Saliency and Tracking in Compressed Video

by

Sayed Hossein Khatoonabadi

M.Sc., Amirkabir University of Technology, 2005

B.Sc., University of Kerman, 2002

Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Doctor of Philosophy

in the
School of Engineering Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2015

APPROVAL

Name: Sayed Hossein Khatoonabadi

Degree: Doctor of Philosophy

Title: Saliency and Tracking in Compressed Video

Examining Committee: **Chair**: Dr. Rodney G. Vaughan

Professor, Simon Fraser University

Dr. Ivan V. Bajić

Associate Professor, Simon Fraser University
Senior Supervisor

Dr. Parvaneh Saeedi

Associate Professor, Simon Fraser University
Supervisor

Dr. Nuno Vasconcelos

Professor, University of California, San Diego
Supervisor

Dr. Jie Liang

Associate Professor, Simon Fraser University
Internal Examiner

Dr. Z. Jane Wang

Professor, University of British Columbia
External Examiner

Date Approved: June 30th, 2015

Abstract

Visual saliency is the propensity of a part of the scene to attract attention. Computational modeling of visual saliency has become an important research problem in recent years, with applications in quality assessment, compression, object tracking, and so on. While most saliency estimation models for dynamic scenes operate on raw video, their high computational complexity is a serious drawback when it comes to practical applications. Our approach for decreasing the complexity and memory requirements is to avoid decoding the compressed bitstream as much as possible. Since most modern cameras incorporate video encoders, this paves the way for in-camera saliency estimation, which could be useful in a variety of computer vision applications. In this dissertation we present compressed-domain features that are highly indicative of saliency in natural video. Using these features, we construct two simple and effective saliency estimation models for compressed video. The proposed models have been extensively tested on two ground truth datasets using several accuracy metrics, and shown to yield considerable improvement over several state-of-the-art compressed-domain and pixel-domain saliency models. Another contribution is a tracking algorithm that also uses only compressed-domain information to isolate moving regions and estimate their trajectories. The algorithm has been tested on a number of standard sequences, and the results demonstrate its advantages over state-of-the-art for compressed-domain tracking and segmentation, with over 30% improvement in F-measure.

Keywords: Compressed-Domain Processing, Video Object Tracking, Visual Attention, Visual Saliency Modeling

To my mother and father for their constant support, encouragement, and love.

To my brothers for their constant friendship and belief in me and my dreams.

Acknowledgements

I am grateful for having the opportunity to experience a wonderful time at SFU, a pioneer research facility, and an exceptionally friendly environment. This has been and will always be a marvelous memorable experience.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Ivan V. Bajić, for his generous guidance and mentorship during my PhD. His insightful comments, invaluable assistance, support and enthusiasm kept me on track with this dissertation and provided me with numerous opportunities to learn and grow. Prof. Bajić's willingness to share his knowledge and the opportunities he has provided me is very much appreciated.

I wish to acknowledge Prof. Nuno Vasconcelos for giving me the opportunity to collaborate with Statistical Visual Computing Laboratory team at the University of California, San Diego under his supervision for six months. His valuable comments and superb ideas have undoubtedly had a great impact on the success of this project. I am very fortunate to have collaborated with and learned from one of the bests in the field.

It is my pleasure to thank my supervisory committee member, Prof. Parvaneh Saeedi, for her insightful comments on my research that greatly improved the clarity and the quality of this work. I would also like to thank Prof. Jie Liang, and Prof. Z. Jane Wang, my internal and external examiners, who offered priceless feedback, support and encouragement for my research. I also express my great thanks to Prof. Rodney G. Vaughan who kindly agreed to coordinate my defense. I hereby, would like to acknowledge the financial support I received from Cisco Systems, Inc. In particular, I owe thanks to Dr. Yufeng Shan for his support and encouragement.

At last, none of this would have been possible without the full support of my friends and family. I would like to thank them for their nonstop support and patronizing during my life. Unquestionably they have done their best to pave the road of success for me which

indeed walked me towards this stage, the place where they can finally see the results of their lifetime endeavors. For now, I can only hope to make them proud and to make sure they know how much I love them. Mom! more than anyone else, this thesis is dedicated to you.

Contents

List of Tables

List of Figures

List of Symbols

I	The intensity channel of an image
R	The red channel of an image
G	The green channel of an image
B	The blue channel of an image
M	The motion channel of a frame
F	The flicker channel of a frame
$\underline{R}$	The tuned color component of red
$\underline{G}$	The tuned color component of green
$\underline{B}$	The tuned color component of blue
$\underline{Y}$	The tuned color component of yellow
$\mathcal{I}$	The intensity feature map
$\mathcal{RG}$	The opponent color feature map of (red,green)
$\mathcal{BY}$	The opponent color feature map of (blue,yellow)
$\mathcal{O}_\theta$	The orientation feature map with the orientation of θ
$\mathcal{M}_\theta$	The motion feature map with the orientation of θ
$\mathcal{F}$	The flicker feature map
$\mathcal{K}$	A feature map
$\overline{\mathcal{I}}$	The intensity conspicuity map
$\overline{\mathcal{C}}$	The color conspicuity map
$\overline{\mathcal{O}}$	The orientation conspicuity map
$\overline{\mathcal{M}}$	The motion conspicuity map
$\overline{\mathcal{F}}$	The flicker conspicuity map
$\overline{\mathcal{K}}$	A conspicuity map
$\mathbf{C}_f$	The conspicuity map of feature f
$\mathcal{G}$	The ground-truth saliency map

$\mathcal{S}$	The predicted saliency map
$\mathcal{S}'$	The normalized saliency map according to normalized scanpath saliency
$\mathcal{S}_s$	The static/spatial saliency map
$\mathcal{S}_t$	The motion/temporal saliency map
$\mathcal{S}'_t$	The temporal saliency map after global motion compensation
$\mathcal{S}_s^{-1}$	The static saliency map of the previous non-predicted frame
$\mathcal{S}_t^{-1}$	The motion saliency map of the previous predicted frame
$\mathcal{S}_{MVE}$	The saliency map of motion vector entropy
$\mathcal{S}_{SRN}$	The saliency map of smoothed residual norm
$\mathbf{M}_k$	The motion saliency map at level k
$\mathbf{E}_m$	The normalized motion magnitude
$\mathbf{E}_g$	The global angle coherence
$\mathbf{E}_s$	The spatial angle coherence
$\mathbf{E}_t$	The temporal angle coherence
$\mathbf{E}_f$	The spatial frequency content
$\mathbf{E}_e$	The edge energy
$\mathbf{X}$	The DCT coefficients
$\mathbf{X}_\mathbf{n}$	The DCT coefficient of the block $\mathbf{n}$
$\mathbf{X}_\mathbf{r}$	The DCT coefficient of the residual block $\mathbf{r}$
$\mathbf{X}_S$	The DCT coefficients of the desired signal
$\mathbf{X}_N$	The DCT coefficients of the undesired signal
$\mathbf{S}_S$	The power spectral densities of the desired signal
$\mathbf{S}_N$	The power spectral densities of the undesired signal
(x, y)	A pixel coordinate
(x, y, t)	A pixel coordinate of frame t
$\mathbf{n}$	A block of pixels in an image
$\mathbf{m}$	A block of pixels in an image
$\mathbf{p}$	A block of pixels in an image
$\mathbf{r}$	A residual block in an image
$\mathbf{v}(\mathbf{n})$	The motion vector assigned to block $\mathbf{n}$
(v_x, v_y)	The motion vector assigned to a block at coordinate (x, y)
$\mathbf{v}'(\mathbf{n})$	The preprocessed motion vector assigned to block $\mathbf{n}$

$\widehat{\mathbf{v}}$	The representative vector of a region's motion
V	A list of motion vectors
$\widehat{V}$	A list of motion vectors containing a subset of V
o^t	The observed information of frame t
κ^t	The block coding mode and partition size of frame t
$\mathcal{T}$	The transfer function
Υ_θ	The shifted image by one pixel orthogonal to the Gabor orientation of θ
D	The difference of Gaussians map
O_θ	The Gabor orientation of an image with the orientation of θ
N_G	The 2-D Gaussian density map
P	The probability distribution
Q	The probability distribution
b	The Bernoulli distribution
H	The histogram
H	The entropy
$\mathcal{L}$	The low-pass filter
$corr$	The Pearson correlation coefficient
cov	The covariance function
$\inf$	The infimum function
$\sup$	The supremum function
F	the set of pixel coordinates of fixations
E	The energy function
ξ	The block-wise energy/error function
Z	The partition function
w	The weighting function
m_i	A global motion parameter (m_1,m_4:translation, m_2,m_6:zoom, m_3,m_5:rotation)
ρ	The strength of camera motion
θ	The orientation/angle
ϑ	The resolution level (scale) of an image
Λ_k	The motion vectors of 8-connected neighbors at level k
$\mathbf{z}$	The quantized transformed residual of a macroblock
dc	The DC value of a block

ω^t	The class labels of frame t
ω_*^t	The optimal label assignment for frame t
Ω^t	The set of all possible label assignments for frame t
ψ	A sample label assignment
Φ	The measure of saliency in the neighborhood
W	The spatial dimension
L	The temporal dimension
μ	The sample mean
$\overline{\mu}$	The weighted sample mean
$\widehat{m}$	The maximum value
σ	The Gaussian parameter or sample standard deviation
$\overline{\sigma}$	The weighted standard deviation
$\gamma(\cdot,\cdot)$	The degree of overlap between two class labels
$c(\cdot,\cdot)$	The consistency between the two locations
$d(\cdot,\cdot)$	The distance function between two locations
$d(\cdot)$	The length of a vector
$d'(\cdot)$	The normalized length of a vector
$\Delta_f(\cdot,\cdot)$	The absolute difference between the feature values of two blocks
$\delta(\cdot)$	The dissimilarity between a motion vector and its neighboring motion vectors
$N(\cdot)$	The neighborhood of a given block
$N^+(\cdot)$	The first-order neighborhood of a given block
$N^\times(\cdot)$	The second-order neighborhood of a given block
$\mathcal{N}(\cdot)$	The normalization operator
ℓ_p	The p-norm
τ	The threshold
$\mathbf{0}$	The matrix of 0s
$\mathbf{1}$	The matrix of 1s
T	Transpose
$\propto$	The proportional relationship
$\forall$	The for all operator
$\overset{2}{\Downarrow}$	The downsampling operator by a factor of two

$\overset{2}{\Uparrow}$	The upsampling operator by a factor of two
$\oplus$	The center-surround combination operator
$\ominus$	The center-surround difference operator
$\odot$	The pointwise multiplication operator
$*$	The convolution operator
$\angle$	The polar angle
$\|\| \cdot \|\|_2$	The Euclidean distance
$\|\| \cdot \|\|_2^s$	The Euclidean distance along the spatial dimension
$\|\| \cdot \|\|_2^t$	The Euclidean distance along the temporal dimension
$\| \cdot \|$	Absolute value operator or cardinality
$\|\cdot\|_{\geq 0}$	The operator that sets the negative values to zero

List of Acronyms

abb	*advert-bbc4-bees*
abl	*advert-bbc4-library*
ai	*advert-iphone*
aic	*ami-ib4010-closeup*
ail	*ami-ib4010-left*
blicb	*bbc-life-in-cold-blood*
bws	*bbc-wildlife-serpent*
ds	*diy-sos*
hp6t	*harry-potter-6-trailer*
mg	*music-gummybear*
mtnin	*music-trailer-nine-inch-nails*
nim	*nightlife-in-mozambique*
ntbr	*news-tony-blair-resignation*
os	*one-show*
pas	*pingpong-angle-shot*
pnb	*pingpong-no-bodies*
ss	*sport-scramblers*
swff	*sport-wimbledon-federer-final*
tucf	*tv-uni-challenge-final*
ufci	*university-forum-construction-ionic*
2-D	Two-Dimensional
3-D	Three-Dimensional
APPROX	Approximation to IKN
ASP	Advanced Simple Profile
AUC	Area Under Curve

AUC$'$	The center-bias-corrected Area Under Curve
AVC	Advanced Video Coding
AWS	Adaptive Whitening Saliency
B-frame	Bi-Predicted frame
BCM	Block Coding Mode
CIF	Common Intermediate Format
CRCNS	Collaborative Research in Computational Neuroscience
DCT	Discrete Cosine Transform
DCT-P	Discrete Cosine Transformation of Pixel blocks
DCT-R	Discrete Cosine Transformation of Residual blocks
DIEM	Dynamic Images and Eye Movements
DIOFM	DKL-color, Intensity, Orientation, Flicker, and Motion
DoG	Difference of Gaussians
EER	Equalized Error Rate
FN	False Negative
FOA	Focus of Attention
FP	False Positive
FPR	False Positive Rate
GAUS-CS	Gaussian Center-Surround
GAUSS	Gaussian center-bias
GBVS	Graph-Based Visual Saliency
GM	Global Motion
GMC	Global Motion Compensation
GME	Global Motion Estimation
GOP	Group-of-Pictures
HEVC	High Efficiency Video Coding
HVS	Human Visual System
ICM	Iterated Conditional Modes
IKN	Itti-Koch-Niebur
I-frame	Intra-coded frame
IO	Intra-Observer
JD	J-Divergence

JSD	Jensen-Shannon Divergence
JSD′	The center-bias-corrected Jensen-Shannon Divergence
KLD	Kullback-Leibler Divergence
MAM	Motion Attention Model
MAP	Maximum a Posteriori
MB	Macroblock
MCSDM	Motion Center-Surround Difference Model
MPEG	Moving Picture Experts Group
MRF	Markov Random Field
MSM-SM	Motion Saliency Map - Similarity Map
MV	Motion Vector
MVF	Motion Vector Field
MVE	Motion Vector Entropy
NSS	Normalized Scanpath Saliency
NSS′	The center-bias-corrected Normalized Scanpath Saliency
OBDL	Operational Block Description Length
P-frame	Predicted frame
PCC	Pearson Correlation Coefficient
PIM-MCS	Perceptual Importance Map based on Motion Center-Surround
PIM-ZEN	Perceptual Importance Map based on Zen method
PMES	Perceived Motion Energy Spectrum
PNSP-CS	Parametrized Normalization, Sum and Product - Center-Surround
PSNR	Peak Signal-to-Noise Ratio
PVM	Polar Vector Median
QCIF	Quarter Common Intermediate Format
QP	Quantization Parameter
RN	Residual Norm
ROC	Receiver Operating Characteristic
SEM	Standard Error of the Mean
SFC	Spatial Frequency Content
SFU	Simon Fraser University
SIF	Source Input Format

SORM	Self-Ordinal Resemblance Measure
STSD	Space Time Saliency Detection
SP	Simple Profile
SR	Stochastic Relaxation
SRN	Smoothed Residual Norm
ST-MRF	Spatio-Temporal Markov Random Field
TP	True Positive
TPR	True Positive Rate

Chapter 1

Introduction

1.1 Background and Motivation

Visual attention in humans is a set of strategies in early stages of vision processing that filters the stream of data collected by eyes. Visual attention enables the visual system to parse complex and highly cluttered scenes rapidly. The Human Visual System (HVS) reduces the complexity of visual scene analysis by automatically shifting the Focus of Attention (FOA) across the scene [128]. This ability allows the brain to restrict high-level processing of a scene to a relatively small part at any given time. Regions that draw attention are called *salient* and are subject to further processing for high-level perception of the scene.

In humans, attention mechanisms are driven by two components: 1) observer biases (so-called top-down attention) that enable high-level perception and 2) visual stimuli (so-called bottom-up attention) that are the characteristics of the scene itself. Top-down attention is a cognitive mechanism for maintaining goal-directed behavior. Bottom-up attention, on the other hand, deals with low-level features of the scene, such as contrast between various regions and their surroundings.

Recently, the development of visual attention models has attracted much interest in the computer vision and image processing communities. Although there has been a couple of attempts to model the cognitive influence in the HVS (top-down attention), most of the efforts have been devoted to model the stimulus-driven component (bottom-up attention), typically through the development of visual saliency models. This has long been believed to be a part of the early stages of vision, via the projection of the visual stimulus along the features computed in the early visual cortex, and to consist of a center-surround operation. In general, regions of the field of view that are distinctive compared to their surroundings attract attention [21]. A potentially more accurate approach to model visual attention could be to combine bottom-up saliency and high-level priors, or even broader perception [127, 107, 76, 24, 69, 47, 141, 61].

Early approaches to computational attention modeled bottom-up saliency using center-surround difference operator [73]. Under these deterministic models, if a given region resembles its surround, then the stimulus would be suppressive, resulting in low saliency, and if it differs from its surround, the stimulus would be excitatory, leading to high saliency. Variations on the details of the center-surround computation have given rise to a multitude of saliency models in the past decade and a half.

Another class of saliency models attempts to describe the principles of bottom-up attention in probabilistic terms [15, 16, 17, 145]. This approach is typically inspired by the cognitive science view where the brain is considered as a probabilistic network [89]. It is widely known in the cognitive science literature that the human brain operates as a universal compression device [16], where each layer eliminates as much signal redundancy as possible from its input, while preserving all the information necessary for scene perception. This principle has led to important developments in signal processing and computer vision techniques, such as wavelet theory [113], sparse representations [130], and, more recently, compression-based models of saliency.

The models that rely on probabilistic reasoning can be divided into two groups:

1. Stimulus-based information maximization

2. Signal compressibility

In the first group, saliency is hypothesized to be stimulus self-information [24, 164, 129]. More specifically, visual attention is governed by information maximization, where each region is assigned self-information [30] with respect to the distribution of certain features in the neighborhood. If a stimulus has low probability according to the feature distribution in the surround, this leads to high self-information and subsequently high saliency, whereas if the stimulus has high probability, self-information is low, leading to low saliency. A similar idea has been cast in a Bayesian framework in [69], where saliency is related to the divergence between the prior feature distribution in the surround, and the posterior distribution computed after observing the features in the center, termed *Bayesian surprise*.

In the second group of probabilistic models, saliency is equated to the reconstruction error of a compressed representation of the stimulus. Particularly, at each location, the stimulus is compressed, for example via principal component analysis [64, 114, 48], wavelet [140], or sparse decomposition [65, 98], and the reconstruction error from this compressed representation is measured. Large reconstruction error indicates incompressibility, which is considered as high saliency. On the other hand, easily compressible regions are considered to have low saliency in this framework. A recent comparative study [22] has shown that saliency models based on the compression principle tend to have excellent accuracy for the prediction of eye fixations. In fact, several of these models predict saliency with accuracy near the probability of agreement among observers. It could thus be claimed that "the bottom-up saliency modeling problem is solved."

There are, nevertheless, three main problems with the current state-of-the-art on visual saliency modeling:

- While it is true that high accuracy has been extensively documented for free viewing of still images, the same is not true for dynamic stimuli, which has received much less attention.

- While many implementations of the compression principle for saliency modeling have been proposed, none has really used a direct measure of compressibility. From a scientific point of view, this weakens the arguments in support of the principle.

- While many implementations of the "saliency as compression" principle have been proposed, much less attention has been devoted to implementation complexity.

The last item above is of critical importance for many real-world applications. For example, consider automatic monitoring of video quality in intermediate network nodes, where the uncompressed video is generally not accessible. According to physiological and psychological evidence, the impact of distortion in salient and non-salient areas is not equally important in terms of perceived quality. For large-scale in-network deployment, video quality monitoring that considers saliency must be of reasonably low complexity and memory requirements. As another example, for anomaly detection [110] or background subtraction [146] in large camera networks, saliency estimation should ideally be performed in the cameras themselves, then the system would only consume the power and bandwidth necessary to transmit video when faced with salient or anomalous events. This, however, requires highly efficient saliency algorithms.

These observations have motivated us to investigate alternative measures of saliency, according to compressed-domain features using the data already computed by the video encoder. In this dissertation, we propose two approaches to extract compressed-domain features that are highly indicative of saliency in natural video. In the first approach, two video features are extracted from the compressed video bitstream using motion vectors (MVs), block coding modes (BCMs), and transformed prediction residuals. In the second approach, the central idea is that there is no need to define new indirect measures of saliency, since a direct measure of compressibility, namely the number of bits, is readily available in the compressed bitstream. In fact, due to the extensive amount of research on video compression over the last decades, modern video compression systems are improving. It follows that the number of bits produced by a modern video codec is a fairly good measure of compressibility of the video being processed. Because modern codecs work very hard to assign bits efficiently to different locations of the visual field, the spatial distribution of bits can be seen as a saliency measure, which directly implements the compressibility principle. Under this view, regions that require more bits to compress are more salient, while regions that require fewer bits are less salient.

1.2 Preview and Contributions

This research is aimed at compressed-domain video processing. The goal is to reduce computational requirements of two important computer vision tasks - visual saliency modeling and region tracking - by reusing, as much as possible, the data already produced by the video encoder. As will be seen, however, the focus on compressed-domain information does not only improve algorithmic efficiency. In the case of saliency modeling, which forms the larger part of the dissertation, it also leads to higher accuracy. In the following, we give a preview of the various chapters in the dissertation and summarize the main contributions.

1.2.1 Bottom-Up Saliency Estimation

Many computational models have been introduced during the past 25 years to estimate visual saliency. An excellent review of the state of the art on pixel-domain saliency estimation is given in [21, 22]. To introduce the main concepts, we briefly review a gold-standard saliency model, the so-called Itti-Koch-Niebur (IKN) model in Chapter 2. In this model, the visual saliency is estimated using the center-surround difference mechanism implemented as the difference between a fine and a coarse resolution for a given feature.

1.2.2 Saliency Model Evaluation Framework

Eye-tracking data is the most typical psychophysical ground truth for both bottom-up and top-down visual attention models [39]. In this dissertation, we compare models' performance using sequences from two popular datasets, SFU [56] and DIEM [2]. In addition, to illuminate various aspects of the models' performance, a number of different comparison metrics are used. The advantages and limitations of the existing evaluation metrics are discussed, and, accordingly, new metrics are introduced that overcome the existing metrics' shortfalls. The details of the datasets and the evaluation metrics used in this research are described in [85, 82], and are discussed in Chapter 3.

1.2.3 A Comparison of Compressed-Domain Saliency Models

The overwhelming majority of existing saliency models operate on raw pixels, rather than compressed images or video. However, a few attempts have been made to use compressed video data, such as MVs, BCMs, motion-compensated prediction residuals, or their transform coefficients, in saliency modeling. The compressed-domain approach is typically adopted for efficiency reasons, i.e., to avoid recomputing information already present in the compressed bitstream. The extracted data is a proxy for many of the features frequently used in saliency modeling. For example, the motion vector field (MVF) is an approximation to optical flow, while BCMs and prediction residuals are indicative of motion complexity. Furthermore, the extraction of these features only requires partial decoding of the compressed video file, while the recovery of the actual pixel values is not necessary. A comparative study of available compressed-domain saliency models is presented in [82] and discussed in Chapter 4.

1.2.4 Proposed Saliency Estimation Methods

In [83, 84], we described two new video features for saliency modeling, namely Motion Vector Entropy (MVE) and Smoothed Residual Norm (SRN), both of which can be computed from the compressed video bitstream using MVs, BCMs, and transformed prediction residuals with partial decoding. The variation of motion and the split size of blocks are used to generate the MVE feature map, while the energy of prediction residuals is used to construct the SRN feature map.

We also proposed a simple compressed-domain video feature called the Operational Block Description Length (OBDL) as a measure of saliency in [86]. The OBDL is the number of bits required to compress a given block of video data under a distortion criterion. This saliency measure addresses the three main limitations of the state of the art. First, it is a direct measure of stimulus compressibility, namely "how many bits it takes to compress." By leveraging decades of research on video compression, this is a far more accurate measure of compressibility than previous proposals, such as Bayesian surprise or mutual information. Second, it is equally easy to apply to images and video. For example, it does not require weighting the contributions of spatial and temporal components, as the video encoder already uses motion estimation and compensation, and performs rate-distortion optimized bit assignment. Finally, because most modern cameras already contain an on-chip video compressor, it has trivial complexity for most computer vision applications. In fact, it only requires the very first step of decoding of the compressed bitstream to determine the number of bits assigned to each region. In Chapter 5, we will show that the three above-mentioned compressed-domain features are powerful enough to discriminate fixation points from non-fixation points in natural video.

A simple and effective saliency estimation method for compressed video can be constructed using the proposed features. In [83], we described a method called MVE+SRN to fuse MVE and SRN feature maps into the final saliency map. We have also proposed an implementation of the OBDL measure in [86], and showed that saliency can be estimated using a simple feature derived from it. However, while video compression systems produce very effective measures of compressibility, this measure is strictly local, since all processing is restricted to image blocks. Saliency, on the other hand, has both a local and global character, e.g. saliency maps are usually smooth. To account for this property we embed the OBDL features in a Markov Random Field (MRF) model. Our extensive experiments show that the resulting MVE+SRN and OBDL-MRF saliency measures make accurate predictions of eye fixations in dynamics scenes. Both methods will be described in Chapter 6.

1.2.5 Compressed-Domain Tracking

In Chapter 7 we describe a method for compressed-domain region tracking, which was first presented in [81]. While the material in this chapter can stand on its own and is applicable to various problems outside of saliency modeling, it opens up interesting possibilities in conjunction with compressed-domain saliency estimation, such as salient region tracking. The proposed tracking framework makes use of only the MVs and BCMs from the H.264/AVC-compressed video bitstream. This method tracks a single object by computing the *maximum a posteriori* (MAP) estimate of a Spatio-Temporal Markov Random Field (ST-MRF) at each P- or B-frame. In Chapter 7, the details of the framework are presented, and the accuracy of the proposed method is evaluated through simulation.

1.3 Scholarly Publications

The research efforts during this Ph.D. study have resulted in the following 8 scholarly publications, comprising 3 journal papers (two accepted, one submitted) and 5 conference papers (all accepted). Some works that have been completed during this period are not presented in this dissertation, particularly the work on still visualization of object motion (conference paper 4). All work has been performed in a reproducible research manner [154]. MATLAB implementation of the proposed methods (including our implementation of several compressed-domain saliency models from the literature) and the evaluation data (ground truth data, as well as implementations of various evaluation metrics) used in this study have been made available online at http://www.sfu.ca/~ibajic/software.html.

1.3.1 Journal papers

1. S. H. Khatoonabadi, I. V. Bajić, and Y. Shan. "Compressed-domain visual saliency models: A comparative study,“ submitted to *IEEE Trans. Image Process.*, 2014.

2. S. H. Khatoonabadi, I. V. Bajić and Y. Shan, “Compressed-domain correlates of human fixations in dynamic scenes,” accepted for publication in *Multimedia Tools and Applications*, Special Issue on Perception Inspired Video Processing, 2015. (Invited)

3. S. H. Khatoonabadi and I. V. Bajić. “Video object tracking in the compressed domain using spatio-temporal Markov random fields,“ *IEEE Trans. Image Process.*, 22(1):300-313, 2013.

1.3.2 Conference papers

1. S. H. Khatoonabadi, N. Vasconcelos, I. V. Bajić, and Y. Shan. “How many bits does it take for a stimulus to be salient?,“ *In Proc. IEEE CVPR'15*, pages 5501-5510, 2015.

2. S. H. Khatoonabadi, I. V. Bajić, and Y. Shan. “Compressed-domain correlates of fixations in video,“ *In Proc. 1st Intl. Workshop on Perception Inspired Video Processing, PIVP'14*, pages 3-8, 2014.

3. S. H. Khatoonabadi, I. V. Bajić, and Y. Shan. “Comparison of visual saliency models for compressed video,“ *In Proc. IEEE ICIP'14*, pages 1081-1085, 2014.

4. S. H. Khatoonabadi and I. V. Bajić. “Still visualization of object motion in compressed video,“ *In Proc. IEEE ICME'13 Workshop: MMIX*, 2013.

5. S. H. Khatoonabadi and I. V. Bajić. “Compressed-domain global motion estimation based on the normalized direct linear transform algorithm,“ *In Proc. International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC'13)*, 2013.

Chapter 2

Bottom-Up Saliency Estimation

The Human Visual System (HVS) is able to automatically shift the Focus of Attention (FOA) to salient regions in the pre-attentive, early vision phase. This ability allows the brain to restrict high-level processing of a scene to a relatively small part at any given time. Many computational models have been introduced to imitate the HVS in order to predict human visual attention. Many models rely on physiological and psychophysical findings [73]. Visual saliency estimation can benefit a large number of applications in image processing and computer vision, such as quality assessment [158, 44, 102, 123, 93, 40, 106, 165, 138, 32], compression [66, 157, 50, 99, 52, 53], guiding visual attention [57, 115, 116], retargeting [41, 105], segmentation [121, 45], anomaly detection [110], background subtraction [146], object recognition [58], object tracking [112], video abstraction [75], error concealment [55], data hiding [78], and so on.

In this chapter, we review one representative pixel-domain saliency estimation method known as the Itti-Koch-Niebur (IKN) model. This is one of the most cited saliency models, regarded as a gold standard in the field. A particular version of this model has recently been found [117] to be the most accurate among several publicly available saliency models on a dataset from [56]. We refer readers to [21, 22] for a more comprehensive overview of pixel-domain saliency estimation.

2.1 The Itti-Koch-Niebur (IKN) Saliency Model

In [73], Itti *et al.* proposed an architecture for building a bottom-up saliency map of visual attention for static images. This biologically-plausible architecture was inspired by

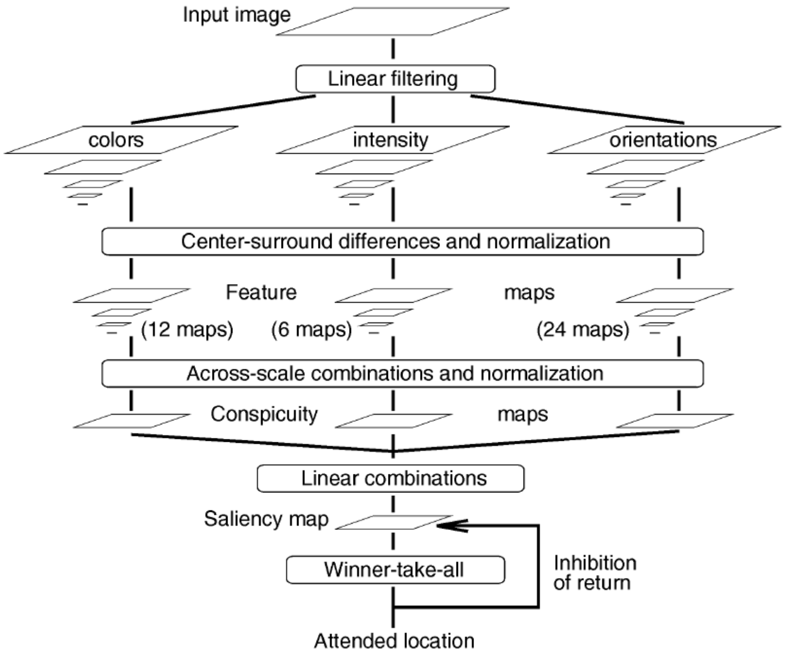


Figure 2.1: Architecture of the IKN model from [73].

the *Feature Integration Theory of Attention*, introduced by Treisman and Gelade [153]. This theory explains how HVS extracts important features and combines them to find the FOA. A set of topographic feature maps is first extracted from a scene. Next, salient spatial locations within each feature map are selected through competition. Multiple image features are then fused into a single topographical saliency map, known as the *Master Saliency Map*.

The competition for saliency was inspired by computational principles in the retina, lateral geniculate nucleus, and primary visual cortex [96], called *Center-Surround* strategy, in which the spatial location (the "center") that stands out relative to its neighborhood ("surround") is located. In the standard IKN model [73], center-surround difference was computed as the difference between fine and coarse resolutions (scales). The center was defined as a pixel at a fine resolution level, and surround was derived from the corresponding pixel at a coarser resolution level. A feature map was then determined by the difference between the maps at the two resolution levels, by interpolating the coarser resolution level to the finer resolution level and subtracting point by point. The architecture of the IKN model is depicted in Fig. 2.1 where the process is elaborated below.

2.1.1 Feature Map Extraction

Three groups of image features, namely intensity, color and orientation, are extracted in the IKN model. Associated with each image feature, a pyramid of the image is constructed at nine levels, by iteratively low-pass filtering and subsampling by a factor of two. The original image is at level "0" and the coarsest resolution image is at level "8." Center-surround feature maps are defined as the differences between a fine level (center) and a coarser level (surround).

Let $I(\vartheta)$ be the intensity of an image at the resolution level $\vartheta \in \{0, 1, ..., 8\}$ of the constructed Gaussian pyramid. A set of six feature maps is extracted corresponding to intensity channel of the image:

$$\mathcal{I}(c, s) = |I(c) \ominus I(s)|, \tag{2.1}$$

where operator $\ominus$ represents the center-surround difference, $c \in \{2, 3, 4\}$ and $s = c + \delta$ with $\delta \in \{3, 4\}$. The center-surround difference is obtained by interpolating the coarser resolution level (s) to the finer resolution level (c) through up-sampling followed by pointwise subtraction. An intensity feature map represents the sensitivity of neurons to intensity contrast (bright center surrounded by a dark neighborhood, or dark center surrounded by a bright neighborhood) that conforms to the functionality of the early HVS [96].

Similar to intensity feature maps, a set of color feature maps is also extracted. First, tuned color components of red ($\underline{R}$), green ($\underline{G}$), blue ($\underline{B}$), and yellow ($\underline{Y}$) are defined as

$$\underline{R} = \max\left(0, R - \frac{G + B}{2}\right), \tag{2.2}$$

$$\underline{G} = \max\left(0, G - \frac{R + B}{2}\right), \tag{2.3}$$

$$\underline{B} = \max\left(0, B - \frac{G + R}{2}\right), \tag{2.4}$$

$$\underline{Y} = \max\left(0, \frac{R + G}{2} - \frac{|R - G|}{2} - B\right), \tag{2.5}$$

where R, G and B are the red, green, and blue components of the image, respectively.

The definition of color feature maps is according to *Color Double-Opponent* model [38], which states that neurons in the center of their receptive fields are activated by one color (e.g., yellow) and inhibited by its opponent color (e.g., blue), or vice versa. The pairs of such opponent colors in HVS are (red, green) and (blue, yellow). Consequently, two color components are extracted based on these opponent colors:

$$\mathcal{RG}\left(c,s\right)=\left|\left\{\underline{R}\left(c\right)-\underline{G}\left(c\right)\right\}\ominus\left\{\underline{G}\left(s\right)-\underline{R}\left(s\right)\right\}\right|, \tag{2.6}$$

$$\mathcal{BY}\left(c,s\right)=\left|\left\{\underline{B}\left(c\right)-\underline{Y}\left(c\right)\right\}\ominus\left\{\underline{Y}\left(s\right)-\underline{B}\left(s\right)\right\}\right|, \tag{2.7}$$

where c and s are defined as before. Given $c \in \{2,3,4\}$ and $s = c+\delta$ with $\delta \in \{3,4\}$, a set of six feature maps is extracted for each color feature. Accordingly, 12 maps are defined for color features.

A set of orientation feature maps is also specified in order to consider receptive field sensitivity profile of orientation-selective neurons in HVS [96]. These maps are created based on the local orientation contrast between the center and surround:

$$\mathcal{O}_{\theta}\left(c,s\right)=\left|O_{\theta}\left(c\right)\ominus O_{\theta}\left(s\right)\right|, \tag{2.8}$$

where $O_{\theta}\left(\vartheta\right)$ is the Gabor orientation, obtained by Gabor filter [34], at the resolution level $\vartheta \in \{0,1,...,8\}$ with the orientation of $\theta \in \{0°, 45°, 90°, 135°\}$. Considering the number of possible orientations and resolution levels, 24 maps are derived for orientation features. Together with 6 intensity feature maps and 12 color feature maps, a total of $6+12+24=42$ feature maps are extracted from the image.

2.1.2 The Master Saliency Map

The master saliency map, or briefly the saliency map, is used frequently in the literature for quantifying the saliency at every location within an image. It is a gray scale image in which the brighter pixels represent the more salient locations. In the IKN model, all 42 extracted feature maps are combined together to create the saliency map. In the following, we review two methods used in various versions of the IKN model for combining feature maps.

MaxNorm normalization

In the standard IKN model [73], all feature maps are first normalized such that the values of each map range from 0 (dark) to $\widehat{m}$ (bright). All local maxima, except the global

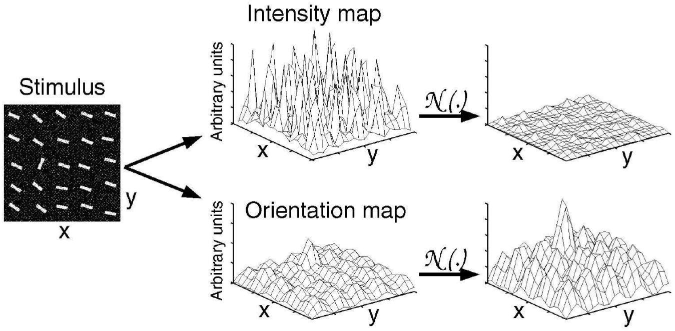


Figure 2.2: An example of MaxNorm normalization of a feature from [73].

maximum value $\widehat{m}$, are then identified within each map, and their sample mean, μ, is computed. Each map is individually normalized by multiplying its values by $(\widehat{m}-\mu)^2$ to intensify strong peaks that stand out from other peaks in the map, or to suppress the map globally if there is no distinctive peak compared to the average of local maxima. This normalization procedure is called MaxNorm normalization and is denoted by $\mathcal{N}(\cdot)$. The MaxNorm normalization attempts to account for the neuro-biological principle that neighboring similar features inhibit each other in HVS [25]. An example of MaxNorm normalization is illustrated in Fig. 2.2.

After MaxNorm normalization, all feature maps within each group of image features are combined, resulting in three separate "conspicuity" maps: $\overline{\mathcal{I}}$ for intensity, $\overline{\mathcal{C}}$ for color, and $\overline{\mathcal{O}}$ for orientation:

$$\overline{\mathcal{I}} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{4} \mathcal{N}\left(\mathcal{I}(c,s)\right), \tag{2.9}$$

$$\overline{\mathcal{C}} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{4} [\mathcal{N}\left(\mathcal{RG}(c,s)\right) + \mathcal{N}\left(\mathcal{BY}(c,s)\right)], \tag{2.10}$$

$$\overline{\mathcal{O}} = \sum_{\theta=\{0,45,90,135\}} \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{4} \mathcal{N}\left(\mathcal{O}_{\theta}\left(c,s\right)\right). \tag{2.11}$$

Combining feature maps is accomplished by subsampling to the resolution level four (the coarsest resolution level of centers) and pointwise summation, denoted by $\oplus$. Eventually, all conspicuity maps are normalized once more by the MaxNorm operation and summed to form the master saliency map:

$$\mathcal{S} = \frac{\sum_{\overline{\mathcal{K}}\in\{\overline{\mathcal{I}},\overline{\mathcal{C}},\overline{\mathcal{O}}\}} \mathcal{N}\left(\overline{\mathcal{K}}\right)}{3}. \tag{2.12}$$

Biologically, features in individual conspicuity map compete for saliency, whereas features in different conspicuity maps support each other. Hence, all feature maps in the same class are first combined and normalized, and then resulting feature maps of different classes are integrated to create the master saliency map.

FancyNorm Normalization

Itti and Koch in [72] proposed an alternative biologically-plausible normalization. The method is called Iterative Localized Normalization, also known as FancyNorm in the literature, and has been shown to offer high accuracy in terms of gaze prediction [117]. MaxNorm relies on the global maximum, while FancyNorm is based on local computations, which is consistent with the local connectivity of cortical neurons [72]. For this reason, FancyNorm might be considered more biologically-plausible than MaxNorm.

In FancyNorm, all feature maps are first normalized to range $[0, 1]$. Then, each feature map is convolved with the difference of Gaussians (DoG) filter given by

$$D(x, y) = \frac{c_{exc}^2}{2\pi\sigma_{exc}^2} \cdot e^{-(x^2+y^2)/2\pi\sigma_{exc}^2} - \frac{c_{inh}^2}{2\pi\sigma_{inh}^2} \cdot e^{-(x^2+y^2)/2\pi\sigma_{inh}^2}, \tag{2.13}$$

where c_{exc}^2 and c_{inh}^2 represent, respectively, the impact of local excitation and inhibition from neighboring locations. Parameters σ_{exc} and σ_{inh} are, respectively, Gaussian parameters for excitation and inhibition.

A given feature map $\mathcal{K}$ can be iteratively normalized by the DoG filter as

$$\mathcal{K} = |\mathcal{K} + \mathcal{K} * D - C_{inh}|_{\geq 0}\,, \tag{2.14}$$

where operator $*$ denotes convolution, and operator $|\cdot|_{\geq 0}$ means setting negative values to zero. C_{inh} is a constant inhibitory term which discards non-salient regions of uniform textures.

Fig. 2.3 shows the result of applying FancyNorm procedure on two feature maps: one containing a strong activation peak surrounded by numerous weaker activation peaks, and the other containing numerous strong activation peaks. In the former, shown at the top, the stronger peak at iteration 0 becomes excessively dominant peak after a few iterations, while in the latter, no peak stands out after a number of iterations. If this function is applied in a single iteration, the resulting normalization is called FancyOne; if two iterations are used, the resulting normalization is called FancyTwo, and so on.

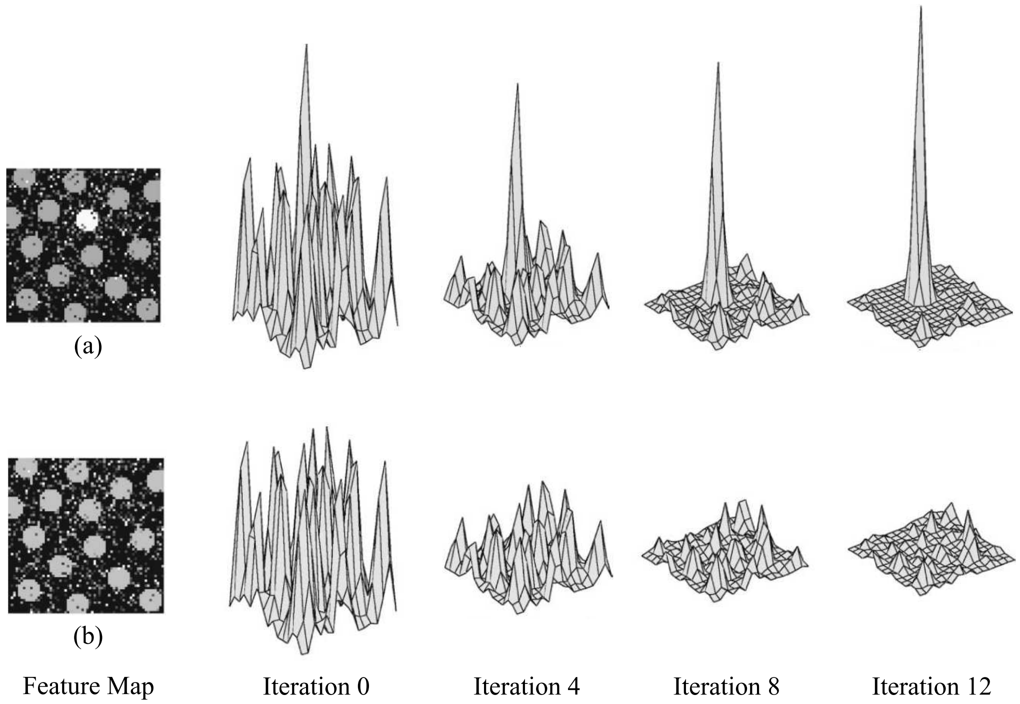


Figure 2.3: FancyNorm of two feature maps, from [72]: (a) contains a strong activation peak and several weaker activation peaks, (b) contains various strong activation peaks.

2.1.3 Focus of Attention

The predicted Focus of Attention (FOA) is the maximum of the master saliency map. To determine FOA jumps from one salient location to the next, for a given saliency map, the IKN model uses a biologically-plausible, 2-D winner-take-all neural network. In this neural network, each neuron is associated with a saliency map pixel, and is described by a capacitance in which the potential of more salient locations increases faster. The capacitance integrates excitatory inputs from the saliency map until one of them (the capacitance of the winner neuron) first reaches the voltage threshold and fires. At that time, the FOA is directed to the location of the winner, the capacitance of all neurons reset, and the area around the winner location (in the IKN model it is a disk with a radius determined by the size of the input image) is transiently deactivated (inhibited). The neurons again integrate the charge until the next FOA is identified and the process repeats. Fig. 2.4 shows an example.

The inhibition-of-return in the IKN model is such that the FOA is inhibited for approximately 500-900 ms, similar to HVS characteristics [133]. In addition, to correlate with

HVS, the voltage threshold of the capacitance should be chosen such that the time interval for jumping between two FOAs is approximately 30-70 ms [133].

2.2 Spatio-Temporal Saliency Estimation

In [71, 66], Itti *et al.* further extended the IKN model to address spatio-temporal saliency estimation. Two new groups of features, namely motion and flicker contrasts, were added to the basic IKN model [73]. Motion features were computed for different orientations, similar to features defined for orientation in static images. To do this, pyramid images of two successive frames were spatially-shifted orthogonal to the Gabor orientation and subtracted based on the Reichardt model [134]:

$$M_\theta^n(\vartheta) = \left| O_\theta^n(\vartheta) \odot \Upsilon_\theta^{n-1}(\vartheta) - O_\theta^{n-1}(\vartheta) \odot \Upsilon_\theta^n(\vartheta) \right|. \tag{2.15}$$

In the above equation, the symbol $\odot$ denotes pointwise multiplication. $M_\theta^n(\vartheta)$ is the motion feature of frame n at the resolution level $\vartheta \in \{0, 1, ..., 8\}$ and orientation $\theta \in \{0°, 45°, 90°, 135°\}$. $\Upsilon_\theta^n(\vartheta)$ is obtained by shifting the pyramid image one pixel orthogonal to the Gabor orientation $O_\theta^n(\vartheta)$. Note that one pixel shift at the coarsest level, $\vartheta = 8$, corresponds to $2^8 = 256$ pixels shift at the finest level, so a wide range of velocities can be handled.

The flicker for frame n was computed as the absolute difference between the intensity of the current frame and that of the previous frame:

$$F^n(\vartheta) = \left| I^n(\vartheta) - I^{n-1}(\vartheta) \right|. \tag{2.16}$$

Again, having six different pairs of (c, s) and four possible values for θ, 24 feature maps for motion and 6 feature maps for flicker were extracted as

$$\mathcal{M}_\theta^n(c, s) = \left| M_\theta^n(c) - M_\theta^n(s) \right|, \tag{2.17}$$

$$\mathcal{F}^n(c, s) = \left| F^n(c) - F^n(s) \right|. \tag{2.18}$$

From the extracted features, two new conspicuity maps were respectively defined for motion and flicker as

$$\overline{\mathcal{M}} = \sum_{\theta=\{0,45,90,135\}} \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{4} \mathcal{N}\left(\mathcal{M}_\theta\left(c, s\right)\right), \tag{2.19}$$

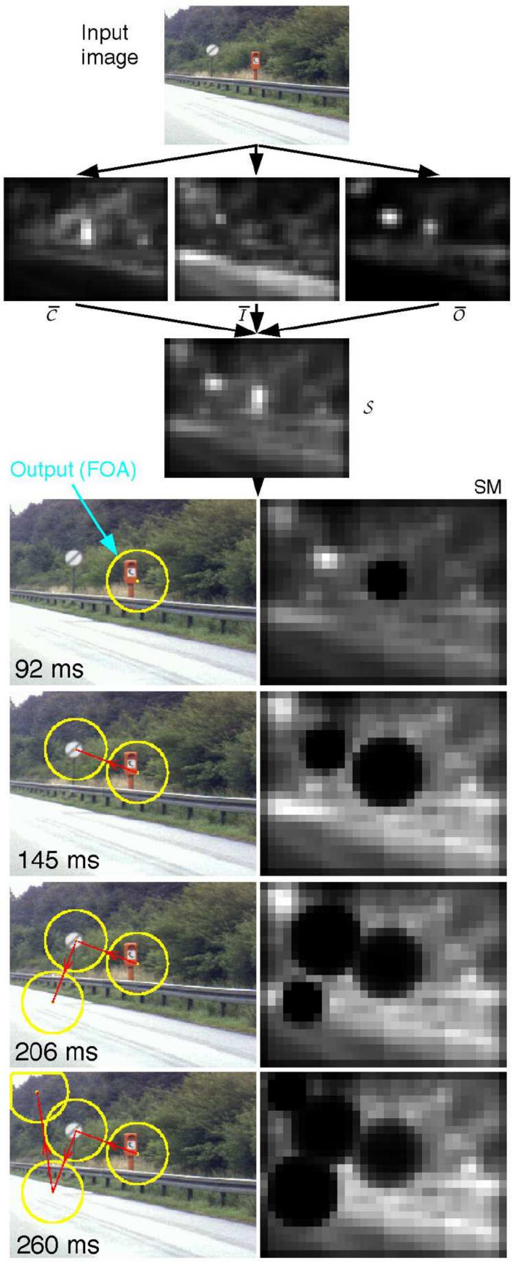


Figure 2.4: From [73]: the processing steps of the IKN model, demonstrated by an example. Three conspicuity maps of color ($\overline{\mathcal{C}}$), intensity ($\overline{\mathcal{I}}$) and orientation ($\overline{\mathcal{O}}$) are derived from image features and integrated to create the master saliency map ($\mathcal{S}$). The FOA is first directed to the most salient location, specified by an arrow in 92 ms simulated time, after which the next FOAs are successively determined considering inhibition-of-return feedback.

$$\overline{\mathcal{F}} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{4} \mathcal{N}\left(\mathcal{F}(c,s)\right). \tag{2.20}$$

Finally, the master saliency map in this model was computed as the summation of 5 conspicuity maps, containing 72 feature maps in total: 6 for intensity, 12 for color, 24 for orientation, 24 for motion and 6 for flicker.

$$\mathcal{S} = \frac{\sum_{\overline{\mathcal{K}} \in \{\overline{\mathcal{I}}, \overline{\mathcal{C}}, \overline{\mathcal{O}}, \overline{\mathcal{M}}, \overline{\mathcal{F}}\}} \mathcal{N}\left(\overline{\mathcal{K}}\right)}{5}. \tag{2.21}$$

Chapter 3

Saliency Model Evaluation Framework

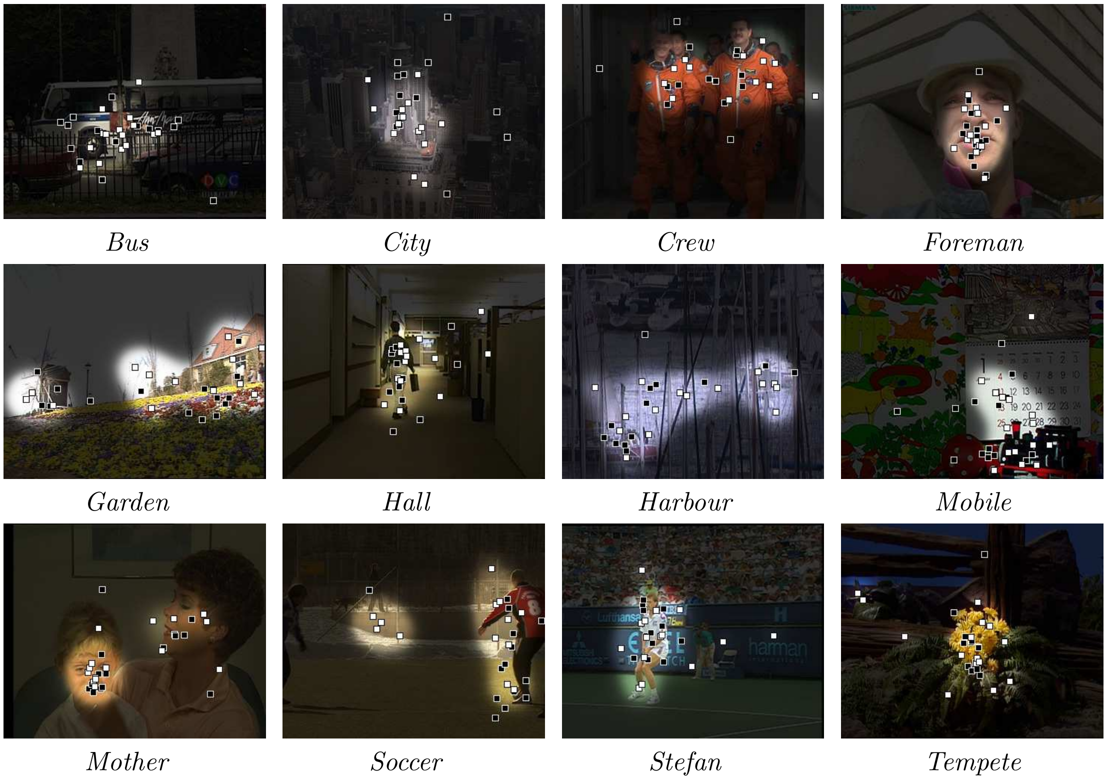
In order to quantify research progress on a particular problem, one needs to be able to compare various solutions in a common framework. In the case of visual saliency model comparison, we have to make decisions about ground-truth datasets, evaluation metrics, and the comparison methodology.

3.1 Eye-Tracking Video Datasets

Eye-tracking data is the most typical psychophysical ground truth for visual saliency models [39]. To evaluate saliency models, each model's saliency map is compared with recorded gaze locations of the subjects. Two recent publicly available eye-tracking datasets were used in this study. The reader is referred to [160] for an overview of other existing datasets in the field.

3.1.1 The SFU Dataset

The Simon Fraser University (SFU) eye-tracking dataset [56, 3] consists of twelve CIF (Common Intermediate Format, 352 × 288) sequences that have become popular in the video compression and communications community: *Bus*, *City*, *Crew*, *Foreman*, *Flower Garden*, *Hall Monitor*, *Harbour*, *Mobile Calendar*, *Mother and Daughter*, *Soccer*, *Stefan*, and *Tempete*. A total of 15 participants watched all 12 videos while wearing a Locarna Pt-mini head-mounted eye tracker [6]. Each participant took part in the test twice, resulting in two sets of viewings per participant for each video. The first viewing is used as ground truth for evaluating the performance of saliency models, whereas the data from the second


Bus *City* *Crew* *Foreman*

Garden *Hall* *Harbour* *Mobile*

Mother *Soccer* *Stefan* *Tempete*

Figure 3.1: Sample gaze visualization from the SFU Dataset. The gaze points from the first viewing are indicated as white squares, those from the second viewing as black squares.

viewing is used to construct benchmark models, as described in Section 3.1.3. The results in [56] showed that gaze locations in the first and second viewings can differ notably, however they remain relatively close to each other when there is a single dominant salient region in the scene (for example, the face in the *Foreman* sequence.) As a result, it is reasonable to expect that good saliency models will produce high scores for those frames where the first and second viewing data agree. A sample frame from each video is shown in Fig. 3.1, overlaid with the gaze locations from both viewings. The visualization is such that the less-attended regions (according to the first viewing) are indicated by darker colors. Further details about this dataset are shown in Table 3.1.

3.1.2 The DIEM Dataset

Dynamic Images and Eye Movements (DIEM) project [2] provides tools and data to study how people look at dynamic scenes. So far, DIEM collected gaze data for 85 sequences of 30 fps videos varying in the number of frames and resolution, using the SR Research Eyelink 1000 eye tracker [8]. The videos were taken from various categories including movie

Table 3.1: Datasets used in our study to evaluate different visual saliency models.

Dataset	**SFU**	**DIEM**
Year	2012	2011
Sequences	12	85
Display Resolution	704×576[*]	varying
Format	RAW	MPEG-4
Frame per Seconds	30	30
Frames	90-300	888-3401
Participants	15	35-53[§]
Viewings	2[†]	2[‡]
Screen Resolution	1280×1024	1600×1200
Screen Diagonal	19"	21.3"
Viewing Distance	80 cm	90 cm

[*] The original video resolution (352×288) was doubled during the presentation to the participants
[§] A total of 250 subjects participated in the study, but not all of them viewed each video; the number of viewers per video was 35-53 [†] Each participant watched each sequence twice, after several minutes
[‡] Viewings for the left/right eye are available

trailers, music videos, documentary, news and advertisements. For the purpose of the study, the frames of the sequences from the DIEM dataset were re-sized to 288 pixels height, while securing the original aspect ratio, resulting in five different resolutions: 352×288, 384×288, 512×288, 640×288 and 672×288. Among 85 available videos, 20 sequences similar to those used in [22] were chosen for the study, and only the first 300 frames were used in the comparison to match the length of the SFU sequences. In the DIEM dataset, the gaze location of both eyes are available. The gaze locations of the right eye were used as ground truth in the study, while gaze locations of the left eye were used to construct benchmark models, as described in Section 3.1.3. Clearly, the gaze points of the two eyes are very close to each other, closer than the gaze points of the first and second viewing in the SFU dataset. A sample frame form each selected sequence, overlaid with gaze locations of both eyes, is illustrated in Fig. 3.2. The visualization is such that the less-attended regions (according to the right eye) are indicated by darker colors.

3.1.3 Benchmark Models

In addition to the computational saliency models, we consider two additional models: Intra-Observer (IO) and Gaussian center-bias (GAUSS). The IO saliency map is obtained by the convolution of a 2-D Gaussian blob (with standard deviation of 1° of visual angle)

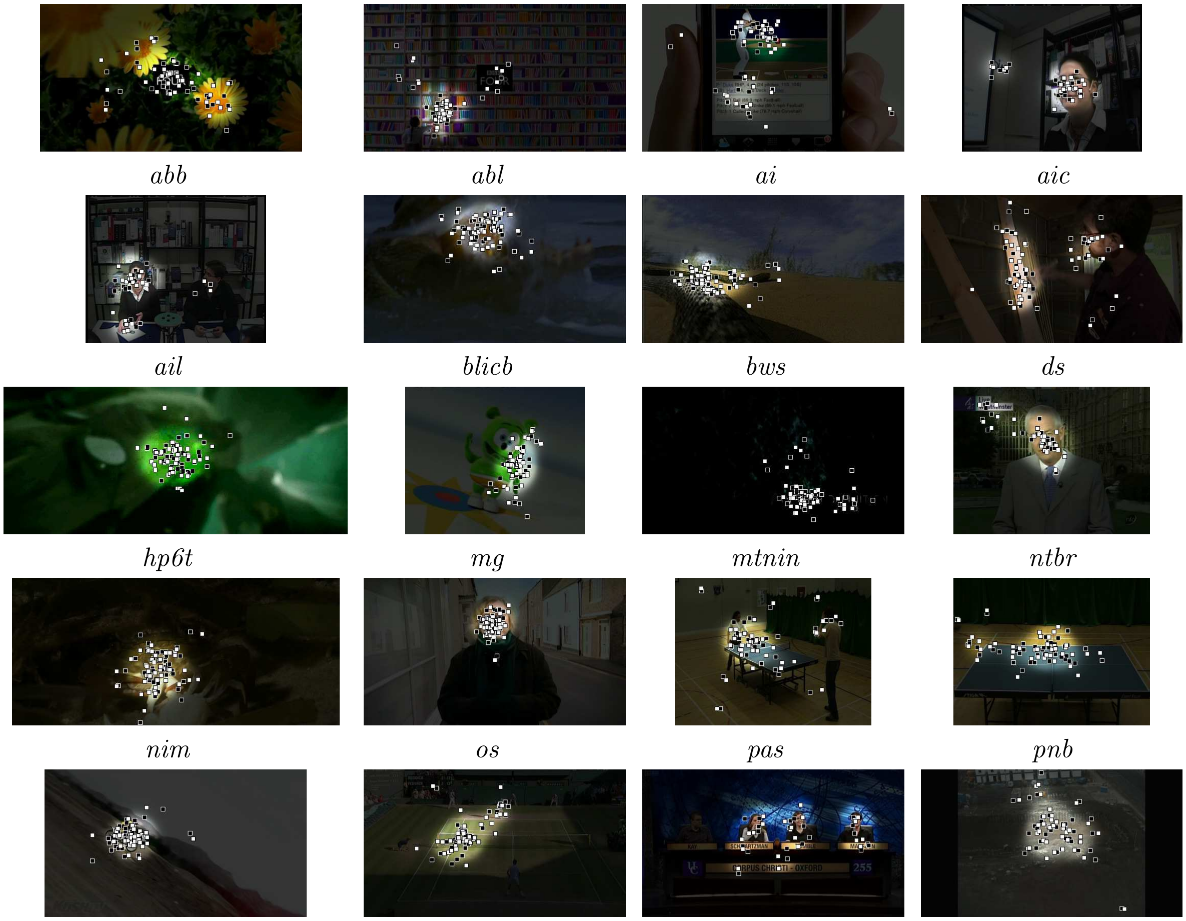


abb *abl* *ai* *aic*

ail *blicb* *bws* *ds*

hp6t *mg* *mtnin* *ntbr*

nim *os* *pas* *pnb*

ss *swff* *tucf* *ufci*

Figure 3.2: Sample gaze visualization from the DIEM Dataset. The gaze points of the right eye are shown as white squares, those of the left eye as black squares.

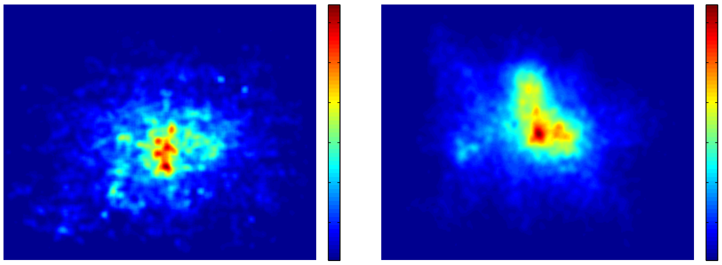

SFU DIEM

Figure 3.3: The heatmap visualization of gaze points combined across all frames and all observers, for the first viewing in the SFU dataset and the right eye in the DIEM dataset. Gaze points accumulate near the center of the frame.

with the second set of gaze points of the same observer within the dataset. Recall that both datasets have two sets of gaze points for each sequence and each observer – first/second viewing in the SFU dataset, right/left eye in the DIEM dataset. So the IO saliency maps for the sequences in the SFU dataset are obtained using the gaze points from the second viewing, while IO saliency maps for the sequences from the DIEM dataset are obtained using the gaze points of the left eye. These IO saliency maps can be considered as indicators of the best possible performance of a visual saliency model, especially in the DIEM dataset where the right and left eye gaze points are always close to each other.

On the other hand, GAUSS saliency map is just a 2-D Gaussian blob with the standard deviation of $1°$ located at the center of the frame. This model assumes that the center of the frame is the most salient point. Center bias turns out to be surprisingly powerful and has been used occasionally to boost the performance of saliency models without taking scene content into account. The underlying assumption is that the person recording the image or video will attempt to keep the salient objects at or near the center of the frame. Fig. 3.3 shows the heatmaps indicating cumulative gaze point locations across all sequences and all participants in the SFU dataset (first viewing) and DIEM dataset (right eye). As seen in the figure, aggregate gaze point locations do indeed cluster around the center of the frame. However, since GAUSS does not take content into account, one could expect a good saliency model to outperform it.

3.2 Accuracy Evaluation

A number of methods have been used to evaluate the accuracy of visual saliency models with respect to gaze point data [21, 22, 37, 67, 68, 95]. Since each method emphasizes a particular aspect of model's performance, to make the evaluation balanced, a collection of methods and metrics is employed in this study. A model that offers high score across many metrics can be considered to be accurate.

3.2.1 Area Under Curve (AUC)

The Area Under Curve (AUC) or, more precisely, the area under Receiver Operating Characteristic (ROC) curve, is computed from the graph of the true positive rate (TPR) versus the false positive rate (FPR) at various threshold parameters [148]. In the context of saliency maps, the saliency values are first divided into positive and negative sets corresponding to gaze and non-gaze points. Then for any given threshold, TPR and FPR are, respectively, obtained as the fraction of elements in the positive set and in the negative set that are greater than the threshold. Essentially, by varying the threshold, the ROC curve of TPR versus FPR is generated, visualizing the performance of a saliency model across all possible thresholds. The area under this curve quantifies the performance and shows how well the saliency map can predict gaze points. A larger AUC implies a greater correspondence between gaze locations and saliency predictions. A small AUC indicates weaker correspondence. The AUC is in the range $[0, 1]$: the values near 1 indicates the saliency algorithm performs well, the value of 0.5 represents pure chance performance, and the value of less than 0.5 represents worse than pure chance performance. This metric is also invariant to monotonic scaling of saliency maps [23].

It is worth mentioning that instead of using all non-gaze saliency values, these are usually sampled [139, 37]. The idea behind this approach is that an effective saliency model would have higher values at fixation points than at randomly sampled points. Control points for non-gaze saliency values are obtained with the help of a nonparametric bootstrap technique [36], and sampled with replacement, with sample size equal to the number of gaze points, from non-gaze parts of the frame, multiple times. Finally, the average of the statistic over all bootstrap subsamples is taken as a sample mean.

3.2.2 Kullback-Leibler Divergence (KLD) and J-Divergence (JD)

The Kullback-Leibler Divergence (KLD) is often used to obtain the divergence between two probability distributions. It is given by the relative entropy of one distribution with respect to another [92]

$$KLD(P\|Q) = \sum_{i=1}^{r} P(i) \cdot \log_b \left(\frac{P(i)}{Q(i)} \right), \tag{3.1}$$

where P and Q are discrete probability distributions, b is the logarithmic base, and r indicates the number of bins in each distribution. Note that KLD is asymmetric. The symmetric version of KLD, also called J-Divergence (JD), is [74]

$$JD(P\|Q) = KLD(P\|Q) + KLD(Q\|P). \tag{3.2}$$

To assess how accurately a saliency model predicts gaze locations based on JD, the distribution of saliency values at the gaze locations is compared against the distribution of saliency values at some random points from non-gaze locations [67, 69, 68]. If these two distributions overlap substantially, i.e., if JD approaches zero, then the saliency model predicts gaze points no better than a random guess. On the other hand, as one distribution diverges from the other and JD increases, the saliency model is better able to predict gaze points.

Specifically, let there be n gaze points in a frame. Another n points different from the gaze points are randomly selected from the frame. The saliency values at the gaze points and the randomly selected points constitute the two distributions, P and Q. A good saliency model would produce a large JD, because saliency values at gaze points would be large, while saliency values at non-gaze points would be small. The process of choosing random samples and computing the JD is usually repeated many times and the resulting JD values are averaged to minimize the effect of random variations. While JD has certain advantages over KLD (see [68, 21] for details), it also shares several problems. One of the problems with both KLD and JD is the lack of an upper bound [91]. Another problem is that if $P(i)$ or $Q(i)$ is zero for some i, one of the terms in (3.2) is undefined. For these reasons, KLD and JD were not used in the present study.

3.2.3 Jensen-Shannon Divergence (JSD)

The Jensen-Shannon divergence (JSD) is a KLD-based metric that avoids some of the problems faced by KLD and JD [100]. For two probability distributions P and Q, JSD is defined as [33]:

$$JSD(P\|Q) = \frac{KLD(P\|R) + KLD(Q\|R)}{2}, \tag{3.3}$$

where

$$R = \frac{P+Q}{2}. \tag{3.4}$$

Unlike KLD, JSD is a proper metric, is symmetric in P and Q, and is bounded in $[0,1]$ if the logarithmic base is set to $b = 2$ [100]. The value of the JSD for the saliency map that perfectly predicts gaze points will be equal to 1. The same sampling strategy employed in AUC and KLD/JD computation can also be used for computing JSD.

3.2.4 Normalized Scanpath Saliency (NSS)

The Normalized Scanpath Saliency (NSS) measures the strength of normalized saliency values at gaze locations [132]. Normalization is affine so that the resulting normalized saliency map has zero mean and unit standard deviation. The NSS is defined as the average of normalized saliency values at gaze points:

$$NSS = \frac{\sum_{(x,y)\in F} \mathcal{S}'(x,y)}{|F|}, \tag{3.5}$$

where F is the set of pixel coordinates of fixations, $|\cdot|$ is cardinality, and

$$\mathcal{S}'(x,y) = \frac{\mathcal{S}(x,y) - \mu}{\sigma}, \tag{3.6}$$

in which μ and σ are the mean and the standard deviation of the saliency map, respectively.

A positive normalized saliency value at a certain gaze point indicates that the gaze point matches one of the predicted salient regions, zero indicates no link between predictions and the gaze point, while a negative value indicates that the gaze point has fallen into an area predicted to be non-salient.

3.2.5 Pearson Correlation Coefficient (PCC)

The Pearson Correlation Coefficient (PCC) measures the strength of a linear relationship between the predicted saliency map $\mathcal{S}$ and the ground truth map $\mathcal{G}$. First, the ground truth map $\mathcal{G}$ is obtained by convolving the gaze point map with a 2-D Gaussian function having the standard deviation of 1° of the visual angle [95]. Then $\mathcal{S}$ and $\mathcal{G}$ are treated as random variables whose paired samples are given by values of the two maps at each pixel position in the frame. The Pearson correlation coefficient is defined as

$$corr(\mathcal{S}, \mathcal{G}) = \frac{cov(\mathcal{S}, \mathcal{G})}{\sigma_{\mathcal{S}}\sigma_{\mathcal{G}}}, \tag{3.7}$$

where $cov(\cdot, \cdot)$ denotes covariance and σ_S and σ_G are, respectively, the standard deviations of the predicted saliency map and the ground truth map. The value of PCC is between -1 and 1; the value of ± 1 indicates the strongest linear relationship, whereas the value of 0 indicates no correlation. If the model's saliency values tend to increase as the values in the ground truth map increase, the PCC is positive. Otherwise, if the model's saliency values tend to decrease as the ground truth values increase, the PCC is negative. In this context, a PCC value of -1 would mean that the model predicts non-salient regions as salient, and salient regions as non-salient. While this is the opposite of what is needed, such model can still be considered accurate if its saliency map is inverted. While PCC is widely used for studying relationships between random variables, in its default form it has some shortcomings in the context of saliency model evaluation, especially due to center bias, as discussed in the next section.

3.3 Data Analysis Considerations

Here, we discuss several considerations about the ground truth data, and the methods and metrics used in the evaluation.

3.3.1 Gaze Point Uncertainty

Eye-tracking datasets usually report a single point (x, y) as the gaze point of a given subject in a given frame. However, such data should not be treated as absolute. There are at least two sources of uncertainty in the measurement of gaze points. One is the eye-tracker's measurement error, which is usually on the order of 0.5° to 1° of the visual angle [6, 8, 122]. The other source of uncertainty is the involuntary eye movement during fixations. The human eye does not concentrate on a stationary point during a fixation, but instead constantly makes small rapid movements to make the image more clear [26].

Depending on the implementation, the eye tracker may filter those rapid movements out, either due to undersampling or to create an impression of a more stable fixation. For at least these two reasons, the gaze point measurement reported by an eye tracker contains some uncertainty. At the current state of technology, the eye tracker measurement errors seem to be larger than the uncertainty caused by involuntary drifts, and so we take them as the dominant source of noise in the ground truth data. To account for this noise, we apply a local maximum operator in a radius of 0.5° of visual angle. In other words, when computing a saliency value of a given point in a frame, the maximum value within its neighborhood is used.

3.3.2 Center Bias and Border Effects

A person recording a video will generally tend to put regions of interest near the center of the frame [151, 131]. In addition, people also have a tendency to look at the center of the image [150], presumably to maximize the coverage of the displayed image by their field of view. These phenomena are known as center bias. Fig. 3.3 illustrates the center bias in the SFU and DIEM datasets by displaying the locations of gaze points accumulated over all sequences and all frames.

Interestingly, Kanan *et al.* [76] and Borji *et al.* [22] showed that creating a saliency map merely by placing a Gaussian blob at the center of the frame may result in fairly high scores. Such high scores are partly caused by using a uniform spatial distribution over the image when selecting control samples. Specifically, the computation of ACU, KLD and JSD for a given model involves choosing non-gaze control points randomly in an image. If these are chosen according to a uniform distribution across the image, the process results in many control points near the border, which, empirically, have little chance of being salient. As a result, the saliency values of those control points tend to be small, resulting in an artificially high score for the model under test. At the same time, since gaze points are likely located near the center of the frame, a centered Gaussian blob would tend to match many of the gaze points, which would make its NSS and PCC scores high.

Additionally, Zhang *et al.* [164] thoroughly investigated the effect of dummy zero borders against evaluation metrics. Adding dummy zero saliency values at the border of the image changes the distribution of saliency of the random samples as well as the normalization parameters in NSS, leading to different scores while the saliency prediction is unchanged. To decrease sensitivity to the center bias and the border effect, Tatler *et al.* [151] and Parkhurst and Niebur [131] suggested to distribute random samples according to the measured gaze points. To this end, Tatler *et al.* [151] distributed random samples from human saccades and choose control points for the current image randomly from fixation points in other images in their dataset. Kanan *et al.* [76] also picked saliency values at the gaze points in the current image, while control samples were chosen randomly from the fixations in other images in the dataset. For both techniques, control points are drawn from a non-uniform random distribution according to the measured fixations, decreasing the effect of center bias. Furthermore, this way, dummy zero borders will not affect the distribution of random samples.

In this thesis, we use a similar approach for handling center bias and border effects. Instead of directly using the accumulated gaze points over all frames in the dataset (Fig. 3.3), we fit an omni-directional 2-D Gaussian distribution to the accumulated gaze points across both SFU and DIEM datasets. Then, control samples are chosen randomly from the fitted 2-D Gaussian distribution. This reduces center bias in AUC and JSD.

To reduce center bias and border effects in NSS, we modify the normalization as

$$\mathcal{S}'(x,y) = \frac{\mathcal{S}(x,y) - \overline{\mu}}{\overline{\sigma}}, \tag{3.8}$$

where

$$\overline{\mu} = \frac{1}{n} \sum_{(x,y)} N_G(x,y) \cdot \mathcal{S}(x,y), \tag{3.9}$$

$$\overline{\sigma} = \sqrt{\frac{1}{n-1} \sum_{(x,y)} \left(N_G(x,y) \cdot \mathcal{S}(x,y) - \overline{\mu}\right)^2}. \tag{3.10}$$

In the above equations, (x, y) are the pixel coordinates, n is the total number of pixels, and $N_G(x, y)$ is the fitted 2-D Gaussian density evaluated at (x, y) normalized such that it sums up to 1. In the normalization described by the above three equations, the pixels located near the center of the image are given more significance due to N_G. In other words, saliency predictions have the same bias as observers' fixations. These accuracy measures that are modified to reduce the center bias and border effects are indicated by prime ($'$) and referred to as NSS$'$, AUC$'$, and JSD$'$.

We summarize all above-mentioned metrics in Table 3.2. Metrics can be divided by symmetry (column 2) or boundedness (column 3). Some metrics favor center-biased saliency

Table 3.2: Summary of evaluation metrics used in the study.

Metric	**Symmetric**	**Bounded**	**Center-biased**	**Applicability**	**Input**
AUC	Yes	Yes	Yes	General	Location
AUC′	Yes	Yes	No	Saliency	Location
KLD	No	No	Yes	General	Distribution
JD	Yes	No	Yes	General	Distribution
JSD	Yes	Yes	Yes	General	Distribution
JSD′	Yes	Yes	No	Saliency	Distribution
NSS	Yes	No	Yes	Saliency	Value
NSS′	Yes	No	No	Saliency	Value
PCC	Yes	Yes	Yes	General	Distribution

models (column 4). Also, some metrics are specific to saliency while others have more general applicability (column 5), e.g., for comparing two distributions. The input data for various metrics comes from three sources (column 6): 1) the locations associated with estimated saliency 2) the distribution of estimated saliency and 3) the values of estimated saliency at fixation points.

Chapter 4

Compressed-Domain Saliency Estimation

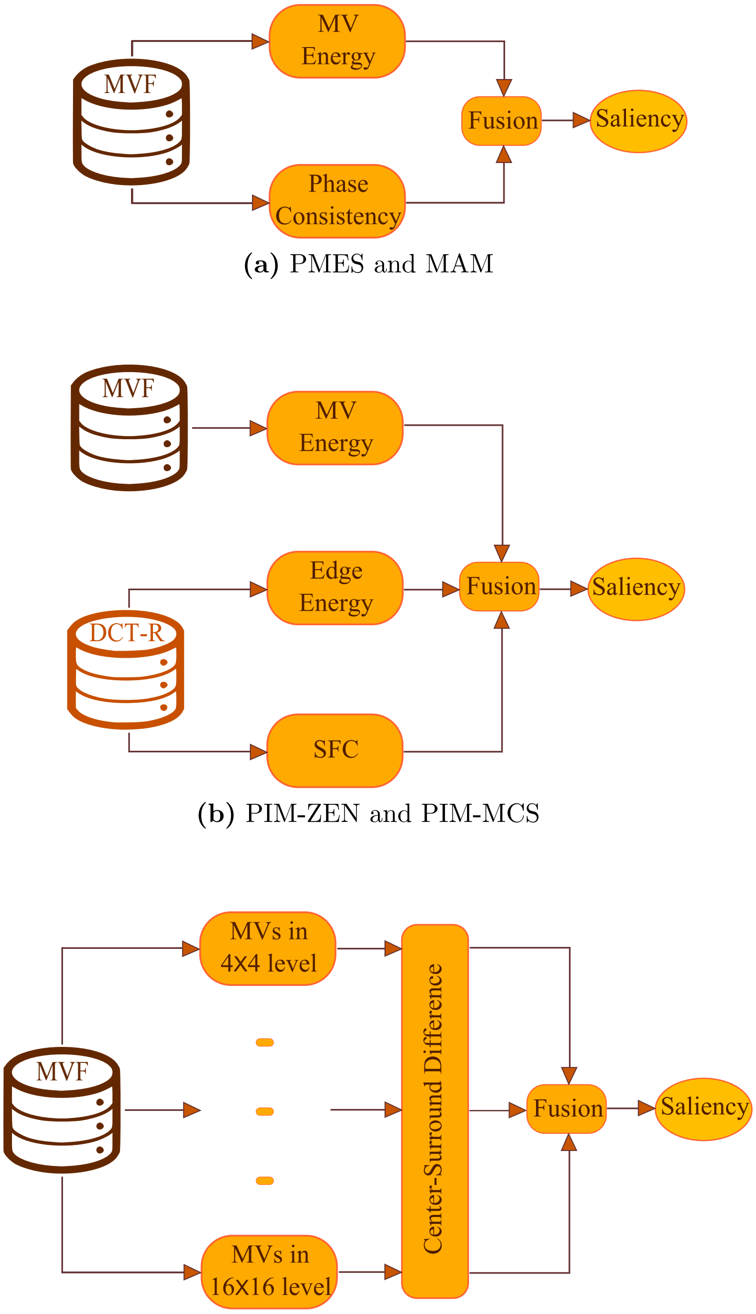
Image and video processing methods can be distinguished by the domain in which they operate: pixel domain vs. compressed domain. The former have the potential for higher accuracy, but also have higher computational complexity. In contrast to the pixel-domain methods, compressed-domain approaches make use of the data from the compressed video bitstream, such as motion vectors (MVs), block coding modes (BCMs), motion-compensated prediction residuals or their transform coefficients, etc. The lack of full pixel information often leads to lower accuracy, but the main advantage of compressed-domain methods in practical applications is their generally lower computational cost. This is due to the fact that part of decoding can be avoided, a smaller amount of data needs to be processed compared to pixel-domain methods, and some of the information produced during encoding (e.g., MVs and transform coefficients) can be reused. Therefore, compressed-domain methods are more suitable for real-time applications.

Due to its potential for lower complexity compared to pixel-domain methods, compressed-domain visual saliency estimation is starting to be used in a number of applications such as video compression [51], motion-based video retrieval [108], video skimming and summarization [109], video transcoding [161, 104, 143], quality estimation [101], image retargeting [41], salient motion detection [126], and so on. Some methods only use MVs [108, 109, 161, 104] while others also take advantage of other data produced during encoding, such as transform coefficients [11, 143, 35, 42]. Although there are relatively few compressed-domain saliency models compared to their pixel-domain counterparts, their potential for practical deployment makes them an important research topic.

In this chapter, our goal is to evaluate visual saliency models for video that have been designed explicitly for, or have the potential to work in, the compressed domain. This means that they should operate with the kind of information found in a compressed video bitstream, such as block-based motion vector field (MVF), BCMs, prediction residuals or their transforms, etc. Finally, the strategies that have shown success in compressed-domain saliency modeling are highlighted, and certain challenges are identified as potential avenues for further improvement. To our knowledge, this is the first comprehensive comparison of compressed-domain saliency models. Preliminary findings have been presented in [82] and a more complete study has been submitted for publication [85].

4.1 Compressed-Domain Visual Saliency Models

In the following, nine prominent models listed in Table 4.1, sorted according to the publication year, are reviewed. Different models assume different coding standards, for example MPEG-1, MPEG-2, MPEG-4 SP (Simple Profile), MPEG-4 ASP (Advanced Simple Profile), and MPEG-4 part 10, better known as H.264/AVC (Advanced Video Coding). For each model, the data used from the compressed bitstream, their intended application, as well as data and evaluation method, if any, are also included in the table. As seen in the table, only a few of the most recent models have been evaluated using gaze data from eye-tracking experiments, which is widely considered to be the ultimate test for a visual saliency model. This fact makes the comparison study all the more relevant. Fig. 4.1 illustrates functional diagrams of the various models in the study, which will complement their brief description below.

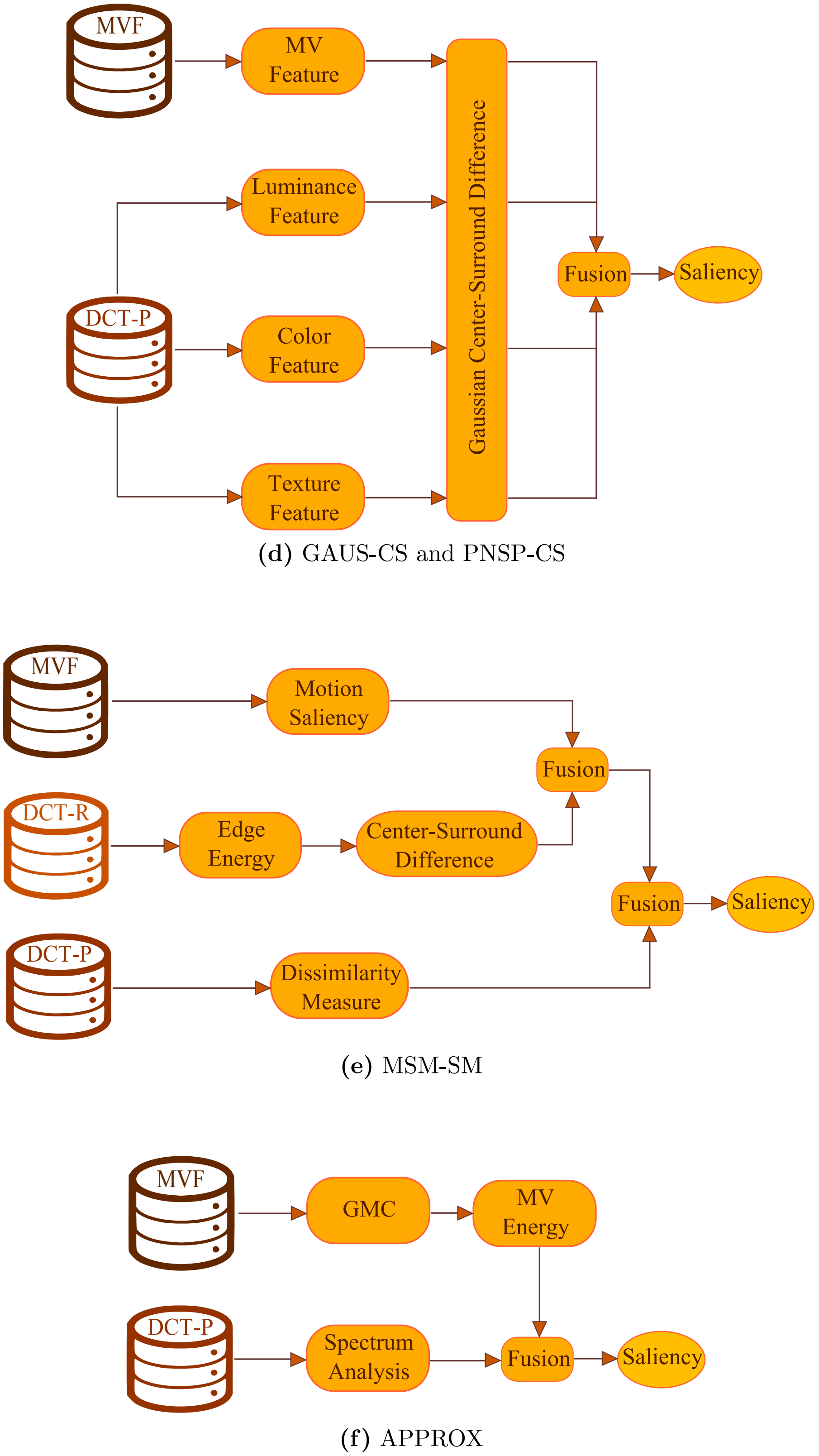


(a) PMES and MAM

(b) PIM-ZEN and PIM-MCS

(c) MCSDM

Figure 4.1: Functional diagrams for various compressed-domain saliency models.



(d) GAUS-CS and PNSP-CS

(e) MSM-SM

(f) APPROX

Figure 4.1: Functional diagrams for various compressed-domain saliency models. *(cont.)*

Table 4.1: Compressed-domain visual saliency models included in the study
(MVF: Motion Vector Field; DCT-R: Discrete Cosine Transformation of residual blocks; DCT-P: Discrete Cosine Transformation of pixel blocks; KLD: Kullback-Leibler Divergence; AUC: Area Under Curve; ROC: Receiver Operating Characteristic)

#	Model	First Author	Year	Codec	Data	Application	Sequences	Gaze data	Metric(s)
1	**PMES**	Ma [108]	2001	MPEG-1/2	MVF	Video Retrieval	MPEG-7 [125]	-	-
2	**MAM**	Ma [109]	2002	MPEG-1/2	MVF	Video Skimming	Specific [109]	-	Human Score
3	**PIM-ZEN**	Agarwal [11]	2003	MPEG-1/2	MVF+DCT-R	ROI Detection	QCIF Standard	-	-
4	**PIM-MCS**	Sinha [143]	2004	MPEG-4 SP	MVF+DCT-R	Video Transcoding	QCIF Various	-	-
5	**MCSDM**	Liu [104]	2009	H.264/AVC	MVF	Rate Control	QCIF Standard	-	-
6	**GAUS-CS**	Fang [42]	2012	MPEG-4 ASP	MVF+DCT-P	Saliency Detection	CRCNS [69],[70]	Yes	KLD
7	**MSM-SM**	Muthuswamy [126]	2013	MPEG-2	MVF+DCT-P/R	Saliency Detection	Mahadevan [111]	-	ROC
8	**APPROX**	Hadizadeh [51]	2013	-	MVF+DCT-P	Video Compression	SFU [56]	Yes	KLD+AUC
9	**PNSP-CS**	Fang [43]	2014	MPEG-4 ASP	MVF+DCT-P	Saliency Detection	CRCNS [69],[70]	Yes	KLD+AUC

4.1.1 Saliency Based on Perceived Motion Energy Spectrum

Ma and Zhang [108] used the magnitude of an object's motion and its duration as cues for detecting salient regions. According to this saliency model, the so-called Perceived Motion Energy Spectrum (PMES) model (Fig. 4.1-a), a motion with a large magnitude attracts human's attention. Moreover, the angle of camera motion in a shot is supposed to be more stable than the angle of a salient object's motion.

The saliency map in [108] was constructed by using MVs only:

$$\mathcal{S} = \mathbf{E}_m \odot \mathbf{E}_g, \tag{4.1}$$

where matrices $\mathbf{E}_m$ and $\mathbf{E}_g$ respectively denote the normalized motion magnitude and global angle coherence, and operator $\odot$ represents the element-wise multiplication. Matrix $\mathbf{E}_m$ was obtained from the average motion magnitude over a constant duration of a shot after removing outliers. Specifically, at each MV, a 3-D spatio-temporal tracking volume from which the motion magnitude is computed was considered. Inlier MVs were filtered by an α-trimmed average filter followed by normalization, that is

$$\mathbf{E}_m(\mathbf{n}) = \begin{cases} \frac{x}{\tau} & x \leq \tau \\ 1 & x > \tau, \end{cases} \tag{4.2}$$

where x is the filtered motion magnitude of block $\mathbf{n}$ and τ denotes the truncating threshold. So, if $\mathbf{E}_m$ of a MV approaches 1, it implies the corresponding block has a large motion magnitude, and therefore probably attracts human's attention.

The global angle coherence of a MV over its corresponding duration of the shot was computed from the normalized entropy:

$$\mathbf{E}_g(\mathbf{n}) = \frac{-\sum P(\theta) \cdot \log P(\theta)}{\log n}, \tag{4.3}$$

where $P(\theta)$ is the probability mass function (i.e., normalized histogram) of MV angles, and n is the number of histogram bins within the associated tracking volume. The denominator used here, $\log n$, is for normalizing the value of $\mathbf{E}_g(\mathbf{n})$ to the range $[0, 1]$. Note that when $P(\theta) = 1/n$ for all θ in the tracking volume associated with the block $\mathbf{n}$, $\mathbf{E}_g(\mathbf{n})$ will be 1, and when $\lim P(\theta) \to 1$ for a certain θ, $\mathbf{E}_g(\mathbf{n})$ approaches 0. Thus, if $\mathbf{E}_g(\mathbf{n})$ approaches 0, it implies the camera motion is dominant, and if it approaches 1, it it indicates the inconsistency of the motion that likely attracts human's attention.

The saliency map computed by (4.1) takes two rules into consideration: the sensitivity of Human Visual System (HVS) to a large magnitude motion and the stability of camera motion. It is unclear whether the assumptions made about HVS in this paper actually hold in practice. Furthermore, experimental evaluation was undertaken in the context of video retrieval, rather than gaze prediction.

4.1.2 Saliency Based on Motion Attention Model

In the Motion Attention Model (MAM) [109] (Fig. 4.1-a), developed by the same group as PMES in Section 4.1.1, two new rules are further added to reduce false detection: the MVs of a moving object tend to have coherent angles, and if the magnitudes of object's MVs are large and their angles are incoherent, the motion information is not reliable. Therefore, the saliency map in the MAM is computed as

$$\mathcal{S} = \mathbf{E}_m \odot \mathbf{E}_t \odot (\mathbf{1} - \mathbf{E}_m \odot \mathbf{E}_s), \tag{4.4}$$

where matrices $\mathbf{E}_m$, $\mathbf{E}_s$ and $\mathbf{E}_t$ are, respectively, the normalized motion magnitude, spatial angle coherence and temporal angle coherence, and $\mathbf{1}$ is a matrix of 1s. Note that operators $\odot$ and $-$ are element-wise operations. Spatial angle coherence was defined within a spatial window, while temporal angle coherence was obtained within a temporal window, both as normalized entropies, similar to (4.3).

The target application of [109] was video skimming. The proposed motion saliency model was evaluated using the following technique. Both the video and the obtained motion saliency model were shown to 10 viewers. Each saliency model was given a score from {0, 50, 100} to each shot. The viewers gave a score of 100 if they felt the saliency model best predicted the Foci of Attention (FOAs); they gave a score of 50 if they believed the estimated saliency model was not the best but satisfactory; otherwise they gave score of 0. No comparison with other saliency estimation methods was reported.

4.1.3 Saliency Based on Perceptual Importance Map

The Perceptual Importance Map [11] based on Zen method [162] (PIM-ZEN) (Fig. 4.1-b) computes the saliency map based on MVs and DCT values. In this model, the saliency map was computed as

$$\mathcal{S} = \mathbf{E}_m + \mathbf{E}_f + \mathbf{E}_e, \tag{4.5}$$

where matrix $\mathbf{E}_m$ is the normalized motion magnitude, matrix $\mathbf{E}_f$ represents the Spatial Frequency Content (SFC), and matrix $\mathbf{E}_e$ indicates the edge energy. The edge energy was calculated by Laplacian of Gaussian over DCT coefficient values in each macroblock (MB). The SFC was also computed for each MB as the number of DCT coefficient values that are larger than a predefined threshold. The biological motivation behind $\mathbf{E}_f$ is that the SFC at the eye-fixated locations in an image is noticeably higher than, on average, at random locations [135].

In contrast to the PIM-ZEN model that gives the same weight to each energy term above, in the Perceptual Importance Map based on Motion Center-Surround (PIM-MCS) model [143] (Fig. 4.1-b) the saliency is estimated by the weighted linear combination of the three energy terms,

$$\mathcal{S} = 4\mathbf{E}_m + 2\mathbf{E}_f + \mathbf{E}_e. \tag{4.6}$$

Sinha *et al.* [143] employed a motion energy term obtained by normalizing $\left(\mathbf{E}_m^1 + \mathbf{E}_m^2\right)$ in which $\mathbf{E}_m^1$ and $\mathbf{E}_m^2$ of the block $\mathbf{n}$ in P- or B-frames are, respectively, defined as

$$\mathbf{E}_m^1\left(\mathbf{n}\right) = \frac{\left\|\mathbf{v}\left(\mathbf{n}\right)\right\|^2}{dc(\mathbf{n})}, \tag{4.7}$$

$$\mathbf{E}_m^2\left(\mathbf{n}\right) = \left|\left\|\mathbf{v}\left(\mathbf{n}\right)\right\| - \overset{2}{\Uparrow}\left(\overset{2}{\Downarrow}\left(\mathcal{L}(\left\|\mathbf{v}\left(\mathbf{n}\right)\right\|)\right)\right)\right|, \tag{4.8}$$

where $dc(\mathbf{n})$ indicates the DC value of the block $\mathbf{n}$, or equivalently, the normalized sum of absolute difference of each decoded block. Here DCT means DCT of the residue for a MB in P- and B-frames. Essentially, large DC value reflects an unreliable MV. Symbol $\mathbf{v}\left(\mathbf{n}\right)$ represents the MV of block $\mathbf{n}$, $\mathcal{L}$ denotes a low-pass filter, and operators $\overset{2}{\Downarrow}$ and $\overset{2}{\Uparrow}$ represent, respectively, downsampling and upsampling by a factor of two. Note that function $\mathbf{E}_m^2$ was inspired by a center-surround mechanism and attempts to measure how different is the magnitude of $\mathbf{v}(\mathbf{n})$ from the motion magnitudes in its neighborhood.

In [143], the objective was video transcoding using the region-of-interest analysis. Unfortunately, there was no evaluation based on eye-tracking data of the performance of saliency estimation.

4.1.4 Saliency Based on Motion Center-Surround Difference Model

Liu *et al.* [104] proposed a method for detecting saliency using MVs based on the Motion Center-Surround Difference Model (MCSDM) (Fig. 4.1-c). Seven different resolutions were constructed according to MV sizes, $k \in \{4\times4, 4\times8, 8\times4, 8\times8, 8\times16, 16\times8, 16\times16\}$, where 4×4 and 16×16 are, respectively, the finest and the coarsest resolution. The MV at any resolution was obtained by averaging the corresponding 4×4 MVs; for example, a MV at resolution 16×16 is computed by averaging the 16 corresponding 4×4 MVs. The motion saliency at level k for the MV $\mathbf{v}(\mathbf{n})$ was defined as the average magnitude difference from 8-connected neighbors at the same level, denoted by $\Lambda_k(\mathbf{n})$:

$$\mathbf{M}_k(\mathbf{n}) = \frac{\sum\limits_{\mathbf{m}\in\Lambda_k(\mathbf{n})} \|\mathbf{v}(\mathbf{n}) - \mathbf{v}(\mathbf{m})\|}{|\Lambda_k(\mathbf{n})|}, \tag{4.9}$$

in which $|\cdot|$ represents cardinality. The final saliency map was defined at the lowest level (MV size 4×4) by averaging multiscale motion saliency maps:

$$\mathcal{S}(\mathbf{n}) = \frac{1}{7}\sum_{k=4\times4}^{16\times16} \mathbf{M}_k(\mathbf{n}) \tag{4.10}$$

In [104], saliency is used in the context of rate control in video coding. There was no evaluation of the saliency model on gaze data.

4.1.5 Saliency Based on Center-Surround Difference

In the Gaussian Center-Surround difference (GAUS-CS) model [42] and the Parametrized Normalization, Sum and Product - Center-Surround difference (PNSP-CS) model [43] (Fig. 4.1-d), saliency is identified through static and motion saliency maps. In this method, assuming a YCrCb video sequence had been encoded by an MPEG-4 encoder, the static saliency was determined from Intra-coded frames (I-frames) while the motion saliency was computed from Predicted frames (P-frames) and Bi-predicted frames (B-frames).

Three features were extracted from I-frames from which static saliency was computed:

1. Luminance feature (L): DC coefficients from the Y channel,

2. Color feature (Cr, Cb): Two DC coefficients, one from each color channel,

3. Texture feature (T): a vector containing the first nine AC coefficients in the Y channel. These coefficients usually contain most of the energy in each DCT block [152].

From P- and B-frames, the set of MVs was extracted as the feature of motion saliency (M).

Based on the *Feature Integration Theory of Attention* [153], a conspicuity map was constructed for each feature $f \in \{L, Cr, Cb, T, M\}$ using the center-surround difference mechanism. The sensitivity of HVS to center-surround differences decreases as the distance between the center and surround is increased. Fang *et al.* simulated this relationship by a Gaussian function:

$$\mathbf{C}_f(\mathbf{n}) = \sum_{\mathbf{m}\neq\mathbf{n}} \frac{1}{\sigma\sqrt{2\pi}} \cdot \Delta_f(\mathbf{n},\mathbf{m}) \cdot \exp \frac{-||\mathbf{n}-\mathbf{m}||_2^2}{2\sigma^2} \tag{4.11}$$

where $\mathbf{C}_f$ denotes the conspicuity of feature f, $\mathbf{m}$ and $\mathbf{n}$ are 2-D block indices, $||\mathbf{n}-\mathbf{m}||_2$ is the Euclidean distance between the centers of blocks $\mathbf{n}$ and $\mathbf{m}$, σ is the parameter of the Gaussian function, and $\Delta_f(\mathbf{n},\mathbf{m})$ represents the absolute difference between the feature values of blocks $\mathbf{n}$ and $\mathbf{m}$.

Feature maps corresponding to luminance, Cr component, Cb component and texture were all averaged into the static saliency map:

$$\mathcal{S}_s(\mathbf{n}) = \frac{1}{4} \sum_{f\in\{L,Cr,Cb,T\}} \mathbf{C}_f(\mathbf{n}). \tag{4.12}$$

On the other hand, the motion saliency map was obtained simply as $\mathcal{S}_t(\mathbf{n}) = \mathbf{C}_M(\mathbf{n})$. The final saliency map was computed separately for I- and P/B-frames:

$$\mathcal{S} = \begin{cases} \frac{\mathcal{S}_s+\mathcal{S}_t^{-1}}{2} & \text{for I} - \text{frames}, \\ \frac{\mathcal{S}_t+\mathcal{S}_s^{-1}}{2} & \text{for P/B} - \text{frames}, \end{cases} \tag{4.13}$$

where $\mathcal{S}_t^{-1}$ and $\mathcal{S}_s^{-1}$ are, respectively, the motion saliency map of the previous predicted frame and the static saliency map of the previous I-frame.

Fang *et al.* reported high accuracy of their proposed saliency method as compared to the well-known saliency models [69, 73, 50] when the comparison was performed on the public database in [69] using KLD and ROC area measurements.

4.1.6 Saliency Based on Motion Saliency Map and Similarity Map

The Motion Saliency Map - Similarity Map (MSM-SM) model [126] (Fig. 4.1-e) consists of two steps for generating the final motion saliency map. In the first step, the edge strength of each block is computed based on low-frequency AC coefficients of the luma and chroma channels [142]. These features (luma edge and chroma edge) then construct spatial saliency map according to center-surround differences. The motion saliency map is the result of refining the spatial saliency map by using an accumulated binary motion map across neighboring frames. Each binary motion frame is obtained by thresholding the magnitude of MVs, and the refining is carried out by element-wise multiplication. In the second step, the dissimilarity of DC images of the luma and chroma channels among co-located blocks over the frames is calculated by entropy. The final saliency map is the product of the spatial and motion saliency maps. Muthuswamy and Rajan [126] evaluated their model on segmented ground-truth video dataset using Equalized Error Rate (EER), the value at which the false alarm rate is equal to the miss rate [111].

4.1.7 A Convex Approximation to IKN Saliency

In the APPROX model of Hadizadeh [51] (Fig. 4.1-f), the goal was to approximate the standard IKN saliency model [73] only using DCT coefficients of image blocks. Specifically, the method is searching for the portion of the image with the normalized frequency range $[\pi/256, \pi/16]$, assuming the image spectrum is in the normalized frequency range $[0, \pi]$. The range $[\pi/256, \pi/16]$ has been chosen based on the analysis of the pyramid structure in the IKN model [54]. The Wiener filter was used to extract the portion of the original image signal in the frequency range $[\pi/256, \pi/16]$ from the spectrum of each block.

The (j, l)-th 2-D DCT coefficient of the block $\mathbf{n}$ is given by

$$\mathbf{X}_{\mathbf{n}}(j,l) = \frac{1}{4} C_j C_l \sum_{x=0}^{W-1} \sum_{y=0}^{W-1} \mathbf{n}(x,y) \cdot \cos\left(\frac{(2x+1)j\pi}{2N}\right) \cdot \cos\left(\frac{(2y+1)j\pi}{2N}\right), \tag{4.14}$$

where W is the width and height of the block $\mathbf{n}$, $C_0 = 1/\sqrt{2}$ and $C_i = 1$ for $i \neq 0$. The Wiener transfer function to separate the desired signal from the undesired signal, when the two signals are statistically orthogonal, has the following form

$$\mathcal{T}(j,l) = \frac{\mathbf{S}_S(j,l)}{\mathbf{S}_S(j,l) + \mathbf{S}_N(j,l)}, \tag{4.15}$$

where $\mathbf{S}_S(j,l)$ and $\mathbf{S}_N(j,l)$ are the power spectral densities of the desired and the undesired signal, respectively.

Hadizadeh used $1/f$-model to compute the Wiener transfer function $\mathcal{T}$. To do this, two deterministic $1/f$ 2-D signals with the size of the original image were constructed such that one covers the frequency band $[\pi/256, \pi/16]$ and the other covers the remainder of the spectrum. Then, a block of a given size (say 16×16) was extracted from the center of each signal, and 2-D DCT was performed, resulting in $\mathbf{X}_S(j,l)$ for the desired signal and $\mathbf{X}_N(j,l)$ for the undesired signal. The DCT-domain Wiener transfer function was obtained as

$$\mathcal{T}(j,l) = \frac{\mathbf{X}_S^2(j,l)}{\mathbf{X}_S^2(j,l) + \mathbf{X}_N^2(j,l)}, \tag{4.16}$$

in which $\mathbf{X}_S^2(j,l)$ and $\mathbf{X}_N^2(j,l)$ are, respectively, the powers of the desired and the undesired signal associated with DCT coefficient (j,l). The Wiener transfer function can be precomputed for typical resolutions and block sizes.

The approximation to the spatial saliency of the block $\mathbf{n}$ was computed as

$$\mathcal{S}_s(\mathbf{n}) = \sum_{(j,l)\in\mathbf{n}} \mathcal{T}^2(j,l) \cdot \mathbf{X}_\mathbf{n}^2(j,l). \tag{4.17}$$

Similarly, the temporal saliency of the block $\mathbf{n}$ was computed as

$$\mathcal{S}_t(\mathbf{n}) = \sum_{(j,l)\in\mathbf{n}} \mathcal{T}^2(j,l) \cdot \mathbf{X}_\mathbf{r}^2(j,l), \tag{4.18}$$

where $\mathbf{X}_\mathbf{r}(j,l)$ is the (j,l)-th 2-D DCT coefficient of the residual block $\mathbf{r}$. The residual block $\mathbf{r}$ was obtained by absolute difference of the block $\mathbf{n}$ in the current frame and its co-located block in the previous frame.

Finally, the overall saliency was approximated by the weighted sum of the obtained spatial and temporal saliency:

$$\mathcal{S} = (1-\alpha) \cdot \mathcal{S}_s + \alpha \cdot \mathcal{S}_t, \tag{4.19}$$

where α is a positive constant.

Hadizadeh also proposed another method for estimating the temporal saliency in case of camera motion. In this method, the camera motion is compensated prior to temporal saliency computation. Let $\mathcal{S}_t'$ be the temporal saliency after global motion compensation (GMC). Then, the overall saliency was refined as

$$\mathcal{S} = (1-\alpha_1) \cdot \mathcal{S}_s + \alpha_1 \cdot \mathcal{S}_t' + \alpha_2 \cdot \mathcal{S}_s \odot \mathcal{S}_t', \tag{4.20}$$

where α_1 and α_2 are some normalizing constants. In (4.20), α_1 trades off between the spatial saliency and the temporal saliency, and α_2 controls the mutual reinforcement.

Hadizadeh [51] evaluated his spatial saliency approximation on two common eye-tracking datasets for images, Toronto [24] and MIT [7], using AUC [151] and KLD [31] measures. Based on the reported results, his spatial saliency method is statistically very close to the standard IKN saliency model for images [73], according to a t-test comparison [118]. For evaluating the combination of spatial and temporal saliency approximation to the IKN model, the KLD on the eye-tracking dataset in [56] containing 12 standard sequences was used. The comparison was made against the IKN model with motion and flicker channels [66], and with MaxNorm normalization. Again, the two methods were shown to be statistically similar. And because saliency based on the models in [51] is convex as a function of image data, these models are referred to as *convex approximations* to the IKN model.

4.2 Experiments

In the literature, existing models have been developed for different applications and their evaluation was based on different datasets and quantitative criteria. Furthermore, models are often tailored to a particular video coding standard, and the encoding parameter settings used in the evaluation are often not reported. All of this makes a fair and comprehensive comparison more challenging. To enable meaningful comparison, in this work we reimplemented all compared methods on the same platform, and evaluated them under the same encoding conditions. The results of the comparison indicate which strategies seem promising in the context of compressed-domain saliency estimation for video, and point the way towards improving existing models and developing new ones.

4.2.1 Experimental Setup

In order to have a unified framework for comparison, we have implemented all models in MATLAB 8.0 on the same machine, an Intel (R) Core (TM) i7 CPU at 3.40 GHz and 16 GB RAM running 64-bit Windows 8.1. Where possible, we verified the implementation by comparing the results with those presented in the corresponding papers and/or by contacting the authors. As seen in Table 4.1, each model assumed a certain video coding standard. However, fundamentally, they all rely on the same type of information – MVs and DCT of residual blocks (DCT-R) or pixel blocks (DCT-P). The main difference is in the size of the blocks to which MVs are assigned or to which DCT is applied. In standards up to MPEG-4 ASP, the minimum block size was 8×8, whereas H.264/AVC allowed block sizes down to 4×4 [159]. In pursuance of a fair comparison, for which all models should accept the same input data, we chose to encode all videos in the MPEG-4 ASP format.

This choice ensured that seven out of nine models in the study did not require modification. Minor modifications were necessary to two models in order for them to accept MPEG-4 ASP input data, as noted below.

First, in MCSDM [104], which is intended to operate on MVs from a H.264/AVC bitstream (Table 4.1), we changed the minimum block size from 4×4 to 8×8. Second, in the APPROX model [51], where the spatial saliency map relies on DCT values of 16×16 pixel blocks, the 16×16 DCT was computed from the 8×8 DCTs using a fast algorithm from [60]. Also, minimum MV block size was set to 8×8.

To encode the videos used in the evaluation, the GOP (Group-of-Pictures) structure was set to IPPP with the GOP size of 12, i.e., the first frame is coded as intra (I), the next 11 frames are coded predictively (P), then the next frame is coded as I, and so on. The MV search range was set to 16 with 1/4-pel motion compensation. The quantization parameter (QP) value was set to 12. For this QP setting, the range of resulting Y-channel peak signal-to-noise ratio (PSNR) was between 28.70 dB (for *Mobile Calendar*) to 39.82 dB (for *harry-potter-6-trailer*). In the decoding stage, the DCT-P values (in I-frames) and DCT-R values (in P-frames), as well as MVs (in P-frames) were extracted for each 8×8 block. Encoding and partial decoding to extract the required data was accomplished using the FFMPEG library [4] (version 2.0.1).

4.2.2 Results

In this section, the performance of the nine compressed-domain saliency models described above is compared amongst themselves, and also against three high-performing pixel-domain models in order to gain insight into the relationship between the accuracy of the current state of the art in pixel-domain and compressed-domain saliency estimation for video.

Among the pixel-domain models, we chose AWS (Adaptive Whitening Saliency) [1], which takes only spatial information into account, as well as MaxNorm [73] (including temporal and flicker channels) and GBVS (Graph-Based Visual Saliency) [59] with DIOFM channels (DKL-color, Intensity, Orientation, Flicker, and Motion), which take both spatial and temporal information into account for estimating the saliency. AWS is frequently reported as one of the top performing models on still natural images [22, 87]. MaxNorm is known as a gold-standard model for saliency estimation. GBVS is another well-known model, often used as a benchmark for comparison. While MATLAB implementations of AWS and GBVS models are available, MaxNorm implementation is only available in C. Therefore, for all pixel-domain and compressed-domain models under study we used MATLAB implementations except MaxNorm.

Before presenting quantitative evaluation, we show a qualitative comparison of saliency maps produced by various models on a few specific examples. Fig. 4.2 shows the saliency maps for frame #150 of *City*, frame #150 of *Mobile Calendar*, frame #300 of *one-show* and frame #300 of *advert-iphone*. In the figure, the MVF of each frame is also shown beneath the corresponding frame.

In *City*, all the motion is due to camera movement. While observers typically look at the building in the center of the frame (cf. IO in Fig. 4.2), all models declare the boundary of the building as salient, where local motion is different from the global motion (GM). Meanwhile, APPROX is also able to detect the central building as salient. Note that APPROX is the only model in the study that employs GMC and its high scores on *City* are an indication that other models could benefit from incorporating GMC.

The salient objects in *Mobile Calendar*, i.e., the ball and the train, are easily detectable due to large and distinctive MVs. Therefore, all models, except AWS and GBVS, are able to successfully estimate saliency in this sequence. Recall that AWS does not use temporal information at all. GBVS also shows weak performance on this frame, in part because of its implicit center-bias applied to the saliency map.

In *one-show*, large noisy MVs in low-texture areas cause all compressed-domain models except MSM-SM to mistakenly declare them as salient regions. Recall that MSM-SM does not directly use motion magnitude but rather uses processed MVs in the form of a motion binary map. In this sequence, observers mostly focus on the face (cf. IO in Fig. 4.2) so a model that was able to perform face detection would have done well in this example. Unfortunately, none of the models is currently able to do face detection in the compressed domain - this seems like a rather challenging problem. All three pixel-domain models – MaxNorm, GBVS and AWS – also declare some part of non-salient regions as salient.

Finally, *advert-iphone* does not have any salient motion. Again, models typically detect noisy MVs as salient. MSM-SM does not detect any saliency at all since none of the MVs are strong enough according to the criteria of this model to activate its motion binary map. In this example, as well as the previous one, the sensitivity of GAUSS-CS and PNSP-CS

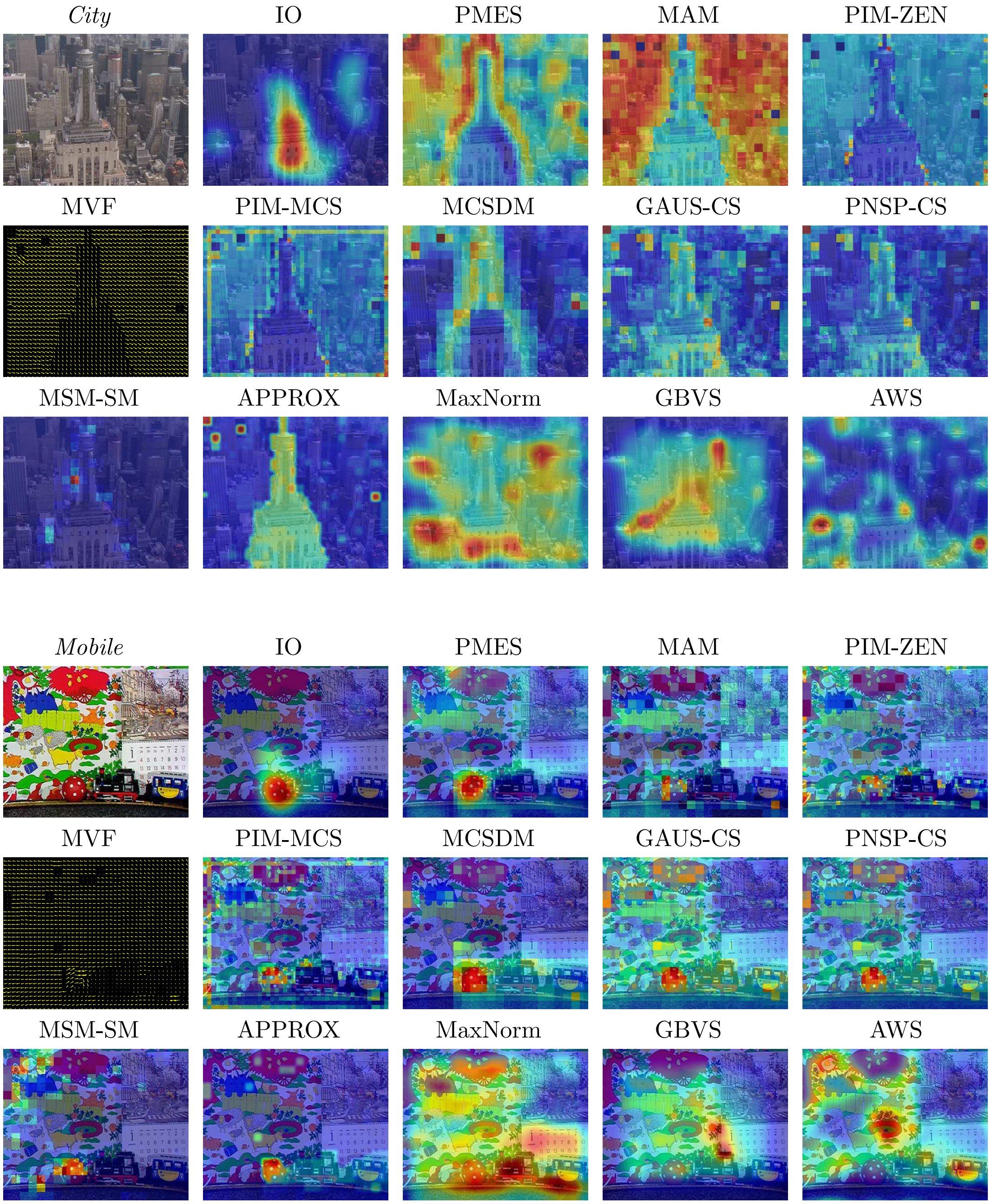


Figure 4.2: Sample saliency maps obtained by various models.

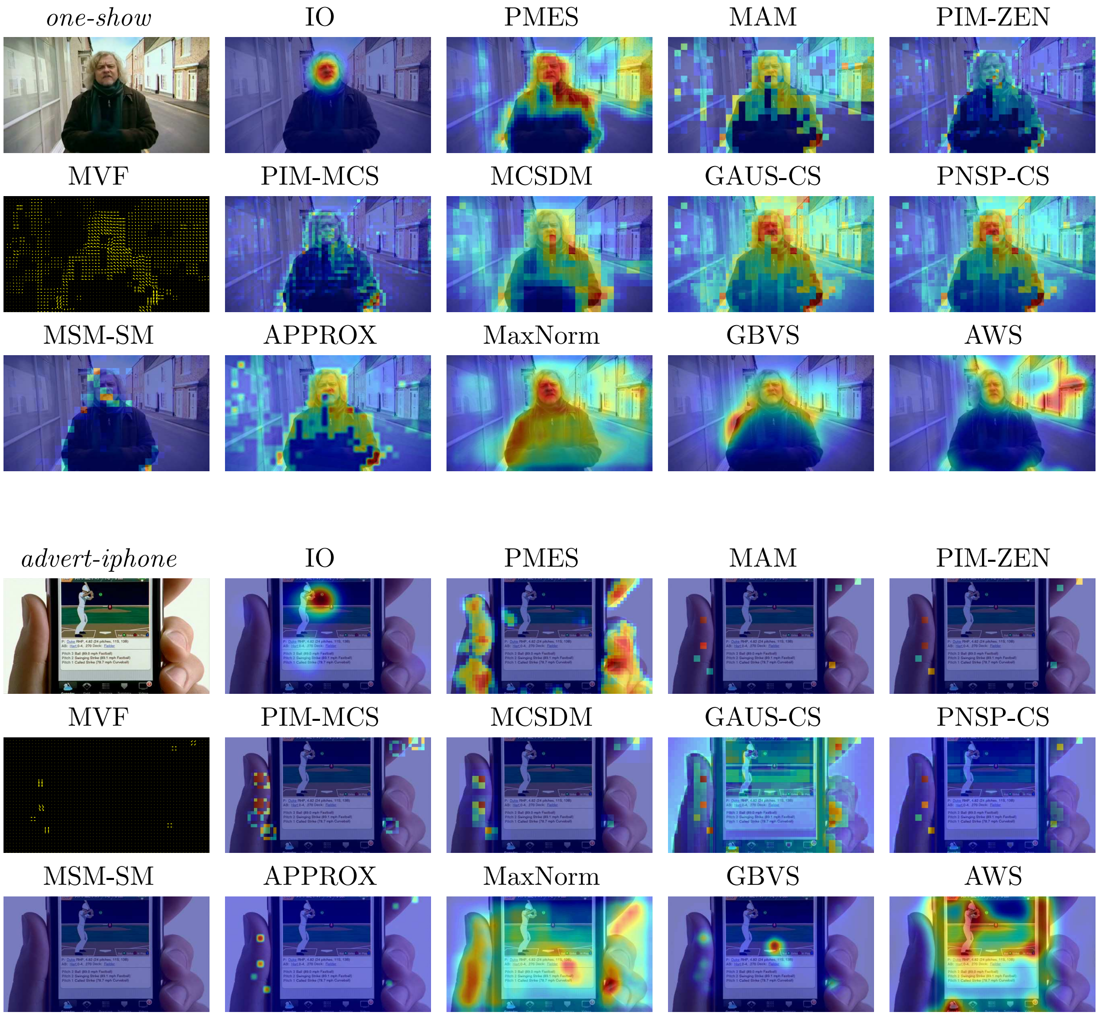


Figure 4.2: Sample saliency maps obtained by various models. *(cont.)*

to spatial saliency is clearly visible. It is not surprising that AWS performs the best in this example, because the true saliency in this example does not depend on motion.

Next, we present quantitative assessment of the saliency models using the data from the SFU and DIEM datasets. We start with the assessment based on AUC′. Fig. 4.3 shows the average AUC′ scores of various models across the test sequences. Note that all models are able to produce saliency maps for P-frames, while only some of them are able to produce a saliency map for I-frames. Hence, Fig. 4.3 (top) shows the average AUC′ scores on I-frames for those models able to handle I-frames, while Fig. 4.3 (bottom) shows the average AUC′ scores for all models on P-frames. Sequences from the SFU dataset are indicated with capital first letter.

As seen in the figure, all models achieved average AUC′ scores between those of IO, which represents a kind of an upper bound (especially on the DIEM dataset), and GAUSS, which represents center-biased, content-independent static saliency map. Note that GAUSS itself has a slightly better AUC′ score than the pure chance score of 0.5. Recall that AUC′ corrects for center bias by random sampling of control points based on empirical gaze distribution across all frames and all sequences. It is encouraging that all models are able to surpass GAUSS and achieve average AUC′ scores around 0.6.

Another interesting point in Fig. 4.3 is an indication of how difficult or easy is saliency prediction in a given sequence according to AUC′. In the figure, the sequences are sorted along the horizontal axis in decreasing order of average AUC′ score across all models. Although the order is not the same for I- and P-frames, overall, it seems that *one-show*, *sport-scramblers*, *advert-bbc4-library*, *bbc-life-in-cold-blood*, and *ami-ib4010-closeup* are the sequences where the saliency is easiest to predict, whereas *City*, *Harbour*, *Tempete*, *news-tony-blair-resignation*, and *university-forum-construction-ionic* are the sequences where saliency is hardest to predict. We will return to this issue shortly. Note that IO has better performance on the sequences from the DIEM dataset. Here, IO saliency maps are formed by the left eye gaze points and represent an excellent indicator of the ground-truth right eye gaze points. In the sequences from the SFU dataset, where IO saliency map is formed from the gaze points of the second viewing, the IO scores are not as high because the second-viewing gaze points are not as good of a predictor of the ground-truth first-viewing gaze points.

A similar set of results quantifying the models' performance according to NSS′, JSD′ and PCC are shown in Figs. 4.4, 4.5 and 4.6. As seen in Figs. 4.3, 4.4, 4.5 and 4.6, the models that are able to handle I-frames (top parts of the figures) achieve similar average scores on the I-frames as they do on the P-frames (bottom parts of the figures). For this reason, in the remainder of the chapter the results for I- and P-frames will sometimes be reported jointly. That is, in such cases, all scores will be the averages across all frames that the model is able to handle. Since the number of I-frames is much smaller than the number of P-frames, for the models that are able to handle I-frames, the effect of I-frame scores on the combined score is relatively small. Note that PCC is not center bias-corrected, so PCC scores for GAUSS are higher than all other models except IO. Also note that apart from IO, which is always ranked first, the ranking of the models depends on the metric. For example, MSM-SM scores well according to NSS′, but poorly according to JSD′.

Table 4.2 shows the ranking of test sequences according to the average scores across all models except IO and GAUSS. The sequences are ranked in decreasing order of average scores – the highest-ranked sequences are those for which the average scores are highest, and therefore seem to be the easiest for saliency prediction. Meanwhile, the lowest-ranked sequences are those for which saliency prediction seems the most difficult. Although the ranking differs somewhat for different metrics, overall, *Mobile Calendar* seems to be among the easiest sequence for saliency prediction, while *City*, *Tempete* and *pingpong-no-bodies* are among the hardest. *Mobile Calendar* contains several moving objects, including a ball and a train. The motion of each of these is sufficiently strong and different from the surroundings that almost all models are able to correctly predict viewers' gaze locations. It should be noted that the background of this sequence involves many static colorful regions that, in the absence of motion, would have the potential to attract attention. It is encouraging that the compressed-based models are generally able to identify the salient moving objects against such colorful and potentially attention-grabbing background. Meanwhile, AWS and GBVS show a relatively poor performance on this sequence.

On the other hand, *City*, *Tempete* and *pingpong-no-bodies* do not contain salient moving objects. In fact, *City* does not contain any moving objects; all the motion in this sequence is due to camera movement. *Tempete* also contains significant camera motion (zoom out) and in addition shows falling yellow leaves that act like motion noise, as they do not attract viewers' attention. *pingpong-no-bodies* does not include any salient moving object at all. While all models get confused by the falling leaves in *Tempete*, APPROX achieves a decent performance on *City* due to its use of GMC. APPROX is the only model in the study

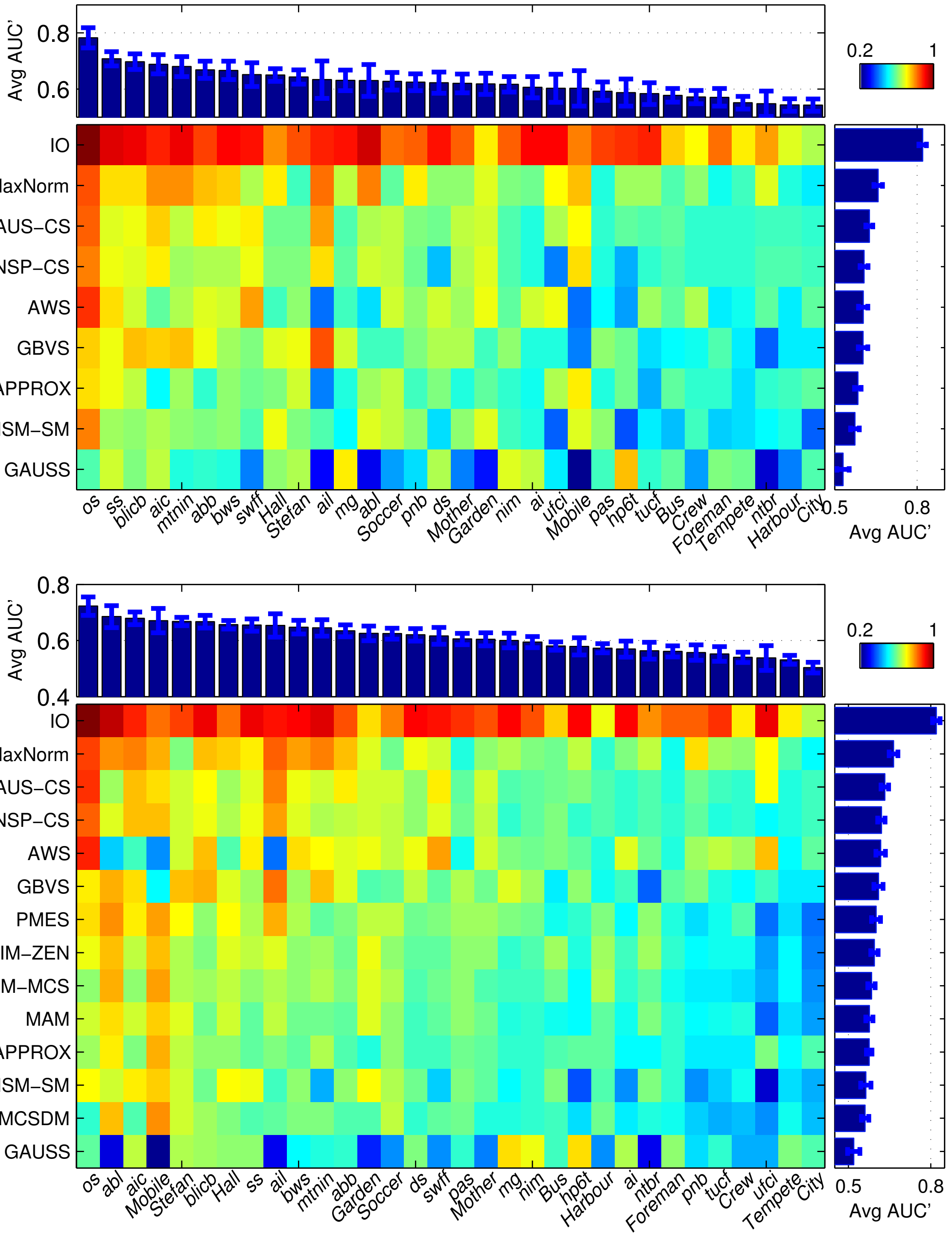


Figure 4.3: Accuracy of various saliency models over SFU and DIEM dataset according to AUC$'$ score for (top) I-frames and (bottom) P-frames. The 2-D color map shows the average AUC$'$ score of each model on each sequence. *Top*: Average AUC$'$ score for each sequence, across all models. *Right*: Average AUC$'$ scores each model across all sequences. Error bars represent standard error of the mean (SEM), $\sigma/\sqrt{n}$, where σ is the sample standard deviation of n samples.

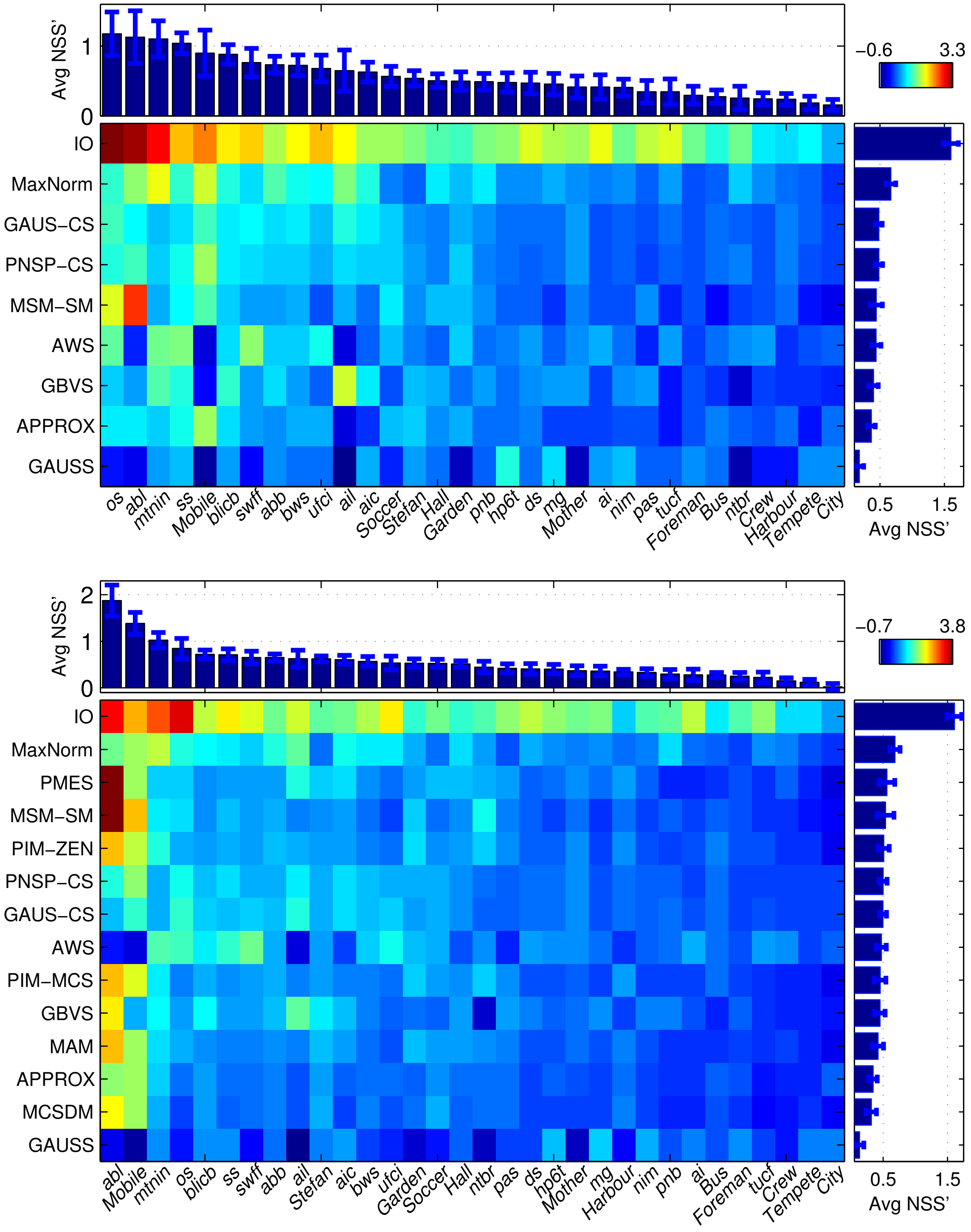


Figure 4.4: Accuracy of various saliency models over SFU and DIEM dataset according to NSS′.

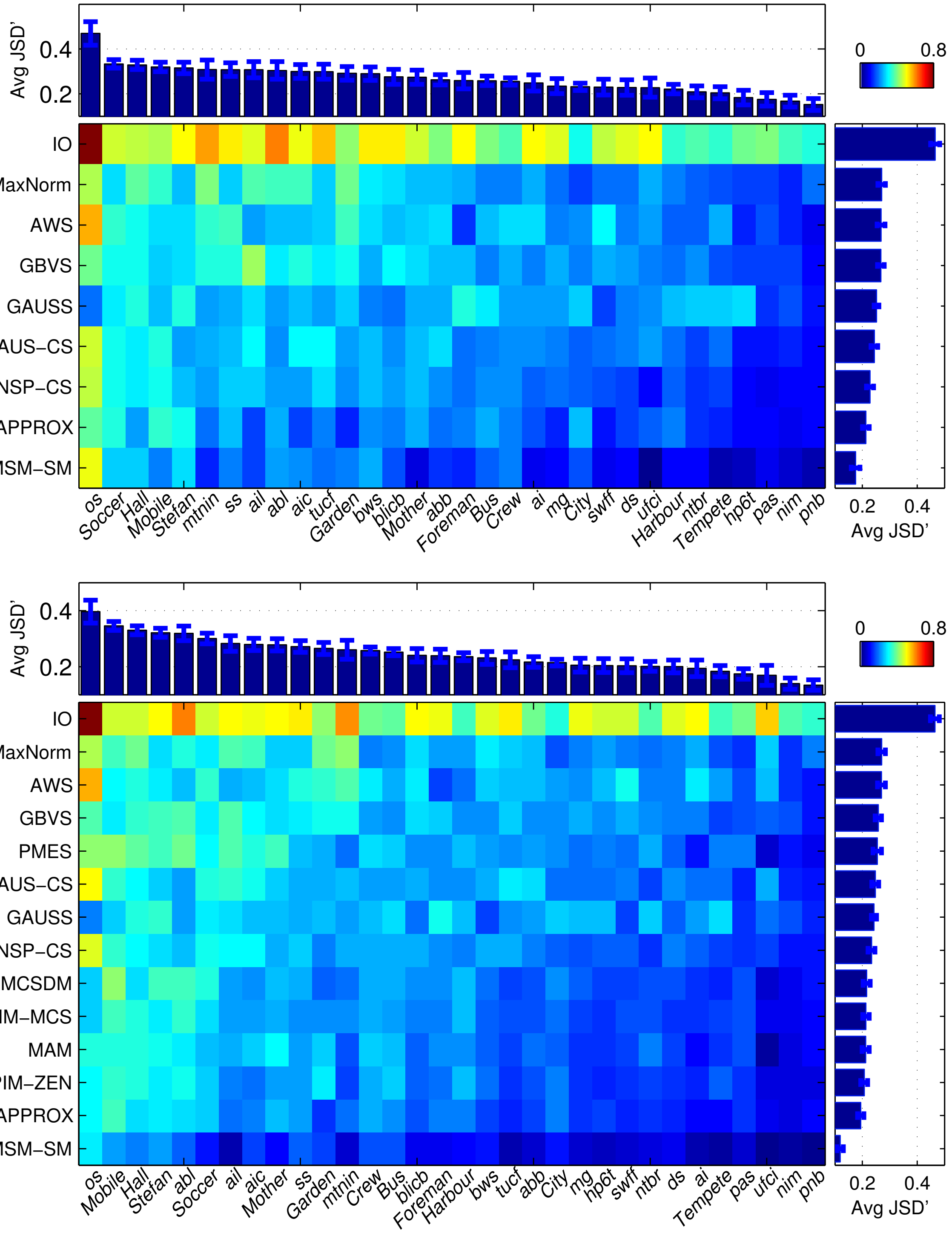


Figure 4.5: Accuracy of various saliency models over SFU and DIEM dataset according to JSD′.

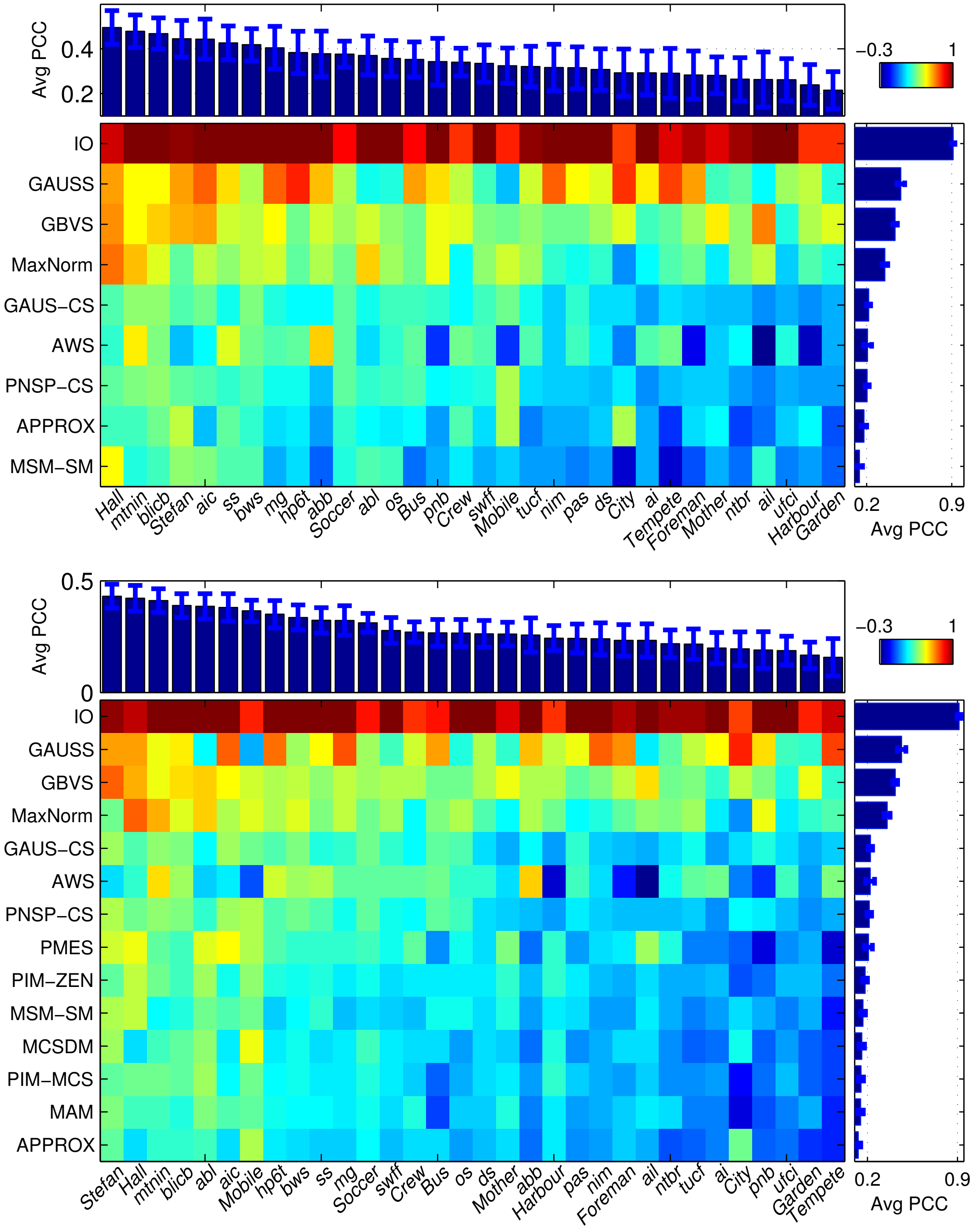


Figure 4.6: Accuracy of various saliency models over SFU and DIEM dataset according to PCC.

Table 4.2: Ranking of test sequences according to average scores across all models excluding IO and GAUSS.

Rank	**AUC′**	**JSD′**	**NSS′**	**PCC**
1	*os*	*os*	*abl*	*Stefan*
2	*abl*	*Mobile*	*Mobile*	*Hall*
3	*Mobile*	*Hall*	*mtnin*	*abl*
4	*Stefan*	*Stefan*	*os*	*mtnin*
5	*ail*	*abl*	*blicb*	*Mobile*
6	*aic*	*Soccer*	*ail*	*blicb*
7	*blicb*	*ail*	*Stefan*	*aic*
8	*Hall*	*Mother*	*ss*	*bws*
9	*ss*	*aic*	*swff*	*Soccer*
10	*Garden*	*ss*	*abb*	*hp6t*
11	*mtnin*	*Crew*	*Garden*	*ss*
12	*bws*	*Garden*	*aic*	*Mother*
13	*Soccer*	*Bus*	*Soccer*	*mg*
14	*abb*	*mtnin*	*bws*	*Crew*
15	*swff*	*blicb*	*Hall*	*swff*
16	*Mother*	*Harbour*	*ufci*	*os*
17	*pas*	*Foreman*	*ntbr*	*ds*
18	*ds*	*bws*	*pas*	*ail*
19	*Harbour*	*City*	*Mother*	*Harbour*
20	*nim*	*tucf*	*Harbour*	*Bus*
21	*mg*	*abb*	*ds*	*abb*
22	*Bus*	*swff*	*hp6t*	*pas*
23	*ntbr*	*ntbr*	*nim*	*Foreman*
24	*Foreman*	*mg*	*mg*	*nim*
25	*hp6t*	*ds*	*Bus*	*ntbr*
26	*ai*	*hp6t*	*pnb*	*tucf*
27	*pnb*	*ai*	*Foreman*	*City*
28	*Crew*	*Tempete*	*ai*	*Garden*
29	*tucf*	*pas*	*Crew*	*ufci*
30	*Tempete*	*ufci*	*tucf*	*ai*
31	*ufci*	*nim*	*Tempete*	*pnb*
32	*City*	*pnb*	*City*	*Tempete*

that employs GMC and its success on *City* is an indication that other models could be improved by incorporating GMC. Note that AWS also scores well on *City* because, as a spatial saliency model, it ignores motion and therefore does not get confused by camera motion in this sequence.

Table 4.3 lists the ranking of sequences according to IO scores. Recall that IO measures the congruence between the left and right eye gaze point in the DIEM dataset, and between the first and second viewing in the SFU dataset. Not surprisingly, the sequences from the DIEM dataset mostly occupy the top of the table, since there is high agreement between the left and right eye gaze point in each frame of each sequence. While there are some

Table 4.3: Ranking of test sequences according to different metrics for IO.

Rank	**AUC′**	**JSD′**	**NSS′**	**PCC**
1	*os*	*os*	*os*	*blicb*
2	*abl*	*abl*	*abl*	*nim*
3	*mtnin*	*mtnin*	*mtnin*	*mg*
4	*ss*	*ufci*	*Mobile*	*os*
5	*ufci*	*ss*	*ufci*	*pnb*
6	*blicb*	*tucf*	*ss*	*bws*
7	*bws*	*ai*	*swff*	*pas*
8	*ai*	*Stefan*	*ail*	*ds*
9	*mg*	*blicb*	*blicb*	*hp6t*
10	*ds*	*Mother*	*ds*	*ai*
11	*swff*	*Foreman*	*ai*	*ufci*
12	*hp6t*	*ail*	*bws*	*ss*
13	*ail*	*mg*	*pas*	*mtnin*
14	*aic*	*bws*	*tucf*	*aic*
15	*tucf*	*aic*	*hp6t*	*abl*
16	*pas*	*ds*	*mg*	*abb*
17	*Stefan*	*Hall*	*Mother*	*ail*
18	*abb*	*Soccer*	*Soccer*	*swff*
19	*nim*	*Mobile*	*abb*	*Stefan*
20	*Mother*	*swff*	*aic*	*ntbr*
21	*Foreman*	*hp6t*	*Stefan*	*tucf*
22	*pnb*	*Garden*	*pnb*	*Foreman*
23	*Mobile*	*pas*	*nim*	*Hall*
24	*Hall*	*abb*	*Foreman*	*Tempete*
25	*Soccer*	*Bus*	*ntbr*	*Mother*
26	*ntbr*	*Crew*	*Hall*	*Soccer*
27	*Bus*	*ntbr*	*Garden*	*Bus*
28	*Garden*	*nim*	*Bus*	*Mobile*
29	*Tempete*	*Harbour*	*Tempete*	*Garden*
30	*Crew*	*Tempete*	*Crew*	*Crew*
31	*Harbour*	*pnb*	*Harbour*	*Harbour*
32	*City*	*City*	*City*	*City*

sequences from the SFU dataset that appear in the top half of the table according to some metrics (for example, *Stefan*, *Mother and Daughter*, and *Foreman* according to JSD′, and *Mobile Calendar* according to NSS′), most SFU sequences are at the bottom of the table. This is not surprising, because the congruence between the first and second viewing of the sequence is much lower than that between the left and right eye gaze point. In particular, *City* is at the bottom of the table according to all four metrics. In this sequence, the central building attracts viewers' gaze in both viewings, but because the building occupies a large portion of the frame, the actual gaze points on the first and second viewing may end up being very far apart, leading to low scores even for IO.

The average scores of saliency models across all sequences in both datasets are shown in Fig. 4.7 for various accuracy metrics. Note that the horizontal axis has been focused on the relevant range of scores. Not surprisingly, IO achieves the highest scores regardless of the metric. At the same time, the effect of center bias is easily revealed by comparing AUC and NSS scores to their center bias-corrected versions AUC$'$ and NSS$'$. For example, the AUC measures the accuracy of saliency prediction of a particular model against a control distribution drawn uniformly across the frame. Since the uniform distribution is a relatively poor control distribution for saliency and easy to outperform, all models achieve a higher AUC score compared to their AUC$'$ score, which uses a control distribution fitted to the empirical gaze points shown in Fig. 3.3. This effect is most visible in the GAUSS benchmark model, which has the AUC score of around 0.8 (higher than all the models except IO), but the AUC$'$ score of only slightly above 0.5 (lower than all other models). This over-exaggeration of the accuracy of a simple scheme such as GAUSS when plain AUC is used was the reason why [131, 76] suggest center bias correction via non uniform control sampling. The center bias-corrected AUC$'$ score is a better reflection of the models' accuracy. Center bias also has a significant effect on NSS, but a less pronounced effect on JSD. It can also be observed that GAUSS (and then GBVS) achieves a higher PCC score than any other method except IO, due to the accumulation of fixations near the center of the frame.

In addition to the average scores, another type of assessment of a model's performance is counting its number of appearances among top performing models for each sequence [117]. To this end, a multiple comparison test is performed using Tukey's honestly significant difference as the criterion [63]. Specifically, for each sequence, we compute the average score of a model across all frames, as well as the 95% confidence interval for the average score. Then we find the model with the highest average score (excluding IO and GAUSS), and find all the models whose 95% confidence interval overlaps that of the highest-scoring model. All such models are considered top performers for the given sequence. Fig. 4.8 shows two examples. In the left panel, MaxNorm has the highest average AUC$'$ score, and its 95% confidence interval does not overlap any of the other models' intervals. Hence, in this case, MaxNorm is the sole top performer. On the other hand, in the right panel, the 95% confidence interval of the top-scoring MCSDM overlaps the corresponding intervals of PMES, PIM-MCS, APPROX and MaxNorm. In this case, all five models are considered top

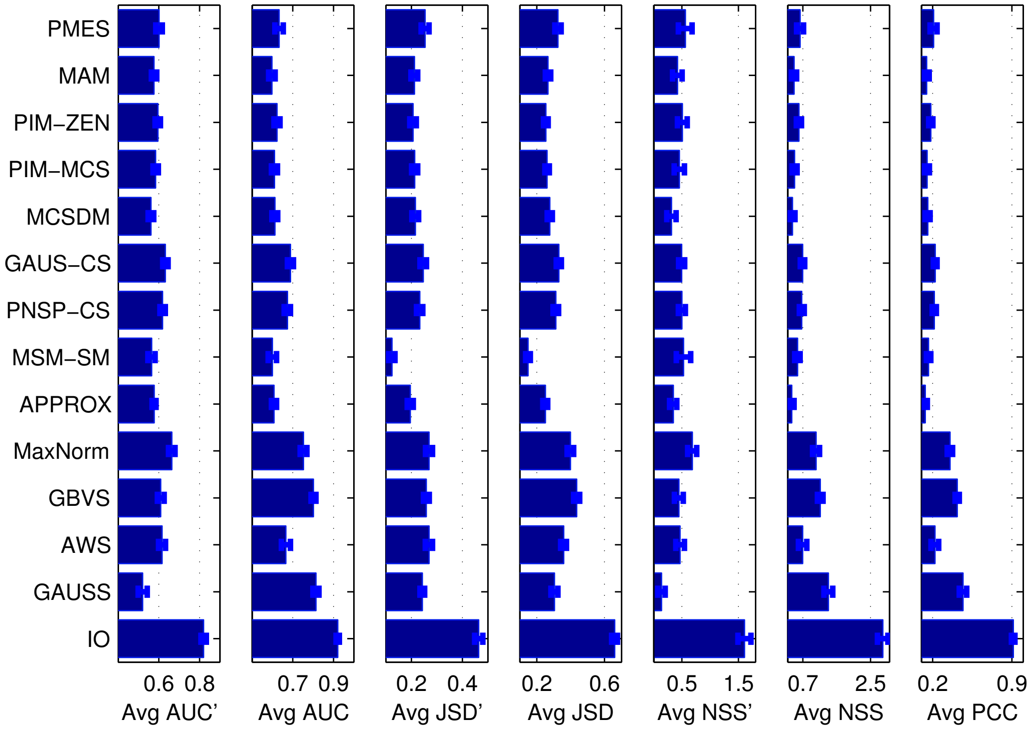


Figure 4.7: Evaluation of models using various metrics.

performers. The number of appearances among top performers for each model is shown in Fig. 4.9. These results show similar trends as average scores, with MaxNorm, AWS, GBVS, PMES, GAUS-CS and PNSP-CS often being among top performers, while APPROX, MAM and MCSDM rarely offering top scores.

Since a compressed video representation always involves some amount of information loss, it is important to determine the sensitivity of the compressed-domain saliency model to the amount of compression. Note that the predictive power of MVs and DCT coefficients could change dramatically across the compression range. To study this issue, we repeated the experiments described above for different amounts of compression, by varying the QP parameter. The quality of encoded video drops as the QP increases, as shown by the PSNR values on the SFU dataset for QP $\in \{4, 8, 12, 16, 20, 24\}$ in Table 4.4.

Fig. 4.10 shows how the average AUC$'$ and NSS$'$ scores change as a function of the QP parameter. In this experiment, MaxNorm, AWS and GBVS were applied to the decoded video, hence they effectively used the same data as compressed-domain models, but in the pixel domain after full video reconstruction. As seen in the figure, the models typically score slightly poorer as the video quality drops, because of the less accurate MVs and DCT

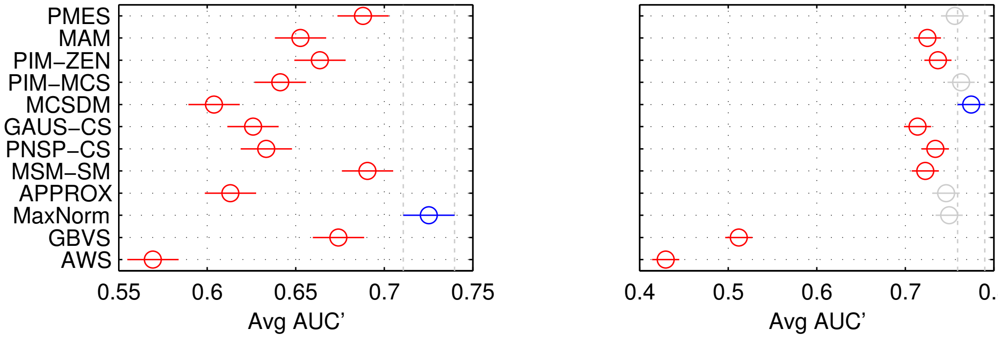


Figure 4.8: Illustration of multiple comparison test for (left) *Hall Monitor* and (right) *Mobile Calendar* sequences using AUC′.

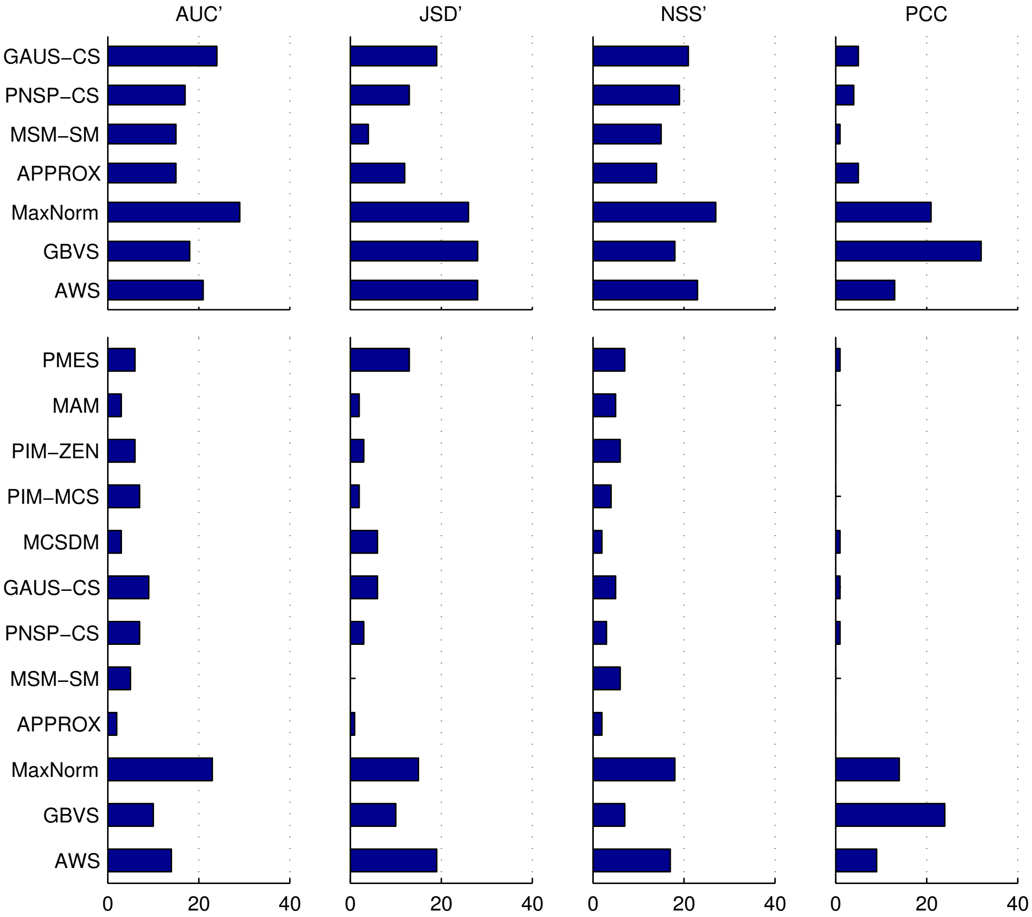


Figure 4.9: The number of appearances among top performers, using various evaluation metrics. Results for I-frames are shown at the top, those for P-frames at the bottom.

Table 4.4: The PSNR value (in dB) for various QP values on sequences from the SFU dataset

Sequence	*Bus*	*City*	*Crew*	*Foreman*	*Garden*	*Hall*	*Harbour*	*Mobile*	*Mother*	*Soccer*	*Stefan*	*Tempete*	**Avg.**
QP=4	36.8	37.6	38.8	38.2	37.3	39.3	36.6	36.2	41.0	38.6	38.0	36.9	**37.9**
QP=8	32.0	33.4	35.0	34.3	32.1	36.1	32.0	31.4	38.0	34.4	33.2	32.4	**33.7**
QP=12	29.4	31.3	33.0	32.4	29.3	34.3	29.6	28.7	36.2	32.4	30.5	30.0	**31.4**
QP=16	27.9	30.1	31.8	31.1	27.4	32.9	28.1	27.0	35.1	31.2	28.7	28.4	**30.0**
QP=20	26.6	29.1	30.9	30.1	26.0	31.8	26.9	25.6	34.2	30.3	27.3	27.3	**28.8**
QP=24	25.8	28.5	30.3	29.4	25.0	30.8	26.2	24.6	33.5	29.6	26.2	26.5	**28.0**

Table 4.5: Average processing time in milliseconds per frame.

Model	AWS	GBVS	MAM	PMES	MaxNorm	GAUS-CS	PNSP-CS	PIM-ZEN	APPROX	PIM-MCS	MSM-SM	MCSDM
Time (ms)	1535	783	166	98	91	57	57	18	11	9	7	4

coefficients at higher compression ratios. Nonetheless, the saliency prediction performance is still reasonably consistent over this range of QP values, leading to the conclusion that the models' performance is not too sensitive to encoding parameters over a reasonable range of video qualities. This observation is consistent with studies taken by Le Meur [94], and Milanfar and Kim [87].

The average processing time per frame on the SFU dataset (CIF resolution videos at 30 fps) is listed in Table 4.5. The time taken for extracting MVs and DCT values from the bitstream is excluded. Please note that these results correspond to MATLAB implementations of the models, except MaxNorm that was implemented in C, and the processing time can be significantly decreased by implementation in a medium-level programming language such as C/C++. Despite this, some of the models are fast enough for real time performance (under 33 ms par frame) even when implemented in MATLAB. Discussion of accuracy and complexity of the models is presented in the next section.

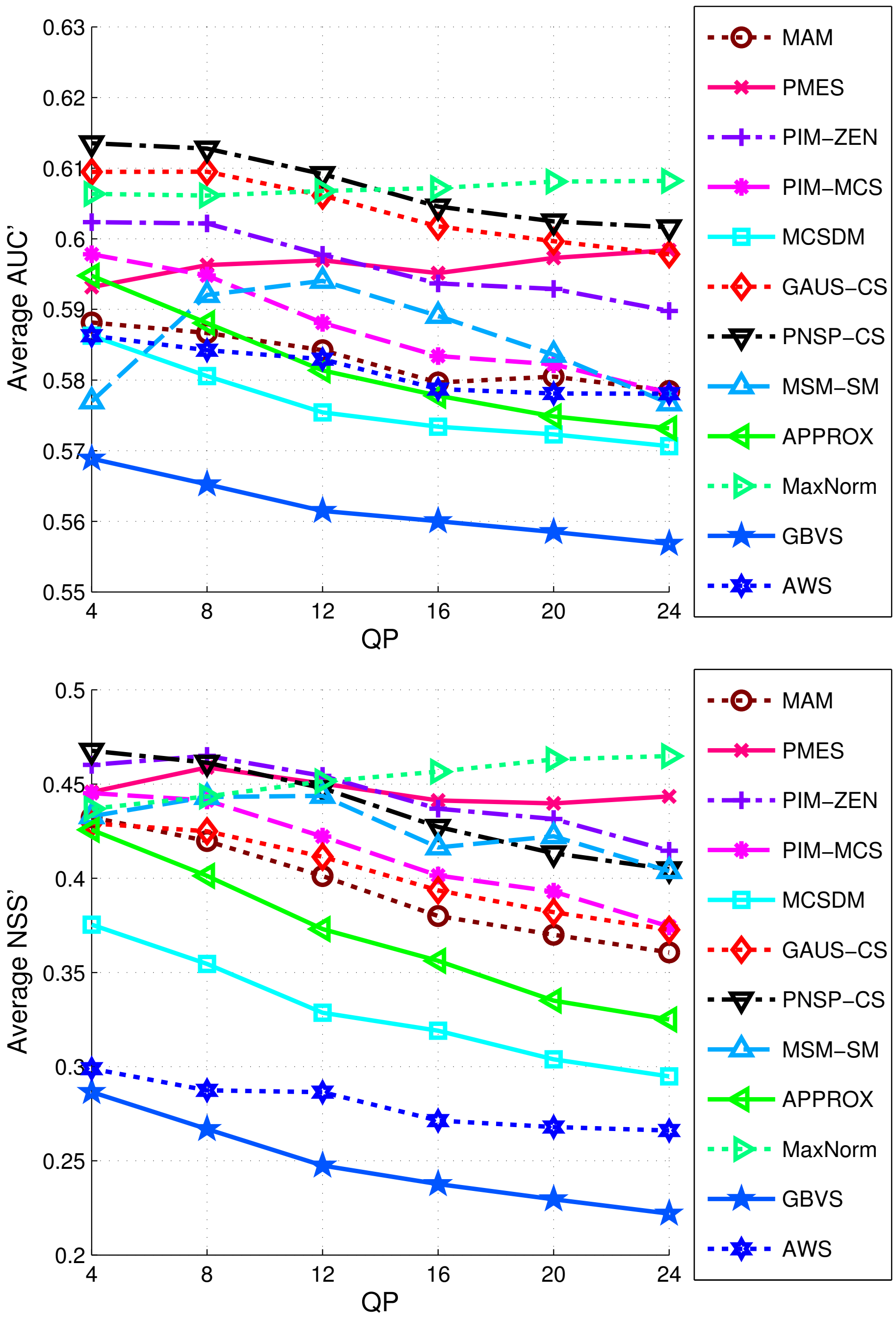


Figure 4.10: The relationship between the QP parameter and the models' accuracy on the SFU dataset.

4.3 Discussion

Considering the results in Fig. 4.7 and Fig. 4.9, MaxNorm, GBVS, AWS, PMES, GAUS-CS and PNSP-CS consistently achieve high scores across different metrics. It is encouraging that the accuracy of some compressed-domain models is competitive with that of pixel-domain models. This was already claimed by Fang *et al.*[43, 42], though based on a smaller study. Note that, in general, achieving a high score with one metric does not guarantee a high score with other metrics. As an example, MSM-SM achieves a relatively high average scores across several metrics, but the lowest JSD and JSD$'$ score. Hence, the fact that MaxNorm, GBVS, AWS, PMES, GAUS-CS and PNSP-CS perform consistently well across all metrics considered in this study lends additional confidence in their accuracy.

PMES was the first compressed-domain saliency model, proposed in 2001, and it only uses MVs to estimate saliency. It is well known that motion is a strong indicator of saliency in dynamic visual scenes [120, 111, 68], so it is not surprising that MVs would be a powerful cue for saliency estimation. PMES estimates saliency by considering two properties: large motion magnitude in a spatio-temporal region, and the lack of coherence among MV angles in that region. These two properties seem to describe salient objects reasonably well in most cases, as demonstrated by the results. Taken together, they resemble a center-surround mechanism where a region is considered salient if it sufficiently "stands out" from its surroundings.

GAUS-CS and PNSP-CS show high accuracy in both I- and P-frames. Both models are based on the center-surround difference mechanism, and both employ MVs for saliency estimation in P-frames and DCT of pixel values in I-frames. The capability of center-surround difference mechanism to predict where people look has been discussed extensively [73], so their success is also not surprising.

Although PIM-MCS and MSM-SM also attempt to employ the center-surround difference mechanism, their scores are not as consistently high as those of GAUS-CS and PNSP-CS. The reason may be that in GAUS-CS and PNSP-CS models, the contrast is inversely proportional to the distance between the current DCT block and all other DCT blocks in the frame, which means that they consider not only the contrast between blocks, but also the distance between them. This seems to be a good strategy for compressed-domain saliency estimation.

According to the results in the previous section, the lowest-scoring models on most metrics were APPROX and MCSDM. Incidentally, both these models were originally developed for a different type of input data and had to be modified for this comparison, which may degrade their performance. Specifically, both models were developed for MVs corresponding to 4×4 blocks, whereas the evaluation in this work employed MVs corresponding to 8×8 blocks. Additionally, APPROX originally assumed DCT coefficients of 16×16 blocks from a raw (uncompressed) frame, whereas in this evaluation, 16×16 DCT was computed from four 8×8 DCTs of compressed frames, which involved quantization noise. Looking at the results in Figs. 4.3 and 4.4, the gap between GAUS-CS (PNSP-CS) and APPROX is smaller in the top parts of the figures (I-frames) than in the bottom parts (P-frames), which indicates that the effect of quantization noise was not as detrimental to the performance of APPROX as the switch from 4×4 MVs to 8×8 MVs.

The influence of global (camera) motion on visual saliency is still an open research problem, with limited work in the literature addressing this issue. Reference [9] studied separately the effect of pan/tilt and zoom-in/-out. It was found that in the case of pan/tilt, the gaze points tend to shift towards the direction of pan/tilt, in the case of zoom-in, they tend to concentrate near the frame center, and in the case of zoom-out, they tend to scatter further out. On the other hand, according to [18], the presence of camera motion tends to concentrate gaze points around the center of the frame "according to the direction orthogonal to the tracking speed vector."

Among the models tested in the present study, only APPROX took GM into account by removing it before the analysis of MVs. This paid off in the case of *City*, which was overall the most difficult sequence for other spatio-temporal saliency models in Figs. 4.3 and 4.4. However, GMC did not help much in the case of *Tempete* or *Flower Garden*. In fact, *Tempete* contains strong zoom-out, which, according to [9], would tend to scatter the gaze points around the frame. However, Figs. 4.3 and 4.4 show that GAUSS, with its simple center-biased saliency map, scores well here (even with center-bias-corrected metrics), suggesting that the gaze points are still located near the center of the frame. This is due to the presence of a yellow bunch of flowers in the center of the frame, which turns out to be highly attention-grabbing. Apparently, the key to accurate saliency estimation in *Tempete* is not in the motion, but rather in the color present in the scene. *Flower Garden* is another example where GMC did not pay off. The viewers' gaze in this sequence is attracted to the objects in the background, specifically the windmill and the pedestrians, whose motion tends to be zeroed out after GMC on 8×8 MVs. Overall, the results suggest that GM is not sufficiently well handled by current compressed-domain methods, and that further research is needed to make progress on this front.

Considering models' complexity and processing time in Table 4.5, MCSDM is the fastest while AWS is the most demanding in terms of processing. While MaxNorm, AWS, GBVS, PMES, GAUS-CS and PNSP-CS all scored highly in terms of accuracy, the processing time of GAUS-CS and PNSP-CS is only half that of PMES and MaxNorm, one fifteenth that of GBVS, and one thirtieth that of AWS, which would make them preferable in real-world applications. Although entropy decoding time of MVs and DCT residuals was not taken into account and although PMES skips I-frames and uses only MVs in P-frames, GAUS-CS and PNSP-CS can also be forced to skip I-frames and rely only on MVs in P-frames, which would make the decoding time required for these models equal to that of PMES, hence the relative processing time in Table 4.5 would still be a good indicator of their relative complexity.

It is interesting to note that the five fastest models (MCSDM, MSM-SM, PIM-MCS, APPROX and PIM-ZEN) are able to offer real-time performance (under 33 ms per frame) on CIF sequences even with a relatively inefficient MATLAB implementation. This suggests that an optimized implementation of some of these models may be a good candidate for low-weight saliency estimation for applications such as real-time video stream analysis and video quality monitoring.

As mentioned before, this study was performed on MPEG-4 ASP bitstreams because this way, the majority of the models (seven out of nine) did not require modification. The two models that required modification, because they were developed for H.264/AVC bitstreams, did not score particularly well, presumably because they have tailored their parameters to smaller MV block sizes in H.264/AVC. Had the test been done on H.264/AVC bitstreams, these two models may have scored higher, but there would have been considerable ambiguity in how to extend the other majority of the models to handle smaller block sizes. Looking towards the future, however, bitstreams that conform to H.264/AVC and the more recent High Efficiency Video Coding (HEVC) [147] standard are of considerable interest, and further research is needed to build saliency models that take advantage of their peculiarities.

4.4 Conclusions

In this chapter we attempted to provide a comprehensive comparison of nine compressed-domain visual saliency models for video. All methods were reimplemented in MATLAB and tested on two eye-tracking datasets using several accuracy metrics. Care was taken to correct for center bias and border effects in the employed metrics, which were issues found in earlier studies on visual saliency model evaluation. The results indicate that in many cases, reasonably accurate visual saliency estimation is possible using only motion vectors from the compressed video bitstream. This is encouraging considering that motion vectors occupy a relatively small portion of the bitstream (usually around 20%) and no further decoding is required. Several compressed-domain saliency models showed competitive accuracy with some of the best currently known pixel-domain models. On top of that, the fastest compressed-domain methods are fast enough for real-time saliency estimation on CIF video even with a relatively inefficient MATLAB implementation, which suggests that their optimized implementation could be used for online saliency estimation in a variety of applications.

Many sequences that have turned out to be difficult for models to handle contain global (camera) motion. The influence of GM on visual saliency is not very well understood, and most models in the study did not account for it. A number of compressed-domain global motion estimation (GME) methods, based on motion vectors alone, have been developed recently (cf. Section 7.2.2), so it is reasonable to expect that compressed-domain saliency models should be able to benefit from these developments.

Chapter 5

Compressed-Domain Correlates of Fixations

In this chapter, we first present two video features called *Motion Vector Entropy* (MVE) and *Smoothed Residual Norm* (SRN) that can be computed from the compressed video bitstream using motion vectors (MVs), block coding modes (BCMs), and transformed prediction residuals. Next, we propose another feature as a measure of incompressibility, called *Operational Block Description Length* ($OBDL$) that is computed directly from the output of the entropy decoder, which is the first processing block in a video decoder. No further decoding of the compressed bitstream is needed for computing OBDL, whereas partial decoding is required to extract MVE and SRN. Finally, the potential of these three features to predict saliency is demonstrated by comparing their statistics around human fixation points in a number of videos against the non-attended points selected randomly away from fixations. The high statistical agreement between feature values and fixation points qualifies these features as *correlates* of fixations. That is to say, these features are highly indicative of attended regions in video.

5.1 Compressed-domain features

Typical video compression consists of motion estimation and motion-compensated prediction, followed by transformation, quantization and entropy coding of prediction residuals and MVs. These processing blocks have existed since the earliest video coding standards, getting more sophisticated over time. Our compressed-domain features are computed from the outputs of these basic processing blocks. For concreteness, we shall focus on the H.264/AVC coding standard [159], but the feature computation can be adjusted to other video coding standards, including the latest High Efficiency Video Coding (HEVC) [147]. Due to the focus on H.264/AVC, our terminology involves 16×16 macroblocks (MBs), block coding modes (BCM: INTER, INTRA, SKIP) for various block sizes (4×4, 8×8, etc.), and the 4×4 integer transform that we shall refer to as "DCT" although it is only an approximation to the actual Discrete Cosine Transform.

5.1.1 Motion Vector Entropy (MVE)

Motion vectors (MVs) in the video bitstream carry important cues regarding temporal changes in the scene. A MV is a two-dimensional vector $\mathbf{v} = (v_x, v_y)$ assigned to a block, which represents its offset from the best-matching block in a reference frame. The best-matching block is found via motion estimation, often coupled with rate-distortion optimization. The MV field can be considered as an approximation to the optical flow.

When a moving object passes through a certain region in the scene, it will generate different MVs in the corresponding spatio-temporal neighborhood; some MVs will correspond to the background, others to the object itself, and object's MVs themselves may be very different from each other, especially if the object is flexible. On the other hand, an area of the scene covered entirely by the background will tend to have consistent MVs, caused mostly by the camera motion. From this point of view, variation of MVs in a given spatio-temporal neighborhood could be used as an indicator of the presence of moving objects, which in turn shows potential for attracting attention.

Before computing the feature that describes the above-mentioned concept, block processing in the frame is performed as follows. SKIP blocks are assigned a zero MV, while INTRA-coded blocks are excluded from analysis. Then all MVs in the frame are mapped to 4×4 blocks; for example, a MV assigned to an 8×8 block is allocated to all four of its constituent 4×4 blocks, etc.

We define a *motion cube* as a causal spatio-temporal neighborhood of a given 4×4 block $\mathbf{n}$, as illustrated in Fig. 5.1. The spatial dimension of the cube (W) is selected to be twice the size of the fovea (2° of visual angle [66]) while the temporal dimension (L) is set to 200 ms. For example, for CIF resolution (352×288) video at 30 frames per second and viewing conditions specified in [56], these values are $W = 52$ pixels and $L = 6$ frames.

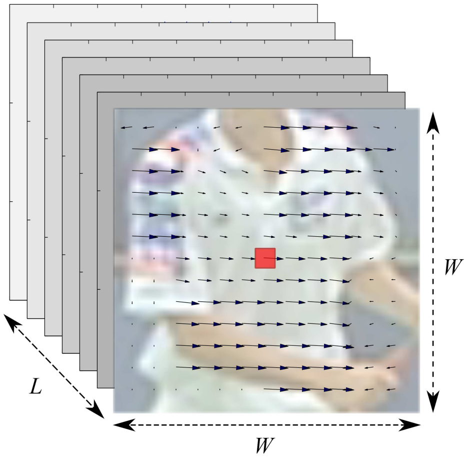


Figure 5.1: Motion cube $N(\mathbf{n})$ associated with block $\mathbf{n}$ (shown in red) is its causal spatio-temporal neighborhood of size $W \times W \times L$.

To give a quantitative interpretation of MV variability within the motion cube, we use the normalized Motion Vector Entropy (MVE) map, defined as

$$\mathcal{S}_{MVE}(\mathbf{n}) = -\frac{1}{\log N} \sum_{i \in \mathrm{H}(N(\mathbf{n}))} \frac{n_i}{N} \cdot \log\left(\frac{n_i}{N}\right), \tag{5.1}$$

where $N(\mathbf{n})$ is the motion cube associated with the 4×4 block $\mathbf{n}$, $\mathrm{H}(\cdot)$ is the histogram, i is the bin index, n_i is the number of MVs in bin i, and $N = \sum_i n_i$. The factor $1/\log N$ in (5.1) serves to normalize MVE so that its maximum value is 1, achieved when $n_i = n_j, \forall i, j$. The minimum value of MVE is 0, achieved when $n_i = 0$ for all i except one.

Histogram $\mathrm{H}(N(\mathbf{n}))$ is constructed from MVs of inter-coded blocks within the motion cube. Depending on the encoder settings, such as search range and MV accuracy (full pixel, half pixel, etc.), each MV can be represented using a finite number of pairs of values (v_x, v_y). Each possible pair of values (v_x, v_y) under the given motion estimation settings defines one bin of the histogram. The cube is scanned and every occurrence of a particular (v_x, v_y) results in incrementing the corresponding n_i by 1.

It has been observed that large-size blocks are more likely to be part of the background, whereas small-size blocks, arising from splitting during the motion estimation, are more likely to belong to moving objects [13] (cf. Fig. 7.6). To take this into account, during block processing, 4×4 INTER blocks are assigned random vectors from a uniform distribution over the motion search range prior to mapping MVs from larger INTER and SKIP blocks to their constituent 4×4 blocks. This way, a motion cube that ended up with many 4×4 INTER blocks during encoding is forced to have high MVE.

5.1.2 Smoothed Residual Norm (SRN)

Large motion-compensated prediction residual is an indication that the best-matching block in the reference frame is not a very good match to the current block. This in turn means that the motion of the current block cannot be well predicted using the block translation model, either due to the presence of higher-order motion or due to (dis)occlusions. Of these two, (dis)occlusions often yield higher residuals. Moreover, (dis)occlusions are associated with surprise, as a new object enters the scene or gets revealed behind another moving object, so they represent a potential attractor of attention. Therefore, large residuals might be an indicator of regions that have the potential to attract attention.

The "size" of the residual is usually measured using a certain norm, for example ℓ_p norm for some $p \geq 0$. In this paper we employ the ℓ_0 norm, i.e., the number of non-zero elements, since it is easier to compute than other popular norms such as ℓ_1 or ℓ_2. For any macroblock (MB), we define *Residual Norm* (*RN*) as the norm of the quantized transformed prediction residual of the MB, normalized to the range $[0, 1]$. For the ℓ_0 norm employed in this paper, RN would be:

$$RN(\mathbf{z}) = \frac{1}{256}\|\mathbf{z}\|_0, \tag{5.2}$$

where $\mathbf{z}$ denotes the quantized transformed residual of a 16 × 16 MB.

Finally, the map of MB residual norms is smoothed spatially using a 3×3 averaging filter, temporally using a moving average filter over previous L frames, and finally upsampled by a factor of 4 using bilinear interpolation. The result is the Smoothed Residual Norm (SRN) map, with one value per 4 × 4 block, just like the MVE map.

5.1.3 Operational Block Description Length (OBDL)

The Operational Block Description Length (OBDL) is computed directly from the output of the entropy decoder, which is the first processing block in a video decoder. No further decoding of the compressed bitstream is needed for computing this feature. The number of bits spent on encoding each MB is extracted and mapped to the unit interval $[0, 1]$, where the value of 0 is assigned to the MB(s) requiring the least bits to code and the value of 1 is assigned to the MB(s) requiring the most bits to code, among all MBs in the frame. The normalized OBDL map is smoothed by convolution with a 2-D Gaussian of standard deviation equal to 2° of visual angle. Although the spatially smoothed OBDL map is already a solid saliency measure, we observed that an additional improvement in the accuracy of saliency predictions is possible by performing further temporal smoothing. This conforms with what is known about biological vision [12, 10, 124], where temporal filtering is known to occur in the earliest layers of visual cortex. Specifically, we apply a simple causal temporal averaging over 100 ms to obtain a feature derived from the OBDL.

The OBDL simplifies many of the previously proposed compression-based measures of saliency. For example, representing the block by a set of DCT coefficients resembles the subspace [64, 48, 114], sparse [65, 98] or independent component [24] decomposition, which are at the core of various saliency measures. The differential encoding of DCT coefficients or, more generally, spatial block prediction, resembles the center-surround operations of [73], while motion-compensated prediction resembles the surprise mechanism of [69]. In fact, given the well known convergence of modern entropy coders to the entropy rate of the source being compressed

$$H = \frac{1}{n} \sum_{i} \log \frac{1}{P(x_i)}, \tag{5.3}$$

where $P(x)$ is the probability of symbol x, the number of bits produced by the entropy coder is a measure of the conditional self information of each block. Hence, a video compressor is a very sophisticated implementation of the saliency principle of Bruce and Tsotsos [24], which evaluates saliency as

$$\mathcal{S}(x) = \log \frac{1}{P(x)}. \tag{5.4}$$

While Bruce and Tsotsos [24] proposed a simple independent component analysis to extract features x from the image pixels, the video compressor performs a sequence of operations involving motion compensated prediction, DCT transform of the residuals, predictive coding of DCT coefficients, quantization, and entropy coding, all within a rate-distortion optimization framework. This results in a much more accurate measure of information and, moreover, is much simpler to obtain in practice, given the widespread availability of video codecs.

The proposed OBDL is even simpler to extract from compressed bitstreams than the other forms of compressed-domain information including our compressed-domain features in Sections 5.1.1 and 5.1.2, because the recovery of MVs or residuals is not required. Overall, the OBDL combines the accuracy of the pixel-domain saliency measures (which will be demonstrated later in the dissertation) with the computational efficiency of their compressed-domain counterparts.

5.2 Discriminative Power of the Proposed Compressed-Domain Features

To assess how indicative of fixations are the three proposed features, we use an approach similar to the protocol of Reinagel and Zador [135], who performed an analogous analysis for two other features – spatial contrast and local pixel correlation – on still natural images. Their analysis showed that in still images, on average, spatial contrast is higher, while local pixel correlation is lower, around fixation points compared to random points.

We follow the same protocol, using two eye-tracking datasets, the SFU [56] and DIEM [2] (cf. Section 3.1 for details on the datasets). In these experiments, each video was encoded in the H.264/AVC format using the FFMPEG library [4] (version 2.3) with a quantization parameter (QP) of 30 and 1/4-pixel MV accuracy with no range restriction, and up to four MVs per MB. After encoding, transformed residuals (DCT), BCMs, MVs and the number of bits assigned to each MB of P-frames were extracted and the three features were computed as explained above.

In each frame, feature values at fixation points were selected as the test sample, while feature values at non-fixation points were selected as the control sample. Specifically, the control sample was obtained by applying a nonparametric bootstrap technique [36] to all non-fixation points of the video frame. Control points were sampled with replacement, multiple times, with sample size equal to the number of fixation points. The average of the feature values over all bootstrap (sub)samples was taken as the control sample mean.

The pairs of values (control sample mean, test sample mean) are shown in Fig. 5.2 for each frame as a green dot. The top scatter plot corresponds to MVE, the middle scatter plot to SRN and the bottom scatter plot to OBDL. From these plots, it is easy to see that, on average, MVE, SRN and OBDL values at fixation points tend to be higher than, respectively, MVE, SRN and OBDL values at randomly-selected non-fixation points. This suggests that MVE, SRN and OBDL could be used as indicators of possible fixations in video.

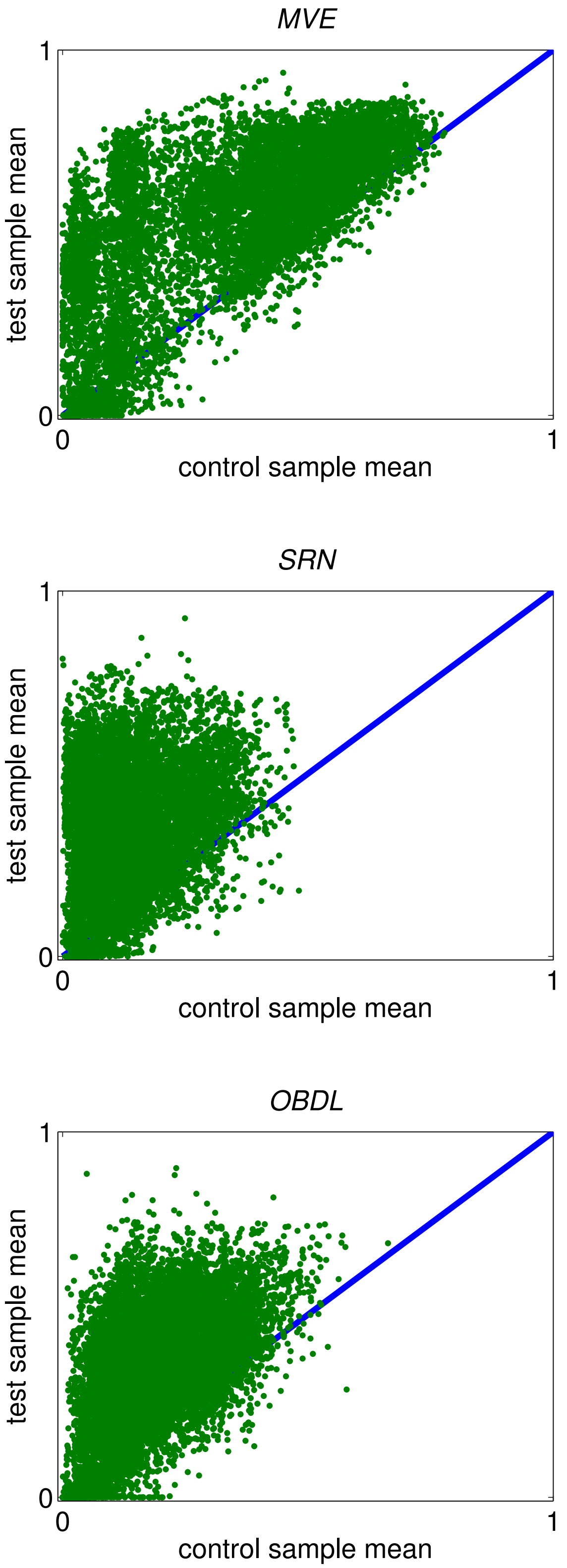


Figure 5.2: Scatter plots of the pairs (control sample mean, test sample mean) in each frame, for MVE (top), SRN (middle) and OBDL (bottom). Dots above the diagonal show that feature values at fixation points are higher than at randomly selected points.

To validate this hypothesis, we perform a two-sample t-test [90] using the control and test sample of each sequence. The null hypothesis was that the two samples originate in populations of the same mean. A separate test is performed for MVE, SRN and OBDL. This hypothesis was rejected by the two-sample t-test, at the 1% significance level, for all sequences and all three features. The p-values obtained for each video sequence are listed in Table 5.1, along with the percentage of frames where the test sample mean is greater than the control sample mean. Note that the p-values in most cases are very low, indicating low overall risk of rejecting the null hypothesis. However, the percentage columns show that, while the test sample mean is higher than the control sample mean in most frames of most sequences, there are also sequences (e.g., *Foreman*, *abb*, *ai*, *pnb*) with a significant percentage of frames where this is not true.

Overall, these results lend strong support to the assertion that MVE, SRN and OBDL are compressed-domain correlates of fixations in natural video. In the next chapter, we describe two simple approaches to visual saliency estimation using the proposed features, and then proceed to compare the proposed approaches against several state-of-the-art visual saliency models.

Table 5.1: Results of statistical comparison of test and control samples. For each sequence, the p-value of a two-sample t-test and the percentage (%) of frames where the test sample mean is larger than the control sample mean are shown.

		MVE		SRN		OBDL	
#	**Seq.**	**p**	%	**p**	%	**p**	%
1	*Bus*	10^{-66}	100	10^{-114}	99	10^{-112}	99
2	*City*	10^{-8}	72	10^{-51}	84	10^{-16}	73
3	*Crew*	10^{-50}	90	10^{-66}	93	10^{-29}	83
4	*Foreman*	10^{-26}	76	10^{-14}	58	10^{-10}	56
5	*Garden*	10^{-20}	83	10^{-53}	88	10^{-52}	90
6	*Hall*	10^{-223}	97	10^{-222}	95	10^{-211}	96
7	*Harbour*	10^{-55}	88	10^{-130}	100	10^{-83}	98
8	*Mobile*	10^{-66}	94	10^{-86}	91	10^{-58}	88
9	*Mother*	10^{-181}	99	10^{-168}	100	10^{-120}	100
10	*Soccer*	10^{-69}	97	10^{-93}	97	10^{-68}	94
11	*Stefan*	10^{-48}	100	10^{-69}	100	10^{-53}	98
12	*Tempete*	10^{-5}	63	10^{-54}	97	10^{-31}	89
13	*abb*	10^{-24}	66	10^{-66}	75	10^{-36}	76
14	*abl*	10^{-119}	93	10^{-144}	96	10^{-86}	92
15	*ai*	10^{-24}	65	10^{-59}	78	10^{-37}	73
16	*aic*	10^{-Inf}	100	10^{-262}	100	10^{-192}	100
17	*ail*	10^{-216}	98	10^{-199}	98	10^{-162}	98
18	*blicb*	10^{-55}	96	10^{-100}	98	10^{-79}	96
19	*bws*	10^{-70}	99	10^{-98}	95	10^{-82}	98
20	*ds*	10^{-49}	95	10^{-46}	84	10^{-50}	92
21	*hp6t*	10^{-31}	92	10^{-62}	97	10^{-42}	84
22	*mg*	10^{-62}	97	10^{-73}	97	10^{-49}	91
23	*mtnin*	10^{-116}	98	10^{-177}	97	10^{-87}	83
24	*ntbr*	10^{-25}	96	10^{-40}	93	10^{-57}	98
25	*nim*	10^{-38}	89	10^{-30}	68	10^{-16}	68
26	*os*	10^{-112}	95	10^{-141}	100	10^{-123}	97
27	*pas*	10^{-78}	94	10^{-92}	98	10^{-112}	99
28	*pnb*	10^{-2}	39	10^{-12}	55	10^{-12}	56
29	*ss*	10^{-129}	100	10^{-177}	99	10^{-135}	96
30	*swff*	10^{-9}	68	10^{-25}	62	10^{-16}	60
31	*tucf*	10^{-8}	85	10^{-48}	92	10^{-22}	86
32	*ufci*	10^{-13}	83	10^{-33}	94	10^{-7}	65

Chapter 6

Proposed Saliency Estimation Algorithms

The three compressed-domain features identified in Chapter 5 as visual correlates of fixations in video suggest that a simple saliency estimate may be obtained without fully reconstructing the video. In this chapter, we present two new visual saliency models for compressed video that have higher accuracy in predicting fixations compared to state-of-the-art models, even the pixel-domain ones. One model, called MVE+SRN, is built upon MVE and SRN, as its name suggests, while another model, called OBDL-MRF, uses only the OBDL feature.

6.1 MVE+SRN Saliency Estimation Model

Fig. 6.1 shows the block diagram of the proposed MVE+SRN algorithm to estimate visual saliency. For each inter-coded frame, motion vectors (MVs), block coding modes (BCMs) and transformed residuals (DCT) are entropy-decoded from the video bitstream. MVs and BCMs are used to construct the Motion Vector Entropy (MVE) map, illustrated in the left branch in Fig. 6.1 for a frame from sequence *Stefan*. Transformed residuals are used to construct the Smoothed Residual Norm (SRN) map, shown in the right branch in Fig. 6.1. Operations performed by various processing blocks were described when the corresponding features were introduced in Sections 5.1.1 and 5.1.2. The final saliency map is obtained by fusing the two feature maps.

A number of feature fusion methods have been investigated in the context of saliency estimation [66, 73]. The appropriate fusion method will depend on whether the features in question are independent, and whether their mutual action reinforces or diminishes saliency.

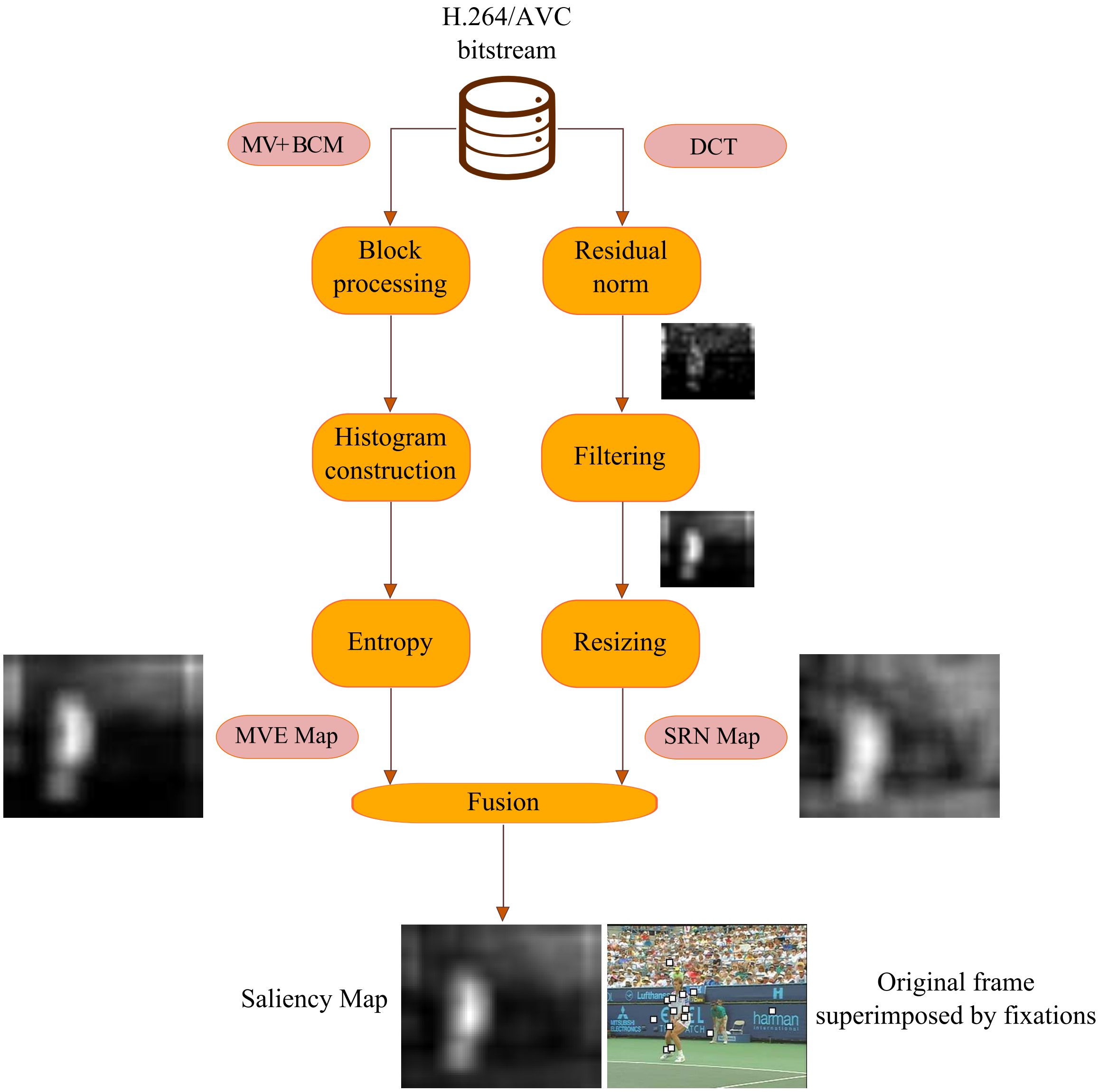


Figure 6.1: Block diagram of the proposed MVE+SRN saliency estimation algorithm.

In our case, we note that MVE and SRN are somewhat independent, in the sense that one could imagine a region in the scene with high MVE and low SRN, and vice versa. Also, their combined action is likely to increase saliency – when both MVE and SRN are large, the region is not only likely to contain moving objects (large MVE), but also contains parts that are surprising and not easily predictable from previous frames (large SRN). Hence, our fusion involves both additive and multiplicative combination of MVE and SRN maps,

$$\mathcal{S} = \mathcal{N}\left(\mathcal{S}_{MVE} + \mathcal{S}_{SRN} + \mathcal{S}_{MVE} \odot \mathcal{S}_{SRN}\right), \tag{6.1}$$

where the symbol $\odot$ denotes pointwise multiplication and $\mathcal{N}(\cdot)$ indicates normalization to the range $[0, 1]$.

6.2 OBDL-MRF Saliency Estimation Model

The next proposed method measures visual saliency based on a Markov Random Field (MRF) model of Operational Block Description Length (OBDL) feature responses.

6.2.1 MRF Model

While video compression algorithms are very sophisticated estimators of local information content, they only produce *local* information estimates, since all the processing is spatially and temporally localized to the macroblock (MB) unit. On the other hand, saliency has both a local and a global component. For example, many saliency models implement inhibition of return mechanisms [73], which suppress the saliency of image locations in the neighborhood of a saliency peak. To account for these effects, we rely on a MRF model [156].

More specifically, the saliency detection problem is formulated as one of inferring the *maximum a posteriori* (MAP) solution of a Spatio-Temporal Markov Random Field (ST-MRF) model. This is defined with respect to a binary classification problem, where salient blocks of 16×16 pixels belong to class 1 and non-salient blocks to class 0. The goal is to determine the class labels $\omega^t \in \{0, 1\}$ of the blocks of frame t, given the labels $\omega^{1 \cdots t-1}$ of the previous frames, and all previously observed compressed information $o^{1 \cdots t}$. The optimal label assignment ω^t_* is that which maximizes the posterior probability $P(\omega^t | \omega^{1 \cdots t-1}, o^{1 \cdots t})$. By application of Bayes rule this can be written as

$$\begin{aligned} P(\omega^t|\omega^{1\cdots t-1}, o^{1\cdots t}) &\propto P(\omega^{1\cdots t-1}|\omega^t, o^{1\cdots t}) \cdot P(\omega^t|o^{1\cdots t}) \\ &\propto P(\omega^{1\cdots t-1}|\omega^t, o^{1\cdots t}) \cdot P(o^{1\cdots t}|\omega^t) \cdot P(\omega^t), \end{aligned} \tag{6.2}$$

where $\propto$ denotes equality up to a normalization constant. Considering the monotonicity of the logarithm, the MAP solution for the saliency labels ω^t is then given by

$$\omega^t_* = \underset{\psi\in\Omega^t}{\arg\min} \left\{ -\log P(\omega^{1\cdots t-1}|\psi, o^{1\cdots t}) - \log P(o^{1\cdots t}|\psi) - \log P(\psi) \right\}, \tag{6.3}$$

where Ω^t denotes the set of all possible labeling configurations for frame t. From the Hammersley-Clifford theorem [19], the probabilities in (6.3) can be expressed as Gibbs distributions $P(x) = \frac{1}{Z}\exp\frac{-E(x)}{C}$ where $E(x)$ is an energy function, C a constant, sometimes referred to as "temperature," and Z a partition function. This enables the reformulation of the MAP estimation problem as

$$\omega^t_* = \underset{\psi\in\Omega^t}{\arg\min} \left\{ \frac{1}{C_t}E(\psi;\omega^{1\cdots t-1}, o^{1\cdots t}) + \frac{1}{C_o}E(\psi;o^{1\cdots t}) + \frac{1}{C_c}E(\psi) \right\}. \tag{6.4}$$

The components $E(\psi;\omega^{1\cdots t-1}, o^{1\cdots t})$, $E(\psi;o^{1\cdots t})$, and $E(\psi)$ of the energy function, respectively, measure the degree of *temporal* consistency of the saliency labels, the *coherence* between labels and feature observations, and the *spatial compactness* of the label field. A more precise definition of these three components is given in the following sections. Finally, the minimization problem (6.4) is solved by the method of Iterated Conditional Modes (ICM) [20] (cf. Section 6.2.5).

6.2.2 Temporal Consistency

Given a block at image location $\mathbf{n} = (x, y)$ of frame t, the spatio-temporal neighborhood $N(\mathbf{n})$ is defined as the set of blocks $\mathbf{m} = (x', y', t')$ such that $|x - x'| \le 1$, $|y - y'| \le 1$ and $t - L < t' < t$ for some L. The temporal consistency of the label field is measured locally, using

$$E(\psi;\omega^{1\cdots t-1}, o^{1\cdots t}) = \sum_{\mathbf{n}} E_t(\mathbf{n}), \tag{6.5}$$

where $E_t(\mathbf{n})$ is a measure of inconsistency within $N(\mathbf{n})$, which penalizes temporally inconsistent label assignments, i.e., $\omega^t(x, y) \ne \omega^{t'}(x', y')$.

The saliency label $\omega(\mathbf{m})$ of block $\mathbf{m}$ is assumed to be Bernoulli distributed with parameter proportional to the strength of features $o(\mathbf{m})$, i.e. $P(\omega(\mathbf{m})) = o(\mathbf{m})^{\omega(\mathbf{m})} \cdot (1 -$

$o(\mathbf{m}))^{1-\omega(\mathbf{m})}$. It follows that the probability $b(\mathbf{n}, \mathbf{m})$ that block $\mathbf{m}$ will bind with block $\mathbf{n}$ (i.e. have label $\psi(\mathbf{n})$) is

$$b(\mathbf{n}, \mathbf{m}) = o(\mathbf{m})^{\psi(\mathbf{n})} \cdot (1 - o(\mathbf{m}))^{1-\psi(\mathbf{n})}. \tag{6.6}$$

The consistency measure weights this probability by a similarity function, based on a Gaussian function of the distance between $\mathbf{n}$ and $\mathbf{m}$,

$$d(\mathbf{n}, \mathbf{m}) \propto \exp\left(\frac{-||\mathbf{m} - \mathbf{n}||_2^s}{2\sigma_s^2}\right) \cdot \exp\left(\frac{-||\mathbf{m} - \mathbf{n}||_2^t}{2\sigma_t^2}\right), \tag{6.7}$$

where $||\cdot||_2^s$ and $||\cdot||_2^t$ are the Euclidean distances along the spatial and temporal dimension, respectively, and σ_s^2, σ_t^2 two normalization parameters. The expected consistency between the two locations is then

$$c(\mathbf{n}, \mathbf{m}) = \frac{b(\mathbf{n}, \mathbf{m}) \cdot d(\mathbf{n}, \mathbf{m})}{\sum_{\mathbf{p} \in N(\mathbf{n})} (b(\mathbf{n}, \mathbf{p}) \cdot d(\mathbf{n}, \mathbf{p}))}. \tag{6.8}$$

This determines a prior expectation for the consistency of the labels, based on the observed features $o(\mathbf{m})$. The energy function then penalizes inconsistent labelings, proportionally to this prior expectation of consistency

$$E_t(\mathbf{n}) = \sum_{\mathbf{m} \in N(\mathbf{n})} c(\mathbf{n}, \mathbf{m}) \cdot (1 - \omega(\mathbf{m}))^{\psi(\mathbf{n})} \cdot \omega(\mathbf{m})^{1-\psi(\mathbf{n})}. \tag{6.9}$$

Note that $E_t(\mathbf{n})$ ranges from 0 to 1, taking the value 0 when all neighboring blocks $\mathbf{m} \in N(\mathbf{n})$ have the same label as block $\mathbf{n}$, and the value 1 when neighboring blocks all have label different than $\psi(\mathbf{n})$.

6.2.3 Observation Coherence

The incoherence between the observation and label fields at time t is measured with an energy function $E(\psi; o^{1\cdots t})$. While this supports the dependence of w^t on all prior observations ($o^{1\cdots t-1}$), we assume that the current labels are dependent only on the current observations (o^t). Incoherence is then measured by the energy function

$$E(\psi; o^{1\cdots t}) = \sum_{\mathbf{n}} \left(\inf_{\mathbf{m}} o(\mathbf{m})\right)^{1-\psi(\mathbf{n})} \cdot \left(1 - \sup_{\mathbf{m}} o(\mathbf{m})\right)^{\psi(\mathbf{n})}, \tag{6.10}$$

where infimum inf and supremum sup are defined over $\mathbf{m} = (x', y')$ such that $|x - x'| \leq 1$, $|y - y'| \leq 1$. This is again in $[0, 1]$ and penalizes the labeling of block $\mathbf{n}$ as non-salient, i.e., $\psi(\mathbf{n}) = 0$ when the infimum of feature value $\inf_{\mathbf{m}} o(\mathbf{m})$ is large, or as salient, i.e., $\psi(\mathbf{n}) = 1$, when the supremum of feature value $\sup_{\mathbf{m}} o(\mathbf{m})$ is small.

6.2.4 Compactness

In general, the probability of a block being labeled salient should increase if many of its neighbors are salient. The last energy component in (6.4) encourages this type of behavior. It is defined as

$$E(\psi) = \sum_{\mathbf{n}} \Phi(\mathbf{n})^{1-\psi(\mathbf{n})} \cdot (1-\Phi(\mathbf{n}))^{\psi(\mathbf{n})}, \tag{6.11}$$

where $\Phi(\mathbf{n})$ is a measure of saliency in the neighborhood of $\mathbf{n}$. This is defined as

$$\Phi(\mathbf{n}) = \alpha \sum_{\mathbf{m}\in N^{+}(\mathbf{n})} \psi(\mathbf{m}) + \beta \sum_{\mathbf{m}\in N^{\times}(\mathbf{n})} \psi(\mathbf{m}), \tag{6.12}$$

where $N^{+}(\mathbf{n})$ and $N^{\times}(\mathbf{n})$ are, respectively, the first-order (North, South, East, and West) and the second-order (North-East, North-West, South-East, and South-West) neighborhoods of block $\mathbf{n}$. In our experiments, we set $\alpha = \frac{1}{6}$ and $\beta = \frac{1}{12}$, to give higher weight to first-order neighbors.

6.2.5 Optimization

The solution of (6.4) can be found with many numerical procedures. Two popular methods are Stochastic Relaxation (SR) [49] and ICM [20]. SR has been reported to have some advantage in accuracy over ICM, but at a higher computational cost [149]. In this work, we adopt ICM, mainly due to its simplicity. The label of each block is initialized according to the corresponding feature value, $o(\mathbf{n})$, i.e. the block is labeled salient if $o(\mathbf{n}) > 0.5$ and non-salient otherwise. Each block is then relabeled with the label (0 or 1) that produces the largest reduction in the energy function. This relabeling is iterated until no further energy reduction is possible. We limit the iterations to eight in our experiment. It is worth mentioning that ICM is prone to getting trapped in local minima and the results are dependent on the initial labeling.

6.2.6 Final Saliency Map

The procedure above produces the most probable, a posteriori, map of salient block labels. To emphasize the locations with higher probability of attracting attention, the OBDL of a block declared salient (non-salient) by the MRF is increased (decreased) according to the OBDLs in its neighborhood. The process is formulated as

$$\mathcal{S}(\mathbf{n}) = \left(\sup_{\mathbf{m}} \{o(\mathbf{m}) \cdot d(\mathbf{m},\mathbf{n})\}\right)^{\psi(\mathbf{n})} \cdot \left(1 - \sup_{\mathbf{m}} \{(1-o(\mathbf{m})) \cdot d(\mathbf{m},\mathbf{n})\}\right)^{1-\psi(\mathbf{n})}, \tag{6.13}$$

where $\mathbf{m} = (x', y', t')$ is defined as the set of blocks such that $|x - x'| \leq 1$, $|y - y'| \leq 1$ and $t - L < t' \leq t$, and $d(\mathbf{m}, \mathbf{n})$ as in (6.7). In this way, a block $\mathbf{n}$ labeled as salient by the MRF inference is assigned a saliency equal to the largest feature value within its neighborhood, weighted by its distance from $\mathbf{n}$. On the other hand, for a block $\mathbf{n}$ declared as non-salient, this operation is applied to the complement of the saliency values within $N_{\mathbf{n}}$. The complement of this value is then assigned as the saliency value of $\mathbf{n}$.

6.3 Experiments

This section presents experimental evaluation of the proposed saliency estimation algorithms and their comparison with several state of the art saliency models.

6.3.1 Experimental setup

The proposed algorithms were compared with a number of state-of-the-art algorithms for saliency estimation in video. These methods are listed in Table 6.1. For each algorithm, the target domain - pixel (pxl) or compressed (cmp) - and the implementation details are also indicated (cf. Section 4.2.1 for more implementation details of cmp-domain algorithms.) Encoding was done using FFMPEG library [4] (version 2.3) with QP $\in \{3, 6, ..., 51\}$ in the baseline profile, with default Group-of-Pictures (GOP) structure. For each MB, there exists up to four MVs having 1/4-pixel accuracy with no range restriction. We avoided the use of I- and B-frames since many of the compressed-domain methods did not specify how to handle these frames. In principle, there are several possibilities for B-frames, such as flipping forward MVs around the frame to create backward MVs (as in P-frames), or forming two saliency maps, one from forward MVs and one from backward MVs, and then averaging them. However, to avoid speculation and stay true to the algorithms the way they were presented, we used only P-frames in the evaluation.

6.3.2 Results

A set of experiments was performed to compare the proposed algorithms to state-of-the-art saliency algorithms. These experiments used quantization parameter QP = 30, i.e. reasonably good video quality - average peak signal-to-noise (PSNR) across encoded sequences of 35.8 dB. Fig. 6.2 illustrates the differences between the saliency predictions

Table 6.1: Saliency estimation algorithms used in our evaluation. D: target domain (cmp: compressed; pxl: pixel); I: Implementation (M: Matlab; P: Matlab p-code; C: C/C++; E: Executable).

#	Algorithm	First Author	Year	D	I
1	**MaxNorm**	Itti [73] (ilab.usc.edu/toolkit)	1998	pxl	C
2	**Fancy1**	Itti [66] (ilab.usc.edu/toolkit)	2004	pxl	C
3	**SURP**	Itti [69] (ilab.usc.edu/toolkit)	2006	pxl	C
4	**GBVS**	Harel [59] (DIOFM channel)	2007	pxl	M
5	**STSD**	Seo [141]	2009	pxl	M
6	**SORM**	Kim [88]	2011	pxl	E
7	**AWS**	Diaz [48]	2012	pxl	P
8	**PMES**	Ma [108]	2001	cmp	M
9	**MAM**	Ma [109]	2002	cmp	M
10	**PIM-ZEN**	Agarwal [11]	2003	cmp	M
11	**PIM-MCS**	Sinha [143]	2004	cmp	M
12	**MCSDM**	Liu [104]	2009	cmp	M
13	**MSM-SM**	Muthuswamy [126]	2013	cmp	M
14	**PNSP-CS**	Fang [43]	2014	cmp	M

of various algorithms for a few sample video frames. In the figure, the motion vector field (MVF), Intra-Observer (IO) and the raw OBDL are also shown for the corresponding frame.

Figs. 6.3 and 6.4 show the average AUC′ and NSS′ score, respectively, of various algorithms across the test sequences. Not surprisingly, on average, pixel-domain methods perform better than compressed-domain ones. However, our proposed compressed-domain methods, MVE+SRN and OBDL-MRF, top all other methods, including pixel-domain ones, on both metrics. Based on these results, while MVE+SRN is the best saliency predictor, OBDL-MRF achieves very close prediction accuracy. Also note that the SRN feature, by itself, has a close prediction capabilities to MVE+SRN. Including the MVE feature into saliency prediction helps on sequences with considerable amount of motion, such as *harry-potter-6-trailer*, but in many cases, SRN is sufficient. Another interesting point is that raw OBDL feature, by itself, achieves comparable scores to state-of-the-art, while incorporating spatial (OBDL-S) and temporal (OBDL-T) filtering as well as MRF inference results in improved predictions, with best results produced by the full-blown OBDL-MRF.

The performance of the various saliency models was also evaluated using a multiple comparison test [63] similar to what we did in Section 4.2.2. For each sequence, the average score of a given model across all frames is computed, along with the 95% confidence interval for the average score. The model with the highest average score is a top performer on that

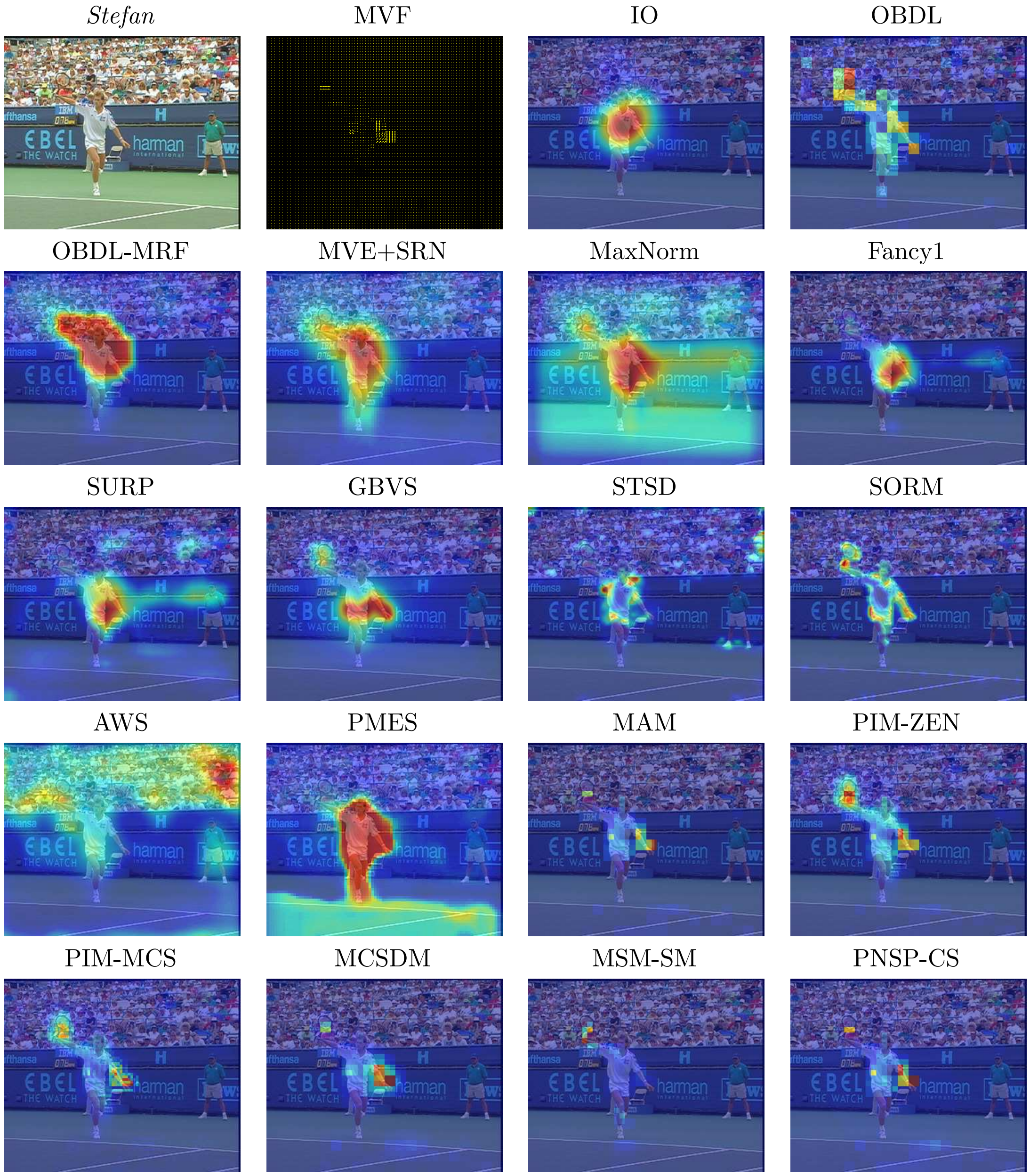
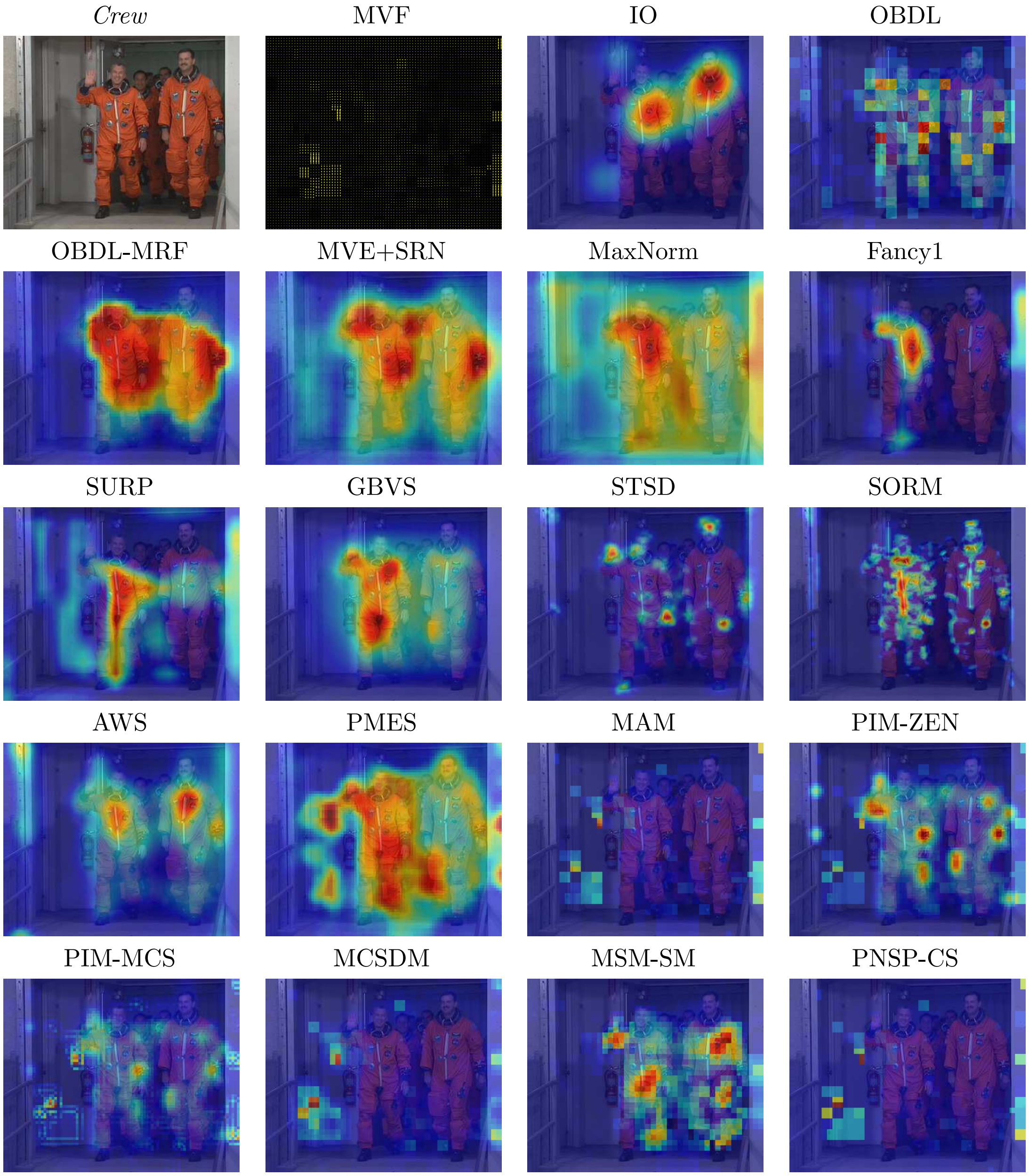


Figure 6.2: Sample saliency maps obtained by various algorithms.



Crew MVF IO OBDL

OBDL-MRF MVE+SRN MaxNorm Fancy1

SURP GBVS STSD SORM

AWS PMES MAM PIM-ZEN

PIM-MCS MCSDM MSM-SM PNSP-CS

Figure 6.2: Sample saliency maps obtained by various algorithms.*(cont.)*

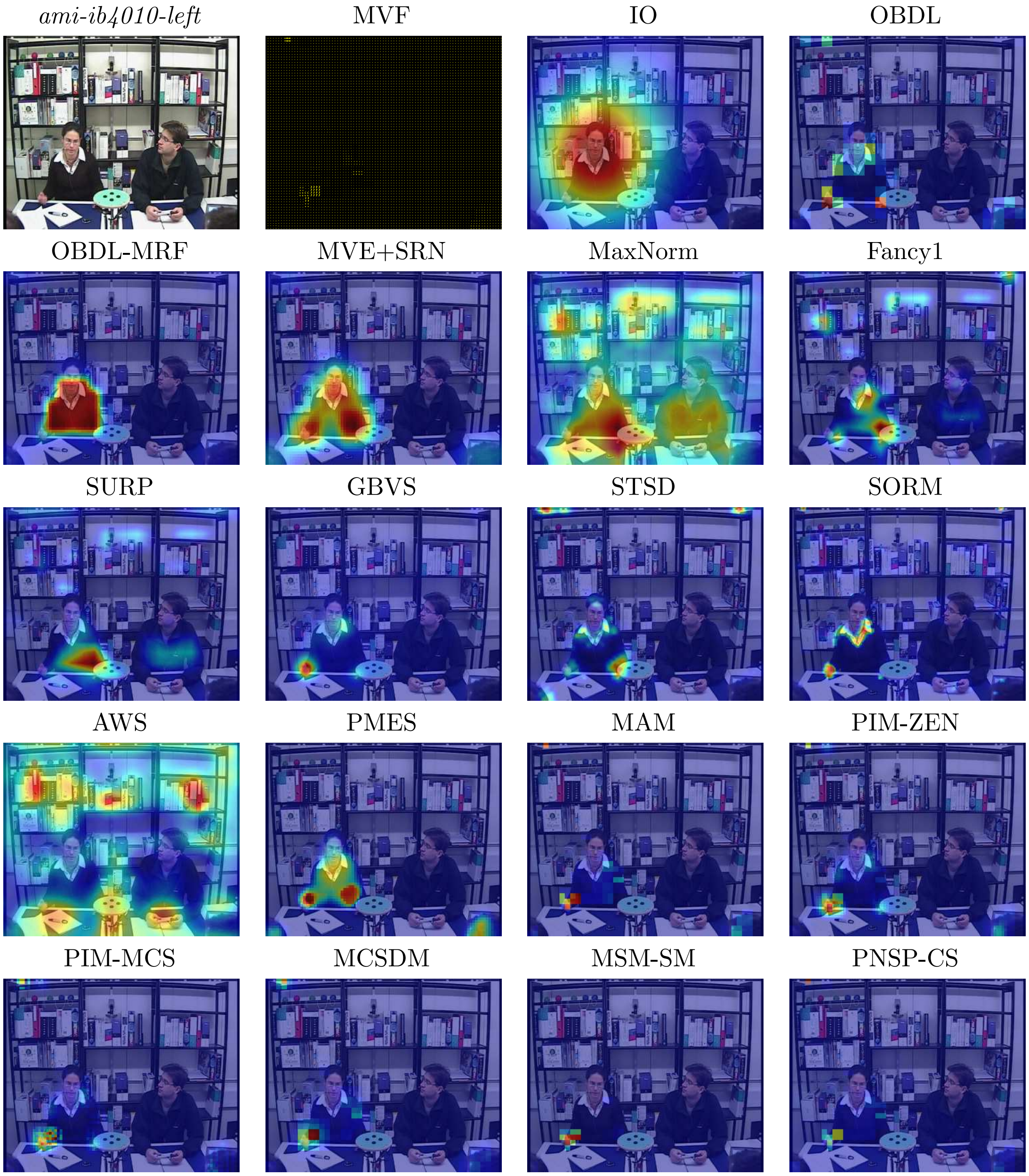
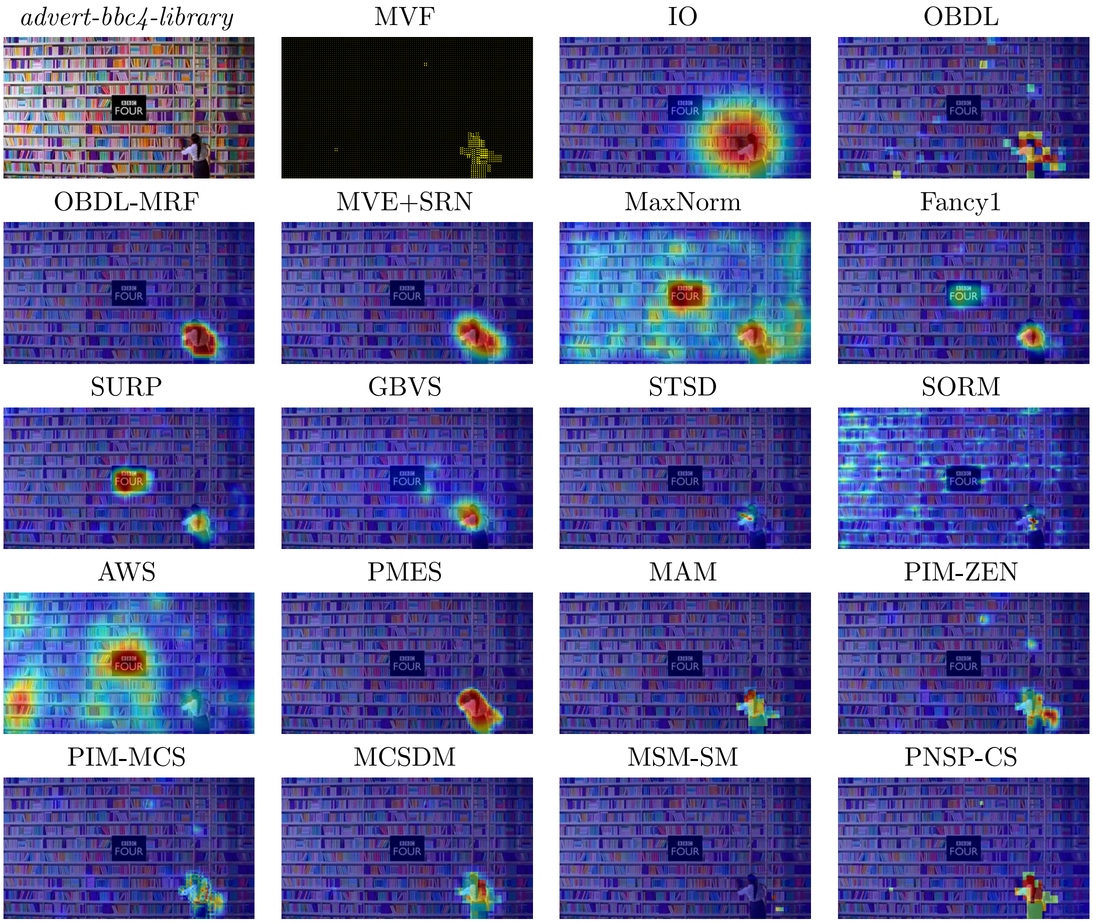


Figure 6.2: Sample saliency maps obtained by various algorithms.*(cont.)*



advert-bbc4-library

MVF

IO

OBDL

OBDL-MRF

MVE+SRN

MaxNorm

Fancy1

SURP

GBVS

STSD

SORM

AWS

PMES

MAM

PIM-ZEN

PIM-MCS

MCSDM

MSM-SM

PNSP-CS

Figure 6.2: Sample saliency maps obtained by various algorithms.*(cont.)*

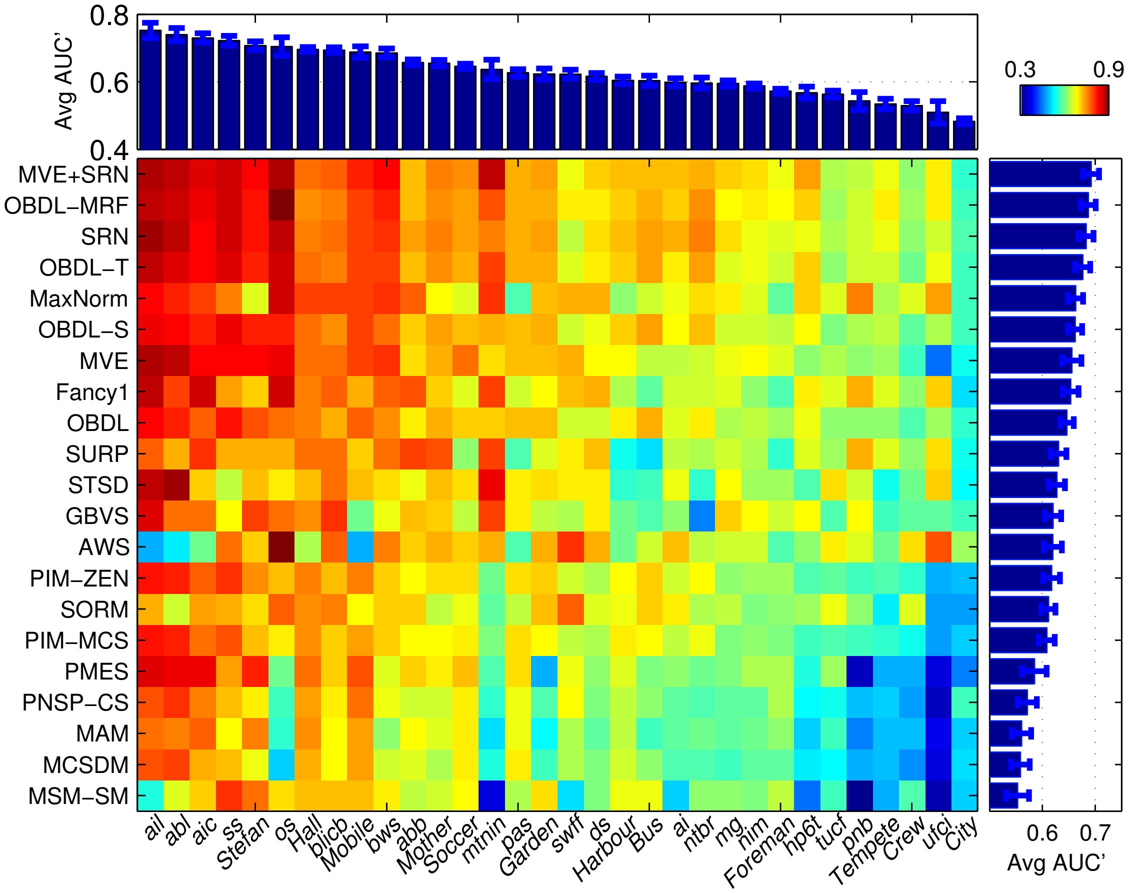


Figure 6.3: Accuracy of various saliency algorithms over the two datasets according to AUC′. Each 2-D color map shows the average AUC′ score of each algorithm on each sequence. The average AUC′ performance across sequences/algorithms shown in the sidebar/topbar. Error bars represent standard error of the mean.

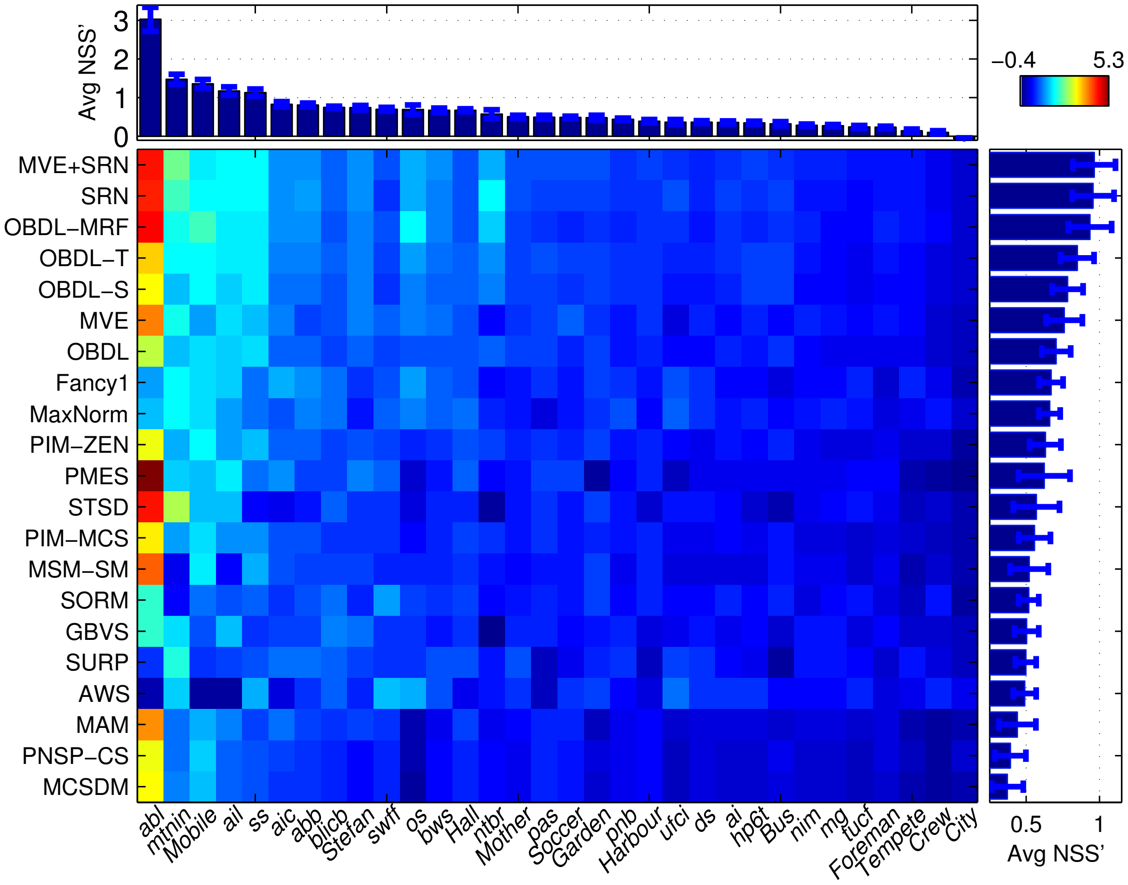


Figure 6.4: Accuracy of various saliency algorithms over the two datasets according to NSS′.

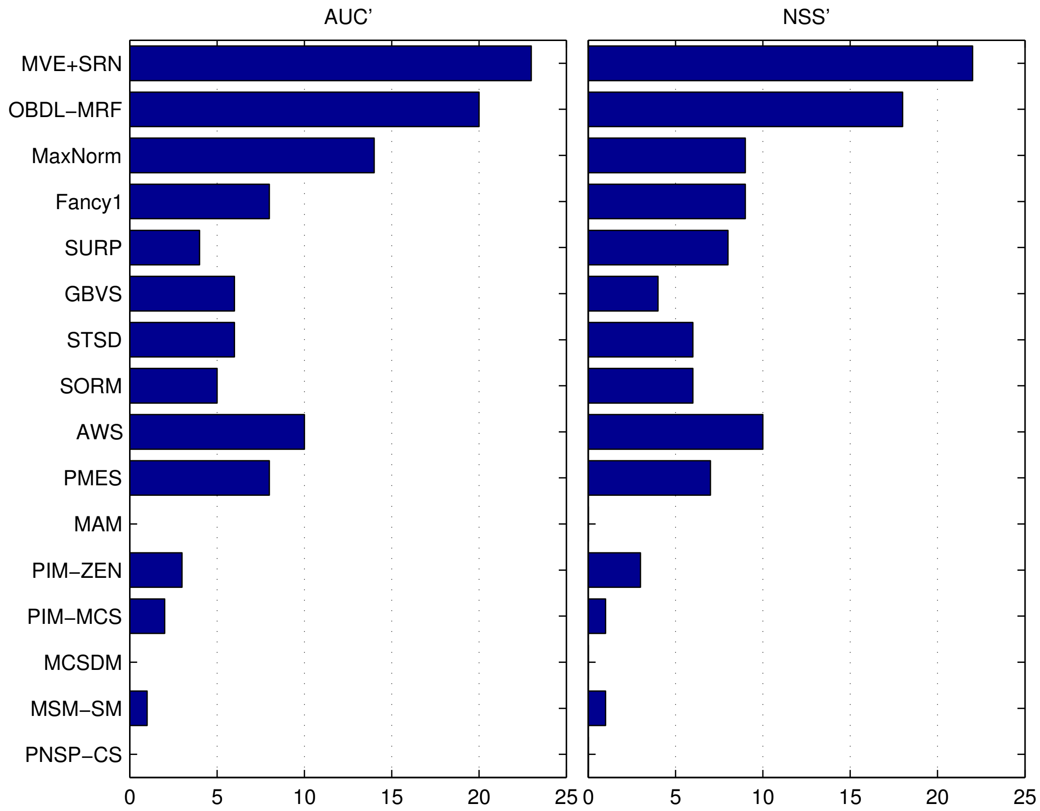


Figure 6.5: The number of appearances among top performers, using AUC′ and NSS′ evaluation metrics.

sequence, however, all other models whose 95% confidence interval overlaps that of the highest-scoring model are also considered top performers on that sequence. The number of appearances among top performers for each model is shown in Fig. 6.5. Again, pixel-domain methods tend to do better than compressed-domain ones, but our methods top both groups. As before, MVE+SNR comes in first in terms of both AUC′ and NSS′, while OBDL-MRF is second.

It is worth mentioning that due to the very weak performance of AWS on sequences where most other methods scored well in Figs. 6.3 and 6.4 (such as *advert-bbc4-library*, *ami-ib4010-left* and *Mobile*), the average AUC′ and NSS′ scores of AWS are not particularly high - the average was dragged down by the low scores on these few sequences. However, in the multiple comparison test in Fig. 6.5, AWS shows strong performance, because it is among top performers on many other sequences. These results corroborate the results of the comparison made in [22] (which was performed on DIEM and CRCNS [69],[70]) where AWS was among the highest-scoring algorithms under study.

We also compare the distribution of saliency values at the fixation locations against the distribution of saliency values at random points from non-fixation locations in Fig. 6.6 in terms of JSD. If these two distributions overlap substantially, then the saliency model predicts fixation points no better than a random guess. On the other hand, as one distribution diverges from the other, the saliency model is better able to predict fixation points. As seen in the Fig. 6.6, MVE+SRN and OBDL-MRF, respectively, generate higher divergence between two distributions compared to other models.

As discussed in Section 4.2.2, the quality of the encoded video, measured in terms of PSNR, drops as QP increases due to the larger amount of compression. Fig. 6.7 shows how the average AUC$'$ and NSS$'$ scores change as a function of the average PSNR (across sequences), by varying QP $\in \{3, 6, ..., 51\}$. Note that pixel-domain methods are also sensitive to compression - they do not use compressed-domain information, but they are impacted by the accuracy of decoded pixel values. Our MVE+SRN and OBDL-MRF achieve their best performance at PSNR around 35 dB. This is excellent news for our proposed models because this range of PSNR is thought to be very appropriate in terms of balance between objective quality and compression efficiency. Overall, MVE+SRN achieves the highest accuracy followed by OBDL-MRF across most of the compression range.

Based on the results in Fig. 6.7, it appears that around the PSNR value of 35 dB, compressed-domain features MVE, SRN and OBDL are all sufficiently informative and sufficiently accurate. As the amount of compression reduces (i.e., quality increases) SRN becomes less informative, since small quantization step-size makes each residual have large ℓ_0 norm. At the same time, MVs may become too noisy, since rate-distortion optimization does not impose sufficient constraints for smoothness. On the other hand, as the amount of compression increases (i.e., quality reduces), both MVE and SRN become less accurate. Both extremes are detrimental to saliency prediction. Since the OBDL feature incorportes the number of bits needed for both transformed prediction residuals and MVs, it is not surprising that the accuracy of OBDL-MRF degrades substantially at the extremes of the compression range. While at low rates there are too few bits to enable a precise measurement of saliency, at high rates there are too many bits available, and all blocks become salient according to this feature.

To assess the complexity of various algorithms, processing time was measured on an Intel (R) Core (TM) i7 CPU at 3.40 GHz and 16 GB RAM running 64-bit Windows 8.1

Table 6.2: Average processing time (ms) per frame.

Algorithm	AWS	GBVS	PNSP-CS	MAM	PMES	SURP	STSD	Fancy1	SORM	MaxNorm	PIM-ZEN	**OBDL-MRF**	**MVE+SRN**	MCSDM	PIM-MCS	MSM-SM
Time	1492	923	895	778	579	323	227	98	92	89	43	**39**	**30**	15	10	8

implemented as indicated in Table 6.1. The results are shown in Table 6.2. As expected, compressed-domain models tend to require far less processing time than their pixel-domain counterparts. The proposed methods, MVE+SRN and OBDL-MRF, implemented in MATLAB, required, respectively, an average of 30 ms and 39 ms per CIF video frame. While this is slower than some of the other compressed-domain algorithms, it enables the computation of saliency within or close to the real-time requirements, even in MATLAB. Note that the decoding time is not included in these results. Recall that OBDL-MRF only requires entropy decoding to get data necessary to process, whereas other compressed-domain algorithms, including MVE+SRN, require additional decoding effort.

6.4 Discussion and conclusions

Using the compressed-domain features in Chapter 5, we constructed two simple visual saliency estimation methods and compared them with fourteen other saliency prediction methods for video. Some of these methods also made use of compressed-domain features, while others operated in the pixel domain. Comparison was made using a number of established metrics. The results showed that both proposed methods outperformed all other methods, including pixel-domain ones.

A natural question to ask is - *how come a compressed-domain method, which seems to be restricted to a relatively constrained set of data, can outperform pixel-domain methods in terms of saliency prediction?* To answer this question, one needs to realize that a compressed-domain method is not a stand-alone entity. Its front end is the video encoder, an extremely sophisticated algorithm whose goal is to provide the most compact representation of video. Surely, such compact representation contains useful information about various aspects of the video signal, including saliency.

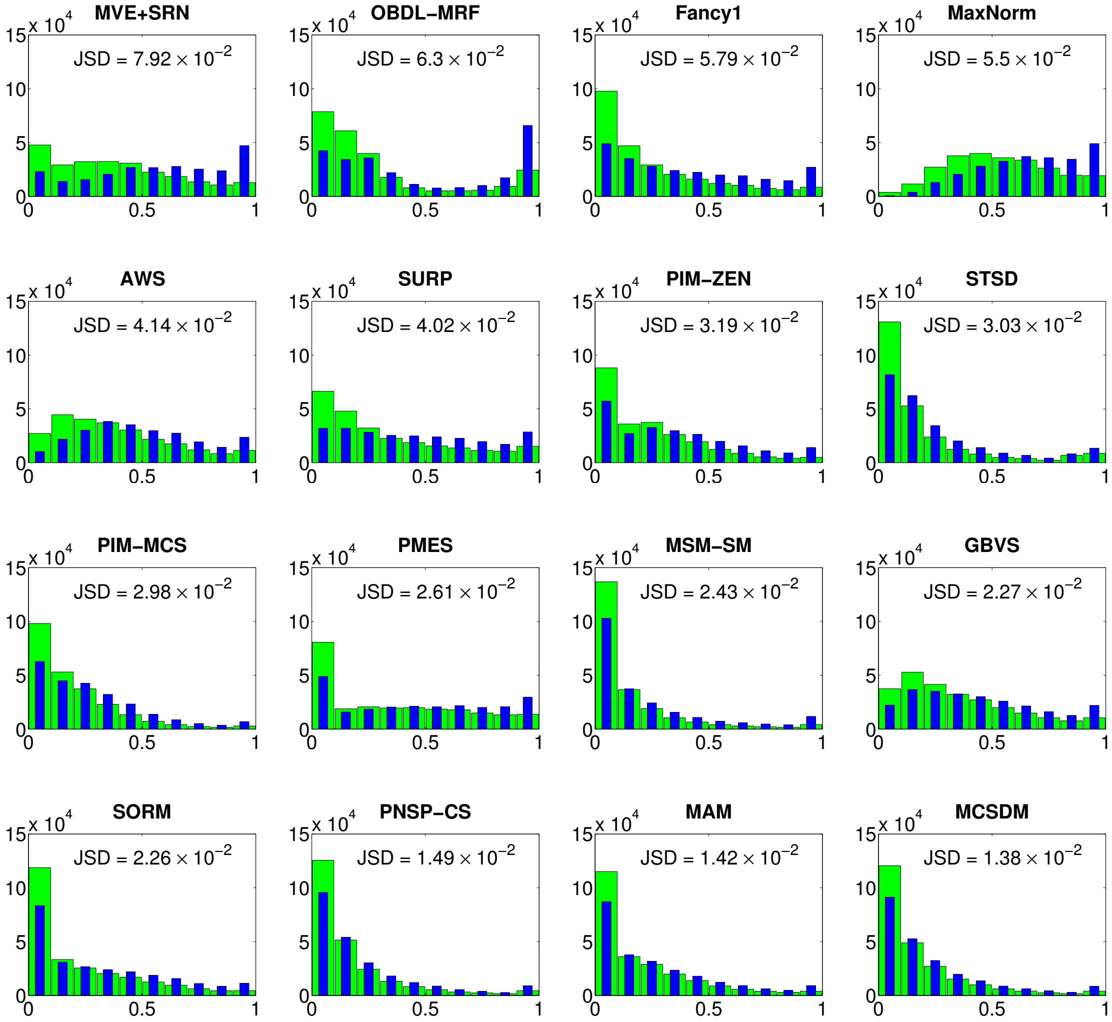


Figure 6.6: The frequencies of saliency values estimated by different algorithms at the fixation locations (narrow blue bars) and random points from non-fixation locations (wide green bars) vs the number of human fixations. The JSD between two distribution corresponding to each algorithm presents how large each distribution diverges from another (histograms were sorted from left to right and top to bottom according to the JSD metric.)

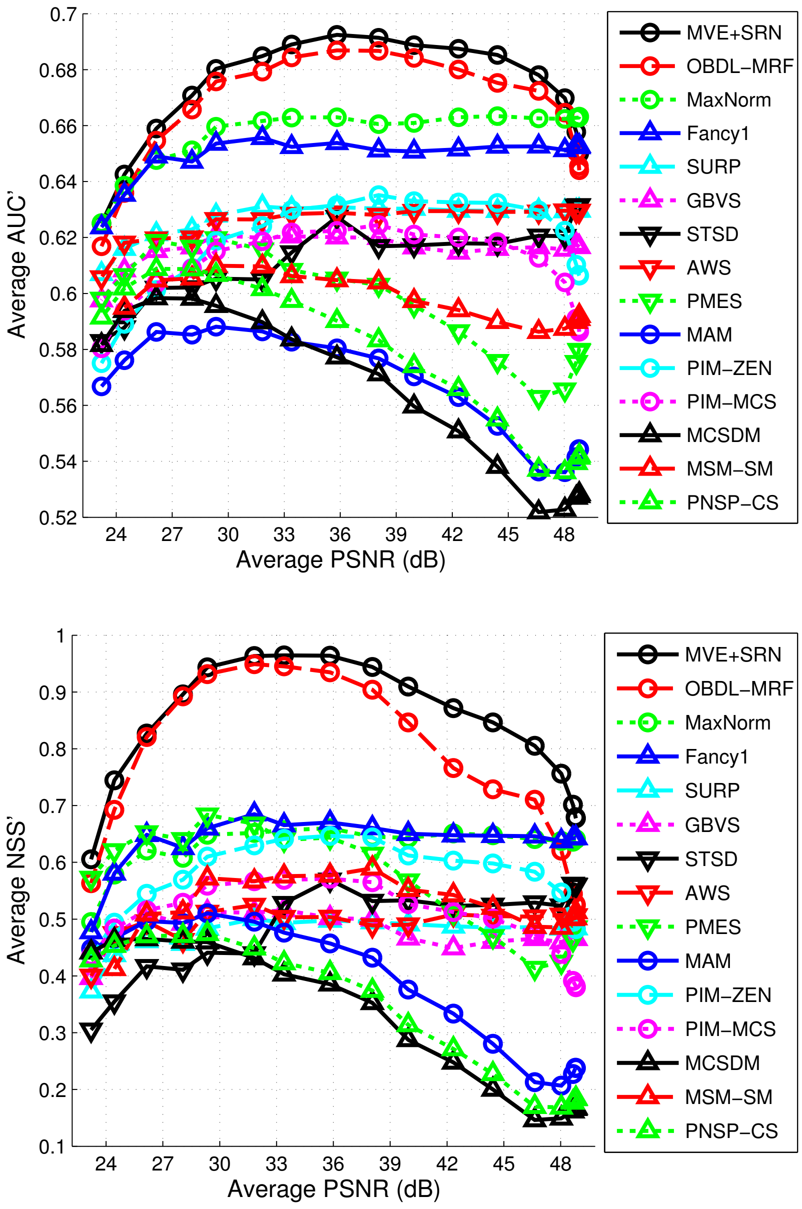


Figure 6.7: The relationship between the average PSNR and the models' accuracy.

Chapter 7

Compressed-Domain Tracking

This chapter presents our approach for region tracking in H.264/AVC-compressed video, which could be used to enhance salient region analysis and, more generally, scene interpretation, in the compressed domain. It is well known that motion has a strong impact on saliency in dynamic scenes [120, 111, 68]. Hence, a number of saliency estimation methods have been proposed that rely on temporal changes [108, 109, 161, 104], as discussed in Chapter 4. These methods estimate saliency in each frame using motion information, extracted from the compressed bitstream, to decrease complexity. In video, the viewer's attention usually remains on a salient region for several consecutive frames. For example, [62] showed that *fixation duration* ranges from 100 ms to 400 ms, mostly around 200 ms, which equals 5 frames for a 25 fps video, or about 7 frames for 30 fps video. Fixation duration is the period of time when the eyes fixate on a single location at the center of the gaze. Note that fixation and attention are tightly inter-linked in normal viewing, although one is able to change their attention independently of where the eyes are fixated [39]. Although the relationship between visual saliency and tracking is an interesting one, in this chapter we develop a general algorithm for region tracking that could be used independently of saliency modeling. The algorithm was first presented in [81].

We focus on tracking a single moving region in the compressed domain using a Spatio-Temporal Markov Random Field (ST-MRF) model. A ST-MRF model naturally integrates the spatial and temporal aspects of the region's motion. Built upon such a model, the proposed method uses only the motion vectors (MVs) and block coding modes (BCMs) from the compressed bitstream to perform tracking. First, the MVs are pre-processed through intra-coded block motion approximation and global motion compensation (GMC).

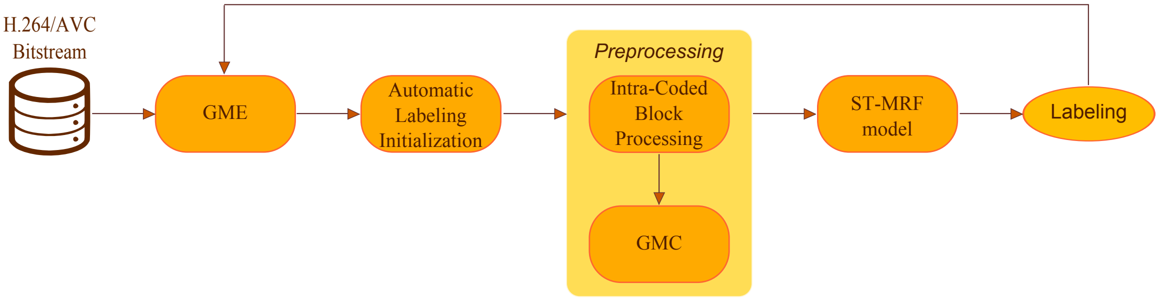


Figure 7.1: The flowchart of our proposed moving object tracking.

At each frame, the decision of whether a particular block belongs to the region being tracked is made with the help of the ST-MRF model, which is updated from frame to frame in order to follow the changes in the region's motion.

The flowchart of our proposed moving region tracking is shown in Fig. 7.1. In each frame, the proposed method first approximates MVs of intra-coded blocks, estimates global motion (GM) parameters, and removes GM from the motion vector field (MVF). The estimated GM parameters are used to initialize a rough position of the region in the current frame by projecting its previous position into the current frame. Eventually, the procedure of Iterated Conditional Modes (ICM) updates and refines the predicted position according to spatial and temporal coherence under the *maximum a posteriori* (MAP) criterion, defined by the ST-MRF model.

7.1 MRF-Based Tracking

A moving rigid object/region is generally characterized by spatial compactness (i.e., not dispersed across different parts of the frame), relative similarity of motion within the region, and a continuous motion trajectory. Our ST-MRF model is based on rigid object/region motion characteristics. We treat moving region tracking as a problem of inferring the MAP solution of the ST-MRF model. More specifically, we consider the frame to be divided into small blocks (4×4 in our experiments). Region blocks will be labeled 1, non-region blocks 0. We want to infer the block labels $\omega^t \in \{0, 1\}$ in frame t, given the labels ω^{t-1} in frame $t-1$, and the observed motion information $o^t = \left(\mathbf{v}^t, \kappa^t\right)$. Here, the motion information o^t consists of the MVs from the compressed bitstream, denoted $\mathbf{v}^t(\mathbf{n})$, and the BCM and partition size $\kappa^t(\mathbf{n})$, where $\mathbf{n} = (x, y)$ indicates the position of the block within the frame. The criterion for choosing "the best" labeling ω_*^t is that it should maximize the posterior probability $P\left(\omega^t|\omega^{t-1}, o^t\right)$. Similar to (6.2) and (6.3), this problem is solved by a Bayesian framework, where we express the posterior probability in terms of the inter-frame likelihood $P\left(\omega^{t-1}|\omega^t, o^t\right)$, the intra-frame likelihood $P\left(o^t|\omega^t\right)$, and the a priori probability $P\left(\omega^t\right)$ as follows

$$\omega_*^t = \arg\max_{\psi\in\Omega}\left\{P(\omega^{t-1}|\psi, o^t)\cdot P(o^t|\psi)\cdot P(\psi)\right\}, \tag{7.1}$$

where Ω denotes the set of all possible labeling configurations for frame t. The solution to the maximization problem in (7.1) is the same as the solution to the following minimization problem

$$\omega_*^t = \arg\min_{\psi\in\Omega}\left\{-\log P(\omega^{t-1}|\psi, o^t) - \log P(o^t|\psi) - \log P(\psi)\right\} \tag{7.2}$$

According to the Hammersley-Clifford theorem [19], the probabilities in (7.2) can be expressed as Gibbs distributions of the form $\frac{1}{Z}\exp\frac{-E(x)}{C}$ for some energy function $E(x)$, partition function Z and normalizing constant C. Hence, we write

$$P(\omega^{t-1}|\psi, o^t) = \frac{1}{Z_\Gamma}\exp\left\{-\frac{1}{C_\Gamma}E(\psi;\omega^{t-1}, o^t)\right\} \tag{7.3}$$

$$P(o^t|\psi) = \frac{1}{Z_\Lambda}\exp\left\{-\frac{1}{C_\Lambda}E(\psi;o^t)\right\} \tag{7.4}$$

$$P(\psi) = \frac{1}{Z_\Phi}\exp\left\{-\frac{1}{C_\Phi}E(\psi)\right\} \tag{7.5}$$

In the above equations, the three energy functions $E(\psi;\omega^{t-1}, o^t)$, $E(\psi;o^t)$, and $E(\psi)$ represent the degree of inconsistency in temporal continuity, spatial context coherence, and compactness, respectively. The parameters C_Γ, C_Λ and C_Φ are scaling constants. In our model, each of the three energy functions E is expressed as the summation of the corresponding block-wise energy terms ξ over the object blocks, so that the optimization problem becomes

$$\omega_*^t = \arg\min_{\psi\in\Omega}\left\{\frac{1}{C_\Gamma}\sum_{\mathbf{n}:\psi(\mathbf{n})=1}\xi(\mathbf{n};\omega^{t-1}, o^t) + \frac{1}{C_\Lambda}\sum_{\mathbf{n}:\psi(\mathbf{n})=1}\xi(\mathbf{n};o^t) + \frac{1}{C_\Phi}\sum_{\mathbf{n}:\psi(\mathbf{n})=1}\xi(\mathbf{n})\right\}. \tag{7.6}$$

The first term measures the temporal discontinuity of labeling between consecutive frames - the larger the difference between the labeling ω^{t-1} and the backwards-projected candidate labeling ψ, the larger this term will be. The second term represents spatial incoherence among region's MVs - the larger the difference among the MVs within the labeled region under the candidate labeling ψ, the larger this term will be. The compactness of region's shape is accounted for in the last term. All three terms will be defined more precisely in the following sections. Finally, the minimization problem (7.6) will be solved by the method of ICM [20].

7.1.1 Temporal Continuity

Temporal continuity is measured by the overlap between the labeling of the previous frame, ω^{t-1}, and the backwards-projected candidate labeling ψ for the current frame. Consider a block $\mathbf{n}$ in the current frame that is assigned to the object by the candidate labeling ψ, i.e. $\psi(\mathbf{n}) = 1$. The block is projected backwards into the previous frame along its MV, $\mathbf{v}^t(\mathbf{n}) = (v_x(\mathbf{n}), v_y(\mathbf{n}))$, and the degree of overlap $\gamma(\mathbf{n} + \mathbf{v}^t(\mathbf{n}), \omega^{t-1})$ is computed as the fraction of pixels within the projected block in the previous frame that carry the label 1 under the labeling ω^{t-1}. The energy term for block $\mathbf{n}$ is taken to be

$$\xi(\mathbf{n}; \omega^{t-1}, o^t) = -\gamma\left(\mathbf{n} + \mathbf{v}^t(\mathbf{n}), \omega^{t-1}\right). \tag{7.7}$$

Our experiments suggest that temporal continuity is a powerful cue for region tracking, and by itself is able to provide relatively accurate tracking in many cases. It is also relatively simple to compute. However, its performance may suffer due to noisy or inaccurate MVs, especially near region boundaries. Hence, a robust tracking algorithm should not rely exclusively on temporal continuity, and should incorporate some spatial properties of the MVF. Two such concepts, corresponding to the second and third terms in (7.6), are discussed next.

7.1.2 Context Coherence

One of the characteristics of rigid object/region motion in natural video is the relative consistency (coherence) of the MVs belonging to the object. This is true even if the motion is not pure translation, so long as the frame rate is not too low. Such motion coherence has frequently been used in compressed-domain segmentation [27, 119, 28, 163, 103]. A popular approach is to model the MVs within a region with an independent bivariate Gaussian distribution whose parameters are estimated from the decoded MVs. Once the parameter estimates are obtained, one can adjust the segmentation by testing how well the MVs fit into the assumed model. A particular problem with parameter estimation in this context is the presence of outliers - incorrect or noisy MVs that are often found in flat-texture regions or near region boundaries. For small regions, even a few outliers can lead to large estimation errors. Sample variance is especially sensitive to outliers [137]. To resolve the problem we employ robust statistics methods [137], which tend to be more resistant to the effect of outliers compared to classical statistics.

A number of robust statistics methods have been proposed to estimate central tendency of the data in the presence of outliers, e.g., Median, Trimmed Mean, Median Absolute Deviation, and Inter Quartile Range [137, 155, 136]. We have tested these methods in the context of our problem, and finally settled on a Modified Trimmed Mean method, as it gave the best results. The main difference between our proposed Modified Trimmed Mean and the conventional Trimmed Mean [137] is that instead of truncating a fixed percentage of the data set from one or both ends of the distribution, in our method the outliers are adaptively recognized and truncated, as explained below.

For the purpose of MV coherence analysis, the region's motion is represented by a single representative MV $\widehat{\mathbf{v}}$. This vector is computed based on the Polar Vector Median (cf. Section 7.2.1) of MVs that are assigned to the region by the candidate labeling ψ, i.e. $\psi(\mathbf{n}) = 1$. At this stage, we are using preprocessed MVs, $\mathbf{v}'(\mathbf{n})$ (cf. Section 7.2). The deviation of a MV of the block $\mathbf{n}$ from the region's representative vector $\widehat{\mathbf{v}}$ is computed as the Euclidean distance between $\mathbf{v}'(\mathbf{n})$ and $\widehat{\mathbf{v}}$:

$$d(\mathbf{n}) = \left\|\mathbf{v}'(\mathbf{n}) - \widehat{\mathbf{v}}\right\|_2 . \tag{7.8}$$

It is observed that the deviation $d(\mathbf{n})$ for the blocks belonging to the object can be modeled reasonably well by the rectified (non-negative) Gaussian distribution, except for the outliers. Thus, we identify the outliers in the data by checking whether $d(\mathbf{n}) > \tau$, where τ is computed as

$$\tau = \max\left\{2 \cdot \sigma_d, 1\right\}, \tag{7.9}$$

and σ_d is the sample standard deviation of $d(\mathbf{n})$ assuming the average value of zero. The distribution of $d(\mathbf{n})$ for the rotating ball object in frame #4 of the *Mobile Calendar* sequence is shown in Fig. 7.2 (top). In this example, the candidate labeling ψ is initially predicted by projecting the previous frame labeling, i.e. ψ^3, via current GM parameters. Two MVs with $d(\mathbf{n}) \cong 24.7$ and one with $d(\mathbf{n}) \cong 29$ are recognized as the outliers based on the threshold defined in (7.9). The fitted rectified Gaussian distributions before and after outlier removal

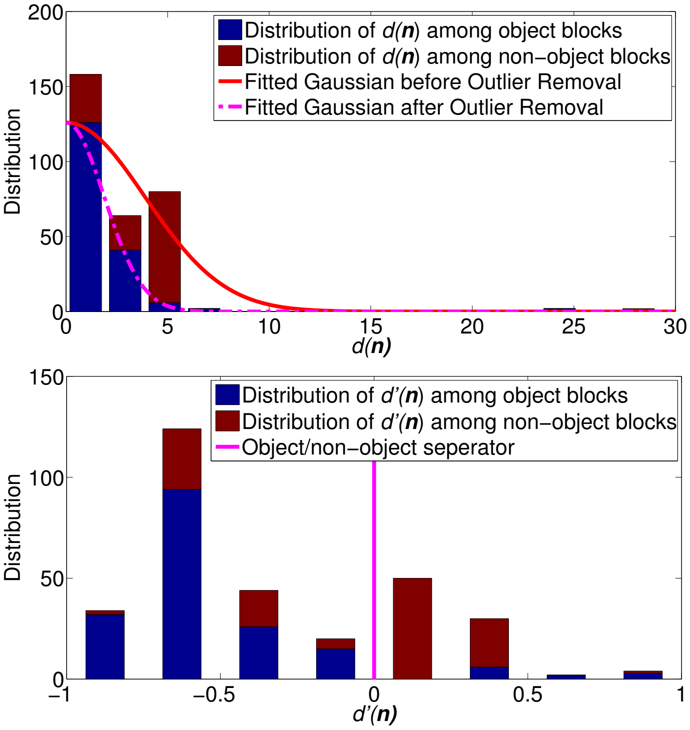


Figure 7.2: Distribution of $d(\mathbf{n})$ and $d'(\mathbf{n})$ for the ball in frame #4 of the *Mobile Calendar* sequence.

are also shown. It can be seen that outliers significantly increase the sample variance and that after their removal, the rectified Gaussian can be used as a reasonable model for $d(\mathbf{n})$.

After removing the outliers, the sample standard deviation σ_d is recalculated. In certain cases, when the region's MVs are close to identical, the recalculated σ_d is close to zero. We clip σ_d from below to 0.5 in order to avoid problems with subsequent computations. After that, the MV deviation is normalized to the interval $[-1, 1]$ as follows

$$d'(\mathbf{n}) = \min\left\{\frac{d(\mathbf{n})/\sigma_d - 2}{2}, 1\right\}. \tag{7.10}$$

The higher the value of the normalized deviation $d'(\mathbf{n})$, the less similar is $\mathbf{v}'(\mathbf{n})$ to $\widehat{\mathbf{v}}$, so the less likely is block $\mathbf{n}$ to belong to the region being tracked, as far as MV similarity goes. Fig. 7.2 (bottom) shows the distribution of $d'(\mathbf{n})$.

The effect of GM is taken into account by dividing $d'(\mathbf{n})$ with a GM parameter ρ defined as

$$\rho = 2 - \exp\left(-c_1 \cdot (\rho_A + \rho_B)\right), \tag{7.11}$$

where

$$\rho_A = c_2 \cdot \left(|m_1| + |m_4|\right)^{c_3}, \tag{7.12}$$

$$\rho_B = |1 - m_2| + |m_3| + |m_5| + |1 - m_6|, \tag{7.13}$$

where $m_i, i = 1, 2, ..., 6$, are the six affine GM parameters (cf. Section 7.2.2) and c_1, c_2 and c_3 are constant values. The logic behind (7.11) is as follows: the larger the camera motion, the closer the value of ρ is to 2; the smaller the camera motion, the closer its value is to 1. Parameters m_1 and m_4 represent translation (value 0 means no translation), m_2 and m_6 represent zoom (value 1 means no zoom), and m_3 and m_5 represent rotation (value 0 means no rotation). Hence, as any component of the affine motion increases, the exponent in (7.11) becomes more negative, and the value of ρ gets closer to 2, which is the highest it can be. In the case of fast camera motion, MVs are less reliable, and the influence of context coherence on tracking decisions should be reduced. Hence, the block-wise context coherence energy term in (7.6) is computed by dividing the normalized MV deviation in (7.10) by ρ, that is

$$\xi(\mathbf{n}; o^t) = d'(n)/\rho. \tag{7.14}$$

7.1.3 Compactness

While there are certainly counterexamples to this observation, most rigid objects/regions in natural video tend to have compact shape, meaning that the chance of a block belonging to the region being tracked is increased if many of its neighbors are known to belong to the same region. We take this observation into account through the last term in (7.6). We employ the 8-adjacency neighborhood with different weights for the first-order and second-order neighbors. The block-wise energy term for compactness is computed by the weighted sum of labels in the neighborhood of the current block:

$$\xi(\mathbf{n}) = -\alpha \cdot \sum_{\mathbf{p} \in N^+(\mathbf{n})} \psi(\mathbf{p}) - \beta \cdot \sum_{\mathbf{p} \in N^\times(\mathbf{n})} \psi(\mathbf{p}), \tag{7.15}$$

where $N^+(\mathbf{n})$ and $N^\times(\mathbf{n})$ are, respectively, the first-order (North, South, East, and West) and the second-order (North-East, North-West, South-East, and South-West) neighborhoods of block $\mathbf{n}$. Recall from (7.6) that inside of the sum, $\psi(\mathbf{n}) = 1$. A neighboring block with label 0 will not change the value of (7.15), while a neighboring block with label 1 will make (7.15) more negative. Hence, the higher the number of neighboring blocks that belong to the region being tracked (label 1), the lower the value of the energy term in (7.15), meaning that the more likely it is that block $\mathbf{n}$ also belongs to that region. We set $\alpha = \frac{1}{6}$ and $\beta = \frac{1}{12}$ in our experiments to give higher weight to first-order neighbors.

7.1.4 Optimization

Now that each term in (7.6) is defined, the optimization problem needs to be solved. As discussed in Section 6.2.5, we use ICM to solve (7.6), mainly due to its simplicity. At the beginning, the label of each block is initialized by projecting the previous frame labeling ω^{t-1} into the current frame using the current GM parameters. After that, each block is relabeled with the label (0 or 1) that leads to the largest reduction in the energy function. This relabeling procedure is iterated until no further energy reduction is achieved. Usually, six iterations are enough to reach a local minimum.

7.2 Preprocessing

Our proposed tracking algorithm makes use of two types of information from the H.264/AVC-compressed bitstream: BCM (partition) information and MVs. Texture data does not need to be decoded in the proposed method. H.264/AVC defines four basic macroblock (MB) modes [159]: 16×16, 16×8, 8×16, and 8×8, where the 8×8 mode can be further split into 8×4, 4×8, and 4×4 modes. Since the smallest coding mode (partition) in H.264/AVC is 4×4, in order to have a uniformly sampled MVF, we map all MVs to 4×4 blocks. This is straightforward in inter-coded blocks, as well as SKIP blocks where the MV is simply set to zero. However, interpreting the motion in the intra-coded blocks is more involved. In this section, we describe two preprocessing steps that are employed before the actual ST-MRF optimization discussed in the previous section. These steps are the management of intra-coded blocks and eliminating GM.

7.2.1 Polar Vector Median for Intra-Coded Blocks

Intra-coded blocks have no associated MVs. However, for the purpose of running the ST-MRF optimization in the previous section, it is useful to assign MVs to these blocks. We propose to do this based on the neighboring MVs using a new method called Polar Vector Median (PVM). For this purpose, we employ MVs of the first-order neighboring MBs (North, West, South, and East) that are not intra-coded. Fig. 7.3 shows a sample

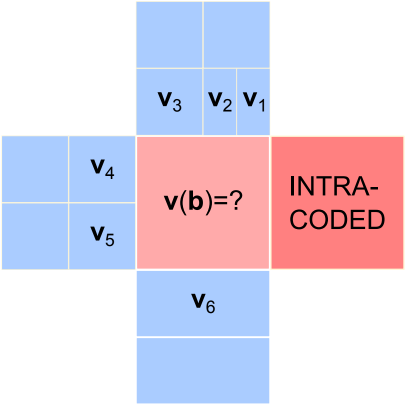


Figure 7.3: MV assignment for an intra-coded MB. One of the first-order neighboring MBs is also intra-coded, and the remaining neighbors have MVs assigned to variable size blocks.

intra-coded MB along with its first-order neighboring MBs. In this example, the goal is to find $\mathbf{v}(\mathbf{b})$ for all blocks $\mathbf{b}$ in the intra-coded MB. We collect MVs of the 4×4 blocks from the neighboring MBs that are closest to the current intra-coded MB and store them in the list V. For this example, the list of MVs is $V = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_4, \mathbf{v}_5, \mathbf{v}_5, \mathbf{v}_6, \mathbf{v}_6, \mathbf{v}_6, \mathbf{v}_6)$.

Note that $\mathbf{v}_6$ appears four times in the list, because it is assigned to four 4×4 blocks along the southern boundary of the intra-coded MB. For the same reason, $\mathbf{v}_3$, $\mathbf{v}_4$, and $\mathbf{v}_5$ appear twice, while $\mathbf{v}_1$ and $\mathbf{v}_2$ appear only once in the list. The list will contain at most 16 vectors, which happens when all first-order neighboring MBs are inter-coded. In the above example, one of the neighboring MBs is intra-coded, so the list contains only 12 vectors.

The next step is to assign a representative vector from this collection of vectors. The standard vector median [14] would be the vector from the list with the minimum total distance to other vectors in the list. The existence of outliers, however, has an adverse effect on the vector median. We therefore propose another method called PVM, which has proved less cost demanding and more robust in our study.

In PVM, the representative vector is computed in polar coordinates as follows. Let $V = (\mathbf{v}_i)_{i=1:n}$ be the list of n input vectors, sorted according to their angle (in radians) from $-\pi$ to $+\pi$. Then, a collection of $m = \lfloor (n+1)/2 \rfloor$ vectors is selected as $\widehat{V} = (\mathbf{v}_i)_{i=k:k+m-1}$, where index k is found as

$$k = \arg\min_j \sum_{i=j}^{j+m-2} \theta_i, \tag{7.16}$$

and θ_i denotes the angle between vectors $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ (let $\mathbf{v}_1 \equiv \mathbf{v}_{n+1}$). The new list $\widehat{V}$ contains approximately half the original number of vectors in V chosen such that the sum

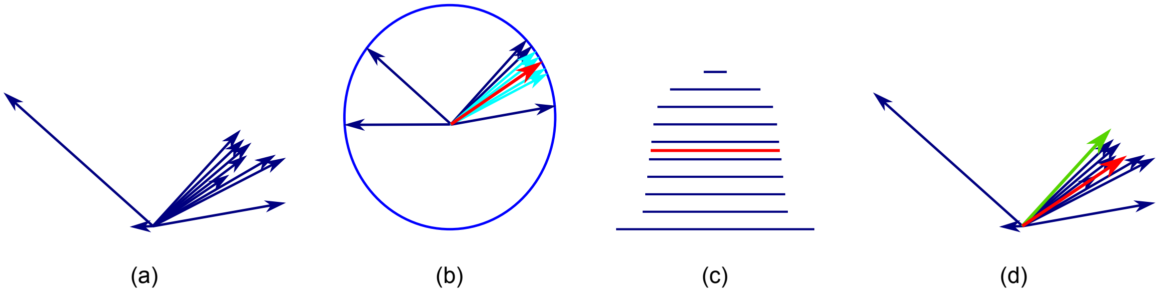


Figure 7.4: Polar Vector Median: (a) original vectors, (b) angles of vectors; cyan vectors: candidate vectors for computing representative angle, red vector: representative angle, (c) lengths of vectors, red line: representative length, (d) result of polar vector median, green vector: standard vector median [14], red vector: polar vector median.

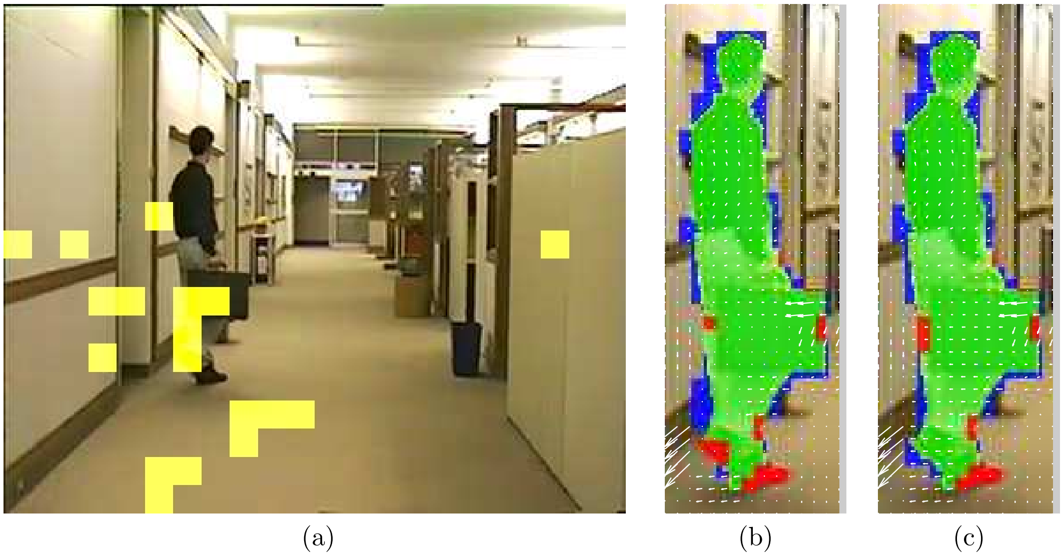
of the angles between them is minimum. Hence, they are clustered in a narrow beam. The PVM $\widehat{\mathbf{v}}$ is constructed as follows: its angle is chosen to be the median of angles of the vectors in $\widehat{V}$, while its magnitude is set to the median of magnitudes of the vectors in V, that is

$$\angle\widehat{\mathbf{v}} = \text{median}(\angle\mathbf{v}_i)_{i=k:k+m-1}, \tag{7.17}$$

$$\|\widehat{\mathbf{v}}\|_2 = \text{median}(\|\mathbf{v}_i\|_2)_{i=1:n}. \tag{7.18}$$

Fig. 7.4 shows an example of computing the PVM. It should be mentioned that zero-vectors are excluded from angle calculations, since their angle is indeterminate. Once PVM $\widehat{\mathbf{v}}$ is computed, it is assigned to all 4 × 4 blocks within the intra-coded MB (e.g., $\mathbf{v}(\mathbf{b})$ in Fig. 7.3).

Fig. 7.5 shows the effect of assigning PVM of neighboring blocks to intra-coded blocks on frame #35 of the *Hall Monitor* sequence. The intra-coded blocks are indicated in Fig. 7.5(a), the labeling with zero MV assignment to intra-coded MBs is shown in Fig. 7.5(b), while the labeling with PVM assignment to intra-coded MBs is shown in Fig. 7.5(c). When the block labeling is computed as discussed in Section 7.1, all pixels in a given block are assigned the label of that block. By comparing with the manually segmented ground truth, one can identify correctly labeled region pixels (true positives - TP), non-region pixels incorrectly labeled as region pixels (false positives - FP), and missed region pixels that are labeled as non-region pixels (false negatives - FN). The numbers are $TP = 3340$, $FP = 580$, $FN = 199$ for Fig. 7.5(c), where PVM is used, against $TP = 3313$, $FP = 591$, $FN = 226$ in Fig. 7.5(b), where zero-vector is used instead of the PVM. It is easy to see that detection



(a) (b) (c)

Figure 7.5: The effect of assigning PVM to intra-coded blocks: (a) intra-coded blocks indicated as yellow squares; (b) tracking result without PVM assignment, (c) tracking result with PVM assignment. TP are shown as green, FP as blue, FN as red.

is improved by PVM assignment around the man's feet in the bottom parts of Fig. 7.5(b) and Fig. 7.5(c).

7.2.2 Global Motion Compensation

Global motion (GM), caused by camera movement, affects all pixels in the frame. Since GM adds to the object's native motion, for accurate tracking, it is important to remove GM from the MV field prior to further processing. In this work we use the 6-parameter affine model [37] to represent GM. Although less flexible than the 8-parameter perspective model, the affine model has been widely used in the literature to represent GM, in part because there are fewer parameters to be estimated, which often leads to higher estimation accuracy. Given the parameters of the affine model, $[m_1, \ldots, m_6]$, a block centered at (x, y) in the current frame will be transformed to a quadrangle centered at (x', y') in the reference frame, where (x', y') is given by

$$x' = m_1 + m_2 x + m_3 y, \quad y' = m_4 + m_5 x + m_6 y. \tag{7.19}$$

The MV due to affine motion is given by

$$\mathbf{v}(x, y) = (x' - x, y' - y) \tag{7.20}$$

To estimate GM parameters $[m_1, \ldots, m_6]$, we use a modified version of the M-Estimator introduced in [13] by Arvanitidou *et al.*, which is an extension of [144]. This method reduces the influence of outliers in a re-weighted iteration procedure based on the estimation error obtained using least squares estimation. This iterative procedure continues until convergence. Chen *et al.* [29] showed that the performance of this approach depends on the tuning constant in weighting factor calculation. In [13], the weighting factor $w(\xi_i)$ for i-th MV, which imposes the strength of outlier suppression, is calculated as

$$w(\xi_i) = \begin{cases} \left(1 - \frac{\xi_i^2}{\tau^2 \cdot \mu_\xi^2}\right)^2 & \xi_i < \tau \cdot \mu_\xi \\ 0 & \xi_i \geq \tau \cdot \mu_\xi \end{cases} \tag{7.21}$$

where τ is the tuning constant, ξ_i is the estimation error calculated as the Manhattan norm between the observed MV and the MV obtained from the estimated model (7.19)-(7.20), and μ_ξ is the average estimation error over all the MVs in the frame. In our slightly modified approach, to obtain the decision boundary for outlier suppression (which is $\tau \cdot \mu_\xi$ in (7.21)), instead of using all the MVs in the frame, we choose only those MVs that are not currently declared as outliers. In particular, the weighting factor is calculated iteratively as

$$w(\xi_i) = \begin{cases} \left(1 - \frac{\xi_i^2}{(\mu_\xi + 2\sigma_\xi)^2}\right)^2 & \xi_i < \mu_\xi + 2\sigma_\xi \\ 0 & \xi_i \geq \mu_\xi + 2\sigma_\xi \end{cases} \tag{7.22}$$

where ξ is the set of estimation errors for MVs with a non-zero weighting factor after each iteration, and σ_ξ is the standard deviation of the new set ξ. Once the weighting factor for a MV becomes zero, it will be discarded from the set ξ in the following iterations. In this approach, the decision boundary for outlier suppression uses both the average value and standard deviation of estimation errors over the set ξ, which is more robust in comparison to the conventional approach where only the average value of estimation errors over all MVs is used.

Arvanitidou *et al.* [13] observed that large block modes are more likely to be part of the background, whereas small block modes, arising from splitting in the motion estimation procedure at the encoder, are more likely to belong to foreground moving objects. As an example, Fig. 7.6 shows small block modes (8×4, 4×8 and 4×4) indicated in red in a couple of frames from *Coastguard* and *Stefan* sequences. Hence, only MVs of large blocks (16×16, 16×8, 8×16 and 8×8) are used in global motion estimation (GME), while MVs of small blocks (8×4, 4×8 and 4×4) and intra-coded blocks are discarded from this

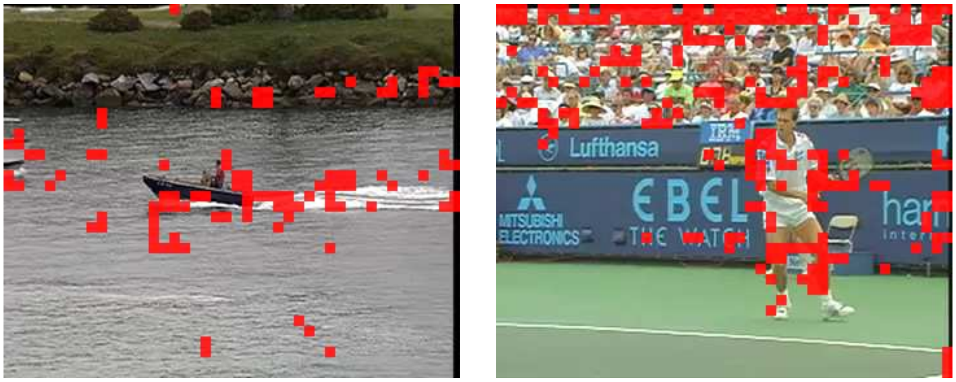


Figure 7.6: Small block modes (8 × 4, 4 × 8 and 4 × 4) of frame #2 for *Coastguard* and *Stefan* sequences.

process. In our approach, for the purpose of GME, we also discard the MVs from the region that was occupied by the object in the previous frame. To get fast convergence to a stable solution and escape from being trapped in many local minima, we initialize the weighting factor for i-th MV based on its dissimilarity to neighboring MVs, denoted by $\delta(\xi_i)$, which is computed as the median of its Manhattan differences from MVs of its neighbors. Therefore, the initial weight for i-th MV is computed by

$$w(\xi_i) = \exp\left(-\delta(\xi_i)\right). \tag{7.23}$$

7.3 Experiments

7.3.1 Experimental setup

A number of standard test sequences were used to evaluate the performance of our proposed approach. Sequences were in the YUV 4:2:0 format, at two resolutions, CIF (Common Intermediate Format, 352 × 288 pixels) and SIF (Source Input Format, 352 × 240 pixels), all at the frame rate of 30 fps. All sequences were encoded using the H.264/AVC JM v.18.0 encoder [5], at various bitrates, with the GOP (Group-of-Pictures) structure IPPP, i.e., the first frame is coded as intra (I), and the subsequent frames are coded predictively (P). Motion and partition information were extracted from the compressed bitstream, and MVs were remapped to 4 × 4 blocks, as explained in Section 7.2.

Some of the characteristics of the proposed approach are its robustness and stability. To show this, we use the same parameters throughout all the experiments, as listed in

Table 7.1: Parameter values used in our experiments

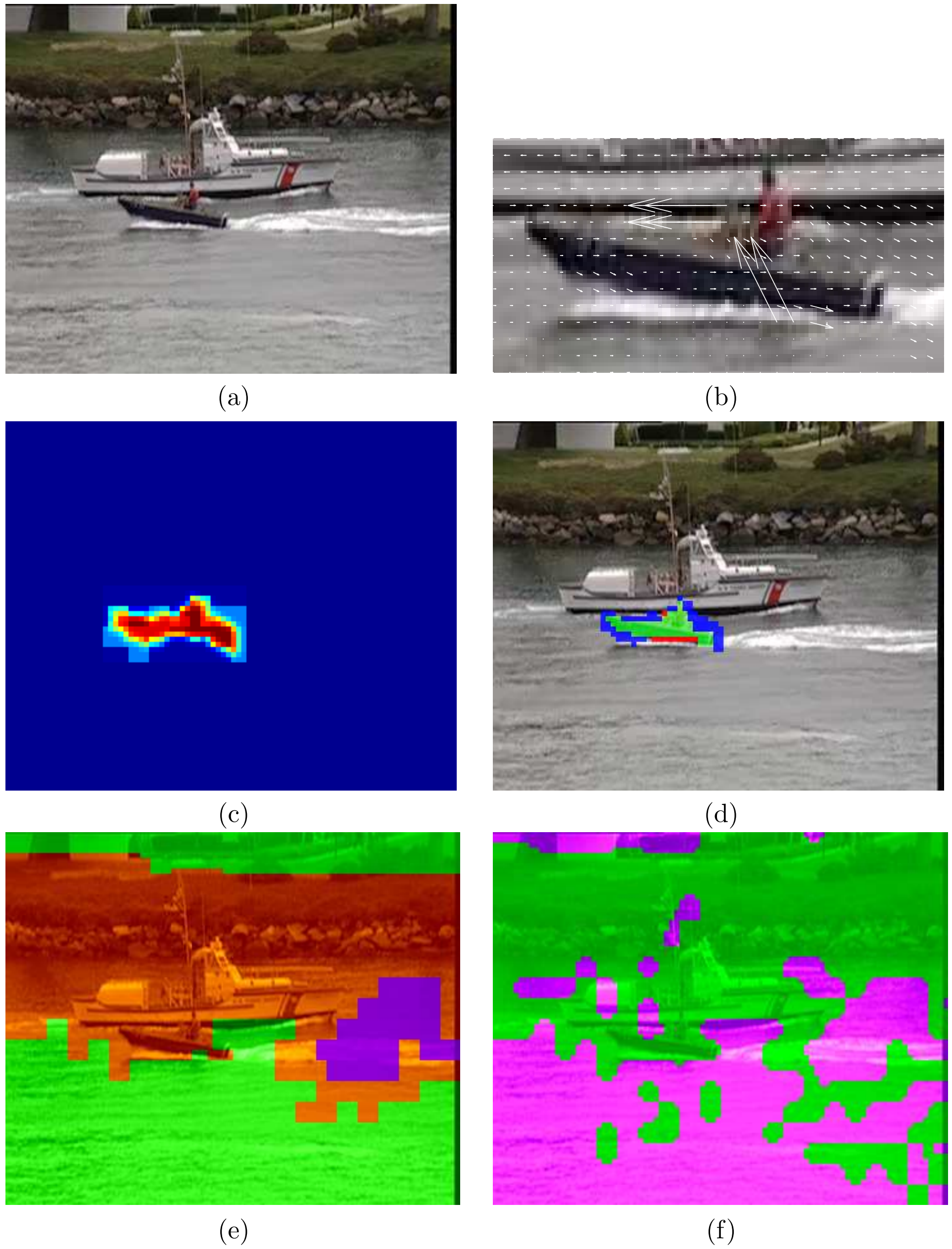
Parameter	C_Γ	C_Φ	C_Λ	c_1	c_2	c_3
Value	1	2/3	0.25	1/128	1	2

Table 7.1. We found that the average performance does not change much if some of these parameter values are changed, especially the parameters c_1, c_2 and c_3 that represent the effect of camera motion on energy function, and only affect a few frames.

7.3.2 Results

Fig. 7.7 illustrates a few intermediate results from the tracking process for a sample frame from *Coastguard*. As seen from Fig. 7.7(b), the MVF around the target (small boat) is somewhat erratic, due to fast camera motion in this part of the sequence. The proposed ST-MRF-based tracking algorithm computes the energy function for the chosen MRF model, which is shown in Fig. 7.7(c). Therefore, despite the erratic MVF, the target seems to be localized reasonably well. Fig. 7.7(d) shows the detected target region after the tracking process has been completed. Pixels shaded with different colors indicate TP, FP and FN, as explained in the caption of Fig. 7.5. As seen here, some erroneous decisions are made around the boundary of the target object, but the object interior is detected well. For comparison purposes, the segmentation result from two other methods, [119] and [163], are illustrated in Fig. 7.7(e) and Fig. 7.7(f), respectively. Clearly, the proposed method has produced a much better result compared to these two methods in the face of sudden and fast camera movement.

In Fig. 7.8 our proposed method is compared with the methods of Liu *et al.* [103], Chen *et al.* [119], and Zeng *et al.* [163] in terms of visual segmentation results on frame #97 of *Mobile Calendar*. The region being tracked is the ball. In frame #97, which corresponds to the moment when the ball touches the train and changes its direction, the blocks within the ball and the train have almost equal MVs. This may cause confusion in the tracking or segmentation task. As a consequence, the proposed method declares some parts of the train as the ball; nonetheless, the ball itself is detected correctly (Fig. 7.8(a)). The method from [103], on the other hand, misses a large part of the ball (Fig. 7.8(b)). It only detects a small part of the ball and misclassifies some parts of the train and the background as the target. Segmentation results of the methods from [119] (Fig. 7.8(c)) and [163] (Fig. 7.8(d))

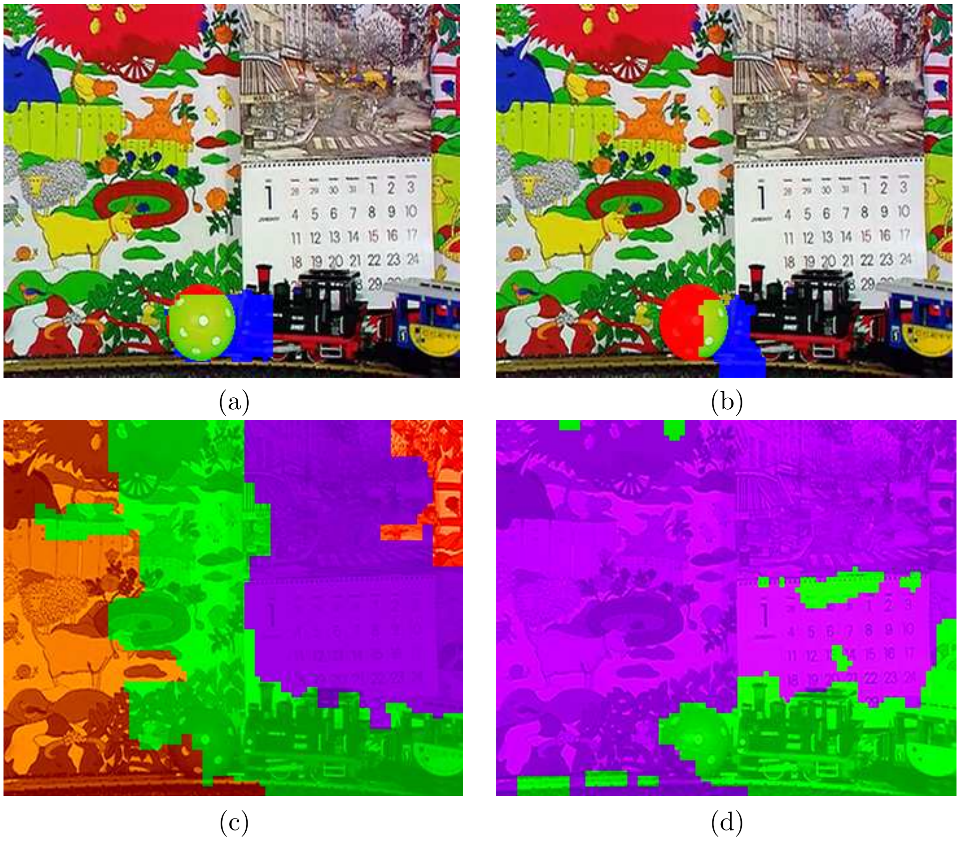


(a) (b)

(c) (d)

(e) (f)

Figure 7.7: Object detection during ST-MRF-based tracking (a) frame #70 of *Coastguard* (b) target superimposed by scaled MVF after GMC, (c) the heatmap visualization of MRF energy, (d) tracking result by the proposed method, (e) segmentation result from [119], and (f) [163].



(a)

(b)

(c)

(d)

Figure 7.8: Object detection/tracking by (a) proposed method, (b) and method from [103] for frame #97 of *Mobile Calendar* (c) Segmentation result from [119], and (d) [163].

methods show that the ball is not separated from the train, which is not surprising, since these methods rely on spatial segmentation of MVs. Another example that illustrates the robustness of the proposed method is the *Coastguard* sequence. Here, our method is able to track the small boat throughout the sequence, even during the fast camera movement, as shown by the trajectory in Fig. 7.9(b). By comparison, none of the other three methods were able to segment and/or track the small boat in the frames #68 to #76, where the camera motion overwhelms the object motion.

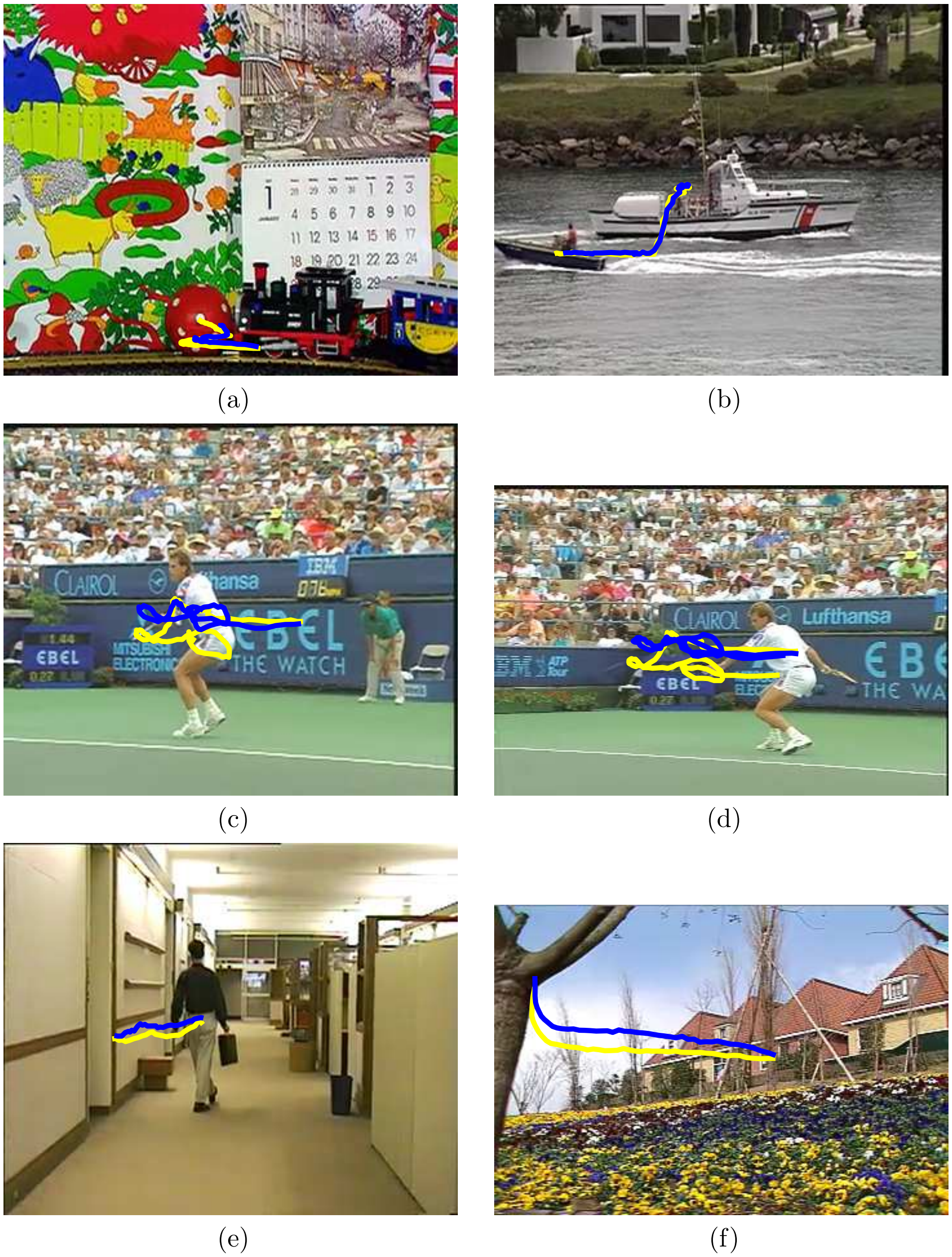
The average values of *Precision*, *Recall* and *F-measure* for several sequences where segmentation ground truth was available are shown in Table 7.2 for [119], [163], and the proposed method. *Precision* is defined as the number of TP divided by the total number of labeled pixels, i.e., the sum of TP and FP. *Recall* is defined as the number of TP divided by the total number of ground truth labels, i.e., the sum of TP and FN. *F-measure* is the harmonic mean of precision and recall. Unfortunately, for the method from [103], we were only able to obtain object masks for *Mobile Calendar* from the authors. As seen in this

Table 7.2: Total average of *Precision* (P), *Recall* (R), and *F-Measure* (F) in percent for different methods

Method	**Measure**	*Mobile*	*Coastguard*	*Stefan* (CIF)	*Stefan* (SIF)	*Hall*	*Garden*	*Tennis*	*City*	*Foreman*	**Avg.**
Proposed	P	75.9	64.3	84.2	84.7	72.8	82.9	94.1	92.9	92.3	**82.7**
	R	88.4	89.4	68.3	67.8	84.4	95.8	88.0	96.5	90.4	**85.5**
	F	81.2	74.4	74.1	74.3	78.1	88.8	90.8	94.6	91.2	**83.0**
[**119**]	P	31.8	4.8	18.5	18.4	27.9	53.2	76.3	86.8	85.8	**44.8**
	R	91.6	86.3	86.0	88.5	91.9	99.0	72.9	96.9	64.3	**86.4**
	F	40.5	8.1	25.5	24.4	37.3	68.8	69.9	91.5	69.9	**48.4**
[**163**]	P	6.6	3.0	10.7	10.5	15.6	34.6	48.9	77.8	81.2	**32.1**
	R	93.3	95.9	87.8	88.4	90.1	99.2	76.2	97.0	65.3	**88.1**
	F	12.1	5.8	18.3	17.9	22.9	50.7	52.2	84.2	69.5	**37.1**

table, the proposed method has the highest *Precision* and *F-measure* across all sequences, averaging almost a two-fold improvement in both of these metrics. The reason why its *Recall* is sometimes lower than that of the other two methods is that it tracks the boundary of the target object fairly closely in a block-based fashion, and therefore excludes from the object those pixels that fall into boundary blocks that get declared as the background, resulting in higher FN. By comparison, the other two methods often include the object and a large portion of the background into the segmented region, resulting in a smaller FN.

The example shown in Fig. 7.9 shows the tracked trajectory produced by the proposed method for several standard sequences, superimposed onto the last frame in the sequence. Trajectory is obtained by connecting the center of gravity of the tracked target through the frames. The blue lines connect the centers of gravity produced by the proposed method, while yellow lines connect the centers of gravity of the ground truth. The trajectory result of ball tracking in *Mobile Calendar* is shown in Fig. 7.9(a). Although the ball has unreliable MVs due to its rotational movement and relatively small size, and even changes its direction twice, the proposed algorithm is able to track it reasonably well. In *Coastguard* in Fig. 7.9(b), the small boat moves from right to left; the camera at first follows the small boat until frame #66, then moves upwards quickly, and after that it follows the big boat starting at frame #75. During the first 66 frames, because the camera motion is in line with the movement of the small boat, the location of the small boat relative to the frame is fairly static. During the frames #67 to #74, when the camera moves upwards quickly, the MVs are rather erratic. The method of [103] has problems in this part of the sequence and fails to track the small boat, as noted in [103]. Similarly, the methods of [163] and [119] fail here as well, since the camera motion completely overwhelms object motion. However, the proposed method is able to track the small boat fairly accurately even through this challenging part of the sequence. Fig. 7.9(c) and Fig. 7.9(d) (*Stefan* CIF and SIF) are good examples of having both rapid movement of the target object (the player) and the camera. Since our algorithm is not able to detect the player's feet as a part of the target region, due to their small size relative to the block size, the center of gravity produced by the proposed algorithm is above the ground truth center of gravity. Other than that, the proposed method correctly tracks the player even in this challenging MVF involving fast camera motion and rapid player's movement. Fig. 7.9(e) shows the trajectory results for *Hall Monitor*, where the camera is fixed and the man is moving from the door to the end of the hall. In Fig. 7.9(f) the camera motion is toward the right, therefore the position of the target (tree trunk) changes from right to left within the frame. The center of gravity goes upwards in the latter part because the bottom of the tree trunk disappears and the center of gravity rises as the result.



(a) (b) (c) (d) (e) (f)

Figure 7.9: Trajectory results of (a) *Mobile Calendar*, (b) *Coastguard* (c) *Stefan CIF* (d) *Stefan SIF* (e) *Hall Monitor* and (f) *Flower Garden* sequences (blue lines: proposed algorithm, yellow lines: ground truth)

Chapter 8

Conclusions and Future Work

8.1 Summary and Conclusions

In this dissertation, we studied two important video processing problems – saliency estimation and tracking – in the compressed domain. Although pixel-based methods have the potential to yield more accurate results compared to compressed-domain methods, their high computational complexity limit their use in practice. Since all digital video content available to the end-users is in a compressed form anyway, pixel-domain algorithms require additional computational effort to decode the video bitstream before they can be applied. On the other hand, compressed-domain algorithms are able to significantly reduce computational complexity by utilizing information from the encoded video bitstream at the expense of losing some accuracy.

In terms of saliency estimation, we proposed three compressed-domain features as a measure of saliency, and showed that they have high correlation with human gaze points. Subsequently, two saliency estimation algorithms have been proposed and shown to have superior accuracy with respect to state-of-the-art algorithms, even pixel-domain ones.

In particular, in Chapter 2 we presented an overview of a popular pixel-domain saliency model, the so-called Itti-Koch-Niebur (IKN). The IKN saliency map is estimated from color images through a biologically plausible manner. This bottom-up model of visual attention analyzes various pre-attentive independent feature channels such as intensity, color and orientation. Each feature is calculated and contrasted within two different resolutions using a center-surround mechanism where each resolution is obtained by progressively low-pass filtering and down-sampling the input image. More specifically, 6 intensity feature maps, 12 color feature maps, and 24 maps for orientation feature maps are created from the image. Finally, all computed feature maps are combined across all resolutions as well as feature channels to create the master saliency map. We also described some of variants of IKN in Chapter 2. Two normalization operators for combining feature maps, namely MaxNorm and FancyOne, were explained. In contrast to the MaxNorm operator, FancyOne is more in tune with the local connectivity of cortical neurons and produces sparser saliency maps with sharper peaks, leading to more accurate prediction of human fixations. In another variation, two new features that are responsible for motion and flicker contrasts were added to the standard IKN model to address spatio-temporal saliency estimation. Particularly, 24 motion feature maps and 6 flicker feature maps are created for various orientations, and combined with the feature maps from basic IKN model to generate the spatio-temporal saliency map.

A unified framework for accuracy measurement of visual saliency models was introduced in Chapter 3. Two popular eye-tracking datasets for video, namely the SFU and DIEM datasets, were used in our study. In addition to these ground-truth data, two saliency maps, i.e., Intra-Observer (IO) and Gaussian center-bias (GAUSS), were used as benchmarks to evaluate computational models. IO saliency map is obtained by the convolution of a 2-D Gaussian blob with the second set of eye tracking fixations of the same observer. IO saliency map was used as a benchmark for top performer saliency models. GAUSS map is simply a 2-D Gaussian blob located at the image center, considered as the output of a fully center-biased model. Several metrics for evaluating the accuracy of visual saliency models were also reviewed in this chapter, and then several novel metrics, namely AUC$'$, NSS$'$ and JSD$'$, were proposed to overcome the shortcomings of conventional evaluation metrics, such as gaze point uncertainty, center bias and border effects. A highly accurate computational saliency model is expected to perform well across most metrics.

A comparison among existing compressed-domain saliency estimation models was given in Chapter 4. Three well-known pixel-domain saliency models, i.e., MaxNorm, GBVS and AWS, were also considered in this comparison to gain insight into the performance gap between the current compressed-domain and pixel-domain state-of-the-art. In total, nine compressed-domain visual saliency models were included in the study. While different video coding standards were assumed by different compressed-domain models, our comparison was carried out using the MPEG-4 ASP format. In this case, only two models (MCSDM and

APPROX) required minor modifications. The experimental results revealed that MaxNorm, AWS, GBVS, PMES, GAUS-CS and PNSP-CS sustain superior performance according to different accuracy metrics. While AWS needs 1535 ms on average per CIF video frame in our environmental setting, GBVS 783 ms, MaxNorm 91 ms, and the three top performer compressed-domain models from 57 ms to 98 ms. It is encouraging that using only compressed-domain information such as motion vectors (MVs), one can obtain comparable accuracy to pixel-domain models with much less complexity. Furthermore, based on our analysis, even though the model's accuracy gets worse as the quality of encoded video drops, the performance of a saliency model is relatively consistent over a meaningful range of Peak Signal-to-Noise Ratio (PSNR).

In Chapter 5, we proposed three novel compressed-domain features – Motion Vector Entropy (MVE), Smoothed Residual Norm (SRN) and Operational Block Description Length (OBDL). MVE is computed by MVs and block coding modes (BCMs), and SRN by transformed prediction residuals. OBDL, on the other hand, is obtained directly from the output of the entropy decoder and is considered as a measure of incompressibility. Both MVE and SRN are computed from the H.264/AVC bitstream in this work, but can be extended to other video coding standards such as newly developed HEVC. The adaptation of OBDL to any encoding standard is even simpler since OBDL is simply the number of bits required to encode a block within a frame. We analyzed the statistics of the proposed compressed-domain features around fixation points and randomly-selected non-fixation points, and validated using a two-sample t-test the hypothesis that on average, feature values at fixation points tend to be higher than at non-fixation points.

In Chapter 6, two saliency estimation models were introduced based on the compressed-domain features proposed in Chapter 5. The first model is called MVE+SRN, and computes the saliency map by fusing MVE and SRN feature maps. In the second model, called OBDL-MRF, OBDL feature maps are filtered spatially and temporally, and the results are incorporated into a Markov Random Field (MRF) model. In other words, the saliency labeling of blocks within a frame at a certain time is solved using a *maximum a posteriori* (MAP) of a MRF model. This allows formulating the log-posterior as the sum of temporal consistency, observation coherence and compactness. To find the optimum label assignment, we used Iterated Conditional Modes (ICM) mainly due to its low computational complexity. The resulting saliency label assignments are then exploited to enhance the initial saliency value. While, at a high level, OBDL-MRF is similar to well-known saliency models, such as those based on self-information and surprise, it has the distinct advantage of being readily available at the output of any video encoder, which already exists in most modern cameras. Furthermore, the compressibility measure now proposed naturally takes into account the trade-off between spatial and temporal information, because the video encoder already performs rate-distortion optimization to minimize the number of bits needed to reconstruct different regions in video. In this sense, the proposed solution is a much more sophisticated measure of compressibility than previous measures based on reconstruction error, or cruder measurements of self-information. We compared our proposed saliency models, including MVE+SRN and OBDL-MRF, with seven compressed-domain saliency models as well as seven prominent pixel-domain counterparts on H.264/AVC-compressed video. The resulting saliency measures were shown highly accurate for the prediction of eye fixations, achieving state-of-the-art results on standard benchmarks. This is complemented by very low complexity, an average of 30 ms for MVE+SRN and 39 ms for OBDL-MRF per CIF video frame in MATLAB, which makes these models appropriate for practical deployment. Our analysis of the relationship between the performance of our saliency models and objective quality showed that both MVE+SRN and OBDL-MRF achieve their highest accuracy at PSNR around 35 dB while their performance degrades dramatically at both extremes of PSNR, below 26 dB and above 42 dB.

Finally, in Chapter 7, we have presented a novel approach to track a moving object/region in a H.264/AVC-compressed video. The only data from the compressed stream used in the proposed method are the MVs and BCMs. As a result, the proposed method has a fairly low processing time, yet still provides high accuracy. After the preprocessing stage, which consists of intra-coded block approximation and global motion compensation (GMC), we employed a MRF model to detect and track a moving target. Using this model, an estimate of the labeling of the current frame was formed based on the previous frame labeling and current motion information. The results of experimental evaluations on ground truth video demonstrate superior functionality and accuracy of our approach against other state-of-the-art compressed-domain segmentation/tracking approaches.

8.2 Future Work

In this research study, our proposed compressed features of saliency were extensively tested on H.264/AVC-compressed video. While H.264/AVC is in wide use today in media and entertainment industry, the new encoding standard, i.e., HEVC, has continued to attract market interests since its publishing in 2013. Additionally, HEVC doubles the compression ratio compared to H.264/AVC at the same level of video quality. Consequently, it would be highly interesting to investigate if there is a stronger correlation between our video features, specifically OBDL, and human fixation points in HEVC compared to H.264/AVC at the same level of quality.

In our proposed MRF model for saliency detection and also for video object tracking, we used fixed temperature parameters. Although our algorithm works well even with fixed parameter values, possibly better performance may be obtained by adaptive tuning, although this would in general increase the complexity. Along these lines, dynamic parameter tuning as proposed by Kato *et al.* [77] would be worth investigating as future work. We also believe that our relatively good results of MRF inference are partly due to reasonably accurate initialization of the ICM, i.e., ICM is able to find good solutions in our case because the labels are initialized to a reasonably good configuration. As a possible future work, we can test this assertion by comparing ICM results to modern energy minimization algorithms for MRF such as graph cuts [149]. In addition, while the MRF model has some weaknesses over Conditional Random Field (CRF) [97], the labeling problem could probably be improved incrementally in the future.

In the MVE+SRN method, we used additive and multiplicative operations to combine the compressed features MVE and SRN with the same weight for each of the three terms in (6.1). Possibly more accurate saliency results can be obtained by setting different weights for different features and operations. For example, since SRN achieves more accurate saliency prediction according to different metrics (Section 6.3.2), it would seem reasonable to assign higher weight to SRN compared to MVE in saliency estimation. Also, one can expect improvement by weighting the impact of mutual reinforcement (multiplication between MVE and SRN).

We discussed various situations where most algorithms fail to estimate saliency. For example, the falling yellow leaves in *Tempete* confuse most of the algorithms as they predict these leaves to be salient, whereas the static object (the flower) in the middle of the frame is what actually attracts human attention. Also, for video sequences such as *City* and *Flower Garden*, where motion is produced only via camera motion, the algorithms generally get confused by the objects closer to the camera. Another example would be dynamic background (e.g., moving trees) which may require many bits to code, and may therefore appear as salient to algorithms relying on compressed-domain features such as OBDL. These are only a few examples where our compressed-domain algorithms, and in fact most existing algorithms, might not perform well. More research on characterizing such situations and finding appropriate solutions is needed.

Our MRF-based region tracking method attempts to track a single moving region, detected in an intra-coded frame, through subsequent inter-coded frames. To be able to use this method in a general setting, however, it should be extended to multiple region tracking. In light of that, each region can be individually tracked by our proposed MRF framework. However, the framework should be able to detect and resolve occlusion. We suggest to use Merge-Split approach [46] for occlusion reasoning. In this approach, once several regions are predicted as being occluded, they are merged and encapsulated into a new region. This new region is then tracked using its new feature characteristic. Upon separation, the region is split into separated regions. An occlusion in the next frame can be predicted by the estimated position of blocks belonging to tracked regions using their current MVs. For tracking an occluded region, MRF framework can still be applied. To do this, all energy terms in Eq. (7.6) can remain the same, except for context coherence energy, because a new encapsulated region consists of various context coherency attributes. To manage context coherence energy in the case of occlusion, our suggestion is to define several representative MVs rather than a single representative MV. In turn, the context coherence energy needs to be redefined to handle several groups of consistent motions.

It would also be interesting to investigate a new compressed-domain saliency model, which detects salient regions in intra-coded frames and simply tracks them using the proposed ST-MRF framework through inter-coded frames for a certain period of time. This would also enable hybrid pixel-compressed-domain saliency estimation, where a pixel-domain estimate is obtained in intra-coded frames, and then tracked and/or refined in subsequent inter-frames. Typically, the fraction of intra-coded frames is relatively small compared to the total number of frames in the video and they require less computational effort for decoding compared to inter-coded frames. This allows a more complex saliency estimation procedure to be employed on intra-coded frames. At the same time, there are highly accurate still image saliency models in the pixel-domain, such as AWS, that could be employed for saliency estimation in these frames.

While most efforts on saliency estimation have been done on high-quality noiseless video, much less attention has been given to low-quality video. Since one of the main application areas of saliency-based processing is video surveillance, it is important for a practical solution to perform well on low-quality video taken by cheap cameras and in unfavorable lighting conditions. Unfortunately, at this time, an eye-tracking dataset including video sequences with different sources of noise is not available, so it is not possible to comprehensively test existing saliency models on such video sequences. Creating such a dataset would therefore enable further progress on this topic.

In addition to the material presented in this dissertation, we have proposed a new method for compressed-domain estimation of global motion (GM) in [79], which has the potential to enhance tracking and possibly saliency estimation accuracy, by incorporating camera motion into saliency computation. The influence of GMC on the accuracy of saliency estimation has been shown by Hadizadeh in [51] and also discussed in Chapter 4, so it is reasonable to expect that proper handling of global motion would enhance saliency estimation.

Finally, in [80] we presented a method for visualizing the motion of an object on a background sprite (mosaic). The background sprite was generated efficiently by using a limited amount of information from the compressed video stream - MVs from each frame, and pixel values from a select subset of frames. As a further contribution, this method can be adjusted for saliency visualization to illustrate trajectories of estimated Foci of Attention (FOAs) over a video sequence on the constructed background sprite.

Bibliography

[1] Adaptive whitening saliency model (AWS). http://persoal.citius.usc.es/xose.vidal/research/aws/AWSmodel.html. [Online]. 44

[2] The dynamic images and eye movements (DIEM) project. http://thediemproject.wordpress.com. [Online]. 5, 20, 70

[3] Eye tracking database for standard video sequences. http://www.sfu.ca/~ibajic/datasets.html. [Online]. 19

[4] FFMPEG project. http://www.ffmpeg.org. [Online]. 44, 70, 80

[5] H. 264/AVC reference software. http://iphome.hhi.de/suehring/tml/. [Online]. 105

[6] Locarna systems. http://www.locarna.com. [Online]. 19, 27

[7] Saliency benchmark datasets. http://people.csail.mit.edu/tjudd/SaliencyBenchmark/. [Online]. 42

[8] SR Research. http://www.sr-research.com. [Online]. 20, 27

[9] G. Abdollahian, Z. Pizlo, and E.J. Delp. A study on the effect of camera motion on human visual attention. In *Proc. IEEE ICIP'08*, pages 693–696, 2008. 62

[10] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am*, 2(2):284–299, 1985. 69

[11] G. Agarwal, A. Anbu, and A. Sinha. A fast algorithm to find the region-of-interest in the compressed MPEG domain. In *Proc. IEEE ICME'03*, volume 2, pages 133–136, 2003. 31, 35, 37, 81

[12] S. M. Anstis and D. M. Mackay. The perception of apparent movement [and discussion]. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 290(1038):153–168, 1980. 69

[13] M. G. Arvanitidou, A. Glantz, A. Krutz, T. Sikora, M. Mrak, and A. Kondoz. Global motion estimation using variable block sizes and its application to object segmentation. In *Proc. IEEE WIAMIS'09*, pages 173–176, 2009. 67, 104

[14] J. Astola, P. Haavisto, and Y. Neuvo. Vector median filters. *Proceedings of the IEEE*, 78(4):678–689, 1990. xiv, 101, 102

[15] F. Attneave. Informational aspects of visual perception. *Psychological Review*, 61:183–193, 1954. 2

[16] H. Barlow. Cerebral cortex as a model builder. In *Models of the Visual Cortex*, pages 37–46, 1985. 2

[17] H. Barlow. Redundancy reduction revisited. *Network: Computation in Neural Systems*, 12:241–253, 2001. 2

[18] E. Baudrier, V. Rosselli, and M. Larabi. Camera motion influence on dynamic saliency central bias. In *Proc. IEEE ICASSP'09*, pages 817–820, 2009. 62

[19] J. Besag. Spatial interaction and the spatial analysis of lattice systems. *Journal of the Royal Statistical Society. Series B*, 36:192–236, 1974. 77, 95

[20] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B*, 48:259–302, 1986. 77, 79, 96

[21] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):185–207, 2013. 2, 5, 9, 24, 25

[22] A. Borji, D. N. Sihite, and L. Itti. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Trans. Image Process.*, 22(1):55–69, 2013. 3, 5, 9, 21, 24, 28, 44, 88

[23] C. D. Brown and H. T. Davis. Receiver operating characteristics curves and related decision measures: A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 80(1):24–38, 2006. 24

[24] N. Bruce and J. Tsotsos. Saliency based on information maximization. *Advances in Neural Information Processing Systems*, 18:155, 2006. 2, 42, 69

[25] M. W. Cannon and S. C. Fullenkamp. A model for inhibitory lateral interaction effects in perceived contrast. *Vision Research*, 36(8):1115–1125, 1996. 13

[26] N. R. Carlson. *Psychology: The Science of Behaviour, 4th edition.* Pearson Education Canada, 2010. 27

[27] Y. M. Chen and I. V. Bajić. A joint approach to global motion estimation and motion segmentation from a coarsely sampled motion vector field. *IEEE Trans. Circuits Syst. Video Technol.*, 21(9):1316–1328, 2011. 96

[28] Y. M. Chen, I. V. Bajić, and P. Saeedi. Motion segmentation in compressed video using Markov random fields. In *Proc. IEEE ICME'10*, pages 760–765, 2010. 96

[29] Y. M. Chen, I. V. Bajić, and P. Saeedi. Moving region segmentation from compressed video using global motion estimation and markov random fields. *IEEE Trans. Multimedia*, 13(3):421–431, 2011. 104

[30] T. M. Cover and J. A Thomas. *Elements of Information Theory.* Wiley-Interscience, 2 edition, 2006. 2

[31] T. M. Cover and J. A. Thomas. *Elements of information theory.* Wiley-Interscience, 2012. 42

[32] D. Culibrk, M. Mirkovic, V. Zlokolica, M. Pokric, V. Crnojevic, and D. Kukolj. Salient motion features for video quality assessment. *IEEE Trans. Image Process.*, 20(4):948–958, 2011. 9

[33] I. Dagan, L. Lee, and F. Pereira. Similarity-based methods for word sense disambiguation. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 56–63. Association for Computational Linguistics, 1997. 26

[34] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Optical Society of America A*, 2(7):1160–1169, 1985. 12

[35] S. Dogan, A. H. Sadka, A. M. Kondoz, E. Kasutani, and T. Ebrahimi. Fast region of interest selection in the transform domain for video transcoding. In *Proc. IEEE WIAMIS'05*, 2005. 31

[36] B. Efron and R. Tibshirani. *An introduction to the bootstrap*, volume 57. CRC press, 1993. 24, 70

[37] W. Einhäuser, M. Spain, and P. Perona. Objects predict fixations better than early saliency. *Journal of Vision*, 8(14), 2008. 24

[38] S. Engel, X. Zhang, and B. Wandell. Colour tuning in human visual cortex measured with functional magnetic resonance imaging. *Nature*, 388(6637):68–71, 1997. 11

[39] U. Engelke, H. Kaprykowsky, H. J. Zepernick, and P. Ndjiki-Nya. Visual attention in quality assessment. *IEEE Signal Process. Mag.*, 28(6):50–59, 2011. 5, 19, 93

[40] U. Engelke and H. J. Zepernick. Framework for optimal region of interest–based quality assessment in wireless imaging. *Journal of Electronic Imaging*, 19(1):1–13, 2010. 9

[41] Y. Fang, Z. Chen, W. Lin, and C. W. Lin. Saliency detection in the compressed domain for adaptive image retargeting. *IEEE Trans. Image Process.*, 21(9):3888–3901, 2012. 9, 31

[42] Y. Fang, W. Lin, Z. Chen, C. M. Tsai, and C. W. Lin. Video saliency detection in the compressed domain. In *Proc. 20th ACM Intl. Conf. Multimedia*, pages 697–700. ACM, 2012. 31, 35, 39, 61

[43] Y. Fang, W. Lin, Z. Chen, C. M. Tsai, and C. W. Lin. A video saliency detection model in compressed domain. *IEEE Trans. Circuits Syst. Video Technol.*, 24(1):27–38, 2014. 35, 39, 61, 81

[44] X. Feng, T. Liu, D. Yang, and Y. Wang. Saliency inspired full-reference quality metrics for packet-loss-impaired video. *IEEE Trans. Broadcast.*, 57(1):81–88, 2011. 9

[45] K. Fukuchi, K. Miyazato, A. Kimura, S. Takagi, and J. Yamato. Saliency-based video segmentation with graph cuts and sequentially updated priors. In *Proc. IEEE ICME'09*, pages 638–641, 2009. 9

[46] P. F. Gabriel, J. G. Verly, J. H. Piater, and A. Genon. The state of the art in multiple object tracking under occlusion in video sequences. In *Advanced Concepts for Intelligent Vision Systems*, pages 166–173, 2003. 117

[47] D. Gao and N. Vasconcelos. On the plausibility of the discriminant center-surround hypothesis for visual saliency. *Journal of Vision*, 8, 7:1–18, 2008. 2

[48] A. Garcia-Diaz, X. R. Fdez-Vidal, X. M. Pardo, and R. Dosil. Saliency from hierarchical adaptation through decorrelation and variance normalization. *Image and Vision Computing*, 30(1):51–64, 2012. 3, 69, 81

[49] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, 1984. 79

[50] C. Guo and L. Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE Trans. Image Process.*, 19(1):185–198, 2010. 9, 40

[51] H. Hadizadeh. *Visual saliency in video compression and transmission.* PhD thesis, Simon Fraser University, Apr. 2013. 31, 35, 41, 42, 43, 44, 118

[52] H. Hadizadeh and I. V. Bajić. Saliency-preserving video compression. In *Proc. IEEE ICME'11*, pages 1–6, 2011. 9

[53] H. Hadizadeh and I. V. Bajić. Saliency-aware video compression. *IEEE Trans. Image Process.*, 23(1):19–33, Jan. 2014. 9

[54] H. Hadizadeh, I. V. Bajić, and G. Cheung. Saliency-cognizant error concealment in loss-corrupted streaming video. In *Proc. IEEE ICME'12*, pages 73–78, 2012. 41

[55] H. Hadizadeh, I. V. Bajić, and G. Cheung. Video error concealment using a computation-efficient low saliency prior. *IEEE Trans. Multimedia*, 15(8):2099–2113, Dec. 2013. 9

[56] H. Hadizadeh, M. J. Enriquez, and I. V. Bajić. Eye-tracking database for a set of standard video sequences. *IEEE Trans. Image Process.*, 21(2):898–903, Feb. 2012. 5, 9, 19, 20, 35, 43, 66, 70

[57] A. Hagiwara, A. Sugimoto, and K. Kawamoto. Saliency-based image editing for guiding visual attention. In *Proc. PETMEI'11*, pages 43–48, Sep. 2011. 9

[58] S. Han and N. Vasconcelos. Biologically plausible saliency mechanisms improve feedforward object recognition. *Vision Research*, 50(22):2295–2307, 2010. 9

[59] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. *Advances in neural information processing systems*, 19:545–552, 2007. 44, 81

[60] Z. He, M. Bystrom, and S. H. Nawab. Bidirectional conversion between DCT coefficients of blocks and their subblocks. *IEEE Trans. Signal Process.*, 53(8):2835–2841, 2005. 44

[61] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282–4286, 1995. 2

[62] T. Ho-Phuoc, N. Guyader, F. Landragin, and A. Guérin-Dugué. When viewing natural scenes, do abnormal colors impact on spatial or temporal parameters of eye movements? *Journal of Vision*, 12(2), 2012. 93

[63] Y. Hochberg and A. C. Tamhane. *Multiple comparison procedures.* John Wiley & Sons, Inc., 1987. 56, 81

[64] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Proc. IEEE CVPR'07*, pages 1–8, 2007. 3, 69

[65] X. Hou and L. Zhang. Dynamic visual attention: Searching for coding length increments. *Advances in Neural Information Processing Systems*, 21:681–688, 2008. 3, 69

[66] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Trans. Image Process.*, 13(10):1304–1318, 2004. 9, 16, 43, 66, 74, 81

[67] L. Itti and P. Baldi. A principled approach to detecting surprising events in video. In *Proc. IEEE CVPR'05*, volume 1, pages 631–637, 2005. 24, 25

[68] L. Itti and P. Baldi. Bayesian surprise attracts human attention. *Vision Research*, 49(10):1295–1306, 2009. 24, 25, 61, 93

[69] L. Itti and P. F. Baldi. Bayesian surprise attracts human attention. *Advances in Neural Information Processing Systems*, 19:547–554, 2006. 2, 25, 35, 40, 69, 81, 88

[70] L. Itti and R. Carmi. Eye-tracking data from human volunteers watching complex video stimuli. 2009. CRCNS.org. 35, 88

[71] L. Itti, N. Dhavale, and F. Pighin. Realistic avatar eye and head animation using a neurobiological model of visual attention. In *Optical Science and Technology, SPIE's 48th Annual Meeting*, pages 64–78. International Society for Optics and Photonics, 2004. 16

[72] L. Itti and C. Koch. Feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging*, 10(1):161–169, 2001. xii, 14, 15

[73] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254–1259, 1998. xii, 2, 9, 10, 12, 13, 16, 17, 40, 41, 42, 44, 61, 69, 74, 76, 81

[74] H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946. 25

[75] Q. G. Ji, Z. D. Fang, Z. H. Xie, and Z. M. Lu. Video abstraction based on the visual attention model and online clustering. *Signal Processing: Image Commun.*, 28(3):241–253, 2013. 9

[76] C. Kanan, M. H. Tong, L. Zhang, and G. W. Cottrell. SUN: Top-down saliency using natural statistics. *Visual Cognition*, 17(6-7):979–1003, 2009. 2, 28, 29, 56

[77] Z. Kato, T. C. Pong, and J. Chung-Mong Lee. Color image segmentation and parameter estimation in a Markovian framework. *Pattern Recognition Letters*, 22(3):309–321, 2001. 116

[78] H. Khalilian and I. V. Bajić. Video watermarking with empirical PCA-based decoding. *IEEE Trans. Image Processing*, 22(12):4825–4840, Dec 2013. 9

[79] S. H. Khatoonabadi and I. V. Bajić. Compressed-domain global motion estimation based on the normalized direct linear transform algorithm. In *Proc. International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC'13)*, 2013. 118

[80] S. H. Khatoonabadi and I. V. Bajić. Still visualization of object motion in compressed video. In *Proc. IEEE ICME'13 Workshop: MMIX*, 2013. 118

[81] S. H. Khatoonabadi and I. V. Bajić. Video object tracking in the compressed domain using spatio-temporal Markov random fields. *IEEE Trans. Image Process.*, 22(1):300–313, 2013. 7, 93

[82] S. H. Khatoonabadi, I. V. Bajić, and Y. Shan. Comparison of visual saliency models for compressed video. In *Proc. IEEE ICIP'14*, pages 1081–1085, 2014. 5, 32

[83] S. H. Khatoonabadi, I. V. Bajić, and Y. Shan. Compressed-domain correlates of fixations in video. In *ACM Multimedia PIVP*, PIVP '14, pages 3–8, 2014. 6

[84] S. H. Khatoonabadi, I. V. Bajić, and Y. Shan. Compressed-domain correlates of human fixations in dynamic scenes. *Multimedia Tools and Applications*, 2015. [Accepted for publication]. 6

[85] S. H. Khatoonabadi, I. V. Bajić, and Y. Shan. Compressed-domain visual saliency models: A comparative study. *IEEE Trans. Image Process.*, 2015. [Submitted]. 5, 32

[86] S. H. Khatoonabadi, N. Vasconcelos, I. V. Bajić, and Y. Shan. How many bits does it take for a stimulus to be salient? In *Proc. IEEE CVPR'15*, pages 5501–5510, 2015. 6

[87] C. Kim and P. Milanfar. Visual saliency in noisy images. *Journal of Vision*, 13(4):1–14, 2013. 44, 59

[88] W. Kim, C. Jung, and C. Kim. Spatiotemporal saliency detection and its applications in static and dynamic scenes. *IEEE Trans. Circuits Syst. Video Technol.*, 21(4):446–456, 2011. 81

[89] D. Knill and W. Richards. *Perception as Bayesian Inference.* Cambridge Univ. Press, 1996. 2

[90] E. Kreyszig. *Introductory mathematical statistics: principles and methods.* Wiley New York, 1970. 72

[91] S. Kullback. *Information theory and statistics.* Courier Dover Publications, 1997. 25

[92] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. 25

[93] E. C. Larson, C. Vu, and D. M. Chandler. Can visual fixation patterns improve image fidelity assessment? In *Proc. IEEE ICIP'08*, pages 2572–2575, 2008. 9

[94] O. Le Meur. Robustness and repeatability of saliency models subjected to visual degradations. In *Proc. IEEE ICIP'11*, pages 3285–3288, 2011. 59

[95] O. Le Meur and T. Baccino. Methods for comparing scanpaths and saliency maps: Strengths and weaknesses. *Behavior Research Methods*, 45(1):251–266, 2013. 24, 27

[96] A. G. Leventhal. *The neural basis of visual function.* Vision and Visual Dysfunction. Boca Raton, CRC Press, 1991. 10, 11, 12

[97] S. Z. Li. *Markov random field modeling in image analysis.* Springer Science & Business Media, 2009. 116

[98] X. Li, H. Lu, L. Zhang, X. Ruan, and M. H. Yang. Saliency detection via dense and sparse reconstruction. In *Proc. IEEE ICCV'13*, pages 2976–2983, 2013. 3, 69

[99] Z. Li, S. Qin, and L. Itti. Visual attention guided bit allocation in video compression. *Image and Vision Computing*, 21(1):1–19, Jan. 2011. 9

[100] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory*, 37(1):145–151, 1991. 26

[101] X. Lin, H. Ma, L. Luo, and Y. Chen. No-reference video quality assessment in the compressed domain. *IEEE Trans. Consum. Electron.*, 58(2):505–512, 2012. 31

[102] H. Liu and I. Heynderickx. Visual attention in objective image quality assessment: based on eye-tracking data. *IEEE Trans. Circuits Syst. Video Technol.*, 21(7):971–982, 2011. 9

[103] Z. Liu, Y. Lu, and Z. Zhang. Real-time spatiotemporal segmentation of video objects in the H. 264 compressed domain. *J. Vis. Commun. Image R.*, 18(3):275–290, 2007. xv, 96, 106, 108, 109

[104] Z. Liu, H. Yan, L. Shen, Y. Wang, and Z. Zhang. A motion attention model based rate control algorithm for H. 264/AVC. In *The 8th IEEE/ACIS International Conference on Computer and Information Science (ICIS'09)*, pages 568–573, 2009. 31, 35, 38, 39, 44, 81, 93

[105] T. Lu, Z. Yuan, Y. Huang, D. Wu, and H. Yu. Video retargeting with nonlinear spatial-temporal saliency fusion. In *Proc. IEEE ICIP'10*, pages 1801–1804, 2010. 9

[106] Z. Lu, W. Lin, E. P. Ong, X. Yang, and S. Yao. PQSM-based RR and NR video quality metrics. In *Proceedings-SPIE The International Society of Optical Engineering*, volume 5150, pages 633–640. International Society for Optical Engineering, 2003. 9

[107] Y. F. Ma, X. S. Hua, L. Lu, and H. J. Zhang. A generic framework of user attention model and its application in video summarization. *IEEE Trans. Multimedia*, 7(5):907–919, 2005. 2

[108] Y. F. Ma and H. J. Zhang. A new perceived motion based shot content representation. In *Proc. IEEE ICIP'01*, volume 3, pages 426–429, 2001. 31, 35, 36, 81, 93

[109] Y. F. Ma and H. J. Zhang. A model of motion attention for video skimming. In *Proc. IEEE ICIP'02*, volume 1, pages 129–132, 2002. 31, 35, 37, 81, 93

[110] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *Proc. IEEE CVPR'10*, pages 1975–1981, 2010. 3, 9

[111] V. Mahadevan and N. Vasconcelos. Spatiotemporal saliency in dynamic scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):171–177, 2010. 35, 41, 61, 93

[112] V. Mahadevan and N. Vasconcelos. Biologically inspired object tracking using center-surround saliency mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(3):541–554, 2013. 9

[113] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 11:674–693, July 1989. 2

[114] R. Margolin, A. Tal, and L. Zelnik-Manor. What makes a patch distinct? In *Proc. IEEE CVPR'13*, pages 1139–1146, 2013. 3, 69

[115] V. A. Mateescu and I. V. Bajić. Guiding visual attention by manipulating orientation in images. In *Proc. IEEE ICME'11*, pages 1–6, Jul. 2013. 9

[116] V. A. Mateescu and I. V. Bajić. Attention retargeting by color manipulation in images. In *Proc. 1st Intl. Workshop on Perception Inspired Video Processing*, PIVP '14, pages 15–20, 2014. 9

[117] V. A. Mateescu, H. Hadizadeh, and I. V. Bajić. Evaluation of several visual saliency models in terms of gaze prediction accuracy on video. In *Proc. IEEE Globecom'12 Workshop: QoEMC*, pages 1304–1308, Dec. 2012. 9, 14, 56

[118] J. T. McClave and T. Sincich. Statistics, 9th edition, 2003. 42

[119] Y. M. Meng, I. V. Bajić, and P. Saeedi. Moving region segmentation from compressed video using global motion estimation and Markov random fields. *IEEE Trans. Multimedia*, 13(3):421–431, 2011. xv, 96, 106, 107, 108, 109

[120] R. Milanese, S. Gil, and T. Pun. Attentive mechanisms for dynamic and static scene analysis. *Optical Engineering*, 34(8):2428–2434, 1995. 61, 93

[121] A. K. Mishra, Y. Aloimonos, L. F. Cheong, and A. Kassim. Active visual segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(4):639–653, 2012. 9

[122] P. K. Mital, T. J. Smith, R. L. Hill, and J. M. Henderson. Clustering of gaze during dynamic scene viewing is predicted by motion. *Cognitive Computation*, 3(1):5–24, 2011. 27

[123] A. K. Moorthy and A. C. Bovik. Visual importance pooling for image quality assessment. *IEEE J. Sel. Topics Signal Process.*, 3(2):193–201, 2009. 9

[124] B. Moulden, J. Renshaw, and G. Mather. Two channels for flicker in the human visual system. *Perception*, 13(4):387–400, 1984. 69

[125] MPEG-7 Output Document ISO/MPEG. MPEG-7 visual part of experimentation model (XM) version 2.0. 1999. 35

[126] K. Muthuswamy and D. Rajan. Salient motion detection in compressed domain. *IEEE Signal Process. Lett.*, 20(10):996–999, Oct. 2013. 31, 35, 40, 41, 81

[127] V. Navalpakkam and L. Itti. Modeling the influence of task on attention. *Vision Research*, 45(2):205–231, 2005. 2

[128] E. Niebur and C. Koch. Computational architectures for attention. chapter 9, pages 163–186. Cambridge, MA: MIT Press, 1998. 1

[129] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006. 2

[130] B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996. 2

[131] D. J. Parkhurst and E. Niebur. Scene content selected by active vision. *Spatial Vision*, 16(2):125–154, 2003. 28, 29, 56

[132] R. J. Peters, A. Iyer, L. Itti, and C. Koch. Components of bottom-up gaze allocation in natural images. *Vision Research*, 45(18):2397–2416, 2005. 26

[133] M. I. Posner and Y. Cohen. Components of visual orienting. *Attention and Performance X: Control of Language Processes*, 32:531–556, 1984. 15, 16

[134] W. Reichardt. Evaluation of optical motion information by movement detectors. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 161(4):533–547, 1987. 16

[135] P. Reinagel and A. M. Zador. Natural scene statistics at the center of gaze. *Network: Computation in Neural Systems*, 10:1–10, 1999. 38, 70

[136] P. J. Rousseeuw and C. Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993. 97

[137] P. J. Rousseeuw and A. M. Leroy. Robust regression and outlier detection. *Wiley Series in Probability and Mathematical Statistics, New York*, 2003. 97

[138] N. G. Sadaka, L. J. Karam, R. Ferzli, and G. P. Abousleman. A no-reference perceptual image sharpness metric based on saliency-weighted foveal pooling. In *Proc. IEEE ICIP'08*, pages 369–372, 2008. 9

[139] E. F. Schisterman, D. Faraggi, B. Reiser, and M. Trevisan. Statistical inference for the area under the receiver operating characteristic curve in the presence of random measurement error. *American Journal of Epidemiology*, 154(2):174–179, 2001. 24

[140] N. Sebe and M. S. Lew. Comparing salient point detectors. *Pattern Recognition Letters*, 24(1):89–96, 2003. 3

[141] H. J. Seo and P. Milanfar. Static and space-time visual saliency detection by self-resemblance. *Journal of Vision*, 9(12):15, 2009. 2, 81

[142] B. Shen and I. K. Sethi. Convolution-based edge detection for image/video in block DCT domain. *J. Vis. Commun. Image R.*, 7(4):411–423, 1996. 40

[143] A. Sinha, G. Agarwal, and A. Anbu. Region-of-interest based compressed domain video transcoding scheme. In *Proc. IEEE ICASSP'04*, volume 3, pages 161–164, 2004. 31, 35, 38, 81

[144] A. Smolic, M. Hoeynck, and J. R. Ohm. Low-complexity global motion estimation from P-frame motion vectors for MPEG-7 applications. In *Proc. IEEE ICIP'00*, volume 2, pages 271–274, 2000. 104

[145] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S. C. Zhu. On advances in statistical modeling of natural images. *Journal of mathematical imaging and vision*, 18(1):17–33, 2003. 2

[146] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE CVPR'99*, pages 246–252, 1999. 3, 9

[147] G.J. Sullivan, J. Ohm, Woo-Jin Han, and T. Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.*, 22(12):1649–1668, 2012. 63, 66

[148] J. A. Swets. *Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers.* Lawrence Erlbaum Associates, Inc., 1996. 24

[149] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields. In *Proc. ECCV'06*, volume 2, pages 16–29, 2006. 79, 116

[150] B. W. Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision*, 7(14), 2007. 28

[151] B. W. Tatler, R. J. Baddeley, and I. D. Gilchrist. Visual correlates of fixation selection: Effects of scale and time. *Vision Research*, 45(5):643–659, 2005. 28, 29, 42

[152] C. Theoharatos, V. K. Pothos, N. A. Laskaris, G. Economou, and S. Fotopoulos. Multivariate image similarity in the compressed domain using statistical graph matching. *Pattern Recognition*, 39(10):1892–1904, 2006. 39

[153] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980. 10, 39

[154] P. Vandewalle, J. Kovacevic, and M. Vetterli. Reproducible research in signal processing. *IEEE Signal Process. Mag.*, 26(3):37–47, 2009. 7

[155] W. N. Venables and B. D. Ripley. *Modern applied statistics with S.* Springer, 2002. 97

[156] C. Wang, N. Komodakis, and N. Paragios. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117(11):1610–1627, 2013. 76

[157] Z. Wang and A. C. Bovik. Foveated image and video coding. In H. R. Wu and B. D. Rao, editors, *Digital Video Image Quality and Perceptual Coding*, pages 431–458. 2005. 9

[158] Z. Wang and Q. Li. Information content weighting for perceptual image quality assessment. *IEEE Trans. Image Process.*, 20(5):1185–1198, 2011. 9

[159] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H. 264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):560–576, 2003. 43, 66, 100

[160] S. Winkler and R. Subramanian. Overview of eye tracking datasets. In *5th Intl. Workshop on Quality of Multimedia Experience (QoMEX)*, pages 212–217. IEEE, 2013. 19

[161] R. Xie and S. Yu. Region-of-interest-based video transcoding from MPEG-2 to H. 264 in the compressed domain. *Optical Engineering*, 47(9):097001–097001, 2008. 31, 93

[162] H. Zen, T. Hasegawa, and S. Ozawa. Moving object detection from MPEG coded picture. In *Proc. IEEE ICIP'99*, volume 4, pages 25–29, 1999. 37

[163] W. Zeng, J. Du, W. Gao, and Q. Huang. Robust moving object segmentation on H. 264/AVC compressed video using the block-based MRF model. *Real-Time Imaging*, 11(4):290–299, 2005. xv, 96, 106, 107, 108, 109

[164] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. SUN: A bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7), 2008. 2, 28

[165] S. H. Zhong, Y. Liu, Y. Liu, and F. L. Chung. A semantic no-reference image sharpness metric based on top-down and bottom-up saliency map modeling. In *Proc. IEEE ICIP'10*, pages 1553–1556, 2010. 9