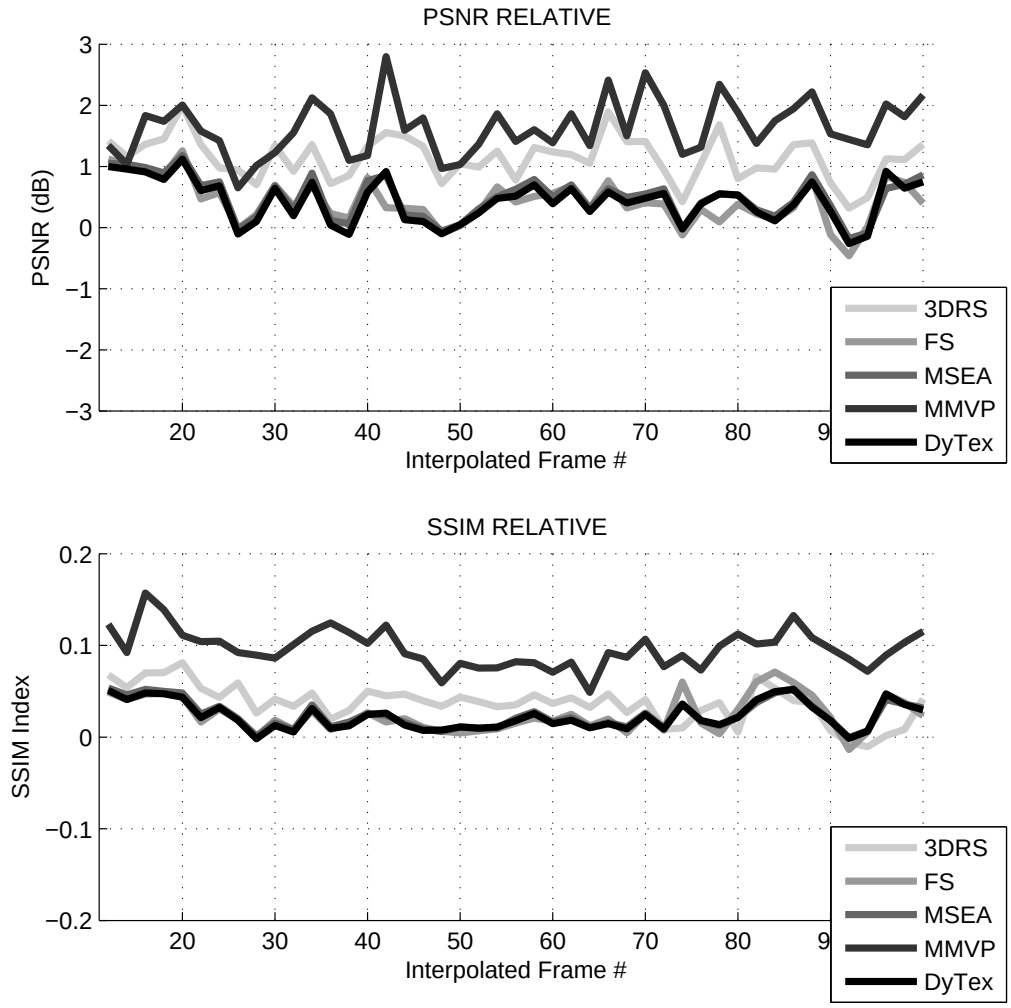


Figure 3.9: Relative performance of the proposed Scale-Aware Saliency based FRUC algorithm compared with previous methods. Improved performance is indicated by a relative score greater than zero for each frame.

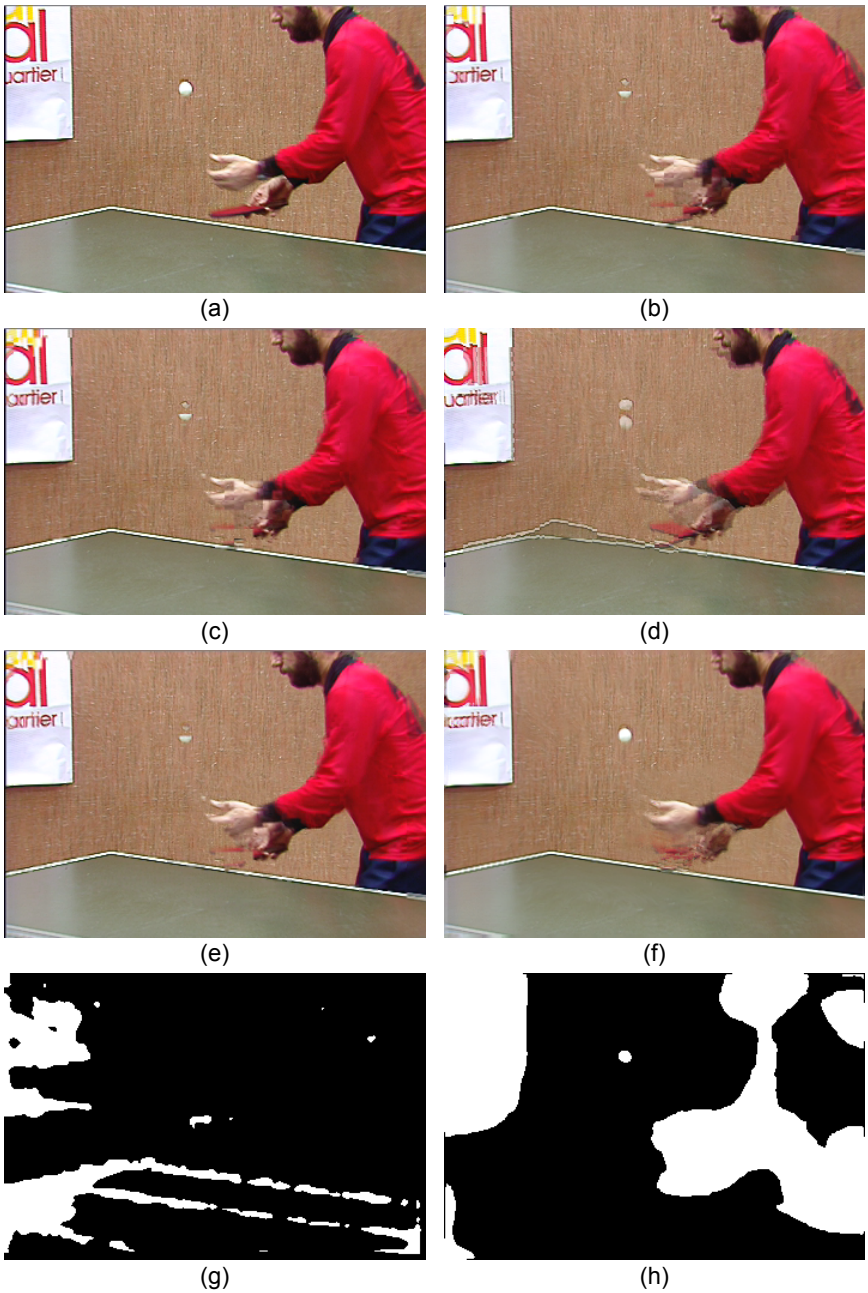


Figure 3.10: Comparison of interpolated frame 50 of the **tennis** sequence: (a) original frame, (b) Full Search, (c) MSEA, (d) MMVP, (e) Method of [48] using motion-based Discriminant Saliency, (f) Proposed method. Saliency maps are included: (g) Discriminant Saliency map using dynamic textures [28] used in [48], (h) proposed scale-aware saliency map

The proposed method is currently implemented in MATLAB on an Intel core-i7 CPU clocked at 3.2 GHz. Processing time is roughly 30 seconds per frame for initial motion estimation as well as all steps of the proposed algorithm, excluding saliency map computation. Runtime for scale-aware saliency computation is roughly 60 seconds per frame. In order to streamline this process and achieve real-time implementation, the saliency map may be computed on a subsampled grid. For mobile application, saliency information can be computed at the encoder side, and transmitted as side-information. In addition, the saliency map could be used to improve compression as well as provide a guide for packet prioritization.

3.7 Conclusion

In this work, we have demonstrated a new biologically-plausible method for saliency detection using scale information. The proposed algorithm has been investigated both for images and video. Excellent results were obtained with the proposed algorithm applied to the task of determining salient points in the Human fixation database. Using the standard metric of area under the ROC curve, the proposed method performed in excess of all competing algorithms. To ascertain performance for video processing applications, the scale-aware saliency algorithm was confronted with the task of Frame Rate Up-Conversion. The ability to detect salient objects at multiple scales allowed for a performance increase over previous methods for FRUC. It is determined from these results that the proposed algorithm for saliency detection is well-suited for general perceptually-based video processing tasks.

The text of Chapter 3 is adapted from *Scale-Aware Saliency for Application to Frame Rate Up-Conversion*, Natan Jacobson, Truong Nguyen, accepted to *IEEE Transactions on Image Processing* in August of 2011. The dissertation author is the primary author of this publication.

Chapter 4

Effect of stereoscopic depth on saliency: evidence from eye movements

"There ought to be something very special about the boundary conditions of the universe and what can be more special than that there is no boundary?" - Stephen Hawking

We proceed to investigate saliency with respect to the stereoscopic depth feature. It is well-known how humans segment and analyze objects in the visible world by means of salient features, such as contrast, color and brightness, but how does depth (e.g., retinal disparity) contribute to saliency? In this work, we analyze the effect of stereopsis on human visual saliency. In particular, we investigate how pre-attentive fixations change with the presence/absence of disparity depth cues. A mirror stereoscope was used to present stimuli with or without a stereoscopic manipulation to subjects. Fixations and saccades were recorded for each subject via an EyeLink II eye-tracking system. Contrast, gradient, and center-surround difference were analyzed for the intensity and disparity features both for the locations of the measured fixations, as well as randomly-sampled positions. Eye movements were recorded while subjects viewed two datasets: one which includes synthetic objects and ground-truth disparity was calculated

for the images while the other includes only natural scenes and the disparity map was less precisely estimated. Results differ between the two datasets. For natural scenes, subjects tend to fixate locations with lower disparity than at random locations. However, for the synthetic database, subjects attend to regions of higher disparity than at randomly-sampled locations. The results are discussed in detail, and all eye-tracking data is made publicly available on our website.

4.1 Introduction

Over the last quarter-century, visual saliency has become an integral part of the image and video processing world. In addition, it has helped us understand how people see the world. Considerable attention has been paid to understanding what features of an image affect human attention. Features which are known to play a major role include: brightness, color, orientation, motion and size [53]. However, the influence of disparity on saliency is less well understood. While significant research has taken place to understand how depth is processed by the human visual system [54–59], the effect on pre-attentive vision remains a mystery. In this work, we aim to further understand the connection between stereoscopic depth and saliency through a set of eye-tracking experiments.

Saliency in the human visual system can be partitioned into two separate classes. The first class is referred to as *pre-attentive* or *bottom-up* saliency [60]. This is the stimulus-driven response of the human visual system to attend to regions which are rich in information content. The visual world contains an excess of information to be able to process everything at once, so this stage is important for quickly understanding the important visual events around us. The second class is referred to as *top-down* saliency, and is goal-driven. Tasks such as recognition, tracking and locating human faces are all encompassed in the top-down stage. In this work, the primary concern is the effect of stereoscopic depth on pre-attentive vision. This allows us to understand how changes in the depth content of a scene will attract early human fixations.

Understanding of the effect of depth on saliency is of considerable interest to the image and video processing fields. Saliency-based video processing techniques already abound for the monocular case. A few examples are noted here. A saliency-based compression algorithm has been proposed [8] which uses a variable blur kernel to smooth out each video frame which is sufficiently far from a salient point. The result is a much higher compression ratio since the spatial prediction has lower residual in highly blurred regions. Saliency has also

been employed for the task of background subtraction [28]. Pixels which fall below a selected saliency threshold are classified as *background* pixels and removed from the scene. Background subtraction is directly applicable to compression and to compositing. Finally, saliency has been applied to Frame Rate Up-Conversion [48]. Here, motion vector refinement is applied selectively to salient scene regions, resulting in a superior motion field prior to Motion-Compensated Frame Interpolation.

In the excellent review of features affecting attention [53], stereoscopic depth is considered as a *probable guiding attribute* for visual attention. It is discussed that a broader feature, such as three-dimensional spatial layout, may include stereopsis but may not itself be a guiding principal of attention. The classification of probable guiding attention implies that more data may help to clear up ambiguities. The later work of [61] explores the relationship between luminance and depth for natural images. A high level of correlation between the two features is found, and a statistical model is proposed for depth statistics. Further analysis of the depth feature was explored in [62], in which mean disparity and disparity contrast were investigated with respect to human attention. It was determined that humans fixate on objects closer to them, while disparity contrast has a small effect on noise images. In addition, subjects fixated on frontal planar regions more frequently when presented with 2D stimuli then when depth information was present. In [63], eye-tracking experiments were performed to further understand the relationship between depth and human fixations. It was determined that regions with a high contrast or gradient in depth tend to repel human fixations. This claim is counter-intuitive, but is supported by the low processing bandwidth of early human vision.

This section is organized as follows. First, previous work in the field of saliency is discussed, including monocular features as well as the contribution of stereo. Following that, our experimental setup is described. Next, our two experiments are introduced. Finally, we conclude by discussing the results obtained through our experiments.

4.2 Previous Work

The study of saliency with respect to monocular features has received an incredible amount of attention over the past 25 years. While pre-attentive vision has been studied by psychologists for decades, the implementation in software is somewhat more recent. A framework for generalized saliency detection was first proposed in [60] in which a winner-take-all (WTA) network is used among a set of features to produce a *conspicuity map*. Later, a center-surround saliency model was proposed in [37]. Here, multi-scale center-surround differences are computed for the intensity, color, and gabor-filter orientation channels of an image to produce a conspicuity map. A saliency map is generated by combining the conspicuity maps using a uniform weighting over the features. Later approaches have considered saliency as an information maximizing process aimed at rapid scene understanding. In [36], Shannon's self-information measure is utilized to estimate the probability of a human fixation given a local patch. Later, a discriminant saliency detector was proposed [20], which defines saliency as the ability of some feature response to discriminate between center and surround regions. The higher the dissimilarity between the two regions, the more salient the location must be, In particular, the discriminant saliency approach has been shown to replicate the human visual response to changes in orientation and contrast.

In addition to purely spatial features, temporal aspects have been investigated as well. A dynamic texture-based approach to discriminant saliency was considered [64] which has proven to be promising for highly dynamic scenes. Another approach to modeling saliency for video was proposed by [40], using self-resemblance of a pixel and its immediate neighborhood to measure saliency. The temporal aspect is modeled using 3D local steering kernels (two spatial dimensions and one time dimension).

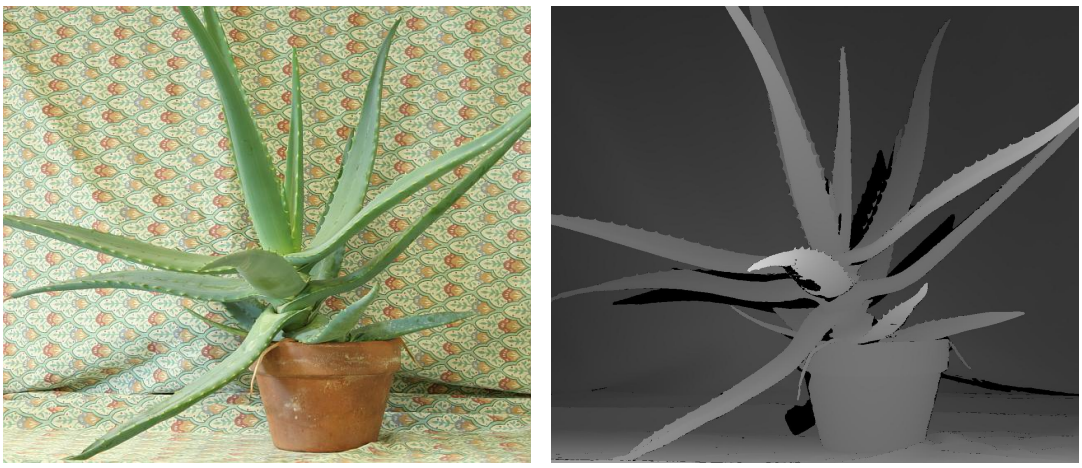
The computation of saliency with respect to the depth feature has received attention, but little consensus. In an early psychological experiment based on the research of [65], feature conjunctions are investigated for stereo.

A feature conjunction occurs when more than one salient feature is changing in a scene. An example for the monocular would be the conjunction of color and orientation, both salient features on their own. It was discovered that the human visual system can perform a search along a single stimulus dimension in parallel, while this is reduced to a serial search for conjunctions. In the work of [66], the conjunction of stereo with other features was investigated; it was determined that the visual system can perform the search across this conjunction in parallel. This conflicts with the earlier findings, and is an indicator that stereoscopic depth is not in itself a salient feature.

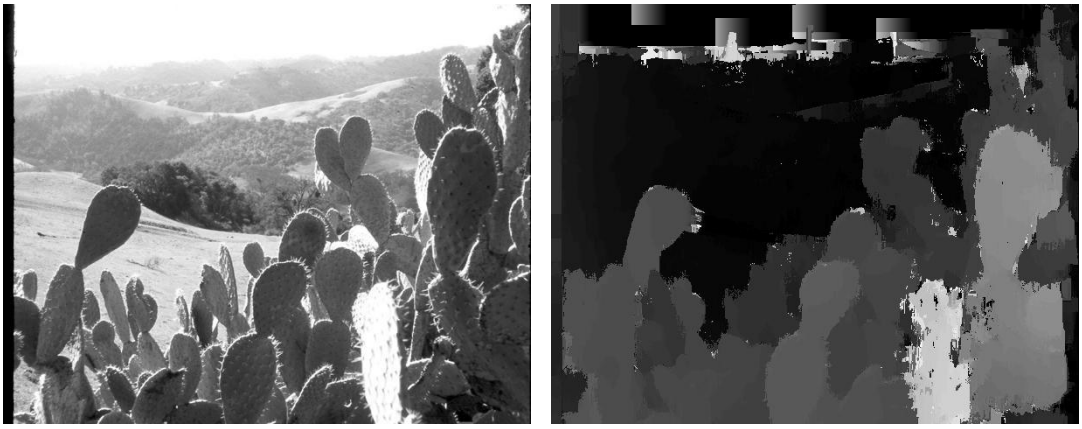
In recent literature, saliency algorithms have been proposed with include the stereoscopic depth feature. The algorithm proposed by [67] extends the earlier work of [37] by considering depth information from a laser range-scanner. Three depth features are considered: absolute depth, mean curvature, and depth gradient. A conspicuity map is generated for each feature using a center-surround difference, and the conspicuity maps are combined with equal weight to generate the saliency map. A simplified model was proposed in [68] wherein monocular saliency maps for the left and right views of a stereo pair are combined, and exponentially weighted based on proximity to the camera. It was determined in [69] that points closer to the observer (larger disparity) tend to be fixated upon more readily than randomly selected points, providing motivation for the exponentially weighted model. This experiment was validated using an autostereoscopic monitor and random dot stereograms. A separate approach [70] produces a "depth-saliency map" from the regions of the scene with highest depth contrast, where depth estimation is performed by computing the pixel-wise difference between the left and right views.

4.3 Experimental Setup

Our experiments were conducted using a single-reflection mirror stereoscope and an SR-Research EyeLink II eye tracking system. The mirror stereoscope allowed us to display separate visual information to each of the subject's



(a) (b)



(c) (d)

Figure 4.1: Two scenes selected from the testing set. **Middlebury** scene is shown in the top row and **yosemite** scene in the bottom row. The left column shows the stimulus while the right column is the associated disparity map.

eyes. For 2D scenes, the same image was displayed to each eye, whereas separate views were displayed for 3D scenes. Testing was performed for a set of 20 subjects, ranging in age and gender. A stereo fly test was used to validate normal stereoscopic vision for each subject. All 20 subjects were capable of stereopsis at a minimum of 100 arc-seconds at the recommended viewing distance of 16 inches.

4.3.1 Hardware

The mirror stereoscope was constructed from off-the-shelf components. A Newport optical alignment table was used as the base of the system, with adjustable mounting posts supporting two 3" square mirrors. A chin rest was placed in front of the table centered between the two mirrors. Two Dell 17" LCD monitors were placed on either side of the subject, facing inwards so that the primary reflection was visible to the subject. The position of the monitors was fixed for all subjects. Monitor resolution was fixed at 1024×768, corresponding to 40 pixels per degree of visual angle. The position of the chin-rest and angle of the mirrors relative to the subject were adjustable to a small degree to accommodate different subjects. The head-mounted EyeLink II eye tracking system was used to capture fixation data. Four infrared LEDs were placed in a rectangular pattern in front of the subject to assist in capturing head pose for the EyeLink system. Fixation data was captured at 500Hz using pupil tracking and a 9-point calibration grid.

4.3.2 Software

The experiment was performed using a set of 50 images, each of which was displayed to the subject either in 2D or 3D. The images were selected from two datasets: the **middlebury** stereo set [71] and the **yosemite** dataset [72]. Example images are displayed in Fig. 4.1. The two dataset have very different properties. The **middlebury** dataset consists of indoor scenes containing man-made objects. In addition to the stereo image pairs, depth information was

(a) (b)

Figure 4.2: Fixations from (a) **middlebury** and (b) **yosemite** aggregated over 20 subjects. 2D fixations are displayed as yellow dots while 3D fixations are displayed as magenta dots. All fixation data is available on our website.

captured using a laser range-finder. This provides ground truth disparity for all scenes within the **middlebury** set. The **yosemite** dataset contains stereo image pairs for outdoor scenes. No man-made objects are visible in this set. Image data is available in luminance only, and the disparity is estimated using a biologically-plausible cross-correlation measure [73].

The experiment was written in MATLAB using the Psych Toolbox [74–76] and EyeLink Toolbox [77]. The goal of the proposed experiment was that of a memory test. Each subject was instructed to study a set of scenes for the purpose of recognizing them upon recall. The eye-tracker was then placed on the subject such that fixations were recorded for the right eye. The subject then performed a 9-point calibration on the EyeLink followed by a validation. Following this, the main test started. The subject was shown a set of stimuli either in 2D or 3D. The order of images was randomly permuted for each subject. For each image, half of the subjects received the 2D version while the other half received the 3D version. Each image was displayed for 3 seconds, followed by a random noise pattern for one second to suppress after-images. After each five images, a drift-correction was performed, with the option of recalibrating

the system if the drift was too large. After the 50 images were completed, the EyeLink was removed from the subject prior to the memory test. The memory test consisted of 10 images, five of which were sampled from the testing set, and the other five were novel images. For this portion, the subject used a keyboard to provide feedback. No time constraints were placed on the subject for completing this portion of the test.

4.4 Experiment 1

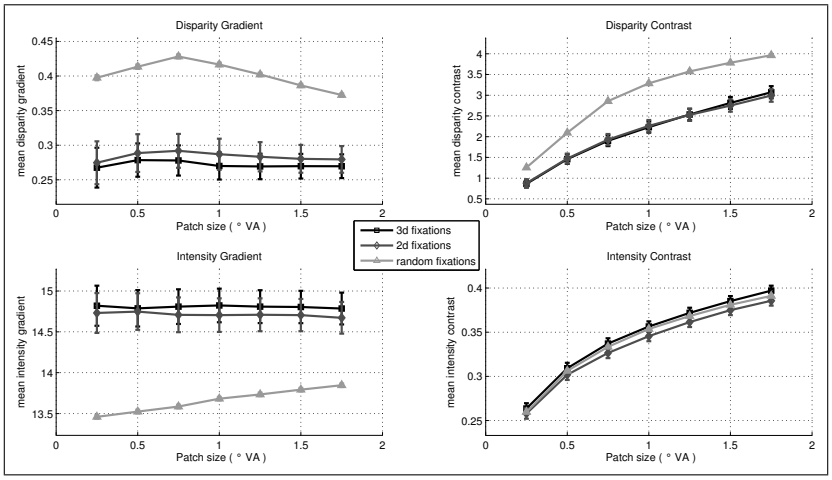


Figure 4.3: **Yosemite**: distribution of disparity gradient, disparity contrast, intensity gradient and intensity contrast. Feature distributions are averaged over a variable patch size, reported with respect to degrees of visual angle. The 95% confidence interval on the mean is plotted for each sample. A patch size of 1°VA corresponds to a 40×40 pixel region. We show a lower disparity gradient and disparity contrast at human fixations when compared with randomly sampled locations. Intensity gradient is significantly higher at fixated locations than at random locations. For intensity contrast, no clear distinction can be made.

The first experiment is based on the research of [63]. Intensity and disparity feature content are compared for human fixated regions as well as

randomly-sampled positions. Our contribution to this research is a larger set of subjects, as well as a distinction between 2D and 3D fixations. The four features investigated are: intensity gradient (G_I), intensity contrast (C_I), disparity gradient (G_D) and disparity contrast (C_D); given in Eqs. 4.1-4.3.

$$G_I = \sqrt{\frac{\partial I^2}{\partial x} + \frac{\partial I^2}{\partial y}}, \quad G_D = \sqrt{\frac{\partial D^2}{\partial x} + \frac{\partial D^2}{\partial y}} \tag{4.1}$$

$$C_I = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{I_i - \bar{I}}{\bar{I}}\right)^2} \tag{4.2}$$

$$C_D = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(D_i - \bar{D}\right)^2} \tag{4.3}$$

where $\bar{I}$ and $\bar{D}$ are the means of local intensity and disparity patches. These features are averaged over a variable patch size ranging from $0.25^\circ - 1.75^\circ$ visual angle. In addition to the direct comparison, ratio tests are performed for each feature measure. The ratio tests help to remove the effect of positive correlation between intensity and disparity in the datasets. Results for the **yosemite** dataset are given in Fig. 4.3. The 95% confidence interval for each data point is computed on the sample mean. Given a sample, there is 95% probability that the mean of the feature measure will be within the confidence interval. The random positions are determined by uniformly sampling over each image in the dataset. In order to decrease variance on the random positions, a large number of positions are sampled from each image. If for image F_i we have collected n_i fixations over the entire subject pool, then we will select $100 \times n_i$ random positions for this image. A considerable difference in the distribution of disparity features is observed in the **yosemite** dataset. We observe that the disparity gradient is considerably lower at human fixations than it is for random positions. The same is true for disparity contrast. For both of these features, the random and human data is easily separable, as there is no overlap in the confidence intervals. The disparity gradient is slightly higher when no depth information is available to the subject, however this change is not statistically

significant. Disparity contrast is nearly identical for 2D and 3D human fixations. The intensity gradient is significantly higher for human fixations than it is for random positions. Again there is little distinction between 2D and 3D human fixations. Finally, there is no statistically significant difference in the intensity contrast feature. Here, there is total overlap between all human fixations and random positions. Overall, these findings validate the earlier results obtained in [63] while considering a larger pool of subjects. Small trends are observed which differentiate the 2D and 3D human fixations for the intensity and disparity gradient feature, although these are not statistically significant. This is a case for monocular depth cues guiding visual attention, while the addition of true depth has little additional effect.

Results for the **middlebury** dataset are displayed in Fig. 4.4. This dataset is comprised of synthetic scenes, and therefore may elicit more of a top-down response from subjects as they identify recognizable objects of interest (text, faces, etc.). We observe a disparity gradient which is larger for human fixations than it is for randomly-selected positions. This is a surprising result, as the opposite trend was identified for **yosemite**. Here, a larger discrepancy between 2D and 3D fixations is exhibited, however the confidence intervals are also somewhat greater than before. As with gradient, the disparity contrast is higher for human fixations than random sampling. For a small patch size, the distribution at 3D fixations exceeds that at 2D fixations, however this trend is reduced as the patch size increases. Results for intensity features also differ significantly between the two datasets. Intensity contrast is higher for fixations, while intensity gradient is indistinguishable between the three trials. Intensity gradient is larger at 2D fixations than it is for 3D fixations.

4.4.1 Ratio Test

Further analysis is performed by comparing the feature discrepancies between fixated positions and random locations separately for each image in the dataset. This is accomplished by performing a ratio test. For each dataset,

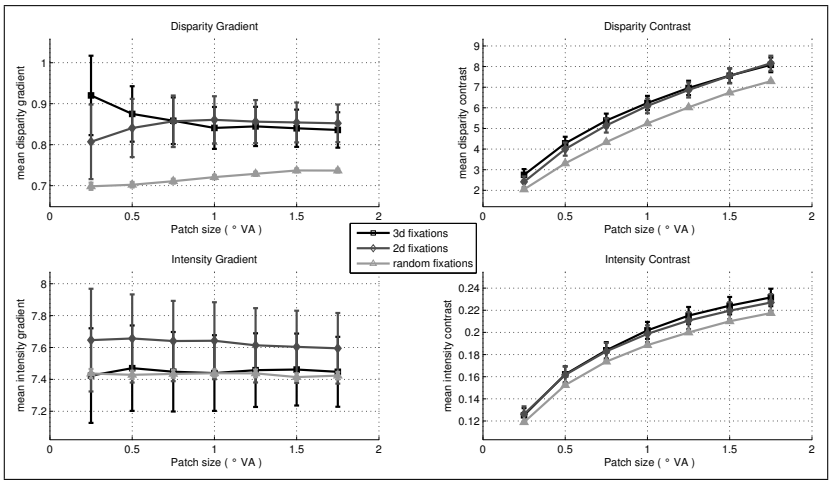


Figure 4.4: **Middlebury**: distribution of disparity gradient, disparity contrast, intensity gradient and intensity contrast. Feature distributions are averaged over a variable patch size, reported with respect to degrees of visual angle. The 95% confidence interval on the mean is plotted for each sample. A patch size of 1°VA corresponds to a 40×40 pixel region. As for the **yosemite** set, the feature content between 2D and 3D fixations cannot be clearly separated, as the confidence intervals are overlapping. The disparity gradient is higher for human fixations than for randomly selected locations. The same is true for disparity contrast and intensity contrast. Surprisingly, the intensity gradient at random positions is inseparable from the human fixations.

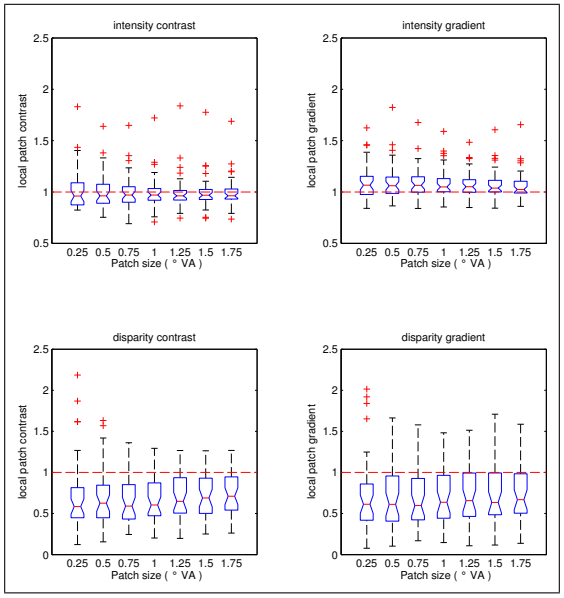


Figure 4.5: **Yosemite**: ratio between 2D human fixations and randomly-selected positions. A ratio above 1 indicates that fixated patches have a higher feature contribution than random.

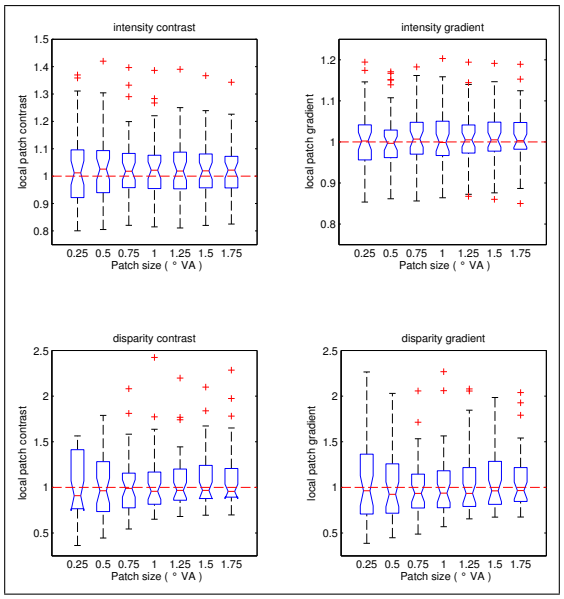


Figure 4.6: **Yosemite**: ratio between 3D and 2D human fixations. A ratio above 1 indicates that the fixated patches when depth information is available have a higher feature contribution than when depth is unavailable.

the ratio between 2D and random positions will be explored, as well as the ratio between 2D and 3D fixations. Intensity gradient/contrast and disparity gradient/contrast are investigated. For intensity contrast, the ratio is computed as follows. For image i, the mean 2D intensity contrast is given by C^i_{2d}:

$$C^i_{2d} = \frac{\sum_{j=1}^{N_{2d}} C^i_I(x_j, y_j)}{N_{2d}} \tag{4.4}$$

where C^i_I is the intensity contrast map, N_{2d} is the number of 2D fixations recorded for image i over the set of 20 subjects. (x_j, y_j) is the position of fixation j. The mean 3D intensity contrast is computed in the same fashion. Next, the mean intensity contrast is computed for random positions:

$$C^i_{ran} = \frac{\sum_{j=1}^{N_{ran}} C^i_I(\tilde{x}, \tilde{y})}{N_{ran}} \tag{4.5}$$

where $N_{ran} = 100 \times N_{2d}$ and $(\tilde{x}, \tilde{y})$ is a uniformly distributed random position within the extent of the image. The mean feature content per image is computed for the other features via the same procedure.

In this work, we explore the difference between human fixations and random positions. For this, the 2D fixations are investigated. For the intensity contrast feature, this ratio is RC^{2d}_I.

$$RC^{2d}_I = \frac{C^i_{2d}}{C^i_{ran}} \tag{4.6}$$

While this ratio has been investigated before, we are able to consider a significantly larger sample size. In addition, we investigate the difference between feature content at 2D and 3D fixations. This ratio is given by RC^{3d}_I.

$$RC^{3d}_I = \frac{C^i_{3d}}{C^i_{2d}} \tag{4.7}$$

For the **yosemite** dataset, results are presented in Figs. 4.5-4.6 using a boxplot. For each patch size, the series of images in the dataset is plotted. The center of the box (red line) is the median of the distribution, while the top and bottom of the box are the 25th and 75th percentiles. Outliers are marked separately. In the case of Fig. 4.5, we see that the ratio for intensity gradient is

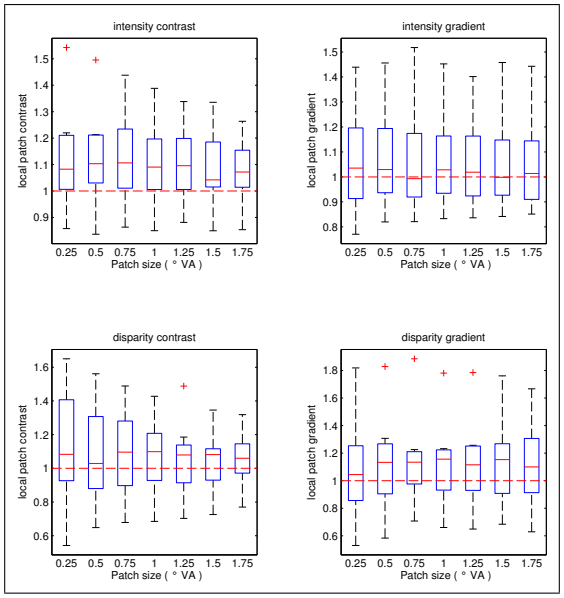


Figure 4.7: **Middlebury**: ratio tests between 2D human fixations and random positions. For each patch size (given in °VA) the distribution of mean feature content is displayed using a box plot. The red line shows the median, while the box displays the 25th and 75th percentiles of the distribution.

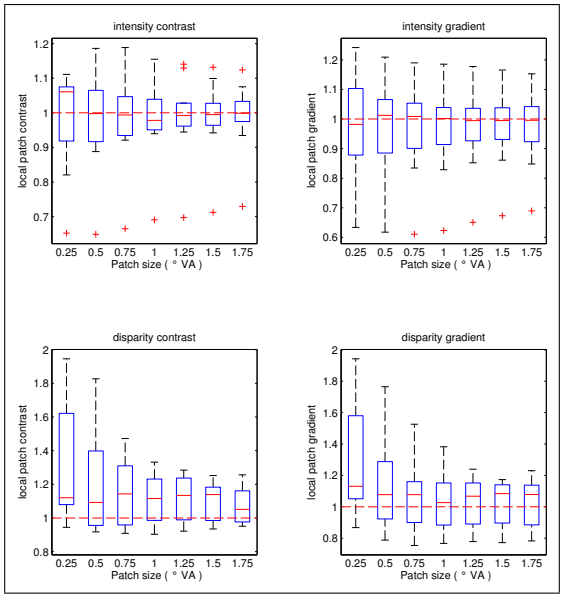


Figure 4.8: **Middlebury**: ratio tests between 3D and 2D human fixations. For each patch size (given in °VA) the distribution of mean feature content is displayed using a box plot. The red line shows the median, while the box displays the 25th and 75th percentiles of the distribution.

greater than 1, while both disparity features are less than 1. This is in concert with the previously obtained results. However, RC_I^{2d} is very close to 1 for all patch sizes. This indicates that local patch intensity is not a good indicator of where the subject will fixate. This result is somewhat surprising, although it is known that attention should be more related to bandpass features than contrast alone [78]. Center-surround differences for intensity will be investigated in experiment 2. In Fig. 4.6, we investigate the difference between features given 2D and 3D fixations. As it turns out, very little discrepancy occurs for any of the four features. This could indicate that salient information in the disparity feature is captured in the 2D case by monocular cues. In this case, the correlation between intensity and disparity is responsible for the similarity in the two trials.

Feature ratio tests were also performed for the **middlebury** dataset in Figs. 4.7-4.8. For the comparison with randomly-sampled positions, the trend is significantly different than it was for the **yosemite** dataset. In this case, $RC_D^{2d} > 1$ and $RG_D^{2d} > 1$ for all patch sizes. In addition, $RG_I^{2d} > 1$ and $RC_I^{2d} \approx 1$. The difference regarding the disparity feature can be due to one of many factors. The content of the two datasets is dramatically different, with recognizable objects in **middlebury** while **yosemite** is comprised solely of *natural* content. Another possibility is that inconsistencies in the disparity estimation for **yosemite** are responsible for lost depth information.

In Fig. 4.8 we investigate the ratio between 3D and 2D human fixations. The median ratio for both intensity features is very close to one, whereas the median ratio for the disparity features is greater than one. This result indicates that, for the **middlebury** dataset, changes in disparity act to attract human attention. In fact, changes in disparity appear to be highly salient, as 3D fixations have a ratio above unity when compared with 2D fixations, which already have a ratio above unity when compared with random positions. This result is very different than the **yosemite** dataset, in which the addition of depth made very little difference with respect to human fixations.

4.5 Experiment 2

In the second experiment, the distribution of center-surround differences was investigated at fixated positions. Analysis of band-pass center-surround features was previously explored in [78]. In addition, the center-surround mechanism has been studied extensively as it relates to saliency and the human visual system [20,79,80]. The biological motivation for this model is the on-center and off-center receptive fields of retinal ganglion cells. Rather than being sensitive to specific intensities or contrasts, pre-attentive vision is responsive to stimuli which exhibit contrasting center and surround responses.

The center-surround differences for intensity and disparity are computed as in [37]. To begin, a Gaussian pyramid is constructed for each image. Nine scales are considered, where the first scale is the original image, and each subsequent scale involves a Gaussian pre-filtering and down-sampling of 2 in each dimension. The ratio between the first and ninth scales will be $256:1$. The center-surround difference is then computed between a *surround* at scale $c+\delta$, and *center* at scale c. The coarser scale is interpolated to be of the same resolution as the finer scale. Finally, the difference is computed element-wise between the two patches.

We select a set of center windows corresponding to $0.125^\circ - 1.0^\circ$VA, and fix the ratio between center and surround to 3.0 as shown in Eq. 4.8.

$$c \in \{3,4,5,6\}, \quad s = c+\delta, \quad \delta = 3 \tag{4.8}$$

Center-surround differences for intensity and disparity are shown for **yosemite** in Fig. 4.9. The mean center-surround intensity measure is higher for human fixations than it is for randomly-sampled positions. For disparity, the measure is lower for human fixations. As we saw in the previous experiment for **yosemite**, there is little difference between the feature distribution at 2D and 3D fixations. Next, results are given for **middlebury** in Fig. 4.10. Here, the center-surround distribution follows the trend of the previous experiment. The contribution is higher at human fixations both for intensity and disparity. Slightly more

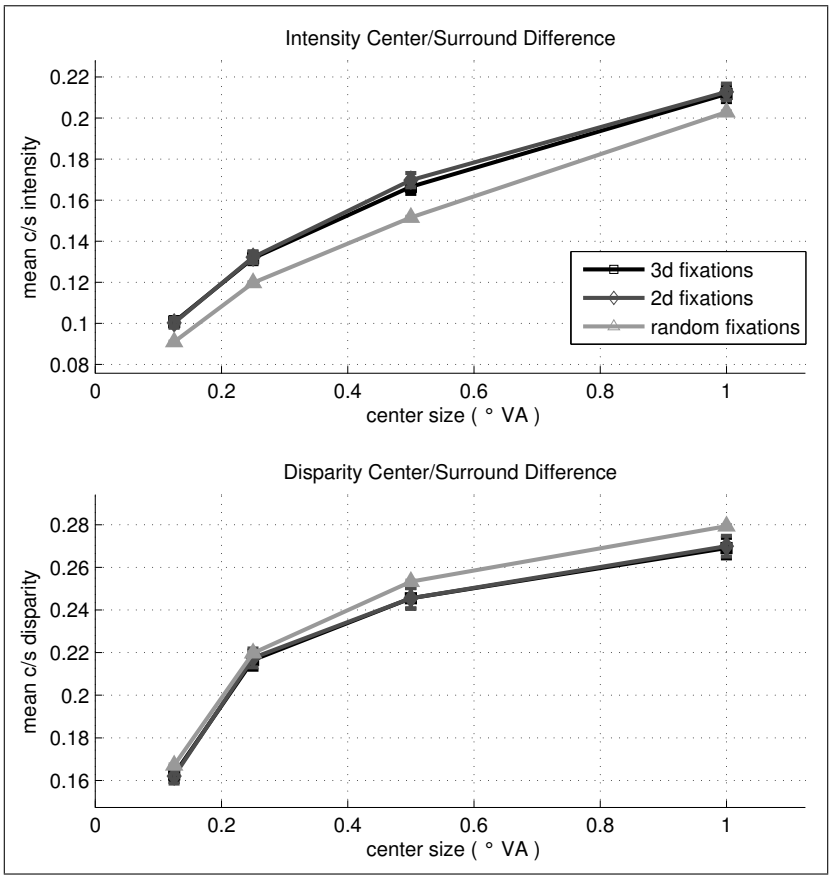


Figure 4.9: **Yosemite**: center/surround difference measure at human fixations and randomly-sampled positions. Features are computed as proposed in [37]. The 95% confidence interval on the mean is shown for each sample. For this dataset, the feature response for intensity is higher at human fixations than it is for randomly-sampled positions. The opposite is true for the disparity feature. Overall, the center-surround difference follows the same trend as the local contrast measure computed in experiment 1. The 2D and 3D fixations for both features are not clearly separable from one another, in that the confidence intervals are overlapping.

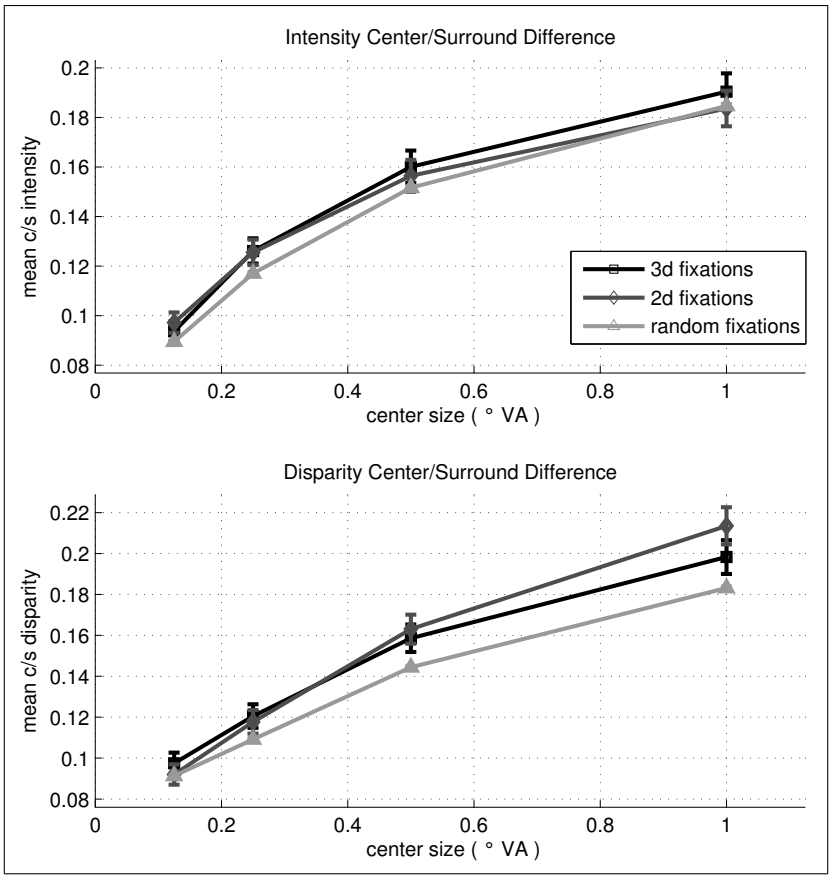


Figure 4.10: **Middlebury**: center/surround difference measure at human fixations and randomly-sampled positions. For both the intensity and disparity features, the measure is higher for attended positions than for random positions. As before, the 2D and 3D fixations are not clearly separable. For the case of disparity, the center/surround difference is clearly separable between the human fixations and random positions. This experiment follows the trend of the contrast measure in experiment 1.

variability is seen for the **middlebury** dataset between 2D and 3D fixations. As before, this data will be further investigated using a ratio test over the set of images.

4.5.1 Ratio Test

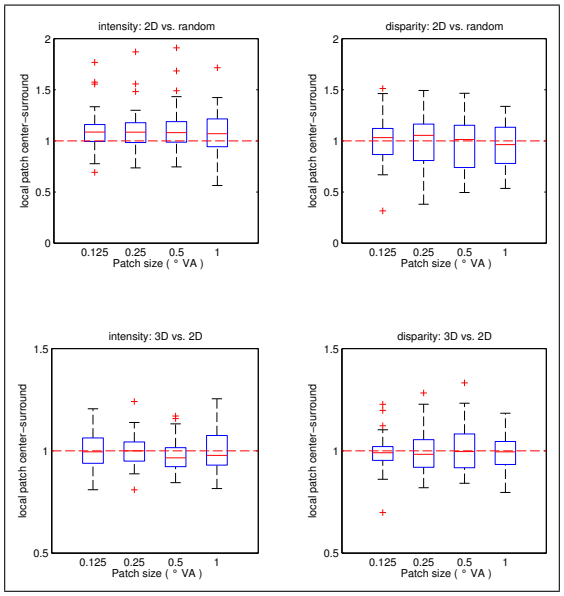


Figure 4.11: **Yosemite**: ratio tests for center-surround difference as a function of the center window size in °VA.

The ratio test is redefined for the second experiment. We examine the center-surround differences for intensity and disparity at varying scale. The ratios for intensity are given as RCS_I^{2d} and RCS_I^{3d}, where the superscript $2D$ is used for the comparison of 2D fixations with random positions and the superscript $3D$ is used for the comparison of 3D and 2D fixations.

$$RCS_I^{2d} = \frac{CS_{2d}^i}{CS_{ran}^i} \tag{4.9}$$

$$RCS_I^{3d} = \frac{CS_{3d}^i}{CS_{2d}^i} \tag{4.10}$$

Here, CS_{2d}^i is the mean of center-surround data for all 2D human fixations for image i. The disparity ratios RCS_D^{2d} and RCS_D^{3d} are computed in

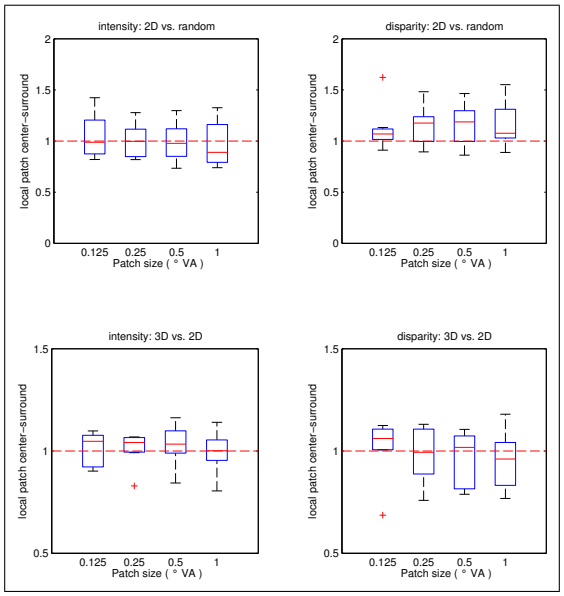


Figure 4.12: **Middlebury**: ratio tests for center-surround difference as a function of the center window size in °VA.

the same way, given the disparity map. All results for **yosemite** are shown in Fig. 4.11. We see that the center-surround difference is higher for intensity at random positions than for random locations. This is to be expected, as the bandpass luminance feature is known to be highly salient [60]. For disparity, the distribution fluctuates as a function of the patch size. Overall, the median is near zero. Next, the ratios for RCS_I^{3d} and RCS_D^{3d} are near unity. This indicates, as in the previous experiment, that depth information does not significantly change the nature of fixations for the **yosemite** dataset.

Finally, results of the ratio test for **middlebury** are shown in Fig. 4.12. The ratio for RCS_I^{2d} is near unity across the range of center sizes, while RCS_D^{2d} exhibits a trend above unity. This is consistent with findings from the previous experiment. It seems that changes in disparity attract human attention for this dataset. The ratio between 3D and 2D fixations is less clear, in this case. For RCS_I^{3d}, the unity line is contained within the 25th-75th percentile box, indicating that no clear trend is present. Finally, RCS_D^{3d} is above one for a center window of size 0.125°VA and near one for larger patch sizes.

4.6 Discussion

In this work, we have explored the feature content of intensity and disparity at human fixations using an eye-tracking system. We have included all recorded data on our website [1]. Eye tracking data was recorded for the **middlebury** [71] and **yosemite** [72] datasets. We investigated the gradient and contrast at fixations, motivated by the work of [63]. In addition, center-surround differences were explored.

For the **yosemite** natural image dataset, our results concur with previous research. We measured a lower disparity gradient and contrast at human fixations when compared with randomly-sampled positions. However, little difference is observed between the feature content at 2D and 3D fixations. Center-surround differences were also investigated for this dataset, yielding a similar trend. For the synthetic **middlebury** dataset, disparity gradient and contrast were both higher for human fixations than random. In addition, intensity contrast was higher and intensity gradient showed no significant trend. Center-surround measures were higher for both intensity and disparity at fixations. This dataset exhibited different feature distributions when depth was removed from the experiment. We show that disparity contrast and center-surround difference attracts attention when it is present, due to a ratio above unity when comparing 3D and 2D fixations.

The difference in construction between the two datasets is the most likely explanation for the varying fixation distributions. The **yosemite** set elicits bottom-up salient cues primarily due to the lack of recognizable objects. As was explained previously, bottom-up vision is inherently low-bandwidth and must rely on simple features for rapid scene understanding. Regions of high disparity contrast are therefore avoided, as these require significantly more processing. Meanwhile, we have experienced the presence of top-down salient cues being fixated in the **middlebury** dataset. An example of this is a large concentration of fixations on a doll's face. This shift in bottom-up to top-down salient features

[1] `http://videoprocessing.ucsd.edu/~NatanHaim`

would explain the investigation of regions which require a higher level of visual processing. In addition, we observe a significantly larger difference between 2D and 3D fixations for the **middlebury** set.

The text of Chapter 4 is adapted from *Effect of Stereoscopic Depth on Saliency: Evidence from Eye Movements*, Natan Jacobson, Elizabeth Schotter, Yang Liu, Alan Bovik, Keith Rayner, Truong Nguyen, in preparation for *Journal of Vision*. The dissertation author is the primary author of this publication.

Chapter 5

Occlusion Boundary Detection using Online Learning

"We are all connected; To each other, biologically. To the earth, chemically. To the rest of the universe atomically." - Neil deGrasse Tyson

In a departure from saliency-based methods, we investigate the video occlusion problem. An occlusion occurs whenever a foreground object covers or uncovers a background object, thereby changing the information content in the scene. Proper detection of occlusion boundaries allows for improvements to a number of video processing algorithms, such as motion estimation and disparity estimation for 3D scenes. In this work, we propose a simple approach to detect occlusion boundaries based on online learning. Unlike many learning-based methods, online learning does not require an expensive training stage, thereby making it easier to use in practice. Our algorithm exhibits improved performance with respect to previous non-training-based methods and is competitive with more expensive training-based methods. In addition, we demonstrate the best overall performance for a synthetic sequence for which no training data is available.

5.1 Introduction

Digital video has become ubiquitous over the last two decades. Demand is growing at an exceptional rate for mobile video, high quality digital television and even 3D broadcast and movies. This demand has necessitated significant research in the fields of compression, motion-compensated interpolation, and disparity estimation. Fundamental to each of these research topics is the concept of occlusion boundary detection; where an occlusion is the region between two overlapping objects with disparate motion. Detecting these events is crucial because many of the video processing assumptions fail at occlusion boundaries. For example, the smoothness constraint of disparity estimation does not hold true across an occlusion boundary. However, if the occlusion boundary is detected, then the smoothness term can be set to zero at these locations.

Occlusions are omnipresent and are crucial for visual understanding in a three dimensional world [81]. By strict definition, an occlusion event occurs whenever one object is covered or uncovered by another object which is spatially closer to the observer. The observer may be a human, still camera or video camera. For humans, occlusion is crucial in order to infer the relative depth of objects in the world. In fact, occlusion boundaries play a central role in human stereopsis [82], which determines how the human visual system perceives three dimensions from stereo vision.

The distinction between *occlusion boundaries* and *appearance edges* is an important one. Here, an *appearance edge* refers to the typical output of an edge detection algorithm (e.g. Canny edge detector [83]) on luminance or color image data, whereas an *occlusion boundary* is explicitly created by objects covering or uncovering one another. An example of an occlusion boundary from the **Tsukuba** stereo image pair is demonstrated in Fig. 5.1. In this stereo pair, the lampshade in the foreground occludes the bookcase in the background. Whereas occlusion boundaries typically occur at appearance edges, the detection of an appearance edge is in no way sufficient to guarantee an occlusion boundary. For example, the significant edge on the nose of the ceramic bust in Fig. 5.1 is not

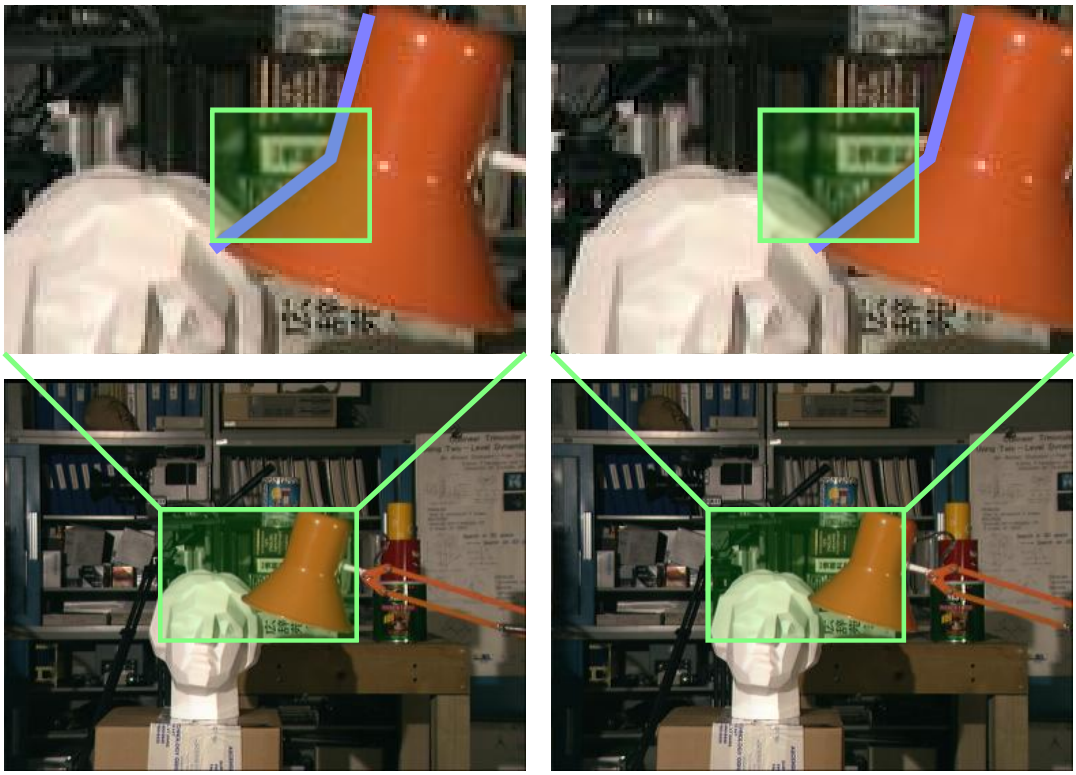


Figure 5.1: Occlusions evident in **Tsukuba** stereo image pair. The lampshade occludes pixels in the left view which are visible in the right view. Occlusion boundary in the selected window is highlighted in blue.

an occlusion boundary, rather it is due to the lighting of the scene. Likewise, edges in the motion field may be an indicator that an occlusion boundary is present, however no guarantee is made. It is, however, evident that valuable information is provided by both the appearance cue and the motion field. In this work, both cues will be considered. To calculate the motion cue, an off-the-shelf dense optical flow method is used [84]. This motion information is combined with pixel-domain information in an online learning framework.

The online learning framework is based on the idea of a panel of experts, where each expert is proficient at a distinct task. Each time a decision needs to be made, each expert is polled and a loss is calculated based on how correct their prediction turned out to be. Over time, the best expert for each task will become evident. In this work, the task of each expert is to detect a certain type

of occlusion event, and the quality of each expert's detection is measured using a loss function in the pixel domain (similar to a standard distortion measure). At a specific location, an occlusion boundary is detected based on the best prediction over all experts. Contrary to all competing approaches, the proposed method does not require any training; rather the classification of occlusion boundaries is based on the relative weighting of the experts, which occurs online.

This section is organized as follows. Previous methods for the detection of occlusion boundaries are described in Section 5.2. The underlying concepts of online learning are presented in Section 5.3. Next, our algorithm is presented in Section 5.4 along with a description of the feature set and experts used. Finally, results are presented in Section 5.5 and we conclude in Section 5.6.

5.2 Previous Work

The presence of occlusions in the image and video processing literature is astoundingly diverse. Research is conducted both to determine occlusion boundaries explicitly and to use this information implicitly in the pursuit of other results.

Implicit determination of occlusion boundaries has been cited in applications of object tracking [85], segmentation [86–88] and disparity estimation [89–91]. In all of these cases, occlusions are handled implicitly due to the information they provide about the scene. Two important concepts related to disparity estimation and pertinent to these works are the *uniqueness constraint* and the *ordering constraint* [91]. The *uniqueness constraint* states that features in the left and right image are in one-to-one correspondence. This concept is extended naturally to monocular video as objects generally do not appear or disappear during a single frame period. The *ordering constraint* states that the ordering of two objects in the left view is maintained in the right view. Because these concepts apply to objects in a stereo pair or video sequence, they must also apply to interactions between objects. For this reason, the two constraints must also apply for occlusion detection.

A number of methods have been applied to the problem of explicitly determining occlusion boundaries. One early approach determines occlusion boundaries for a set of known object types on a small pixel grid [92]. This method showed promising performance assuming no noise and a set of a priori known objects. Later, occlusion detection was combined with motion estimation to classify occluded areas based on a photometric mismatch between frames [93]. The drawback of this method is that any errors in the motion vector field are likely to cause false detections of occlusion boundaries. Around the turn of the 21st century, learning-based research was conducted which used two separate models to describe scene motion: a two-parameter translational model for regular motion and a six-parameter generative model for occlusion boundaries [94]. A graph-cuts approach has also been considered in which the uniqueness constraint is utilized to guarantee proper occlusion handling [95]. In a separate approach, occlusion events are determined by the presence of T-junctions in a spatiotemporal volume created from a video sequence [96]. Geometric approaches have also been considered analyzing the motion field alone to determine the presence of occlusions [97]. A more recent direction makes use of local appearance cues as well as motion information to detect occlusion boundaries [98, 99]. It is demonstrated that the combination of cues performs better than any cue used independently. In a separate project, researchers have applied a probabilistic framework for considering occlusion information across multiple frames [100]. The drawback of these learning-based methods is that, in order to obtain good detection performance, significant training data must be available. Additionally, this training data must be very similar in content to the test data in order for the trained features to contain sufficient discriminating power.

Additional motivation for this research was obtained from the excellent Particle Video research paper [44]. In this work, motion estimation is posed as a particle tracking problem using particle appearance and inter-particle distortion. Here, a significant improvement is demonstrated over standard optical flow methods.

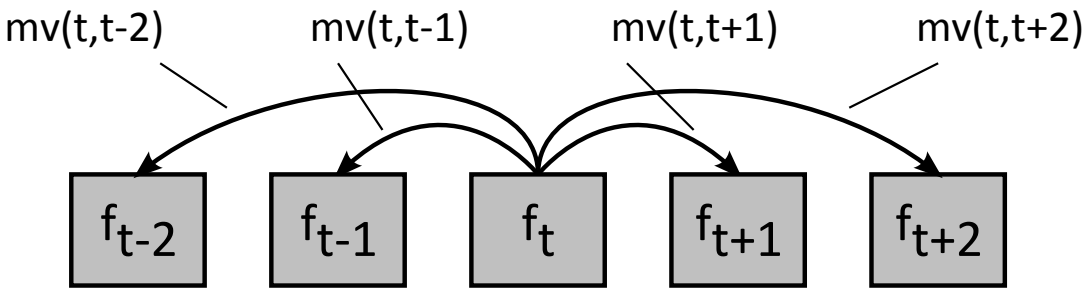


Figure 5.2: Notation used for motion estimation. $mv\,(t,s)$ represents the motion field between frames f_t and f_s.

5.3 Online Learning

We pose occlusion boundary detection as a problem of prediction over a video sequence. This formulation is well-suited for video as the correctness of each prediction at frame f_t will be revealed by the subsequent frame, f_{t+1}. The proposed framework is based on the Hedge online learning algorithm [101]. Rather than requiring a training stage, as do most conventional learning algorithms, the Hedge algorithm may be run against a video sequence without seeing any previous input. The Hedge algorithm is based on a panel of experts, where each expert is tuned to perform a simple decision at each frame instance. Decisions are made using the motion vector field (MVF) and pixel-domain information. The MVF is computed using an optical flow technique [84]. The flow field between frames f_t and f_s will be denoted as $mv\,(t,s)$ as shown in Fig. 5.2. The correctness of each decision is then measured using a loss function. Experts are weighted based on their loss functions such that good performance (low loss) is rewarded with a higher weighting. Finally, the Hedge algorithm makes its prediction based on the weighting of all experts. The flexibility of the algorithm allows us to choose the number of experts, and to determine how each expert makes its decision. In this work, the experts have been designed such that each one is tuned to detect a specific type of occlusion boundary. As will be presented, the loss function is measured using the Sum of Absolute Differences (SAD) between predicted image patches and image patches in the

current frame.

5.3.1 Pixels And Particles

In this work, we distinguish between *pixels* and *particles* to be as explicit as possible. Whereas a pixel has the standard meaning, we use the term *particle* to refer to a picture element which will be tracked through the video sequence, and will maintain certain properties. In the first time-step of the video sequence, a large number of particles will be initialized such that one particle exists for each pixel in the image, minus the boundary. The online learning framework will then be applied to each particle, which will be classified based on the prediction and loss over the set of experts. Between time-steps, each particle will be propagated based on the calculated optical flow. In this way, each particle maintains its history, and the classification will gain confidence over time. Special care is taken to ensure the particle tracking grid remains dense between time-steps, as is discussed in Section 5.4.

5.3.2 Notation

The following notation is used for all further discussion involving Hedge and the proposed algorithm. The subscript i will be used to index the set of experts while subscript m is used to index the set of particles. The time-step is denoted by the superscript t. For example, the variable $w_{i,m}^{t}$ will refer to the weighting of expert i at time-step t for particle m. A full listing of variables is presented in Table 5.1.

Further description and analysis of the experts is presented in the following section. A complete description of the proposed algorithm is then provided in Section 5.4.

Table 5.1: Variable list

f_t	frame t of the input sequence
m	particle index
i	expert index
t	time-step
η	learning rate
$X\,(m)$	X and Y coordinates for particle m
$D\,(m)$	indicator vector for inactive particles
$C\,(m)$	occlusion classification for particle m
e_i	expert i
$l_{i,m}^t$	instantaneous loss
$L_{i,m}^t$	cumulative loss
$w_{i,m}^t$	attributed weight
$p_{i,m}^t$	normalized weight (probability)
c	size in pixels of ignored image boundary
θ	parameter to specify occlusion boundary angle
α	parameter to specify covering/uncovering
$\Gamma(\theta, \alpha)$	occlusion type for parameters θ, α
δ	patch size for calculation of expert loss
τ	multiplicative weight for no-occlusion expert

5.3.3 Description Of Experts

The goal of each *expert* is to detect a specific type of occlusion boundary. In the online learning framework, this is accomplished by each expert predicting content in the subsequent frame congruous with its specific occlusion boundary type. Parameters must be defined so that each occlusion boundary type is explicitly stated. Once this is accomplished, the calculation of *loss* can be defined, ultimately leading to the proposed algorithm for occlusion boundary detection.

Each expert is associated with two parameters which determine its prediction. These parameters are angle of the occlusion boundary, and whether the foreground object is covering or uncovering the background. The first parameter is illustrated in Fig. 5.3, where the angle of occlusion boundary is in the set $\theta \in \left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\right\}$. Note that additional angles may be included if the user desires a larger set of experts. The next parameter distinguishes between a *covering*

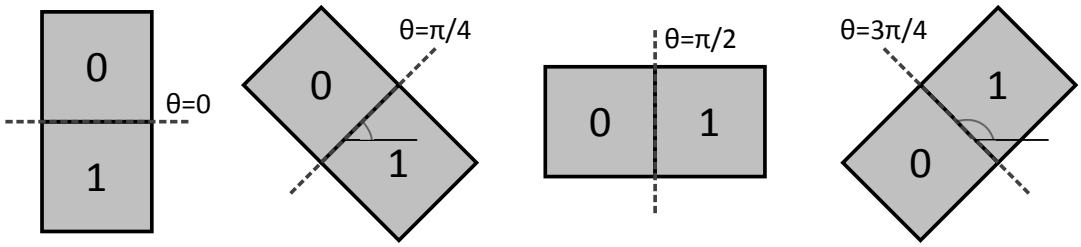


Figure 5.3: Occlusion type parameter θ for the angle of the occlusion boundary. For each angle, the foreground object at the occlusion boundary can either be *covering* or *uncovering* the background. The labels $0, 1$ are used to make explicit the two sides of the occlusion boundary.

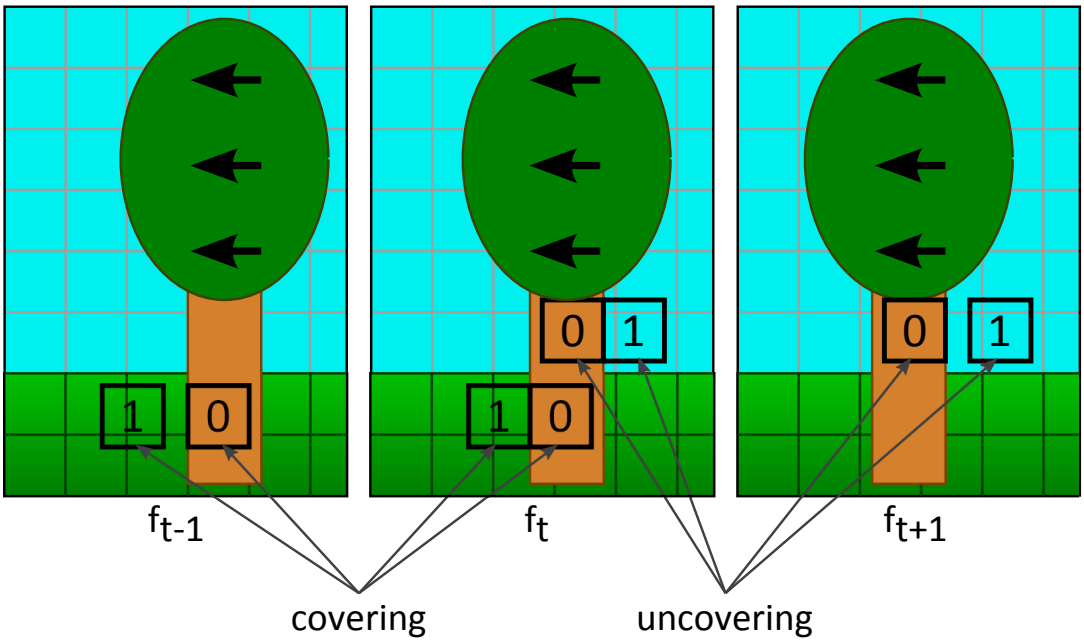


Figure 5.4: *covering* ($\alpha = 1$) and *uncovering* ($\alpha = 0$) occlusion boundaries for frame f_t. Both boundaries are vertical ($\theta = \frac{\pi}{2}$) with the tree in the foreground. The expert for the covering occlusion boundary uses frame f_{t-1} as reference with motion field $mv\,(t, t-1)$ while the expert for the uncovering case uses frame f_{t+1} with motion field $mv\,(t, t+1)$.

and an *uncovering* occlusion event, as illustrated in Fig 5.4. Parameter $\alpha = 1$ denotes the foreground object covering the background, while $\alpha = 0$ denotes

the foreground object uncovering the background. To make these concepts more concrete, a series of *occlusion types* are defined, where each is fixed to a specific set of parameters. Denote an occlusion type as $\Gamma(\theta, \alpha)$, where the parameters θ, α are as described in this section. For example, the two experts in Fig. 5.4 detect a vertical covering occlusion type $\left(\Gamma\left(\frac{\pi}{2}, 1\right)\right)$, and a vertical uncovering occlusion type $\left(\Gamma\left(\frac{\pi}{2}, 0\right)\right)$, respectively. The patch labels $0, 1$ are included to explicitly distinguish between the two sides of the occlusion boundary.

It is clear from this formulation that the introduction of each additional angle will create two unique occlusion types. Analysis has been conducted which demonstrates a decreasing marginal benefit to performance as the set of angles is increased. This is discussed further in Section 5.5. For the time being, it should be mentioned that for presented results, the four occlusion boundary angles in Fig. 5.3 are considered. Therefore, a total of 8 occlusion types are included in the framework, along with a default case which assumes no occlusion event. Because each of these types is predicted by a unique expert, the current formulation of the problem includes a total of $N = 9$ experts.

5.3.4 Instantaneous Loss Calculation

The instantaneous loss of each expert is measured using the Sum of Absolute Differences (SAD) error metric, as is demonstrated in Fig. 5.5. The SAD is computed between predicted and revealed image patches either adjacent to or centered on the particle. For all occlusion types which predict an occlusion boundary, the patches are adjacent to the boundary, with one patch on either side. For the null case which predicts no occlusion boundary, the patch will be centered on the particle. If, for example, the image patch is of size $\delta \times \delta$, and the expert is of type $\Gamma\left(\frac{\pi}{2}, 0\right)$, then the centers of the two patches will be located at $\left(x(m) \pm \lceil \frac{\delta}{2} \rceil, y(m)\right)$ where $(x(m), y(m))$ is the location of particle m. Denote $x_0 = x(m) - \lceil \frac{\delta}{2} \rceil$ and $x_1 = x(m) + \lceil \frac{\delta}{2} \rceil$, and consider the motion vector pair u_0, v_0 to be the x and y components of the motion field at x_0, and u_1, v_1 to be the components at x_1. Finally, it should be mentioned that for an

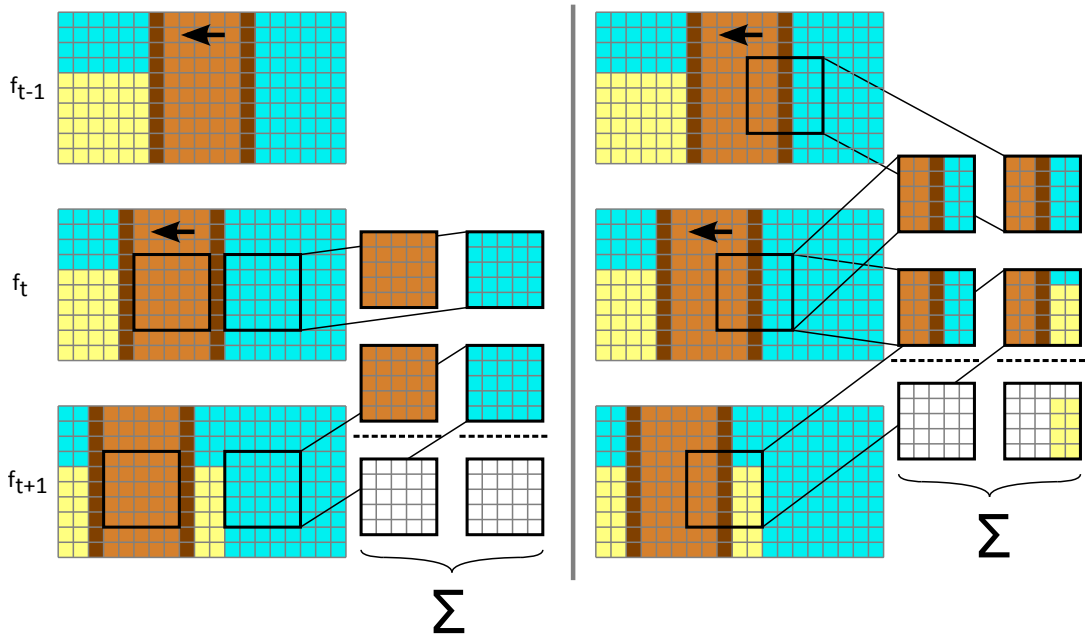


Figure 5.5: Example of instantaneous loss calculation for two experts. The tree is moving to the left against a static background. On the left side of the figure, occlusion type $\Gamma\left(\frac{\pi}{2}, 0\right)$ is shown and the instantaneous loss is computed for patches on either side of the occlusion boundary. On the right side of the figure, the null occlusion type is shown and instantaneous loss is calculated for a centered patch. For both occlusion types, instantaneous loss is calculated as the SAD between patches in frame f_t and predicted patches in frames $f_{t\pm1}$. In the case of the null occlusion type, prediction error occurs for the background, indicating that this expert will have a larger instantaneous loss than the expert on the left.

uncovering occlusion boundary event, the motion field $mv\,(t, t+1)$ will be used with predicted patches in frame f_{t+1}. However, for a *covering* occlusion type, the motion field $mv\,(t, t-1)$ will be used instead with frame f_{t-1}. For the null case, both motion fields will be used.

For the occlusion type $\Gamma\left(\frac{\pi}{2}, 0\right)$, the SAD for the two patches, l_0 and l_1 are calculated as:

$$l_0 = \sum_{a=-\lfloor\frac{\delta}{2}\rfloor}^{\lfloor\frac{\delta}{2}\rfloor} \sum_{b=-\lfloor\frac{\delta}{2}\rfloor}^{\lfloor\frac{\delta}{2}\rfloor} |f_t\left(x_0 + a, y(m) + b\right) - f_{t+1}\left(x_0 + a + u_0, y(m) + b + v_0\right)| \tag{5.1}$$

$$l_1 = \sum_{a=-\lfloor\frac{\delta}{2}\rfloor}^{\lfloor\frac{\delta}{2}\rfloor} \sum_{b=-\lfloor\frac{\delta}{2}\rfloor}^{\lfloor\frac{\delta}{2}\rfloor} |f_t\left(x_1 + a, y(m) + b\right) - f_{t+1}\left(x_1 + a + u_1, y(m) + b + v_1\right)|$$

The total instantaneous loss for the expert is the sum of the two patches: $l_{i,m}^t = l_0 + l_1$. Instantaneous loss for the other experts is calculated likewise, however the location and orientation of the patches will depend on the angle θ of the occlusion boundary and the user-defined parameter δ.

For the null expert which predicts the lack of an occlusion boundary, instantaneous loss is calculated for a patch cenetered on the particle with reference frames f_{t-1} and f_{t+1}. Here, the SAD for the two patches are calculated as:

$$l_0 = \sum_{a=-\lfloor\frac{\delta}{2}\rfloor}^{\lfloor\frac{\delta}{2}\rfloor} \sum_{b=-\lfloor\frac{\delta}{2}\rfloor}^{\lfloor\frac{\delta}{2}\rfloor} |f_t\left(x(m) + a, y(m) + b\right) - f_{t+1}\left(x(m) + a + u_0, y(m) + b + v_0\right)| \tag{5.2}$$

$$l_1 = \sum_{a=-\lfloor\frac{\delta}{2}\rfloor}^{\lfloor\frac{\delta}{2}\rfloor} \sum_{b=-\lfloor\frac{\delta}{2}\rfloor}^{\lfloor\frac{\delta}{2}\rfloor} |f_t\left(x(m) + a, y(m) + b\right) - f_{t-1}\left(x(m) + a + u_1, y(m) + b + v_1\right)|$$

and $l_{i,m}^t = \tau\left(l_0 + l_1\right)$. The parameter τ controls the sensitivity of the detector to occlusion boundaries. We vary this parameter in order to generate the precision-recall plots shown in Section 5.5. For the proposed work, we fix the parameter $\delta = 7$ for all experiments. This selection was made to capture sufficient

local pixel-domain information without requiring a huge number of pixel calculations for each expert, which would dramatically increase the computational complexity.

5.4 Proposed Algorithm

A flowchart in Fig. 5.6 depicts the three main portions of the proposed algorithm. In the first portion, the particle tracking grid is initialized (if it is the first time-step), and optical flow fields are computed. Next, the Hedge algorithm is used to compute the probability distribution over the experts for each particle. In the third portion, each particle is classified and the particle tracking grid is propagated.

In the first stage of the proposed algorithm, motion vector information is obtained for the video sequence using an optical flow technique [84]. As is standard in optical flow methods, the brightness constancy and gradient constancy assumptions are combined to obtain a motion field between two subsequent frames. A third term is included in [84] which enforces spatio-temporal smoothness while preserving spatial discontinuities. The resulting dense motion vector field $mv\,(t, t \pm 1)$ is defined for each pixel in the sequence.

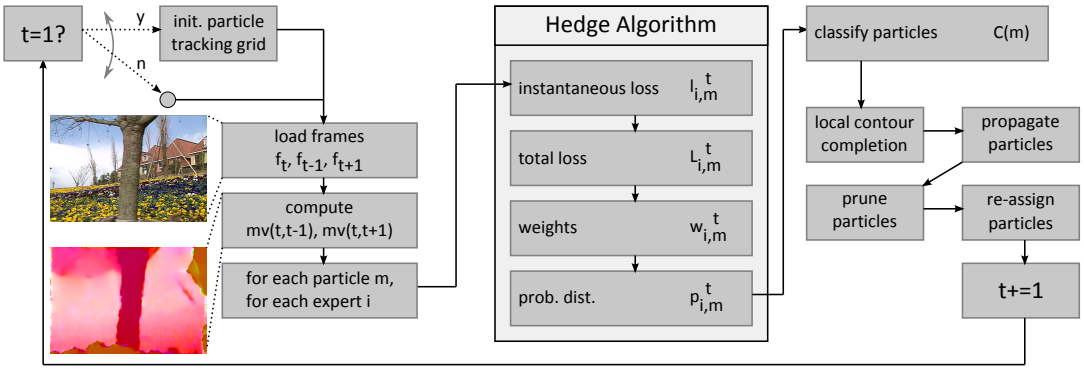


Figure 5.6: Flowchart for the proposed occlusion boundary detection algorithm.

The Hedge algorithm [101], presented in Algorithm 2, is responsible for

Algorithm 2 Hedge

initialize expert weights to be uniform: $w_i^1 = \frac{1}{N}, \forall i$
for $t = 1, \ldots, T$ **do**
  for $m = 1, \ldots, M$ **do**
    for $i = 1, \ldots, N$ **do**
      expert e_i makes prediction
      instantaneous loss calculated: $l_{i,m}^t$
      if exponentially discounted loss **then**
        total loss: $L_{i,m}^t = (1-\alpha)\sum_{s=1}^{t-1} l_{i,m}^s + l_{i,m}^t$
      else
        total loss: $L_{i,m}^t = \sum_{s=1}^{t} l_{i,m}^s$
      end if
      weights updated: $w_{i,m}^t = w_{i,m}^1 \exp^{-\eta L_{i,m}^t}$
      form prob. dist. over experts: $p_{i,m}^t = \frac{w_{i,m}^t}{\sum_{j=1}^{N} w_{j,m}^t}$
    end for
  end for
end for

detecting occlusion boundaries using a set of experts, each of which is tuned to detect a separate occlusion type. The input to the algorithm is the pixel-domain information $f_t, f_{t\pm1}$ as well as the motion vector fields $mv\,(t, t \pm 1)$. The output is a probability distribution over the set of experts for each particle. This results in a classification of each particle into an occlusion type.

The algorithm is initialized with a dense grid of particles in the first frame of the video sequence. The particle grid is the same resolution as the video sequence with a border of $c = 10$ pixels removed to avoid edge effects. Particles are indexed by $m \in \{1, \ldots, M\}$ where $M = (W - 2c) \times (H - 2c)$ and W, H are the width and height in pixels, respectively, for the video sequence. Particle locations are stored in the vector $\mathbf{X} \in \mathbb{Z}^{\mathbf{M}\times\mathbf{2}}$ where each row of $\mathbf{X}$, denoted $X\,(m)$ contains the x and y coordinates of particle m. A set of experts $e_1, \ldots, e_N$ predicts local patch information in f_{t-1} or f_{t+1} based on the occlusion type. The instantaneous loss of each expert is calculated using the SAD error metric. Next, cumulative loss is calculated using an exponentially discounted loss function and stored in $\mathbf{L} \in \mathbb{R}^{\mathbf{M}\times\mathbf{N}}$. The exponentially discounted loss function is:

$$L_{i,m}^{t} = (1 - \alpha)\, L_{i,m}^{t-1} + l_{i,m}^{t} \tag{5.3}$$

where $L_{i,m}^{t}$ is the cumulative loss of expert i at time step t for particle m. Each expert is re-weighted at each time-step based on the cumulative loss and a tunable *learning rate*, η.

$$w_{i,m}^{t} = \frac{1}{N} \exp^{-\eta L_{i,m}^{t}} \tag{5.4}$$

Weights $w_{i,m}^{1}$ are initialized to be uniform. That is, $w_{i,m}^{1} = \frac{1}{N}$, $\forall i$. The learning rate is set to $\eta = \sqrt{\frac{2 \ln N}{T}}$ as proposed in [101]. This learning rate is selected as it guarantees an upper bound on the cumulative loss of the Hedge algorithm:

$$L_{\text{hedge}}\,(\eta) \leq \min_{i} L_i + \sqrt{2T \ln N} + \ln N \tag{5.5}$$

where N is the total number of experts and T is the number of time-steps. This means that Hedge will always achieve a cumulative loss close to that of the best

expert at time t. Next, the weights are normalized to produce a probability distribution over the experts. Intuitively, this is a measure of how well each expert is performing.

$$p_{i,m}^{t} = \frac{w_{i,m}^{t}}{\sum_{j=1}^{N} w_{j,m}^{t}} \tag{5.6}$$

At each time-step and for each particle, the weights attributed to the N experts are compared. Classification is performed based on the expert with the largest weighting at time-step t. The classification function is:

$$C\left(m\right) = \underset{i}{\operatorname{argmax}} \left(p_{i,m}^{t}\right) \tag{5.7}$$

Therefore if expert e_i is tuned to detect occlusion type $\Gamma\left(\frac{\pi}{4}, 1\right)$, then particle m will be classified as a covering occlusion boundary of angle $\theta = \frac{\pi}{4}$ if $p_{i,m}^{t}$ has the largest value over the vector p.

5.4.1 Local Contour Completion

Subsequent to the Hedge update at each time-step, a contour completion stage is conducted. This is similar to the binary morphological operation of dilation. We have determined that this operation significantly improves detection performance while adding little complexity. Performance improvement is observed because the Hedge method may result in small gaps in the detected occlusion boundary due to factors such as noisy pixel data and a slow learning rate. For each particle which has registered an occlusion event, a short segment of potential particles are aligned based on the classified occlusion type of the particle. An example is shown in Fig. 5.7 for an occlusion type of angle $\theta = \frac{\pi}{4}$. If more than half of these particles also observe an occlusion event tuned to the same angle, then the entire line segment is classified identically.

5.4.2 Particle Propagation

Between two adjacent time-steps, the particles are propagated via the motion vector field. If particle m is located at position $\mathbf{X}\left(m\right) = \left(x\left(m\right), y\left(m\right)\right)$

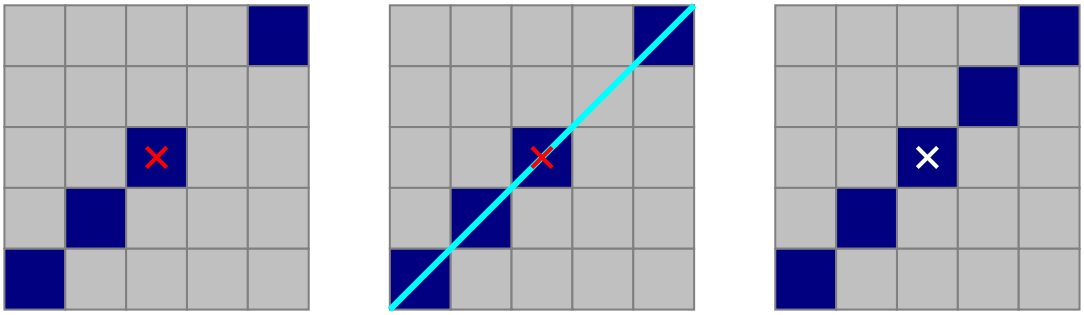


Figure 5.7: Contour completion at angle $\theta = \frac{\pi}{4}$ using four neighbors and threshold of 50%. Center pixel marked with red X.

in frame f_t, then it will be propagated to:

$$\mathbf{X}(m) = (x(m) + u, y(m) + v) \quad \text{in } f_{t+1} \tag{5.8}$$

where u, v are the components of the motion vector, located in the motion vector field $\mathbf{mv}(t, t+1)$ at position $(x(m), y(m))$.

5.4.3 Particle Pruning and Reassignment

The number of particles is kept constant by merging particles which belong to the same pixel, and introducing new particles where appropriate. In addition, particles which are propagated to within $c = 10$ pixels of the image boundary will be pruned, as it is unnecessary to track particles which are exiting the frame. If multiple particles are propagated to the same pixel, all but one of them will be pruned, and the average of the cumulative losses will be assigned to the remaining particle. After particle pruning is complete, new particles will be assigned such that the particle tracking grid remains dense. For time-step $t+1$, the particle tracking grid is examined to determine if there are any pixel locations such that no particle exists. If one is found, a new particle is added to the tracking grid with uniform weights over the set of experts: $w_{i,m}^{t+1} = \frac{1}{N}$, $\forall i$

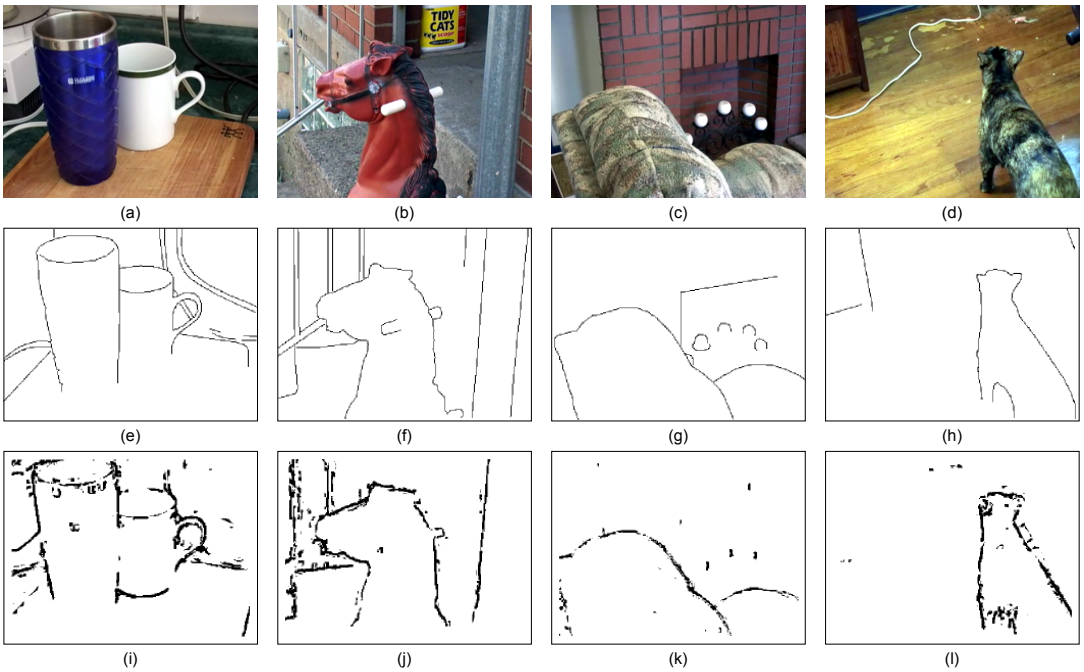


Figure 5.8: Comparison of the proposed algorithm with ground truth occlusion boundaries for four sequences of the CMU database. For each sequence, the frame is shown in the top row, ground truth in the middle row, and the result of the proposed occlusion boundary detector in the bottom row. Columns from left to right: **mugs2** ($F = 49.10\%$), **rocking horse** ($F = 54.50\%$), **couch corner** ($F = 54.31\%$), **zoe1** ($F = 50.29\%$)

5.5 Simulation Results

The proposed occlusion boundary detection algorithm has been tested against sequences obtained from the *CMU Video Dataset for Occlusion/Object Boundary Detection* [98]. This dataset is comprised of 30 short video sequences, each of which contains labeled ground truth occlusion boundaries. In addition, a synthetic sequence has been created which is significantly different than the sequences available in the CMU dataset. This will test the robustness of learning-based methods for which cross-validation is not possible. Additional results, including detection results for full video sequences, are included on the

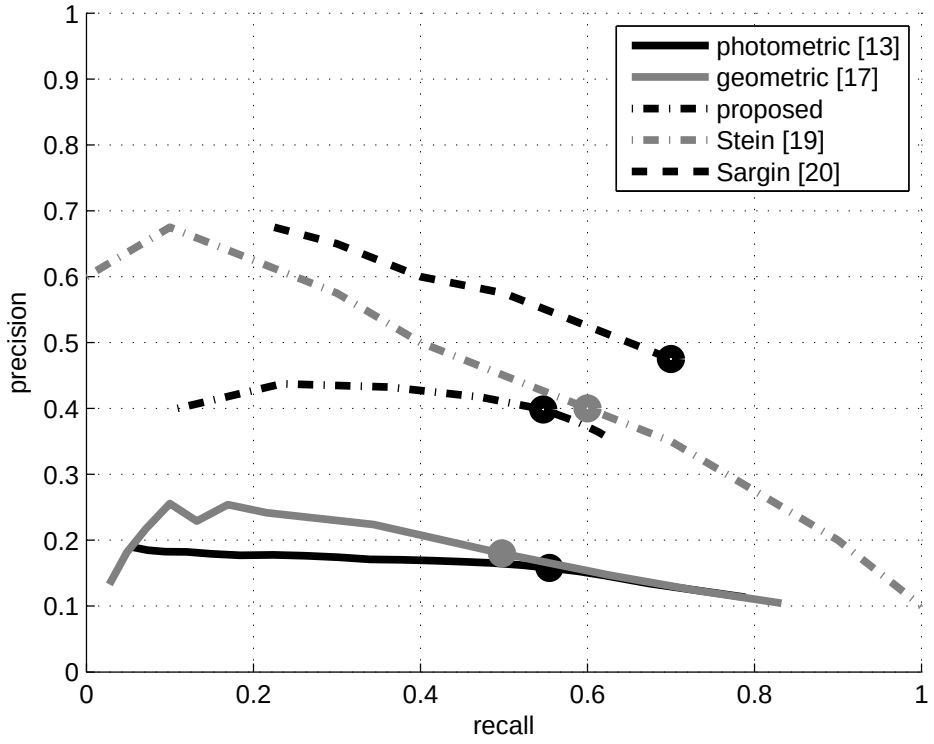


Figure 5.9: Comparison of occlusion boundary detection performance using precision-recall. Point of maximum F-score marked for each curve. Best performance of each method: Photometric: $F_{max} = 22.637\%$, Geometric: $F_{max} = 24.636\%$, Proposed: $F_{max} = 43.857\%$, Stein [99]: $F_{max} = 48.000\%$, Sargin [100]: $F_{max} = 56.596\%$

author's website[1].

5.5.1 CMU Occlusion Dataset

Each sequence in the CMU dataset consists of between 5 and 30 frames and includes hand-labeled ground truth occlusion boundaries for the center frame. An objective comparison is performed between each occlusion boundary detector and the ground truth data from the dataset using precision-recall. An

[1] `http://videoprocessing.ucsd.edu/~NatanHaim/TIP_2011a/`

example of this is demonstrated in Fig. 5.8 where four frames from the dataset are shown along with the ground truth occlusion boundaries and the result of the proposed online detection algorithm. Precision and recall scores are calculated as follows: let $\{x_{gt}\}$ denote the set of occlusion boundary pixels in the ground truth data and $\{x_d\}$ denote the set of occlusion boundary pixels detected by the proposed algorithm. Then precision, recall, and F-score values are calculated as:

$$p = \frac{|x_{gt}| \wedge |x_d|}{|x_d|}, \quad r = \frac{|x_{gt}| \wedge |x_d|}{|x_{gt}|} \quad F = \frac{2pr}{p+r} \tag{5.9}$$

where the F-score F can equivalently be calculated as the ratio between the common ground truth and detected occlusion boundaries, and the mean between the two, that is: $F = \frac{2(|x_{gt}| \wedge |x_d|)}{|x_{gt}|+|x_d|}$. In general, results are presented using this single metric. The proposed algorithm is compared with four competing methods from the occlusion literature. The first approach we compare with is based on the photometric difference between adjacent frames [93] and is denoted as the *photometric* approach. Here, the mismatch in intensity between two adjacent frames is measured using the motion field. For two adjacent frames f_t and f_{t+1} and the forward and backward motion fields $d_f = mv\,(t, t+1)$ and $d_b = mv\,(t+1, t)$, the motion-compensated prediction errors are given by:

$$\begin{aligned} \epsilon_f\,(\mathbf{x}) &= f_t\,(\mathbf{x}) - f_{t+1}\,(\mathbf{x} + d_f\,(\mathbf{x})) \\ \epsilon_b\,(\mathbf{x}) &= f_{t+1}\,(\mathbf{x}) - f_t\,(\mathbf{x} - d_b\,(\mathbf{x})) \end{aligned} \tag{5.10}$$

The absolute value of these two errors are compared with a threshold Θ to determine the presence of occlusion boundaries. The precision-recall curve in Fig. 5.9 is produced by sweeping the threshold in the range $\Theta \in [0, 255]$. In the second *geometric* approach, uncovered regions are detected by locating areas in the reference frame for which no motion candidates exist in the current frame [97]. If we denote a sampling lattice in frame f_t as Λ, and we denote $S = \{\mathbf{y} : \mathbf{y} = \mathbf{x} + d_f\,(\mathbf{x})\,, \mathbf{x} \in \Lambda\}$ as the set of spatial positions in f_{t+1} achieved by motion-compensating the sampling lattice of f_t, then an indicator function

may be defined as:

$$\xi_i(\mathbf{x}) = \begin{cases} 1, & \|\mathbf{x} - \mathbf{z_i}\| \leq r \\ 0, & \text{otherwise} \end{cases} \qquad \mathbf{x} \in \Lambda, \mathbf{z_i} \in S \tag{5.11}$$

where $\mathbf{x}$ and $\mathbf{z_i}$ are in f_{t+1}, and the radius r is defined by the user (in this work, as in [97], $r = 2$). Finally, $M(\mathbf{x}) = \sum_{i=1}^{|S|} \xi_i(\mathbf{x})$ measures the density of projections. This value is thresholded to generate the precision-recall curve. In order to detect covered regions, the reverse process is computed from $f_{t+1} \rightarrow f_t$. The third approach considered is the training-based method of [99] in which numerous appearance and motion features are considered in the training of a binary occlusion boundary classifier. Finally, the method of [100] is explored in which a discriminative learning step is used to learn the relation between low-level features and labeled occlusion boundaries for the CMU dataset.

Results for these methods as well as the proposed occlusion boundary detector are displayed in Fig. 5.9. Note that the proposed detector outperforms the other two non training-based methods by roughly 20%. In fact, our performance approaches that of [99], achieving within 5% of a method which was trained using the CMU dataset. The further work of [100] is able to increase performance further, but is not well-suited to novel video sequences, as will be demonstrated in the next section. For the two training-based methods, learning was performed using ground truth on half of the CMU dataset, while testing on the other half. This was repeated such that all sequences were tested.

5.5.2 Synthetic Sequence

To further assess the performance of the proposed method, we have constructed a 60 frame synthetic sequence with a dominant occlusion boundary. Textures for this sequence were obtained from the publicly available Vision Texture homepage [102]. The proposed method for occlusion boundary detection is compared with the methods of [93, 97, 100] in Fig. 5.10. Here, objective results are provided in Fig. 5.10(g) by computing the maximum F-score for each

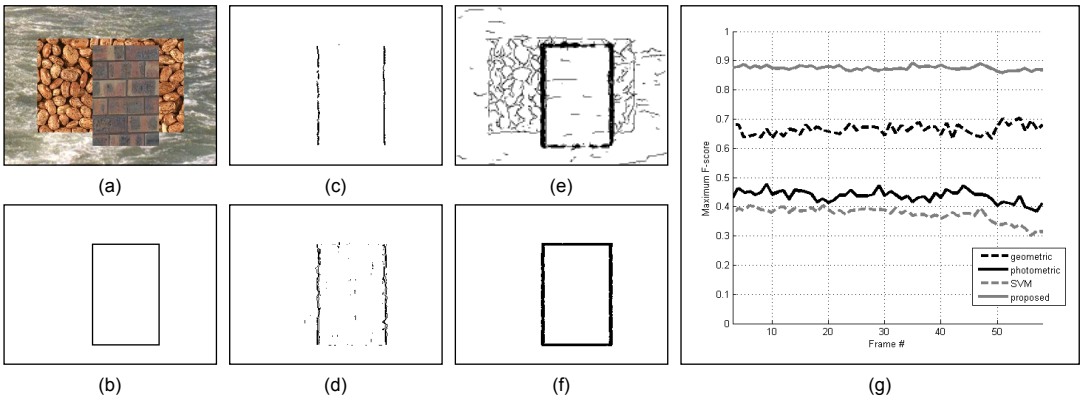


Figure 5.10: Results for synthetic sequence: (a) frame 30, (b) ground truth, (c) geometric method [97], (d) photometric method [93], (e) trained SVM [100], (f) proposed, (g) comparison of methods: F_{max} vs. frame index.

method at each frame of the sequence. It is clear from this simulation that the best performance is obtained by the proposed detector. It is also clear that while the training-based method may perform well when cross-validation is possible, it may not perform as well in general. In particular, a large number of false positive occlusion boundaries are detected due to edges in the texture maps.

Results for the two competing non-training based methods are achieved using a threshold of $\Theta = 80$ for the photometric method [93] and $\Theta = 11$ for the geometric method [97]. These parameters were selected because they produced the maximum F-score for each frame. We implemented the method of [100] using a Support Vector Classifier (LibSVM [103]) which is trained on the CMU database. The feature set for training is identical to that used in [100]. It is worth noting that the same optical flow field is used for both the proposed method and the feature set for the SVM classifier. In this way, it cannot be stated that one method of computing optical flow is superior to another, thus providing for an unfair comparison.

The SVM Classifier is trained using the following procedure. First, the optical flow field is computed between each adjacent pair of frames using the method due to [84]. Denote the motion field in the x and y directions as u and v, respectively. The gradients of the optical flow field are then $u_{dx}, u_{dy}, v_{dx}, v_{dy}$.

The feature set is comprised of the following five features:

- magnitude of the optical flow gradient

$$\sqrt{u_{dx}^2 + u_{dy}^2 + v_{dx}^2 + v_{dy}^2} \tag{5.12}$$

- motion estimation error

$$|F_t - F_{t+1}(x+u, y+v)| \tag{5.13}$$

- divergence of optical flow

$$|u_{dx} + v_{dy}| \tag{5.14}$$

- minimum eigenvalue of the spatio-temporal structure tensor (Eq. 5.15)

$$T = \begin{bmatrix} F_{dx}^2 & F_{dx}F_{dy} & F_{dx}F_{dt} \\ F_{dy}F_{dx} & F_{dy}^2 & F_{dy}F_{dt} \\ F_{dt}F_{dx} & F_{dt}F_{dy} & F_{dt}^2 \end{bmatrix} \tag{5.15}$$

  where F_{dt} is computed using the following formula: $uF_{dx} + vF_{dy} + F_{dt} = 0$

- edge intensity map using the pB edge detector [104]

A grid search is performed on the training set in order to obtain good training performance. Since the number of negative example far exceeds the number of positive examples, a random resampling procedure is employed for the negative examples to produce the training set. Each feature is normalized to be in the range $[0, 1]$. Training time is on the order of 2 hours for all 30 sequences in the CMU dataset using an Intel core-i7 965 processor with 12GB of RAM.

5.5.3 Performance And Occlusion Types

As the runtime of our proposed algorithm is linear with respect to the number of experts, it is important to determine what effect the expert count has on performance. To test this, the F-score of the **chair** sequence from the

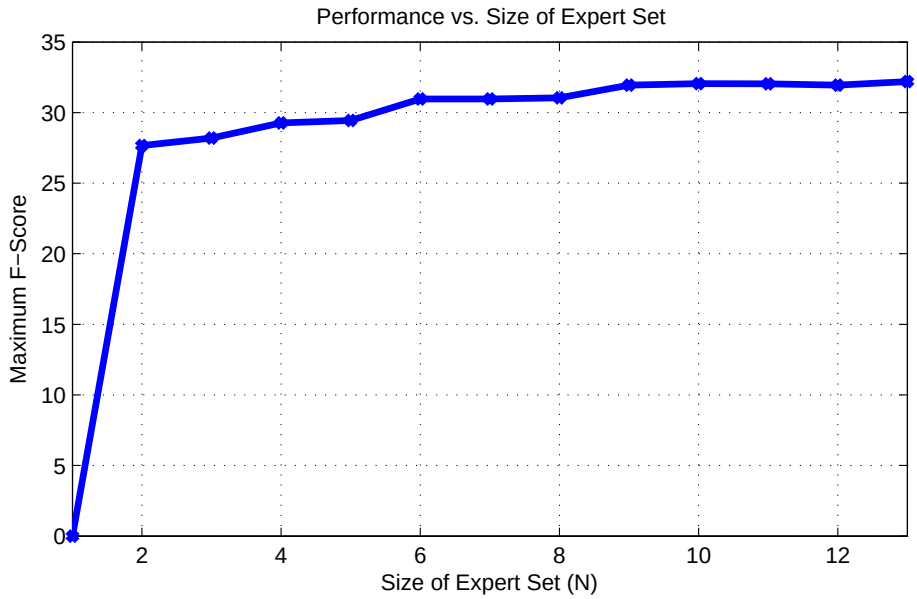


Figure 5.11: Proposed algorithm performance as a function of expert set size for **chair** sequence. We have selected a set of $N = 9$ experts, as selecting further occlusion types yields diminishing performance returns.

CMU dataset was examined with a variable size of the expert set. Results are displayed in Fig. 5.11. As N is increased, additional occlusion types are enabled in the order $\{\Gamma(\theta, 0), \Gamma(\theta, 1)\}$ for each angle θ. Angles are added in the order: $\theta = \left\{\frac{\pi}{2}, 0, \frac{\pi}{4}, \frac{3\pi}{4}, \frac{\pi}{8}, \frac{5\pi}{8}\right\}$. It is observed that the marginal performance increase above $N = 9$ is diminishing, thus the selection of $N = 9$ for all experiments conducted in the proposed work.

5.5.4 Occlusion Boundary Classification

In addition to occlusion boundary detection, the proposed algorithm can provide classification results based on the occlusion types specified. An example of this is demonstrated in Fig. 5.12 for the **rocking horse** sequence of the CMU dataset. Further classification can be performed based on the angle of the occlusion type, if desired. This technique is well suited for any application which will treat covering and uncovering occlusion types separately.

Figure 5.12: Occlusion boundary classification into two classes: (left) uncovering, (right) covering.

5.6 Conclusion

In this work, an efficient online learning approach to occlusion boundary detection has been presented. This method boasts a runtime linear with respect to the number of tracked particles and number of experts. In addition, the proposed algorithm does not require training, making it much simpler to use than competing methods, and much more suitable for novel video sequences when training data is unavailable. Despite the lack of training, the algorithm has demonstrated excellent performance both on the CMU Occlusion Dataset and on a synthetic video sequence. We outperform previous approaches which do not require a training stage, while approaching the performance of a fully-trained classifier. The efficiency of the proposed algorithm makes it well-suited as an off-the-shelf implementation which can be used on its own, or as a preprocessing step for other video processing tasks such as disparity estimation, motion vector

refinement and frame rate up-conversion.

Future work for this research may include an improved algorithm which detects object scale in addition to occlusion boundaries. In addition, NormalHedge [105] may be used to remove the learning rate parameter, η as well as to allow a larger set of experts.

5.7 Acknowledgements

The authors appreciate the assistance of Dr. Sargin in providing comparison data to our method using a trained SVM classifier based on his thesis work. Without his patience and guidance, the comparison to training-based methods would not have been possible.

The text of Chapter 5 is adapted from *An Online Learning Approach to Occlusion Boundary Detection*, Natan Jacobson, Yoav Freund and Truong Nguyen, accepted to *IEEE Transactions on Image Processing* in July of 2011. The dissertation author is the primary author of this publication.

Chapter 6

Conclusion

"Science is the belief in the ignorance of experts." - Richard Feynman

In this work, we have demonstrated the application of saliency to FRUC, yielding an improvement both in PSNR as well as subjective video quality. In addition, we proposed a new method for saliency detection using scale-space information. This further improved our saliency-based FRUC method. Finally, we investigated the effect of stereoscopic depth on saliency through an eye-tracking experiment. The data obtained in this experiment has been made publicly available online. Going forward, this data may lead to saliency detection methods for 3D scenes which properly handle the depth/disparity feature.

Outside of the saliency realm, we proposed an online-learning based method for occlusion boundary detection. Future work for this project could involve the combination of our method with a disparity estimation scheme to more robustly handle occlusions for 3D video. One area which would greatly benefit from this research is multiview synthesis, in which a stereo image pair is extrapolated to produce an arbitrary number of views from different vantage points. The quality of the extrapolated pictures would greatly increase.

Finally, it is our hope that research continues on saliency-based video processing methods. As saliency detection algorithms improve in accuracy and performance, they will become increasingly useful throughout the field.

Bibliography

[1] E. Shechtman, Y. Caspi, and M. Irani, "Space-time super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 531–545, April 2005.

[2] G. de Haan, P. Biezen, H. Huijgen, and O. Ojo, "True-motion estimation with 3-d recursive search block matching," *IEEE Trans. Circuits Syst. Video Tech,*, vol. 3, no. 5, pp. 368–379, 388, October 1993.

[3] J. Wang, D. Wang, and W. Zhang, "Temporal compensated motion estimation with simple block-based prediction," *IEEE Trans. Broadcasting*, vol. 49, no. 3, pp. 241–248, September 2003.

[4] M. Biswas and T. Nguyen, "A novel motion estimation algorithm using phase plane correlation for frame rate conversion," in *2002 Asilomar Conf. on Sig. Syst. Comp.*, vol. 1, November 2002, pp. 492–496 vol.1.

[5] A.-M. Huang and T. Nguyen, "Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation," *IEEE Trans. Image Process.*, vol. 18, no. 4, pp. 740–752, April 2009.

[6] S.-C. Tai, Y.-R. Chen, Z.-B. Huang, and C.-C. Wang, "A multi-pass true motion estimation scheme with motion vector propagation for frame rate up-conversion applications," *J. Disp. Tech.*, vol. 4, no. 2, pp. 188–197, June 2008.

[7] X. Gao, Y. Yang, and B. Xiao, "Adaptive frame rate up-conversion based on motion classification," *Signal Process.*, vol. 88, no. 12, pp. 2979–2988, 2008.

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

[9] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, pp. 1395–1407, 2006.

[10] W. Einhaeuser, T. N. Mundhenk, P. F. Baldi, C. Koch, and L. Itti, "A bottom-up model of spatial attention predicts human error patterns in rapid scene recognition," *J. of Vis.*, vol. 7, no. 10, pp. 1–13, July 2007.

[11] D. Gao and N. Vasconcelos, "Bottom-up saliency is a discriminant process," *2007 IEEE Int. Conf. Comp. Vis.*, pp. 1–6, October 2007.

[12] B. K. P. Horn and B. G. Schunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[13] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (darpa)," in *Proceedings of the 1981 DARPA Image Understanding Workshop*, April 1981, pp. 121–130.

[14] W. B. Thompson, "Combining motion and contrast for segmentation," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 2, no. 6, pp. 543–549, November 1980.

[15] T. Y. Tian and M. Shah, "Motion estimation and segmentation," *Machine Vision and Applications*, vol. 9, no. 1, pp. 32–42, January 1996.

[16] M. Chang, A. Tekalp, and M. Sezan, "Simultaneous motion estimation and segmentation," *IEEE Trans. Image Process.*, vol. 6, no. 9, pp. 1326–1333, September 1997.

[17] S. Khan and M. Shah, "Object based segmentation of video using color, motion and spatial information," *2001 IEEE Int. Conf. Comp. Vis. Pattern Recognition*, vol. 2, p. 746, 2001.

[18] D. Cremers and S. Soatto, "Variational space-time motion segmentation," vol. 2, October 2003, pp. 886–893.

[19] S. Fogel, "Segmentation-based method for motion-compensated frame interpolation," U.S. Patent 6 008 865, December 28, 1999.

[20] D. Gao, V. Mahadevan, and N. Vasconcelos, "On the plausibility of the discriminant center-surround hypothesis for visual saliency," *J. Vis.*, vol. 8, no. 7, pp. 1–18, June 2008.

[21] S. Soatto, G. Doretto, and Y. N. Wu, "Dynamic textures," *2001 IEEE Int. Conf. Comp. Vis. Pattern Recognition*, vol. 2, pp. 439–446, 2001.

[22] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *Int. J. Comput. Vis.*, vol. 43, no. 1, pp. 7–27, 2001.

[23] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 22, no. 8, pp. 888–905, August 2000.

[24] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using brightness and texture," 2002.

[25] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, no. 5802, pp. 91–97, March 1981.

[26] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2004, pp. 551–556.

[27] J. Youn, M.-T. Sun, and C.-W. Lin, "Motion vector refinement for high-performance transcoding," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 30–40, March 1999.

[28] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 32, pp. 171–177, 2009.

[29] M.-J. Chen, L.-G. Chen, and T.-D. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 4, no. 5, pp. 504–509, October 1994.

[30] X. Gao, C. Duanmu, and C. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 501–504, March 2000.

[31] A.-M. Huang and T. Nguyen, "A multistage motion vector processing method for motion-compensated frame interpolation," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 694–708, May 2008.

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

[33] "Methodology for the subjective assessment of the quality of television pictures," *ITU-R Recommendation BT.500-11*, 2002.

[34] G. Mori, X. Ren, A. Efros, and J. Malik, "Recovering human body configurations: combining segmentation and recognition," *2004 Int. Conf. Comp. Vis. Pattern Recognition*, vol. 2, pp. II–326–II–333 Vol.2, June 2004.

[35] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Adv. Neural Information Process. Syst.*, 2007, pp. 545–552. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.70.2254

[36] N. Bruce and J. Tsotsos, "Saliency based on information maximization," in *Adv. in Neural Inf. Process. Syst.*, no. 18, June 2006, pp. 155–162.

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

[38] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," jun. 2007, pp. 1 –8.

[39] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 185 –198, jan. 2010.

[40] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *J. Vis.*, vol. 9, no. 12, pp. 1–27, 2009.

[41] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *2010 IEEE Int. Conf. Comp. Vis. Pattern Recognition*, 2010. [Online]. Available: http://www.ee.technion.ac.il/\~{}ayellet/Ps/10-Saliency.pdf

[42] A. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 30, no. 5, pp. 909–926, May 2008.

[43] J. Huang and D. Mumford, "Statistics of natural images and models," in *1999 IEEE Int. Conf. Comp. Vis. Pattern Recognition*, vol. 1, 1999, pp. 541–547.

[44] P. Sand and S. Teller, "Particle video: Long-range motion estimation using point trajectories," in *2006 IEEE Int. Conf. Comp. Vis. Pattern Recognition*, vol. 2, 2006, pp. 2195–2202.

[45] P. he Tseng, R. Carmi, I. G. M. Cameron, D. P. Munoz, and L. Itti, "Quantifying center bias of observers in free viewing of dynamic natural scenes," *Journal of Vision*, vol. 9, no. 7, pp. 1–16, July 2009.

[46] B. W. Tatler, "The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions," *J. of Vis.*, vol. 7, no. 14, pp. 1–17, November 2007.

[47] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention." *Vision research*, vol. 40, no. 10-12, pp. 1489–1506, 2000. [Online]. Available: http://view.ncbi.nlm.nih.gov/pubmed/10788654

[48] N. Jacobson, Y. Lee, V. Mahadevan, N. Vasconcelos, and T. Q. Nguyen, "A novel approach to fruc using discriminant saliency and frame segmentation," *IEEE Trans. Image Process.*, November 2010.

[49] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 24, no. 5, pp. 603–619, August 2002. [Online]. Available: http://dx.doi.org/10.1109/34.1000236

[50] C. M. Christoudias, B. Georgescu, P. Meer, and C. M. Georgescu, "Synergism in low level vision," in *Int. Conf. Pattern Recognition*, 2002, pp. 150–155.

[51] Y.-L. Lee and T. Nguyen, "Fast one-pass motion compensated frame interpolation in high-definition video processing," nov. 2009, pp. 369 –372.

[52] D. Wang, A. Vincent, P. Blanchfield, and R. Klepko, "Motion-compensated frame rate up-conversion – part ii: New algorithms for frame interpolation," *IEEE Trans. Broadcasting*, vol. 56, no. 2, pp. 142–149, June 2010.

[53] J. M. Wolfe and T. S. Horowitz, "What attributes guide the deployment of visual attention and how do they do it?" *Nat. Rev. Neurosci.*, vol. 5, no. 6, pp. 495–501, 2004.

[54] L. Cormack, S. Stevenson, and C. Schor, "Interocular correlation, luminance contrast and cyclopean processing," *Vision Res.*, vol. 31, no. 12, pp. 2195–2207, 1991.

[55] D. Fleet, H. Wagner, and D. Heeger, "Neural encoding of binocular disparity: energy models, position shifts and phase shifts," *Vision Res.*, vol. 36, no. 12, pp. 1839–1857, June 1996.

[56] A. Anzai, I. Ohzawa, and R. Freeman, "Neural mechanisms for processing binocular information ii. complex cells," *J. Neurophysiol.*, vol. 82, no. 2, pp. 909–924, August 1999.

[57] B. Backus, D. Fleet, A. Parker, and D. Heeger, "Human cortical activity correlates with stereoscopic depth perception," *J. Neurophysiol.*, vol. 86, no. 4, pp. 2054–2068, October 2001.

[58] S. Georgieva, R. Peeters, H. Kolster, J. T. Todd, and G. A. Orban, "The processing of three-dimensional shape from disparity in the human brain," *J. Neurosci.*, vol. 29, no. 3, pp. 727–742, January 2009.

[59] I. Ohzawa, G. DeAngelis, and R. Freeman, "Stereoscopic depth discrimination in the visual cortex: neurons ideally suited as disparity detectors," *Science*, vol. 249, no. 4972, pp. 1037–1041, August 1990.

[60] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human Neurobiol.*, vol. 4, no. 4, pp. 219–227, 1985.

[61] Y. Liu, L. K. Cormack, and A. C. Bovik, "Luminance, disparity, and range statistics in 3d natural scenes," in *Human Vision and Electronic Imaging*, 2009.

[62] L. Jansen, S. Onat, and P. Knig, "Influence of disparity on fixation and saccades in free viewing of natural scenes," *J. of Vis.*, vol. 9, no. 1, pp. 1–19, January 2009.

[63] Y. Liu, L. K. Cormack, and A. C. Bovik, "Dichotomy between luminance and disparity features at binocular fixations," *J. Vis.*, vol. 10, no. 12, pp. 1–17, October 2010.

[64] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 171–177, 2010.

[65] A. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psych.*, vol. 12, no. 1, pp. 97–136, January 1980.

[66] K. Nakayama and G. Silverman, "Serial and parallel processing of visual feature conjunctions," *Nature*, vol. 320, no. 6059, pp. 264–265, March 1986.

[67] N. Ouerhani and H. Hugli, "Computing visual attention from scene depth," in *In Proc. 2000 Int. Conf. Pattern Recog.*, vol. 1, 2000, pp. 375 –378 vol.1.

[68] S. Jeong, S.-W. Ban, and M. Lee, "Stereo saliency map considering affective factors and selective motion analysis in a dynamic environment," *Neural Networks*, vol. 21, no. 10, pp. 1420–1430, 2008, iCONIP 2007.

[69] T. Jost, N. Ouerhani, R. v. Wartburg, R. Mri, and H. Hugli, "Contribution of depth to visual attention: comparison of a computer model and human behavior," in *In Proc. Early Cog. Vis. Workshop*, 2004.

[70] M. Aziz and B. Mertsching, "Fast depth saliency from stereo for region-based artificial visual attention," in *Adv. Concepts Intell. Vis. Syst.*, ser. Lecture Notes in Computer Science, J. Blanc-Talon, D. Bone, W. Philips, D. Popescu, and P. Scheunders, Eds. Springer Berlin / Heidelberg, 2010, vol. 6474, pp. 367–378.

[71] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comp. Vis.*, vol. 47, pp. 7–42, 2002.

[72] P. Hoyer and A. Hyvrinen, "Independent component analysis applied to feature extraction from colour and stereo images," *Net. Comp. Neural Syst.*, vol. 11, no. 3, pp. 191–210, August 2000.

[73] H. R. Filippini and M. S. Banks, "Limits of stereopsis explained by local cross-correlation," *J. Vs.*, vol. 9, no. 1, pp. 1–18, January 2009.

[74] D. Brainard, "The psychophysics toolbox," *Spat. Vis.*, vol. 10, no. 4, pp. 433–436, 1997.

[75] D. G. Pelli, "The videotoolbox software for visual psychophysics: transforming numbers into movies," *Spat. Vis.*, vol. 10, no. 4, pp. 437–442, 1997.

[76] M. Kleiner, D. Brainard, and D. Pelli, "What's new in psychtoolbox-3?" in *1997 European Conf. Vis. Percept.*, vol. 36, August 2007.

[77] F. Cornelissen, E. Peters, and J. Palmer, "The eyelink toolbox: eye tracking with matlab and the psychophysics toolbox," *Behav. Res. Methods. Instrum. Comput.*, vol. 34, no. 4, pp. 613–617, November 2002.

[78] U. Rajashekar, I. van der Linde, A. Bovik, and L. Cormack, "Foveated analysis of image features at fixations," *Vision Res.*, vol. 47, no. 25, pp. 3160–3172, September 2007.

[79] L. Olzak and P. Laurinen, "A framework for understanding center-surround interactions in apparent contrast and fine spatial discriminations," *J. Vis.*, vol. 5, no. 12, December 2005.

[80] S. Tajima, M. Watanabe, C. Imai, K. Ueno, T. Asamizuya, P. Sun, K. Tanaka, and K. Cheng, "Opposing effects of contextual surround in human early visual cortex revealed by functional magnetic resonance imaging

with continuously modulated visual stimuli," *J. Neurosci.*, vol. 30, no. 9, pp. 3264–3270, March 2010.

[81] P.-S. Toh and A. Forrest, "Occlusion detection in early vision," in *1990 Proc. Int. Conf. Comp. Vis.*, December 1990, pp. 126–132.

[82] K. Nakayama and S. Shimojo, "Da vinci stereopsis: Depth and subjective occluding contours from unpaired image points," *Vis. Research*, vol. 30, no. 11, pp. 1811–1825, 1990.

[83] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 8, no. 6, pp. 679–698, November 1986.

[84] M. J. Black and P. Anandan, "The robust estimation of multiple motions: parametric and piecewise-smooth flow fields," *Comput. Vis. Image Underst.*, vol. 63, no. 1, pp. 75–104, 1996.

[85] Y. Fu, A. Erdem, and A. tekalp, "Tracking visible boundary of objects using occlusion adaptive motion snake," *IEEE Trans. Image Process.*, vol. 9, no. 12, pp. 2051–2060, December 2000.

[86] P. Aguiar and J. Moura, "Figure-ground segmentation from occlusion," *IEEE Trans. Image Process.*, vol. 14, no. 8, pp. 1109–1124, August 2005.

[87] A. Ogale, C. Fermuller, and Y. Aloimonos, "Motion segmentation using occlusions," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 27, no. 6, pp. 988–992, June 2005.

[88] D. Feldman and D. Weinshall, "Motion segmentation and depth ordering using an occlusion detector," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 30, no. 7, pp. 1171–1185, July 2008.

[89] C. Zitnick and T. Kanade, "A cooperative algorithm for stereo matching and occlusion detection," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 22, no. 7, pp. 675–684, July 2000.

[90] O. Williams, M. Isard, and J. MacCormick, "Estimating disparity and occlusions in stereo video sequences," in *2005 IEEE Int. Conf. Comp. Vis. Pattern Recognition*, vol. 2, June 2005, pp. 250 – 257 vol. 2.

[91] Z. jie Zhu, Y. er Wang, G. yi Jiang, and Q. wen Zhang, "Novel scheme for disparity estimation and occlusion detection based on variable line-segment primitive," in *2006 Int. Conf. Signal Process.*, vol. 2, November 2006.

[92] J. Ullmann, "Analysis of 2-d occlusion by subtracting out," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 14, no. 4, pp. 485–489, April 1992.

[93] R. Depommier and E. Dubois, "Motion estimation with detection of occlusion areas," in *1992 IEEE Conf. Acoust., Speech, and Signal Process.*, vol. 3, Mar. 1992, pp. 269 –272 vol.3.

[94] M. J. Black and D. J. Fleet, "Probabilistic detection and tracking of motion boundaries," *Int. J. Comput. Vision*, vol. 38, no. 3, pp. 231–245, 2000.

[95] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *2001 Int. Conf. Comp. Vis.*, 2001, pp. 508–515.

[96] N. Apostoloff and A. Fitzgibbon, "Learning spatiotemporal t-junctions for occlusion detection," *2005 IEEE Int. Conf. Comp. Vis. Pattern Recognition*, vol. 2, pp. 553–559, 2005.

[97] S. Ince and J. Konrad, "Geometry-based estimation of occlusions from video frame pairs," in *2005 IEEE Int. Conf. Accoust. Speech and Signal Process.*, vol. 2, 2005, pp. ii/933 – ii/936 Vol. 2.

[98] A. Stein and M. Hebert, "Combining local appearance and motion cues for occlusion boundary detection," in *British Mach. Vis. Conf.*, September 2007.

[99] A. Stein, "Occlusion boundaries: Low-level detection to high-level reasoning," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2008.

[100] M. Sargin, L. Bertelli, B. Manjunath, and K. Rose, "Probabilistic occlusion boundary detection on spatio-temporal lattices," in *2009 IEEE Int. Conf. Comp. Vis.*, 2009, pp. 560–567.

[101] Y. Freund, R. E. Schapire, Y. Singer, and M. K. Warmuth, "Using and combining predictors that specialize," in *1997 Proc. ACM Symposium Theory of Computing*. New York, NY, USA: ACM, 1997, pp. 334–343.

[102] Vision texture homepage. [Online]. Available: http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html

[103] C. C. Chang and C. J. Lin, "Libsvm: a library for support vector machines," 2001.

[104] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 26, no. 5, pp. 530–549, May 2004.

[105] K. Chaudhuri, Y. Freund, and D. Hsu, "A parameter-free hedging algorithm," *Compt. Research Repository*, vol. abs/0903.2851, pp. 1–9, 2009.

[106] Y. Zhang, G. Jiang, M. Yu, and K. Chen, "Stereoscopic visual attention model for 3d video," in *Adv. in Multimedia Modeling*, ser. Lecture Notes in Computer Science, S. Boll, Q. Tian, L. Zhang, Z. Zhang, and Y.-P. Chen, Eds. Springer Berlin / Heidelberg, 2010, vol. 5916, pp. 314–324.