UNCERTAINTY ESTIMATION OF VISUAL ATTENTION MODELS USING SPATIOTEMPORAL ANALYSIS

A Thesis
Presented to
The Academic Faculty

by

Tariq Alshawi

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
August 2018

UNCERTAINTY ESTIMATION OF VISUAL ATTENTION MODELS USING SPATIOTEMPORAL ANALYSIS

Approved by:

Professor Ghassan AlRegib, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Biing Juang
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor David Anderson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Christopher Barnes
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Berdinus Bras
The George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: May 9, 2018

Dedicated to my family.

Without your constant love, support, and motivation,

I would not have made it this far.

ACKNOWLEDGEMENTS

I would like to express my special appreciation and thanks to my advisor Prof. Ghassan AlRegib for his continuous guidance, support, and invaluable advice. I would like to thank the members of my committee Prof. Juang, Prof. Anderson, Prof. Barnes, and Prof. Bras for providing helpful comments and suggestions for my research. I also thank Dr. Zhiling Long and Dr. Can Temel for their helpful suggestions and motivation during my journey. Special thanks to Motaz Alfarraj, Yazeed Alaudah, Muhammad Amir Shafiq, Min-Hung (Steve) Chen, and all my labmates for their help, suggestions, and friendship.

I'm deeply grateful to Prof. Alshebeili from King Suad University for his mentorship, friendship, and for providing me a wonderful research experience since the days of my undergraduate studies.

I'm greatly indebted to my family who supported me during the entire process, always believed in my potential, and acted as invaluable sources of encouragement, advice, tough-love, and sympathetic ears when I needed them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Computational video saliency detection attempts to highlight interesting regions or objects that might attract human attention when watching a video. Many video and image processing applications such as object segmentation, compression, and quality assessment utilize video saliency to efficiently reduce the dimensionality of the input videos and focus only on regions and objects that are interesting to human visual attention. However, there has been no explicit design of a saliency-based video processing framework nor an analysis of the saliency maps reliability.

In this dissertation, we focus on developing a systematic saliency-based video processing framework that is based on the study of the reliability or the confidence of the generated saliency maps. To develop such framework, we investigate uncertainty estimation within the context of human visual attention. We, first, analyze eye tracking data and video content to discover general patterns of human visual attention that can be used for uncertainty estimation including map consistency and scene motion. Based on such analysis, we introduce a procedure to estimate the correlation between eye-fixation data of a given video by using its corresponding optical flow map. We, also, utilize the eye-fixation correlation analysis to design an unsupervised video feature for uncertainty estimation based on local spatiotemporal neighborhoods.

We combine our findings from eye-fixation correlation study and the analysis of the unsupervised uncertainty estimation feature for video saliency with a data-driven approach to directly obtain a multi-factor estimation model that is both computationally-efficient and effective in estimating uncertainty in the application of video saliency detection.

CHAPTER I

INTRODUCTION

The widespread availability of image and video capturing devices has produced an unprecedented amount of multimedia content that keeps growing. On YouTube, for example, 80,000 hours of video content is being uploaded every day [84]. In 2015, Internet video surveillance traffic exceeded 500 petabytes [16]. With such volume, automated algorithms must process millions of videos to label, categorize, and sometimes summarize their content, which poses challenging research problems. At the same time, video resolution is increasing fast, while demand for efficient video processing, for hand-held devices, is growing rapidly. These two trends push researchers to adopt mechanisms to deal with both data dimensionality and volume. Dimensionality reduction, similarity metrics, and content analysis are examples of such trends.

Typically, algorithms are designed and tested in a *lab-controlled environment* with carefully chosen datasets that might not reflect real world applications. For example, algorithms can be fitted to a specific dataset regardless of the underlying phenomena. Additionally, comparing different algorithms requires a mechanism to evaluate *statistical significance*. A 0.1% improvement in performance might be meaningful given 0.01 statistical significance, while 1% might be meaningless if the statistical significance is larger that 1. Moreover, *application-specific parameters* such as quality of experience (QoE), risk assessment, and profit need to be considered in the overall performance of these algorithms when implemented in the real world. To address all these problems, an uncertainty-based framework is necessary. Such uncertainty can only be estimated by carefully analysing the algorithm itself and its assumption within the context of the application.

Uncertainty in its most basic form is the degree of belief or confidence in a specific information or event. Train delay, weather conditions, and wait time are uncertain as most events in our lives. We are inclined to believe that the train will not be delayed but we cannot say with 100% confidence that it will not. To represent such uncertainty, we typically use probability framework to encode our confidence or belief. Such representation might be qualitative like “It is unlikely that the train will be delayed today” or quantitative like “The probability of the train being delayed is 5%”. Probability provides a well-established framework that can model random phenomena and extract useful information. However, it might be useful, or even necessary, to model uncertainty as a lack of knowledge rather than a random variability. For example, our uncertainty in the train delay could be reduced if we check a transportation website that tracks trains. The latter type of uncertainty, caused by lack of knowledge, is often called *epistemic uncertainty*, while the former, caused by variability or randomness, is called *aleatory uncertainty* [54].

In the context of image and video processing, many applications can benefit from quantifying and evaluating uncertainty. For example, algorithms modeled based on human visual attention try to overcome the shortage of information about the human vision system (HVS) by validating their assumptions based on eye tracking data. These algorithms typically rely on Saliency to enhance the accuracy and efficiency of image and video processing algorithms. By relying on Saliency, the researchers goal is to approach, or exceed, HVS capabilities of handling significant amount of visual data encountered every day, with minimal computational power. HVS can achieve such efficiency by selectively attending to important (salient) details and suppressing redundant or irrelevant information [45]. For example, it has been shown that HVS can spot and recognize an object as an animal in $120ms$ but cannot identify its type until further processing [91]. Indeed, Human visual attention modeling and understanding [45, 40, 12, 34, 95, 41, 61] has been shown to be effective in analyzing big visual data as well as improving the computation efficiency of visual data processing. Numerous applications for saliency-based algorithms have been proposed and currently investigated, such as object detection and recognition [74], scene understanding [7], and multimedia summarization [71].

The majority of existing research efforts focus on computational saliency models [86, 53, 43, 64, 60, 55]. However, less attention has been given to evaluating the generated saliency maps [24, 4, 3]. The validity of such maps is crucial for integrating visual attention in various image and video processing applications. It is a common practice to consider the validity of a saliency detection model, at every pixel, to be directly related to the overall performance of the detection model on image and video datasets. In other words, a saliency detection model is, first, evaluated using typical saliency detection datasets such as CRCNS [47], MSRA [59], MIT [49], and SAVAM [29]. Then, algorithms that detect salient regions effectively, according to a predefined ground truth in the dataset, are assumed to perform well when used in various applications. However, such saliency detectors might fail to produce reliable results in certain contexts or situations, despite their superior performance in other contexts.

The objective of this dissertation is two folds (i) analyze uncertainty in computational video saliency and (ii) design an effective uncertainty estimation algorithm tailored for video saliency detection. More specifically, we analyze eye tracking data and video content to discover general patterns of human visual attention that can be used for uncertainty estimation including: map consistency, and scene motion. Based on such analysis, we design an uncertainty estimation algorithm and show its effectiveness in the application of video saliency detection.

The rest of this dissertation is organized as follows: in Chapter 2, we provide an introduction to uncertainty concepts and common frameworks for its representation, introductory background on computational visual saliency, a literature survey on eye fixation analysis, and state-of-the-art uncertainty analysis in the context of image and video processing. Chapter 3 presents analysis of eye-fixation maps and their self-correlation. In Chapter 4, we show how the self-correlation of eye-fixation maps can be predicted by relying on motion cues. Chapter 5 shows the design of video feature for uncertainty analysis and uncertainty estimation using spatiotemporal cues. Chapter 6 shows a multi-factor uncertainty estimation algorithm that utilizes features from the saliency map as well as video frames. Finally, conclusions and future work are presented in Chapter 7.

CHAPTER II

BACKGROUND AND LITERATURE SURVEY

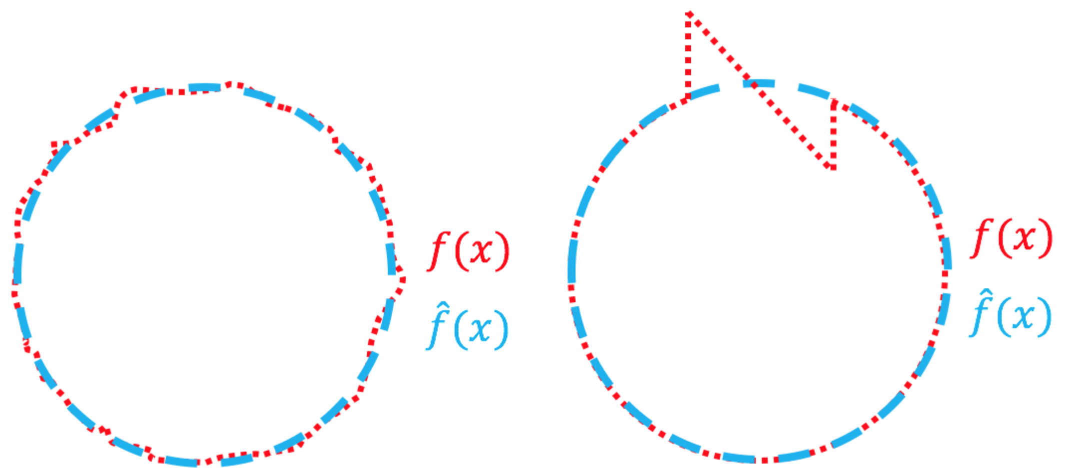
2.1 Uncertainty Representation

Uncertainty is prevalent in every event we encounter and every decision we make [57]. Whether there is a flight delay or a traffic jam on the highway is often uncertain and imprecise. Nevertheless, we have to make do with this incomplete knowledge to be productive and move our lives forward. Uncertainty, also called reliability or confidence, encapsulates our belief of the mismatch between what we know and what we expect; the more mismatch there is the more uncertain we are about this knowledge. Scientifically representing uncertainty is crucial to an informed decision making process and to mitigate possible risks in such process [92]. Before diving into mathematical formulation of uncertainty representation, we ought to explore a precise definition of what is *uncertainty*.

In the literature, the term *uncertainty* is used in different ways to describe, often, different things. However, in our work, we use the term *uncertainty* to describe the phenomena of imperfect knowledge about the state of an event or the outcome of a process, rather than a specific measure or a metric [37]. Causes of such imperfect knowledge are classified into two types: *aleatory uncertainty*, and *epistemic uncertainty* [54]. The first type typically refers to the inherent variability or randomness of an event. For example, in situations where complete knowledge of the forces that determine the results of coin-flipping are inaccessible, the outcome of flipping a coin is treated as random and additional knowledge cannot eliminate this uncertainty. Thus, aleatory uncertainty is often called *irreducible uncertainty* or *objective uncertainty* since the source of uncertainty is the randomness of the event itself rather than the quantification process [68]. For visual representation of aleatory uncertainty, we show in Figure 1(a) $f(x)$ an example of a function that generates values that falls on a circle circumference and an estimation model $\hat{f}(x)$. Since $f(x)$ is inherently random, there exist errors between true values of $f(x)$, shown in red dots, and the estimation values of $\hat{f}(x)$, shown in blue dashes. On the other hand, epistemic uncertainty typically refers to the lack of knowledge to characterize a process or an event. For example, measuring the dimensions of an object often includes epistemic uncertainty because more precise measuring devices can reduce the gap in knowledge of the true dimensions of the object. Thus, this type of uncertainty is called *reducible uncertainty* or *subjective uncertainty* since the source of uncertainty is the measurement process [68]. Again for visual representation, we show in Figure 1(b) $f(x)$ and $\hat{f}(x)$ under epistemic uncertainty. Unlike the first case, in most values the estimate $\hat{f}(x)$ closely matches $f(x)$, however, in some range of $f(x)$, $\hat{f}(x)$ does not match $f(x)$, due to missing knowledge in the estimation model $\hat{f}(x)$. It is important to note here that this distinction is not absolute and depends on the situation at hand. Going back to the two examples earlier, one can argue that a precise modeling of a controlled coin-flipping experiment can deterministically predict the outcome and eliminate its variability. On the other hand, measuring the dimensions of an object can be made more precise all the way to atomic or sub-atomic accuracy but it will always be bounded by technology. However, given a specific situation and a precise description of the present and accessible knowledge, such distinction between uncertainty causes is often consistent and useful to inform further analysis [97].

2.1.1 Probability Theory

There are many frameworks, in the literature, to represent and propagate uncertainty. To illustrate the motivation and assumptions of each framework, we use a simple example of an event A; a measurable subset of the sample space Ω, that contains values



(a) Example of Aleatory Uncertainty (b) Example of Epistemic Uncertainty

Figure 1: Visual Representation of Uncertainty Types

of a given variable Y. Among the various uncertainty frameworks, probability is by far the most used framework to represent uncertainty [68]. In probability framework, uncertainty about the value of Y is encoded into a measure function, i.e. probability distribution function (PDF), that maps every value $y \in \Omega$ to a single value $p_Y(y)$ between 0 and 1 [57]. Figure 2 shows a visual representation of Probability Theory, which oversimplifies the framework but is useful to compare between different uncertainty representation frameworks. The probability of event $A \subseteq \Omega$ to occur ($P(A)$) is equal to sum (or integral) of all Singleton probabilities of the values $y \in A$. Within probability framework, specifying $P(A)$ results in specifying $P(\overline{A})$ as well, since the two probabilities has to sum to 1. This is known as self-duality property; the specification of the likelihood of an event implies the likelihood of its complement event. This stems from the fact that the sample space Ω is well defined and the total probability assigned to Ω, $P(\Omega)$, equals one. Therefore, allocating some of the probability weight to the even A, inadvertently, allocates the rest to the complement event $\bar{A}$.

In the literature, there are three main concerns regarding the use of probability framework to represent uncertainty [97, 68, 54, 37]. First, utilizing probability

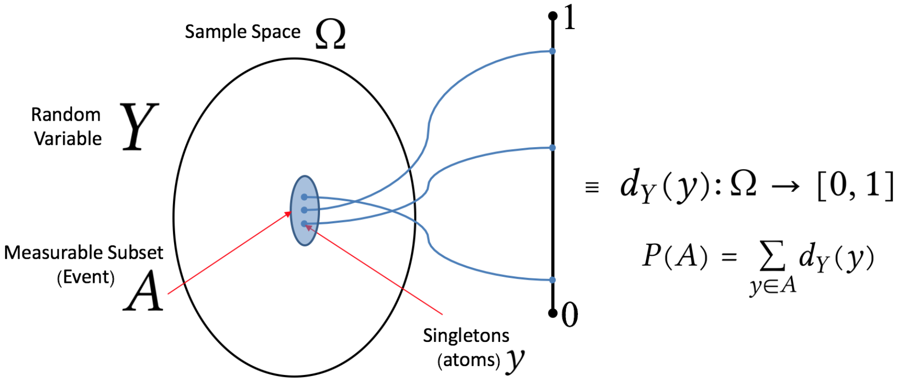

Sample Space Ω
Random Variable Y
Measurable Subset (Event) A
Singletons (atoms) y
1
0
$\equiv d_Y(y) : \Omega \to [0,1]$
$P(A) = \sum_{y \in A} d_Y(y)$

Figure 2: Visual Representation of Probability Theory

framework requires additional assumptions that might not be strongly supported by data. For example, if the only information about Y is its range, we have to assume its mean and variance to build its PDF. Even the so-called *uninformed prior*; i.e. uniform PDF, assumes that the mean of Y is at the center of the interval. Second, representation of uncertainty using probability does not differentiate between aleatory and epistemic uncertainties. In fact, probability framework attributes the cause of uncertainty based on the probability interpretation itself. In the frequentist view, probability is the *chance* of event A to occur given an infinite number of trails. Using this interpretation, uncertainty is attributed mainly to the variation of the outcome rather than the lack of knowledge. On the the other hand, subjective (Bayesian) interpretation views probability as an expression of purely epistemic uncertainty. In both interpretations, quantifying the part of the resulting variation due to epistemic (reducible) uncertainty is not possible because the framework does not distinguish between stochastic variation and ignorance. The third concern regarding the use of probability framework to represent uncertainty is that probability abstracts uncertainty into a single precise value regardless of the available data. For example, given the available data, we are inclined to believe $P(A)$ is somewhere between 0.2 to 0.4, however, such knowledge cannot be accommodated in probability framework and $P(A)$ must be assigned a single precise value, say 0.3.

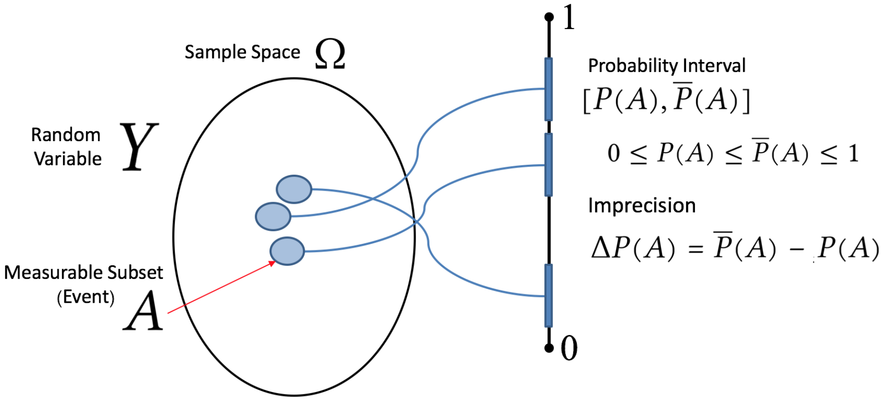
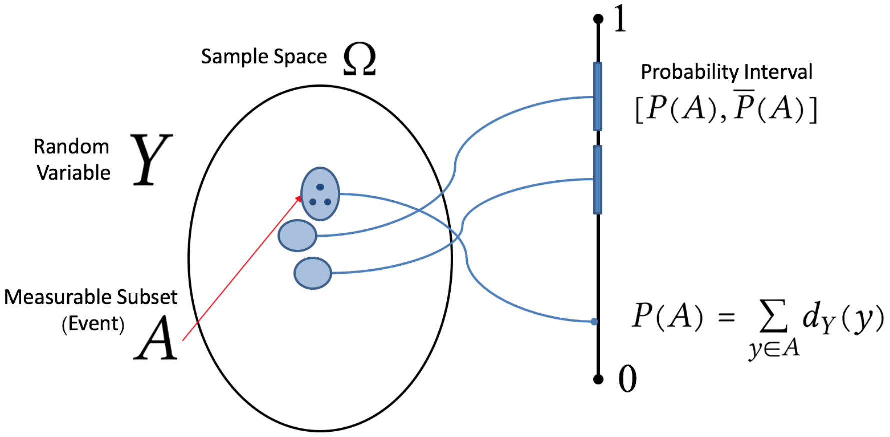

1
Sample Space Ω
Probability Interval
$[P(A), \overline{P}(A)]$
Random Variable Y
$0 \leq P(A) \leq \overline{P}(A) \leq 1$
Imprecision
$\Delta P(A) = \overline{P}(A) - P(A)$
Measurable Subset (Event) A
0

Figure 3: Visual Representation of Interval Analysis

2.1.2 Interval Analysis

To address concerns regarding probability framework representation of uncertainty, other frameworks have been proposed [85, 78, 20, 19, 25]. Interval analysis, or imprecise probability, attempts to overcome some of the shortcomings of probability framework by relaxing its strict precision and required assumptions, such as distribution, and correlation, by proposing a probability interval instead of probability value [85], as shown in the visual representation in Figure 3. For an event A, we can represent the uncertainty of the occurrence of A by an interval $[P(A), \overline{P}(A)]$ where $\overline{P}(A)$ is the upper probability and $P(A)$ is the lower probability. The size of the such interval is known as *imprecision* $\Delta P(A)$, which is used to represent the epistemic uncertainty in this framework. Unlike in probability theory, interval analysis starts by assigning probability intervals directly to events and does not define singletons events or atoms, which makes interval analysis able to incorporate subjective opinions rather easy, at least compared to probability theory.

Imprecise probability framework employs *interval arithmetic* to compound the uncertainty of multiple variables, which generates a probability interval at the end, making the results simple and easy to interpret [69]. Additionally, imprecise probability can represent aleatory and epistemic uncertainty separately making it more



1
Sample Space Ω
Probability Interval
$[\underline{P}(A), \overline{P}(A)]$
Random Variable Y
Measurable Subset (Event) A
$P(A) = \sum_{y \in A} d_Y(y)$
0

Figure 4: Visual Representation of Probability Bound Analysis

specific than probability theory in identifying the source of uncertainty. However, the imprecision of interval analysis is also a disadvantage, because the framework is incapable of taking into account information like distributions, and correlations or dependencies if such information is available. Additionally, the rigorous nature of interval arithmetic makes the computed interval analysis results grow in imprecision vary quickly, making the computed results less-specific and in-turn less useful.

2.1.3 Probability Bound Analysis

To address the problems of probability framework and imprecise probability, probability bound analysis combines both frameworks to create a precise and flexible representation framework that adapts to the available information [25]. In probability bound analysis, the parameters which their aleatory uncertainty can be estimated accurately are formulated using traditional probability framework. On the other hand, the parameters which their aleatory uncertainty cannot be estimated accurately, imprecise probability framework is employed, as shown in Figure 4.

Propagating uncertainty, within probability bound framework, results in generating what is called *probability boxes* where two cumulative distribution functions (CDF) enclose a region of all possible CDFs of a variable, an example is shown in

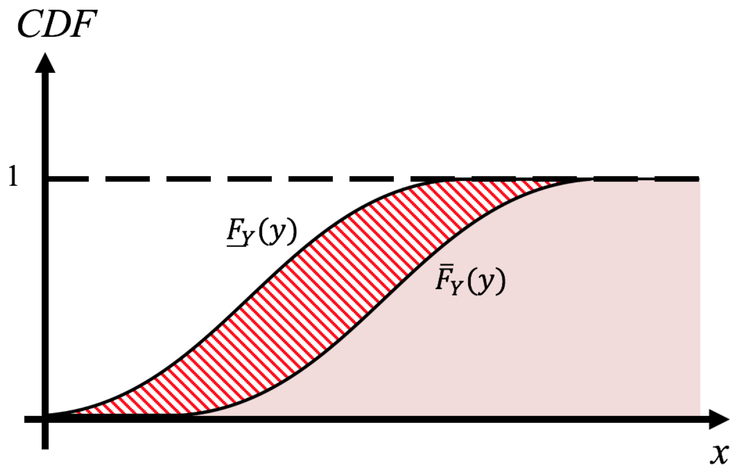


Figure 5: A Toy Example of Probability Box generated by Probability Bound Analysis

the red striped region between $\underline{F}_Y(y)$ and $\bar{F}_Y(y)$ in Figure 5 one of which will be the true CDF. Naturally, probability bound analysis inherits the advantages of both probability theory and interval analysis by enabling precise representation when data is available and the flexibility of interval assignment otherwise. Additionally, probability bound analysis guarantees bounded answers, which gets narrower with better empirical information [54]. However, similar to interval analysis, probability bound analysis does not show the most likely values within probability boxes, also, these bounds might not be the tightest possible bound given the available information.

2.1.4 Evidence Theory

Evidence theory allows for representation of both aleatory and epistemic uncertainty, and produces consistent representation regardless of the level of details in the available information [19, 78]. Specifically, evidence theory is well suited for representing *incomplete* information. Using fuzzy measures, *belief* (Bel) and *plausibility* (Pl); evidence theory encodes epistemic uncertainty about variable Y by distributing the mass of evidence, *basic probability assignment* (bpa), on the subsets of the power set $\mathcal{P}(Y)$ of the space of all possible values of Y called *Universe of Discourse* (U_Y), as

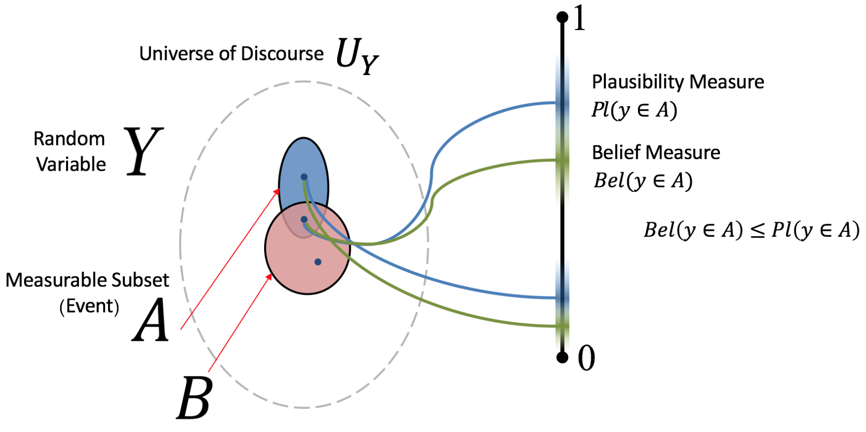

Universe of Discourse U_Y
Random Variable Y
Measurable Subset (Event) A
B
1
0
Plausibility Measure $Pl(y \in A)$
Belief Measure $Bel(y \in A)$
$Bel(y \in A) \leq Pl(y \in A)$

Figure 6: Visual Representation of Evidence Theory

shown in Figure 6. Unlike probability theory and its variations, universe of discourse U_Y in evidence theory is not necessarily well defined. In other words, when formulating a problem using probability-based framework, the sample space Ω must list all possible values of the random variable Y, which leads to two results. The first result is that the probability of the sample space $P(\Omega)$ is one, which is not always the case in evidence theory since the probability mass is distributed over the power set $\mathcal{P}(Y)$ and not directly to the universe of discourse U_Y. The second result of a well defined sample space Ω is the self-duality property; $P(A) + P(\bar{A}) = 1$. Evidence theory on the other hand gives the possibility of unknown event which might be undiscoverable using the available information. Therefore, the two fuzzy measures used in evidence theory, Bel and Pl, can account for *ignorance* in the available information by having $Bel(A) + Bel(\bar{A}) \leq 1$ and $Pl(A) + Pl(\bar{A}) \geq 1$

In evidence theory, aggregating multiple sources of evidence is done using *Dempster's rule* [97]. However, researchers have shown that some results generated using *Dempster's rule* are counter-intuitive especially in situation with highly conflicting pieces of evidence [94]. Nevertheless, evidence theory provides a rigours framework for decision making based on available information regardless of its completeness.

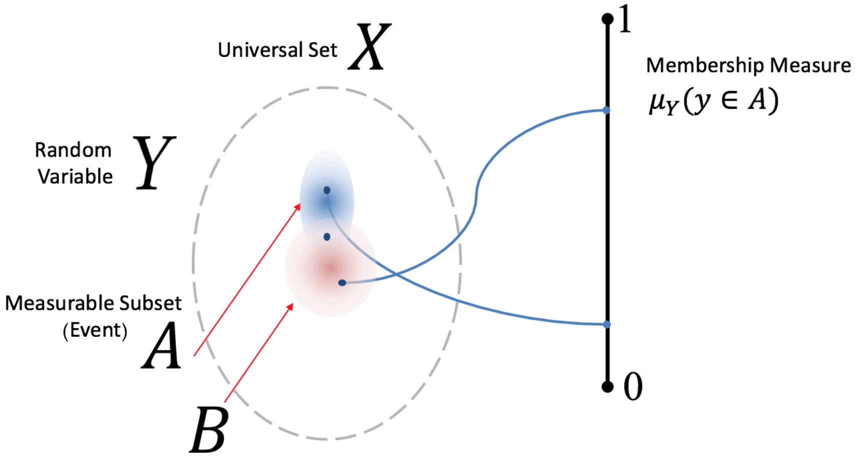


Figure 7: Visual Representation of Fuzzy Sets Theory

2.1.5 Fuzzy Sets Theory

Fuzzy sets theory attempt to represent uncertainty in the problem formulation itself rather than the random variable under study [93]. For example, we can formulate the problem of estimating the room temperature as random variable Y and every possible value for Y, in the sample space Ω, as y. The uncertainty in estimating Y comes from our incomplete knowledge about its exact value, but we are certain that $Y = y_{true}$. Now, by reformulating the problem to be identifying the state of the room temperature to be either *cold*, *warm*, or *hot*, which requires the representation of uncertainty in the sets themselves. In fuzzy sets theory, the sets boundaries are not well defined while the membership measure (μ_A) is precise, as shown in Figure 7, unlike the case in evidence theory. For example, room temperature can be descried as *warm* or *hot*, however, there is no clear boundary between the two descriptions (sets). So, the room temperature can be precise but the set it belongs to is fuzzy. A higher $\mu_A(y)$ indicates that y is more likely to be a member of the set A and vice versa. This gives a rise to *fuzzy fusion* rules to aggregate information from multiple sources[52].

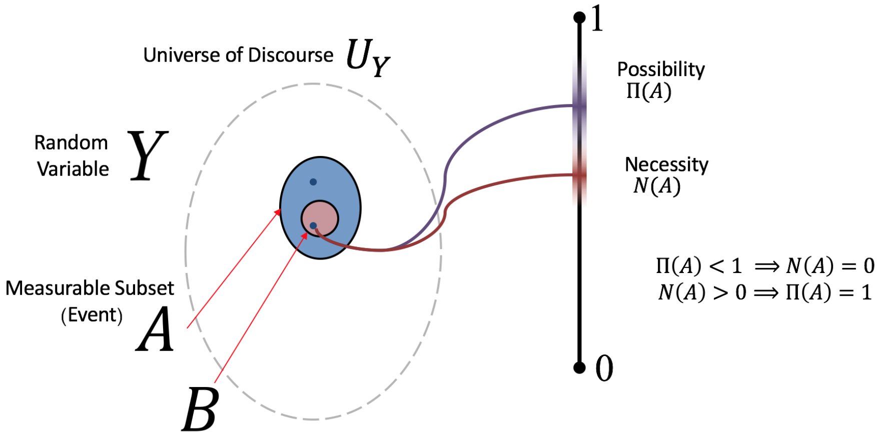

1
Universe of Discourse U_Y
Possibility $\Pi(A)$
Random Variable Y
Necessity $N(A)$
$\Pi(A) < 1 \Rightarrow N(A) = 0$
$N(A) > 0 \Rightarrow \Pi(A) = 1$
Measurable Subset (Event) A
0
B

Figure 8: Visual Representation of Possibility Theory

2.1.6 Possibility Theory

Another framework for representing uncertainty is possibility theory [20]. Using the so-called *possibility distribution*, possibility theory enables epistemic uncertainty representation using a family of probability distributions [97]. Possibility theory replaces the fuzzy measure used in evidence theory, *belief* and *plausibility*, and derives two new measures, *necessity* (N) and *possibility* (Π), that defines possibility distribution. In fact, possibility theory is considered a special case of evidence theory, where sets that have nonzero probability mass assignment are nested sets of each other [97]. Due to this nesting, possibility theory computes what is known as *consonant* body of evidence where $Bel(A \cap B) = min[Bel(A), Bel(B)]$ and $Pl(A \cup B) = max[Pl(A), Pl(B)]$, which is refered to as N and Π, respectively.

2.2 Computational Visual Saliency

Human visual attention modeling has been an active research area in the last few decades. Psychologists, computational neurophysiologists, and computer vision scientists have all contributed to this field. Thus, there are a lot of aspects to human visual attention research, however, we only focus on computational visual saliency in this thesis. Typically, the term *attention* covers the factors influencing the mechanisms of HVS to selectively attend to specific details, often called *selection mechanisms*. Generally, attention research is divided into bottom-up attention, driven by the scene, and top-down attention, driven by expectation or the task [9]. The term *saliency* is often used to describe some part of the scene that stands out relative to its neighboring parts, usually in the context of bottom-up models. In these models, features such as color, intensity, structure, surprise, and motion, often called low-level features, are used to predict the saliency of an image or a video [9]. On the other hand, top-down models rely on high-level features and task-oriented analysis such as HVS ability to tune-in to red striped shirts when searching for Waldo, for example [17]. Visual attention is, also, specific to the media used in the experiments. Attention patterns of subjects examining images, static stimuli, are inherently different than patterns associated with videos, dynamic stimuli [67]. Video sequences with standard frame rate, $24-30$ frames per second, are perceived as a single scene rather a collection of images. Each frame has less than 0.04 seconds to be perceived, thus only certain aspects such as motion and flicker stand out at this high rate [70, 67]. Additionally, in visual attention experiments, subjects are shown images for a short period of time to only allow low-level features to influence the eye-fixation [47]. Such practice produces the so-called *center bias*, where the majority of results show salient regions around the center of images, regardless of the content [8]. Videos, on the other hand, have the same bias only in the first few frames because subjects are able to perceive the scene in that time [5].

2.2.1 Theories

Many psychophysical theories have been proposed to explain the mechanism and process of attention in HVS. Here, we briefly describe the five of the most popular theories [36]:

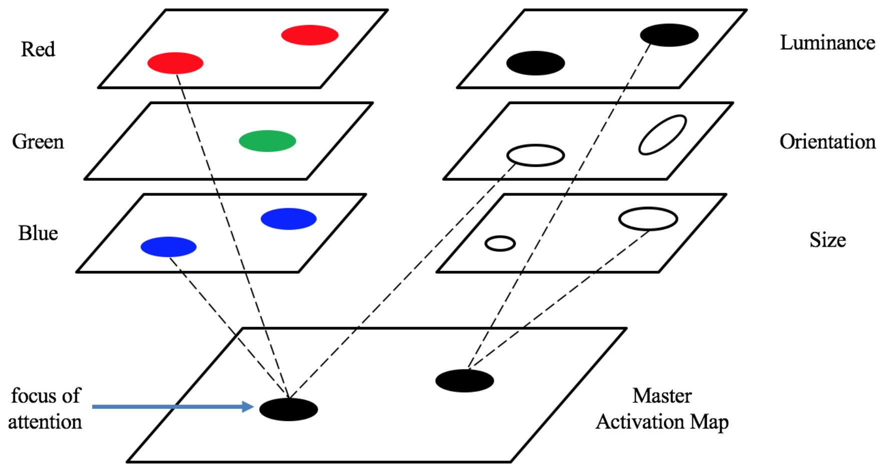


Figure 9: Treisman Feature Integration Model [36].

2.2.1.1 Feature Integration Theory (1980)

Treisman et al. proposed a feature-activation model to explain visual attention in HVS [83]. In their model, experimentally-verified preattentive features are processed on parallel to compute feature-specific activation maps as shown in Figure 9. In this model, salient objects tend to activate more feature maps than others in the image. Further experimental analysis of their model revealed that the amount of attention is relative to the *target-nontarget* difference [82]. For example, a long vertical line can detected immediately among a group of short lines, but a medium-length line may take longer to see. Interestingly, experimental results shows that attention to some features is asymmetric. For example, a sloped line in a sea of vertical lines can be detected preattentively, but a vertical line in a sea of sloped line is more difficult to detect.

2.2.1.2 Textons (1981)

Julesz proposed that HVS preattentively detects three kinds of features, called *textons*; Elongated blobs (includes lines, rectangular, or ellipses), terminators (ends of line segments), and crossings of line segments [51]. In his theory, textons represent

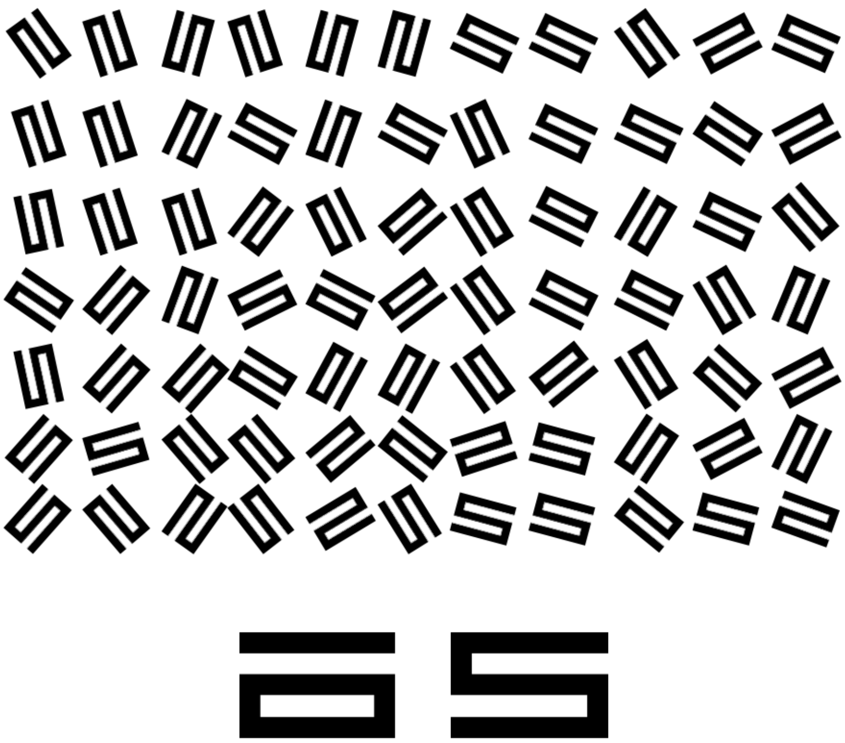
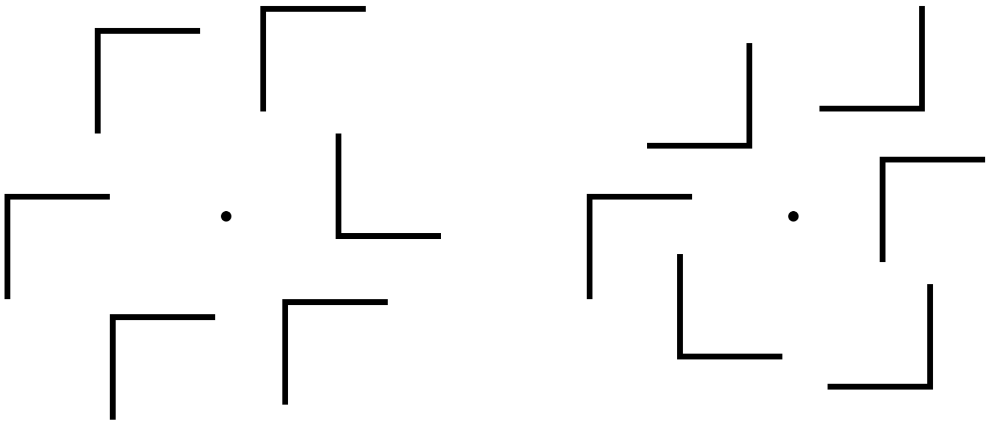


Figure 10: An Example of Julesz's texton features [51].

the basis set for perception, and the variation in order statistics of these textons determine their saliency [50]. The experiments show that first order statistic, such as contrast, are highly salient. While higher order statistics like orientation and regularity (2nd order), and curvature (3rd order) are less salient and needs further processing by the HVS. To verify his hypothesis, Julesz used texture images, like the one shown in Figure 10, and showed that even though two textons appear different in isolation, they cannot be distinguished preattentively when shown in a group.

2.2.1.3 Similarity (1989)

Quinlan et al. investigated the factors that affect conjunction search [72]. They hypothesized that the search time may depend of the amount of information required to identify a target and the similarity between a target and its distractors. Duncan et al. extended this work by quantifying the factors that affect search time



(a) Homogeneous; high N-N similarity (b) Heterogeneous; low N-N similarity

Figure 11: An Example that shows the effect of N-N similarity on search efficiency [21].

and they found that two important criteria determine search time; Target-Nontarget (T-N) similarity, and Nontarget-Nontarget (N-N) similarity [21]. They showed by experiment that search time increase as T-N similarity increase or as N-N similarity decrease. Interestingly, their experiments show that when either T-N similarity is high or N-N similarity is low, changing the other factor has little effect on search time. For example, experimental results show that a pattern such as the one shown in Figure 11(a) are easier to identify the target (blob) than patterns like the Figure 11(b). Form these experiments, Duncan et al. proposed a three-step theory of visual selection: (1)Segmentation of the visual field into homogeneous structural units, (2) Structural units that are closer to the target template are granted more resources of HVS such as access to visual short-term memory, and (3) Groups of structural units that are similar get efficiently rejected if one or more have low correlation with the target template [21].

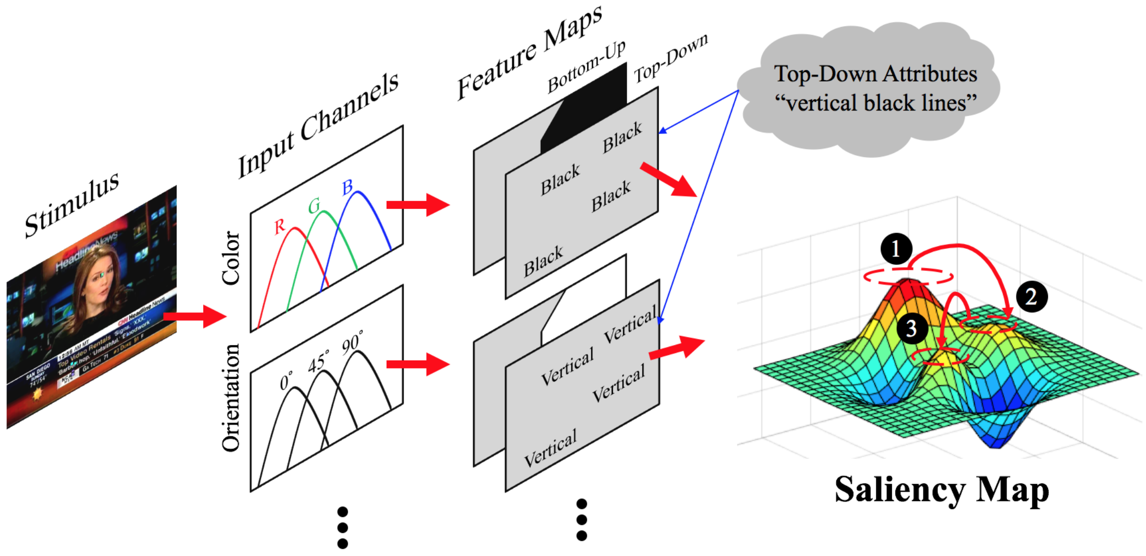


Figure 12: Illustration of Guided Search Theory [89].

2.2.1.4 Guided Search (1994)

Wolf et al., first proposed the theory of guided search in [90] and years later formalized it in [89]. Guided search theory was the first attempt to incorporate the goals of the viewer *Top-Down* mechanism into the model visual attention. As shown in Figure 12, the theory divides the visual stimulus into primary feature maps such as color and orientation. Afterwards, the viewer filters out these feature maps by a combination of bottom-up saliency cues and top-down objectives. Figure 12, shows an example of such mechanism where the viewer has the objective to find the attributes *"vertical black lines"* which are reflected in the Top-Down map search. Finally, a saliency map emerges by combining the Top-Down and Bottom-Up maps, shown on the far right of Figure 12. The attention of the viewer lands on the global maxima activation, first, and then jumps from one local maxima to another in decreasing order of activation value.

2.2.1.5 Boolean Maps (2007)

More recently, Huang et al. proposed a new model of low-level vision by dividing the visual search task into two steps: selection and access [44]. The theory suggests that during the visual search process, the viewer starts by selecting elements that poses a property relevant to the immediate task, then, the visual system can access certain properties of the selected elements. By doing so, the visual system works by selecting according to specific feature then excluding irrelevant objects that do not match the other desired features of the target object, hence the name *Boolean* maps theory. An example is shown in Figure 13. The theory suggests that the viewer, when examining the stimulus Figure 13(a), first selects all the elements with the label *Red*, Figure 13(b), and then creates a desired feature map with the elements with label *Vertical*, Figure 13(c), finally, target objects with both labels *Red* and *Vertical* are identified by finishing the boolean intersection between the two maps (b) and (c).

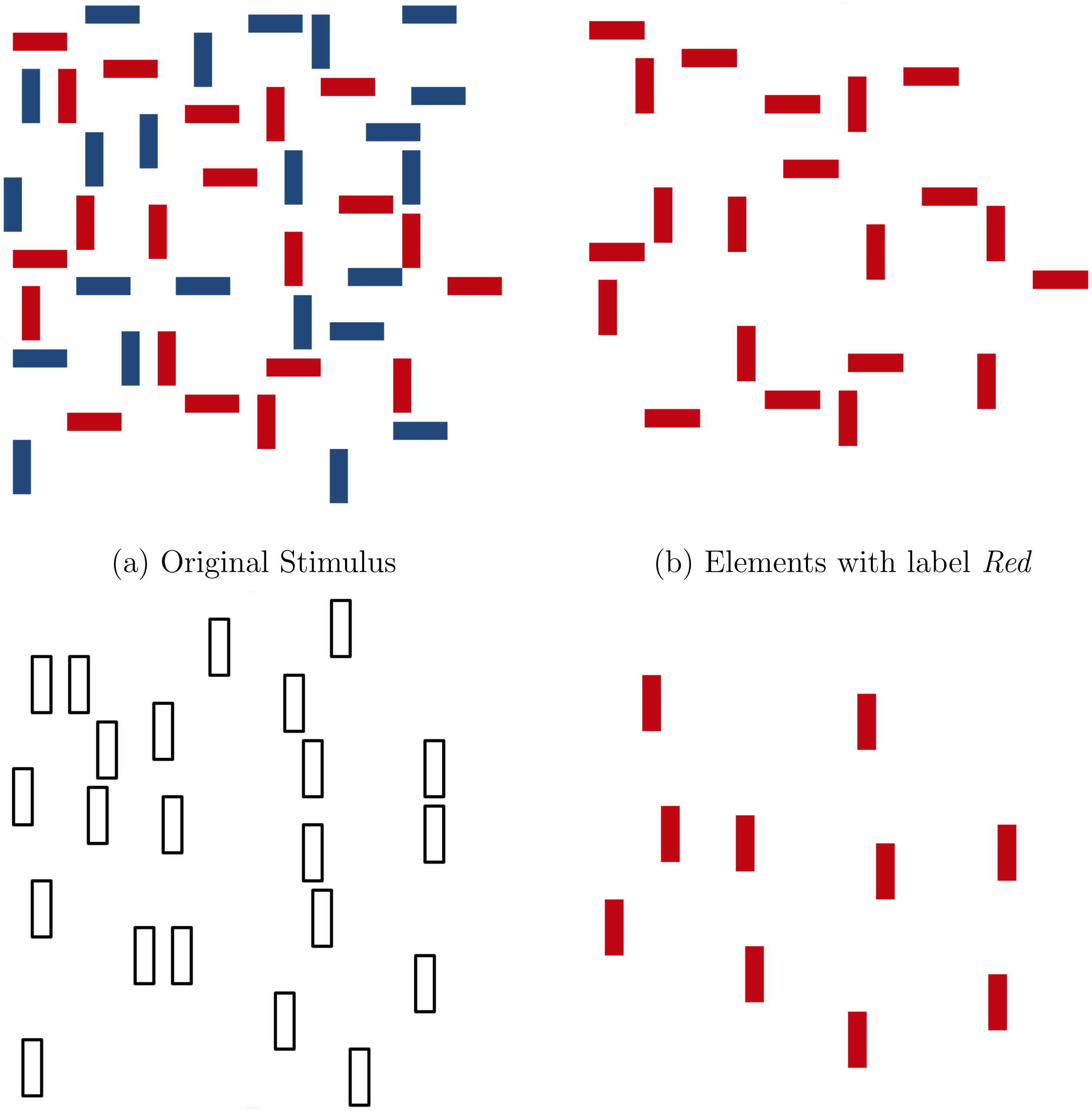
2.2.2 Models

2.2.2.1 Static and Space-time Visual Saliency Detection by Self-Resemblance (STSR)

Seo et al. proposed using local steering kernels (LSK) as features for saliency detection [77]. The proposed algorithm, illustrated in Figure 14, computes feature vectors $\mathbf{f}_i$, at pixel x_i, by vectorizing a normalized version of the local steering kernel function $\boldsymbol{K}(\mathrm{x}_l - \mathrm{x}_i)$, which is computed according to:

$$\boldsymbol{K}(\mathrm{x}_l - \mathrm{x}_i) = \frac{\sqrt{det(\boldsymbol{C}_l)}}{h^2} exp\left\{ \frac{(\mathrm{x}_l - \mathrm{x}_i)^T \boldsymbol{C}_l (\mathrm{x}_l - \mathrm{x}_i)}{-2h^2} \right\} \tag{1}$$

where $l \in \{1, ..., P\}$, P is the number of pixels in a local neighborhood, h is a global smoothing parameter, and $\boldsymbol{C}_l$ is a covariance matrix estimated from a collection of spatial gradient vectors within the local analysis window around a position x_l. For an image (or frame) of pixels x_i, where $i = \{1, ..., M\}$, the proposed algorithms assembles feature matrices $\mathbf{F}_i = [\mathbf{f}_i^1, ..., \mathbf{f}_i^L]$. These $\mathbf{F}_i$ matrices are used to construct



(a) Original Stimulus

(b) Elements with label *Red*

(a) Elements with label *Vertical*

(b) Boolean Map output between (b) and (c)

Figure 13: Example of the low-level vision process according to Boolean Maps theory [44].

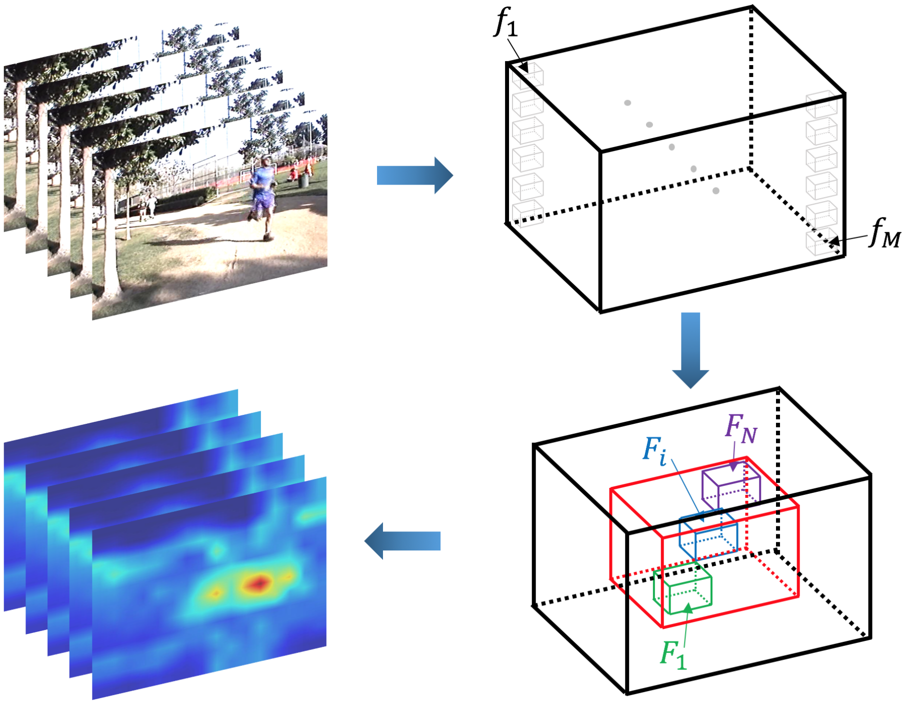


Figure 14: Illustration of Spatio-temporal Saliency Detection by Self-Resemblance algorithm proposed by Seo et al. [77].

the center-surround comparison model, which compares feature matrix $\mathbf{F}_i$, at pixel x_i, with *center+surround* feature matrices $\{\mathbf{F}_1, ..., \mathbf{F}_N\}$. Finally, the saliency map is computed according to:

$$S_i = \left(\sum_{j=1}^{N} exp \left(\frac{-1 + \rho(\mathbf{F}_i, \mathbf{F}_j)}{\sigma^2} \right) \right)^{-1} \tag{2}$$

where N is the size of the local neighborhood, $\rho(\mathbf{F}_i, \mathbf{F}_j)$ is the matrix cosine similarity (MCS) measure, and σ is a parameter controlling the fall-off of weights.

2.2.2.2 Spatio-temporal Saliency detection using phase spectrum of quaternion Fourier transform (PQFT)

Guo et al. proposed using phase-only Fourier transform reconstruction of the quaternion representation of images and video frames to detect saliency [31]. The main idea behind the proposed algorithm is the fact that phase-only reconstruction of the Fourier transform produces spikes in the locations that corresponds to sudden signal changes (i.e. when the signal change is not common in its immediate neighborhood the phase-only reconstruction of the signal produces a spike in the corresponding location in the reconstructed signal). For example, the signals in Figure 15 change over time, but only the ones that have sharp *uncommon* change would have a spike in their corresponding location in the reconstruction results. To include information from color, intensity, and motion features, the authors propose using quaternion system to represent images [22]. The proposed algorithm uses two color channels, $RG(t)$ and $BY(t)$, to represent color information, and two additional channels for intensity $I(t)$ and motion $M(t)$, which are computed according to:

$$RG(t) = \frac{3}{2}\big(r(t) - g(t)\big) \tag{3}$$

$$BY(t) = \frac{1}{2}\big|r(t) - g(t)\big| \tag{4}$$

$$I(t) = \frac{r(t) + g(t) + b(t)}{3} \tag{5}$$

$$M(t) = \big|I(t) - I(t - \tau)\big| \tag{6}$$

where $r(t)$, $g(t)$, and $b(t)$ are the red, green, and blue, respectively, color channels in the input image, and τ is a latency coefficient. These four channels are used to represent the quaternion image $q(t)$ as follows:

$$q(t) = f_1(t) + f_2(t)\mu_2 \tag{7}$$

$$f_1(t) = M(t) + RG(t)\mu_1 \tag{8}$$

$$f_2(t) = BY(t) + I(t)\mu_1 \tag{9}$$

where μ_1, i = 1,2 satisfies $\mu_i^2 = -1$, $\mu_1 \perp \mu_2$. After representing the video frames in their quaternion representation, the proposed algorithm computes the quaternion Fourier transform (QFT) of the image $q(n, m, t)$ according to [22]:

$$Q[u, v, t] = F_1[u, v, t] + F_2[u, v, t]\mu_2 \tag{10}$$

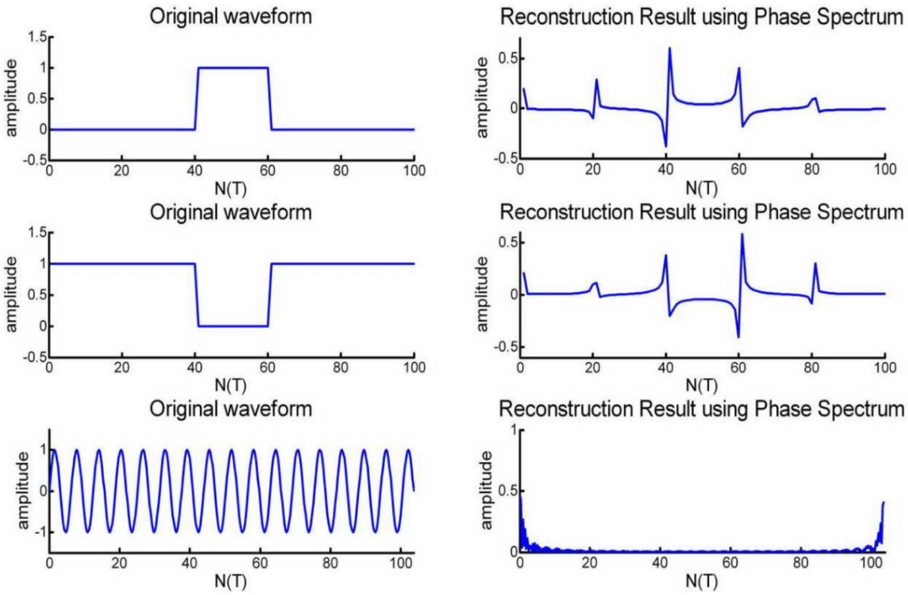


Figure 15: Examples of phase-only spectrum reconstruction [31].

$$F_i[u,v,t] = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e^{-\mu_1 2\pi(\frac{mv}{M}+\frac{nu}{N})} f_i(n,m,t) \tag{11}$$

After constructing $Q(t)$, the amplitude $||Q(t)||$ is removed and the phase of the spectrum $\Phi(t)$ is used to for reconstruction according to:

$$Q^{'}(t) = e^{\mu\Phi(t)} \tag{12}$$

then the inverse quaternion Fourier transform is used to obtain phase-only reconstructed image $q^{'}(t)$ according to:

$$f_i(n,m,t) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e^{\mu_1 2\pi(\frac{mv}{M}+\frac{nu}{N})} F_i[u,v,t] \tag{13}$$

Finally, the saliency map $sM(t)$ is computed according to:

$$sM(t) = g * ||q^{'}(t)||^2 \tag{14}$$

where g is a 2D Gaussian filter.

2.2.2.3 Saliency Detection for Videos Using 3D FFT Local Spectra (3DFFT)

Long et al. proposed using 3D FFT local spectra to detect saliency in videos [61]. Unlike other saliency detection algorithms that use frequency domain local features

[40, 31], the authors proposed detecting saliency by efficiently computing the local spectral of videos in 3D FFT directly instead of computing the spatial and temporal saliency separately. This approach provided a significant computational savings but results in producing scrambled saliency results. To solve this problem, the authors proposed using spectral decomposition as shown in Figure 16. When the point $M(a_0, b_0, c_0)$ is closer to the f_t-axis, it will mostly contain information about the temporal change of the scene. On the other hand, when $M(a_0, b_0, c_0)$ is closer f_x-f_y plane, the information it contains will be more related to spatial changes in the scene [61]. Therefore, the authors calculate the temporal component of the 3D FFT spectra as follows:

$$F_t(a_0, b_0, c_0) = F(a_0, b_0, c_0) \times sin\theta = F(a_0, b_0, c_0) \times \frac{c_0}{\sqrt{a_0^2 + b_0^2 + c_0^2}} \tag{15}$$

similarly, the spatial component is computed according to:

$$F_s(a_0, b_0, c_0) = F(a_0, b_0, c_0) \times cos\theta = F(a_0, b_0, c_0) \times \frac{\sqrt{a_0^2 + b_0^2}}{\sqrt{a_0^2 + b_0^2 + c_0^2}} \tag{16}$$

Using the temporal and spatial components of 3D FFT local spectra, the temporal saliency map is computed using the center-surround model [77] according to:

$$S_t(i, j, k) = \frac{1}{N} \sum_{i_0, j_0, k_0} \left| E_t^n(i, j, k) - E_t^n(i + i_0, j + j_0, k + k_0) \right| \tag{17}$$

where S_t is the temporal saliency map, N is the number of pixels in a window of size $L_1 \times L_2 \times L_3$ centered at (i, j, k), E_t^n is the normalized energy of the FFT spectra in the frame at time t. Similar procedure is used compute the spatial saliency map S_s. Finally, the combined saliency map is computed by averaging the temporal saliency map and the spatial saliency map, as shown in Figure 17.

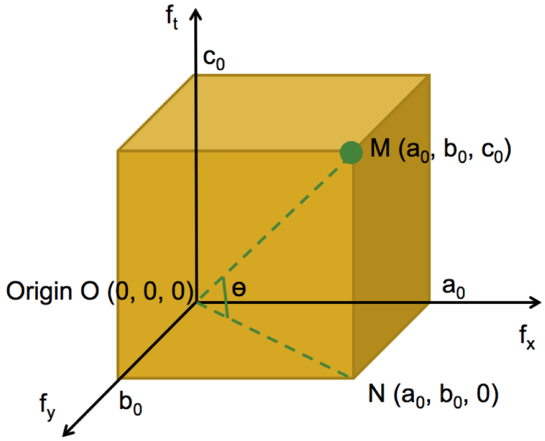


Figure 16: Illustration of FFT spectral decomposition used in 3DFFT algorithm [61].

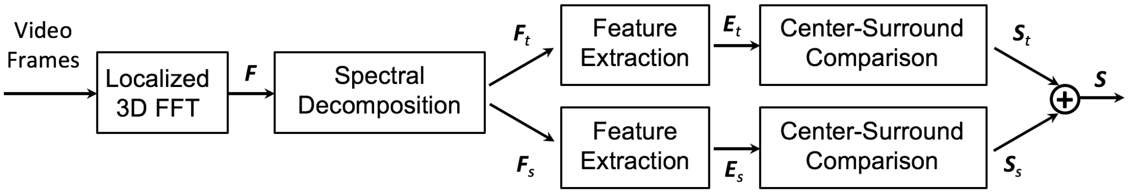


Figure 17: Block diagram of Saliency Detection for Videos Using 3D FFT Local Spectra [61].

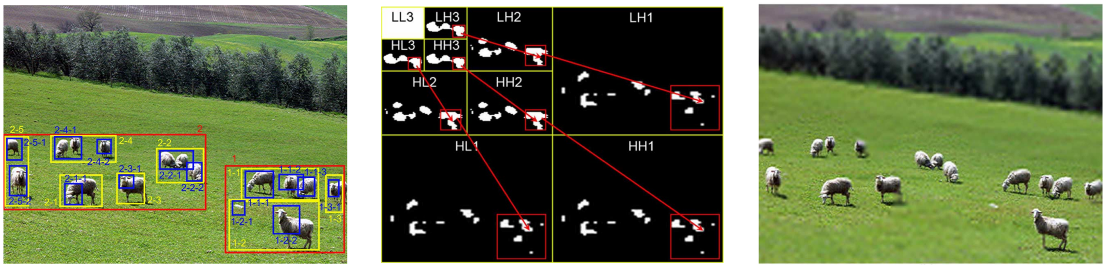
2.2.3 Applications

2.2.3.1 Compression

Guo et al. demonstrated saliency-based image compression by developing a scheme for Hierarchical Selection (HS) that extends the wavelet domain foveation weighting (WDFW) and uses phase spectrum of quaternion Fourier transform (PQFT) for saliency detection [32]. Using multiresolution saliency maps, the authors iteratively select the most prominent objects by starting from the saliency map with lowest resolution, which focuses on large objects in the scene, to the saliency map with the highest resolution, which identifies objects on finer scale. For example, in Figure 18(a), the HS scheme selects two large groups of sheep, red rectangles 1 and 2, as the first level of hierarchy. Then, the higher resolution saliency maps are able to identify smaller groups shown in yellow rectangles. Finally, individual sheep are detected and shown in blue rectangles. Using the identified salient objects, the authors propose applying multiresolution masks, that correspond to each hierarchy level, to the wavelet domain foveation weighting model proposed in [88]. Figure 18(b) continues the procedure on the example image in Figure 18(a) by applying the generated masks from HS scheme to the wavelet domain coefficients of the input image. The results of this process, on the example image, is shown in Figure 18(c). The proposed method achieves 41.56% compression rate compared to 52.2% achieved by state-of-the-art algorithm for output images with perceptional similar image quality scores.

2.2.3.2 Auto-Cropping

Stentiford proposed an algorithm for auto-cropping images based on a measure computed using saliency maps [81]. The *informativeness* measure reflects the region of the image with the maximum average saliency score. This measure is computed by averaging the sum of the saliency scores of individual pixels inside a region over the



(a) (b) (c)

Figure 18: An Example that shows the saliency-based compression procedure proposed by Guo et al. [32]. (a) Hierarchical Selection (HS) scheme, (b) Multiresolution wavelet domain foveation model, (c) Compressed image

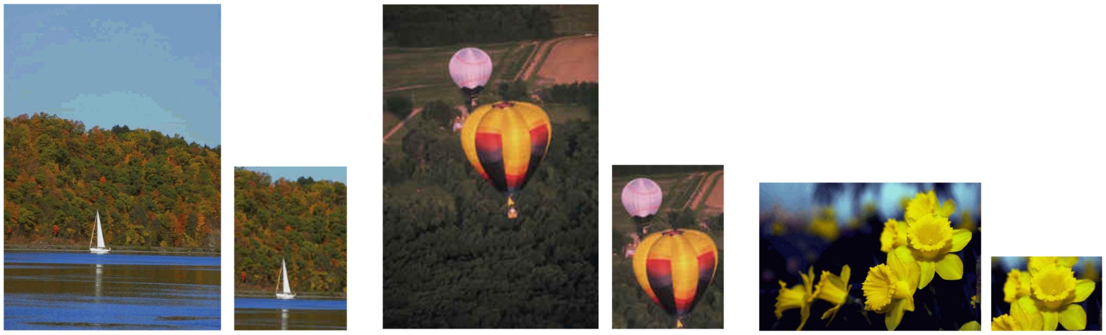


Figure 19: Examples of auto-cropped images using saliency-based auto-cropping algorithm proposed by Stentiford [81].

size of that region. Using informativeness measure, Stentiford proposes using a user-specified window size to search over the whole image and identify the window with the maximum informativeness, which is claimed to correspond to the optimum cropping window. Figure 19 show three examples of images along with their automatically cropped versions.

2.2.3.3 Rendering

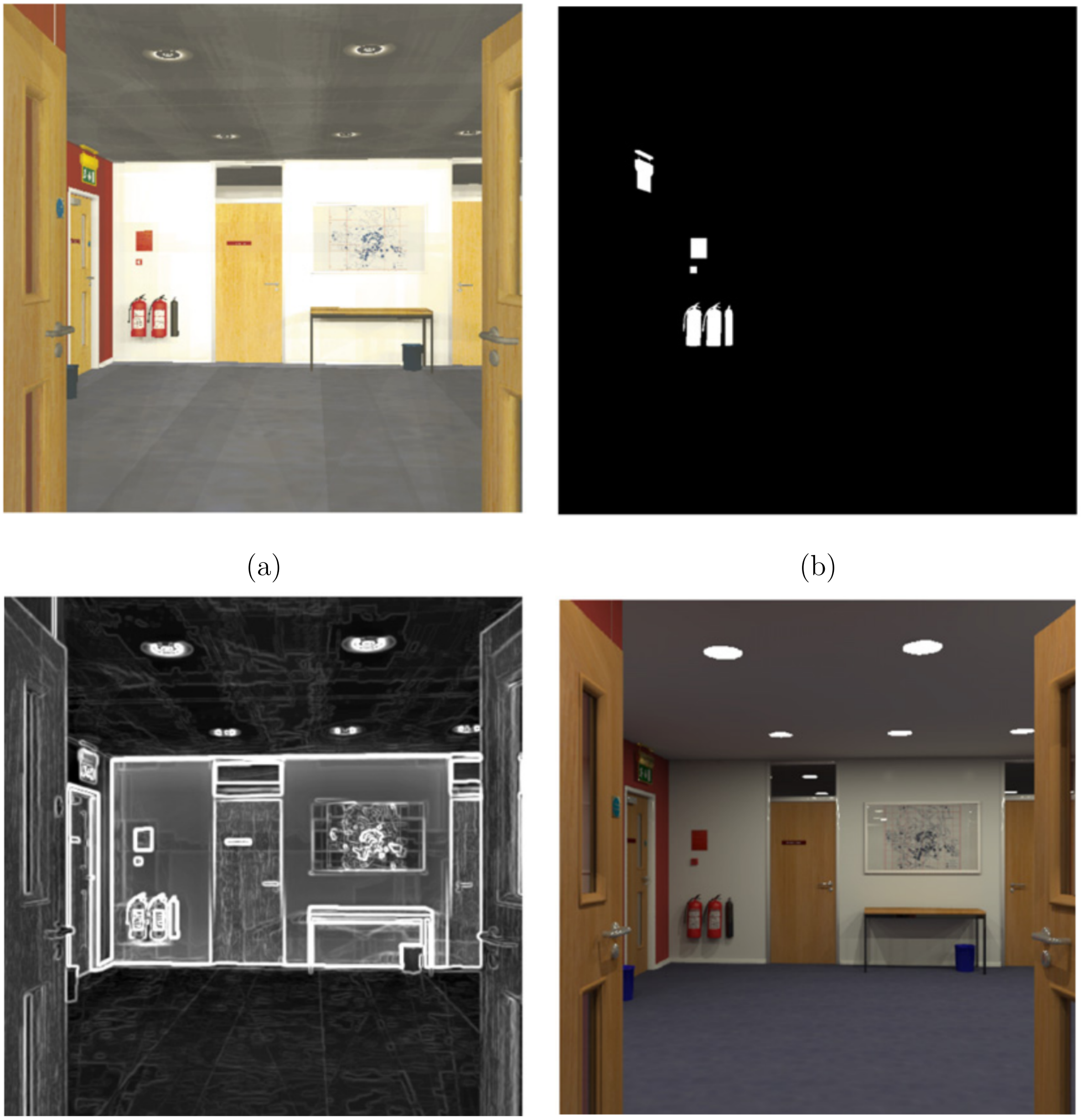
Debattista et al. proposed using visual attention models to selectively render high-fidelity virtual environment [18]. To avoid the reduce expense of global illumination computation, the authors propose using *importance maps* that attempt to identify the parts of an environment that are attended to by the viewer. By computing these importance maps, the rendering engine can be guided, during the *selective guidance* stage, to only afford computational resources to the parts that are deemed perceptually more important. The proposed algorithm works by, first, rendering and image preview that is used to compute *task map*, which corresponds to top-down feature maps, and a *saliency map* that encoded the bottom-up attention cues. Figure 20 shows an example of the proposed algorithm. In Figure 20(a), an image preview is rendered with fast and rough rendering settings. Next, the image preview is used to identify the task map, Figure 20(b), and saliency map, Figure 20(c), which both are combined into importance map. The importance map, in turn, is used to guide the rendering engine which generates the final image shown in Figure 20(d).

2.2.3.4 Image Quality Assessment

Lin et al. proposed using visual saliency to improve image quality assessment (IQA) metrics by giving hire weights to the patches that are deemed salient by the saliency detection algorithm [56]. The proposed algorithm divides the input image to non-overlapping patches and then computes a normalized *visual saliency degree* for each patch, which is used later as a weight for that patch. The overall visual-saliency-enhanced IQA metrics are computed by weight-averaging the local IQA metric for the image patches. For example, the patches of the input image in Figure 21(a) that overlaps with the boat as well as the coastline would have higher weights because the saliency map, shown in Figure 21(b), identify these patches as visually prominent areas. The reported experiments show an improvement in the subjective mean opinion score (MoS) compared to the originally proposed version of the IQA metrics.

2.2.3.5 Scene Understanding

Bharath et al. proposed a saliency-guided framework for scene understanding [7]. The proposed algorithm uses saliency detection for providing possible candidates for



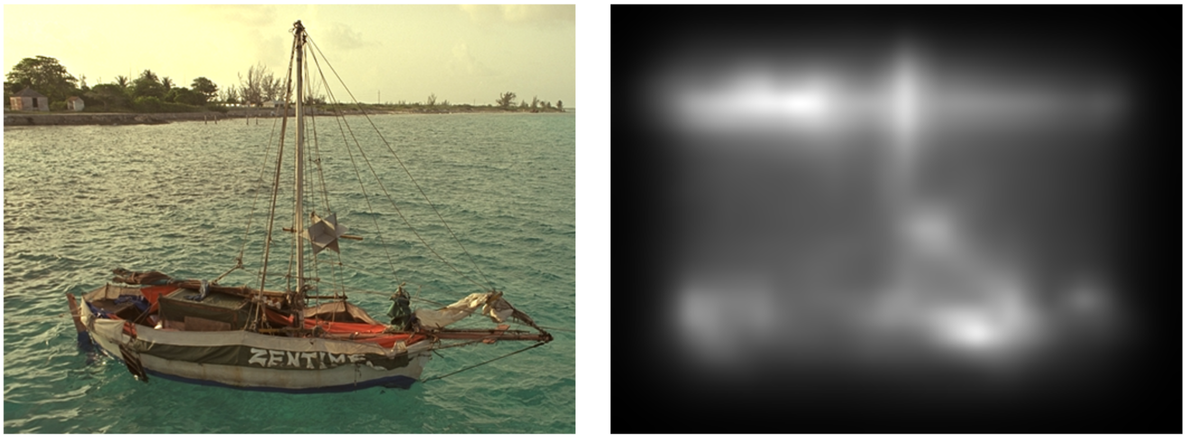
(a)

(b)

(c)

(d)

Figure 20: Example of saliency-based selective rendering algorithm proposed by Debattista et al. [18].



(a) Input image

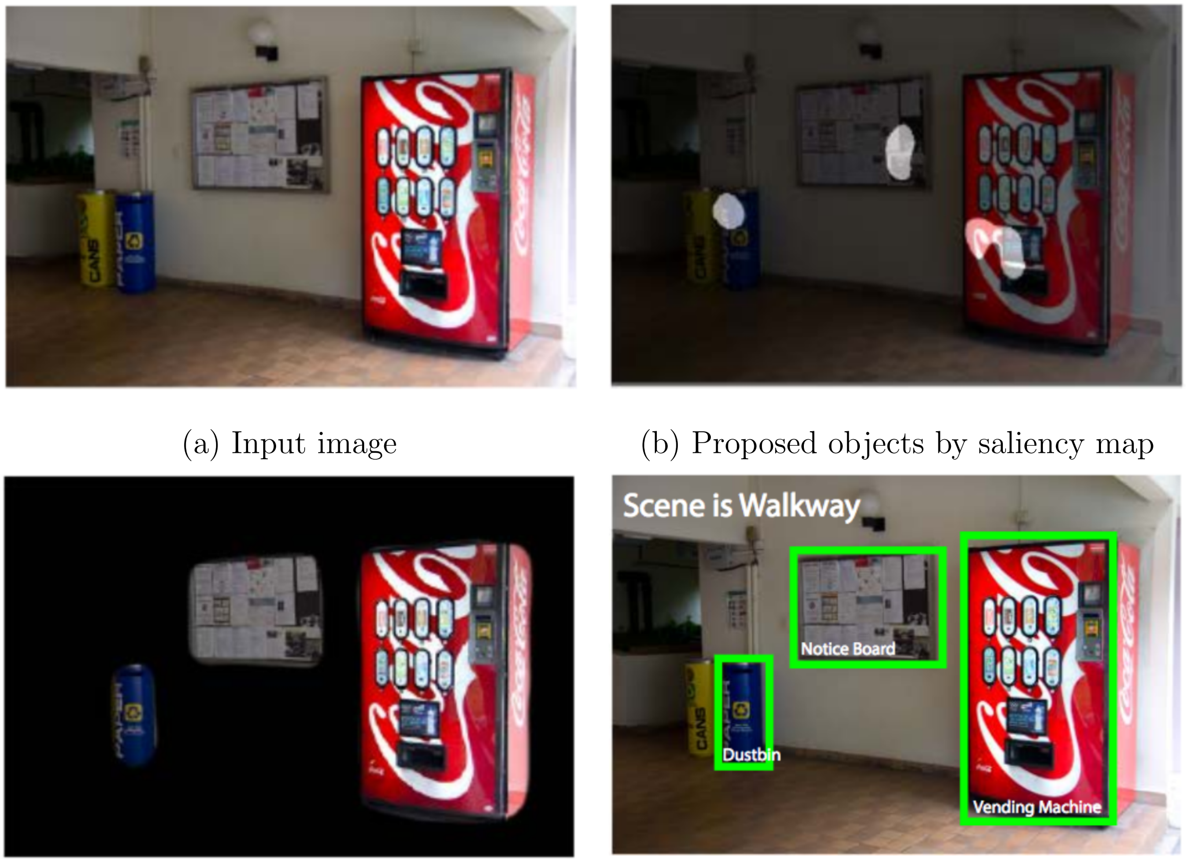
(b) Saliency map

Figure 21: Example of saliency-based image quality assessment algorithm proposed by Lin et al. [56].

objects in a given scene. Even though other methods for detecting objects exist, the author argue that saliency detection provides faster and more efficient results than the other methods without much degradation in accuracy, which makes the framework as whole more scalable. An overview of the framework is shown in Figure 22. First, the input image, Figure 22(a), is used to compute a saliency map which is used to identify possible objects, Figure 22(b), in the scene using an automatic region of interest (RoI) detector. Afterwards, objects are segmented using graph-cut based segmentation algorithm and classified using bag-of-features model [7] as shown in Figure 22(c). Finally, the scene is classified based on the recognized objects using a decision tree model as shown in Figure 22(d).

2.2.3.6 Surveillance Video Summarization

Salehin et al. proposed an efficient method for summarizing surveillance video [75]. In video summarization, repetitive and uninformative frames are dropped from the input video and ultimately the algorithm generates a short summary video that provides complete description of the content of the input video. The proposed method utilizes



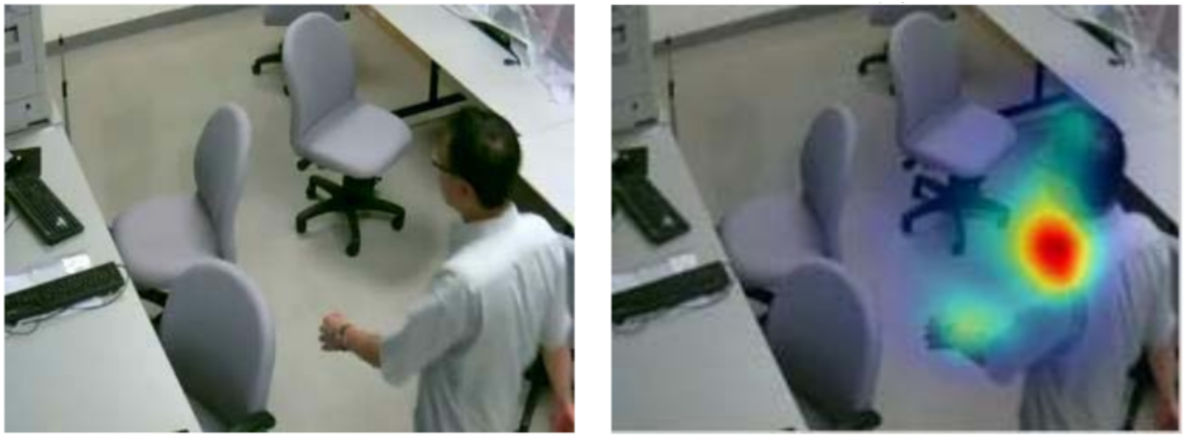
(a) Input image

(b) Proposed objects by saliency map

(a) Segmented objects

(b) Recognized scene

Figure 22: Example of saliency-based scene understanding algorithm proposed by Bharath et al. [7].



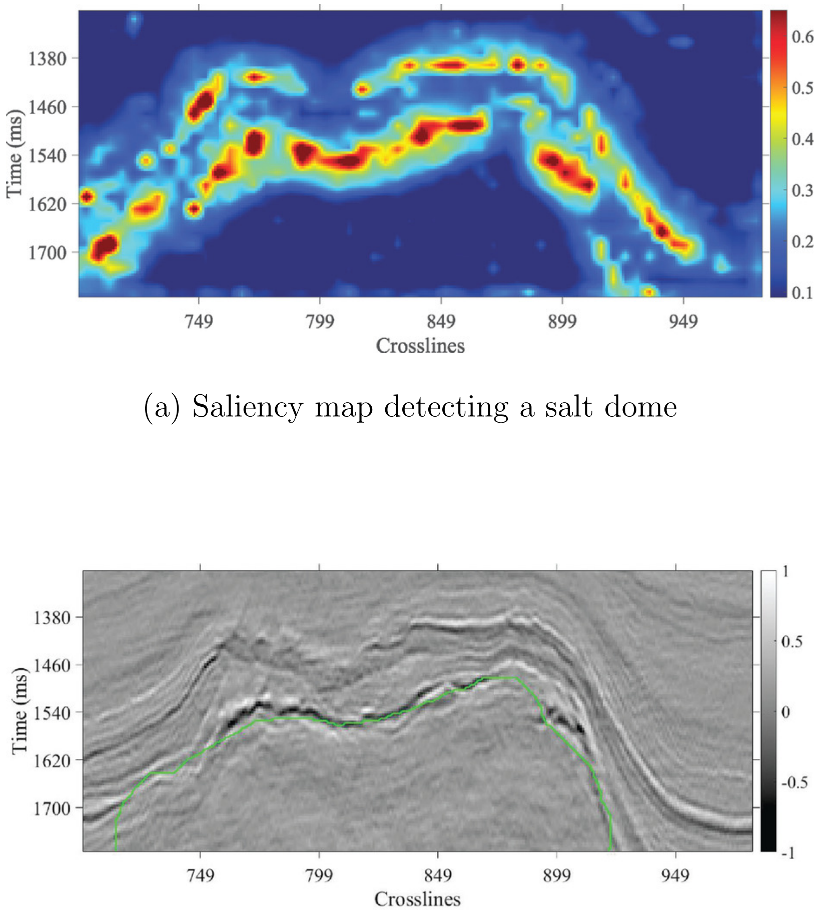
(a) Input Frame (b) Saliency map difference

Figure 23: Example of saliency-based surveillance video summarization algorithm proposed by Salehin et al. [75].

features from foreground objects detection by using Gaussian mixture-based dynamic background modeling (BGM) and subtracting the raw frame, frame-to-frame motion information by thresholding frame-to-frame difference, and visual saliency difference between consecutive frames. The three feature sets are used to train a support vector machine (SVM) that employs a radial basis function (RBF) which can be used to classify frames into informative and uninformative frames. Figure 23 shows an example frame with its corresponding visual saliency feature map.

2.2.3.7 Seismic Interpretation

Shafiq et al. proposed a new seismic attribute for computational seismic interpretation and verified by experiments on F3 block from the North Sea dataset that the proposed attribute is effective for salt dome delineation [79]. The proposed attribute is designed based on the modeling of human visual system. After computing the saliency map of the migrated seismic volume, the map is thresholded and semi-automated region growing algorithm is used to capture the area of the salt dome. Finally, post-processing morphological operation, including dilation and perimeter



(a) Saliency map detecting a salt dome

(b) The delineation results after post-processing

Figure 24: Example of saliency-based seismic interpretation algorithm proposed by Shafiq et al. [79].

extraction, are used to generate the delineation results. Figure 24 shows an example of seismic slice that shows a salt dome and the corresponding saliency map detection.

2.3 Analysis of Eye-fixation Data

Little work has been done to analyze eye-fixation maps separately from visual stimuli. In [80], authors use Judd's et al. image dataset[49] to study spatiotemporal eye-fixation data of 15 subjects looking at 1003 images in that dataset. The analysis utilizes Singular Value Decomposition (SVD) to compute the eigenvectors of the correlation matrix of eye-fixation data. In short, the authors proposed using spatiotemporal histograms of eye-fixation data for each subject as single column in data matrix of size $m \times h_r \times h_c \times n$, where m is the number of subjects, h_r and h_c are the dimensions of 2D histogram, and n is number of time intervals taken into account. The correlation matrix, constructed from the data matrix, is decomposed using SVD procedure and eigenvalues and vectors are extracted and mapped back to image space. The proposed analysis shows that the first Eigen vector accounts for 21% of the eye-fixation data and correspond to highly salient locations in the images. Interestingly, the decomposition of the correlation matrix verified that the correlation matrix is full rank, $rank = m$, which suggest that the spatiotemporal viewing patterns of m subjects looking at the same image are basically independent, which may be attributed to image complexity as have been suggested in [48]. Furthermore, the authors in [80] demonstrated that a set of salient locations and their time sequences corresponding to the first eigenvector can be used to evaluate computational saliency models such as the one proposed in [45]. In another work [2], the authors draw on the analysis procedure proposed in [80] to analyze eye-fixation data for single subject across different images. They found that 23 percent of the data can be accounted by a single eigenvector regardless of the image content, which it turns out to be correlated with image center location. Thus, authors proposed an evaluation metric, robust Area-Under-Curve (rAUC), for computational saliency models that takes such viewing patterns into account.

In [8], the authors used statistics about eye-fixation data to decode the image category. They used a subset of the NUSEF dataset [73] containing five categories over a total of 409 images. The feature vector includes fixation points histogram, fixation duration, saccade length, orientation, duration, and velocity in addition to saliency maps generated from state-of-the-art saliency detection algorithms. The feature vector and image labels are used to train a multi-class Support Vector Machine

(SVM) with Radial Basis Function (RBF) kernel. The decoding results show that saliency map generated by Itti et al. [45], along with eye-fixation data statistics, achieved highest accuracy, 2.5 times higher than random chance. The authors of [8] show that it is feasible to decode image category from feature vector of saliency, saccade, and fixation statistics. This could be, they believe, due to similar saliency patterns across scenes of a category or semantic biases of fixation in each category.

2.4 Uncertainty Analysis in Image and Video Processing Applications

Recently, there has been some research on uncertainty specific to image and video processing applications. In [30], authors proposed using an active learning algorithm based on one-versus-one (OVO) strategy support vector machine (SVM) to solve multi-class image classification. The results of OVO SVM are combined according to a cost function that maximizes the diversity of the chosen set of examples and minimizes the uncertainty of the classification of this set. The uncertainty in this work is estimated using the difference in number of votes between the highest votes class and the second highest class. As the difference in number of votes increases, it is more likely that the highest votes class is the true representative class, so the uncertainty is lower. In the context of medical image registration, Saygili et al. [76] proposed a confidence measure that reflects the accuracy of the registration process of a pair of images. The proposed measure relates the confidence of the registration process at each pixel to the global minima and the steepness of a predefined cost function. The registration at a given pixel is expected to be more reliable if the associated cost function produces a global minima at that location and the cost function in its local region is very steep. In the context of stereo vision and depth estimation, numerous confidence measures have been proposed in literature [42]. Typically, these measures associate the confidence of pixel's match with the shape of the matching cost function, e.g. sum of absolute differences, around that pixel. Haeusler et al. [33]

proposed applying random decision forest framework on a large set of diverse stereo confidence measures to improve the performance of stereo solvers.

In the context of saliency detection, there has been very limited work to address the problem of quantifying uncertainty. Directly applying uncertainty and confidence measures proposed for other image and video processing applications might not take into consideration characteristics of human visual attention mechanisms which are crucial for saliency detection. The authors in [24] proposed a supervised method to estimate the uncertainty associated with detected saliency of a video pixel. The method uses binary entropy function to measure uncertainty according to the probability of a pixel being salient given the distance of the target pixel from the center of mass $p(s|d)$, and connectedness of the target pixel $p(s|c)$. The coordinates of the center of mass of saliency map $[x_c, y_c]$ are first calculated using the ground truth map. Then, the Euclidean distance, d, is calculated for each pixel in the computed saliency map. Similarly, the connectedness feature, c, is calculated by counting the number of salient neighbors. The probability densities $p(s|d)$ and $p(s|c)$ are fitted using salient object segmentation ground truth from images dataset by Achanta et al. [1].

CHAPTER III

CORRELATION OF EYE-FIXATION MAPS IN VIDEOS

As discussed in the previous chapter, uncertainty formulation and representation depends on the uncertainty representation framework and the its assumptions. For a framework-independent formulation of uncertainty, we seek to abstract the problem and its application and compute objective parameters that encode domain-specific knowledge and can be used to estimate uncertainty. These parameters can be regarded as features to feed into uncertainty representation frameworks. In computational video saliency, such features can be computed from the input video, the saliency detection algorithm, the generated saliency maps, or a combination of them. Thus, studying common patterns of saliency maps would be useful to formulate features for uncertainty estimation. Unlike most work reported in the literature, we analyze saliency maps as a separate entity away from its input video rather than an output response of HVS to the input video. To do so, we use eye-fixation data as ground truth for saliency maps and analyze such data to gain insights into the structure and dynamics of the saliency maps. We focus on the relation between the saliency of a pixel and that of its direct neighbors, without making any assumption about the structure of the eye-fixation maps. By employing some basic concepts from information theory, the analysis shows substantial correlation between the saliency of a pixel and the saliency of its neighborhood.

3.1 Preparing Eye-fixation Maps

This study uses eye-fixation maps from the public CRCNS dataset [47]. The dataset includes 50 videos of diverse nature including street scenes, TV sports, and video games; 12 categories in total. The videos are 480 × 640 in size, 5 to 90 seconds in duration, and 30 frames per second (fps). Most of the videos have realistic distortions including camera movement from handheld devices as well as TV camera panning, low light, significant motion distortion, and compression artifacts. Details about experiment setup is provided in [47]. The eye tracking data is collected from eight subjects using an ISCAN RK-464 eye-tracker at 240 Hz sampling rate, which was calibrated every five clips using 9-point calibration. The stimuli were displayed on 22" CRT monitor at $80cm$ viewing distance with mean screen luminance of 30 cd/m^2. Eye tracking data are provided for each human subject separately in a string of eye gaze coordinates, which span 0 to 639 in the horizontal direction and 0 to 479 in the vertical direction with location (0,0) being at the top left corner of the monitor. Labels are available for each eye-gaze sample, e.g., fixation, saccade, and during blink, just to name a few.

For a given video sequence, we transform the eye-tracking data to an eye-fixation map according to the following procedure:

1. Initialize a 480 × 640-frame with zeros

2. Exclude *Saccade* or *loss-of-tracking* samples from further processing. Only *fixation* or *smooth pursuit* samples are processed

3. Increment the corresponding pixel value, of every sample, by one.

4. Process the eye-tracking data frame by frame

5. After processing all frames, construct an eye-fixation map with the same size and number of frames as the video sequence.

Additionally, video frames are often downsampled, i.e. reduced in size, to satisfy application constrains. Thus, we analyze eye-fixation maps at various scales of the original map by reducing its size. For an original map $F[m, n, k]$, the s-scale map,

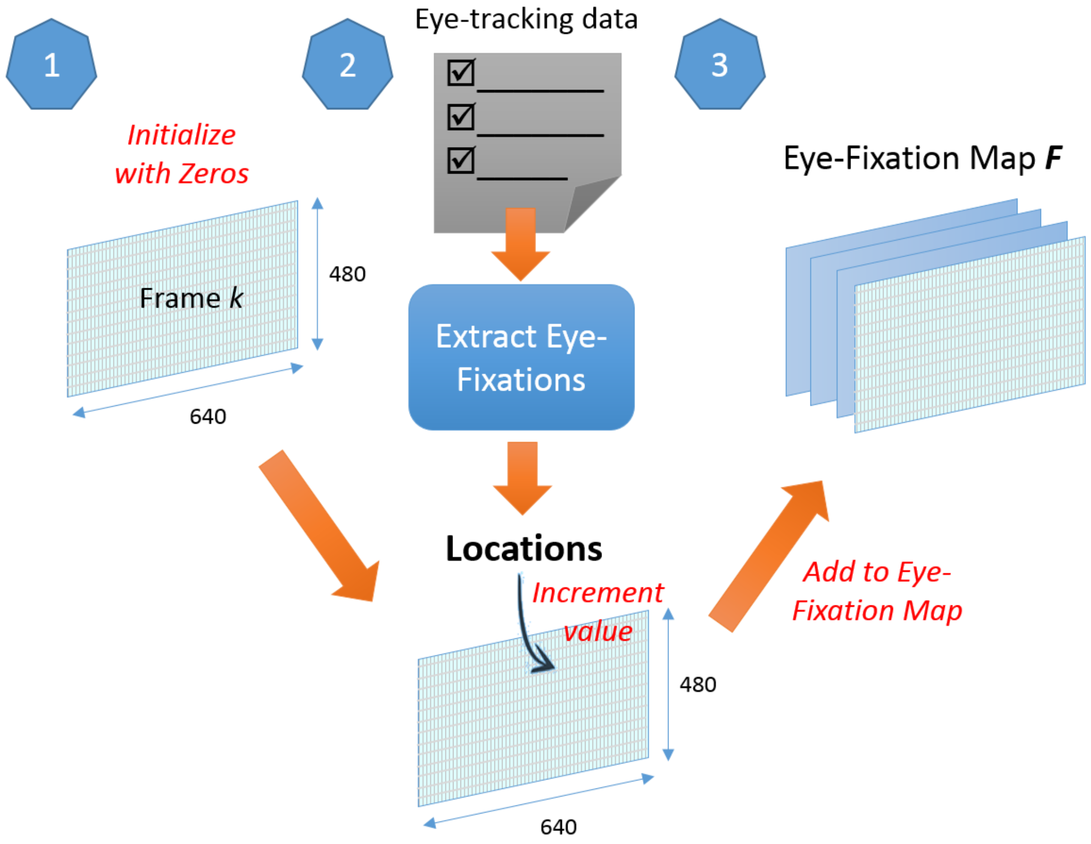


Figure 25: Illustration of the procedure for preparing the eye-fixation data.

$F^{(s)}[m,n,k]$, is formed as

$$F^{(s)}[m,n,k] = \sum_{\forall i,j \in R_s} F[i,j,k] \tag{18}$$

where R_s is a window that defines the set of pixels in $F[m,n,k]$ that correspond to pixel $[m,n,k]$ in $F^{(s)}[m,n,k]$.

3.2 Correlation Results

3.2.1 Overall Map

For a $M \times N \times K$-eye-fixation map, F, where M is the height, N is the width, and K is the depth in frames; every pixel $x[m,n,k] \in F$ is considered an instance of a discrete integer random variable, X, that is:

$$X : \Omega \to E, \tag{19}$$

where Ω is the set of all possible outcomes of the eye tracking experiments, E is the observed set, and $x[m,n,k] \in \{0,1,2,...,L\}$ enumerates all possible outcomes using $L+1$ symbols. For such a eye-fixation map F, we compute the Shannon entropy of X as follows:

$$H(X) = E[I(X)] = -\sum_{i=0}^{L} P(x_i)\log_2 P(x_i), \tag{20}$$

where $E[\cdot]$ is the expectation operator, $I(X)$ is the self-information of X, and $P(x_i)$ is the probability mass function of X.

3.2.2 Spatiotemporal Neighbors

We are interested in examining the relationship between the pixels and their neighbors in eye-fixation maps. There are 9 pixels from frame $k-1$, 9 pixels from frame $k+1$, and 8 pixels from the current frame k; 26 direct spatiotemporal neighbors altogether, as shown in Figure 26. These neighbors are labeled $Y^{(j)}$, where $j \in \{1,2,...,26\}$. The conditional entropy of X, the center pixel, given the average of its direct neighbors

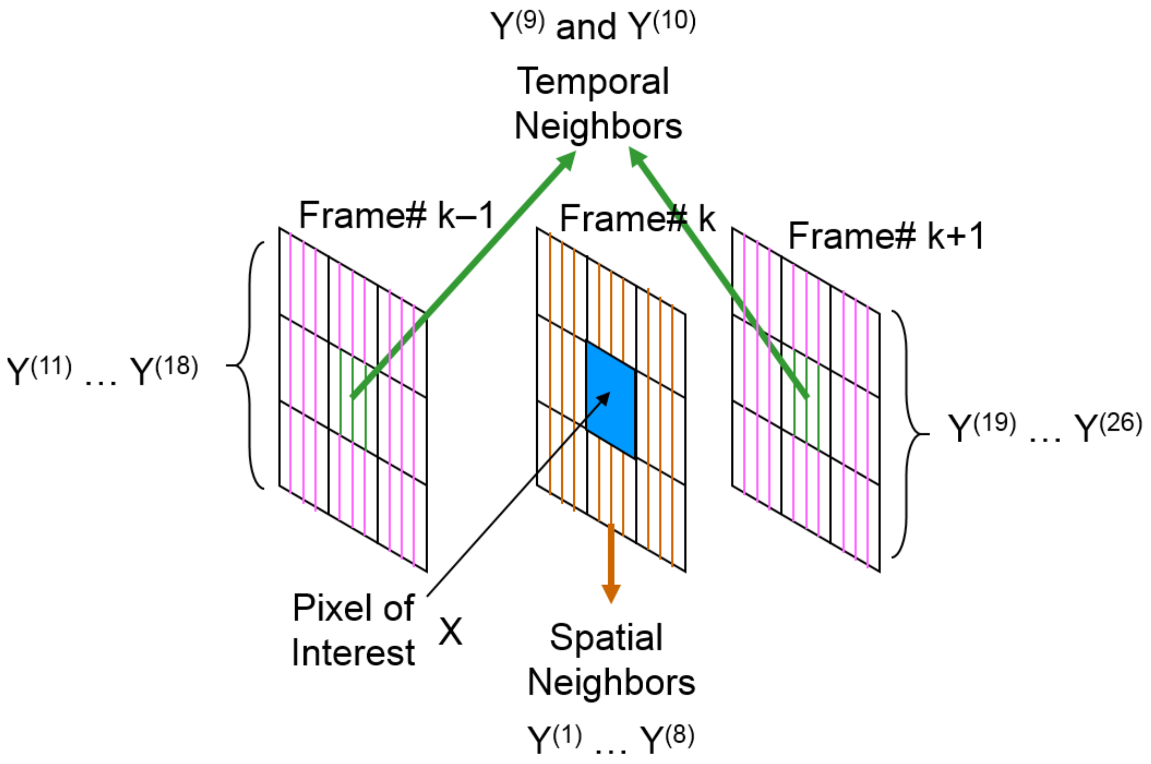


Figure 26: Illustration of neighborhood pixels grouping. Spatial neighbors $Y^{(1)}...Y^{(8)}$ of pixel X are hashed in orange color, temporal neighbors $Y^{(9)}$ and $Y^{(10)}$ of X are shown in green color, and the rest of spatiotemporal neighbors $Y^{(11)}...Y^{(26)}$ are shown in purple.

is:

$$H(X|Z) = \sum_{\forall x_i, z_j} P(x_i, z_j) \log_2 \frac{P(z_i)}{P(x_i, z_j)} \tag{21}$$

where $Z = f\big(Y^{(1)}, \ldots, Y^{(26)}\big)$ is the arithmetic mean of the 26 direct neighbors, and $P(x_i, z_j)$ is the joint probability mass function for X (the center pixel) and Z (the mean of its direct neighbors). A property of conditional entropy states that:

$$0 \leq H(X|Z) \leq H(X). \tag{22}$$

The equality between $H(X|Z)$ and $H(X)$ holds only when X is completely independent of Z; alternatively, $H(X|Z) = 0$ if Z completely determines X. In our experiments, the window size $R_s = 40 \times 40$, in Eq.(1), to reduces the time for computation while at the same time generates similar results to those obtained using the original map size. First, we evaluate the correlation between a map pixel and its direct spatiotemporal neighbors. Figure 27 shows the entropy values computed for each of the 50 video sequences in the dataset. As shown in the figure, the entropy of the eye-fixation drops when the spatiotemporal neighborhood average is considered (red curve). To have a basis for comparison, the entropy of the eye-fixation conditioned on a uniformly-distributed random variable (yellow curve) is shown in the same figure.

A significant 50% reduction in the entropy values, in most videos, indicates a strong correlation. Despite the variation among these videos, the entropy reduction is consistent across all videos in the dataset. In Figure 27, the average entropy reduction is 0.0815 bits with variance 3.2416×10^{-05}. Such low entropy is caused by the sparsity of the eye-fixation maps which most of its pixels are zero. However, the skewness of the probability mass function does not affect the analysis because the probability mass is concentrated in a single symbol. This mass can be redistributed equally among the remaining symbols which would moves the entropy (and the conditional entropy) up or down but does not change its shape. Gamecube02, gamecube06, and gamecube13 are the three video sequences with the highest entropy values because

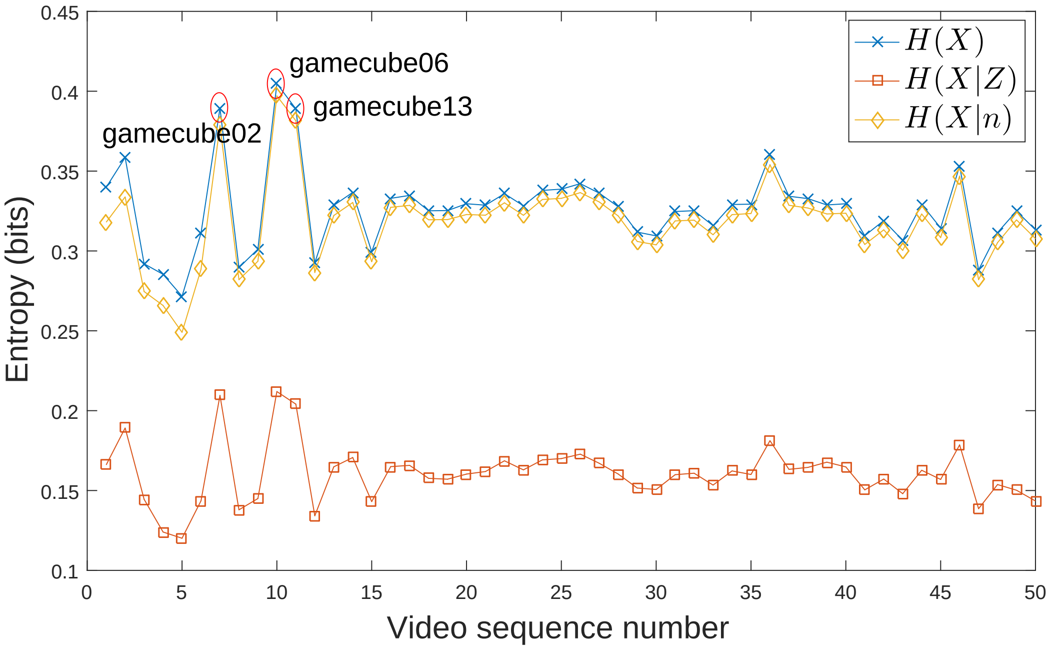


Figure 27: Entropy calculation for all video sequences in CRCNS dataset.

they are relatively longer and contain engaging content. These characteristics could contribute to the higher entropy because it might engage more cognitive processes than other sequences. This difference between categories of video content can be seen clearly in Figure 28.

3.2.3 Spatial Neighbors

To investigate the effect of the video content on the correlation between a pixel and its neighbors, we extend the model introduced above. We study the correlation between a map pixel in a given spatial location and its direct spatial neighborhood. For every location $[m, n]$ in the eye-fixation map, all pixels at that location, across all K frames, are considered instances of a random variable $X[m, n]$. Then, we use mutual information between two random variables to measure the correlation between a given $X[m, n]$ and $Q[m, n]$, the arithmetic mean of its eight direct spatial neighbors

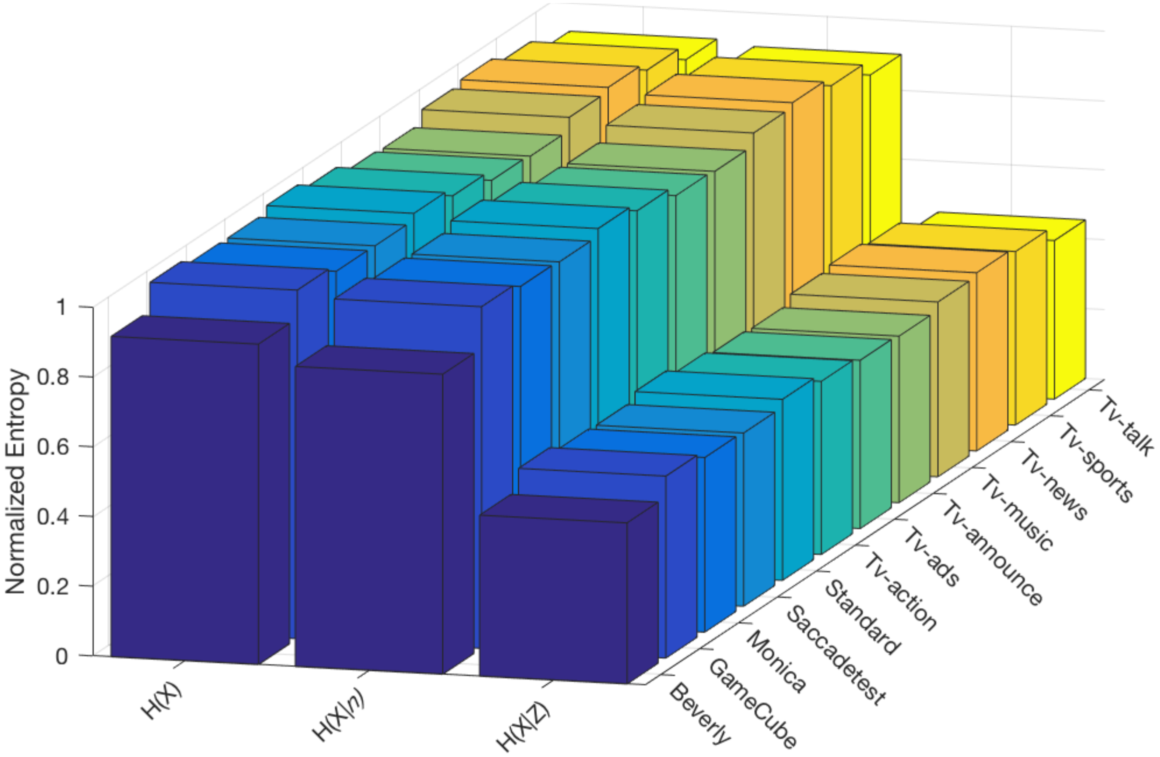


Figure 28: Entropy reduction across all videos in CRCNS dataset. Results reported here are computed using *Scale 1* saliency map of size 12×16.

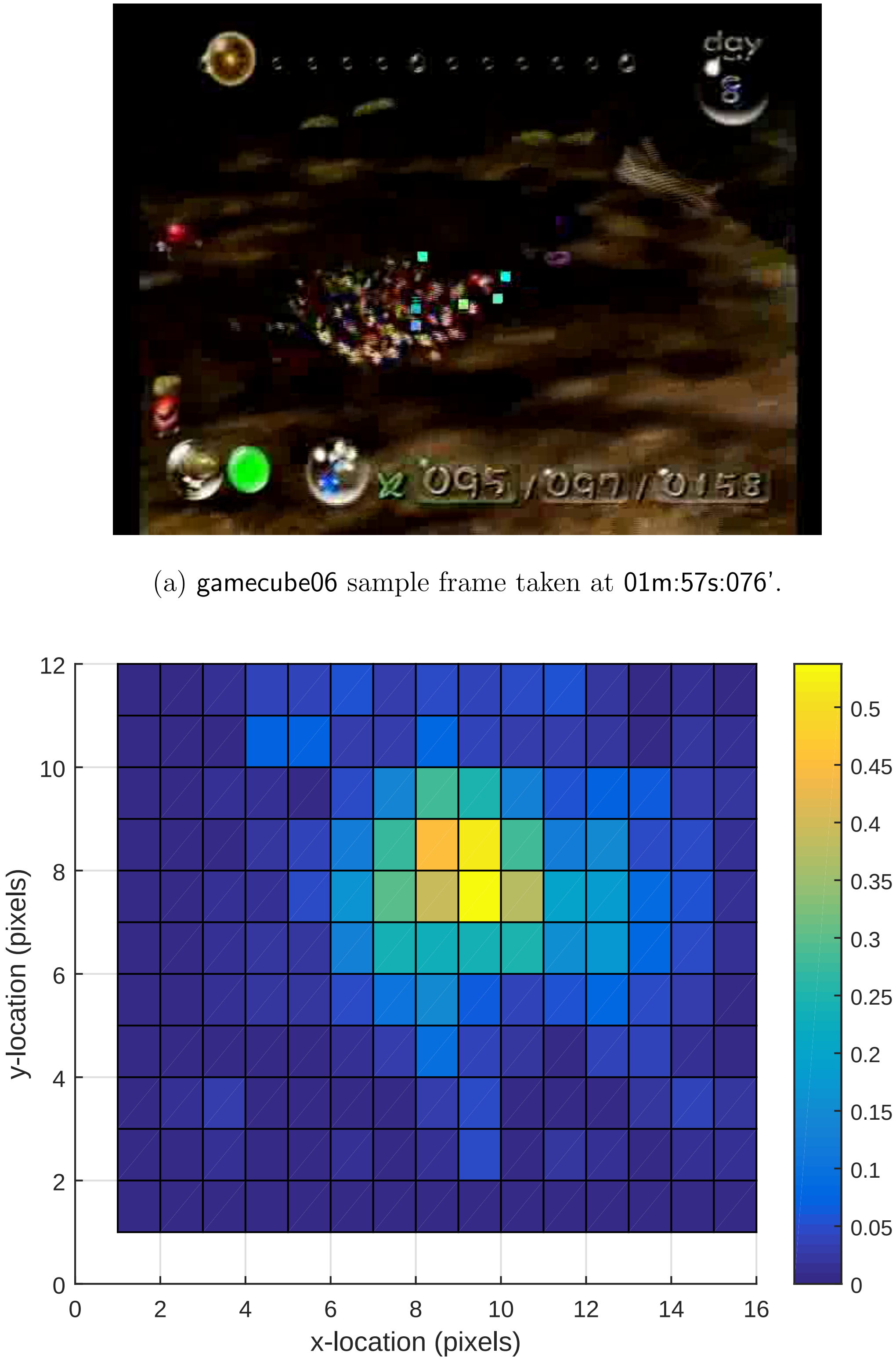
$X[m+i, n+j]$, where i and $j \in \{1, 0, -1\}$, as follows:

$$I(X[m,n]; Q[m,n]) = \sum_{\forall x_i, q_j} P(x_i, q_j) \log_2 \frac{P(x_i, q_j)}{P(x_i)P(q_j)}, \tag{23}$$

where $P(x_i)$ is the probability mass function of random variable $X[m,n]$, $P(q_j)$ is the probability mass function of $Q[m,n]$, the arithmetic mean of the spatial neighbors, and $P(x_i, q_j)$ is the joint probability mass function of $X[m,n]$ and $Q[m,n]$.

It would be interesting to evaluate the impact of the famous center-bias phenomenon [28] on the correlation of eye fixation maps. Research shows that such bias exist in most eye-fixation data related to images. In fact, in addition to naturally occurring bias, observed in images [28], every video sequence, in CRCNS, is preceded by a blinking cross, that lasts for 1 *sec*, in the middle of the screen, exactly at [239,319], which should further enforce the center-bias. However, our results show that *lack of knowledge* center-bias has a small effect on the overall correlation and is only present in the first few frames of almost all videos. The videos that have a center-bias, in this dataset, are the ones with *photography* center-bias which significantly influences the end results. This bias is caused by the tendency of photographers to place the object(s) of interest around the center of the video frames. Such bias is mostly eliminated from visual attention image datasets by shifting the object(s) of interest away from the center, doing the same for videos is difficult. This photography center-bias is particularly obvious in the gamecube videos, a sample frame is shown in Figure 29.(a), since the in-game camera system is designed to place the game character(s) in the center of the video.

When we compute the mutual information between a pixel at a given location and its direct spatial neighbors, the results are shown in Figure 29.(b). Gamecube06 results shows a high correlation, as the case in most gamecube videos, which can be attributed to the photography center-bias. Although textual information has been shown to attract human attention [14], this is not the case with gamecube videos.



(a) gamecube06 sample frame taken at 01m:57s:076'.

(b) gamecube06 mutual information given spatial location, across all frames.

Figure 29: gamecube06 sample frame alone with mutual information given spatial location, involving only spatial neighbors

Gameplay related text, located at the corners of the screen, does not attract eye-fixation for prolonged periods. We believe such pattern is due to the relatively low-information content of this kind of text.

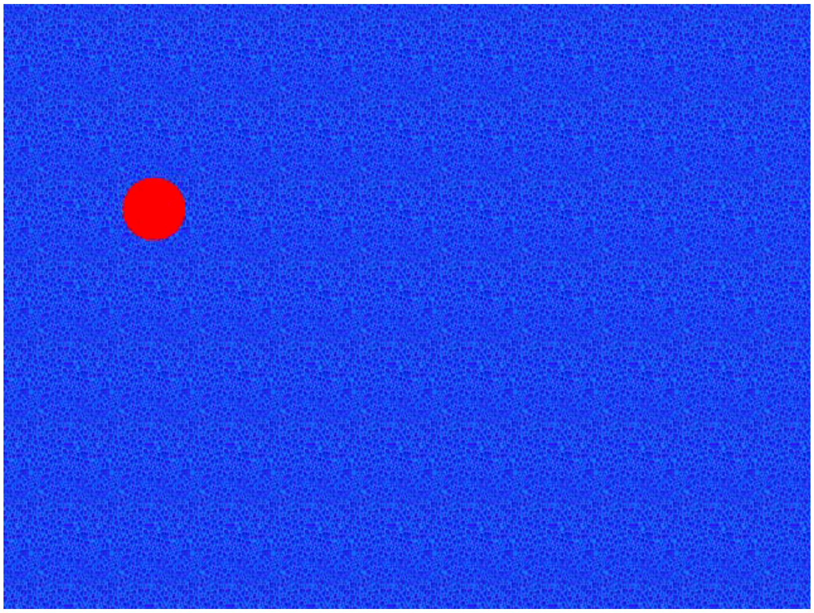
On the other hand, videos that lack photography center-bias exhibit totally different behaviour. Figure 30.(a) shows a sample frame of saccadetest video that consists of a blue textured background and a diagonally moving red dot. The eye-fixation map correlation, Figure 30.(b), is highest when there is a smooth pursuit following the red dot, due to high sampling rate and low spatial displacement of the object of interest. Similar trends can also be observed in video sequences such as beverly06, beverly07, and beverly08.

Additionally, interesting trends can be observed when there are multiple salient objects present in the scene, such as in the tv-news03 video, a sample frame of which shown in Figure 31.(a). The space-localized mutual information map, shown in Figure 31.(b), exhibits two centers of attention. One corresponds to the most semantically informative object in the scene, i.e., the news anchor's face. The other is the textual messages in the lower banner. Since the human subjects spend considerable periods of time looking at these two locations, the correlation is significantly higher than other locations in the eye-fixation map. The example results, especially those from the latter two without obvious center-bias, demonstrate that the higher correlation areas match very well with the human attention.

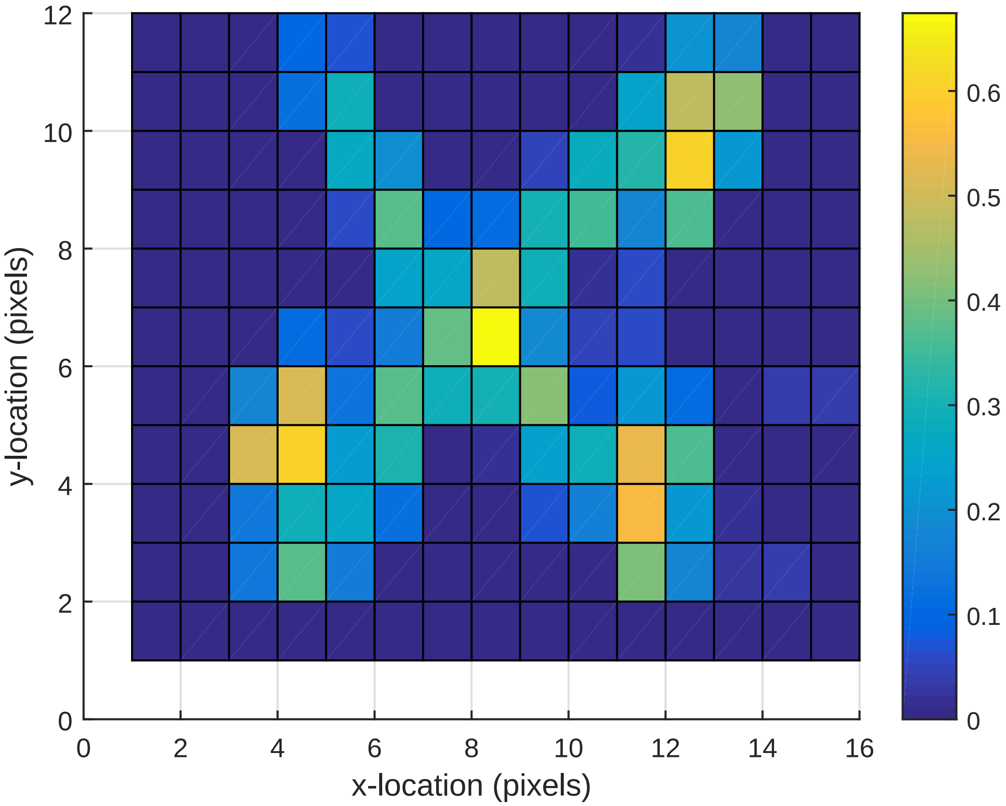
3.2.4 Temporal Neighbors

To analyze the correlation with temporal neighbors, each pixel in frame k of an eye-fixation map, $F(k)$, is considered as an instance of a random variable X_k. Then, we compute mutual information between X_k and W_k, a pixel-wise arithmetic mean of adjacent frames $F(k+D)$ where $D \in \{\pm1, \pm2, \pm3, \dots\}$, as follows:

$$I(X_k; W_k) = \sum_{\forall x_i, w_j} P(x_i, w_j) \log_2 \frac{P(x_i, w_j)}{P(x_i)P(w_j)}, \tag{24}$$

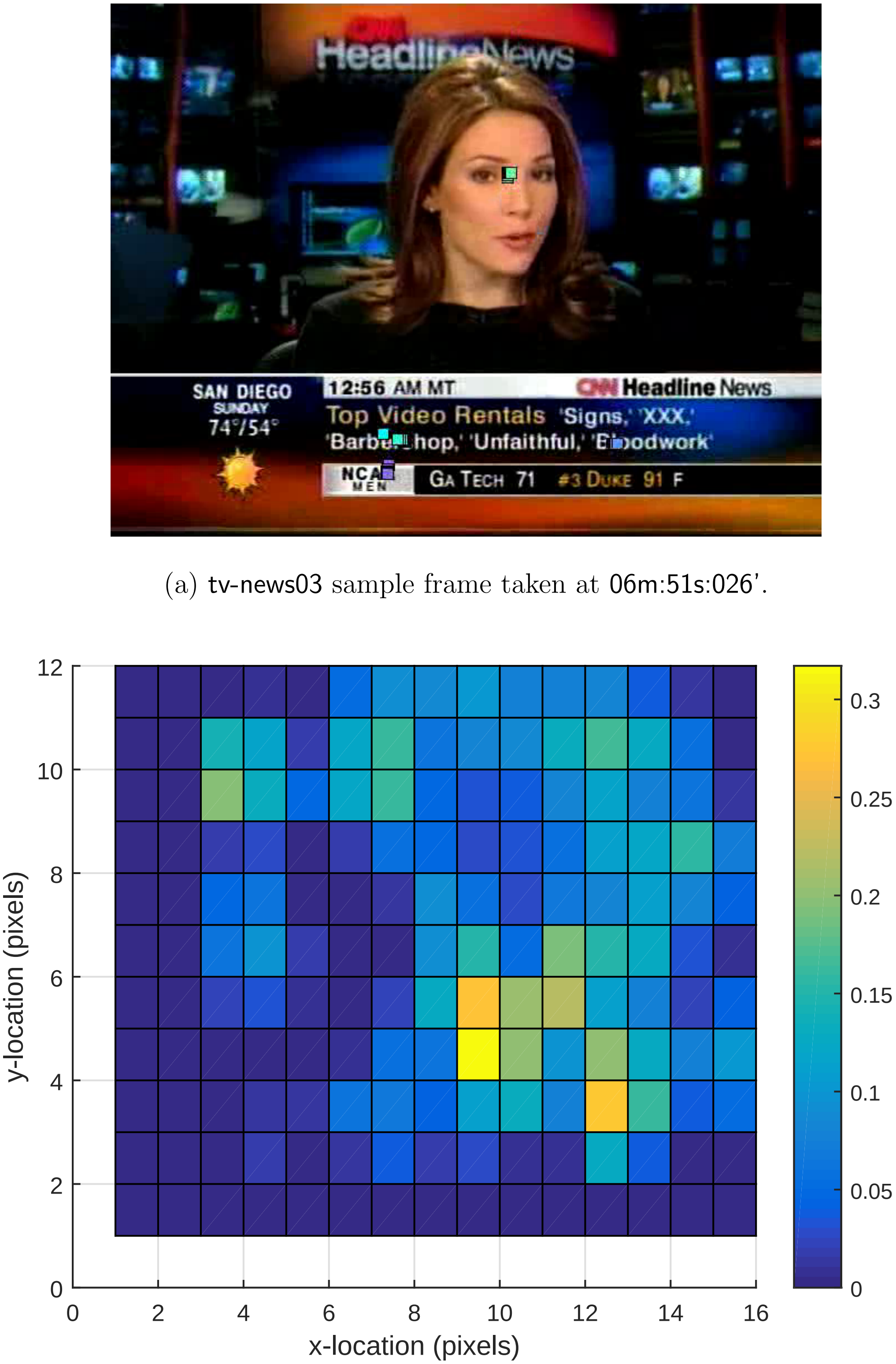


(a) saccadetest sample frame taken at 00m:07s:606'.



(b) saccadetest mutual information given spatial location, across all frames.

Figure 30: saccadetest sample frame alone with mutual information given spatial location



(a) tv-news03 sample frame taken at 06m:51s:026'.

(b) tv-news03 mutual information given spatial location, across all frames.

Figure 31: tv-news03 sample frame alone with mutual information given spatial location

where $P(x_i)$ is the probability mass function of X_k, $P(w_j)$ is the probability mass function of W_k, and $P(x_i, w_j)$ is the associated joint probability mass function.

By restricting W_k, the average of temporal neighbors, to only two frames $F(k+D)$ and $F(k-D)$, we can study the correlation change over time, at distance D from frame $F(k)$. Mutual information between a frame and its direct neighbors (i.e., $D = 1$) is significant, as shown in Figure 32, compared to the information shared with distant frames. Roughly 50% of information is shared between adjacent neighboring frames for all videos, regardless of the content (recall that, as shown in Figure 27, the average information content in an eye-fixation map is about 0.3 bits). However, various video content affect the rate of change. In saccadetest, for example, the rate of change is small (i.e. slow decay), while in tv-ads and tv-sport the rate of change is large (i.e. fast decay). This variability might be attributed to the level of complexity of these videos. More correlated frames in the eye-fixation map of simple stimuli videos (e.g., saccadetest) are caused by subjects fixating on a single target. On the other hand, to comprehend the scene of complex stimuli videos (e.g., tv-ads and tv-sports), subjects are required to exert more effort to examine the scene and actively search for visual information.

Moreover, we compute the correlation between a given frame $F(k)$ and the average of its N direct neighbors. As shown in Figure 33, mutual information between a given frame and its nearest neighbors contains most of the information shared with the rest of the frames. For most categories, the nearest $5-6$ neighbors contain almost all the correlated information in the eye-fixation map. Therefore, including more frames in the neighborhood average does not necessarily add any more useful information. This trend is observed in every category in the dataset, regardless of the content. However, some categories (such as monica and gamecube) yield higher mutual information than other categories, suggesting that the video content makes a difference. For most categories, the mutual information levels-off after 6-8 frames, with the exception of

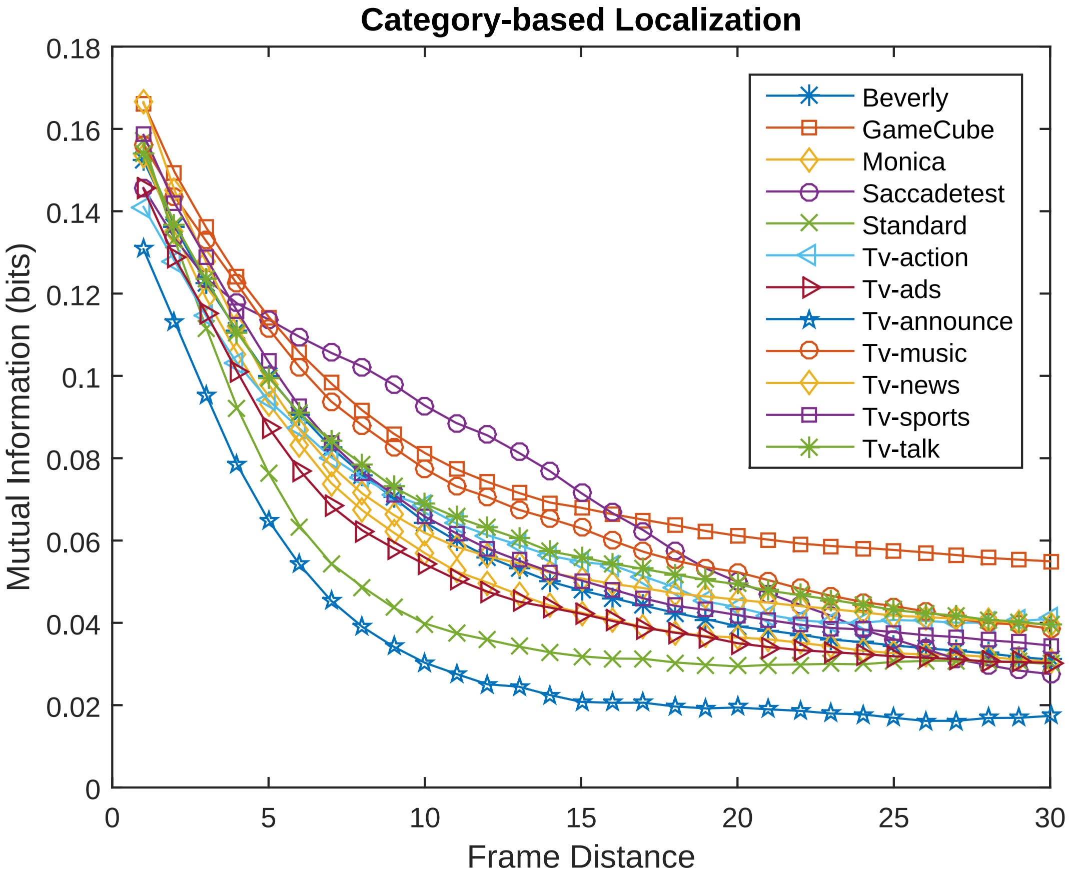


Figure 32: Average mutual information between $F(k)$ and temporal neighbors $F(k+D)$ and $F(k-D)$, where D is the frame distance.

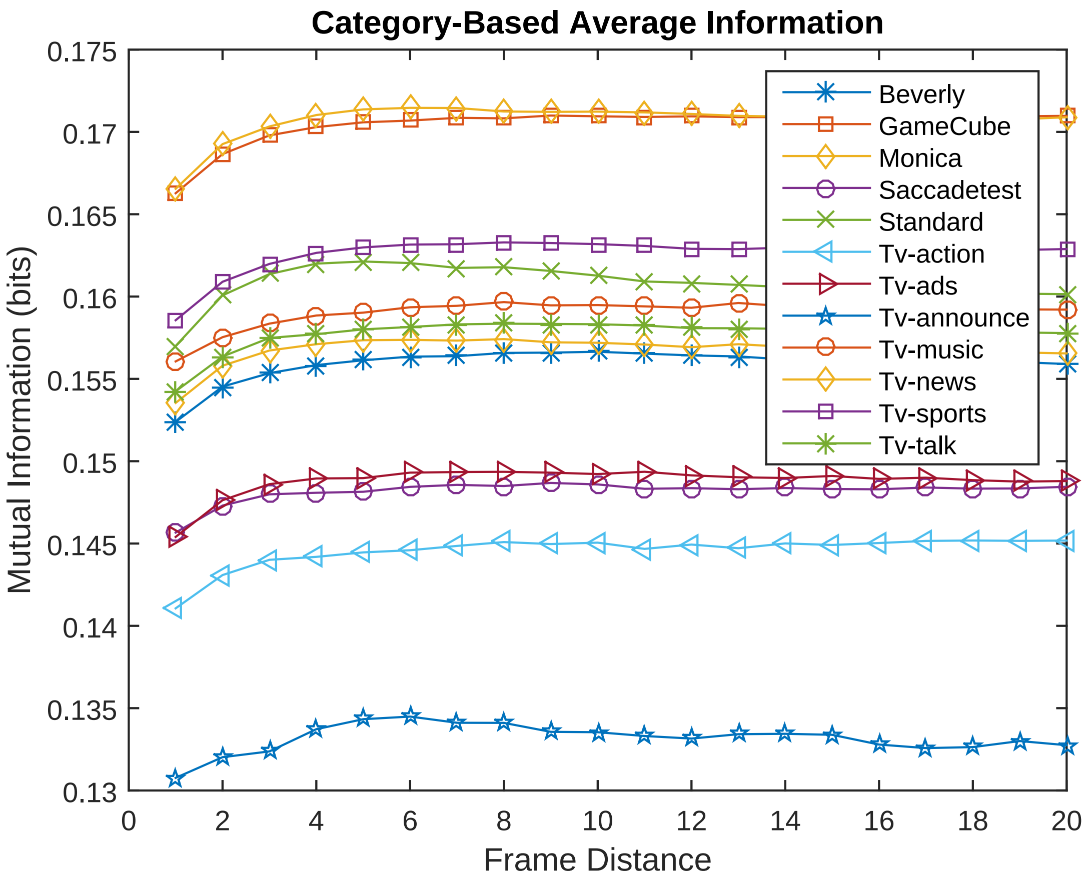


Figure 33: Average mutual information between $F(k)$ and the average of its N temporal neighbors up to a certain frame distance.

standard. This can be attributed to the process of averaging that may cause some mutual information in the nearest neighbors to be marginalized as the number of frames included gets greater.

CHAPTER IV

PREDICTING THE CORRELATION OF EYE-FIXATION MAPS IN VIDEO USING OPTICAL FLOW

In the previous chapter, we showed correlation analysis of eye-fixation data independent from the visual stimuli in order to gain insight into the structure of the eye-fixation data of natural scenes as well as provide us with a better understanding of visual attention mechanisms. However, it would be interesting to see which features in the visual stimuli contributed to such correlation. Correlation analysis of spatial neighbors discussed earlier showed that scene complexity plays an important role in determining the locations of saliency clusters, which is shared between static and dynamic visual stimuli. Scene complexity in images has been shown to be a major contributing factor to saliency [40]. On the other hand, correlation analysis of temporal neighbors shows that temporal complexity affects the correlation of eye-fixation data. The size of correlated neighborhoods in a sequences is inversely proportional to its temporal complexity. Although, the correlation analysis using spatial and temporal neighbors show a snap-shot of motion contribution on eye-fixation data, the analysis does not quantify the motion contribution. In this section, we quantify such contribution using motion cues extracted from optical flow maps and a simple linear regression model.

4.1 Optical Flow Basics

In Optical Flow estimation, we attempt to estimate the 2D velocities for visible surface points in a sequence of images as projected on the image plan from 3D moving surface points in natural scenes [26]. These 2D velocities are typically called 2D motion field, and the goal of optical flow estimation is to compute an approximation to the motion field from time-varying image intensity [38]. The problem of optical flow in its basic formulation is to compute the x and y translation of pixel intensities between frames. That is, given a pixel intensity in time t, $I(x, y, t)$, the following equation holds, given *Brightness constancy* [6]

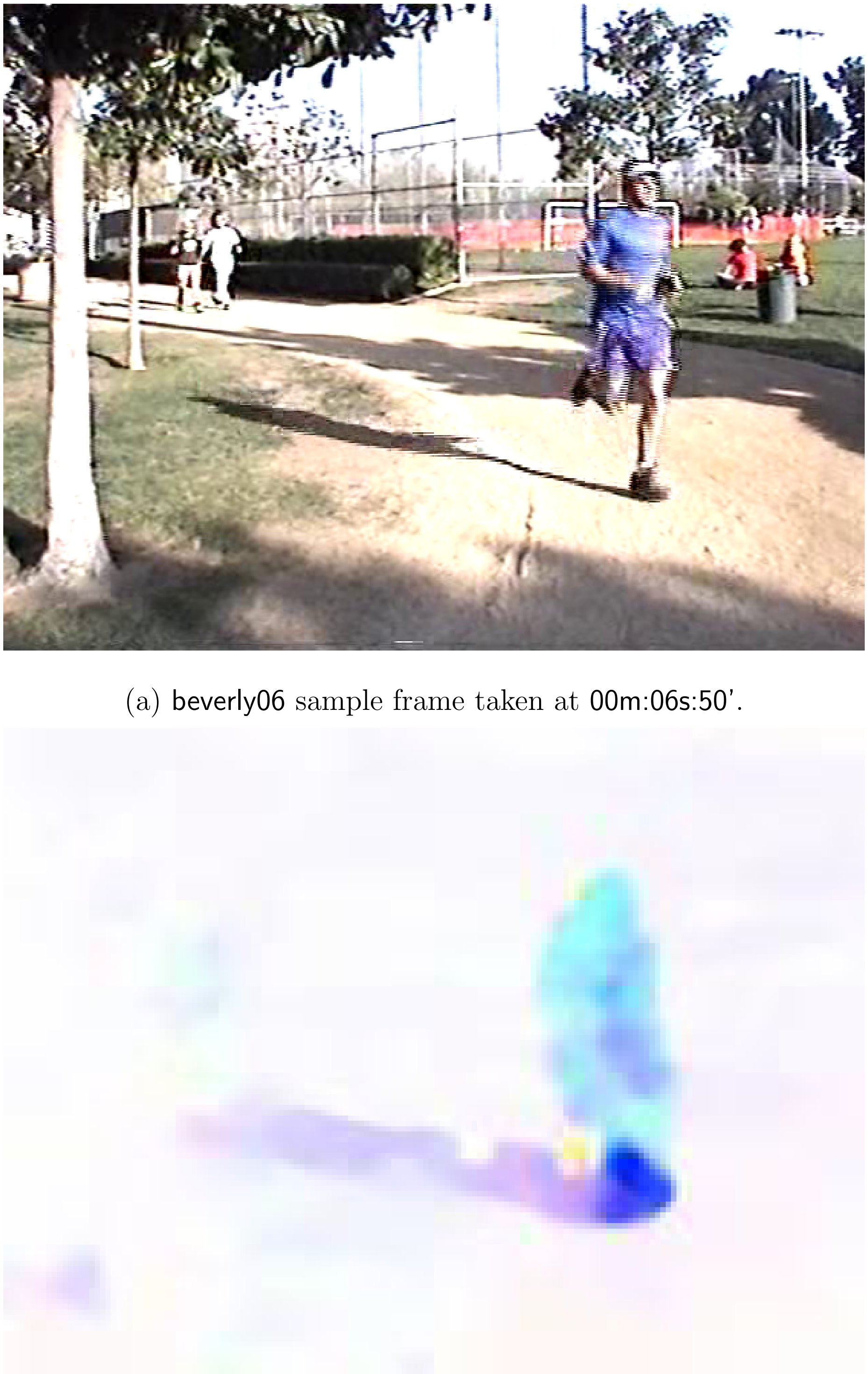
$$I(x, y, t) = I(x + u, y + v, t + 1), \tag{25}$$

which simplifies to *Optical Flow Constraint*

$$u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0. \tag{26}$$

To solve this under-constrained system, optical flow algorithms impose several choices of prior, the majority of which are smoothness priors [6].

Several optical flow algorithms have been proposed, and continue to be, in the last few decades [6]. Two cornerstone approaches to optical flow estimation are Horn-Schunck [39] and Lucas-Kanade [62]. Horn-Schunck method assumes a global smoothness of motion field over the whole image. To solve the flow equation, Horn-Schunck method tries to minimize distortions by solving the associated multi-dimensional Euler-Lagrange equations. On the other hand, Lucas-Kanade takes a local approach to the problem and assumes that the flow is consistent in local patches, then solves the over-constrained system using least squares. More recently, authors of [13] proposed combining global and local smoothing approaches in the so called *combines local-global Method* (CLG). In this paper, we used optical flow estimation implemented by [58], which combines CLG from [13] with warping technique [11] by means of minimizing a non-linearized constancy assumption using fixed point iteration.



(a) beverly06 sample frame taken at 00m:06s:50'.

(b) beverly06 optical flow map generated using publicly available code of [58].

Figure 34: beverly06 sample frame and associated optical flow map

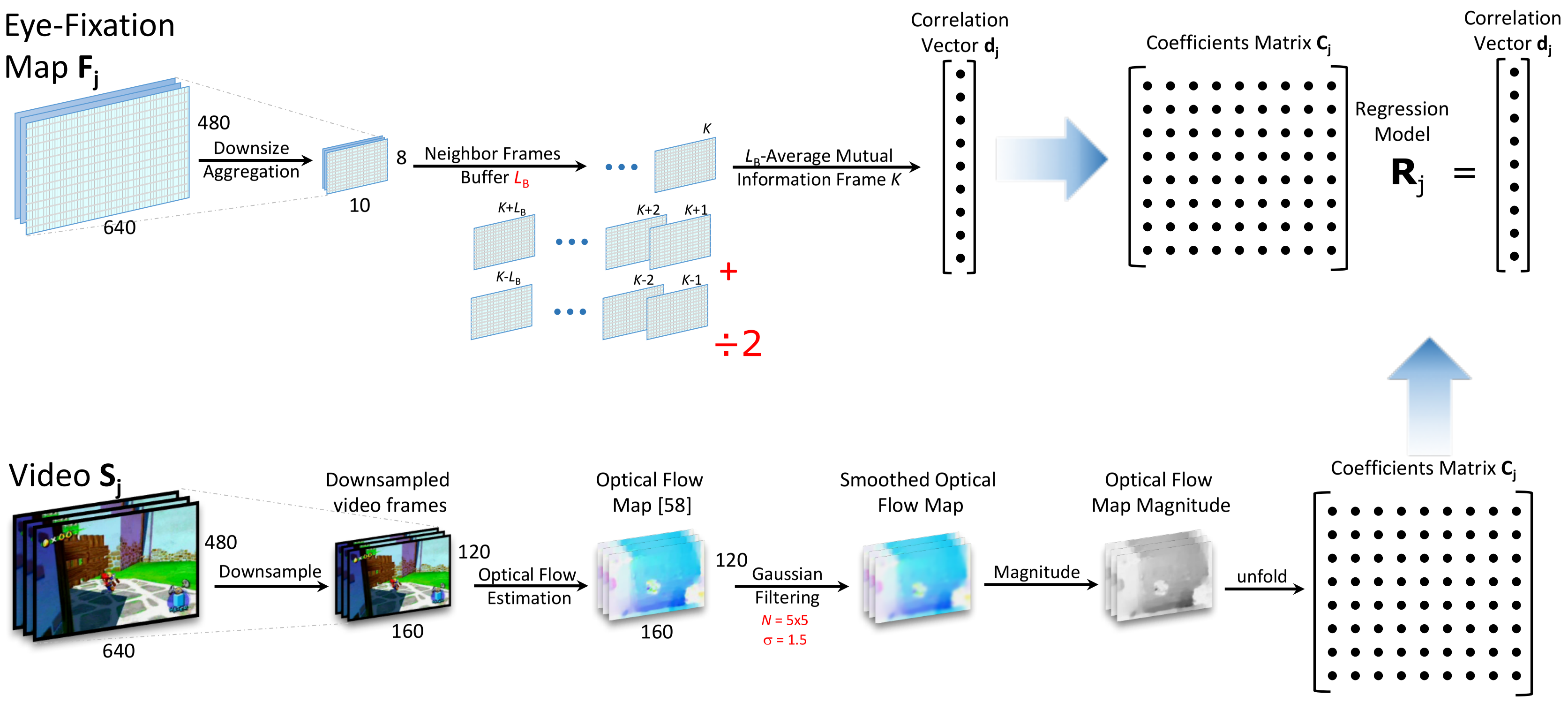


Figure 35: Block diagram of the proposed Algorithm.

4.2 Correlation Prediction using Optical Flow

In this part, we show the possibility of predicting eye-fixation map correlation from video frames using motion cues computed using optical flow. First, we compute correlation vector $\boldsymbol{d}$ of length K from eye-fixation map $\boldsymbol{F}$ of size $M \times N \times K$. This is done by reducing the size of eye-fixation map by a factor α using aggregation in Eq.(18). This aggregation affects the sparsity and variability of the eye-fixation map, which ultimately affects model fitness. Then, we compute mutual information between frame k and the pixel-wise arithmetic mean of adjacent frames $F(k+D)$, where $D \in \{\pm 1, \pm 2, \pm 3, \dots\}$, according to Eq.(24), for all values of $1 \leq D \leq L_B$, where L_B is frame buffer length. Different values of frame buffer length affects the spread of data points over the correlation range, as will be highlighted in the next section. Finally, the k^{th} entry in $\boldsymbol{d}$ correlation vector is computed by averaging all mutual information values for frame k and all frame pairs in frame buffer. As for coefficient matrix $\boldsymbol{C}$ of size $K \times W$, we generate motion cues, i.e optical flow, using [58]. Given video $\boldsymbol{S}$ of size $M \times N \times K$, first, we downsample video frames by a factor β, then compute optical flow map. Afterwards, we use Gaussian filter to smooth out optical flow map discontinuities, which are due to compression and interlacing artifices in the video dataset [47]. Finally, k^{th} row of coefficients matrix $\boldsymbol{C}$ is produced by computing the magnitude of the optical flow of map, then, unfolding frames k, $k+1$, and $k-1$ into a single row.

To quantify the relation between motion cues computed from video frames and correlation between eye-fixation map frames, we use linear least squares method to fit coefficient matrix $\mathbf{C}$ to correlation vector $\mathbf{d}$. Linear least squares is one of the simplest, yet effective, data fitting approach in literature. Linearity assumption is reasonable since the majority of systems and processes are inherently linear or can be approximated by a linear model reasonably well. Another advantage of using a linear model is to reduce the possibility of data over-fitting compared to nonlinear models.

Therefore, we compute the optimal values of regression model $\mathbf{x}$, in the least square sense [35], such that misfit error is minimized:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \left\| \mathbf{C}\mathbf{x} - \mathbf{d} \right\|_2 , \tag{27}$$

where $\mathbf{C}$ is the coefficient matrix extracted from optical flow map, $\mathbf{d}$ is the correlation vector computed from eye-fixation map. The analytical solution for the regression model $\mathbf{x}$ is:

$$\mathbf{x} = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{d}. \tag{28}$$

4.3 Regression Fitting Results

Using data model defined in the previous section, we process all 50 video in CRCNS dataset [47] and compute the predicted frame correlation from motion cues and compare it to that computed from mutual information of eye-fixation map frames. In Figure 36, we show a plot of correlation results where every video is represented using a single point, for which the x-coordinates are the average frame correlation computed from the eye-fixation map and y-coordinates are the average frame correlation predicted by motion cues from optical flow map. As seen in Figure 36, most points are close to the 45-degree line indicating that there is, indeed, strong correlation between mutual information of different frames of the eye-fixation map and motion cues in most videos. In addition to average frame correlation results, we show frame correlation results for eight different videos (four close to the 45-degree line, in Figure 37, and another four away from the 45-degree line, in Figure 38). Each of the bottom plots represent a single video, in which every point represent a frame in that video.

In the first four videos, saccadetest, standard05, tv-talk05, and tv-announce01, we notice a good spread of the data across frame correlation range, where saccadetest and tv-talk05 are gravitated towards higher correlation values due to their simple semantic content while standard05 and tv-announce01 are generally lower in correlation values

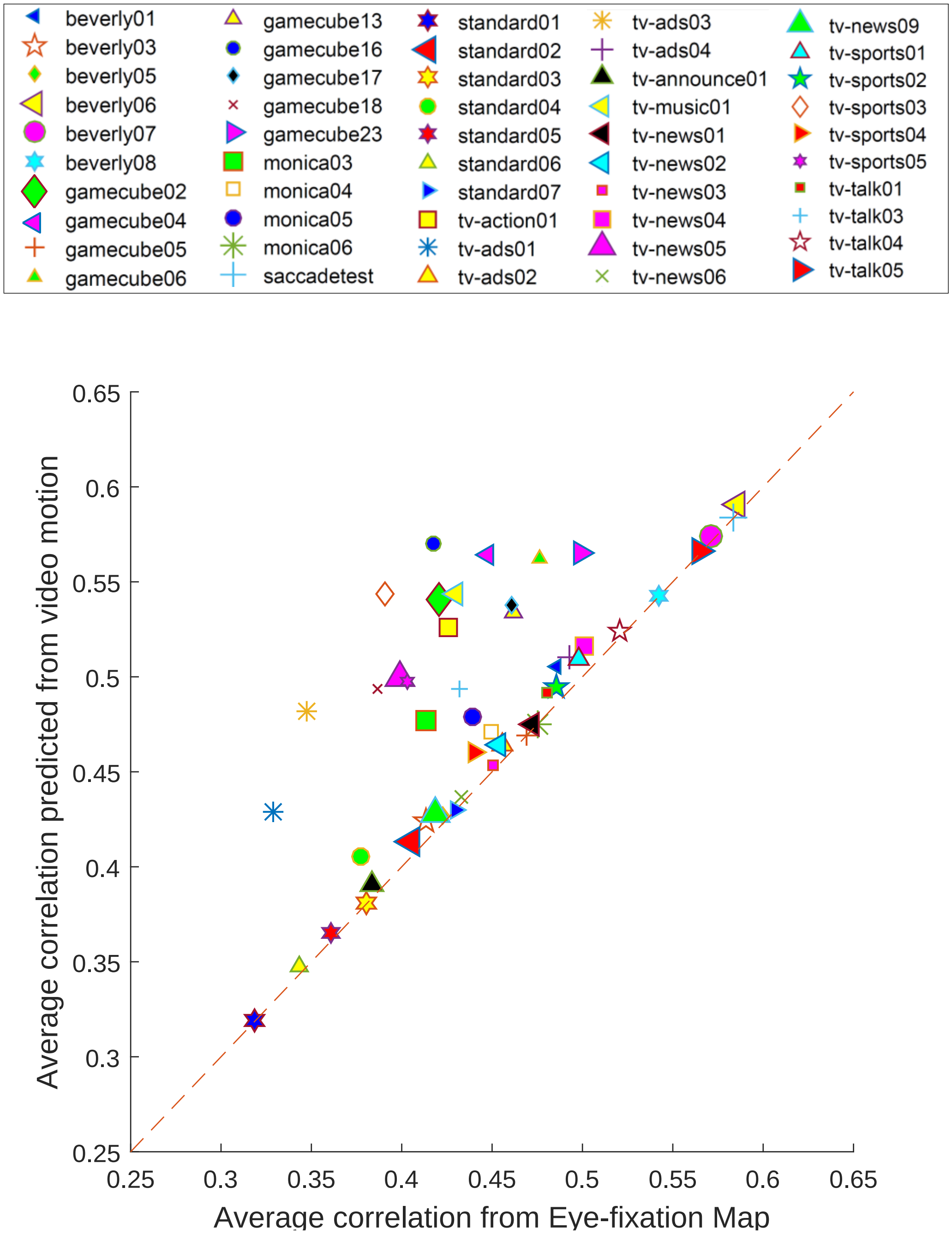


Figure 36: Each video in the dataset is represented as a single point with average frame correlation computed from 8×10 eye-fixation map (x-axis) and average predicted correlation from 120×160 optical flow map (y-axis)

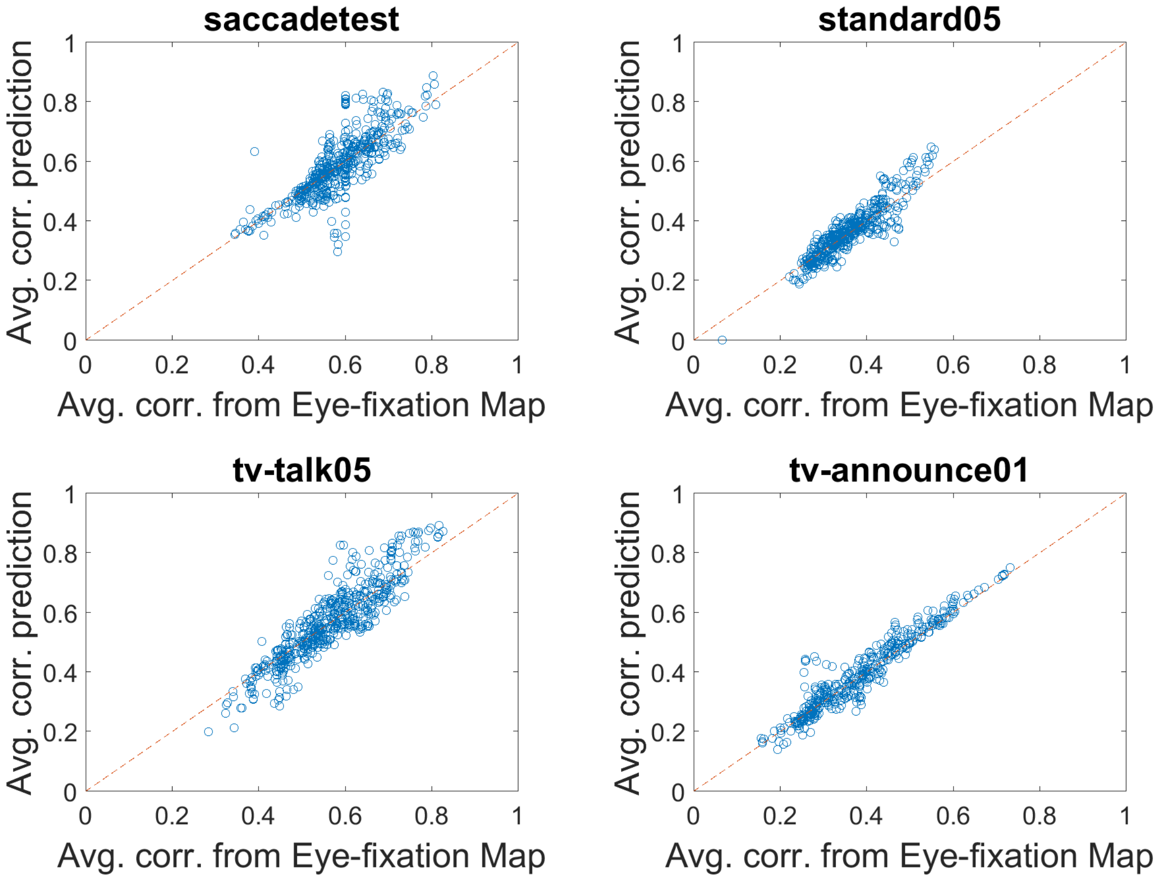


Figure 37: Frame Correlation results of the proposed prediction algorithm.

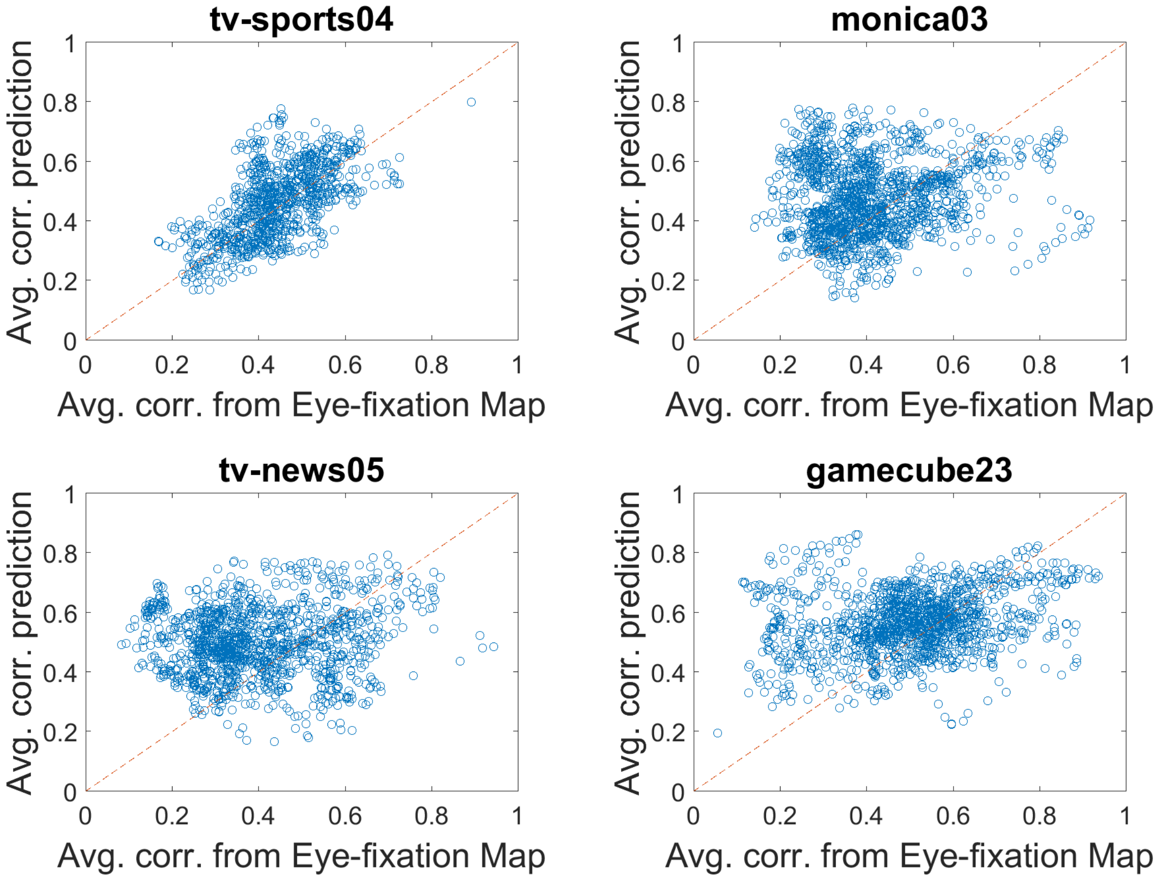


Figure 38: Frame Correlation results of the proposed prediction algorithm.

due to their constantly unique information presented in every frame. Also, we can see that even though the predicted values have maximum misfit error of $10 - 20\%$, the model represent the data well since majority of points fall in a narrow ellipsoid around the 45-degree line. As for the second half of the bottom plots, in tv-sports04, monica03, we notice that the average frame correlation predictions are further away from the 45-degree line compared to the previous four cases due to wider spread of prediction results and the bulk of misfit error in $20-40\%$ range. This might be caused by major motion events biasing the optical flow map, thus, small motion cues related to the main object of interest in these two videos are missed. In tv-sports04, motion of basket players and news graphics obstruct the motion of the small basketball (the main object of attention). Similarly, in monica03, cars and buses motion obstructs motion of hand gestures of the police officers. Next, tv-news05 and gamecube23 are among the videos with correlations furthest away from the 45-degree line. As evident from the predictions scatter plot, the model basically failed to reliably predict frame correlation, where maximum misfit error reaches up to 70%. When closely examining the content, one can observe that tv-news05 contain many high-level features that are not explained by motion, which is discussed in more details in *High-Level Features* section below. gamecube23, on the other hand, suffers from a different problem; a combination of *Photographers Bias* [8] and background motion perpendicular to image plane, details provided in *Camera Motion* section below.

4.3.1 Effect of Frame Buffer Size on Model Fitness

To examine the effect of frame buffer size on the results, we show, in Figure 40, the results of model fitting proposed in earlier for frame buffer lengths $L_B = 3$, 7, 11, and 21. As seen in Figure 40, as the frame buffer size increases, the average frame correlation of videos start to spread across wider range. This is because considering more neighboring frames of eye-fixation map when computing frame correlation makes it easier to distinguish unique eye-fixation patters. In contrast, examining only direct neighbors makes eye-fixation patters appear more random, and as a result, it makes majority of eye-fixation map frames look the same.

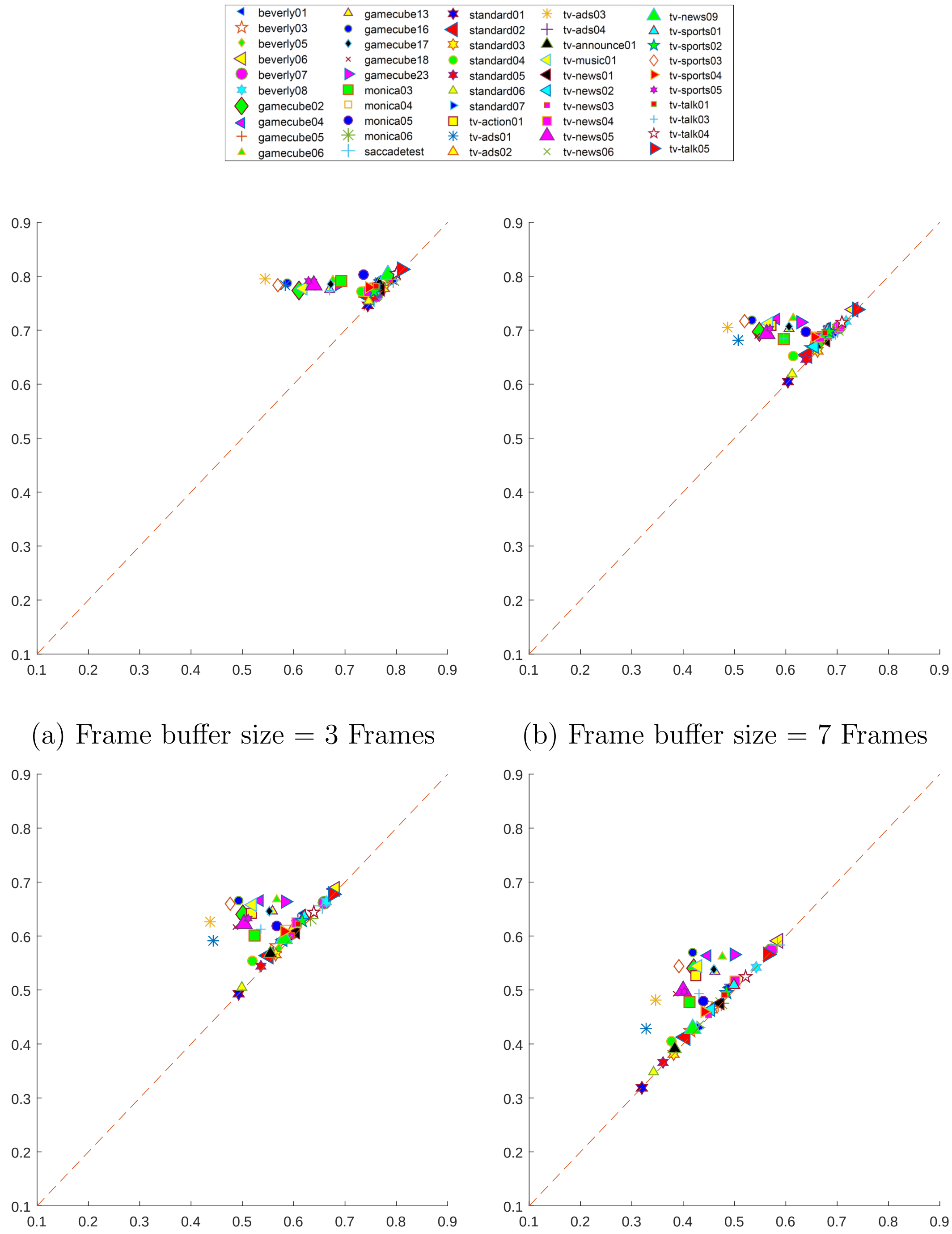
4.3.2 Effect of Eye-fixation Map-Aggregation Size on Model Fitness

We examine the effect of Eye-fixation maps size on the results. In Figure 42, we observe that data points fitting get better as we increase the size of the eye-fixation map used in computing correlation values. This might be due to the fact that meaningful data structures are lost during the aggregation process to generate extremely small eye-fixation map as the case in 3×4 maps.

As the size of eye-fixation map gets larger, fitting algorithm converges to model the data structure reasonably well, as the case when size= 6×8 and 8×10. Furthermore, as eye-fixation map size gets even larger, the data fitting parameters start to over-fit and generate predictions with zero misfit errors, as the case with 12×14 maps. Another aspect to consider when examining this behavior is the fact that as aggregation process reduces the size of the eye-fixation map, the variability of resultant map entries increases compared to original size map. Thus, the effect of eye-fixation map size of model fitness can be due to the variability of the smaller sizes of the eye-fixation map, hence, struggle of data fitting algorithm to capture suitable representation of the smaller sizes of the eye-fixation maps compared to the larger ones.

4.3.3 Effect of High-Level Features on Model Fitness

In some videos in CRCNS database [47], scenes mainly consist of a single actor with little to no motion in the scene, as in tv-news05 shown in Figure 43(a). As seen in the *Regression Fitting Results* section, the proposed data fitting algorithm is not able to converge to a suitable model for tv-news05. We believe this is caused by the video scene, which consists of a news anchor speaking with screen-bottom messages



(a) Frame buffer size = 3 Frames

(b) Frame buffer size = 7 Frames

(c) Frame buffer size = 11 Frames

(d) Frame buffer size = 21 Frames

Figure 39: Effect of frames buffer size on frame correlation predictions. Each video in the dataset is represented as a single point with average frame correlation computed from 8×10 eye-fixation map (x-axis) and average predicted correlation from 120×160 optical flow map (y-axis).

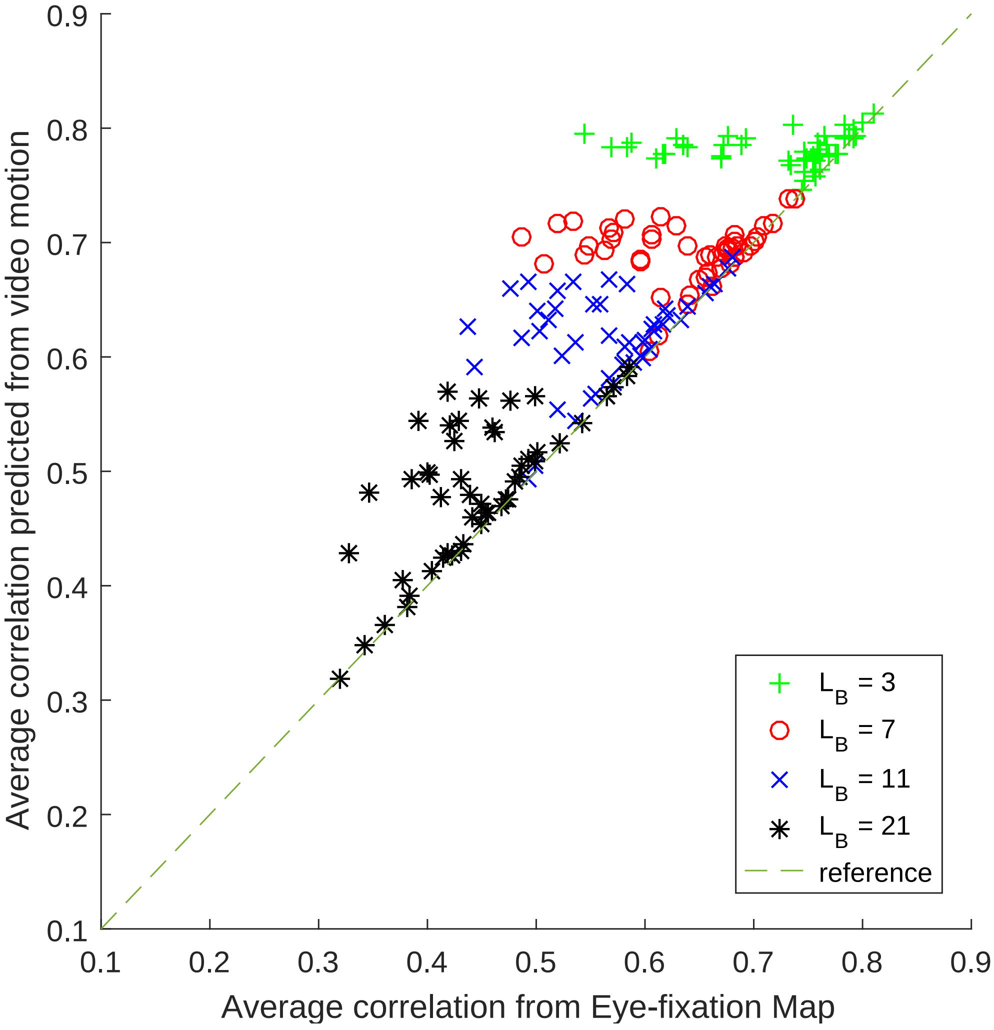
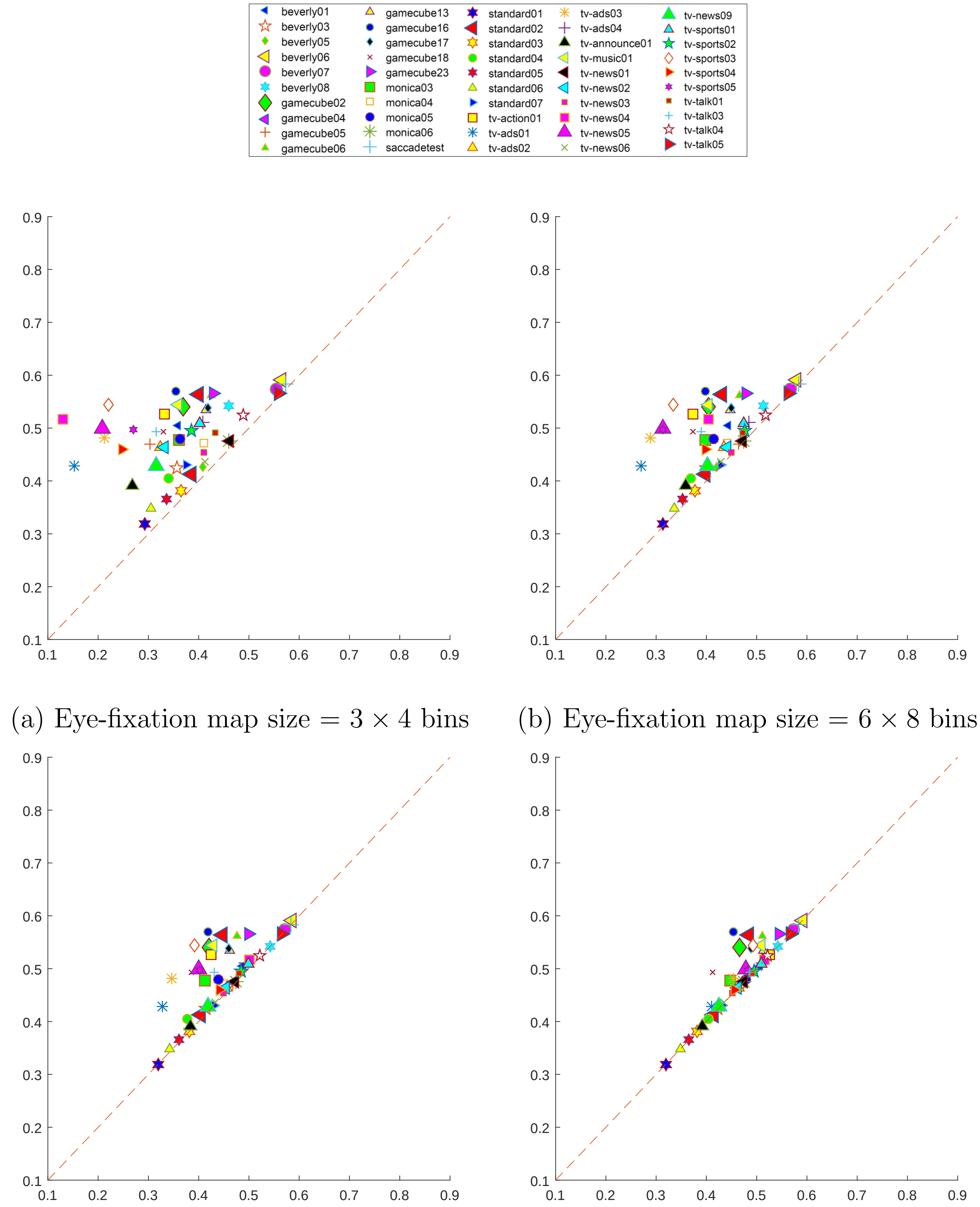


Figure 40: Effect of frames buffer size on frame correlation predictions. Eye-fixation map size is 8×10.



(a) Eye-fixation map size = 3 × 4 bins (b) Eye-fixation map size = 6 × 8 bins

(c) Eye-fixation map size = 8 × 10 bins (d) Eye-fixation map size = 12 × 16 bins

Figure 41: Effect of Eye-fixation map aggregation size on prediction results. Frame buffer size for all plot $L_B = 21$ Frames.

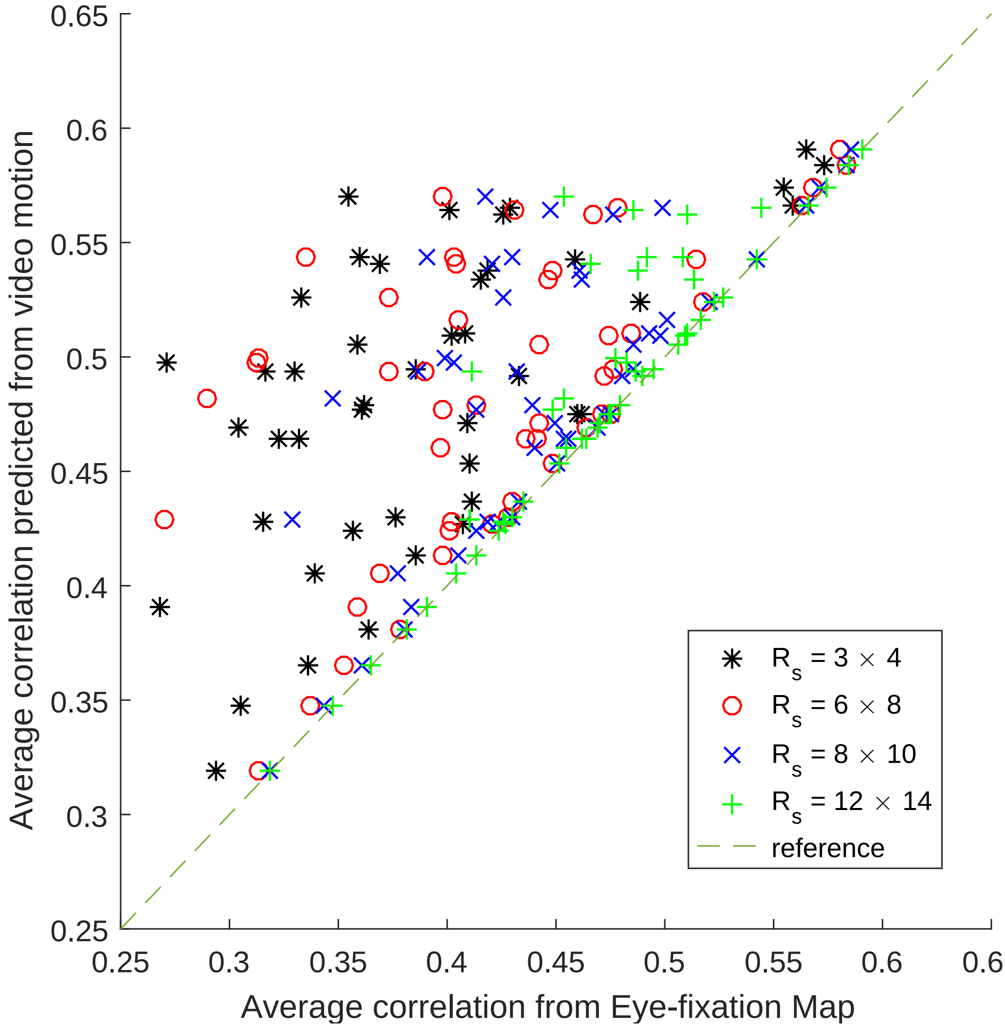
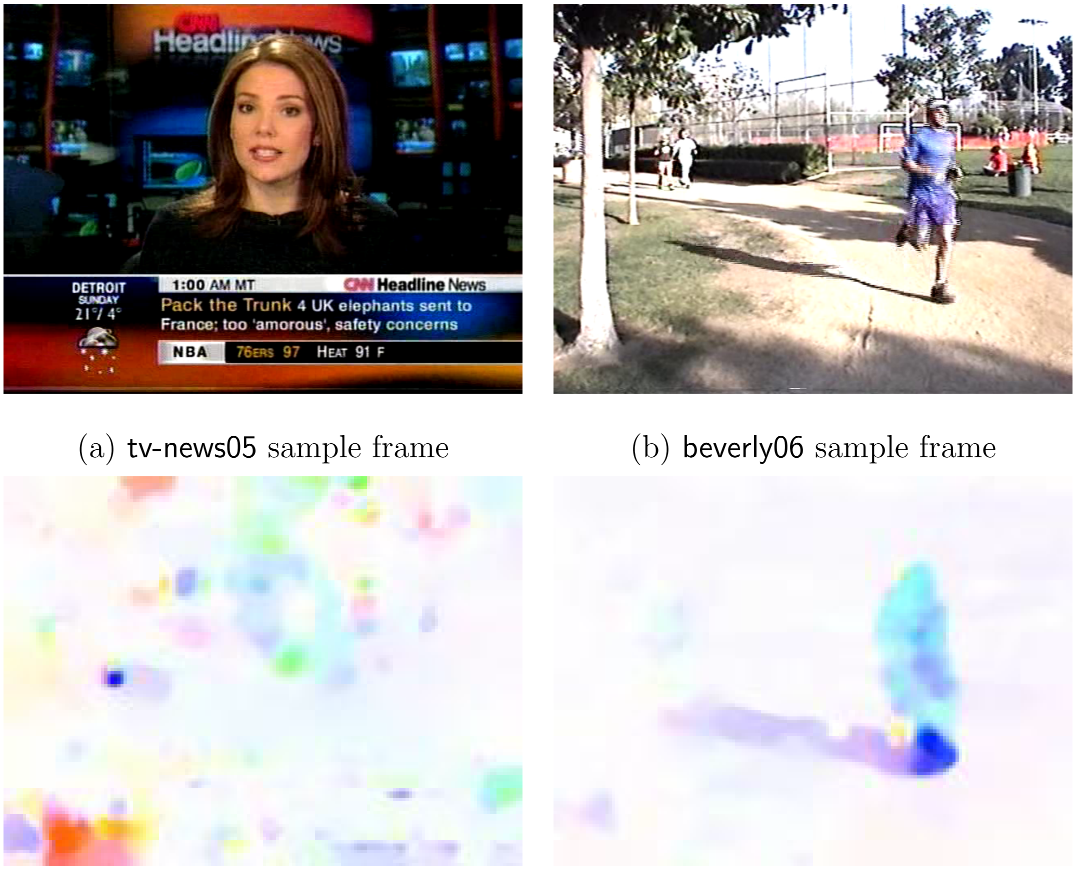


Figure 42: Effect of Eye-fixation map aggregation size on prediction results. Frame buffer size $L_B = 21$ Frames.



(a) tv-news05 sample frame

(b) beverly06 sample frame

(c) tv-news05 optical flow

(b) beverly06 optical flow

Figure 43: Attention to high-level saliency stimuli suppresses the saliency of motion.

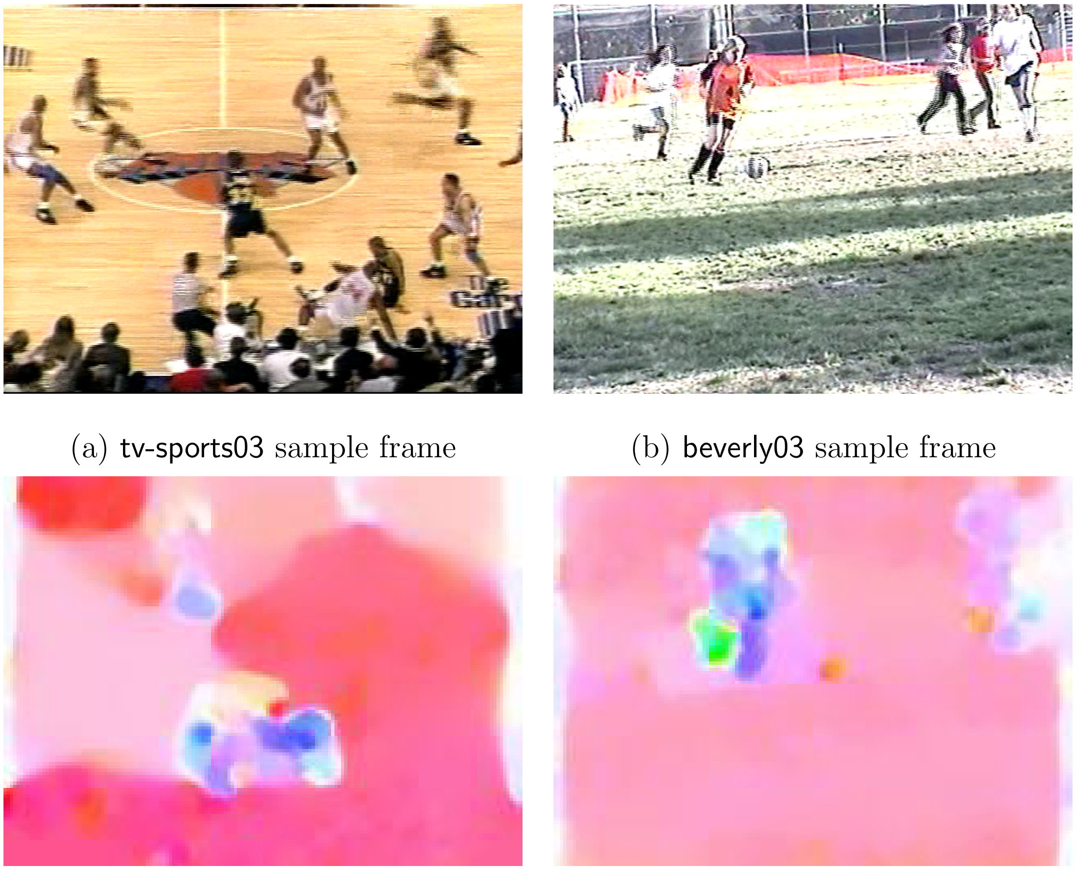
and banners changing every few seconds, being dis-proportionally filled with high-level saliency stimuli such as text and faces. Research has shown that faces and text attract gaze of HVS independent of the task given to subjects and play a major role in explaining the visual attention patterns [14]. In fact, including face detection as part of saliency detection algorithm improves results greatly [15]. Therefore, it would be expected that visual attention attributed to high-level saliency stimuli are not captured properly using motion only. On the other hand, eye-fixation correlation in scenes with low semantic content, such as beverly06 shown in Figure 43(b), can be predicted reliably using motion-based models. In these type videos, the scene context is entirely explained by motion and, thus, attention can be modeled extremely well. In the case of tv-news05 and beverly06, simply examining the optical flow map, Figure 43(c) and Figure 43(d), shows that optical flow map of beverly06 highlights the most prominent object in the scene, while optical flow map of tv-news05 has no identifiable structure and conveys no information about the scene. On contrary, optical flow map of tv-news05 highlights the weather animation in the lower left corner which has no semantic meaning and is of little relevance to tv-news05 context.

4.3.4 Effect of Motion Complexity on Model Fitness

Even in cases where motion plays a major role in explaining the video, the complexity of motion and subjects' prior information can affect the predictability of visual attention in these videos. For example, tv-sports03 (scene from TV coverage of basketball game) shown in Figure 44(a) and beverly03 (scene of amateur soccer game) shown in Figure 44(b) both are sports scenes. However, according to prediction results earlier, beverly03 has a much simpler model to predict compared to tv-sports03. We believe this is due to complexity of motion in tv-sports03 where different player constantly change their positions across the basketball court to optimize their game play. In tv-sports03, the attention of subjects is mainly concentrated on the small basket ball rather than the players, which is not properly highlighted in tv-sports03 optical flow map. Also, since tv-sports03 is a record of professional game, subjects' prior information, such as players or teams recognition, might be affect their viewing patterns. Additionally, tv-sports03 is a TV coverage of sport event, so subjects might be instinctively looking for score board or remaining time clock which they could be familiar with from previous experience. On the other hand, beverly03 does not have similar problems nor any associated prior information available to subjects. Therefore, even though optical flow maps of tv-sports03 and beverly03, shown in Figure 44(c) and Figure 44(d) respectively, look similar in capturing semantically significant motion in both scenes, their predictability is affected by motion complexity and subjects prior information. Naturally, such explanation is far from conclusive and further experiments and annotated videos are required to verify the effect of prior knowledge on the predictability of self-correlation using motion. It is important to note here that both scenes suffer from camera motion and non-stationary backgrounds, however, this fact is more relevant to the discussion next.

4.3.5 Effect of Camera Motion on Model Fitness

In addition to factors discussed earlier, camera motion plays a considerable role in controlling viewing patterns. Let us consider gamecube02 (a scene of video game play), shown in Figure 45(a), and standard05 (a scene of camera sweep recording a social event), shown in Figure 45(b), to examine camera motion effect on the predictability of subjects viewing patterns. In gamecube02 as the player moves the character throughout the video game, the in-game camera system keeps track of the changes and constantly centers the character in middle of the screen, known as *Photographers Bias* [8]. Such setting leads optical flow map of gamecube02, shown in Figure 45(c), to label all the surrounds of the video game character as in-motion, while the character itself is relatively static, completely the opposite of the viewing patters of subjects.



(a) tv-sports03 sample frame

(b) beverly03 sample frame

(c) tv-sports03 optical flow

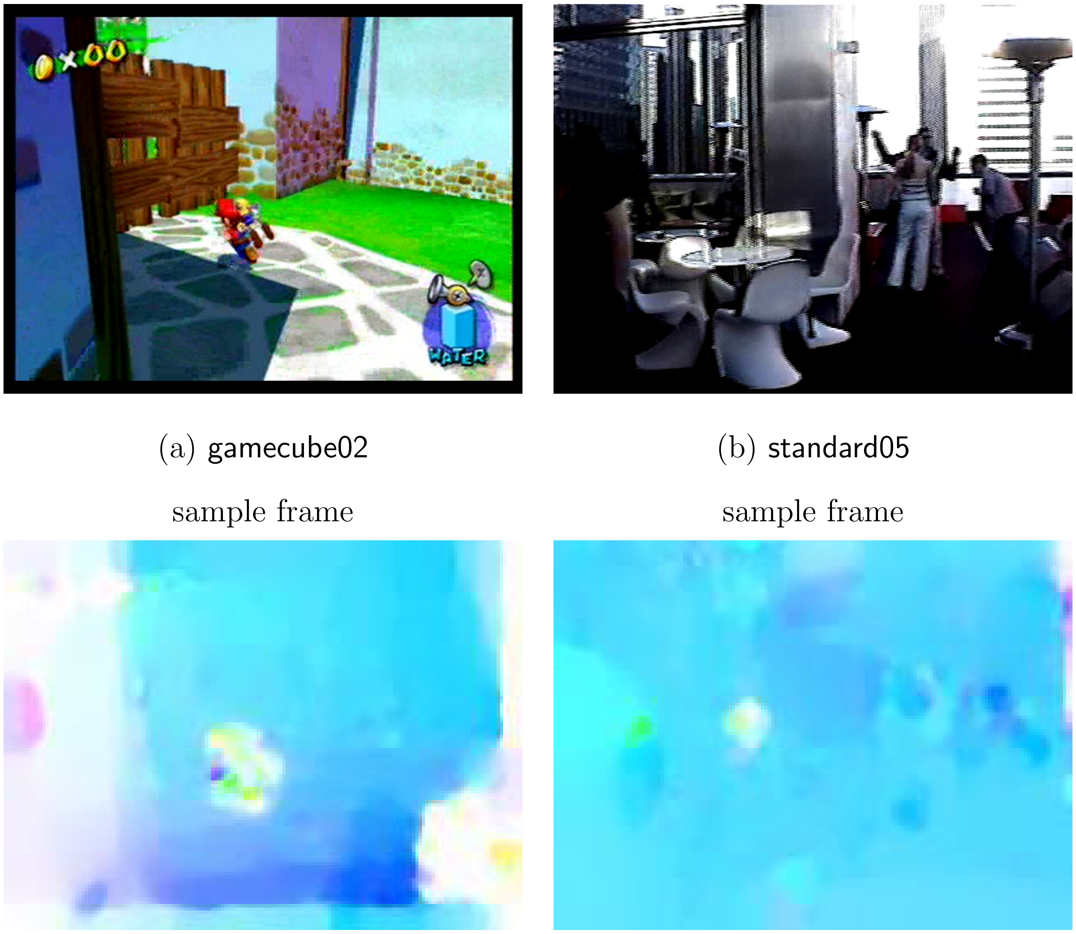
(b) beverly03 optical flow

Figure 44: Motion complexity and subjects prior information affects the predictability of eye-fixation map frames correlation

Table 1: Summary of the factors that affect the predictability of eye-fixation correlation from motion cues

Factors		Effect on the prediction performance
Algorithm Parameters	*Regression Model*	**More** complex models would **improve** prediction performance at the cost of **overfitting**
	Frame Buffer	**Longer** buffer length would **improve** prediction performance and **reduce** clustering effect
	Aggregation Size	**Larger** aggregated eye-fixation maps **improve** prediction performance at the cost of **overfitting**
Content Parameters	*High-level Features*	Presence of higher-level features **reduces** prediction results
	Motion Complexity	**Less** complex motion **improve** prediction results
	Camera Motion	Camera motion **parallel** to image plane **improve** prediction results

Camera motion that is perpendicular to image plane cannot be interpreted easily by computer vision algorithms and certainly not by optical flow field estimation. On the other hand, camera motion parallel to and perpendicular to salient objects can be easily modeled as in the case of standard05 as the mosaic shown in Figure 46. In this video, subjects attention is motivated by continuous flow of new information about the scene details, which is caused by camera motion. Hence, standard05 prediction results based on motion cues closely resemble the correlation in the eye-fixation data.



(a) gamecube02

sample frame

(b) standard05

sample frame

(c) gamecube02

optical flow map

(b) standard05

optical flow map

Figure 45: camera/background motion with fixed main actor vs. camera/background motion parallel to image plane

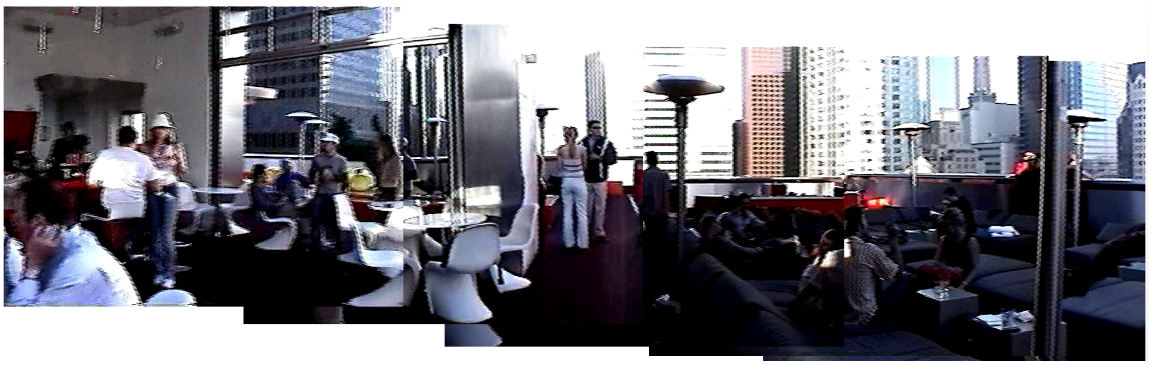


Figure 46: Mosaic of standard05 frames showing motion parallel to image plane.

CHAPTER V

UNSUPERVISED VIDEO FEATURE FOR ESTIMATING UNCERTAINTY

5.1 Uncertainty-based Framework for Video Saliency Applications

In an attempt to mimic the advanced processing capability of the human vision system (HVS), saliency detection has been incorporated into various image and video processing algorithms for improved performance. The diversified applications include but are not limited to compression [32], segmentation [87], object recognition [74], tracking [63], and quality assessment [96]. However, there has been no explicit design of a saliency-based video processing framework, to the best of our knowledge. Most of the proposed methods do not evaluate the validity of saliency maps generated online, but rather design or choose a saliency detection algorithm that exhibits good performance in evaluation datasets, and then hope for the best when the algorithm goes online. Therefore, we propose a unified framework for enhancing video processing algorithms using saliency that is neither application- nor algorithm-specific and can be reliable in real world scenarios. The proposed uncertainty-based framework is depicted in Figure 47. It evaluates the saliency map and produces associated uncertainty map that describes the level of confidence in the generated saliency map. By reliably estimating uncertainty, we can expand the framework to include a systematic decision-making procedure that makes application-specific decisions. Additionally, having a separate module for decision making helps clarify which assumptions are application-specific and which ones are saliency related. Knowledge about the application space can influence the design of this module without making drastic changes

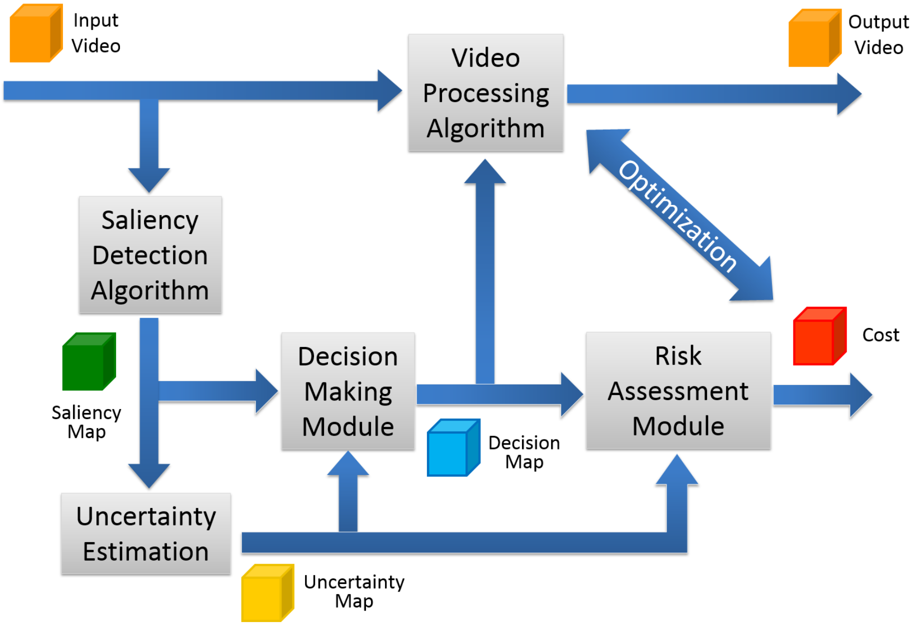


Figure 47: Uncertainty-based framework for improving saliency-enabled video processing algorithms

to the whole framework. Similarly, the availability of uncertainty estimations allows for risk assessment that can be used to guide the optimization of video processing algorithms.

5.2 Unsupervised Uncertainty Estimation Using Spatiotemporal Cues

As discussed in the previous chapter, pixels in eye-fixation maps are correlated and such dependency can be exploited to identify unlikely occurrences in the computational saliency maps. Basically, we assume that visual saliency is consistent and changes in saliency values happen gradually. Thus, sudden changes in saliency value should lower our trust in that particular spatiotemporal event. Thus, saliency map pixels that are significantly different from their neighborhood are most likely uncertain and should be examined more carefully.

However, the size of local neighborhoods crucially depends on the video content.

For example, fast action videos would most likely have a small group of contiguous correlated pixels, in the saliency map, around location of the main scene actor. In contrast, a slow changing scene gives the viewers more freedom to explore different parts of the video frame, thus, the corresponding eye fixation map would have a larger group of pixels that are correlated. Therefore, it is important to include uncertainty cues from the appropriate scales in order to more reliably capture context-based events.

In most video saliency detection algorithms, the processing of video frames usually consumes significant computation time. Hence, a common practice is to resize the input video frames to several sizes and define saliency maps generated in terms of the frame scale. It is worth noting that saliency maps generated from size-reduced video frames differ from saliency maps downsampled from saliency maps of higher scale. In the first case, video details lost in the downsampling process are not included in the downsampled saliency map, while in the second case, downsampled saliency maps still maintain such details. Generally, uncertainty estimation should take advantage of saliency maps of multiple scales to enhance the estimation performance. One way to approximate the contribution to uncertainty estimation from different scales is to generate a multi-scale uncertainty map that is a weighted combination of uncertainty generated from different scales. In this paper, we focus our study on how to estimate uncertainty from a single scale.

Formally, given a saliency map $\boldsymbol{S}^{(d)}$ of scale d and size $M \times N$ and of depth K frames, we seek to estimate an uncertainty map $\boldsymbol{U}^{(d)}$ of the same scale, size and depth as $\boldsymbol{S}^{(d)}$ that is roughly approximated by saliency value divergence from spatiotemporal local neighborhood mean. The estimation is efficiently computed by processing the map $\boldsymbol{S}^{(d)}$ according to Eq.(29):

$$\boldsymbol{U}^{(d)} = \gamma \left| \alpha \boldsymbol{S}^{(d)} * W^{L_1 \times L_2 \times L_3} \right|, \tag{29}$$

where $|.|$ is the operation to find the absolute value and $d = 1, 2, ...D$ is the scale label, $L_1 \times L_2 \times L_3$ is the size of the spatiotemporal kernel $W^{L_1 \times L_2 \times L_3}$, α is a scaling factor for the saliency map to fix its range to be [0,1], and γ is a scaling factor for the uncertainty map to ensure the output range is [0,1]. In this paper, we use a simple averaging kernel defined as follows

$$W^{L_1 \times L_2 \times L_3} = \begin{cases} \frac{R-1}{R} & \text{at the center} \\ -\frac{1}{R}, & otherwise, \end{cases} \tag{30}$$

where $R = L_1 \times L_2 \times L_3$. The design of $W^{L_1 \times L_2 \times L_3}$ can be viewed as the difference between saliency value and a moving average window of size $L_1 \times L_2 \times L_3$. With appropriate size, $W^{L_1 \times L_2 \times L_3}$ can follow the changes in the scene and, to some extent, approximates the common trend of pixel saliency change over time.

In order to systematically analyze spatiotemporal uncertainty estimation, we study the contribution of spatial neighbors separate from temporal neighbors which might lead to a better understanding of spatial context in saliency maps. Thus, we introduce in the following subsections two special cases of the proposed algorithm: uncertainty estimation from temporal cues and uncertainty estimation from spatial cues. Relying only on temporal neighbors, the proposed algorithm estimates the uncertainty of a pixel in frame k by studying its correlation with its neighbors in the same location across all K frames, as we have proposed in [4]. By dividing the saliency map into temporal neighborhoods, we can treat each pixel location as separate 1-D signal that can be processed using a simple 1-D filter of length L_t to calculate pixel-neighborhood divergence. Similarly, we can divide the saliency map into spatial neighborhoods that span $L_{s_1} \times L_{s_2}$ pixels in a single frame, as we have reported in [3].

5.2.1 Uncertainty Estimation from Temporal Cues

For a given saliency map $\boldsymbol{S}$ of size $M \times N$ and of depth K frames, we decompse the map into 1-D signals as follows

$$\boldsymbol{S} = \begin{bmatrix} s[1,1] & s[1,2] & \dots & s[1,n] & \dots & s[1,N] \\ s[2,1] & s[2,2] & \dots & s[2,n] & \dots & s[2,N] \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ s[m,1] & s[m,2] & \dots & s[m,n] & \dots & s[2,N] \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ s[M,1] & s[M,2] & \dots & s[M,n] & \dots & s[M,N] \end{bmatrix}, \tag{31}$$

where $m = 1, 2, ..., M$, $n = 1, 2, ..., N$, are the spatial coordinates of the saliency map.

We seek to construct an uncertainty map $\boldsymbol{U}$ of the same size and depth as $\boldsymbol{S}$ by iteratively processing 1-D signals $\boldsymbol{s}$ located at saliency map pixel $[m, n]$ according to

$$U[m,n] = \gamma \Big| \alpha S[m,n] * W^{L_t} \Big|, \tag{32}$$

where $m = 1, 2, ..., M$, $n = 1, 2, ..., N$, are the spatial coordinates of both the saliency map and uncertainty map, α and γ are scaling factors, and W^{L_t} is the temporal filter of length L_t, defined by

$$W^{L_t} = [\frac{-1}{L_t} ... \frac{-1}{L_t}, \frac{L_t - 1}{L_t}, \frac{-1}{L_t} ... \frac{-1}{L_t}], \tag{33}$$

5.2.2 Uncertainty Estimation from Spatial Cues

Similar to the temporal neighborhood case, given a saliency map $\boldsymbol{S}$ (Eq. (34)) of size $M \times N$ and of depth K frames, we construct an uncertainty map $\boldsymbol{U}$ (Eq. (35)) of the same size and depth as $\boldsymbol{S}$ by iteratively processing saliency frames S_k using a 2-D averaging kernel $W^{L_{s_1} \times L_{s_2}}$ (Eq.(36)) of size $L_{s_1} \times L_{s_2}$.

$$\boldsymbol{S} = \begin{bmatrix} S_1 & S_2 & \dots & S_K \end{bmatrix}, \tag{34}$$

$$\boldsymbol{U} = \begin{bmatrix} U_1 & U_2 & \dots & U_K \end{bmatrix}, \tag{35}$$

$$U_k = \gamma \big| \alpha S_k * W^{L_{s_1} \times L_{s_2}} \big|, \tag{36}$$

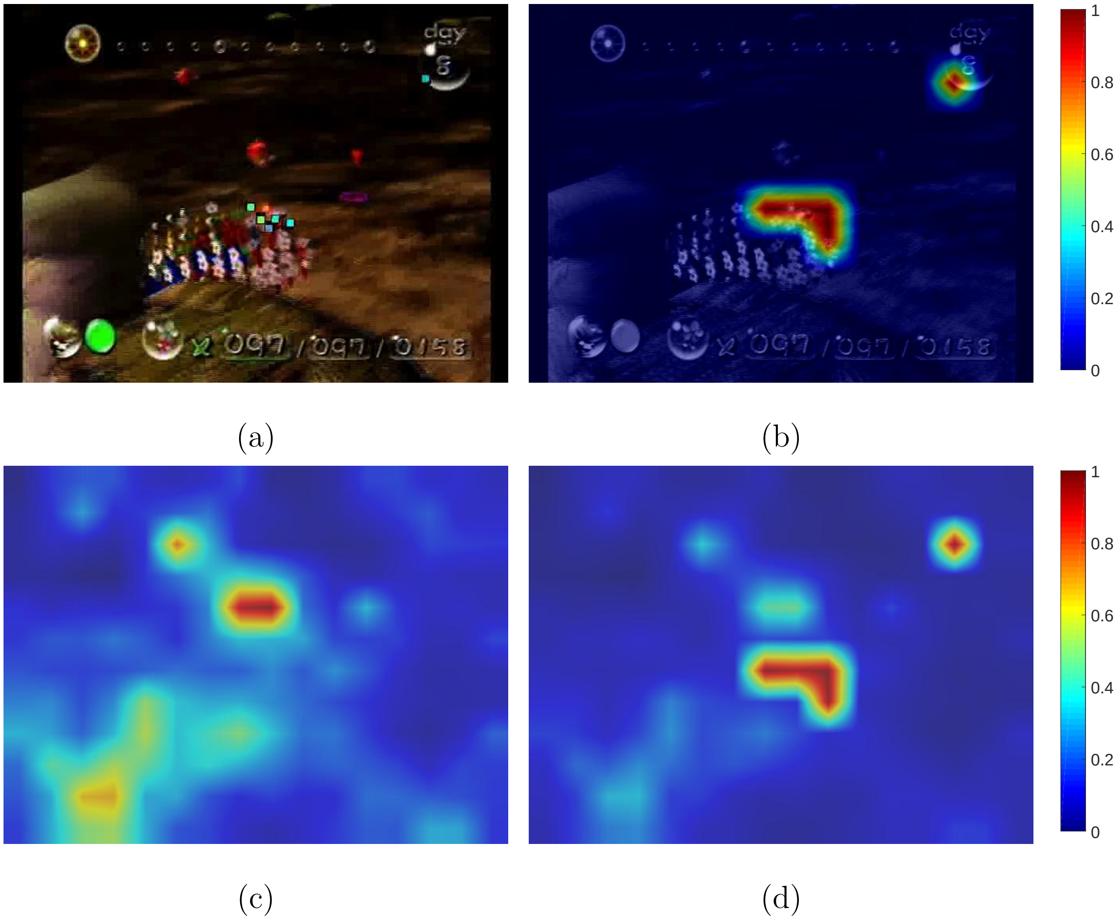
where $k = 1, 2, ..., K$ is the frame index, $W^{L_{s_1} \times L_{s_2}}$ is a spatial filter similar to averaging kernel W^{L_t}, symmetrical around its center and has a size of $L_{s_1} \times L_{s_2}$, α and γ are scaling factors.

5.3 Methods for Ground Truth Generation and Performance Evaluation

To objectively evaluate the performance of an uncertainty estimation algorithm, ideally we need to compare the estimated uncertainty against the ground truth, or the true uncertainty. However, such true uncertainty data is not readily available.

5.3.1 Computing True Uncertainty

Available databases for saliency detection research usually contain ground truth data recording eye fixations of human subjects viewing the images or videos. Based on the eye fixation data, as we proposed in [4], the following method is used to generate the true uncertainty data. Figure 48 illustrates this procedure with some examples while the block diagram is shown in Figure 49. First, we compile the fixation data from all subjects in CRCNS dataset into a single map $\hat{\boldsymbol{F}}^{tr}$ of size M', N', and K being the height, width, and the total number of frames, respectively. We add 1 to $\hat{F}^{tr}[i, j, k]$ for every eye fixation that corresponds to pixel location $[i, j, k]$. Second, we resize the fixation map $\hat{\boldsymbol{F}}^{tr}$ to M, N and K; the respective height, width, and depth of the saliency map $\boldsymbol{S}$ from a saliency detection algorithm. This resizing is necessary because many saliency detection techniques work on downsampled video frames for computational efficiency. However, for the binary map $\hat{\boldsymbol{F}}^{tr}$, the resizing is not exactly a downsampling procedure.Denoted as $\boldsymbol{F}^{tr}$, the resized binary fixation



(a) (b)

(c) (d)

Figure 48: Examples illustrating true uncertainty data. (a) Original video frame with eye fixation superimposed (small color squares in the center and top-right corner); (b) Resized eye fixation map superimposed on the original frame; (c) Saliency detection results; (d) True uncertainty. We note that the color display is only for a better illustration, which involves some interpolation causing the discrete resized fixation map to appear continuous.

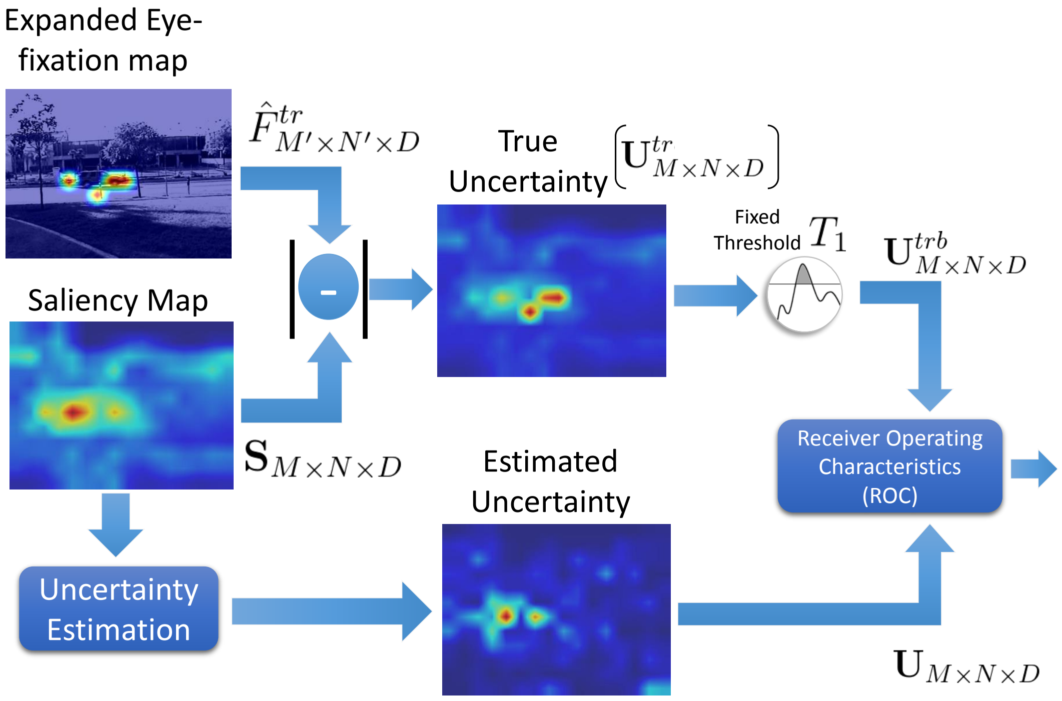

Expanded Eye-fixation map
$\hat{F}^{tr}_{M'\times N'\times D}$
True Uncertainty
$\mathbf{U}^{tr}_{M\times N\times D}$
Fixed Threshold T_1
$\mathbf{U}^{trb}_{M\times N\times D}$
Saliency Map
-
$\mathbf{S}_{M\times N\times D}$
Receiver Operating Characteristics (ROC)
Estimated Uncertainty
Uncertainty Estimation
$\mathbf{U}_{M\times N\times D}$

Figure 49: Evaluation methodology [4].

map is obtained as follows

$$F^{tr}[m,n,k] = \sum_{\forall(i,j)\in\Phi[m,n,k]} \hat{F}^{tr}[i,j,k], \tag{37}$$

where $\Phi[m,n,k]$ is an indexing function that points to the set of pixels in $\hat{F}^{tr}$ that corresponds to pixel $[m,n,k]$ in F^{tr} map. Here, we use the sum of eye-fixation points from all subjects so that salient locations agreed upon by majority of subjects have the highest saliency, but at the same time sparse "1"s in the original fixation truth data are not lost. Finally, assuming that the saliency map $\boldsymbol{S}$ is normalized, we normalize $\boldsymbol{F}^{tr}$ and calculate the true uncertainty as

$$\boldsymbol{U}^{tr} = \left|\boldsymbol{S} - \boldsymbol{F}^{tr}\right|. \tag{38}$$

Obviously, $\boldsymbol{U}^{tr}$ shows how far each saliency estimate is from the recorded fixations. Thus, it can serve as a measure of the estimation uncertainty. Even though the individual eye-fixation data is binary, the aggregated fixation maps $\hat{\boldsymbol{F}}^{tr}$, $\boldsymbol{F}^{tr}$, the derived true uncertainty data $\boldsymbol{U}^{tr}$, and the saliency detection results $\boldsymbol{S}$ are continuous values.

5.3.2 Performance Measurement

With the true uncertainty data available, we use a detection theory-based scheme for the performance evaluation [4]. The scheme generates an ROC curve and uses AUC as the performance metric [35]. Since our true uncertainty data $\boldsymbol{U}^{tr}$ is continuous, it needs to be converted to binary data, denoted as $\boldsymbol{U}^{trb}$, as the ROC curve is intended for binary classifiers. This conversion is conducted by applying a threshold T_1. To generate the ROC curve, the uncertainty estimates $\boldsymbol{U}$ are also thresholded by T_2 into a binary form, $\boldsymbol{U}^{b}$, and compared against $\boldsymbol{U}^{trb}$. Thus, both the true detection rate (TDR) and the false positive rate (FPR) are obtained. When we change the value of T_2, sweeping through its whole range, pairs of TDR and FPR are obtained to yield an ROC curve plotted as TDR vs. FPR. Then, the AUC is easily computed. AUC ranges between 0 and 1, with a greater value indicating better performance, and 0.5 indicating a performance equivalent to random classifier.

5.4 Experimental Results

We conducted three sets of experiments to study several aspects of the proposed algorithm. In the first set, we compare the relative performance based on the neighborhood selection. We evaluate and compare the performance of the proposed algorithm using:

- Spatiotemporal neighborhood as described in 5.2, labeled Spatiotemporal Uncertainty (STU)
- Temporal neighborhood as described in 5.2.1, labeled Temporal Uncertainty (TU) [4]
- Spatial neighborhood as described in 5.2.2, labeled Spatial Uncertainty (SU) [3]
- Naive fusion of Spatial and Temporal Uncertainty (SU+TU), a pixel-wise addition of TU and SU maps
- Entropy-based Uncertainty (EU) [23]
- Local variance of spatiotemporal neighborhood, labeled *Baseline*

The performance of these algorithms is quantified in terms of Area-Under-the-Curve (AUC) values of their corresponding Receiver-Operating-Characteristic (ROC) curves. We, also, show effects of saliency map scale as well as kernel size on the proposed algorithm's performance. Details on data and experiments procedure are provided in the dataset section and the performance evaluation methodology section, respectively. The second set of experiments are designed to show performance of the proposed uncertainty estimation algorithm given different categories of videos. Also, we show the distinct effects of kernel size on the proposed algorithm performance given radically different video contents. The third set of experiments verifies the performance of the proposed algorithms using additional datasets and saliency detection models.

5.4.1 Datasets

We tested the proposed unsupervised uncertainty estimation algorithm using three publicly available databases: CRCNS [46], DIEM [66], and AVD [65]. The CRCNS [46] database includes 50 videos, with the resolution being 480×640 and the duration ranging from 5 to 90 seconds with 30 frames per second. The videos contents are diverse with a total of 12 categories ranging from street scenes to video games and from TV sports to TV news. In many cases the videos contain variations of lighting conditions, severe camera movements, and high motion blur effects. Eye fixation data are provided with each video, recorded for a group of eight human subjects watching the videos under task-free view condition. The DIEM [66] database includes 85 videos, with varying resolutions and duration up to 130 seconds with 30 frames per second. The videos content are mainly limited to TV and film content including film trailers, music videos, and advertisement. The eye fixation data are collected from 250 participants under task-free view conditions. The AVD [65] database includes 148 videos, with varying resolutions and mean duration of 22 seconds with 30 frames per second. The video contents are limited to moving objects, landscape, and faces. The eye fixations data are collected from 176 observers. The AVD dataset contains two sets of videos of the same visual content but one with audio and the other without. According to their findings on the effect of audio on the attention of the participants, we only select the videos without associated audio.

For our experiments, we generated saliency maps for the videos using a recent algorithm based on 3D FFT local spectra (3DFFT) [61]. However, for validation, we also share the results from two additional saliency models: STSR [77] and PQFT [31], which are shown at the end of this section. Unless stated otherwise, saliency maps used in all experiments are generated using 3DFFT. In most of our experiments, the saliency maps are reduced in size to three different scales. *Scale 1* is of size 12×16; a downscale of frames original size 480×640, where every 40×40 region in the original frame corresponds to a single pixel in *Scale 1*. Similarly, *Scale 2* saliency maps are 24×32, where every pixel is equivalent to 20×20 region of pixels in the original sized frame, and *Scale 3* saliency maps are 48×64, where every pixel is equivalent to 10×10 regions.

5.4.2 Threshold Selection

The performance evaluation procedure described earlier utilizes a fixed threshold T_1 to transform the continuous valued true uncertainty $\boldsymbol{U}^{tr}$ to binary ground truth. First, we examine the impact of changing the value of T_1. The algorithms under consideration are: Temporal Uncertainty (TU), Spatial Uncertainty (SU), Fused Spatial and Temporal Uncertainty (SU+TU), Spatiotemporal Uncertainty (STU), Spatiotemporal local variance (*Baseline*) computed on the same neighborhood as STU, and Entropy-based Uncertainty (EU). Figure 50 shows the performance of these algorithms in terms of AUC versus T_1. As shown in Figure 50, T_1 directly affects AUC value; as the value of T_1 increases, the AUC value of all algorithms considered here decreases. It is also interesting to point out that the gradient of AUC levels-off as T_1 reaches higher values. Although we can see that T_1 value significantly changes AUC, conclusions based on relative AUC values are consistent regardless of the value of T_1. As shown in Figure 50, STU outperforms all other algorithms while EU is performing the worst in this experiment. Please note that the reported AUC results are for *Scale 1* maps with averaging kernel of length 5 for TU, of size 5×5 for SU, and $5 \times 5 \times 5$ for STU.

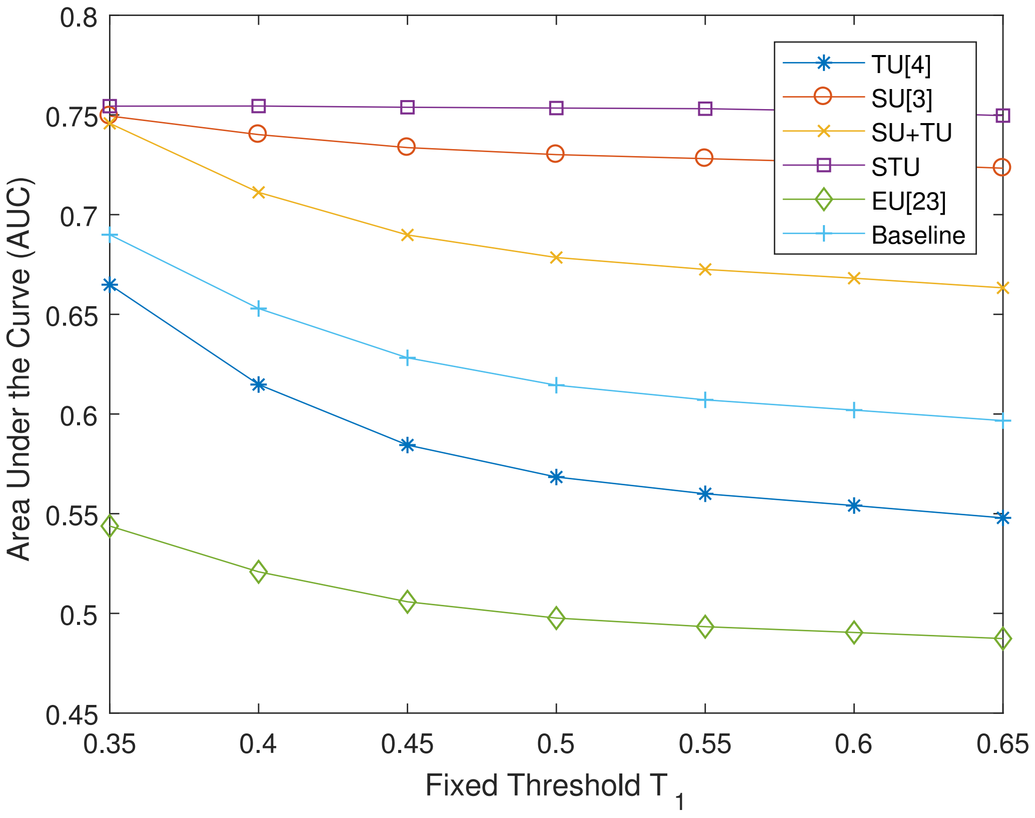


Figure 50: Examples illustrating that relative uncertainty estimation performance is independent of fixed threshold T_1 applied to true uncertainty. Results reported here were generated using *Scale 1* maps with averaging kernel of length 5 for TU, of size 5×5 for SU, and $5 \times 5 \times 5$ for STU.

5.4.3 Neighborhood Selection

As shown earlier, in addition to the threshold T_1, the neighborhood selection affects AUC value. Additionally, scale of the saliency maps and size of the processing kernels affect the performance of proposed estimation algorithm as well. In Figure 53, we show the AUC values for the algorithms under test using different saliency map scales. The experiment is conducted using saliency maps of *scale 1*, *2* and *3* and an averaging kernel. In order to fix the kernel size relative to the support region size in the original frame, we use different kernel size for each scale, as illustrated in Figure 51. In Figure 53, *Scale 1* experiment uses 5×5 for SU and $5 \times 5 \times 5$ for STU. Similarly, for *Scale 2*: 11×11 for SU and $11 \times 11 \times 5$ for STU, and for *Scale 3*: 21×21 for SU and $21 \times 21 \times 5$ for STU. The length of TU kernel is fixed $L_t = 5$. We can see that the change in AUC value is relatively small, thus, shows the effectiveness of the proposed uncertainty algorithm even when saliency maps are considerably small size. This feature of the proposed estimation algorithm can be exploited to reduce the required computations, thus speeding up the estimation process without much sacrifice in terms of performance. Please note that AUC value for EU algorithm changes over different scales, due to true uncertainty $\boldsymbol{U}^{tr}$ containing more details as the scale increases.

Moreover, kernel size affects the performance of the proposed algorithm as well. Figure 52 shows the performance of the estimation algorithms under test, in terms of AUC values, when the estimation kernel size is changed. The experiment is conducted using *scale 2* saliency map and variable kernel size r (r for TU, $r \times r$ for SU, and $r \times r \times r$ for STU). As shown in Figure 52, AUC of the proposed algorithm changes as the size of the kernel changes. However, the change in TU performance is significantly smaller than that of SU and STU because the number of pixels added into SU and STU kernels is significantly more than the number of pixels added to TU kernel. There is, however, a slight degradation in TU performance as the kernel size increases

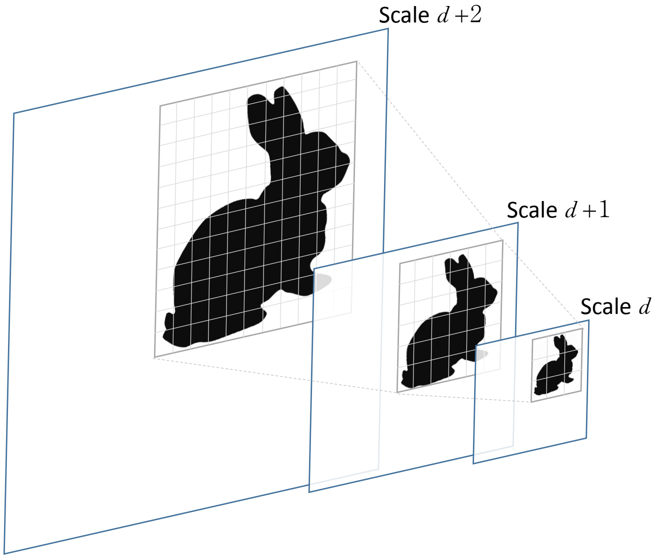


Figure 51: Kernel size changes between scales according to support region size.

(starting from $L_t = 13$ onwards), which can be attributed to including less relevant pixel in the estimation process as the kernel size increase. For kernels of sizes $3 \times 3 \times 3$ till $11 \times 11 \times 11$, it can be seen that STU achieves higher AUC than SU. However, such trend inverts starting from kernel size $13 \times 13 \times 13$ onwards. This could be explained by noting the similar trend in TU as the kernel size increases in time domain due to inclusion of pixels that might be less relevant. The performance degradation in STU (and $Baseline$ as well) is more profound than TU because, for a kernel size of $n \times n \times n$, n^2 pixels are added to STU estimation process for every additional frame while only a single pixel is added for TU estimation. It is important here to clarify that these results are obtained for the whole dataset (50 videos). Thus, trends that are observed here are not necessarily true for every video type. We discuss in details the performance as related to the video categories in the next section.

5.4.4 Video Categories

Given the diverse nature of scenes and dynamics in the dataset, we evaluate the performance of our proposed algorithm for each category in the dataset. For these

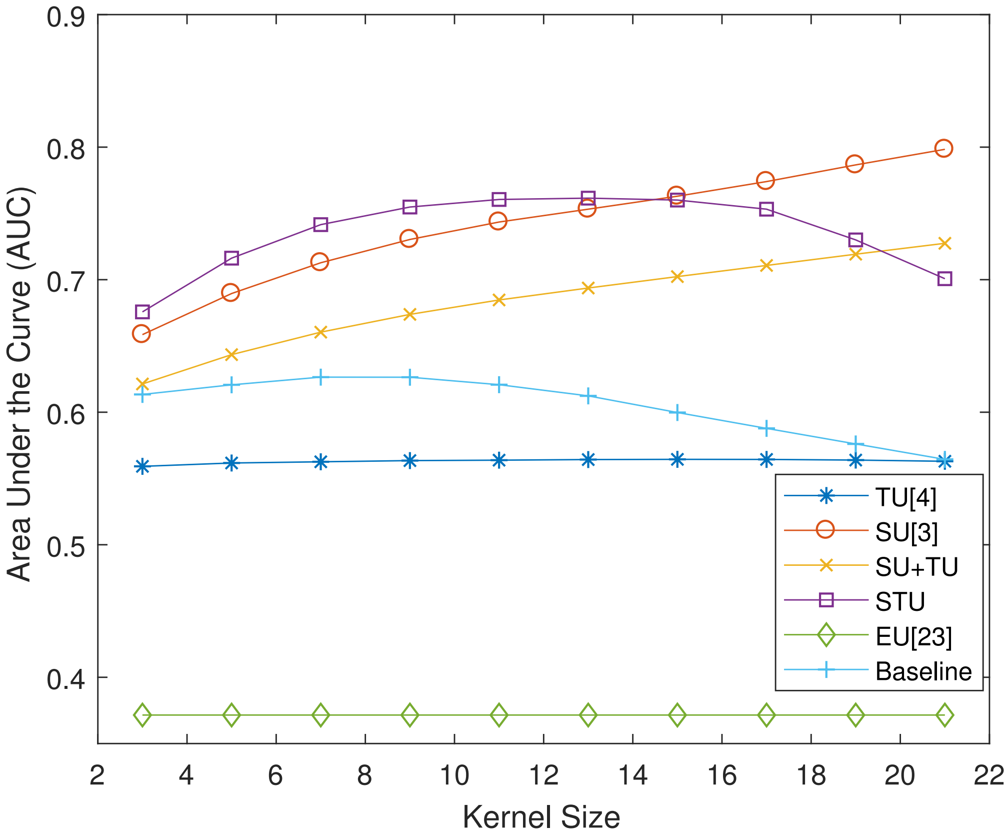


Figure 52: AUC value is affected by the choice of the kernel size at the same scale. Results reported here use *Scale 2* saliency maps and $T_1 = 0.55$.

experiments, we set $T_1 = 0.55$ and use *Scale 1* saliency maps. Table 2 shows AUC values for *TU* ($L_t = 5$), *SU* ($L_{s_x} = 5$), *ST+SU*, *STU* ($L_{st_x} = 5$), *EU*, and *Baseline* ($L_{st_x} = 5$), for each category, separately. As shown in Table 2, AUC values for the proposed algorithm are above 0.5, indicating that the proposed algorithm is advantageous over random guessing. Additionally, the algorithm performs better than *EU* in every category and in some by a wide margin. One interesting result is that AUC for Saccadetest video is significantly higher than other categories for all algorithms considered here. This can be attributed to its non-complex structure, which shows a disk moving against a light textured background. Notably, *STU* achieves highest performance in every category except Saccadetest. This could be attributed to its relative constant scenes in the first segment of the video.

Moreover, we explore the effect of kernel size on the estimation performance. In these experiments, we focus on *STU*, however, *TU*, *SU*, and *SU+TU* exhibit similar behavior. Figure 54 shows AUC for *STU* estimation algorithm on three video categories; saccadetest, tv-talk, and gamecube for kernel sizes: $L_{st_x} = 3$, 7, 11, and 15, using *Scale 2* saliency maps and $T_1 = 0.55$. As shown in Figure 54, as the kernel size increases, *STU* performance on saccadetest degrades indicating that the relevance saliency context in saccadetest video is strictly local and including more pixels than direct neighbors degrades uncertainty estimation performance. Indeed, the structure of saccadetest video justifies these results due to its simplicity. In contrast, gamecube video uncertainty estimation results increase as the kernel size increase. This indicates that the set of correlated saliency pixels for gamecube is larger than its direct neighbors. The large set of correlated saliency pixels in gamecube might be explained by its complex structure and the fact that these videos contain multiple salient actors in the same scene making it more difficult to capture saliency context from small local neighborhoods. On the other hand, *STU* performance in estimating uncertainty for tv-talk reaches maximum level in intermediate kernel sizes and then decreases as we increase the kernel size, indicating that the most appropriate kernel size to capture relevant saliency context is half the frame size.

5.4.5 Comparison across various datasets and saliency models

In this section, we present evaluation results for the proposed algorithm across various datasets. We compare the performance of the proposed algorithm using videos from three datasets: CRCNS [46], DIEM [66], and AVD [65]. Figure 55 shows the the AUC values of the five uncertainty estimation methods using videos from the three datasets. In Figure 55, STU performance is the highest among all datasets. In general, the trend and ranking between the uncertainty estimation methods is consistent across the three datasets.

Additionally, we present the evaluation results for the proposed algorithm across using three saliency models: 3DFFT [61], STSR [77], and PQFT [31]. Figure 56 shows the AUC values of the five uncertainty estimation algorithms. In Figure 56, a consistent trend and ranking between the five algorithms exist across all three saliency models, where STU achieves the highest AUC value.

Moreover, we evaluate the proposed algorithm, in terms of the computed uncertainty map distribution versus uncertainty ground truth maps distribution, using four distribution-based metrics; Jeffrey Divergence (JD), Jensen-Shannon divergence (JS), Histogram Intersection (HI), L2-norm. As shown in Table.3, the proposed algorithm provides the closest distribution to that of the ground truth maps across all four metrics and all datasets.

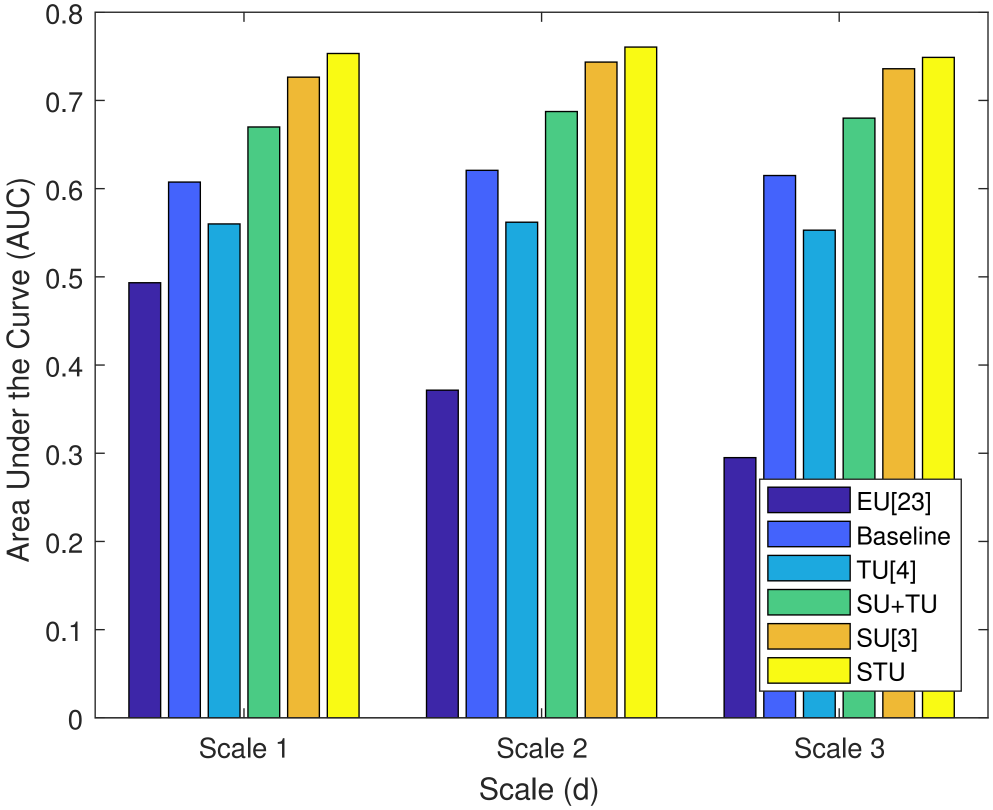


Figure 53: The impact of scale change with constant support region (using different kernel sizes). AUC value is relatively the same when processing different scales. Results reported here use threshold $T_1 = 0.55$.

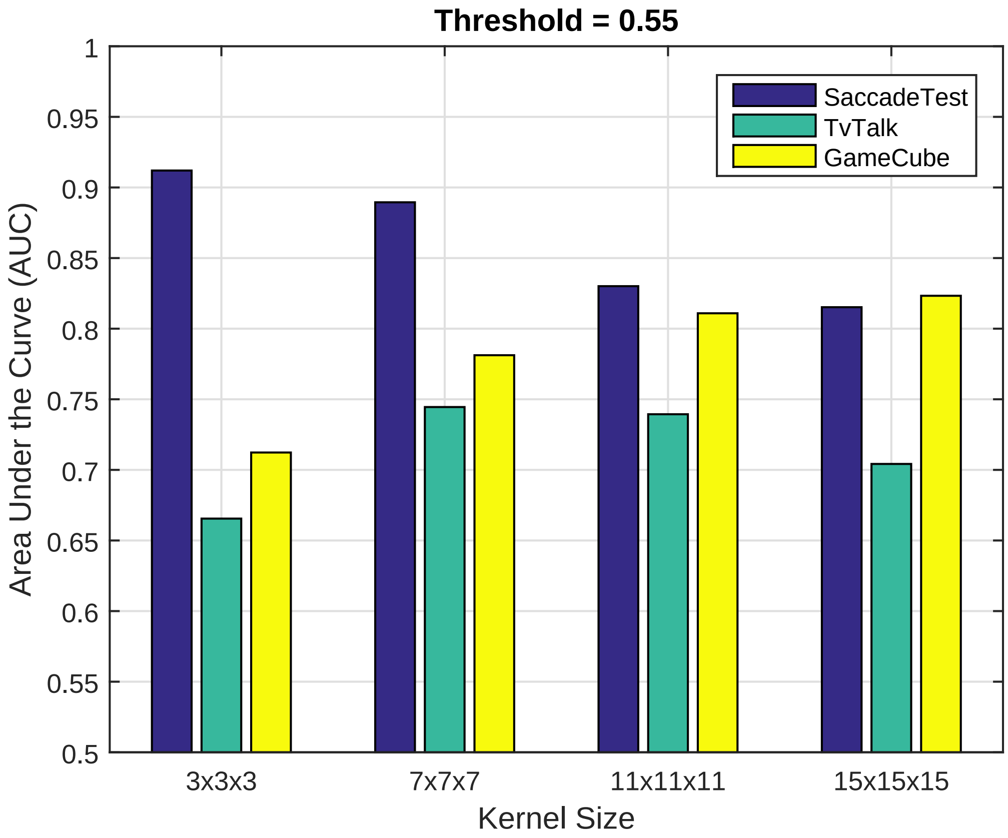


Figure 54: Examples illustrating the effect of kernel size on the estimation performance using *STU* extracted from uncertainty maps of *Scale 2*.

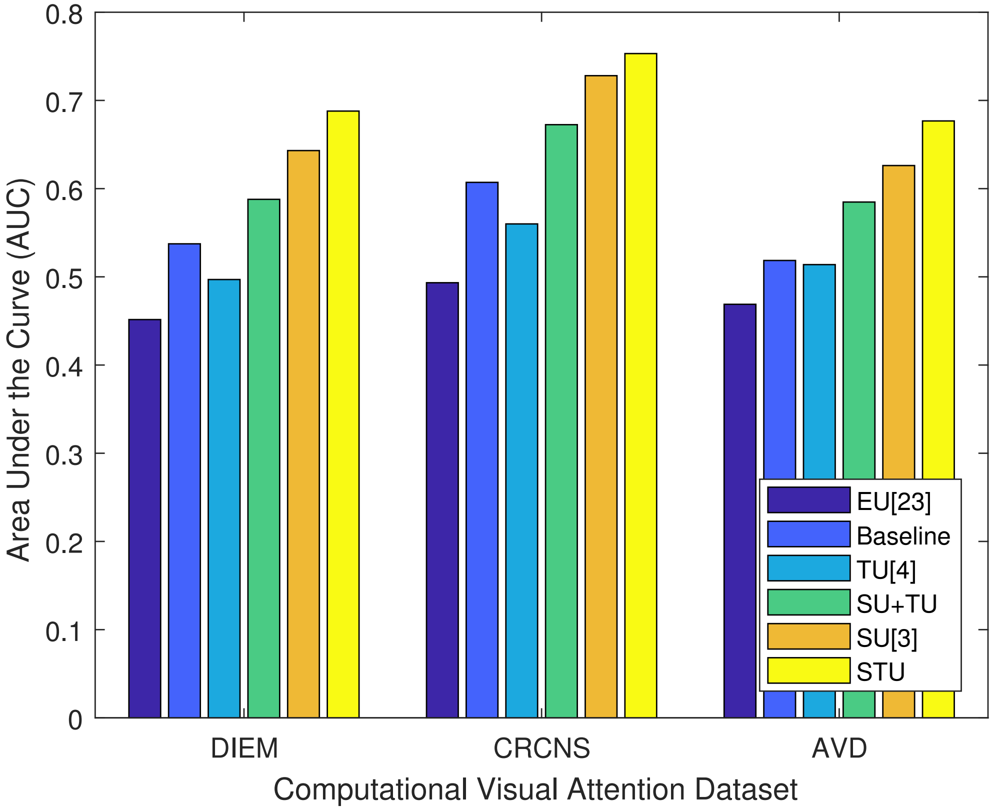


Figure 55: The performance of the proposed algorithm across the datasets CRCNS [46], DIEM [66], and AVD [65].

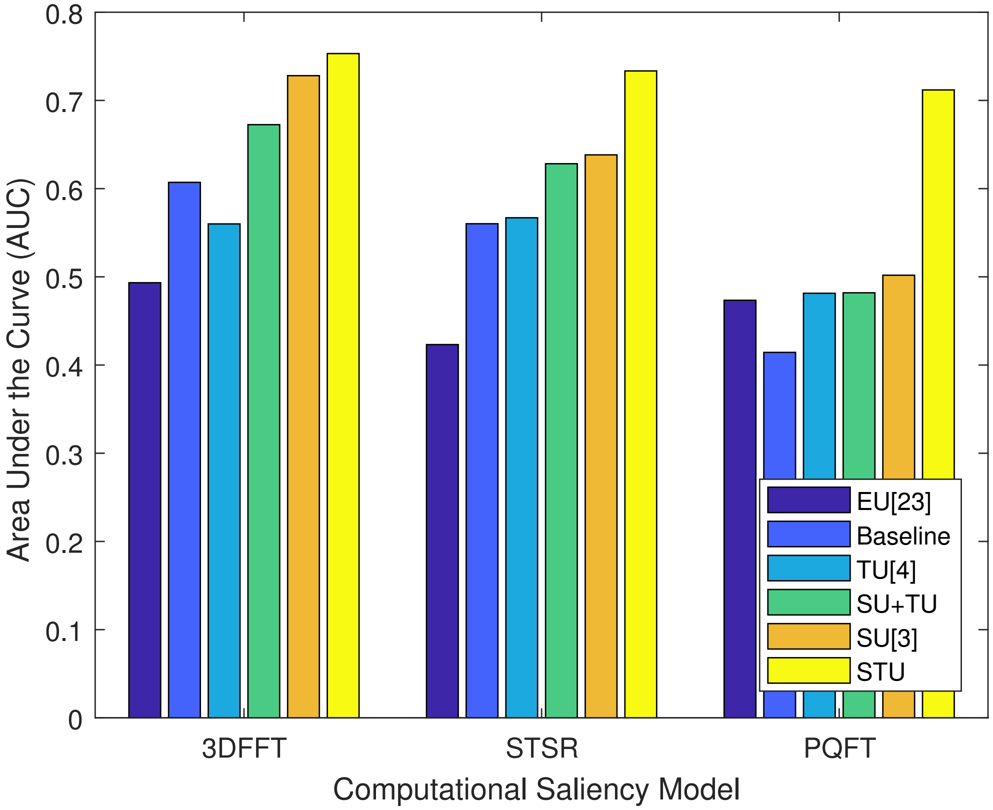


Figure 56: The performance of the proposed algorithm using the saliency models 3DFFT [61], STSR [77], PQFT [31]

Table 2: List of AUC value for different categories using fixed threshold $T_1 = 0.55$ and *Scale 1* saliency maps. Note that the highest AUC value in each category is labeled in green and lowest AUC value in red. Also, the category with the highest AUC in the dataset is shown in bold

	TU [4]	SU [3]	SU+TU	STU	EU [23]	*Baseline*
beverly	0.5793	0.8088	0.7174	0.8130	0.5835	0.6915
gamecube	0.5987	0.7636	0.7155	0.7913	0.5906	0.6834
monica	0.6152	0.7801	0.7240	0.7994	0.5728	0.6506
saccadetest	**0.7722**	**0.8734**	**0.8216**	0.8587	**0.8458**	**0.8308**
standard	0.5866	0.7190	0.6609	0.7462	0.5165	0.5841
tv-action	0.7481	0.8466	0.7970	**0.8667**	0.7245	0.6491
tv-ads	0.5565	0.7248	0.6565	0.7476	0.5228	0.5360
tv-announce	0.4555	0.6679	0.5550	0.7321	0.4434	0.5818
tv-music	0.5548	0.6721	0.6236	0.7427	0.4471	0.5771
tv-news	0.5051	0.6497	0.5885	0.6947	0.4861	0.5029
tv-sports	0.5156	0.6746	0.6170	0.7172	0.5020	0.5368
tv-talk	0.5692	0.7142	0.6393	0.7364	0.5299	0.5250

Table 3: Estimated distances using distribution-based metrics for the proposed feature in comparison with the state-of-the-art algorithms using *Scale 1* saliency maps computed using 3DFFT algorithm [61]. Note that the highest value in each distance metric is labeled in green and the lowest value in red. Also, the distance values for the proposed algorithm is shown in bold

	CRCNS				DIEM				AVD			
Algorithms	JS	JD	HI	L2	JS	JD	HI	L2	JS	JD	HI	L2
TU [4]	0.34	0.68	0.64	0.25	0.16	0.32	0.30	0.12	0.33	0.66	0.60	0.24
SU [3]	0.14	0.29	0.32	0.12	0.05	0.09	0.13	0.04	0.09	0.18	0.24	0.08
SU+TU	0.14	0.27	0.32	0.12	0.05	0.11	0.14	0.05	0.10	0.20	0.28	0.10
STU	**0.08**	**0.15**	**0.24**	**0.08**	**0.03**	**0.05**	**0.10**	**0.04**	**0.06**	**0.12**	**0.21**	**0.08**
EU [23]	0.51	1.02	0.82	0.46	0.23	0.45	0.36	0.19	0.45	0.91	0.71	0.41
Baseline	0.38	0.76	0.68	0.32	0.15	0.30	0.29	0.13	0.29	0.58	0.57	0.25

CHAPTER VI

UNCERTAINTY ESTIMATION USING SPATIOTEMPORAL ANALYSIS

6.1 Enhanced Uncertainty-based Framework for Video Saliency Applications

As mentioned before, video saliency application tend to utilize the computed saliency maps without an evaluation of its accuracy. To enable a systematic analysis of the computed saliency maps, we proposed an uncertainty-based framework, section 5.1, that aims to improve the current pipeline by estimating the uncertainty of the computed saliency maps and utilizing the estimated uncertainty to inform the decision making process. We have also shown, in Chapter 5, that an effective uncertainty estimation can be achieved by relying on the consistency of the saliency map which reveals some aspects regarding the accuracy of the computed saliency map. However, a significant amount of information about the video saliency is embedded in the video frames themselves which might be useful for uncertainty estimation. Additionally, we have shown, in Chapter 3, that motion information, both the scene motion and camera motion, contributes significantly to the visual attention. Therefore, we expand our uncertainty-based framework, proposed in section 5.1, to include motion information extracted from the video frames as shown in Figure 57.

Similar to the original uncertainty-based framework in Figure 47, the extended framework, Figure 57, enables a separation between the saliency detection and the decision making model by incorporating uncertainty information. However, the uncertainty estimation itself takes as its input both the saliency map coming from the saliency detection algorithm and motion information coming from an *abstraction*

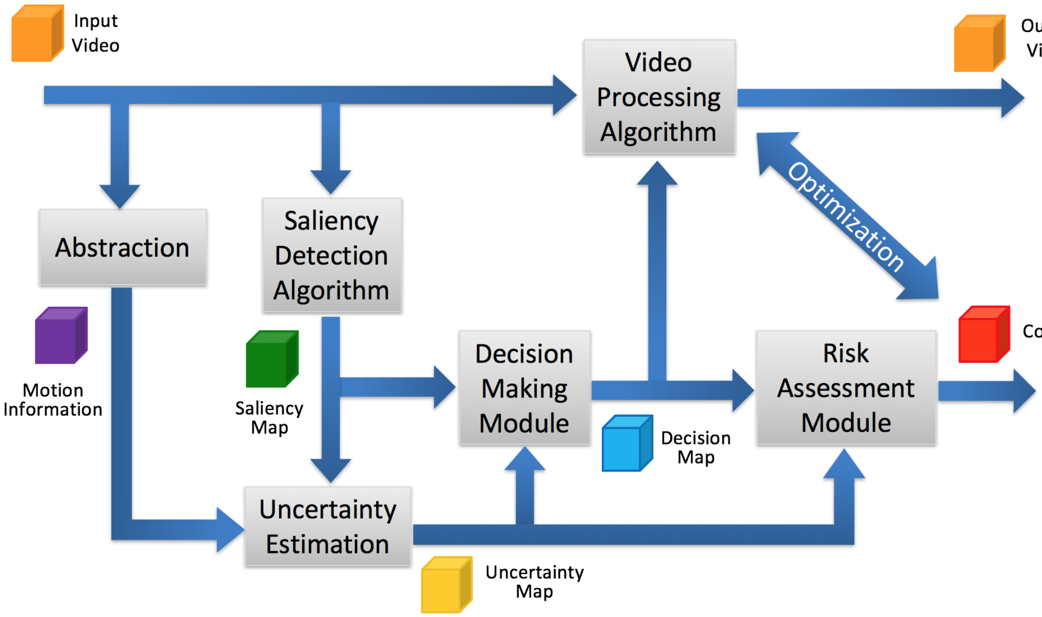


Figure 57: Enhanced Uncertainty-based framework for improving saliency-enabled video processing algorithms

layer. This *abstraction* step is used here to be a general representation of feature extraction algorithms rather than a specific algorithm. Unlike the case where video frames are fed directly to the uncertainty estimation algorithm, separating uncertainty estimation from extracting uncertainty-relevant information from video frames enables tackling each problem separately.

6.2 Multi-factor Uncertainty Estimation

As discussed in Chapter 3, pixels in eye-fixation maps are correlated and visual saliency is mostly consistent where changes in saliency values happen gradually. Thus, sudden changes in saliency value, compared to its local neighborhoods, should lower our trust in that particular spatiotemporal event. Based on these assumption, we proposed in Section 4.2 an uncertainty estimation feature based on spatiotemporal cues that takes these assumption into account.

Formally, given a saliency map $\boldsymbol{S}$ of size $M \times N$ and of depth K frames, the spatiotemporal uncertainty feature map $\boldsymbol{F_{STU}}$ of the same size and depth as $\boldsymbol{S}$ approximates the saliency value divergence from its spatiotemporal local neighborhood mean. The estimation is efficiently computed by processing the map $\boldsymbol{S}$ according to Eq.(39):

$$\boldsymbol{F_{STU}} = \gamma \Big| \alpha \boldsymbol{S} * W^{L_1 \times L_2 \times L_3} \Big|, \tag{39}$$

where $\big|.\big|$ is the operation to find the absolute value and $d = 1, 2, ...D$ is the scale label, $L_1 \times L_2 \times L_3$ is the size of the spatiotemporal kernel $W^{L_1 \times L_2 \times L_3}$, α is a scaling factor for the saliency map to fix its range to be [0,1], and γ is a scaling factor for the uncertainty map to ensure the output range is [0,1]. We use a simple averaging kernel defined as follows

$$W^{L_1 \times L_2 \times L_3} = \begin{cases} \frac{R-1}{R} & \text{at the center} \\ -\frac{1}{R}, & otherwise, \end{cases} \tag{40}$$

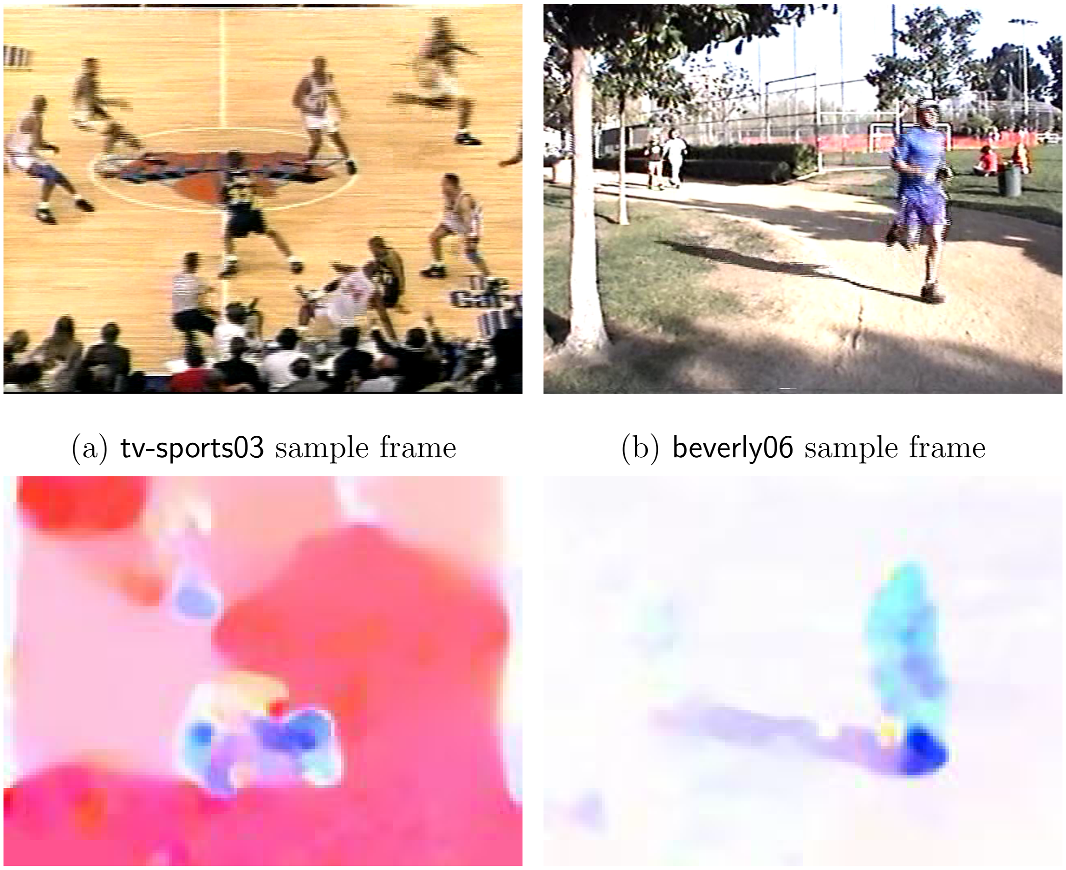
where $R = L_1 \times L_2 \times L_3$. The design of $W^{L_1 \times L_2 \times L_3}$ can be viewed as the difference between saliency value and a moving average window of size $L_1 \times L_2 \times L_3$. Appropriately sized, $W^{L_1 \times L_2 \times L_3}$ can follow the changes in the scene and, to some extent, approximates the common trend of pixel saliency change over time.

Additionally, we showed, in Chapter 4, that motion plays a major role in explaining video scenes and how the complexity of motion and subjects' prior information can affect the predictability of visual attention in these videos. For example, tv-sports03 (scene from TV coverage of basketball game) shown in Figure 58(a) and beverly06 (scene of a park with people walking and running) shown in Figure 58(b) both contain motion. However, according to prediction results shown in Figure 59, beverly06 is much easier to predict using a simple regression model compared to tv-sports03, as shown in Section 4.3. We believe this is due to complexity of motion in tv-sports03 where different players constantly change their positions across the basketball court to optimize their game play. In tv-sports03, the attention of subjects is mainly concentrated on the small basket ball rather than the players, which is not properly highlighted in tv-sports03 optical flow map. On the other hand, beverly06 shows a scene with static background and only few objects moving. Therefore, the optical flow map of beverly06, shown in Figure 58(c), corresponds to the visual saliency in the scene, which explains the accuracy of the optical flow based prediction. Therefore, we use optical flow of the video frames as one of the factors to estimate uncertainty. Formally, given a video $\boldsymbol{V}$ of size $M \times N \times K$, first, we downsample the video frames by a factor β, then compute optical flow feature map $\boldsymbol{F_{OF}}$ using [58] according to Eq. 41:

$$\boldsymbol{F_{OF}} = \Big|\Big|\boldsymbol{G} * (\boldsymbol{V}^{\beta})^{OF}\Big|\Big|, \tag{41}$$

where $\big|\big|.\big|\big|$ is the operation of computing the magnitude, $\boldsymbol{G}$ is a Gaussian kernel to smooth out the optical flow map discontinuities, which are due to compression and interlacing artifices in the video dataset [47], $*$ is the convolution operator, $(\boldsymbol{V}^{\beta})^{OF}$ is the optical flow of the β-downsampled video frames.

Furthermore, we have shown, in Chapter 4, that camera motion plays a considerable role in controlling viewing patterns. Let us consider gamecube02 (a scene of video game play), shown in Figure 60(a), and standard05 (a scene of camera sweep recording a social event), shown in Figure 60(b), to examine camera motion effect on the predictability of subjects viewing patterns. In gamecube02 as the player moves the character throughout the video game, the in-game camera system keeps track of the changes and constantly centers the character in middle of the screen leading to out-of-plane motion On the other hand, camera motion parallel to and perpendicular to salient objects can be easily modeled as in the case of standard05 as the mosaic shown in Figure 60(b). In standard05, subjects attention is motivated by continuous flow of new information about the scene details, which is caused by camera motion. Hence,



(a) tv-sports03 sample frame

(b) beverly06 sample frame

(c) tv-sports03 optical flow

(d) beverly06 optical flow

Figure 58: Motion complexity affects the correlation between the frames of eye-fixation map and may lowers the confidence in the computed saliency maps

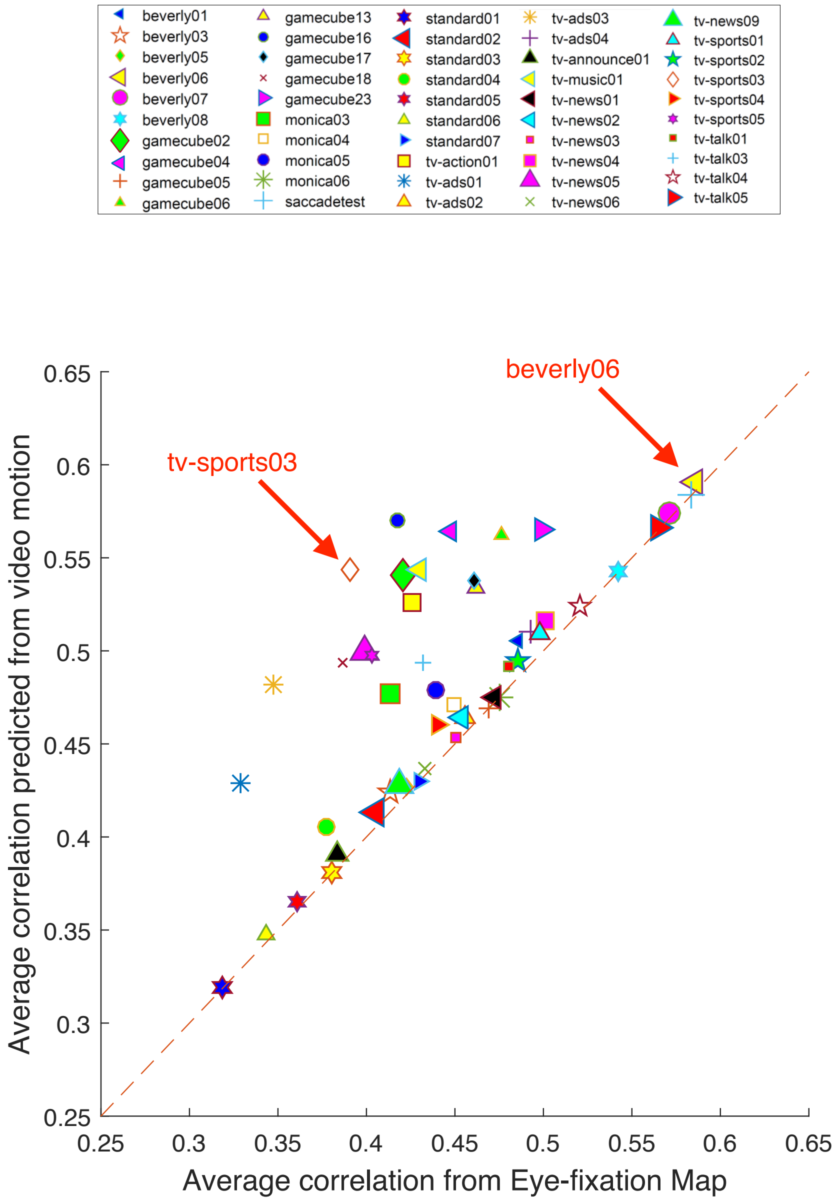


Figure 59: Correlation analysis between eye-fixation map frames shows that motion complexity plays a significant role in visual attention.

standard05 prediction results, shown in Figure 61, based on motion cues closely resemble the correlation in the eye-fixation data unlike the case of gamecube02. Therefore, we use camera motion as one of the factors to estimate uncertainty. However, instead of a detailed frame-by-frame camera motion estimation, we propose using a general label for the whole video as one of five choices (*Top*, *Bottom*, *Left*, *Right*, *Center*). To represent these labels numerically, we use linearly changing maps, as shown in Figure 62, to indicate our prior knowledge regarding the camera motion direction. Formally, given a video $\boldsymbol{V}$ of size $M \times N \times K$, we compute camera motion feature map $\boldsymbol{F_{CM}}$, of the same size and depth as $\boldsymbol{V}$, according to Eq. 42:

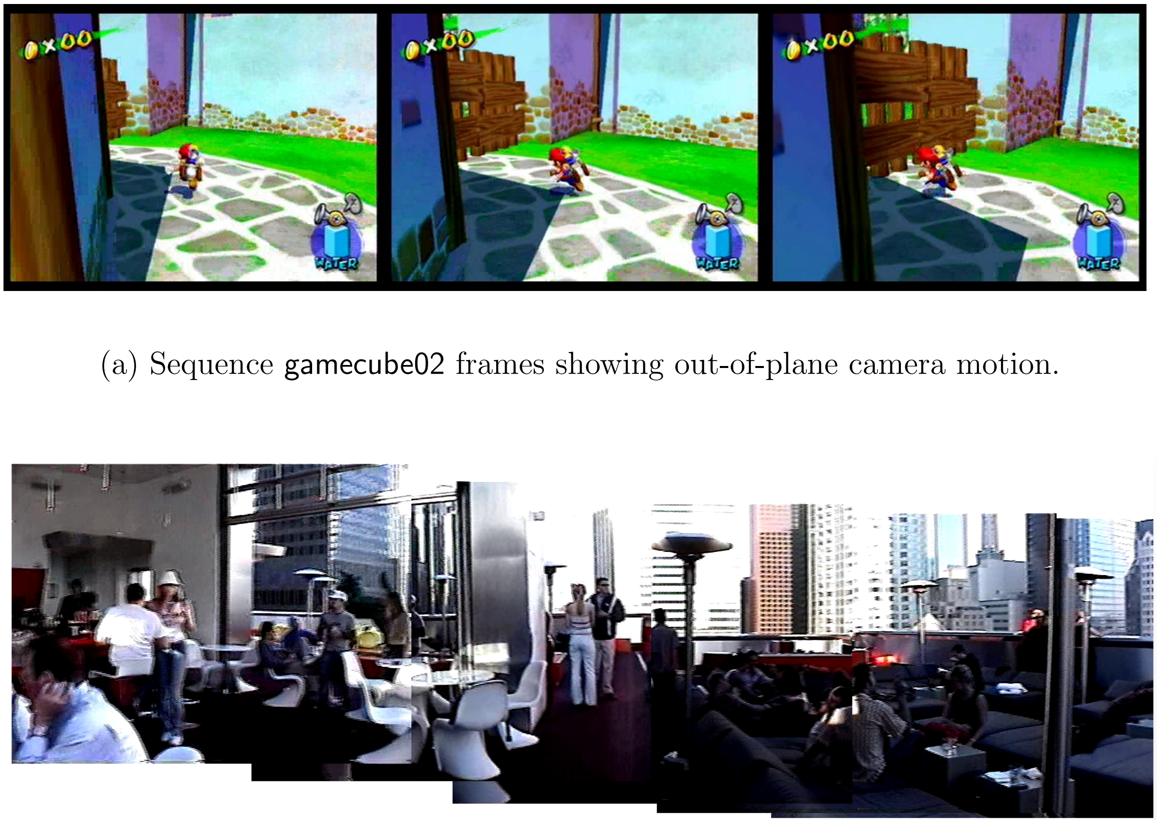
$$\boldsymbol{F_{CM}}(m, n, k) = C_1 + C_2 C_m m + C_3 C_n n + C_4 \sqrt{(m - m_{center})^2 + (n - n_{center})^2}, \quad (42)$$

where C_1, C_2, C_3, and C_4 are values used to differentiate between the various camera motion directions, C_m and C_n are normalization factors to ensure that the feature map $\boldsymbol{F_{CM}} \in [0, 1]$, $m = 1, 2, ..., M$, $n = 1, 2, ..., N$, and $k = 1, 2, ..., K$, and m_{center} and n_{center} are the spatial coordinates of the center pixel in $\boldsymbol{F_{CM}}$.

Based on these features, we propose a multi-factor uncertainty estimation algorithm using Gradient Boosting Trees (GBT) [27] summarized in Figure 63. After computing the three feature maps using Eq. 39, Eq. 41, and Eq. 42, we construct a feature vector $x_i \in R^d$, where $i = 1, 2, ..., M \times N \times K$ and x is a d-length real-valued vector, by unfolding an $L_1 \times L_2$ neighborhood around pixel i in each of the feature maps $\boldsymbol{F_{STU}}$, $\boldsymbol{F_{OF}}$, and $\boldsymbol{F_{CM}}$ that corresponds to the label y_i. Using the training and optimization procedures specified in [27], we train an ensemble of classification and regression trees (CARTs) [10] to find the parameters that minimize the objective function as follows:

$$\min \sum_{i=1}^{MNK} (y_i - \hat{y}_i)^2 + \sum_{i=1}^{T} \boldsymbol{\Omega}(f_i) \quad (43)$$

where M, N ,and K are the dimensions of the video $\boldsymbol{V}$, y_i is the true label for the



(a) Sequence gamecube02 frames showing out-of-plane camera motion.

(b) Mosaic of standard05 frames showing motion parallel to image plane

Figure 60: Camera/background motion with fixed main actor vs. camera/background motion parallel to image plane

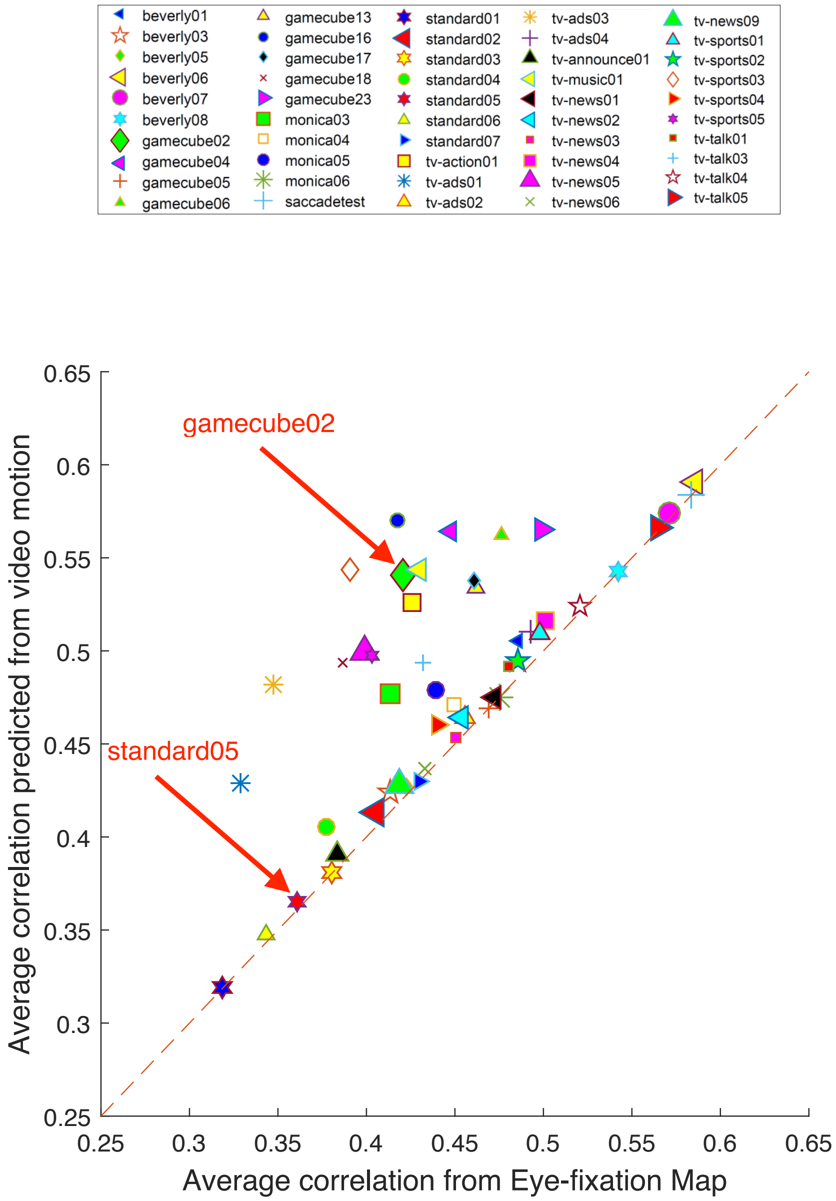
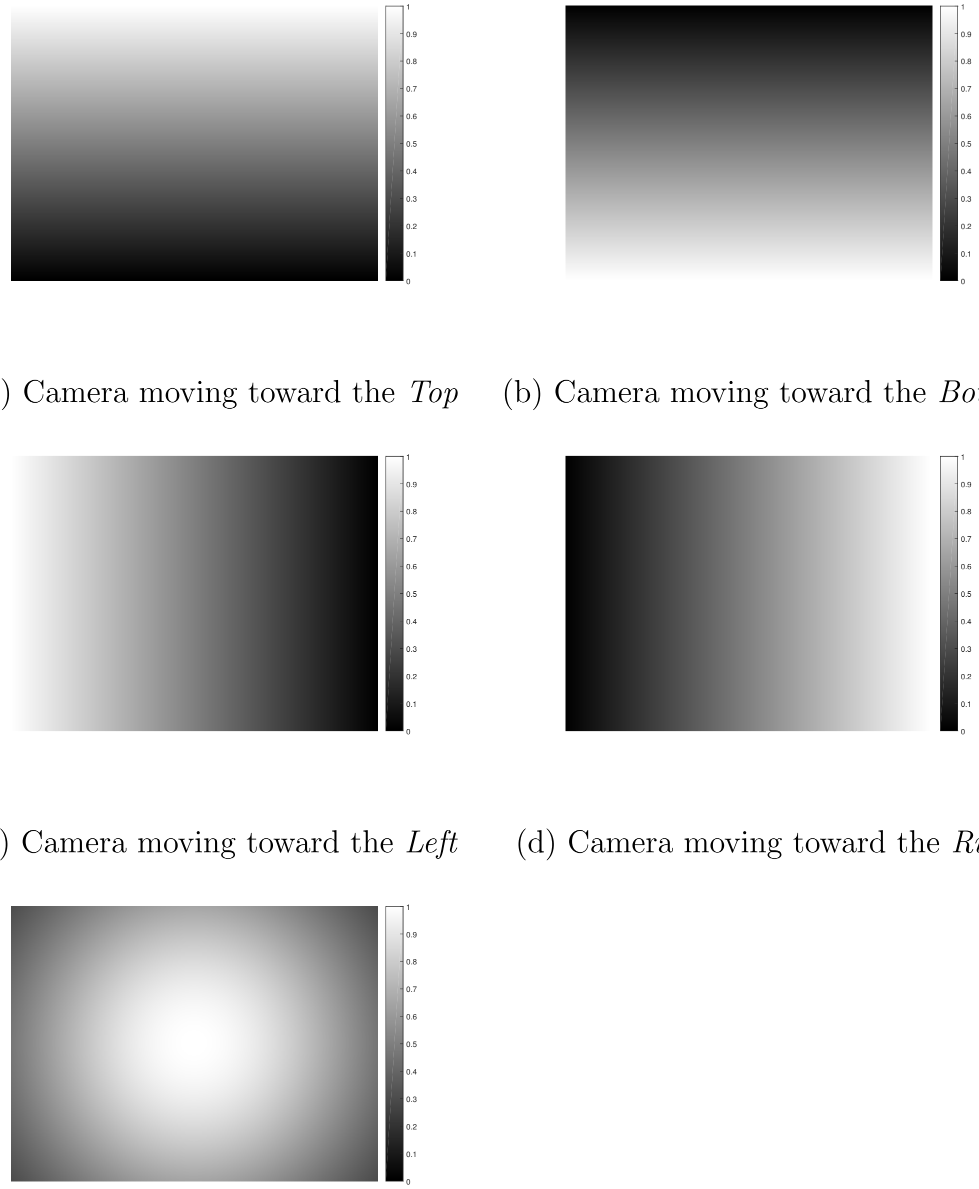


Figure 61: Correlation analysis between eye-fixation map frames shows that camera motion plays a significant role in visual attention.



(a) Camera moving toward the *Top*

(b) Camera moving toward the *Bottom*

(c) Camera moving toward the *Left*

(d) Camera moving toward the *Right*

(e) Camera moving around the *Center*

Figure 62: Camera motion feature maps

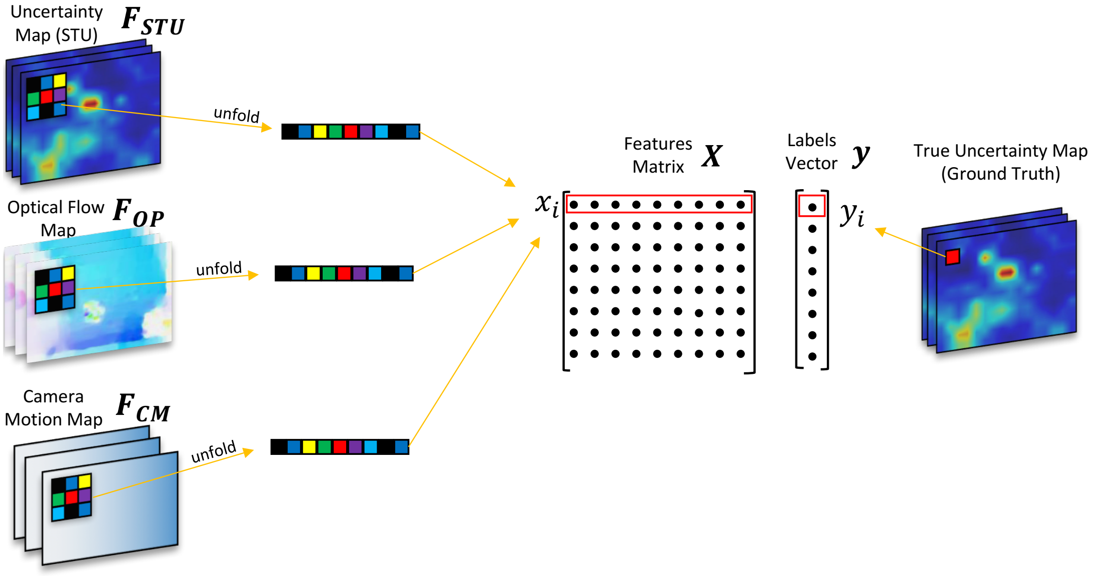


Figure 63: Illustration of the proposed Multi-factor Uncertainty Estimation.

i-th instance , $\mathbf{\Omega}$ is the regularization term to minimize the complexity of the trained CARTs (f_i's), T is the total number of CARTs in the trained ensemble and $\hat{y}_i$ is the estimated label for training which is computed according to:

$$\hat{y}_i = \sum_{k=1}^{T} f_k(x_i), \quad f_k \in \mathfrak{F} \tag{44}$$

where x_i is the feature vector, f_k's are the trained CARTs, T is the total number of CARTs in the trained ensemble, and $\mathfrak{F}$ is the set of all possible CARTs.

6.3 Experimental Results

We conducted three sets of experiments to study several aspects of the proposed algorithm. In the first set, we compare the relative performance based on the fixed threshold T_1, according to the evaluation framework proposed in 5.3. We evaluate and compare the performance of the proposed algorithm using:

- Spatiotemporal Uncertainty (STU) as described in 5.2
- Temporal Uncertainty (TU) as described in [4]

- Spatial Uncertainty (SU) as described in [3]
- Naive fusion of Spatial and Temporal Uncertainty (SU+TU), a pixel-wise addition of TU and SU maps
- Entropy-based Uncertainty (EU) [23]
- Local variance of spatiotemporal neighborhood, labeled *Baseline*
- Multi-factor Uncertainty (mU) as described in 6.2

The performance of these algorithms is quantified in terms of Area-Under-the-Curve (AUC) values of their corresponding Receiver-Operating-Characteristic (ROC) curves. We, also, show effects of kernel size of the feature map $\boldsymbol{F_{STU}}$ on the proposed algorithm's performance. Details on data and experiments procedure are provided in the datasets section while the performance evaluation methodology is detailed in 5.3.The second set of experiments are designed to show performance of the proposed uncertainty estimation algorithm given different categories of videos. The third set of experiments verifies the performance of the proposed algorithms using additional datasets and saliency detection models.

6.3.1 Datasets and Experiment Setup

We tested the proposed multi-factor uncertainty estimation algorithm using three publicly available databases: CRCNS [46], DIEM [66], and AVD [65]. The CRCNS [46] database includes 50 videos, with the resolution being 480×640 and the duration ranging from 5 to 90 seconds with 30 frames per second. The videos contents are diverse with a total of 12 categories ranging from street scenes to video games and from TV sports to TV news. In many cases the videos contain variations of lighting conditions, severe camera movements, and high motion blur effects. Eye fixation data are provided with each video, recorded for a group of eight human subjects watching the videos under task-free view condition. The DIEM [66] database includes 85 videos, with varying resolutions and duration up to 130 seconds with 30 frames per second. The videos content are mainly limited to TV and film content including film trailers, music videos, and advertisement. The eye fixation data are collected from 250 participants under task-free view conditions. The AVD [65] database includes 148 videos, with varying resolutions and mean duration of 22 seconds with 30 frames per second. The video contents are limited to moving objects, landscape, and faces. The eye fixations data are collected from 176 observers. The AVD dataset contains two sets of videos of the same visual content but one with audio and the other without. According to their findings on the effect of audio on the attention of the participants, we only select the videos without associated audio.

For our experiments, we generated saliency maps for the videos using a recent algorithm based on 3D FFT local spectra (3DFFT) [61]. However, for validation, we also share the results from two additional saliency models: STSR [77] and PQFT [31], which are shown at the end of this section. In most of our experiments, the saliency maps are reduced in size to three different scales. *Scale 1* is of size 12×16; a downscale of frames original size 480×640, where every 40×40 region in the original frame corresponds to a single pixel in *Scale 1*. Similarly, *Scale 2* saliency maps are 24×32, where every pixel is equivalent to 20×20 region of pixels in the original sized frame, and *Scale 3* saliency maps are 48×64, where every pixel is equivalent to 10×10 regions. Unless stated otherwise, saliency maps used in all experiments are generated from downscaled frames to be *Scale 1* and generated using 3DFFT. As for the algorithm parameters, we are using an ensemble of size 65, 3×3 local neighborhood of the feature maps which results in feature vector $x_i \in \mathbb{R}^{27}$, and the reported results are computed using 5-fold cross-validation tests.

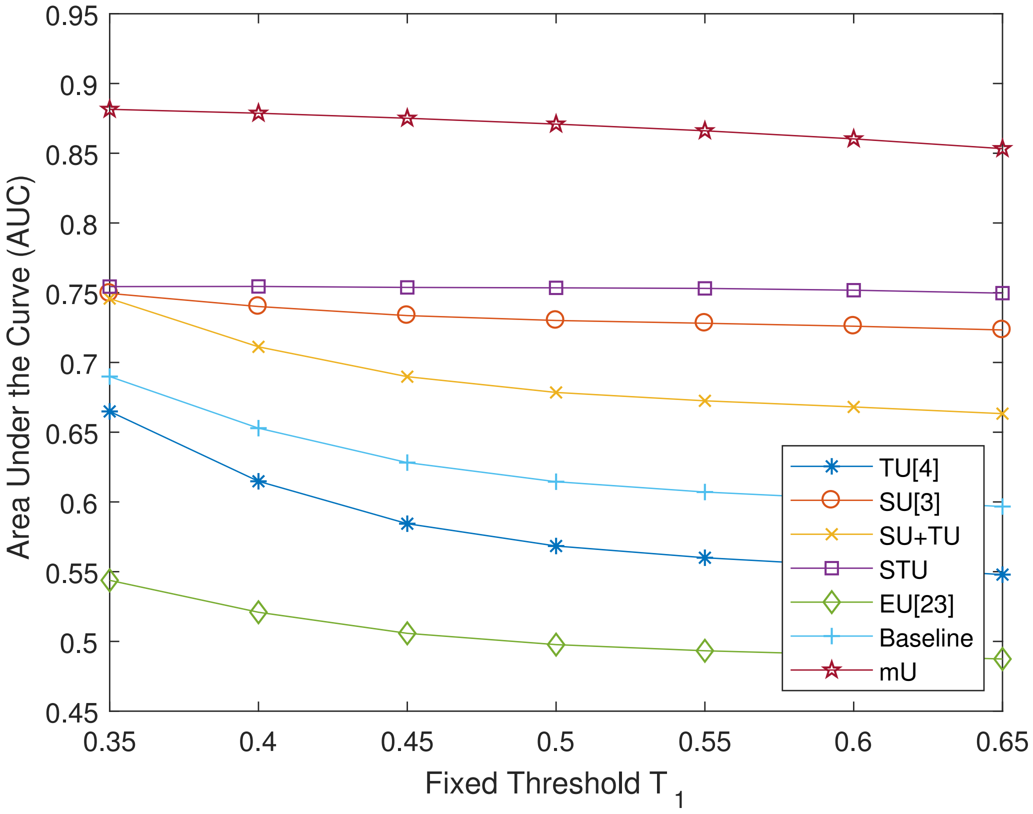


Figure 64: Examples illustrating that relative uncertainty estimation performance is independent of fixed threshold T_1 applied to true uncertainty.

6.3.2 Threshold Selection

The performance evaluation procedure described earlier utilizes a fixed threshold T_1 to transform the continuous valued true uncertainty $\boldsymbol{U}^{tr}$ to binary ground truth. First, we examine the impact of changing the value of T_1. The algorithms under consideration are: Temporal Uncertainty (TU), Spatial Uncertainty (SU), Fused Spatial and Temporal Uncertainty (SU+TU), Spatiotemporal Uncertainty (STU), Spatiotemporal local variance ($Baseline$) computed on the same neighborhood as STU, Entropy-based Uncertainty (EU), and Multi-factor Uncertainty (mU). Figure 64 shows the performance of these algorithms in terms of AUC versus T_1. As shown in Figure 64, T_1 directly affects AUC value; as the value of T_1 increases, the AUC

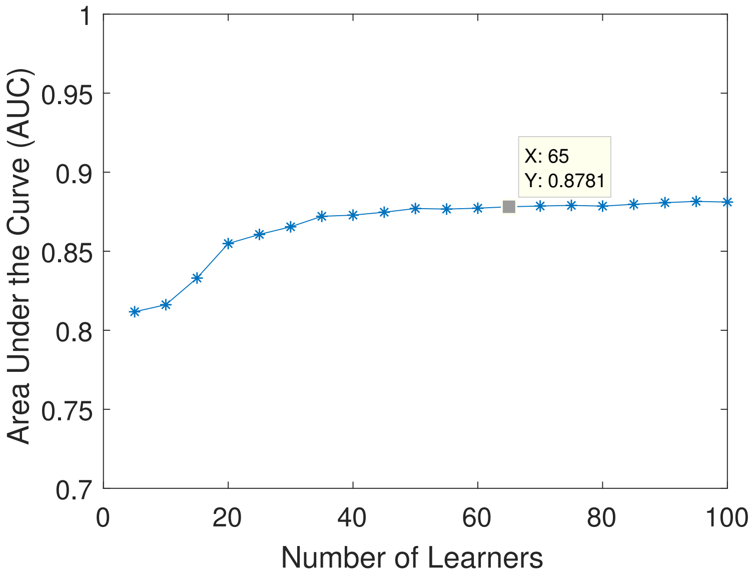


Figure 65: The effect of learning ensemble size on the performance of the proposed algorithm.

value of all algorithms considered here decreases. It is also interesting to point out that the gradient of AUC levels-off as T_1 reaches higher values. Although we can see that T_1 value significantly changes AUC, conclusions based on relative AUC values are consistent regardless of the value of T_1. As shown in Figure 64, mU outperforms all other algorithms, by a big margin, while EU is performing the worst in this experiment. Please note that the reported AUC results are for *Scale 1* maps with averaging kernel of length 5 for TU, of size 5×5 for SU, $5 \times 5 \times 5$ for STU, $3 \times 3 \times 3$ for mU.

6.3.3 Number of Learners

In most of our experiments, we use an ensemble of size 65, which we found to produce the highest results. As shown in Figure 65, the effect of the ensemble size is noticeable. As we increase the size of the ensemble, the AUC values increase until an ensemble of size 50, after which the AUC values levels off and doesn't change much.

6.3.4 Kernel Size

Moreover, kernel size affects the performance of the proposed algorithm as well. Figure 66 shows the performance of the estimation algorithms under test, in terms of AUC values, when the estimation kernel size is changed. The experiment is conducted using *scale 2* saliency map and variable kernel size r (r for TU, $r \times r$ for SU, and $r \times r \times r$ for both STU and mU). As shown in Figure 66, AUC of the proposed algorithm changes as the size of the kernel changes. However, the change in TU performance is significantly smaller than that of SU and STU because the number of pixels added into SU and STU kernels is significantly more than the number of pixels added to TU kernel. There is, however, a slight degradation in TU performance as the kernel size increases (starting from $L_t = 13$ onwards), which can be attributed to including less relevant pixel in the estimation process as the kernel size increase. For kernels of sizes $3 \times 3 \times 3$ till $11 \times 11 \times 11$, it can be seen that STU achieves higher AUC than SU. However, such trend inverts starting from kernel size $13 \times 13 \times 13$ onwards. However, the negative effects of the kernel size increase, in the case of STU, does not affect the proposed mU. In fact, as the size of the kernel increase the performance of mU improves. This might be explained by noting that the estimation in mU algorithm, unlike STU, is highly non-linear because of the use of an ensemble of decision trees which are able to infer higher order relationships between features [10].

6.3.5 Video Categories

Given the diverse nature of scenes and dynamics in the CRCNS dataset, we evaluate the performance of our proposed algorithm for each category in the dataset. For these experiments, we set $T_1 = 0.55$ and use *Scale 1* saliency maps. Table 4 shows AUC values for TU ($L_t = 5$), SU ($L_{s_x} = 5$), *ST+SU*, STU ($L_{st_x} = 5$), *EU*, *Baseline* (L_{st_x} $= 5$), and mU ($L_{st_x} = 5$), for each category, separately. As shown in Table 4, AUC

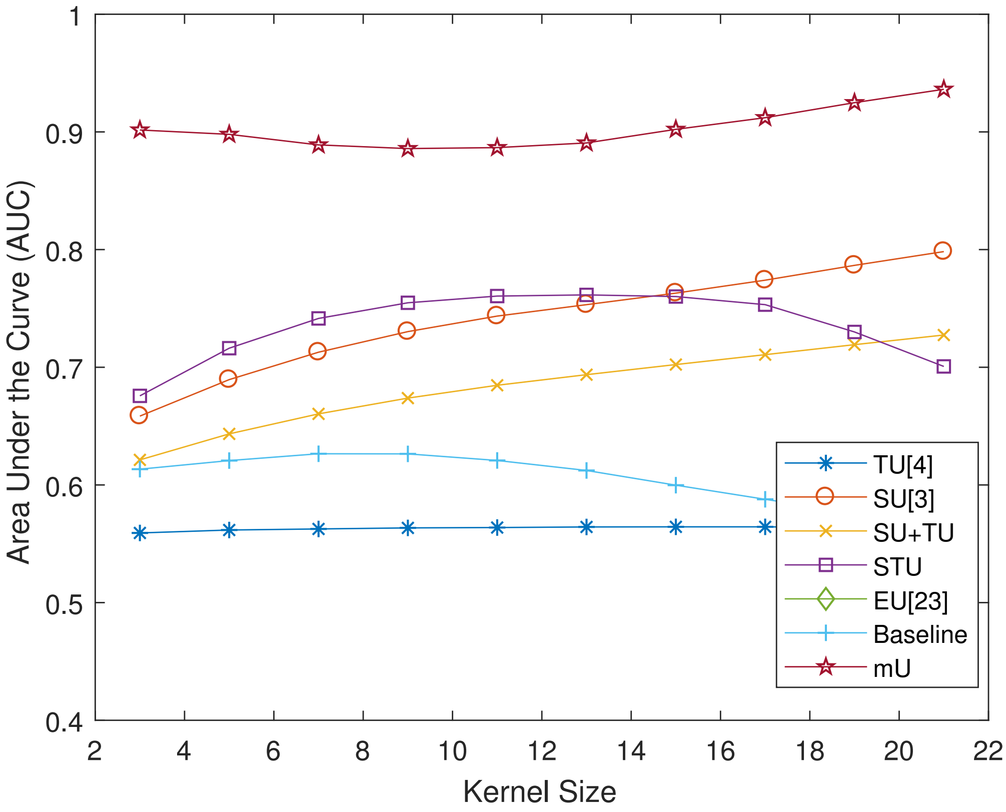


Figure 66: AUC value is affected by the choice of the kernel size at the same scale. Results reported here use *Scale 2* saliency maps and $T_1 = 0.55$.

values for the proposed algorithm are above 0.5, indicating that the proposed algorithm is advantageous over random guessing. Additionally, the algorithm performs better than EU in every category and in some by a wide margin. One interesting result is that AUC for Saccadetest video is significantly higher than other categories for all algorithms considered here. This can be attributed to its non-complex structure, which shows a disk moving against a light textured background. Notably, mU achieves highest performance in most categories. Comparing STU and mU, we find that mU outperforms STU in the categories where motion plays a major role in explaining the scene, like gamecube, saccadetest, and tv-sports.

6.3.6 Comparison across various datasets and saliency models

In this section, we present evaluation results for the proposed algorithm across various datasets. We compare the performance of the proposed algorithm using videos from three datasets: CRCNS [46], DIEM [66], and AVD [65]. Figure 68 shows the the AUC values of the six uncertainty estimation methods using videos from the three datasets. In Figure 68, mU performance is the highest across all datasets. In general, the trend and ranking between the uncertainty estimation methods is consistent across the three datasets.

Additionally, we present the evaluation results for the proposed algorithm using three saliency models: 3DFFT [61], STSR [77], and PQFT [31]. Figure 67 shows the AUC values of the six uncertainty estimation algorithms. In Figure 67, a consistent trend and ranking between the six algorithms exist across all three saliency models, where mU achieves the highest AUC value.

Moreover, we evaluate the proposed algorithm, in terms of the computed uncertainty map distribution versus uncertainty ground truth maps distribution, using four distribution-based metrics; Jeffrey Divergence (JD), Jensen-Shannon divergence (JS), Histogram Intersection (HI), L2-norm. As shown in Table.5, the proposed algorithm

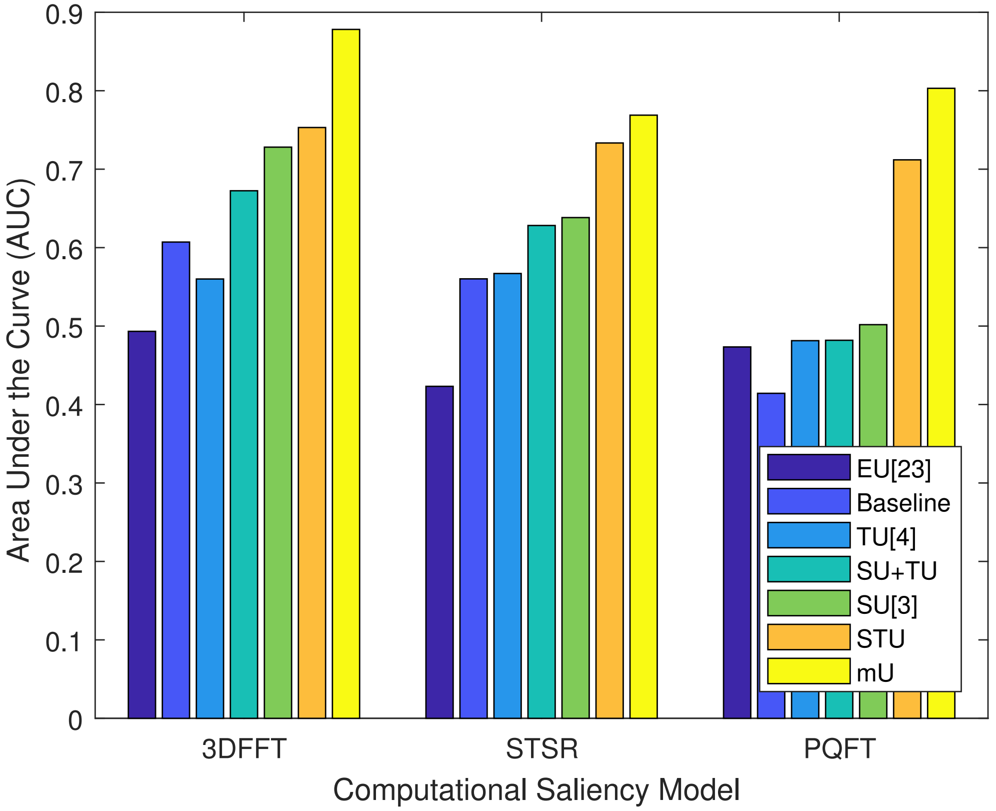


Figure 67: The performance of the proposed algorithm using the saliency models 3DFFT [61], STSR [77], PQFT [31]

is among the best algorithms that provides the closest distribution to that of the ground truth maps. Interestingly, even though mU has a higher performance, than STU, in AUC metric, this is not reflected in the distribution-based metrics. This is caused by the splitting and partitions of the feature space in CART models which produces a spiky histogram of the estimated labels [35], as shown in histogram plots in Figure 69. Even though the profile of the histogram of mU matches that of the ground truth closer than STU in many examples, it suffers from the problem of partitions of the feature space in CART models which generates the same estimation for a large range of input features. This leads to the spiky shape of the histogram shown in Figure 69.

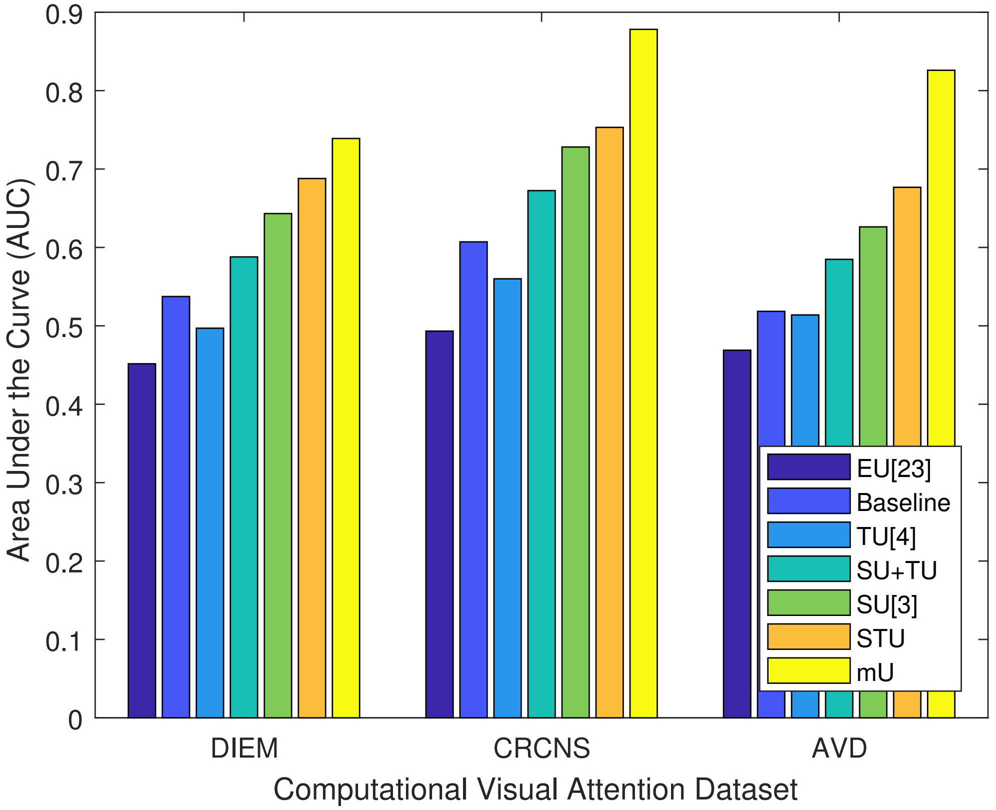


Figure 68: The performance of the proposed algorithm across the datasets CRCNS [46], DIEM [66], and AVD [65].

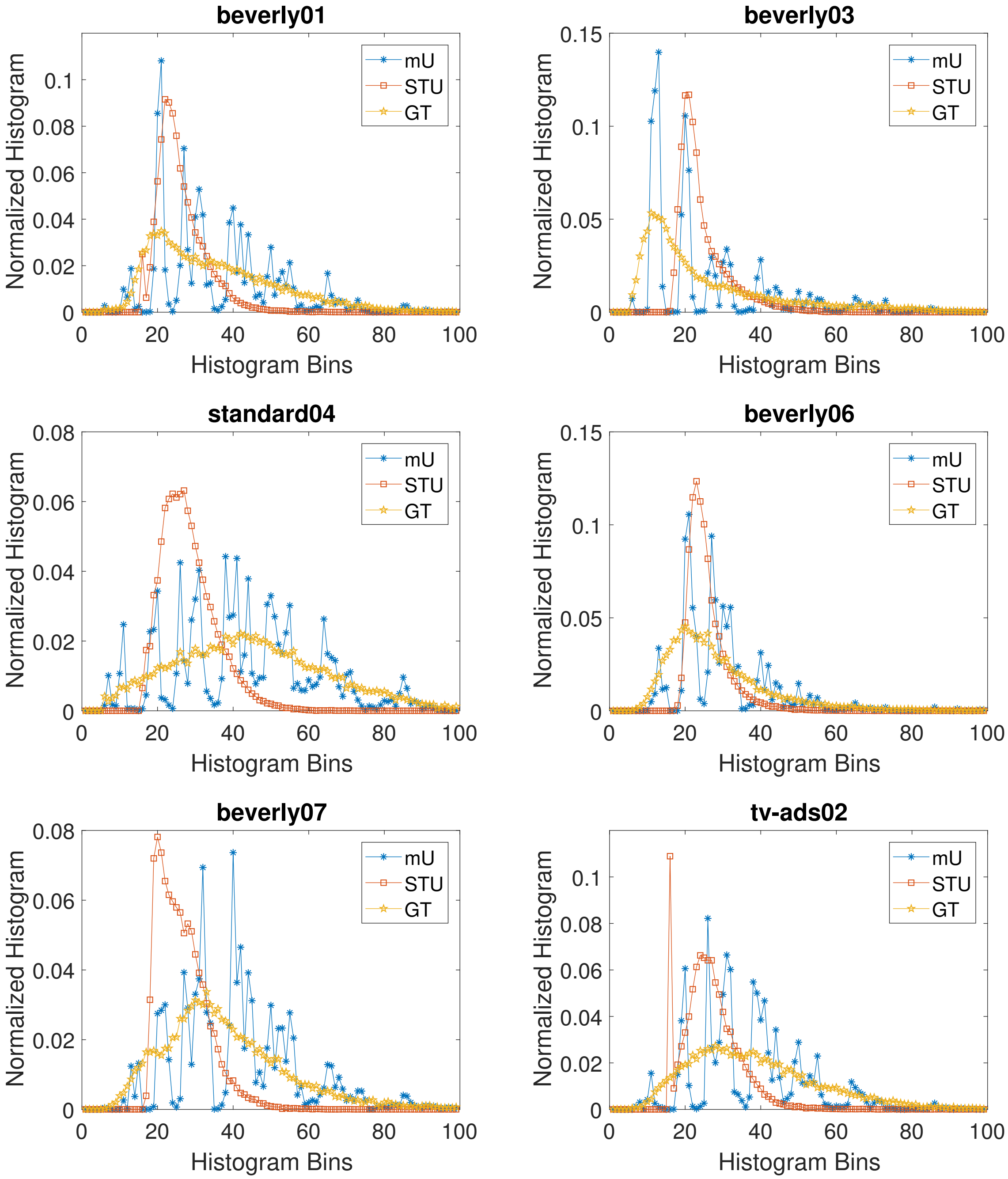


Figure 69: Comparing the histograms of the estimation results of the proposed prediction algorithm vs STU algorithm.

Table 4: List of AUC value for different categories using fixed threshold $T_1 = 0.55$ and *Scale 1* saliency maps. Note that the highest AUC value in each category is labeled in green and lowest AUC value in red. Also, the category with the highest AUC in the dataset is shown in bold

	TU [4]	*SU* [3]	*SU+TU*	*STU*	*EU* [23]	*Baseline*	*mU*
beverly	0.5793	0.8088	0.7174	0.8130	0.5835	0.6915	0.7748
gamecube	0.5987	0.7636	0.7155	0.7913	0.5906	0.6834	0.8562
monica	0.6152	0.7801	0.7240	0.7994	0.5728	0.6506	0.8014
saccadetest	**0.7722**	**0.8734**	**0.8216**	0.8587	**0.8458**	**0.8308**	**0.9259**
standard	0.5866	0.7190	0.6609	0.7462	0.5165	0.5841	0.7261
tv-action	0.7481	0.8466	0.7970	**0.8667**	0.7245	0.6491	0.8433
tv-ads	0.5565	0.7248	0.6565	0.7476	0.5228	0.5360	0.8342
tv-announce	0.4555	0.6679	0.5550	0.7321	0.4434	0.5818	0.7855
tv-music	0.5548	0.6721	0.6236	0.7427	0.4471	0.5771	0.8513
tv-news	0.5051	0.6497	0.5885	0.6947	0.4861	0.5029	0.7732
tv-sports	0.5156	0.6746	0.6170	0.7172	0.5020	0.5368	0.8251
tv-talk	0.5692	0.7142	0.6393	0.7364	0.5299	0.5250	0.8134

Table 5: Estimated distances using distribution-based metrics for the proposed algorithm in comparison with the state-of-the-art algorithms using *Scale 1* saliency maps computed using 3DFFT algorithm [61]. Note that the highest value in each distance metric is labeled in green and the lowest value in red. Also, the distance values for the proposed algorithm is shown in bold

	CRCNS				DIEM				AVD			
Algorithms	JS	JD	HI	L2	JS	JD	HI	L2	JS	JD	HI	L2
TU [4]	0.34	0.68	0.64	0.25	0.16	0.32	0.30	0.12	0.33	0.66	0.60	0.24
SU [3]	0.14	0.29	0.32	0.12	0.05	0.09	0.13	0.04	0.09	0.18	0.24	0.08
SU+TU	0.14	0.27	0.32	0.12	0.05	0.11	0.14	0.05	0.10	0.20	0.28	0.10
STU	0.08	0.15	0.24	0.08	0.03	0.05	0.10	0.04	0.06	0.12	0.21	0.08
EU [23]	0.51	1.02	0.82	0.46	0.23	0.45	0.36	0.19	0.45	0.91	0.71	0.41
Baseline	0.38	0.76	0.68	0.32	0.15	0.30	0.29	0.13	0.29	0.58	0.57	0.25
mU	**0.14**	**0.28**	**0.42**	**0.15**	**0.06**	**0.12**	**0.17**	**0.06**	**0.07**	**0.13**	**0.25**	**0.09**

CHAPTER VII

CONCLUSIONS

7.1 Contributions

Saliency detection has been incorporated into various image and video processing algorithms to improve performance and mimic the advanced processing capability of the human vision system (HVS). However, there has been no explicit design of a saliency-based video processing framework, to the best of our knowledge. Most of the proposed methods do not analyze the validity of saliency maps generated online, but rather design or choose a saliency detection algorithm that exhibits good performance in evaluation datasets, and then hope for the best when the algorithm goes online. Therefore, we propose a unified framework for enhancing video processing algorithms using saliency that is neither application- nor algorithm-specific and can be reliable in real world scenarios. The proposed framework analyzes the saliency map and produces associated uncertainty map that describes the level of confidence in the generated saliency map. By reliably estimating uncertainty, we can expand the framework to include a systematic decision-making procedure that makes application-specific decisions. Additionally, having a separate module for decision making helps clarify which assumptions are application-specific and which ones are saliency related. Knowledge about the application space can influence the design of this module without making drastic changes to the whole framework. Similarly, the availability of uncertainty estimations allows for risk assessment that can be used to guide the optimization of video processing algorithms. We expand the framework beyond relying only on saliency maps for estimating uncertainty, to rely on the motion information extracted from video frames.

We address the problem of quantifying the uncertainty of computational saliency for videos by, first, exploring the spatial correlations in both the saliency map and the eye-fixation map. Then, we learn the spatiotemporal correlations that define a reliable saliency map. We study spatiotemporal eye-fixation data from the public CRCNS dataset and investigate a common feature in human visual attention, without making any assumption about the structure of the eye-fixation maps. By employing basic concepts from information theory, the analysis shows a substantial correlation between the saliency of a pixel and the saliency of its neighborhood. Our experiments showed a reduction of roughly 50%, across all videos, in pixel entropy when conditioned on its local neighbors' average. The experiment shows that local correlation exists in saliency perceived by HVS. Thus, saliency map's pixels ought to be highly correlated with their local neighbors. The analysis also provides insights into the structure and dynamics of the eye-fixation maps.

Motivated by this analysis, we investigated the predictability of spatial correlation of eye-fixation maps using motion cues extracted from optical flow maps. We showed that motion in some video sequences dominates as the main salient stimuli, while other sequences have different stimuli. Our research might provide an alternative quantitative approach to describing human attention. We believe such an approach is very important for many saliency applications. For example, uncertain ground truth data can be validated based on this approach. The various correlations discussed in the paper can also be used as measures of the reliability of detected saliency, thus being a guide for optimizing saliency-based video processing.

Based on the self-correlation study, we design a feature that estimates a pixel-wise uncertainty map that reflects our supposed confidence in the associated computational saliency map by relating a pixel's saliency to the saliency of its direct neighbors. To estimate such uncertainties, we measure the divergence of a pixel, in a saliency map, from its local neighborhood. Additionally, we propose a systematic procedure to evaluate uncertainty estimation performance by explicitly computing uncertainty ground truth as a function of a given saliency map and eye fixations of human subjects. In our experiments, we explore multiple definitions of locality and neighborhoods in spatiotemporal video signals. In addition, we examine the relationship between the parameters of our proposed feature and the content of the videos. The proposed feature is developed in an unsupervised fashion, making it more suitable for generalization to most natural videos. Also, it is computationally efficient and flexible for customization to specific video content. Experiments using three publicly available video datasets show that the proposed feature outperforms state-of-the-art uncertainty estimation methods with improvement in accuracy up to 75% and offers efficiency and flexibility that make it more useful in practical situations.

Finally, we combine the analysis and the features designed in our experiments on saliency maps into a unified multi-factor uncertainty estimation algorithm based on Gradient Boosting Trees (GBT). The proposed algorithm utilizes spatiotemporal cues computed from the saliency maps, scene motion estimated by optical flow maps, and camera motion maps. The proposed algorithm outperforms state-of-the-art uncertainty estimation method by a wide margin and achieves up to 93% accuracy. To summarize, we have the following contributions:

1. Uncertainty-based framework for improving saliency applications reliability

2. Correlation analysis of eye-fixation maps on video datasets

3. Novel Unsupervised video feature for uncertainty estimation

4. Prediction model for eye-fixation correlation using optical flow maps

5. Multi-factor uncertainty estimation of video saliency maps

6. Application-independent uncertainty evaluation procedure

7.2 Prospective Research Directions

Given the limited work in uncertainty estimation in visual saliency applications, there are many aspects that needs further study and analysis. In Chapter 3, we presented a study on the self-correlation of eye-fixation maps and saliency maps without relying on assumptions regarding the structure of eye-fixation maps. However, further analysis on this correlation is needed to gain a deeper understanding. For example, a high level model that links spatiotemporal events in the eye-fixation maps with each others and how they affect the global attention given the well-known Winner Takes All (WTA) phenomenon in visual attention research.

In Chapter 4, we showed that the self-correlation of eye-fixation maps can be predicted from motion cues using a simple linear regression model. Therefore, a continuation of this work would be to analyze the predictability of the self-correlation maps by analyzing higher-order visual attention targets such as faces and text.Such extension can improve our understanding of the predictability of eye-fixation maps and ultimately improves our ability to estimate uncertainty in saliency maps.

REFERENCES

[1] Achanta, R., Hemami, S., Estrada, F., and Susstrunk, S., "Frequency-tuned salient region detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1597–1604, June 2009.

[2] Alsam, A. and Sharma, P., "Robust metric for the evaluation of visual saliency algorithms," *JOSA A*, vol. 31, no. 3, pp. 532–540, 2014.

[3] Alshawi, T., Long, Z., and AlRegib, G., "Unsupervised uncertainty analysis for video saliency detection," in *the 49th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, Nov. 8-11*, IEEE, 2015.

[4] Alshawi, T., Long, Z., and AlRegib, G., "Unsupervised uncertainty estimation in saliency detection for videos using temporal cues," in *IEEE Global Conf. on Signal and Information Processing (GlobalSIP), Orlando, Florida, Dec. 14-16*, SPIE, 2015.

[5] Alshawi, T., Long, Z., and AlRegib, G., "Understanding spatial correlation in eye-fixation maps for visual attention in videos," in *2016 IEEE International Conference on Multimedia and Expo (ICME), Seattle, WA*, IEEE, 2016.

[6] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. J., and Szeliski, R., "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[7] Bharath, R., Nicholas, L., and Cheng, X., "Scalable scene understanding using saliency-guided object localization," in *Control and Automation (ICCA), 2013 10th IEEE International Conference on*, pp. 1503–1508, June 2013.

[8] Borji, A., Tavakoli, H., Sihite, D., and Itti, L., "Analysis of scores, datasets, and models in visual saliency prediction," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 921–928, Dec 2013.

[9] Borji, A. and Itti, L., "State-of-the-art in visual attention modeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 185–207, 2013.

[10] Breiman, L., *Classification and regression trees.* Routledge, 1984.

[11] Brox, T., Bruhn, A., Papenberg, N., and Weickert, J., "High accuracy optical flow estimation based on a theory for warping," in *European conference on computer vision*, pp. 25–36, Springer, 2004.

[12] Bruce, N. and Tsotsos, J., "Saliency based on information maximization," in *Advances in neural information processing systems*, pp. 155–162, 2005.

[13] Bruhn, A., Weickert, J., and Schnörr, C., "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[14] Cerf, M., Frady, E. P., and Koch, C., "Faces and text attract gaze independent of the task: Experimental data and computer model," *Journal of vision*, vol. 9, no. 12, pp. 10–10, 2009.

[15] Cerf, M., Harel, J., Einhäuser, W., and Koch, C., "Predicting human gaze using low-level saliency combined with face detection," in *Advances in neural information processing systems*, pp. 241–248, 2008.

[16] cisco, "The Zettabyte Era Trends and Analysis," tech. rep., Cisco Virtual Network Index, 06 2016.

[17] Clarke, A. D. F., Elsner, M., and Rohde, H., "Where's wally: the influence of visual salience on referring expression generation," *Frontiers in psychology*, vol. 4, p. 329, 2013.

[18] Debattista, K., Chalmers, A., Gillibrand, R., Longhurst, P., Mastoropoulou, G., and Sundstedt, V., "Parallel selective rendering of high-fidelity virtual environments," *Parallel Computing*, vol. 33, no. 6, pp. 361–376, 2007.

[19] Dempster, A. P., "Upper and lower probabilities induced by a multivalued mapping," *The annals of mathematical statistics*, pp. 325–339, 1967.

[20] Dubois, D., "Possibility theory and statistical reasoning," *Computational statistics & data analysis*, vol. 51, no. 1, pp. 47–69, 2006.

[21] Duncan, J. and Humphreys, G. W., "Visual search and stimulus similarity.," *Psychological review*, vol. 96, no. 3, p. 433, 1989.

[22] Ell, T. A. and Sangwine, S. J., "Hypercomplex fourier transforms of color images," *IEEE Transactions on image processing*, vol. 16, no. 1, pp. 22–35, 2007.

[23] Fang, Y., Wang, Z., Lin, W., and Fang, Z., "Video saliency incorporating spatiotemporal cues and uncertainty weighting," *IEEE Transactions on Image Processing*, vol. 23, pp. 3910–3921, Sept 2014.

[24] Fang, Y., Wang, Z., and Lin, W., "Video saliency incorporating spatiotemporal cues and uncertainty weighting," in *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pp. 1–6, IEEE, 2013.

[25] Ferson, S. and Ginzburg, L. R., "Different methods are needed to propagate ignorance and variability," *Reliability Engineering & System Safety*, vol. 54, no. 2-3, pp. 133–144, 1996.

[26] Fleet, D. and Weiss, Y., "Optical flow estimation," in *Handbook of mathematical models in computer vision*, pp. 237–257, Springer, 2006.

[27] Friedman, J. H., "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

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

[29] Gitman, Y., Erofeev, M., Vatolin, D., Andrey, B., and Alexey, F., "Semiautomatic visual-attention modeling and its application to video compression," in *Image Processing (ICIP), 2014 IEEE International Conference on*, pp. 1105–1109, Oct 2014.

[30] Gu, Y., Jin, Z., and Chiu, S. C., "Active learning combining uncertainty and diversity for multi-class image classification," *IET Computer Vision*, vol. 9, no. 3, pp. 400–407, 2015.

[31] Guo, C., Ma, Q., and Zhang, L., "Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2008.

[32] Guo, C. and Zhang, L., "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *Image Processing, IEEE Transactions on*, vol. 19, pp. 185–198, Jan 2010.

[33] Haeusler, R., Nair, R., and Kondermann, D., "Ensemble learning for confidence measures in stereo vision," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 305–312, June 2013.

[34] Harel, J., Koch, C., and Perona, P., "Graph-based visual saliency," in *Advances in neural information processing systems*, pp. 545–552, 2006.

[35] Hastie, T., Tibshirani, R., and Friedman, J., *The elements of statistical learning: data mining, inference and prediction.* Springer, 2005.

[36] Healey, C. and Enns, J., "Attention and visual memory in visualization and computer graphics," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 7, pp. 1170–1188, 2012.

[37] Helton, J. C., Johnson, J. D., and Oberkampf, W. L., "An exploration of alternative approaches to the representation of uncertainty in model predictions," *Reliability Engineering & System Safety*, vol. 85, no. 1, pp. 39–71, 2004.

[38] Horn, B., *Robot Vision.* Cambridge, MA, USA: MIT Press, 1986.

[39] Horn, B. K. and Schunck, B. G., "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[40] Hou, X. and Zhang, L., "Saliency detection: A spectral residual approach," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2007.

[41] Hou, X., Harel, J., and Koch, C., "Image signature: Highlighting sparse salient regions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 1, pp. 194–201, 2012.

[42] Hu, X. and Mordohai, P., "A quantitative evaluation of confidence measures for stereo vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 2121–2133, Nov 2012.

[43] Huang, C. R., Chang, Y. J., Yang, Z. X., and Lin, Y. Y., "Video saliency map detection by dominant camera motion removal," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, pp. 1336–1349, Aug 2014.

[44] Huang, L. and Pashler, H., "A boolean map theory of visual attention.," *Psychological review*, vol. 114, no. 3, p. 599, 2007.

[45] Itti, L., Koch, C., and Niebur, E., "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1254–1259, Nov 1998.

[46] Itti, L., "Eye-tracking data from human volunteers watching complex video stimuli."

[47] Itti, L., "Quantifying the contribution of low-level saliency to human eye movements in dynamic scenes," *Visual Cognition*, vol. 12, no. 6, pp. 1093–1123, 2005.

[48] Judd, T., Durand, F., and Torralba, A., "Fixations on low-resolution images," *Journal of Vision*, vol. 11, no. 4, pp. 14–14, 2011.

[49] Judd, T., Ehinger, K., Durand, F., and Torralba, A., "Learning to predict where humans look," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2106–2113, Sept 2009.

[50] Julesz, B., "Experiments in the visual perception of texture," *Scientific American*, vol. 232, no. 4, pp. 34–43, 1975.

[51] Julesz, B., "A theory of preattentive texture discrimination based on first-order statistics of textons," *Biological cybernetics*, vol. 41, no. 2, pp. 131–138, 1981.

[52] Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N., "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[53] Kim, H., Kim, Y., Sim, J. Y., and Kim, C. S., "Spatiotemporal saliency detection for video sequences based on random walk with restart," *IEEE Transactions on Image Processing*, vol. 24, pp. 2552–2564, Aug 2015.

[54] Li, Y., Chen, J., and Feng, L., "Dealing with uncertainty: A survey of theories and practices," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2463–2482, 2013.

[55] Liang, M. and Hu, X., "Predicting eye fixations with higher-level visual features," *IEEE Transactions on Image Processing*, vol. 24, pp. 1178–1189, March 2015.

[56] Lin, J. Y., Liu, T. J., Lin, W., and Kuo, C.-C. J., "Visual-saliency-enhanced image quality assessment indices," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pp. 1–4, IEEE, 2013.

[57] Lindley, D. V., *Understanding uncertainty.* John Wiley & Sons, 2006.

[58] Liu, C., *Beyond pixels: exploring new representations and applications for motion analysis.* PhD thesis, Citeseer, 2009.

[59] Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., and Shum, H.-Y., "Learning to detect a salient object," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, pp. 353–367, Feb 2011.

[60] Liu, Z., Zhang, X., Luo, S., and Meur, O. L., "Superpixel-based spatiotemporal saliency detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, pp. 1522–1540, Sept 2014.

[61] Long, Z. and AlRegib, G., "Saliency detection for videos using 3D fft local spectra," in *Human Vision and Electronic Imaging XX, SPIE Electronic Imaging*, SPIE, 2015.

[62] Lucas, B. D., Kanade, T., and others, "An iterative image registration technique with an application to stereo vision.," in *International Joint Conference on Artificial Intelligence*, vol. 81, pp. 674–679, 1981.

[63] Mahadevan, V. and Vasconcelos, N., "Biologically inspired object tracking using center-surround saliency mechanisms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, pp. 541–554, March 2013.

[64] Mahapatra, D., Gilani, S. O., and Saini, M. K., "Coherency based spatio-temporal saliency detection for video object segmentation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, pp. 454–462, June 2014.

[65] Marighetto, P., Coutrot, A., Riche, N., Guyader, N., Mancas, M., Gosselin, B., and Laganiere, R., "Audio-visual attention: Eye-tracking dataset and analysis toolbox," in *Image Processing (ICIP), 2017 IEEE International Conference on*, pp. 1802–1806, September 2017.

[66] Mital, P. K., Smith, T. J., Hill, R., and Henderson, J. M., "Clustering of gaze during dynamic scene viewing is predicted by motion," *Cognitive Computation*, vol. 3, pp. 5–24, march 2011.

[67] Mital, P. K., Smith, T. J., Hill, R. L., and Henderson, J. M., "Clustering of gaze during dynamic scene viewing is predicted by motion," *Cognitive Computation*, vol. 3, no. 1, pp. 5–24, 2011.

[68] Moens, D. and Vandepitte, D., "A survey of non-probabilistic uncertainty treatment in finite element analysis," *Computer methods in applied mechanics and engineering*, vol. 194, no. 12, pp. 1527–1555, 2005.

[69] Moore, R. E., *Methods and applications of interval analysis*, vol. 2. Siam, 1979.

[70] Nygård, G. E., Sassi, M., and Wagemans, J., "The influence of orientation and contrast flicker on contour saliency of outlines of everyday objects," *Vision Research*, vol. 51, no. 1, pp. 65–73, 2011.

[71] Peng, J. and Xiao-Lin, Q., "Keyframe-based video summary using visual attention clues," *IEEE MultiMedia*, vol. 17, no. 2, pp. 64–73, 2010.

[72] Quinlan, P. T. and Humphreys, G. W., "Visual search for targets defined by combinations of color, shape, and size: An examination of the task constraints on feature and conjunction searches," *Perception & psychophysics*, vol. 41, no. 5, pp. 455–472, 1987.

[73] R., S., Katti, H., Sebe, N., Kankanhalli, M., and Chua, T., "An eye fixation database for saliency detection in images," in *European Conference on Computer Vision (ECCV), 2010*, 2010.

[74] Ren, Z., Gao, S., Chia, L.-T., and Tsang, I.-H., "Region-based saliency detection and its application in object recognition," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 24, pp. 769–779, May 2014.

[75] Salehin, M. M. and Paul, M., "Summarizing surveillance video by saliency transition and moving object information," in *Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on*, pp. 1–8, IEEE, 2015.

[76] Saygili, G., Staring, M., and Hendriks, E. A., "Confidence estimation for medical image registration based on stereo confidences," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 539–549, Feb 2016.

[77] Seo, H. J. and Milanfar, P., "Static and space-time visual saliency detection by self-resemblance," *Journal of Vision*, vol. 9, no. 12, p. 15, 2009.

[78] Shafer, G. and others, *A mathematical theory of evidence*, vol. 1. Princeton university press Princeton, 1976.

[79] Shafiq, M. A., Alshawi, T., Long, Z., and AlRegib, G., "The role of visual saliency in the automation of seismic interpretation," *Geophysical Prospecting*.

[80] Sharma, P., Cheikh, F., and Hardeberg, J., "Spatio-temporal analysis of eye fixations data in images," in *Image Processing (ICIP), 2014 IEEE International Conference on*, pp. 1150–1154, Oct 2014.

[81] Stentiford, F., "Attention based auto image cropping," in *Workshop on Computational Attention and Applications*, p. 2, ICVS, 2007.

[82] Treisman, A. and Gormican, S., "Feature analysis in early vision: evidence from search asymmetries.," *Psychological review*, vol. 95, no. 1, p. 15, 1988.

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

[84] Turek, R., "What youtube looks like in a day," 2015.

[85] Walley, P., "Towards a unified theory of imprecise probability," *International Journal of Approximate Reasoning*, vol. 24, no. 2-3, pp. 125–148, 2000.

[86] Wang, W., Shen, J., and Shao, L., "Consistent video saliency using local gradient flow optimization and global refinement," *IEEE Transactions on Image Processing*, vol. 24, pp. 4185–4196, Nov 2015.

[87] Wang, W., Shen, J., and Porikli, F., "Saliency-aware geodesic video object segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 3395–3402, June 2015.

[88] Wang, Z., Lu, L., and Bovik, A. C., "Foveation scalable video coding with automatic fixation selection," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 243–254, 2003.

[89] Wolfe, J. M., "Guided search 2.0 a revised model of visual search," *Psychonomic bulletin & review*, vol. 1, no. 2, pp. 202–238, 1994.

[90] Wolfe, J. M., Cave, K. R., and Franzel, S. L., "Guided search: an alternative to the feature integration model for visual search.," *Journal of Experimental Psychology: Human perception and performance*, vol. 15, no. 3, p. 419, 1989.

[91] Wu, C. T., Crouzet, S. M., Thorpe, S. J., and Fabre-Thorpe, M., "At 120 msec you can spot the animal but you don't yet know it's a dog," *Journal of Cognitive Neuroscience*, vol. 27, pp. 141–149, Jan 2015.

[92] Wu, J., Apostolakis, G., and Okrent, D., "Uncertainties in system analysis: probabilistic versus nonprobabilistic theories," *Reliability Engineering & System Safety*, vol. 30, no. 1-3, pp. 163–181, 1990.

[93] Zadeh, L. A., "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[94] Zadeh, L. A., "A simple view of the dempster-shafer theory of evidence and its implication for the rule of combination," *AI magazine*, vol. 7, no. 2, p. 85, 1986.

[95] Zhang, L., Tong, M. H., Marks, T. K., Shan, H., and Cottrell, G. W., "Sun: A bayesian framework for saliency using natural statistics," *Journal of vision*, vol. 8, no. 7, pp. 32–32, 2008.

[96] Zhang, W., Borji, A., Wang, Z., Le Callet, P., and Liu, H., "The application of visual saliency models in objective image quality assessment: A statistical evaluation," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[97] Zio, E. and Pedroni, N., *Literature review of methods for representing uncertainty.* FonCSI, 2013.

VITA

Tariq Alshawi received an M.S. degree with a minor in Mathematics in 2013 from the school of Electrical Engineering and Computer Science in University of Michigan, Ann Arbor, and a PhD degree in 2018 from the school of Electrical and Computer Engineering in Georgia Institute of Technology, Atlanta. Mr. Alshawi worked on various projects including perceived image quality assessment, human visual attention-based image and video processing and computer vision, uncertainty representation and estimation, medical imaging, and seismic interpretation. Mr. Alshawi's research interests goes beyond image and video processing to include electroencephalogram(EEG) seizure prediction, interference cancellation in multi-user wireless communication systems, and low-complexity real-time realization of signal processing systems.