Thèse présentée à la faculté des sciences pour
l'obtention du grade de docteur ès sciences

Computer models of dynamic visual attention

Alexandre Bur

Acceptée sur proposition du jury :
Prof. Heinz Hügli, directeur de thèse
Prof. Pierre-André Farine, co-directeur de thèse
Prof. Jean-Philippe Thiran, rapporteur
Prof. René Müri, rapporteur
Dr. Olivier Le Meur, rapporteur
Soutenue le 3 mars 2009

Institut de Microtechnique
Université de Neuchâtel

Université de Neuchâtel

FACULTE DES SCIENCES
Secrétariat-Décanat de la faculté
Rue Emile-Argand 11
CP 158
CH-2009 Neuchâtel

IMPRIMATUR POUR LA THESE

Computer models of dynamic visual attention

Alexandre BUR

UNIVERSITE DE NEUCHATEL

FACULTE DES SCIENCES

La Faculté des sciences de l'Université de Neuchâtel,
sur le rapport des membres du jury

MM. H. Hügli (directeur de thèse),
P.-A. Farine (co-directeur de thèse),
R. Müri (Uni. Berne), J.-P. Thiran (EPF-Lausanne),
O. Le Meur (Thomson, France)

autorise l'impression de la présente thèse.

Neuchâtel, le 28 avril 2009

Le doyen :
F. Kessler

UNIVERSITE DE NEUCHATEL
FACULTE DES SCIENCES
Secrétariat - décanat de la faculté
Rue Emile-Argand 11 - CP 158
CH-2009 Neuchâtel

Felix Kessler

Abstract

To perceive the environment efficiently, the human vision system proceeds by selecting salient targets. The targets are explored successively by means of saccadic eye movements, which are responsible for shifting the fovea onto the current fixated target. Defined as selective attention, this mechanism can be seen as a preprocessing step, which reduces the amount of information that will be processed later by the brain.

A topic in computer vision deals with the modeling of visual attention. While most investigations concentrate on static computer model, i.e. a computer system selecting salient targets from a still image, only some recent works deal with dynamic computer model, i.e. a computer system selecting salient targets from video streams. Such a paradigm is an attractive solution to reduce complexity issues in computer vision applications. Extending such a computer system to video sequences will lead to promising perspectives. Given the importance of video sequences today, the application potential is huge and covers domains like video compression, video quality assessment, mobile robot navigation, monitoring and surveillance.

The purpose of such model is to provide an automatic selection of potential regions of interest all over the sequence duration. The selection process relies on motion as well as static feature contrasts. It encompasses the feature extraction from the video sequence and its integration in a competitive way to define the resulting saliency map. This scalar map indicates salient locations, in the form of a saliency distribution. At the end, most salient regions of interest are defined from the saliency map using a selection process based on a neural network.

This thesis investigates the design of dynamic computer VA modeling, which relies on three main axes: (i) the static model, (ii) the motion model, and (iii) the map integration scheme to fuse both static and motion channels.

First, the static model relies extensively on previous works that have reported impressive findings on biological and artificial visual attention. The proposed static model shares similar concepts, with some improvements regarding the feature integration strategies.

Second, the design of the motion model is discussed. Research in neuroscience provides plausible hypothesis on motion analysis in the human brain. These mechanisms provide the core of the computer model. We present several computer models highlighting motion contrasts of different nature. Two novel approaches are proposed, namely the vector model which highlights relative motion contrast, and the phase & magnitude model which decouples phase and magnitude contrasts.

Third, the integration of the static and motion models is discussed. Several motion integration strategies are presented, including the novel motion priority scheme as alternative to the classical competitive scheme.

Finally, psycho-physical experiments are used to evaluate the performances of the proposed models. The experimental frame consists in showing a set of video sequences to a population of human subjects, while an eye tracker system is recording their eye movement patterns. The experimental data are then compared to the prediction of the computer models for the purpose of qualitative and quantitative evaluations. The set of video sequences includes various categories (synthetic and real scenes, fixed and moving background), showing advantages and inconveniences of each model.

Keywords: visual attention, computer model, bottom-up, motion, video sequence, psycho-physical experiments, eye movement recording.

Acknowledgments

First, I would like to express my gratitude to Prof. Heinz Hügli, who gave me the opportunity to work in his research group. I really appreciated the pleasant working atmosphere, as well the long scientific discussions we had all over these four years. In addition, I would like to thank him for having guided me to the end of my thesis and for all contributions we brought together.

I am also deeply grateful to Prof. René Müri, who gave me the opportunity to collaborate with his group and who accepted to evaluate my work. This fruitful collaboration constitutes an important part of my work, which allowed me to validate experimentally all the theoretical work. In addition, I sincerely thank Pascal Wurtz for the excellent work he did to set up all the experiments, the recruitment of the volunteers, and all the article's reviews. I really appreciated to work with him in a friendly atmosphere. In addition, I also thank all the members of the Eyelab, and particularly Roman Von Wartburg.

I would like to thank also Prof. Jean-Philippe Thiran, Dr. Olivier Le Meur and Prof. Pierre-André Farine for having accepted to evaluate my work.

I am grateful to my colleagues of the pattern recognition group for the agreeable working atmosphere and the fruitful scientific discussions: James Mure-Dubois, Iva Bogdanova and Thierry Zamofing. I am really happy to have met them and consider them as friends. I also thank all the colleagues of the signal processing group.

I would like to thank also Dr. Nabil Ouerhani, who gave me the motivation to work in this field of research, after having done a semester and master project with him as mentor. I really enjoyed working with him and really appreciate the time we spent both as friends and office colleague.

My special thanks goes to my sweet Kim-Anne for her continuous encouragements and strong support. I thank you Kim-Anne with all my love for having supporting me all the time.

Finally, I thank all my friends, the beer partners who will probably recognize themselves and all my family who strongly encouraged me.

Contents

1 Introduction **9**

1.1 Motivation . . . 9

1.2 Scope . . . 10

1.3 Contribution . . . 11

1.4 Thesis outline . . . 12

2 State of the art **13**

2.1 Static model . . . 14

2.2 Motion model . . . 15

2.2.1 Neuroscience point of view . . . 15

2.2.2 Dynamic computer models . . . 16

2.3 Motion integration strategy . . . 17

3 Static visual attention model **19**

3.1 Chapter introduction . . . 19

3.2 Saliency-based model of visual attention . . . 19

3.3 Map integration strategies . . . 21

3.3.1 State of the art . . . 22

3.3.2 Peak-to-peak normalization scheme $\mathcal{N}_{PP}$ . . . 22

3.3.3 Contents-based global amplification map transform $\mathcal{N}_{lin}$ . . . 23

3.3.4 Non-linear iterative map transform $\mathcal{N}_{iter}$ . . . 24

3.3.5 Long-term normalization scheme $\mathcal{N}_{LT}$ . . . 24

3.3.6 Non-linear exponential map transform $\mathcal{N}_{exp}$ . . . 25

3.4 Static model . . . 26

4 Motion visual attention model **29**

4.1 Chapter introduction . . . 29

4.2 Motion estimation: region-based block matching . . . 30

4.2.1 High resolution BMA . . . 31

4.2.2 Non-overlapped BMA . . . 33

4.2.3 Partial overlapped BMA . . . 33

4.2.4 Implementation of block matching algorithm 34
4.2.5 BMA methods: results and comparison 39
4.2.6 Motion field computation 42
4.2.7 Summary: BMA motion estimation 43
4.3 Motion models: DoG filtering approach 44
4.3.1 Multi-scale contrast computation 44
4.3.2 Motion magnitude model 45
4.3.3 Motion model based on direction decomposition 46
4.3.4 Motion vector model 47
4.3.5 Phase & magnitude motion model 48
4.3.6 Summary: Motion models (DoG approach) 48
4.4 Motion models: an implementation based on motion pyramid 51
4.4.1 Motion pyramid computation: two alternatives 52
4.4.2 Comparison of the two alternatives 53
4.4.3 Motion conspicuity operators 56
4.4.4 Motion magnitude model based on motion pyramid 56
4.4.5 Motion direction model based on motion pyramid 57
4.4.6 Motion vector model based on motion pyramid 58
4.4.7 Phase & magnitude model based on motion pyramid 59
4.5 Chapter summary 60

5 Dynamic model: motion integration schemes 63
5.1 Chapter introduction 63
5.2 Competitive scheme 63
5.3 Motion priority scheme 64

6 Model validation methodology 67
6.1 Chapter introduction 67
6.2 Overview of the methodology 67
6.2.1 Eye movement patterns classification 68
6.2.2 Human saliency map 72
6.3 Evaluation Metrics 74
6.3.1 Fixation-to-chance distance 74
6.3.2 Correlation coefficient 75
6.3.3 Kullback-Leibler divergence 75
6.3.4 ROC curve analysis: AUC value 75

7 Static model evaluation 77
7.1 Chapter introduction 77
7.2 Qualitative evaluation 78
7.2.1 Normalization schemes: $\mathcal{N}_{PP}$ and $\mathcal{N}_{LT}$ 78

7.2.2 Map transforms: $\mathcal{N}_{lin}$, $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$ 81
7.3 Quantitative evaluation 81
7.4 Chapter summary 85

8 Motion model evaluation 87
8.1 Chapter introduction 87
8.2 Experiments 88
8.2.1 Set of video sequences 88
8.3 Human saliency analysis 90
8.4 Qualitative model evaluation 93
8.4.1 Fixed background 93
8.4.2 Moving background 96
8.5 Quantitative model evaluation 99
8.5.1 Fixed background 100
8.5.2 Moving background 104
8.6 Additional discussions 108
8.6.1 Pop-out effect 108
8.6.2 Top-down and bottom-up influence 111
8.7 Chapter summary 112

9 Motion integration scheme evaluation 115
9.1 Chapter introduction 115
9.2 Qualitative model evaluation 115
9.3 Quantitative model evaluation 116

10 Conclusions and perspectives 121
10.1 Static model 121
10.2 Motion model 122
10.3 Motion integration schemes 124

A Application 1: robot navigation 127

B Application 2: omnidirectional visual attention 133

Chapter 1

Introduction

Computer vision is an applied science whose main objective is to provide computer with the function present in human vision. Typical applications range from video surveillance, medical imaging, industrial control quality, robot navigation and human computer interaction. Despite the impressive progress made in this field during the last decades, the current available computer vision solutions by far underlay the human visual system regarding robustness and performance. We briefly explain some reasons why the human vision system is so efficient.

Human vision can basically be divided into two main phases, low-level vision and high-level vision. The former phase can be seen as a preprocessing step, which reduces the amount of information that will be processed later by high-level and complex tasks, typically recognition tasks. Indeed, the amount of information collected on the retina is of the order of 10^8 bits per second [1], a flow of information that the brain is not able to process entirely. Therefore, a mechanism able to extract relevant information at early processing stage is required to ensure efficient perception.

Specifically, to perceive the environment efficiently, the human vision system proceeds by selecting salient targets. The targets are explored successively by means of saccadic eye movements, which are responsible for shifting the fovea onto the current fixated target. This mechanism is defined as selective attention.

One possibility to improve the performance of technical systems in computer vision is to seek inspiration from biological systems and to simulate their mechanisms. This thesis investigates computer modeling of selective attention, specifically on the aspects related to motion.

1.1 Motivation

In the human vision system, visual attention (VA) can be controlled in a voluntary manner [2]. Named as top-down attention, it is driven by the mental state of the subject, that means expectation, cognitive knowledge, or the intention of realizing a task [3]. For example, when you are looking for a specific object, your attention focusses on regions of interest containing the specific characteristics of the object.

Alternatively, attention can be controlled in an automatic and unconscious way. For example, when you are walking in a public garden, your attention is involuntarily directed to salient flowers, containing strong contrast of color. Suddenly, a biker is coming from the left and your attention focusses on the moving biker. This involuntary visual reflex is called bottom-up attention. It is of particular behavioral importance, since it constitutes an efficient alerting system, which is essential for adequate interaction with the environment.

In the field of computer vision, a recent research topic focusses on computer modeling of visual attention. Such a paradigm is an attractive solution to reduce complexity issue in computer vision applications. Indeed, it can be conceived as a preprocessing step which allows a rapid selection of a subset of the available sensory information. Once selected, the salient targets become the specific scene locations on which higher level computer vision tasks can focus.

Regarding bottom-up computer modeling, most investigations concentrate on static computer model, i.e. a computer system selecting salient targets in still image. This paradigm is used in various applications including object recognition, image segmentation and robot navigation.

Only recently, some works deal with dynamic computer model, i.e. a computer system selecting salient targets in video sequences. Extending such computer systems to video sequences will lead to promising perspectives, specifically in numerous video applications such as video compression, video quality assessment, monitoring and surveillance.

1.2 Scope

This thesis investigates the design of bottom-up computer VA models dedicated to video sequences. The purpose of such models is to provide an automatic selection of potential regions of interest all over the sequence duration. The selection process relies on motion as well as on static feature contrasts. It encompasses the feature extraction from the video sequence and its integration in a competitive way to define the resulting saliency map. This scalar map indicates salient locations, in the form of a saliency distribution. At the end, the most salient regions of interest are defined from the saliency map.

The design of a dynamic computer VA model can be divided in three main axes:

- Static model design;
- Motion model design;
- Integration of both models.

Therefore, modeling and implementation issues are discussed according to the three axes, with a main focus on the motion model and its integration in the resulting dynamic model. Moreover, the methodology used to evaluate experimentally the model performances is described.

First, the computer static model is presented, which relies on low-level feature extraction such as color, intensity and orientation. The model relies on previous works that have reported impressive findings on biological and artificial VA [4, 5]. The static model proposed in this thesis shares the same concepts, with some improvements related to the feature integration strategies.

Second, the design of the computer motion model is discussed. Research in neuroscience provides plausible hypothesis on motion analysis in the human brain. These mechanisms provide the core of the computer model. Therefore, we present the nature of motion contrasts that are visually attractive. For modeling issues, we propose several computer models highlighting motion contrasts of different nature.

For implementation issues, the motion model incorporates two parts. The first part is the motion estimation, which defines the motion field. We discuss the requirements to estimate motion accurately, using region-based matching technique. The second part is the motion contrast computation, which applies a contrast detection method to the motion field.

Third, the integration of the static and motion models is discussed to define the dynamic model. Several motion integration strategies are presented, which can be classified in two categories: the competitive scheme and the priority scheme.

1.3 Contribution

The main contributions are related to the motion model and its integration, including modeling as well as implementation aspects. Several models highlighting motion contrasts of different nature are proposed. In addition, the integration of both static and motion models is discussed. The main contributions are the following:

- Modeling and comparison of several motion models, highlighting motion contrasts of different nature, namely magnitude and phase motion contrasts. These models include two novel approaches, the vector model highlighting relative motion contrast, and the phase & magnitude model which decouples phase from the magnitude conspicuities.

- Modeling and comparison of several motion integration schemes used to fuse both static and motion components, including the novel priority scheme as an alternative to the competitive scheme [6, 7].

- An evaluation of dynamic models using psycho-physical experiments, including various categories of video sequences (synthetic and real sequences, acquired with fixed and moving background), showing advantages and inconveniences of each model as well as preferred domain of application [8].

- Novel map integration strategies, namely the long-term and the non-linear exponential normalizations [9], as alternative to those proposed in the classical model [5].

- Two computer vision applications based on visual attention. The first one is an original robot localization system encompassing panoramic vision and attention guided feature detection (appendix A). The second one is a novel computational approach of visual attention for omnidirectional images. The processing is performed in spherical geometry, and is thus applicable for any omnidirectional image that can be mapped on the sphere, typically images acquired with an hyperbolic or parabolic mirror (appendix B).

1.4 Thesis outline

The remainder of the thesis is structured into three main parts distributed in ten chapters. The first part is dedicated to the state of the art, presenting the works related to three axes namely, (i) the static model design, (ii) the motion model design and (iii) the integration of both models.

The second part is the modeling part described from Chapter 3 to 5. Chapter 3 presents the static model. A detailed description of the saliency-based model of VA [4] is given. We include some improvements related to the feature integration strategies.

Chapter 4 discusses the design of the motion model. Several motion models are proposed. First motion estimation using region-based matching technique is described. Then different motion contrast detection methods applied to the motion field are proposed to define the considered models. Chapter 5 investigates the integration of both motion and static information. Several motion integration strategies are presented, including the competitive scheme and the priority scheme.

The third one is the model evaluation part, described from Chapter 6 to 9. Chapter 6 presents the methodology used to compare and evaluation the performances of computer models. In the evaluation, psycho-physical experiments are used to compare experimental data to computer data.

Chapter 7, 8 and 9 present respectively the evaluation of the static models, the motion models and the motion integration strategies.

Finally, Chapter 10 concludes the work by providing a summary of the main concepts, discussing the advantages and limitations of the considered computer models, and providing an outlook of future works.

Chapter 2

State of the art

VA is the ability of the human vision system to rapidly select the most relevant information of the visual field. It is a concept of human perception resulting from visual processing in the brain. Several mechanisms involve the interaction of different areas in the brain. A detailed background on the neurobiology of VA is provided in [1, 10].

In neurobiology and psychology, the scientific community has made impressive advances towards understanding and modeling VA. In the field of computer vision, these advances have allowed the development of computer models simulating VA. Related investigations in computer vision are motivated by two distinctive objectives [10]. The first is to better understand human perception and provide a framework to assess experimentally the suitability of psycho-physical models or theories. The second objective is to develop a technical system which represents a useful front-end for higher-level processing, providing faster and more robust vision applications. Recent fields of application include image and video compression [11], object recognition [10, 12], image segmentation [13], and robot localization [14, 15, 16]. We note that there exists an overlap of the objectives. Indeed, the psycho-physical models might be used in computational applications, while technical systems might be well suited to explain psycho-physical data. Regarding our investigations, we focus on both objectives, with a main emphasis on the development of a technical system.

In this chapter, we present the state of the art of computer VA models. It is composed of three main parts, corresponding to the three axes of investigation:

- State of the art of static models (Section 2.1);
- State of the art of motion models (Section 2.2);
- State of the art of motion integration (Section 2.3).

2.1 Static model

We have previously mentioned in the introduction that attention can be controlled either in an unconscious manner or in a voluntary manner. The former is called bottom-up attention. It is the main scope of the thesis and it will be therefore exposed into details. The latter is called top-down attention and, compared to the former is less studied in the scientific community. Indeed, bottom-up processing is better investigated, and for computational systems, easier to realize. Recent works present attention systems combining both top-down and bottom-up information [17, 1, 18]. In [10], the author proposes an attention system that operates alternatively in an exploration mode (bottom-up) and in a recognition mode (top-down) based on prior knowledge. We refer the reader to a complete state of the art presented in [10] for more related details and references regarding top-down attention.

Attention models can be divided in two categories [10]: (i) the connectionist models [19, 20] which are based on neural network units and (ii) the filter models that use classical filtering methods [21, 22, 23, 10]. On one hand, the connectionist models claim to be more biologically plausible, since they are based on single units corresponding to neurons in the human brain. Each unique unit is therefore able to be treated differently and shows different behavior. Conversely, the filter models usually consider each pixel as a unit that is processed identically. It is an advantage in terms of computational efficiency and the filter models are especially well suited for computer vision applications. In this thesis, the proposed computer models belong to the second category.

While some computer models concentrate more on psycho-physical and biologically plausible aspects [24, 23, 25], other models focus more on efficient computation dedicated to computer vision applications. Those models belong to the filter models. Proposed in [4], the first computational architecture of bottom-up VA includes several concepts that are supported by neurobiologically evidences. These concepts include the feature integration theory, the center-surround mechanisms, the saliency map, winner-take-all network (WTA) and inhibition of return (IOR). Several models are based on these concepts. In [26], the authors develop one of the first implementation including center-surround difference based on classical filtering. A relaxation process is used for the map integration. Such an implementation results however in high computational cost.

One of the most actual model is presented in [5]. It is particularly interesting for its efficient approximation of center-surround differences. Center-surround contrast detection is based on gaussian image pyramid and cross-scale difference. In addition, the relaxation process is replaced by a weighting scheme for the map integration, resulting in faster computation.

This model [5] will be used as framework regarding the design of the static model. Therefore, the proposed static model relies on the same approach, with some improvements regarding the feature integration strategies. This will be the scope of chapter 3.

2.2 Motion model

2.2.1 Neuroscience point of view

Motion is of fundamental importance in biological vision systems. Specifically, motion is clearly involved in visual attention, where rapid detection of moving objects is essential for adequate interaction with the environment [27].

Advanced research in neurophysiology have studied motion analysis in the primate cortex. Today, it is generally admitted that motion processing in the monkey cortex goes through a serie of areas (V1, MT, MST, 7a) connected in a hierarchical way [28]. Each area is specialized in particular motion features, generally from simple to more complex and with smaller to larger receptive fields higher up in the hierarchy. In [29], the authors propose a complete motion representation including selectivities of the mentioned areas (Figure 2.1):

- V_1 **area**: selective for particular local speed and direction of motion.
- **MT area**: selective similarly to V_1 with larger receptive fields. In addition MT is selective for a particular angle between local movement direction and spatial velocity gradient.
- **MST area**: selective for complex motion patterns such as expansion (zoom-in), contraction (zoom-out) and rotation.
- **7a area**: selective to four different types of patterns: translation and spiral motion as in MST, full field rotation and radial motion (expansion or contraction) within largest receptive fields.

This hierarchical motion representation illustrates that attention is linked to motion contrasts of different nature: (i) motion contrast in magnitude, (ii) motion contrast in phase. Motion contrast in magnitude is motivated by the existence of several speed range selectivities (three in the representation), while motion contrast in phase is motivated by the existence of a set of direction selectivities (twelve in the representation). We notice that the number of range and direction selectivities are not defined from neuro-biologically evidence. The representation illustrates a concept rather than an exact model.

The hierarchical motion representation and the motion contrasts described above, have a particular importance in the motion model design. Motion contrasts in phase and magnitude constitute the basic elements that will be used to define the proposed models. This will be the scope of Chapter 4.

The next subsection presents the state of the art of dynamic computer models, and puts the emphasis on motion models. We prefer providing a state of the art on dynamic models rather than motion models, since any attention system requires both static and motion information in the design [6].

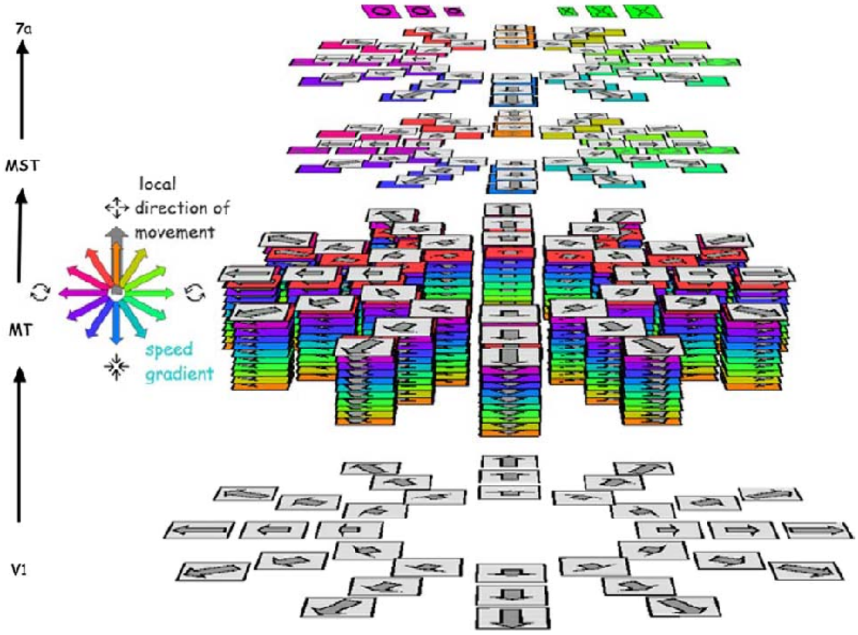


Figure 2.1: The full motion hierarchy. This shows the set of neural selectivities that comprise the entire pyramidal hierarchy covering visual areas V_1, MT, MST and 7a (from [29]).

2.2.2 Dynamic computer models

The related research in neuroscience permits to better understand how motion is processed in the brain. Over the last decade, such knowledge in neurophysiology has made possible computer modeling of visual attention. While numerous computer models have been developed for still image (static model), only few investigations focusses on computer modeling of dynamic visual attention. In order to deal with video sequences, dynamic models generally integrate additional motion components to the classical saliency-based model, a model proposed by Koch and Ullman [4]. A brief description of the related state of the art of dynamic computer models is given below.

In [30], the author considers a dynamic model combining static features (color, intensity, orientations) and dynamic features (temporal changes, four motion directions (0°, 45°, 90° and 135°) and a comparison with human vision is performed experimentally, by comparing the models with respect to the eye movement patterns of human subjects.

This investigation concludes with three points. First, motion contrast is much more relevant than any other features for predicting human attentional behavior. Second, the results show that a model including all the features, fits better to average human visual behavior than any model comprising one single channel. Finally, the study concludes that attentional allocation is strongly influenced by the low-level visual features during free-viewing of dynamics color scenes.

In [31], the authors propose a dynamic model that is based on the motion contrast, computed as the difference between local (hierarchical block matching) and dominant motion (2D affine motion model with M-estimators). The motion contrast computation includes an efficient weighting scheme, which promotes the motion map according to the rate of relative motion. This work confirms that attentional allocation is strongly influenced by the low-level visual features, as well as the model that incorporates all features is the most suitable.

In addition, the authors investigate the influence of viewing time and illustrate experimentally that in a free viewing time, attentional allocation is continuously and strongly driven by the low-level visual features.

In [32], the authors describe a dynamic model using affine motion estimation [33] and motion camera compensation as motion model. Thereby, this approach highlights conspicious moving regions, which are different from the background motion.

Another dynamic model proposed in [34] includes a motion model based on motion contrasts computed from planar motions, which are estimated by point correspondences using SIFT [35].

The next section presents the state of the art of the motion integration strategies.

2.3 Motion integration strategy

Motion integration strategy refers to the map integration strategy used to fuse both static and motion saliency maps. The resulting dynamic saliency map is defined as:

$$S_{dyn} = f(S_{static}, S_{motion}), \tag{2.1}$$

where $f(.)$ reflects the map integration strategy. We briefly mention several works that have influenced our own investigations.

In [21], the authors propose and compare three strategies used to integrate a set of conspicuity maps in a resulting saliency map. We mention them in the state of the art of this chapter, since such strategies can be applied to any kind of feature maps: (i) the straightforward normalization summation, (ii) the contents-based global amplification normalization and (iii) the more biologically plausible non-linear iterative normalization. Moreover, the last strategy is used in [30] to integrate color, intensity, orientation and motion features.

In [31], the authors present a dynamic model based on local and dominant motion. Their proposed map integration scheme relies on both intra-map and inter-map competition, a concept inspired from [26]. It consists in the summation of two contributions. The intra-map contribution, which promotes the feature maps having a sparse distribution and demotes feature maps having numerous conspicious locations. The second contribution is the inter-map contribution. It is based on complementarities (saliency induced by one feature only) and redundancies (saliency induced by conjunction of multiple features).

Another dynamic model proposed in [34] includes a motion model based on planar motion, which is estimated by point correspondences using SIFT [35]. Regarding the map fusion, the final saliency map results from the weighting summation of both maps. The weights are determined in terms of a pseudo-variance, defined as a function of maximum value and median value of the motion saliency map.

In [32], the authors describe a dynamic model using affine motion estimation [33] and motion camera compensation as motion model. Motion integration is performed by applying a maximum operator, which takes maximum value between both maps. The saliency map indicates therefore both most salient static and moving features. This method has the inconvenience not to perform an inter-map competition.

Regarding our own contributions, we present in [6] a comparative study of various visual attention models combining both static and dynamic features in different ways. We define several strategies, which are classified in two distinctive schemes, the competitive and motion priority schemes. Chapter 5 will therefore cover the mentioned motion integration schemes.

Chapter 3

Static visual attention model

3.1 Chapter introduction

This chapter defines the static model of VA, which will be used in the design of the dynamic computer model of VA. The classical model of VA computes a saliency map in a process that encompasses several map integration steps. Compared to the classical one, we propose a static model that shares the same concepts, with several differences regarding the map integration strategies.

Several map integration strategies are presented, the former are issued from the state of the art, while the later are alternatives to the former in order to solve inconveniences that we will expose later. All these strategies will be compared and evaluated experimentally in Chapter 7.

As first contribution, the non-linear exponential map transform is proposed as alternative to the non-linear DoG iterative map transform [21]. As second contribution, the long-term normalization is proposed as an alternative to the peak-to-peak normalization.

This chapter is organized as follow. First, Section 3.2 presents the classical saliency-based model of VA as a framework of the considered static model. Second, Section 3.3 discusses several map integration strategies. A state of the art is first presented (Subsection 3.3.1). Then several map integration schemes are defined, both issued from the classical model (Subsections 3.3.2, 3.3.3 and 3.3.4). Their advantages and inconveniences are discussed. After that, our contributions, namely the long-term normalization (Subsection 3.3.5) and the non-linear exponential map transform (Subsection 3.3.6) are detailed. Finally, Section 3.4 defines the map integration strategies that will be considered in the evaluation. The most suitable strategy will therefore be used to define the static part of dynamic computer model.

3.2 Saliency-based model of visual attention

The saliency-based model of visual attention was proposed by Koch and Ullman [4]. It is based on three major principles: (i) visual attention acts on a multi-featured input; (ii) saliency of locations is influenced by the surrounding context; (iii) the saliency of locations is represented on a scalar saliency map. Several works [26, 5, 25] have dealt with the realization of this model. Typically, the saliency map results from 3 cues (intensity, orientation and chromaticity) and the cues stem from 7 features. We note that other cues and features are possible (e.g. depth [36]). The different steps of the model are illustrated in Figure 3.1 and are detailed below.

First, 7 features are extracted from the scene by computing the so-called feature maps from an RGB color image. The features are typically: (a) one intensity feature F_1, (b) two chromatic features based on the two color opponency filters red-green F_2 and blue-yellow F_3 and (c) four local orientation features $F_{4..7}$.

In a second step, each feature map is transformed into its conspicuity map: the multiscale analysis decomposes each feature F_j in a set of components $F_{j,k}$ for resolution levels k=1...6; the center-surround mechanism produces the multiscale conspicuity maps $\mathcal{M}_{j,k}$ to be combined, in a competitive way, into a single ***feature conspicuity map*** C_j in accordance with:

$$C_j = \sum_{k=1}^{K} \mathcal{N}(\mathcal{M}_{j,k}), \tag{3.1}$$

where $\mathcal{N}(.)$ is a map transformation function that is used to integrate in a competitive way the different scale maps $\mathcal{M}_{j,k}$ into the conspicuity map C_j. A detailed description of the different map integration strategies is provided in Section 3.3.

In the third step, using the same competitive map integration scheme as above, the seven features are then grouped, according to their nature, into the three cues intensity, color and orientation. Formally, the ***cue conspicuity maps*** are thus:

$$C_{int} = C_1; \quad C_{chrom} = \sum_{j\epsilon\{2,3\}} \mathcal{N}(C_j); \quad C_{orient} = \sum_{j\epsilon\{4,5,6,7\}} \mathcal{N}(C_j). \tag{3.2}$$

In the fourth step of the attention model, the cue conspicuity maps are integrated, into a ***saliency map*** S, defined as:

$$S = \sum_{cue\epsilon\{int,chrom,orient\}} \mathcal{N}(C_{cue}). \tag{3.3}$$

The largest values in the saliency map indicate the most salient areas in the image. Finally, the spot of attention selection is performed by applying iteratively on the saliency map Winner-take-all mechanism (WTA) and inhibition of return (IOR). The idea consists in detecting successively the locations of the maxima. First, the most salient spot is detected as the maximum location in the map. Local inhibition is then activated at the current maximum location, which prevents the next spot from returning to previously attended location. Then, the next spots are detected by repeating WTA and IOR.

We have presented the framework of the static model. As we can see, the model encompasses a map integration step at different levels (multiscale, feature conspicuity and cue conspicuity levels (Figure 3.1)). We will see in the next section several map integration strategies.

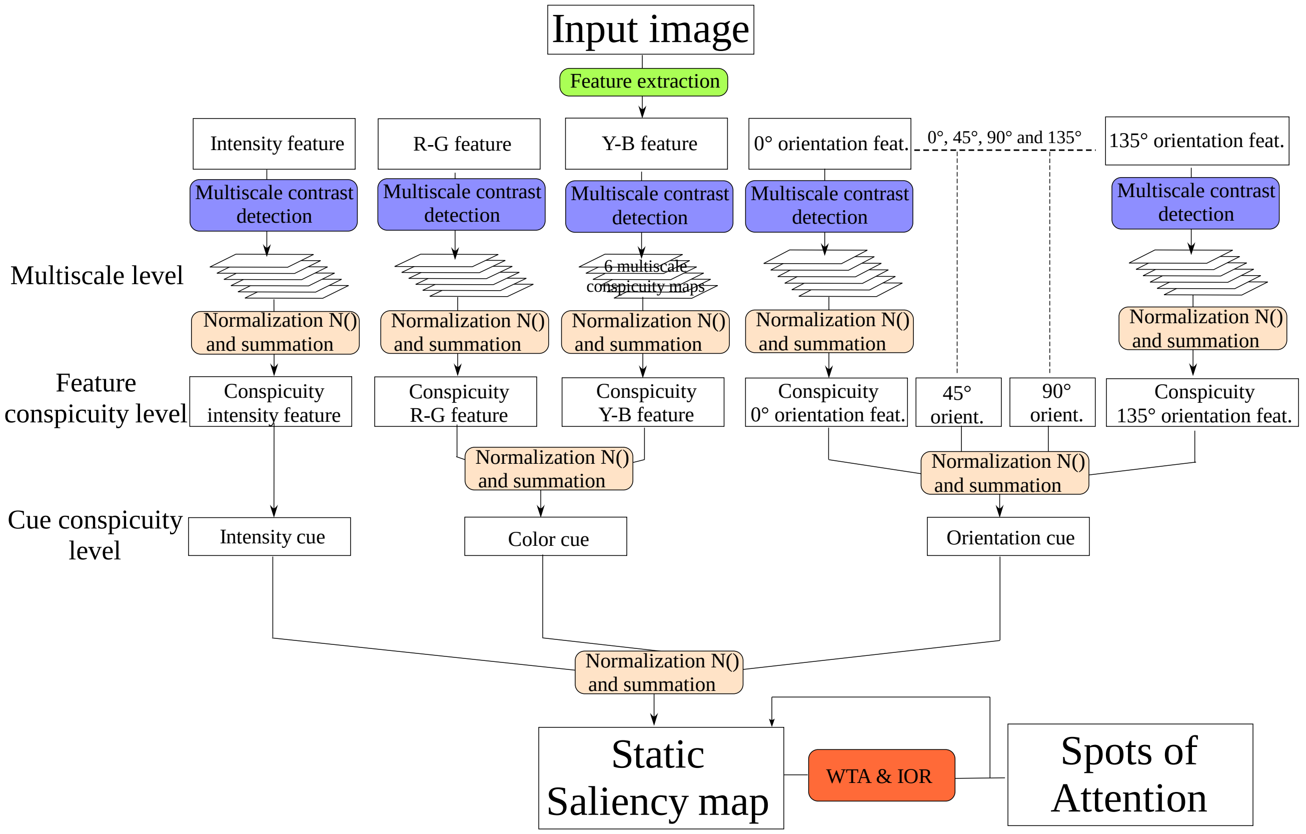

Input image
Feature extraction
Intensity feature
R-G feature
Y-B feature
0° orientation feat.
0°, 45°, 90° and 135°
135° orientation feat.
Multiscale contrast detection
Multiscale contrast detection
Multiscale contrast detection
Multiscale contrast detection
Multiscale contrast detection
Multiscale level
6 multiscale conspicuity maps
Normalization N() and summation
Normalization N() and summation
Normalization N() and summation
Normalization N() and summation
Normalization N() and summation
Feature conspicuity level
Conspicuity intensity feature
Conspicuity R-G feature
Conspicuity Y-B feature
Conspicuity 0° orientation feat.
45° orient.
90° orient.
Conspicuity 135° orientation feat.
Normalization N() and summation
Normalization N() and summation
Cue conspicuity level
Intensity cue
Color cue
Orientation cue
Normalization N() and summation
Static Saliency map
WTA & IOR
Spots of Attention

Figure 3.1: Saliency-based model of visual attention proposed by Koch and Ullman [4]

3.3 Map integration strategies

To compute the saliency map, several maps of different nature are combined at the multiscale, feature conspicuity and cue conspicuity levels. Several map integrations are possible and the strategy used to combine the maps can be different from a level to another. The map integration can generally be divided in two steps. First, a normalization scheme $\mathcal{N}_1(.)$ scales the value ranges of the different maps to a comparable value. Indeed, each map may exhibit different value ranges due to feature extraction mechanisms of different nature (intensity, color, orientation). Second, a map transform $\mathcal{N}_2(.)$ is applied to simulate a competition mechanism between the maps to be integrated. Formally, the resulting map C after the map integration procedure can be defined as:

$$C = \sum_i \mathcal{N}_2(\mathcal{N}_1(C_i)), \tag{3.4}$$

where C_i refer to the maps to be integrated.

The remainder of the section is described as follow. First, we present the state of the art.

Then we focus on the strategies proposed in the classical model [21]. Their advantages and inconveniences are discussed, and therefore, we propose alternative strategies. Finally, we define six map integration strategies that will be compared and evaluated experimentally. The most suitable strategy will be used to define the static part of the dynamic VA model.

3.3.1 State of the art

In [31], the authors propose a map integration scheme that relies on both intra-map and inter-map competition, a concept inspired from [26]. It consists in the combination of two contributions. The intra-map contribution, which promotes the feature maps having a sparse distribution and demotes feature maps having numerous conspicious locations. The second contribution is the inter-map contribution. It is based on complementarities (saliency induced by one feature only) and redundancies (saliency induce by conjuction of multiple features).

In the investigation of the classical model [5], the authors compare three strategies to integrate a set of conspicuity maps in a resulting saliency map [21]. Each of the three strategies applies a preliminary peak-to-peak normalization in order to scale the dynamic range at the same value. This method will be considered in the normalization scheme evaluation and is therefore presented in Subsection 3.3.2.

The first strategy is the straightforward normalization summation, which consists in summing the maps. It has the drawback to include irrelevant noise in the saliency map, for example an homogeneous map. We do not consider it for its evident inaccuracy.

The second strategy is the contents-based global amplification map transform. A weighting scheme is applied in order to simulate inter-map competition. The idea is to promote maps having a sparse distribution of saliency and to suppress maps having numerous conspicuous locations. This method will be considered in the evaluation of the map transforms. It is presented in Subsection 3.3.3.

The third strategy is the more biologically plausible non-linear iterative map transform. DoG filtering is used iteratively and has the advantage to perform intra-map competition, which tends to suppress the noise located around strong conspicious locations. This method is also considered in the map transform evaluation. It is presented in Subsection 3.3.4.

3.3.2 Peak-to-peak normalization scheme $\mathcal{N}_{PP}$

This normalization scheme scales all maps to the same value range in order to eliminate across-modality amplitude difference due to dissimilar feature extraction mechanisms.

In this section and the following ones, we will use the following notation: the conspicuity map C is defined as set of pixels $\{c_i\}$. Formally, the peak-to-peak normalization scheme is defined as:

$$\mathcal{N}_{PP}(c) = \frac{c - c_{min}}{c_{max} - c_{min}}, \quad where \;\; c_{min} = \min_{c \epsilon C}(c) \;\; and \;\; c_{max} = \max_{c \epsilon C}(c). \tag{3.5}$$

3.3.3 Contents-based global amplification map transform $\mathcal{N}_{lin}$

The basic idea is to simulate a competition mechanism between the maps (inter-map competition). A scalar weight w is assigned to each map C that holds for its individual contribution. Each map is weighted by its corresponding weight, which will increase or decrease the contribution of the map in the combination process. The weight is computed from the conspicuity map itself and tends to catch the global interest of the map.

This strategy is defined in three steps. First, a normalization scheme is applied. All maps are normalized to the same value range by applying a peak-to-peak normalization according to Eq. 3.5.

Second, a weight $w(.)$ is computed as a function of M and $\overline{m}$, which are respectively the global maximum value and the average of all the other local maxima:

$$w_1(C) = (M - \overline{m})^2. \tag{3.6}$$

Third, the map transform $\mathcal{N}_{lin}(.)$ is applied. Each map is multiplied by its corresponding weight:

$$\mathcal{N}_{lin}(c) = w_1(C) \cdot c. \tag{3.7}$$

Formally, the content-based global amplification map transform $\mathcal{N}_{lin/PP}$ is finally defined as:

$$\mathcal{N}_{lin/PP}(c) = \mathcal{N}_{lin}(\mathcal{N}_{PP}(c)). \tag{3.8}$$

As mentioned in [1], this strategy compares the maximum activity in the entire map to the average activity. In other words, the weight measures how different the most active location is from the average. It is computationally very simple and thus ideal for real-time implementation. Three drawbacks are mentioned: first, the strategy is not biologically plausible, since global maximum computation is not possible using local connection of neurons. Second, this strategy is not able to enhance a feature map in which a unique location is significantly more salient than all other. The unique salient location is generally surrounded by a noise background, which should be ideally suppressed. Third, the corresponding weight of a map with two equally strong spots without activity in the background would suppress the map, while both spots are expected to be salient. The authors therefore propose a sophisticated iterative approach, that is presented in 3.3.4.

We briefly mention that there exists in the literature other ways to define the weight to overcome the previous drawback [10]. We also propose to compute the weight as follow:

$$w_2(C) = \frac{c_{max}}{\bar{c}}, \tag{3.9}$$

where c_{max} and $\bar{c}$ are respectively the maximum value and the mean value of C. In this way, the weight $w_2(.)$ has the advantage to promote a map with two equally strong spots without activity in the background, while $w_1(.)$ would suppress the map.

We note that when the conspicuity map is normalized, its global maxima value is directly equal to the maximum value range.

Another non-mentioned drawback is the use of a peak-to-peak normalization. It scales the value range of each map to its full range, regardless of the effective amplitude of the map. The next section presents the iterative approach.

3.3.4 Non-linear iterative map transform $\mathcal{N}_{iter}$

The non-linear iterative strategy [21] is more biologically plausible and tends to solve some inconveniences mentioned above. It is composed of two steps. First, a normalization scheme is applied. All maps are normalized to the same value range by applying a peak-to-peak normalization (Eq. 3.5). Second, a map transform $\mathcal{N}_{iter}(.)$ is applied. A filtering based on difference of gaussian (DoG) is applied iteratively to each map according to:

$$\mathcal{N}_{iter}(c) = c_n \quad with \quad C_m = |C_{m-1} + C_{m-1} * DoG - \varepsilon|_{\geq 0}, \quad m = 1...n, \tag{3.10}$$

where the filtering, initiated by $C_0 = C$, iterates n times and produces the iterative mapping $\mathcal{N}_{iter}(C)$. Formally, the iterative non-linear map transform $\mathcal{N}_{iter/PP}$ is finally defined as:

$$\mathcal{N}_{iter/PP}(c) = \mathcal{N}_{iter}(\mathcal{N}_{PP}(c)). \tag{3.11}$$

This non-linear operator promotes the major peak while suppressing less conspicious locations. In addition to the fact that it is inspired from human vision[37], this map transform has the advantage to keep unique salient locations, while suppressing the lesser important values forming the background. It has however the drawback to be time consuming, since the operator is iterative and based on spatial convolution. Further, this strategy uses again the inconvenient peak-to-peak normalization to scale the range value. We mention that both map transforms $\mathcal{N}_{lin}(.)$ and $\mathcal{N}_{iter}(.)$ can be used with other normalization schemes.

To summarize, both strategies above have the drawback to use a peak-to-peak normalization to scale the value range of the maps to be fused. In addition, while the non-linear iterative map transform is more accurate and more biologically plausible, the iterative difference of gaussian filtering is very heavy in term of computation costs.

Therefore, we present our two contributions as alternatives: Subsection 3.3.5 presents the long-term normalization scheme as alternative to the peak-to-peak normalization and Subsection 3.3.6 presents the non-linear exponential map transform as alternative to the iterative non-linear map transform.

3.3.5 Long-term normalization scheme $\mathcal{N}_{LT}$

Considering the general problem of fusing a set of feature maps into a unique map, one must consider the different value ranges of the feature maps and thus a normalization step is mandatory. One straightforward way is to scale each map to the same value range (e.g., between 0 and 1), as performed in the two previous map integration strategies. Such a normalization removes important information about the magnitude on the map. For example, a low feature response will be scaled at a comparable response to a high feature response, which is inappropriate. Before introducing the long-term normalization as alternative, we will see that the map integration strategy is not necessary the same from a level to another one. Indeed, when the integration concerns maps issued from similar features, their value range is similar and the maps can be combined directly without peak-to-peak normalization. This is the case for integrating multiscale maps (Eq. 3.1) and also for the integrating similar features conspicuity maps (Eq. 3.2). However, integrating several cues into the saliency map (Eq. 3.3) is different because the channels intensity, chrominance and orientation have different mechanism extraction and may exhibit completely different value ranges. A normalization step is in this case mandatory. To summarize, using for example the iterative non-linear map transform results in the following strategies for the different levels (Figure 3.1):

$$\begin{cases} \mathcal{N}_{iter/ID}(c) = & \mathcal{N}_{iter}(c) & (multiscale\ level) \\ \mathcal{N}_{iter/ID}(c) = & \mathcal{N}_{iter}(c) & (feature\ conspicuity\ level) \\ \mathcal{N}_{iter/PP}(c) = & \mathcal{N}_{iter}(\mathcal{N}_{PP}(c)) & (cue\ conspicuity\ level), \end{cases} \tag{3.12}$$

where $\mathcal{N}_{iter/ID}(.)$ refers to the iterative map transform without peak-to-peak normalization $\mathcal{N}_{PP}(.)$.

We will see now an alternative to the peak-to-peak normalization. The idea is to normalize each channel with respect to a maximum value which has universal meaning [38, 9]. The procedure, named long-term normalization, scales the cue map with respect to a universal or long-term cue specific maximum $\overline{m}_{cue}$ by

$$\mathcal{N}_{LT}(c) = \frac{c}{\overline{m}_{cue}}. \tag{3.13}$$

Practically, the long-term cue maximum can be estimated for instance by learning from a large set of images. The current procedure computes it from the cue maps C_{cue} of a large set of more than 500 images of various types (lanscapes, traffic, fractals, art, ...) by setting it equal to the average of the cue map maxima.

Another approach as alternative to the peak-to-peak normalization is mentioned in [31]. The value range of each feature map are normalized by using the theoretical maximum of the considered feature computed in a heuristic way.

The next subsection describes an alternative to the iterative non-linear map transform.

3.3.6 Non-linear exponential map transform $\mathcal{N}_{exp}$

As alternative to the more biologically plausible but time consuming iterative non-linear map transform, we propose the non-linear exponential map transform [39, 9]. It is computed in two steps: intra-map competition is performed by applying the exponential operator, then, a weighting scheme for inter-map competition is applied to the map:

$$\mathcal{N}_{exp}(c) = w_2(C_{exp}) \cdot c_{exp} \quad and \quad c_{exp} = c \cdot \left(\frac{c}{c_{max}}\right)^{\gamma}, \tag{3.14}$$

where $w_2(.)$ refers to the weighting scheme that simulates the inter-map competition. The mapping has exponential character imposed by $\gamma > 1$: it promotes the higher conspicuity values and demotes the lower values; it therefore tends to suppress the lesser important values forming the background. Compared to the iterative map transform, it has the advantage to be more efficient in term of computation costs.

3.4 Static model

We have introduced previously three map transforms. The contents-based global amplification $\mathcal{N}_{lin}$ and the non-linear iterative $\mathcal{N}_{iter}$ are issued from the state of the art, while the non-linear exponential $\mathcal{N}_{exp}$ is a proposed alternative to $\mathcal{N}_{iter}$. In addition, we have presented two normalization schemes, the peak-to-peak $\mathcal{N}_{PP}$ and the alternative long-term $\mathcal{N}_{LT}$.

As expected, not all perform equally well. To define the static model that will be integrated in the design of the dynamic model of VA, an evaluation of the different strategies [9] is performed using psycho-physical experiments. We define six map integration strategies issued from the combination of the three map transforms ($\mathcal{N}_{lin}$, $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$) with one of the two normalization schemes ($\mathcal{N}_{PP}$ and $\mathcal{N}_{LT}$), as illustrated in Table 3.1.

Table 3.1: Six map integration strategies issued from the combination of the three map transforms ($\mathcal{N}_{lin}$, $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$) with one of the normalization schemes ($\mathcal{N}_{PP}$ and $\mathcal{N}_{LT}$).

	$\mathcal{N}_{PP}$	$\mathcal{N}_{LT}$
$\mathcal{N}_{lin}$	$\mathcal{M}_1$	$\mathcal{M}_4$
$\mathcal{N}_{iter}$	$\mathcal{M}_2$	$\mathcal{M}_5$
$\mathcal{N}_{exp}$	$\mathcal{M}_3$	$\mathcal{M}_6$

Formally, the six map integration strategies used in the performance evaluation are defined as:

$$Configuration\ \mathcal{M}_1 : \begin{cases} \mathcal{N}_{lin/ID}(c) = & \mathcal{N}_{lin}(c) & (multiscale\ level) \\ \mathcal{N}_{lin/ID}(c) = & \mathcal{N}_{lin}(c) & (feature\ conspicuity\ level) \\ \mathcal{N}_{lin/PP}(c) = & \mathcal{N}_{lin}(\mathcal{N}_{PP}(C)) & (cue\ conspicuity\ level), \end{cases} \tag{3.15}$$

$$Configuration\ \mathcal{M}_2 : \begin{cases} \mathcal{N}_{iter/ID}(c) = & \mathcal{N}_{iter}(c) & (multiscale\ level) \\ \mathcal{N}_{iter/ID}(c) = & \mathcal{N}_{iter}(c) & (feature\ conspicuity\ level) \\ \mathcal{N}_{iter/PP}(c) = & \mathcal{N}_{iter}(\mathcal{N}_{PP}(C)) & (cue\ conspicuity\ level), \end{cases} \tag{3.16}$$

$$Configuration\ \mathcal{M}_3 : \begin{cases} \mathcal{N}_{exp/ID}(c) = & \mathcal{N}_{exp}(c) & (multiscale\ level) \\ \mathcal{N}_{exp/ID}(c) = & \mathcal{N}_{exp}(c) & (feature\ conspicuity\ level) \\ \mathcal{N}_{exp/PP}(c) = & \mathcal{N}_{exp}(\mathcal{N}_{PP}(C)) & (cue\ conspicuity\ level), \end{cases} \tag{3.17}$$

$$Configuration\ \mathcal{M}_4 : \begin{cases} \mathcal{N}_{lin/ID}(c) = & \mathcal{N}_{lin}(c) & (multiscale\ level) \\ \mathcal{N}_{lin/ID}(c) = & \mathcal{N}_{lin}(c) & (feature\ conspicuity\ level) \\ \mathcal{N}_{lin/LT}(c) = & \mathcal{N}_{lin}(\mathcal{N}_{LT}(C)) & (cue\ conspicuity\ level), \end{cases} \tag{3.18}$$

$$Configuration\ \mathcal{M}_5 : \begin{cases} \mathcal{N}_{iter/ID}(c) = & \mathcal{N}_{iter}(c) & (multiscale\ level) \\ \mathcal{N}_{iter/ID}(c) = & \mathcal{N}_{iter}(c) & (feature\ conspicuity\ level) \\ \mathcal{N}_{iter/LT}(c) = & \mathcal{N}_{iter}(\mathcal{N}_{LT}(C)) & (cue\ conspicuity\ level), \end{cases} \tag{3.19}$$

$$Configuration\ \mathcal{M}_6 : \begin{cases} \mathcal{N}_{exp/ID}(c) = & \mathcal{N}_{exp}(c) & (multiscale\ level) \\ \mathcal{N}_{exp/ID}(c) = & \mathcal{N}_{exp}(c) & (feature\ conspicuity\ level) \\ \mathcal{N}_{exp/LT}(c) = & \mathcal{N}_{exp}(\mathcal{N}_{LT}(C)) & (cue\ conspicuity\ level), \end{cases} \tag{3.20}$$

The evaluation of the six mentioned configuration are presented in Chapter 7. The most suitable strategy will be used to define the static part of the dynamic VA model.

To summarize, this chapter presented the static VA model. Several map integration strategies have been considered, combining a map transform with a normalization scheme. Two map transforms ($\mathcal{N}_{lin}$, $\mathcal{N}_{iter}$) are issued from the state of the art, while the non-linear exponential $\mathcal{N}_{exp}(.)$ have been proposed as an alternative to the non-linear iterative $\mathcal{N}_{iter}$ for computational cost issues.

In addition, two normalization schemes have been considered, the inappropriate peak-to-peak $\mathcal{N}_{PP}$ and the proposed alternative long-term $\mathcal{N}_{LT}$. Finally, we defined the six configurations that are considered for the evaluation of Chapter 7 .

While this chapter deals with the static model, the next chapter presents the motion model that will be integrated to the static model to define the dynamic architecture.

Chapter 4

Motion visual attention model

4.1 Chapter introduction

Previously, we described a computer model used to highlight salient locations in still images. Motion is clearly involved in VA mechanisms, therefore a computer model of VA designed for video sequences must consider static information as well as motion information. This chapter investigates the design of the motion VA model.

The design of the proposed models relies on recent findings in neuroscience, specifically on motion processing in the brain. In [29], the authors propose a complete motion representation. Details are given in the state of art (Chapter 2, Subsection 2.2.1). Two categories of motion contrasts are salient in the motion representation, the former in magnitude, the latter in phase (Figure 4.1). The former is discriminant in terms of magnitude. Such contrasts is defined as a difference of speed magnitude between the center and surrounding motion. The latter is discriminant in term of phase, in other word, a difference of speed direction.

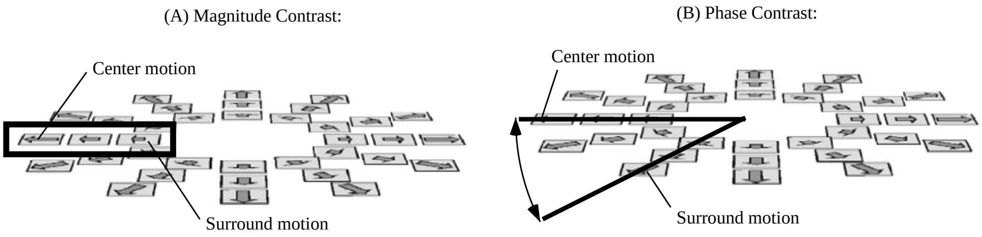


Figure 4.1: Two categories of motion contrasts: (A) magnitude motion contrast, (B) phase motion contrast.

The motion models that will be presented in this chapter, have the capability to highlight the motion contrasts mentioned above. Four models are presented: *the motion magnitude model* highlighting magnitude contrasts, and three motion models highlighting motion contrasts both in phase and magnitude, namely, *the motion model based on direction decomposition*, *the vector motion model*, and *the phase & magnitude motion model*. While the first two models are issued from the state of the art, the two latter models represent our own contributions.

Motion models can be described in two steps:

- The estimation of the motion field, i.e. attribution of a motion vector for each image pixel location. The motion field corresponds to the input feature which will be processed to detect motion contrasts.
- Applying a contrast detection method to the motion field. The method differs according to the considered motion model.

Therefore, the structure of the chapter is defined as follow: first, Section 4.2 describes the motion estimation. Region-based matching technique is used to define the motion field. We will discuss the requirements to estimate motion accurately. Second, Section 4.3 presents the four models highlighting the mentioned motion contrasts. All use a contrast detection method, which is based on difference of Gaussian. Finally, Section 4.4 presents an alternative implementation of the considered models: this method is based on motion pyramid, on which a motion conspicuity operator is applied to detect motion contrast.

4.2 Motion estimation: region-based block matching

From a physical point of view, motion $\mathbf{v}$ at a spatial location $\mathbf{x}$ and time t is defined as the derivative of $\mathbf{x}$ over the time:

$$\mathbf{v}(\mathbf{x}, t) = \lim_{\Delta t \to 0} \frac{\Delta \mathbf{x}}{\Delta t} = \frac{d\mathbf{x}}{dt}. \tag{4.1}$$

Motion can be seen as the ratio of the displacement $\Delta\mathbf{x}$ by an infinitesimal time interval Δt.

From an image processing point of view, each pixel of an image $f(\mathbf{x}, t)$ corresponds to the intensity value, obtained by projection of the 3-D space onto the image plane, sampled at a given time t. In a video sequence, intensity variations over the time are mainly induced by moving patterns in the image. Translation pattern refers to a set of pixels that are moving coherently at a given velocity. Due to the discrete nature of video stream, motion $\mathbf{v}(\mathbf{x})$ is defined as the displacement $\Delta\mathbf{x}$ of the translating pattern between two consecutive images:

$$\mathbf{v} = \frac{\Delta \mathbf{x}}{\Delta t}. \tag{4.2}$$

Motion in the image plane is commonly represented by a vector field $\mathbf{M} = \{\mathbf{v}(\mathbf{x})\}$. Motion can be induced either by the displacement of the objects in the 3-D space, either by the displacement of the camera plane or both.

To compute the motion field, several motion estimation techniques are available in the literature. They are divided into three groups:

- **Region-based matching techniques** [40, 41, 42, 43]. Commonly used in video compression, these methods estimate motion by matching blocks from two consecutive frames, minimizing (or maximizing) a metric, which evaluates a dissimilarity (or similarity). These techniques are investigated in this chapter.

- **Gradient-based matching techniques** [44, 45, 46]. Motion is estimated from derivatives of image intensity over space and time, using the assumption that the image luminance is invariant along the motion trajectories. These techniques will be discussed in the next chapter.

- **Transform-domain techniques** [47, 48, 49, 50]. These methods are based on Fourrier transform or Gabor transform and compute motion using the fact that a translation in the image induce a phase variation in the transform domain.

In this dissertation, we investigate the design of a VA computer model using one of the three techniques. We do not intend to provide a review of all these techniques, since the literature is already well established. Instead, we investigate the requirements of the region-based matching technique in order to design a reliable computer model.

We will now expose the general approach of region-based matching techniques. Such methods are commonly used in video compression [51, 52, 53]. The general idea of the *block matching algorithm (BMA)* is to evaluate a motion vector for each pixel location of the image, by block matching between two successive grayscale frames. Generally the matching is performed by minimizing a distance measure, which evaluates the dissimilarity between two blocks.

The resulting motion field depends on the block partitioning method. Three methods are possible to partition the image into blocks. The first one define a block for each pixel location. We name this method the *high resolution block matching algorithm*. Named as *non-overlapped BMA*, the second one partitions the image into non-overlapped blocks. Finally, the third one partition the image in block that are partially overlapped. This approach named *partial overlapped BMA* can be seen as a good trade-off between the computationally heavy *high resolution BMA* and the less suitable *non-overlapped BMA*.

The remainder of this section is defined as follow: Subsection 4.2.1, 4.2.2 and 4.2.3 present respectively, the *high resolution BMA*, the *non-overlapped BMA* and the *partial overlapped BMA*. Subsection 4.2.4 presents the details of the *BMA* implementation. Then, a performance evaluation of the three methods is covered in Subsection 4.2.5. Finally, the motion field, that is used as input feature of the motion VA model, is defined in Subsection 4.2.6 by combining multiple intermediate motion fields.

4.2.1 High resolution BMA

This method defines a block for each pixel of the image and therefore computes a motion vector for each pixel location. Let us define $\mathbf{x}$ the pixel location of an image of size $n \cdot m$, a square block $\mathbf{W}$ of size w and a square neighborhood $\mathbf{L}$ of size l centered at the $\mathbf{x}$ location (Figure 4.2). For simplicity, the blocks as well as the neighborhood are square.

For each pixel location $\mathbf{x}$, we define a block $B_t(\mathbf{x})$ in the frame (t) and a block $B_{t+1}(\mathbf{x}+\mathbf{k})$ in the frame $(t{+}1)$, which is shifted by $\mathbf{k}$ compared to location of the current block $B_t(\mathbf{x})$:

$$B_t(\mathbf{x}) = \{b_t(\mathbf{x},\mathbf{y})\} \qquad \mathit{and} \qquad B_{t+1}(\mathbf{x}+\mathbf{k}) = \{b_{t+1}(\mathbf{x}+\mathbf{k},\mathbf{y})\}, \tag{4.3}$$

where $\mathbf{y}$ is the block index. For each possible $\mathbf{k}$ in the neighborhood $\mathbf{L}$, both blocks are compared by matching. A motion vector is attributed to the current block, corresponding to the best match according to a distance measure $f(\mathbf{k})$:

$$f(\mathbf{k}) = D(B_t(\mathbf{x}), B_{t+1}(\mathbf{x}+\mathbf{k})). \tag{4.4}$$

$D(.)$ is a distance function which measures the dissimilarity between both blocks B_t and B_{t+1}. Mean square error or mean absolute error are the most usual distances. We choose the second one for its simplicity:

$$f(\mathbf{k}) = \frac{1}{w^2}\sum_{y\epsilon W} |b_t(\mathbf{x},\mathbf{y}) - b_{t+1}(\mathbf{x}+\mathbf{k},\mathbf{y})|. \tag{4.5}$$

The motion vector (or displacement vector) $\mathbf{k}_{min}$ is estimated by finding the minimum of the matching criterion $f(\mathbf{k})$ among all block candidates $\epsilon\ L$:

$$\mathbf{k}_{min} = \arg\min_{\mathbf{k}\epsilon\ L} f(\mathbf{k}). \tag{4.6}$$

We note that alternatives would be to use cross-correlation or correlation coefficient metrics and maximization. Implementation details regarding the minimum detection and vector estimation are described in Subsection 4.2.4. Finally, the steps are repeated for each pixel location, resulting to the 2D motion field $\mathbf{M}$ composed of $n \cdot m$ vectors:

$$\mathbf{M} = \{\mathbf{k}(\mathbf{x})\}. \tag{4.7}$$

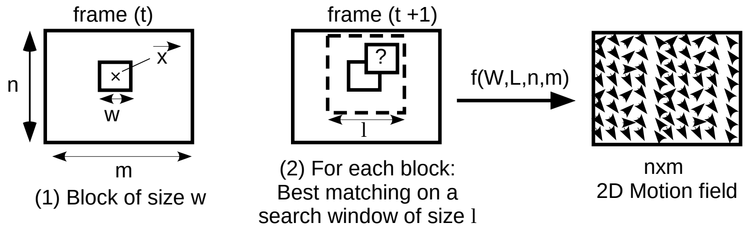


Figure 4.2: Region-based matching technique: $(n \cdot m)$ image size, W size of block and L size of the neighborhood.

Thereby, this method defines a vector for each of the $n \cdot m$ overlapped block and has a computational complexity equal to $C \sim w^2 \cdot l^2 \cdot n \cdot m$, which is very high. On the other hand, this method is the most suitable one for motion estimation. The next method proposes a more efficient alternative to this computationally heavy method.

4.2.2 Non-overlapped BMA

Instead of defining a block for each pixel location, an alternative is to define a reduced number of blocks. The *non-overlapped block matching algorithm* segments the image into a set of non-overlapped blocks (Figure 4.3) and applies the same steps mentioned in the previous subsection. The image is partitioned into blocks $B(i,j)$ of size $w \times w$, with $i = \{1, ..., \frac{n}{w}\}$ and $j = \{1, ..., \frac{m}{w}\}$. This leads to the 2D motion field composed of $\frac{n \cdot m}{w^2}$ vectors.

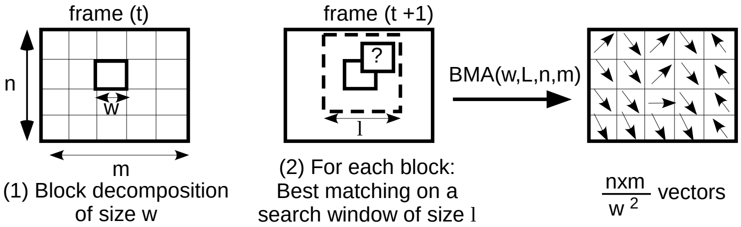

frame (t)
n
w
m
(1) Block decomposition of size w
frame (t +1)
?
l
(2) For each block: Best matching on a search window of size l
BMA(w,L,n,m)
nxm / w 2 vectors

Figure 4.3: Non-overlapped Block matching algorithm (BMA)

Compared to the previous one, this method has the advantage of reducing the complexity to $C \sim l^2 \cdot n \cdot m$. We notice that the complexity becomes independent of the block size w. While it is more efficient in terms of computation costs, it has the inconvenience to decrease the resolution of the motion field, and it therefore provides a bad motion estimation (typically at the border of two non-overlapped blocks). This drawback will be illustrated through some examples in Subsection 4.2.5.

4.2.3 Partial overlapped BMA

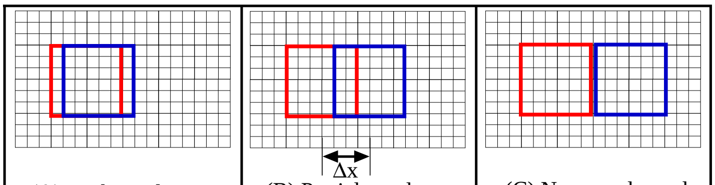
Another possibility which can be seen as a trade-off between both previous methods is to define blocks that are partially overlapped [54, 55]. We name it *partial overlap BMA*. The idea is to reduce the number of blocks compared to the *high resolution BMA* in order to reduce the time computation, while maintaining a minimal block overlapping in order to improve the accuracy compared to the *non-overlapped BMA*.

Let us define Δx the distance between two neighboring blocks. The image is partitioned into overlapped blocks $B(i,j)$ of size $w \times w$, with $i = \{1, ..., \frac{n}{\Delta x}\}$ and $j = \{1, ..., \frac{m}{\Delta x}\}$, resulting to $\frac{n \times m}{\Delta x^2}$ overlapped blocks. The overlapping ratio is defined as the overlapped area divided by block area:

$$\frac{A_{overlap}}{A_{block}} = \frac{w^2 - (2\Delta x - w)^2}{w^2}. \tag{4.8}$$

Regarding the computation costs, the complexity of the *partial overlapped BMA* is proportional to $C \sim \frac{w^2 \cdot l^2 \cdot n \cdot m}{\Delta x^2}$.

To summarize, we have presented three different methods to partition the image into blocks. Figure 4.4 illustrates these three methods: (A) the *high resolution BMA* with its high computational cost and accurate motion estimation, (C) the *non-overlapped BMA* with its efficient computation contrasted with its less accurate motion detection and (B) the *partial overlap BMA* as a trade-off between both previous methods.



	(A) High resolution BMA	(B) Partial overlap BMA	(C) Non-overlapped BMA
Resolution [px]	n x m	$\frac{n \text{ x } m}{\Delta x^2}$	$\frac{n \text{ x } m}{w^2}$
$\frac{A_{overlap}}{A_{block}}$	~100 %	$\frac{w^2\text{-}(2\Delta x\text{-}w)^2}{w^2}$	0 %
Complexity	$w^2 \text{ x } l^2 \text{ x } n \text{ x } m$	$\frac{w^2 \text{ x } l^2 \text{ x } n \text{ x } m}{\Delta x^2}$	$l^2 \text{ x } n \text{ x } m$

Figure 4.4: Block matching algorithm (BMA) using three different block partitioning.

We have described the general principles of region-based matching techniques. Further, three different methods to partition the image have been considered. In Subsection 4.2.4, the details of the implementation of the BMA are exposed. Then Subsection 4.2.5 presents some results illustrating motion estimation based on the three BMA methods.

4.2.4 Implementation of block matching algorithm

The general approach of BMA has been described by three different ways for partitioning the image into blocks. Here we provide the implementation details for computing the motion vector field **M**. The considered implementation estimates motion with a precision of one pixel. Higher precision could be obtained by interpolating the image at fractional pixel location [56]. In addition, the considered implementation is based on full search algorithm, i.e. all possible block candidates inside the search window L are taken into account. We briefly mention that fast search techniques [57, 58] are alternatives to decrease the computation costs. However, these techniques can lead to minimum convergence problems, when the matching criterion $f_{\mathbf{k}}$ is not a monotonic function. For this reason, we use the full search algorithm.

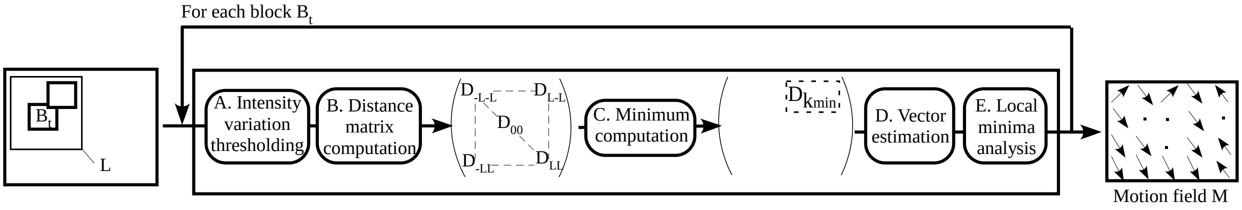


Figure 4.5: Description of the implementation of BMA.

To estimate motion, let us define the motion field $\mathbf{M}$ that can take two possible values for each block location:

$$\mathbf{M}(\mathbf{x}) = \{\mathbf{k}(\mathbf{x})\},\ \ \mathbf{k}(\mathbf{x}) \quad \epsilon \quad \{\mathbb{N}^2, nil\}. \tag{4.9}$$

The motion vector can be either defined as a vector or either undefined (nil). The proposed implementation constitutes five steps (Figure 4.5):

(A) Intensity variation inside the block is estimated by computing standard deviation σ_B of the intensity values. The idea is to discard homogeneous blocks when the variation σ_B is inferior to a given threshold T_σ (typically, 4% of the dynamic range). Formally:

$$\mathbf{k}(\mathbf{x}) \ \ = nil \ \ if \ \ \sigma_B < T_\sigma. \tag{4.10}$$

The remaining steps are applied to blocks satisfying the preliminary condition ($\sigma_B > T_\sigma$).

(B) The matrix distance is computed using mean absolute error as dissimilarity measure. The distance is computed for all block candidates. This leads to a $l \times l$ distance matrix $D_{\mathbf{k}}(\mathbf{x}) = \{d(\mathbf{k}, \mathbf{x})\}$, resulting from to Equ. 4.5.

(C) Minimum of the matrix $d_{\mathbf{k}_{min}}$ is computed using Equ. 4.6.

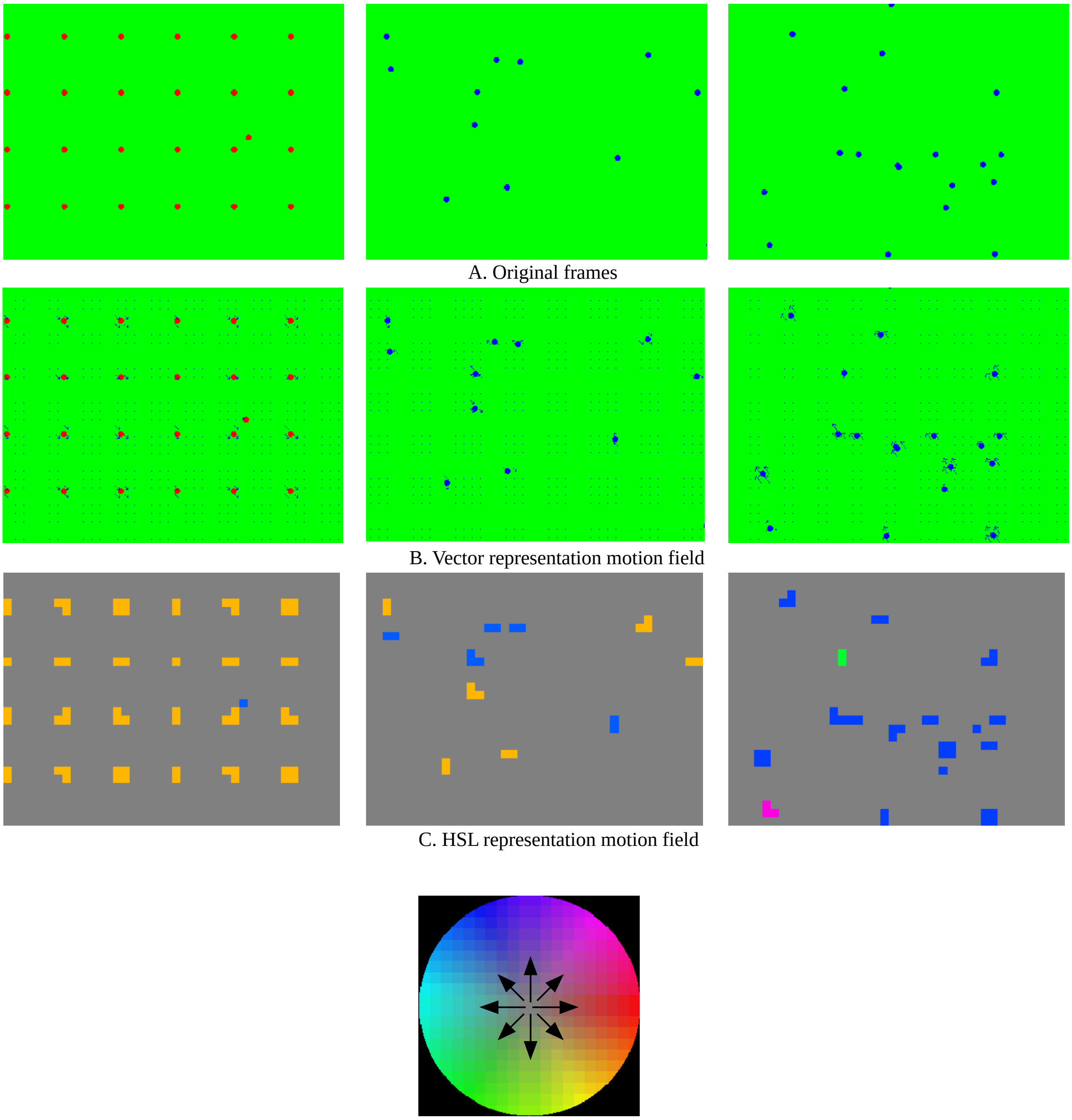
(D) The motion vector is attributed to the current block if the distance minima is inferior to a given threshold, otherwise the motion vector is undefined:

$$\mathbf{k}(\mathbf{x}) \ \ = \begin{cases} \mathbf{k}_{min} & if \quad d_{\mathbf{k}_{min}} < T \\ nil & otherwise. \end{cases} \tag{4.11}$$

(E) A local minima analysis is used to discard the motion vector if the distance matrix contains one or several local minima of same order as the global minima $d_{\mathbf{k}_{min}}$. Local minima d_{local} are defined with the distance matrix as follow:

$$d_{\mathbf{k}} \ \epsilon \ d_{local} \quad if \quad (d_{\mathbf{k}} < T_{percent} \cdot d_{\mathbf{k}_{min}}) \cap (\|\mathbf{k} - \mathbf{k}_{min}\| > T_D). \tag{4.12}$$

The first condition implies the same magnitude order (typically $T_{percent} = 1.5$), while the second one implies local minima detected out of a neigborhood of radius T_D to prevent minima being



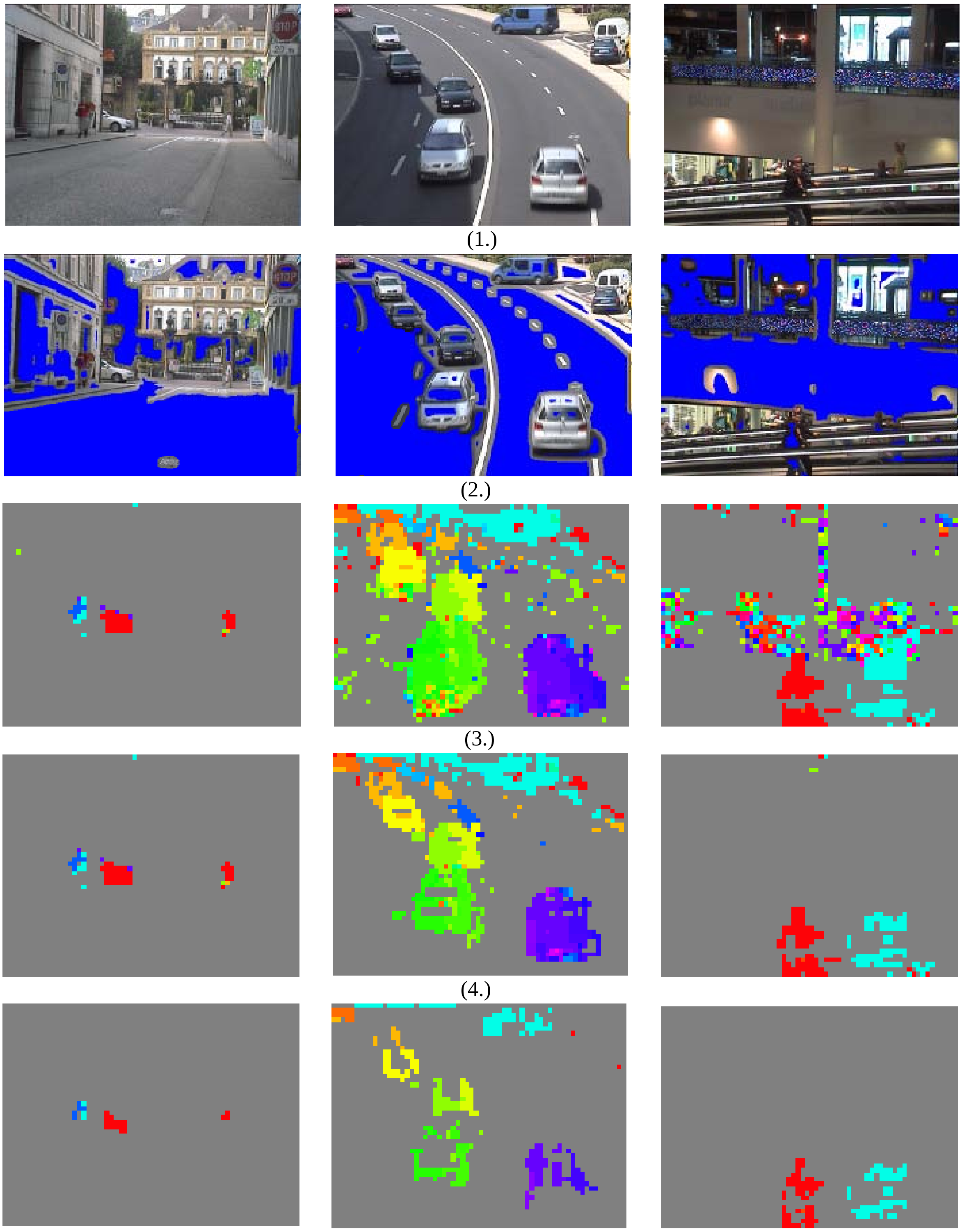
A. Original frames

B. Vector representation motion field

C. HSL representation motion field

D. HSL color representation

Figure 4.6: Motion estimation: an example of non-overlapped BMA with two representation of the motion field. (B.) represents motion field with vectors, while (C.) with HSL color space.



(1.)

(2.)

(3.)

(4.)

(5.)

Figure 4.7: Illustration of the BMA implementation: (1.) original frames, (2.) intensity variation thresholding, (3.) motion field without step (A) and (E), (4.) motion field without step (E), (5.) motion field including all steps. Motion fields are represented with HSL colorspace.

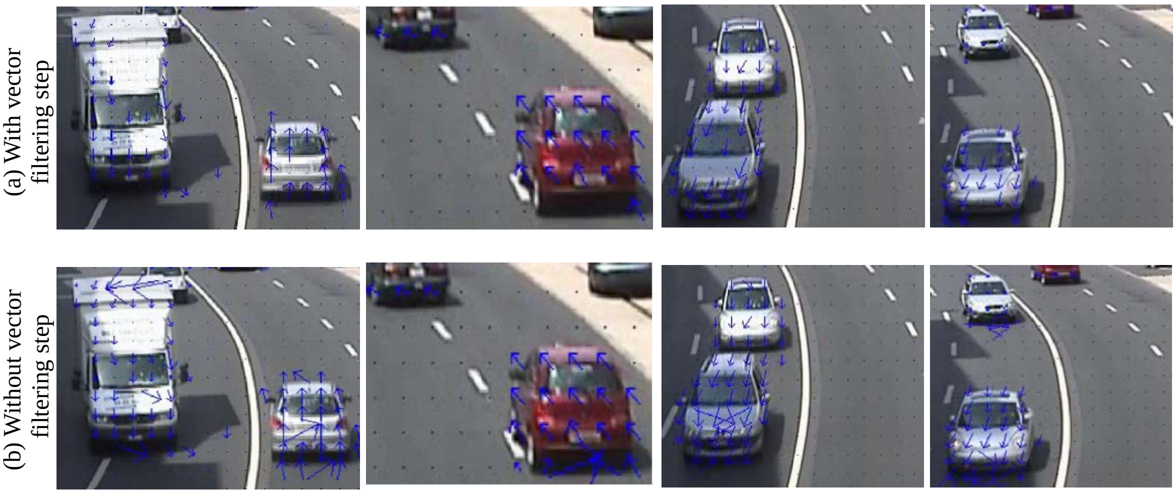

(a) With vector filtering step
(b) Without vector filtering step

Figure 4.8: Examples of motion estimation with and without the local minima analysis step (E).

close to the global minimum. Finally, the motion vector field is computed by applying the steps (A) to (E) for each block B_t.

In order to visualize the motion field, we represent it according to two distinct representations: the vector field representation and the HSL color representation. The former is straightforward and prints a motion vector at each block location. The latter uses HSL color to represent the motion field. Vector phase corresponds to hue, vector norm to saturation and luminance is fixed to a constant value. HSL color space is presented in Figure 4.6 (D). Frames of three different synthetic sequences illustrate the two representations of motion field based on non-overlapped BMA.

Next we present in Figure 4.7 some results of the BMA implementation, including intermediate results illustrating the different steps. Partial overlapped BMA has been used. First, in Figure 4.7 (2.), we illustrate step (A), which discards homogeneous blocks. Block locations that are discarded are represented in blue. The remaining block locations that satisfy the condition have sufficient intensity contrast and are therefore further processed. This step is mandatory in order to suppress false motion vector due to the absence of contrast. (3.) and (4.) present respectively motion estimation without and with intensity contrast thresholding. We notice the clear improvement of motion field accuracy using step (A). In addition, this step has the advantage to speeds up the computation, especially for images containing large homogeneous regions. We present in (5.) the motion estimation including all the steps. Compared to (3.) and (4.), motion field is more suitable and represents accurately motion sequence content. Finally, Figure 4.8 illustrates the advantage of using the local minima analysis (step (E)).

4.2.5 BMA methods: results and comparison

This subsection presents some results in order to illustrate and compare the three considered BMA methods using different block partitioning (high resolution BMA, partial overlapped BMA and non-overlapped BMA). These results motivate the choice of the motion estimation method that will be used in the motion VA model.

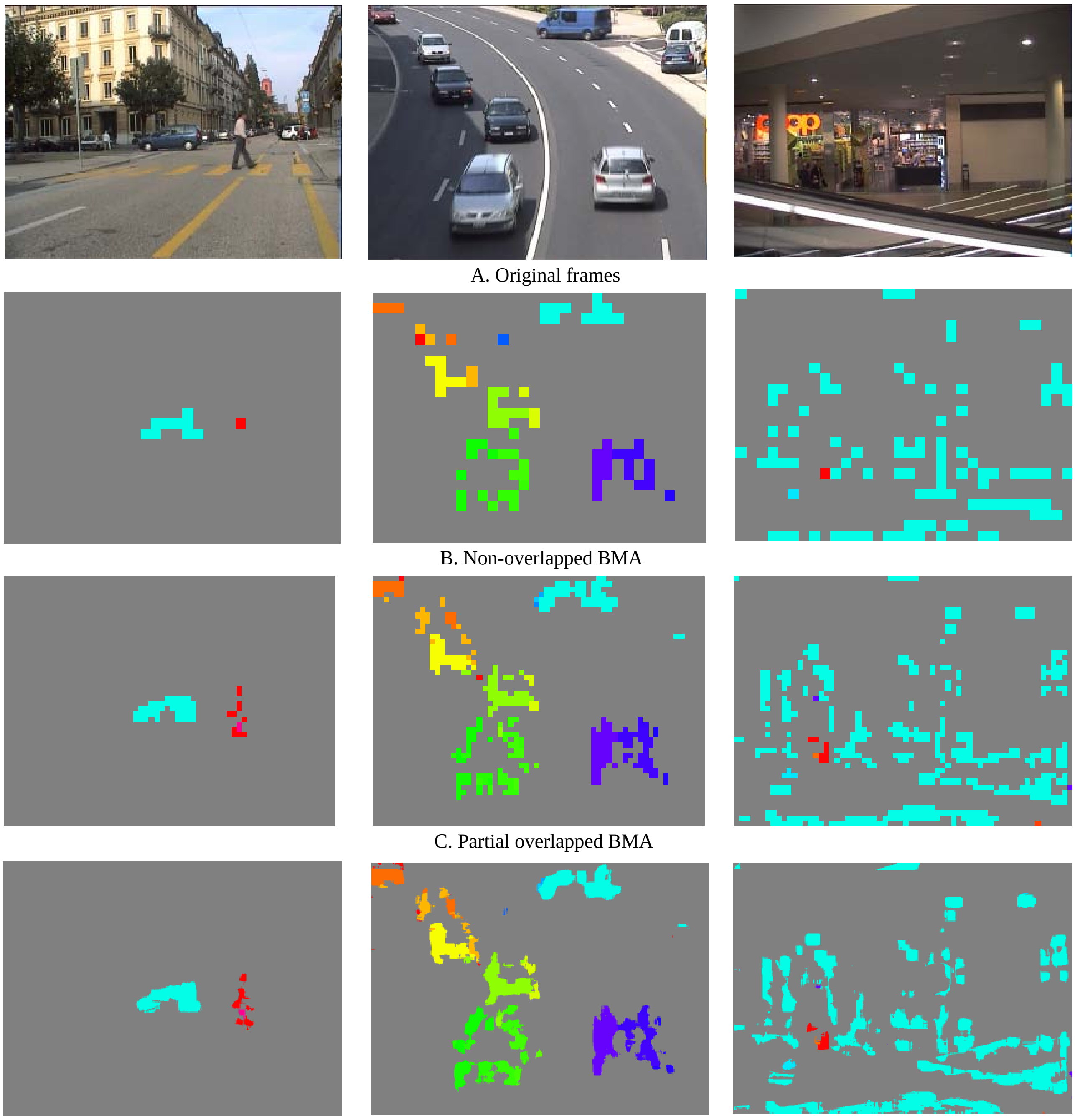
We present in Figure 4.9 motion estimation examples on natural real scenes for the three BMA methods. Since high resolution BMA provides dense motion field, HSL color representation has been used to compare the different BMA methods. Here are some technical details of the considered methods. Image resolution is 512×384 pixels, block size w and search window size l are respectively 16×16 and 15×15 pixels. Motion field resolution is 512×384 for high resolution BMA, 64×48 (50% overlapped area corresponding to $\Delta x = w/2$) and 32×24 for non-overlapped BMA. Figure 4.9 (B), (C) and (D) shows the motion estimation for the three methods.

The results illustrate higher accuracy of the high resolution BMA. Two reasons are exposed. First, with its dense motion field, shape and contours of moving regions are detected accurately, while we can see a degradation for the other methods. Artifacts due to the coarser resolution of non-overlapped and partially overlapped BMAs are visible. As example, we can see in Figure 4.6 (C) non uniform detection of identical circles moving in the same direction. Therefore, high motion resolution ensures accurate motion estimation.

Second, high resolution BMA, which has the highest rate of block overlapping, allows continuous motion detection, i.e. motion detected continuously and independently of the object location relative to the block locations. Typically, non-overlapped block partitioning does not allow to detect motion continuously when moving objects are located at the block borders. Motion continuity is only possible when blocks are partially or fully overlapped. The more overlapped the blocks are, the more continuous the motion is. Figure 4.10 illustrates the motion continuity problem. Motion estimation is presented for three successive frames. The sequence shows a pedestrian crossing the street from left to right. By analyzing motion estimation, we can see higher motion continuity for high resolution and partial overlapped BMAs, while motion estimation is degraded for non-overlapped BMA. Important motion discontinuities are visible.

Finally, Table 4.1 summarizes the evaluation of the three BMA methods. High resolution BMA is the most suitable method to estimate motion, while non-overlapped BMA is the least suitable one. Regarding time computation, the latter is much more efficient compared to the former. With its suitable motion estimation, the partial overlapped BMA is an attractive method, which is trade-off between performances and time computation. In addition, a VA model highlights coarse salient regions (typically the size of the fovea) and therefore it does not require the highest motion resolution. For these reasons, we use the partial overlapped BMA to compute motion estimation in the VA model.

The motion estimation method having been selected, the next subsection defines the motion field that is used as input feature in the motion VA model. We will see that the motion field results from the recombination of multiple intermediate motion fields, in order to cover a large range of displacement scales.



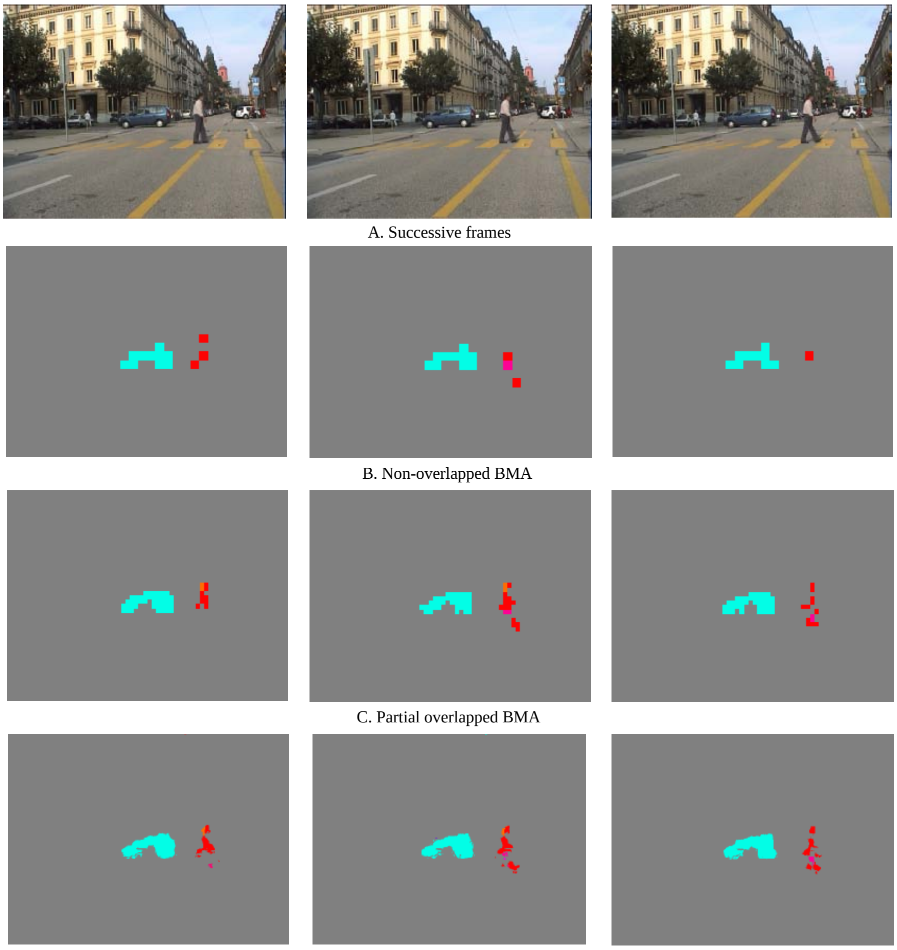
A. Original frames

B. Non-overlapped BMA

C. Partial overlapped BMA

D. high resolution BMA

Figure 4.9: Examples of motion estimation on natural real scenes for the three BMA methods (HSL representation).



A. Successive frames

B. Non-overlapped BMA

C. Partial overlapped BMA

D. High resolution BMA

Figure 4.10: A comparison of BMA methods: accurate motion estimation with high resolution BMA. Motion continuity is more accurate for high resolution and partial overlapped BMAs, while important discontinuities are visible for the non-overlapped BMA.

Table 4.1: Comparison of the *pyramid-based architecture* and *variable block size architecture.*

	Motion continuity accuracy	Shape, contours accuracy	computation Time
Non-overlapped BMA	low	low	fast
partial overlapped BMA	high	intermediate	intermediate
High resolution BMA	high	high	slow

4.2.6 Motion field computation

This subsection defines the motion field $\mathbf{M}$ that is used as input feature in the motion VA model. The motion field $\mathbf{M}$ refers to a 2D vectorial field, represented by a motion vector $\mathbf{v}(\mathbf{x})$ for each pixel location of the image I:

$$\mathbf{M}(\mathbf{x}) = \{\mathbf{v}(\mathbf{x})\}, \ \ \mathbf{v}(\mathbf{x}) \quad \epsilon \quad \{\mathbb{N}^2, nil\}, \quad \forall \quad \mathbf{x} \quad \epsilon \ \ I. \tag{4.13}$$

BMA methods define a motion field that represents moving blocks of specific size. Therefore, estimating the motion field $\mathbf{M}$ directly by using BMA restricts the detection to one specific scale, while video sequences may include moving areas of variable scales (i.e. large and fine areas in motion). The following experiment illustrates the problem.

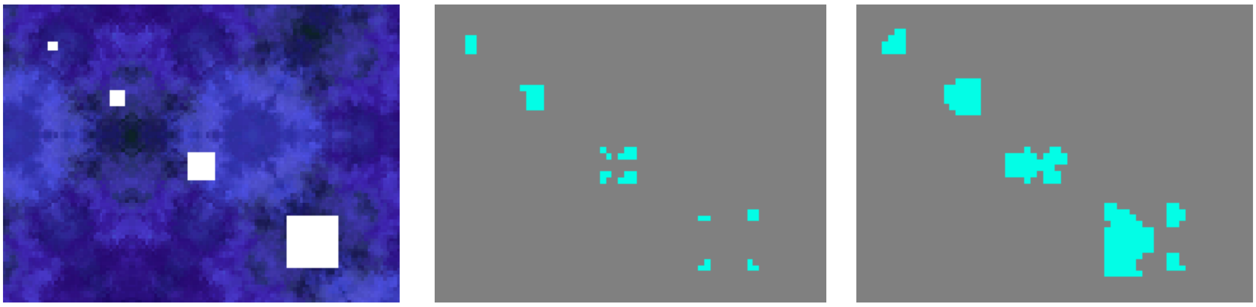
In this experiment, we estimate motion using BMA and we use one given block size. We use a synthetic sequence composed of several squares of variable size in order to investigate the scale effects (Figure 4.11 (a)). The squares are moving from right to left (4 pixels per frame). Motion estimation is computed from partial overlapped BMA using a block size of 8 pixels.

Motion estimation is presented in Figure 4.11 (b). We can see that the moving squares are not detected in the same way. Small squares are fully detected while large ones are detected only in the corners. In addition, motion is not defined inside the large squares. For this reason, estimating the motion field from one motion level is not an appropriate approach.

Therefore, an approach based on several motion levels that highlight a large range of displacement scales is required.

First, we compute several intermediate motion fields, each one representing motion at a given scale. This step can be performed using BMA and varying the block size. Formally, the intermediate maps $\mathbf{M}_i(w_i)$ are computed using BMA with variable block size $w_i = 2^{(i-1)} \cdot w_1$, where w_1 corresponds to the initial block size. The more the index i increases, the larger the block is.

Second, we fuse the intermediate maps into a unique motion field $\mathbf{M}$ using a recombination step (Figure 4.12). The recombination step computes the resulting motion field $\mathbf{M}(i, j)$ by prioritizing the intermediate motion field $\mathbf{M}_i(i, j)$ having the lowest index i, in other words,



(a) Original frame (b) one motion level (c) motion field after recombination

Figure 4.11: Motion field estimation: (b) method based on one motion level, (c) method based on 6 intermediate motion fields after the recombination step.

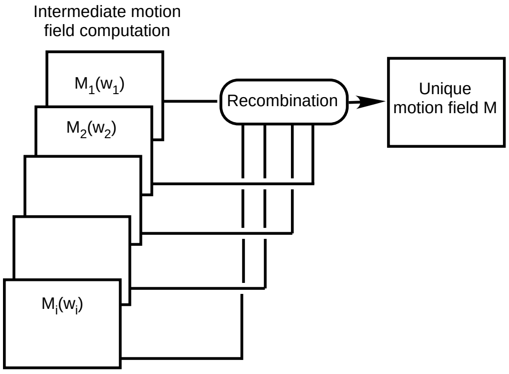


Figure 4.12: Multi-scale motion estimation: the unique motion field **M** results from the fusion of several intermediate motion fields.

small block size has the priority on large block size. Formally:

$$M(\mathbf{x}) \;=\; \begin{cases} \mathbf{M}_1(\mathbf{x}) \;\; if \;\; \mathbf{M}_1(\mathbf{x}) \neq nil \\ \mathbf{M}_2(\mathbf{x}) \;\; if \;\; (\mathbf{M}_2(\mathbf{x}) \neq nil) \cap \;(\mathbf{M}_1(\mathbf{x}) = nil) \\ \mathbf{M}_i(\mathbf{x}) \;\; if \;\; (\mathbf{M}_i(\mathbf{x}) \neq nil) \cap (\mathbf{M}_j(\mathbf{x}) = nil) \;\; \forall \;\; j < i. \end{cases} \tag{4.14}$$

Figure 4.11 (c) shows an example of motion field using the recombination step. We can see that this method improves the detection compared to the method based on one motion level.

4.2.7 Summary: BMA motion estimation

To summarize, this section presented the estimation of the motion field using BMA. Different BMA methods according to three block partitioning methods are presented: (i) the high resolution BMA, (ii) the non-overlapped BMA, and (iii) the partial overlapped BMA. An evaluation has been performed to define the motion estimation method that will be used in the VA model. With its suitable motion estimation, the partial overlapped BMA has been chosen for its trade-off between performances and time computation. Finally, the motion field used as input feature in the motion VA model has been defined as a combination of intermediate motion fields.

The next section presents the four models highlighting the motion contrasts introduced at the beginning of the chapter, namely the magnitude contrast and the phase contrast.

4.3 Motion models: DoG filtering approach

As mentioned previously, motion contrasts of different nature are visually salient. Such motion contrasts can be classified in two distinct categories: (1) motion contrast in magnitude and (2) motion contrast both in phase and magnitude. In this subsection, we present several ways for computing the motion conspicuity related to these two categories. Since visual attention mechanisms are linked to center-surround differences, the four proposed models are based on difference of gaussian filtering (DoG). Each model computes the motion conspicuity from the motion field $\mathbf{M}$ (Equ. 4.13).

While we defined previously the motion field computation, we describe in the next subsections the conspicuity computation methods, which applied on the motion field. First, we present the multi-scale contrast computation. This approach, originally used to compute the static conspicuities, is extended to motion. Then, we present the four motion models. The first one highlights magnitude contrast (Subsection 4.3.2). The three remaining models highlight both phase and magnitude contrasts. The second model is based on a direction decomposition scheme (Subsection 4.3.3), the third one on vectorial convolution (Subsection 4.3.4) and finally, the last one decouples phase and magnitude contrasts by computing separately the phase and magnitude conspicuity (Subsection 4.3.5).

4.3.1 Multi-scale contrast computation

Inspired by the human vision system, center-surround difference refers to contrast between a center and surround region according to a specific feature (Figure 4.13(a)). For the purpose of modeling, the contrast computation leading to the conspicuity map C_{cs} is performed by convolution of the feature $\mathcal{F}$ with a difference of gaussian function $DoG_{(\sigma_c,\alpha)}$, defined respectively by the center standard deviation σ_c and the center-surround ratio α:

$$C_{cs}(\mathcal{F},\sigma_c,\alpha) = |\mathcal{F} **DoG_{(\sigma_c,\alpha)}| = |\mathcal{F}**G_{\sigma_c} - \mathcal{F}**G_{\sigma_s}|, \tag{4.15}$$

$$G_\sigma = \frac{1}{2\pi\sigma^2}e^{\frac{x^2+y^2}{2\sigma^2}}, \quad and \quad \sigma_s = \alpha\sigma_c. \tag{4.16}$$

where the $**$ operator refers to the bi-dimensional convolution. The surround standard deviation σ_s is imposed by σ_c and α.

Therefore, center-surround contrast can be seen as a difference between a center and a surround contribution, each one computed by bi-dimensional convolution of the feature with a gaussian function G_σ of specific size.

In the human visual system, the variable size of the receptive fields allows to highlight contrasts of variable center-surround ratio, i.e. several contrasts between center and surround regions of different size (Figure 4.13(b)). Therefore, the multi-scale contrast computation is performed by computing several intermediate conspicuity maps $C_{cs(i,\alpha)}$ by varying σ_c and the center-surround ratio α:

$$C_{cs(i,\alpha)}(\mathcal{F}) = C_{cs}(\mathcal{F}, \sigma_c(i), \alpha) \tag{4.17}$$

Finally, all intermediate maps $C_{cs(i,\alpha)}$ are integrated into the resulting conspicuity map C. Formally, we define the multi-scale contrast computation as:

$$C = \sum_{i,\alpha} \mathcal{N}(C_{cs(i,\alpha)}(\mathcal{F})). \tag{4.18}$$

Typically, six intermediate maps $C_{cs(i,\alpha)}$ can be used, resulting from three standard deviations $\sigma_c = 2^i \sigma_0, i \epsilon \{0,1,2\}$ combined with two different ratio $\alpha = \{8,16\}$, where σ_0 is the initial standard deviation.

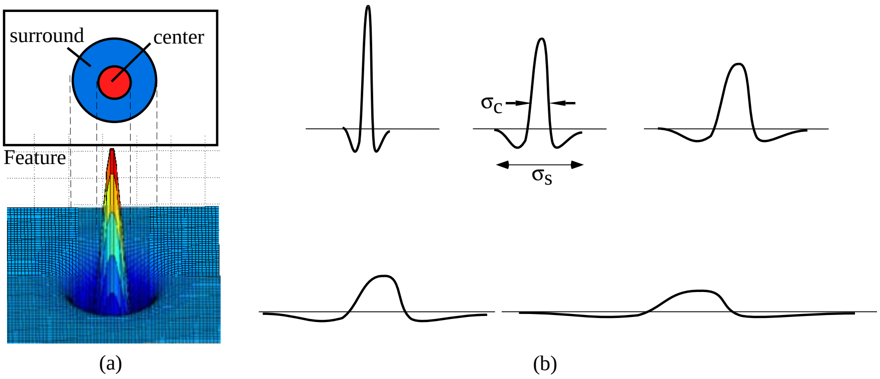


Figure 4.13: (a) Center-surround contrast performed by difference of gaussian (DoG), (b) different size of DoG, which model variable size of the receptive fields.

Thereby, this approach defines the conspicuity map by varying the parameters of the filter, σ_c and α. This approach is commonly used to compute color or intensity conspicuities. Therefore, one straightforward way is to use a similar approach for computing the motion conspicuity.

4.3.2 Motion magnitude model

The motion magnitude model (Figure 4.15 (1)) is composed of one scalar motion conspicuity map C_{magn}, which results from the convolution of one scalar feature $\mathcal{F}_{magn}$.

Let us define a motion vector for each pixel location of the image, corresponding to the motion field $\mathbf{M}$. First, the magnitude feature $\mathcal{F}_{magn}$ is computed as follows:

$$\mathcal{F}_{magn}(\mathbf{x}) = \|\mathbf{M}(\mathbf{x})\|, \tag{4.19}$$

where $\|.\|$ is the vector norm. At this point, we have a scalar feature $\mathcal{F}_{magn}$. Finally, the resulting magnitude conspicuity map C_{magn} is computed according to the multi-scale contrast computation (Equ. 4.18):

$$C_{magn} = \sum_{i,\alpha} \mathcal{N}(C_{cs(i,\alpha)}(\mathcal{F}_{magn})). \tag{4.20}$$

This way for computing the motion conspicuity highlights only magnitude motion contrast and is thus restrictive. Similarly to the conspicuity computation of static features, the motion conspicuity of the magnitude model is scalar.

While static features are scalar, motion is vectorial, and thus is represented by a 2D motion field. This may imply differences regarding conspicuity computation. Indeed, motion contrast includes contrast both in phase and magnitude, while static feature is restricted to magnitude only. One possible way to highlight both contrasts is to define a model selective to specific directions, using direction decomposition. This is the scope of the next subsection.

4.3.3 Motion model based on direction decomposition

Current knowledge in neuroscience mentions that one area in the primate cortex responsible for motion analysis has receptive fields sensitive to specific motion directions. [30] proposes a dynamic model that includes four oriented motion energy maps $(0°, 90°, 180°, 270°)$. Highly biologically motivated, the sophisticated approach proposed in [29] and briefly explained in Subsection 2.2.1 also considers the motion directions.

Here, we consider a motion model composed of a set of m scalar features $\mathcal{F}_{\theta_j}$ sensitive to specific directions $\theta_j = \frac{2\pi}{m}(j-1),\ j \epsilon \{1, 2, ..., m\}$ (Figure 4.15 (2)). Then each feature is transformed using DoG filtering, resulting to a set of m direction conspicuity maps $\mathcal{C}_{\theta_j}$, which are finally integrated into the final conspicuity map.

First, the motion field $\mathbf{M}$ is projected onto the m feature maps $\mathcal{F}_{\theta_j}$. Each motion vector $\mathbf{v}(x, y)$ activates both nearest direction activation maps $\mathcal{F}_{\theta_j}$ using a parallel projection (Figure 4.14). Formally, $\mathcal{F}_{\theta_j}$ is computed as follows:

$$\mathcal{F}_{\theta_j} = proj_{\mathbf{v}_\theta}(\mathbf{v}(x, y)) \quad if \quad \mathbf{v}_\theta \in \{\mathbf{v}_{\theta A}, \mathbf{v}_{\theta B}\}. \tag{4.21}$$

$\mathbf{v}_\theta$ is one of the m-direction and $\{\mathbf{v}_{\theta A}, \mathbf{v}_{\theta B}\}$ are both nearest neighboring directions of the motion vector $\mathbf{v}(x, y)$. At this point, we have a set of m scalar feature maps $\mathcal{F}_{\theta_j}$.

Second, for each direction, we compute a conspicuity map $\mathcal{C}_{\theta_j}$ using the multi-scale contrast computation (Equ. 4.18):

$$C_{\theta_j} = \sum_{i,\alpha} \mathcal{N}(C_{cs(i,\alpha)}(\mathcal{F}_{\theta_j})). \tag{4.22}$$

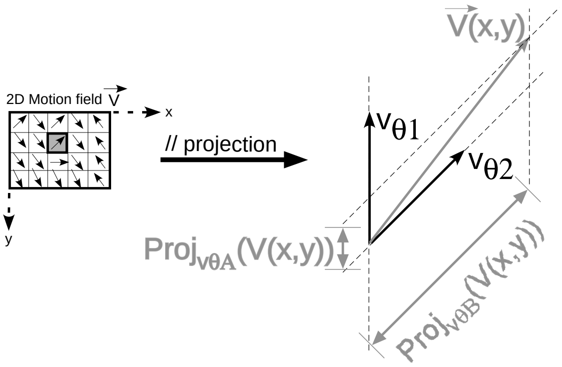

2D Motion field $\vec{V}$
x
y
// projection
$\vec{V}(x,y)$
$v_{\theta 1}$
$v_{\theta 2}$
$Proj_{v\theta A}(V(x,y))$
$Proj_{v\theta B}(V(x,y))$

Figure 4.14: Parallel projection into both nearest direction activation maps. $\mathbf{v}_{\theta A}$ and $\mathbf{v}_{\theta B}$ are both nearest neighboring directions of $\mathbf{v}(x, y)$.

Finally, all maps C_{θ_j} are combined in a competitive way into the motion direction map C_{dir} using the classical map integration scheme:

$$C_{dir} = \sum_j \mathcal{N}(C_{\theta_j}). \tag{4.23}$$

Even if the existence of several direction maps in the model is highly biologically plausible, this approach is heavy in terms of resources (storage of m-activation maps) as well as in terms of computation costs (center-surround filtering based on *DoG* applied to each activation maps). Therefore, this motivates an approach without any direction maps. This will be the scope of the next subsection.

4.3.4 Motion vector model

This subsection presents the second motion conspicuity model highlighting both phase and magnitude contrasts. While the magnitude motion model (Section 4.3.2) applies the convolution operator on a scalar feature (motion magnitude feature), this novel approach applies the convolution operator on a vectorial feature in order to consider both phase and magnitude information (Figure 4.15 (3)). The motion vector model computes the vectorial intermediate conspicuity maps $\mathbf{C}_{cs(i,\alpha)}$ from one vectorial feature, the motion field $\mathbf{M}$.

First, the intermediate conspicuity maps $\mathbf{C}_{cs(i,\alpha)}$ are obtained by convolution of the motion vector field $\mathbf{M}$:

$$\mathbf{C}_{cs_{i,\alpha}}(\mathbf{M}) = \mathbf{C}_{cs}(\mathbf{M}, \sigma_c, \alpha), \tag{4.24}$$

where

$$\mathbf{C}_{cs}(\mathbf{M}, \sigma_c, \alpha) = |\mathbf{M} * *(DoG)_{\sigma_c,\alpha}|. \tag{4.25}$$

Compared to Equ. 4.15, the convolution operates in the vector space.

$\mathbf{C}_{cs(i,\alpha)}$ is vectorial and can be interpreted as a difference between the average center and surround vectors, representing respectively the motion of the center and surround regions. Since the saliency map is scalar, the next step transforms the vectorial map into a scalar one. Formally, the scalar intermediate conspicuity map $C_{cs(i,\alpha)}$ is computed as the norm of $\mathbf{C}_{cs(i,\alpha)}$:

$$C_{cs(i,\alpha)} = \|\mathbf{C}_{cs(i,\alpha)}\|. \tag{4.26}$$

Finally, the motion vector conspicuity C_{vector} is computed by integrating the intermediate scalar conspicuity maps $C_{cs(i,\alpha)}$ using the classical map integration scheme.

$$C_{vector} = \sum_{i,\alpha} \mathcal{N}(C_{cs(i,\alpha)}) \tag{4.27}$$

4.3.5 Phase & magnitude motion model

In this subsection, we present the third motion conspicuity model highlighting both phase and magnitude contrasts. The vector difference can be decomposed in two components, one in phase and the other in magnitude. While both magnitude and phase contributions are not dissociated in the motion vector model, this novel approach decouples the phase and magnitude differences by computing separately the phase and magnitude conspicuity maps.

The phase & magnitude motion model is composed of two scalar conspicuity maps, computed from the motion field $\mathbf{M}$ (Figure 4.15 (4)). First, the magnitude conspicuity map C_{magn} is computed according to Equ. 4.19 and 4.20 (conspicuity model defined in Subsection 4.3.2).

Second, the phase channel is processed. The phase feature map $\mathcal{F}_{phase}$ is computed from the motion field $\mathbf{M}$:

$$\mathcal{F}_{phase}(\mathbf{x}) = arg(\mathbf{M}(\mathbf{x})). \tag{4.28}$$

Then, the resulting phase conspicuity map is computed using the multi-scale contrast computation (Equ. 4.18):

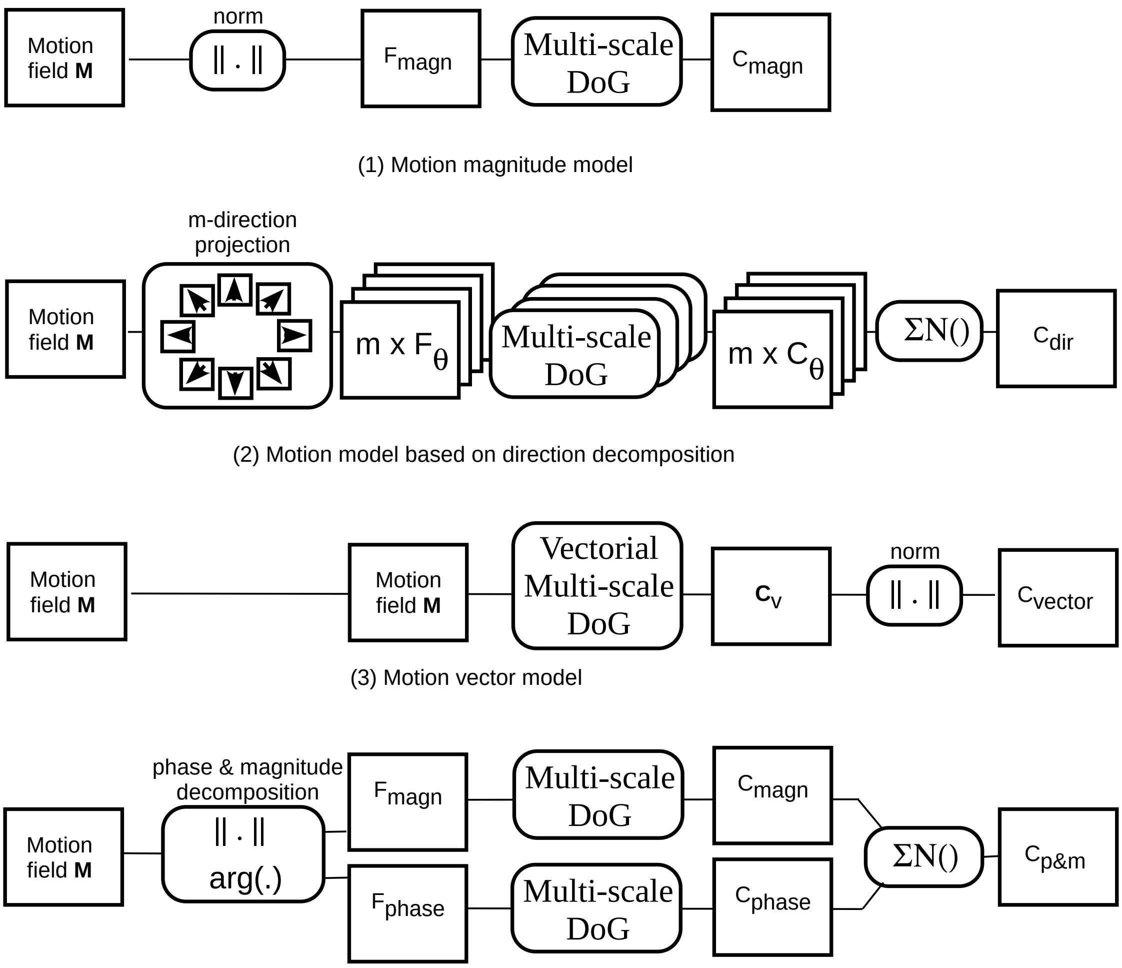
$$C_{phase} = \sum_{i,\alpha} \mathcal{N}(C_{cs(i,\alpha)}(\mathcal{F}_{phase})). \tag{4.29}$$

Finally, the phase & magnitude motion conspicuity $C_{p\&m}$ is obtained by combining both conspicuity maps using the classical map integration scheme:

$$C_{p\&m} = \mathcal{N}(C_{phase}) + \mathcal{N}(C_{magn}). \tag{4.30}$$

4.3.6 Summary: Motion models (DoG approach)

To summarize, four motion models using *DoG* convolution have been presented (Table 4.2 and Figure 4.15). (i) The motion magnitude model, which computes one scalar motion feature, highlights magnitude contrasts. The three other models highlight motion contrasts both in



(1) Motion magnitude model

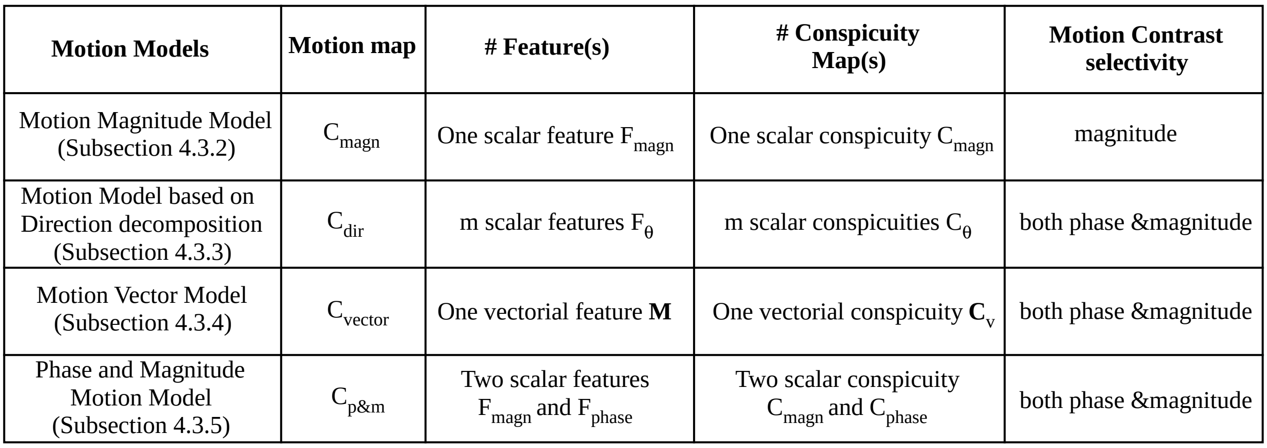
(2) Motion model based on direction decomposition

(3) Motion vector model

(4) Motion phase&magnitude model

Figure 4.15: Four motion models.

Table 4.2: An overview of the four motion models.



Motion Models	Motion map	# Feature(s)	# Conspicuity Map(s)	Motion Contrast selectivity
Motion Magnitude Model (Subsection 4.3.2)	C_{magn}	One scalar feature F_{magn}	One scalar conspicuity C_{magn}	magnitude
Motion Model based on Direction decomposition (Subsection 4.3.3)	C_{dir}	m scalar features F_{θ}	m scalar conspicuities C_{θ}	both phase &magnitude
Motion Vector Model (Subsection 4.3.4)	C_{vector}	One vectorial feature **M**	One vectorial conspicuity $\mathbf{C}_v$	both phase &magnitude
Phase and Magnitude Motion Model (Subsection 4.3.5)	$C_{p\&m}$	Two scalar features F_{magn} and F_{phase}	Two scalar conspicuity C_{magn} and C_{phase}	both phase &magnitude

phase and magnitude. (ii) The motion direction model uses several scalar features sensitive to specific directions. This approach has been proposed previously in [30]. We note that the authors include four oriented motion energy maps in their model, while the proposed one includes eight scalar direction features. (iii) The motion vector model, which is based on vectorial convolution highlights relative motion contrast. (iv) Finally, the phase & magnitude motion model is based on two scalar motion features which decouple phase and magnitude contrasts.

The two last models represent novel approaches as alternative to the biologically-plausible motion direction model, which has the inconvenience to be heavy in term of resources and computation costs. Indeed, the time consuming multi-scale motion contrast computation is repeated for m direction features. Conversely, the motion vector model applies once the multi-scale motion contrast computation in the vector space which is equivalent in term of computation time to the phase & magnitude model, that applies twice the motion contrast computation.

Regarding implementation issues, we mention two drawbacks. First, all the models are based on difference-of-gaussian filtering to detect center-surround contrasts. Such an approach relies on spatial convolution and is therefore heavy in terms of computational complexity. Second, a unique 2D motion field is required, which results from the recombination of several intermediate motion fields. This recombination is not straightforward.

For these reasons, we present in the next section an alternative implementation based on motion pyramid. The pyramid approach combined with cross-scale differences is an efficient approximation of the DoG approach. In addition, this implementation computes directly center-surround differences from the motion pyramid without any recombination step. It constitutes an advantage compared to the DoG approach, which requires that step.

4.4 Motion models: an implementation based on motion pyramid

In the sense of VA, center-surround contrast refers to a difference between a center and surround region. To highlight motion contrasts, one way is to use an approach based on DoG filtering. This method has however the inconvenience of being heavy in terms of computation costs, especially for computing center-surround contrasts at numerous scales. Indeed, the complexity of the spatial convolution is proportional to the square of the kernel size.

In [5], an alternative approach, used to compute static conspicuities, approximates the center-surround filtering by using the image pyramid and cross-scale differences. Regarding motion, a parallel approach can be used similarly. While we presented the motion models based on DoG filtering in Subsection 4.3, we present here an alternative to the *DoG* filtering and propose an implementation based on motion pyramid.

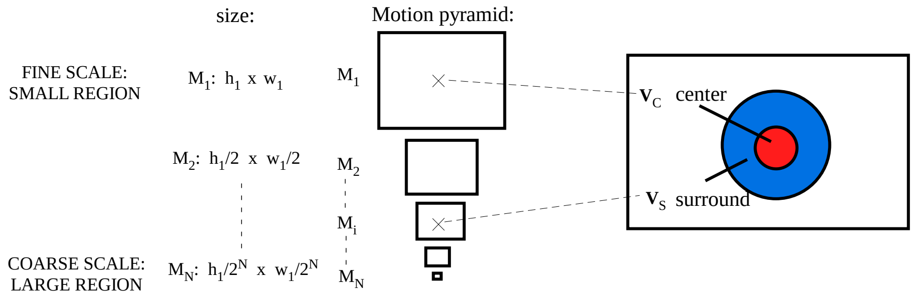


Figure 4.16: Average motion vectors $\mathbf{v}_c$ and $\mathbf{v}_s$ are estimated from the motion pyramid Π_M.

In order to compute motion contrasts, the idea is basically to define two average motion vectors $\mathbf{v}_c$ and $\mathbf{v}_s$ from the motion pyramid, representing respectively the motion of center and surround regions.

The motion pyramid Π_M (Figure 4.16) is composed of N multi-scale motion fields M_i, $i \epsilon \{1, 2, ..., N\}$, corresponding to motion estimation at different scales. Coarse scale maps detect motion of large regions while fine scale maps detect motion of small regions. The initial resolution of the first level $\mathbf{M}_1$ is $h_1 \times w_1$ and the resolution of the other levels is decreasing over the pyramid by factor of 2 between two consecutive levels. The average motion vectors $\mathbf{v}_c$ and $\mathbf{v}_s$ are obtained from $\mathbf{M}_n$ according to their corresponding levels.

In this section, we describe the motion conspicuity computation based on motion pyramid. It is composed of three main parts. First, Subsection 4.4.1 presents the computation of the motion pyramid used to estimate the average motion vectors $\mathbf{v}_c$ and $\mathbf{v}_s$. There are two possibilities to compute the motion pyramid. We will see through an experiment that one method is more appropriate than the other one (Subsection 4.4.2). Second, Subsection 4.4.3 presents the motion conspicuity operators that are used to compute the intermediate conspicuity maps. We will see different operators highlighting the different motion contrasts. These operators are designed with the motion component expressed as the speed magnitude on one hand and as the speed vector on the other hand. Once the basis elements are defined, we present in Subsections 4.4.4 to 4.4.7 the pyramid-based implementation for the four motion models presented in Subsection 4.3.

4.4.1 Motion pyramid computation: two alternatives

As mentioned previously, average motion vectors $\mathbf{v}_c$ and $\mathbf{v}_s$ can be estimated using a motion pyramid.

To compute the motion pyramid Π_M, two alternatives are possible. The first one uses an approach based on image pyramid, while the second one applies BMA to the successive images using variable block size in order to define the different levels of the motion pyramid.

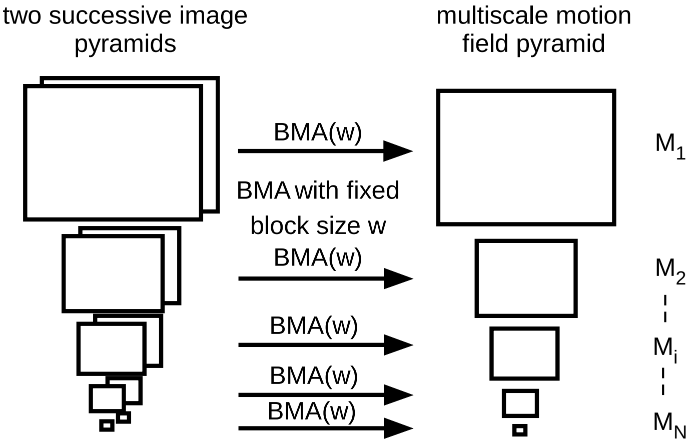

two successive image pyramids
multiscale motion field pyramid
BMA(w)
BMA with fixed block size w
BMA(w)
BMA(w)
BMA(w)
BMA(w)
M1
M2
Mi
MN

Figure 4.17: First approach, *pyramid-based architecture*. The motion pyramid Π_M is computed from the image pyramid by BMA with fixed block size w.

The first one (*pyramid-based architecture*) computes two image pyramids from the successive frame (t) and $(t+1)$ and then applies BMA (one of the three BMA proposed in Subsection 4.2 can be used) with fixed block size w at each level of the pyramid for computing the motion pyramid Π_M (Figure 4.17). In this approach, fine scale maps detect fine displacement of small moving regions while coarse scale maps detect large displacement of large moving regions.

The second one (*variable block size architecture*) is not based on image pyramid and computes multi-scale motion fields $\mathbf{M}_n$ by applying BMA with variable block size (Figure 4.18). The idea is to increase the block size in order to detect large moving regions with large blocks and fine moving regions with fine blocks. The block size depends on the level of $\mathbf{M}_i$ and is computed according to the following equation:

$$w_i = 2^{(i-1)} \cdot w_1, \tag{4.31}$$

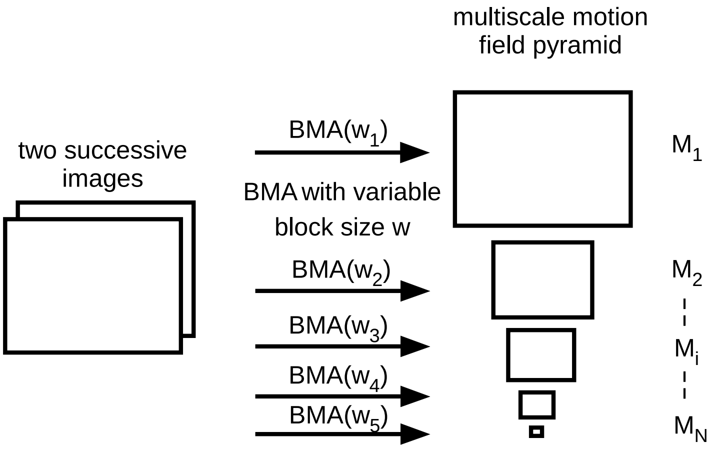


Figure 4.18: Second approach, *variable block size architecture.* The motion pyramid Π_M is computed by BMA with variable block size w.

where w_1 is the initial block size at the first level.

4.4.2 Comparison of the two alternatives

In this subsection, we perform an experiment in order to compare both alternatives used to compute the motion pyramid. The experiment consists in estimating motion from both methods for a set of textures of various patterns that are moving with specific speed ranges. We study the influence of the type of textures as well as the speed magnitude using synthetic video sequences.

Heterogeneous and homogeneous textures (Figure 4.19 (a)(b)) are used. The textures are translated during the sequence with specific speed magnitude. Eight different textures with 3 speed magnitudes (4, 8 and 16 pixels/frame) results in 24 synthetic video sequences. For each sequence, we estimate a motion pyramid Π_M (N=5) using non-overlapped BMA.

The results are the following: when the textures are heterogeneous and the speed range high, both architectures perform identically well. Conversely, both architectures do not perform similarly under other conditions.

First, when the speed range is low, only the *variable block size architecture* estimates motion accurately at each level (Figure 4.19 (e)), while motion accuracy of the *pyramid-based architecture* is decreasing with the pyramid levels (Figure 4.19 (c)). Motion is not detected at coarse levels, which is due to down-sampling effect. Low displacement is not perceptible at coarse image resolution.

Second, when the texture is homogeneous (i.e. low variation in intensity), the second method performs accurately at each level (f), while motion accuracy of the first one is degraded at the coarse levels of the pyramid (d). This is explained by the texture contrast, which decreases with the pyramid level.

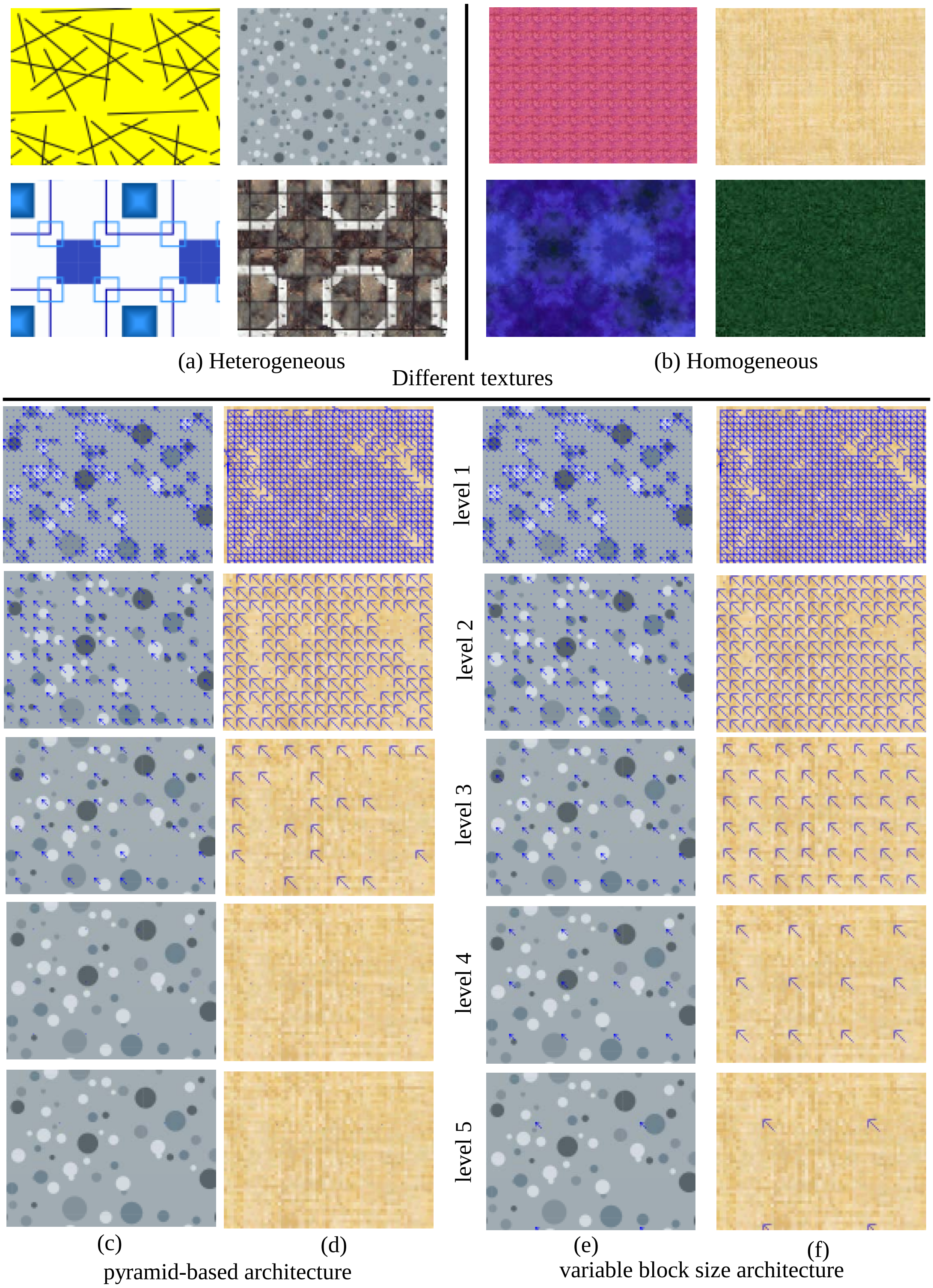
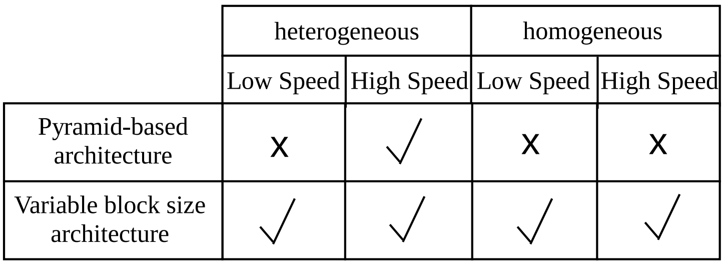


Figure 4.19: Comparison of two motion pyramid computation methods. (a) and (b): textures used to create the synthetic sequences. (c) and (d): two motion pyramids for the *pyramid-based architecture*. (e) and (f): two motion pyramids for the *variable block size architecture*.

Table 4.3: Result overview.



	heterogeneous		homogeneous	
	Low Speed	High Speed	Low Speed	High Speed
Pyramid-based architecture	X	✓	X	X
Variable block size architecture	✓	✓	✓	✓

Table 4.4: Comparison of *pyramid-based architecture* and *variable block size architecture.*

	pyramid-based architecture	*variable block size architecture*
Complexity (non-overlap BMA)	$\sim (\sum_{i=0}^{N-1} \frac{1}{2^{2(i)}}) \cdot n \cdot m$	$\sim N \cdot n \cdot m$
Sensitivity to motion	low sensitivity	high sensitivity

Therefore, this experiment illustrates the higher suitability of the *variable block size architecture* (Table 4.3). This method provides high and constant motion sensitivity over each level, i.e. motion is detected accurately at each level with the same speed range. Conversely, the *pyramid-based architecture* has a low sensitivity with low resolution at coarse levels.

On the other hand, the *pyramid-based architecture* has the advantage to be more efficient in terms of computation time. Indeed, the resolution of the image is decreasing over the image pyramid for this architecture, while the image has a constant resolution for the other one. The complexity is therefore lower for the former. Typically, the *pyramid-based architecture* is four times faster than *variable block size architecture* for computing a motion pyramid of eight levels (N=8) using non-overlapped BMA.

Table 4.4 summarizes the advantages and inconveniences of both architectures. To design a suitable computer model, the motion estimation have to be accurate. Specifically, each level of the pyramid requires to detect motion with a constant speed range and with a constant motion resolution Δv. For these reasons, the *BMA with variable block size architecture* is chosen for computing the motion pyramid.

The motion pyramid computation having been defined, the next subsection presents conspicuity operators that are applied to the motion pyramid in order to highlight motion contrasts.

4.4.3 Motion conspicuity operators

Once average motion vectors $\mathbf{v}_c$ and $\mathbf{v}_s$ have been estimated from the motion pyramid, the next step for computing motion conspicuity is to apply on $\mathbf{v}_c$ and $\mathbf{v}_s$ a motion conspicuity operator in order to detect center-surround contrasts. In this subsection, three conspicuity operators of different natures are presented. The first one is the conspicuity magnitude operator A_{cs} and detects motion contrast in magnitude. This operator computes the norm of $\mathbf{v}_c$ and $\mathbf{v}_s$ and absolute difference:

$$A_{cs}(\mathbf{v}_c, \mathbf{v}_s) = \left| \, \|\mathbf{v}_c(x,y)\| - \|\mathbf{v}_s(x,y)\| \, \right|. \tag{4.32}$$

The second one is the phase conspicuity operator P_{cs} and detects motion contrast in phase. It consists in computing the phase of each vector and performing absolute difference. Formally, the conspicuity operator P_{cs} is defined as follows:

$$P_{cs}(\mathbf{v}_c, \mathbf{v}_s) = Compl\left(\left| tan^{-1}\left(\frac{v_{cy}}{v_{cx}}\right) - tan^{-1}\left(\frac{v_{sy}}{v_{sx}}\right)\right|\right), \tag{4.33}$$

$$Compl(\alpha) = \begin{cases} \alpha & if \quad \alpha < \pi \\ (2\pi - \alpha) & otherwise, \end{cases} \tag{4.34}$$

where (v_{cx}, v_{cy}) and (v_{sx}, v_{sy}) are respectively the xy-motion vector components of $\mathbf{v}_c$ and $\mathbf{v}_s$. $Compl(.)$ is an operator used to shift the phase difference when the phase exceed π. This operator detects the phase contrast.

The third one is the vector conspicuity operator and detects motion contrast both in phase and magnitude. This operator computes the norm of the vector difference:

$$D_{cs}(\mathbf{v}_c, \mathbf{v}_s) = \|\mathbf{v}_c(x,y) - \mathbf{v}_s(x,y)\|. \tag{4.35}$$

In the previous subsections, a framework has been defined, including the computation of the motion pyramid, the estimation of the average motion vectors $\mathbf{v}_c$ and $\mathbf{v}_s$(Subsection 4.4.1) and three motion conspicuity operators (Subsection 4.4.3). The next four subsections will describe implementations (based on motion pyramid) of the four motion models previously defined in subsection 4.3.

4.4.4 Motion magnitude model based on motion pyramid

Here, we present an alternative implementation of the model highlighting motion contrast in magnitude (Subsection 4.3.2). Illustrated in Figure 4.20, this implementation computes the conspicuity map from the motion pyramid Π_M. The magnitude conspicuity operator A_{cs} (Equ. 4.32) is applied at the different levels of the pyramid to compute the intermediate conspicuity maps $C_{A_{ij}}$:

$$C_{A_{ij}}(x,y) = \left| \, \|\mathbf{v}_i(x,y)\| - \|\mathbf{v}_j(x,y)\| \, \right|, \tag{4.36}$$

where $\mathbf{v}_i$ is directly the vector of the center level $\mathbf{M}_i$ and $\mathbf{v}_j$ corresponds to the vector after up-sampling of the surround level $\mathbf{M}_j$ at the corresponding resolution. Notice that up-sampling is necessary to perform point-by-point substraction. For a motion pyramid of $n = 6$ levels, each intermediate conspicuity map $C_{A_{ij}}$ is obtained from a center level $i \epsilon \{1, 2, 3, 4\}$ and a surround level $j = i + \delta$ with $\delta \epsilon \{1, 2, 3\}$. δ corresponds to the scale difference between the center and surround level. Therefore, 11 center-surround differences are computed at different scales (1-2, 1-3, 1-4, 2-3, 2-4, 2-5, 3-4, 3-5, 3-6, 4-5, 4-6). Each intermediate conspicuity map has a resolution corresponding to its center level.

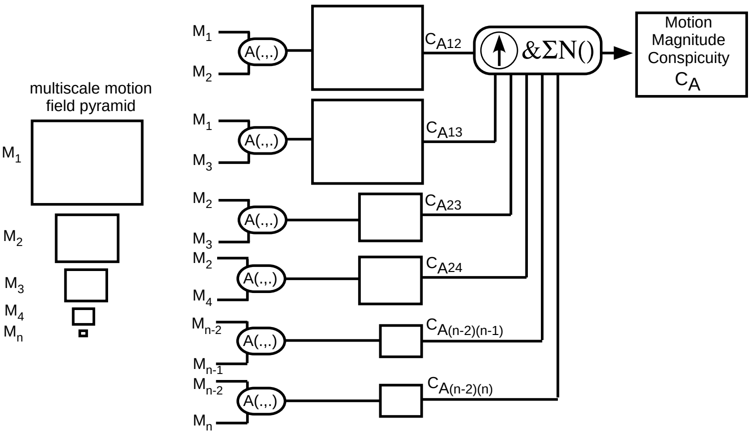


Figure 4.20: the motion magnitude model from motion pyramid.

Finally, all maps are up-sampled at the initial resolution and are integrated into the magnitude conspicuity map C_{magn} in a competitive way using a classical map integration scheme:

$$C_{magn} = \sum_{i,j} \mathcal{N}(C_{A_{ij}}). \tag{4.37}$$

4.4.5 Motion direction model based on motion pyramid

In this subsection, we present an implementation based on motion pyramid for the direction decomposition motion model (Subsection 4.3.3). As reminder, this model uses m direction activation maps, each sensitive to one specific direction (Figure 4.21). Each map is activated by a motion field, resulting to the level recombination of the motion pyramid. Then, a gaussian image pyramid is computed for each direction feature. In order to define the direction conspicuity map $C(\theta)$, the image pyramid is processed by applying a conspicuity operator. Finally, all maps $C(\theta)$ are combined in a competitive scheme into the resulting motion map.

Here are the details of the architecture. First, the motion field $\mathbf{M}$ is computed by recombination of the motion pyramid $\mathbf{M}_n$. The recombination step is defined by Eq. 4.14. Note that each level of the pyramid is preliminary up-sampled at the highest resolution. Second, the motion field $\mathbf{M}$ is projected onto the m activation maps according to Eq. 4.21. Third, a gaussian pyramid is computed for each activation maps M_{v_θ}. Fourth, a direction conspicuity map $C(\theta)$ is computed for each direction by applying the magnitude operator A_{cs} (Eq. 4.32) to the pyramid:

$$C(\theta) = \sum_{i,j} \mathcal{N}(C_{ij}(\theta)). \tag{4.38}$$

Finally, all maps $C(\theta)$ are combined in a competitive way into the motion direction map C_{dir} according to the following equation:

$$C_{dir} = \sum_{\theta} \mathcal{N}(C_\theta). \tag{4.39}$$

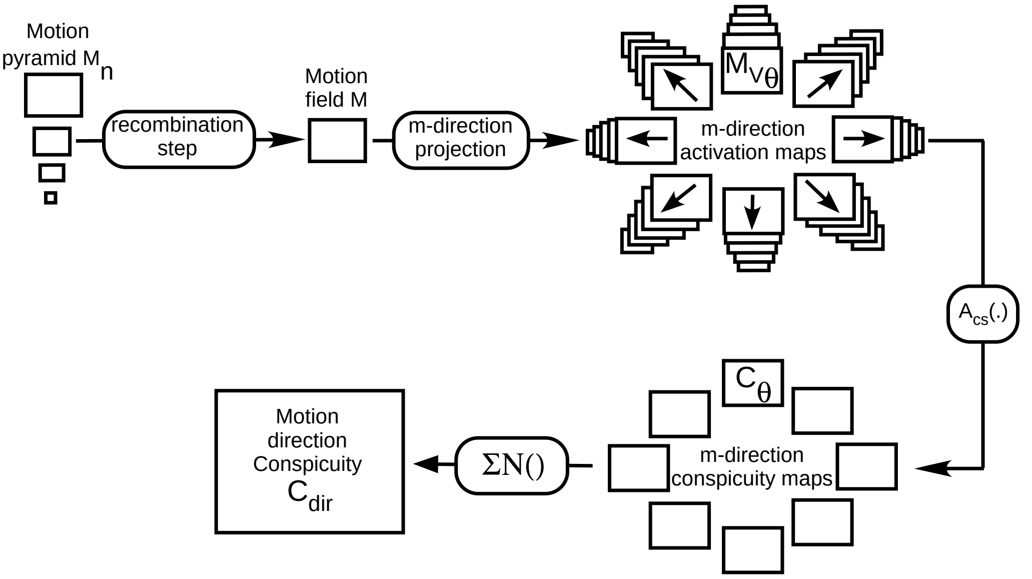

Motion
pyramid M n
recombination step
Motion field M
m-direction projection
M vθ
m-direction activation maps
A cs(.)
C θ
m-direction conspicuity maps
ΣN()
Motion direction Conspicuity C dir

Figure 4.21: Direction decomposition conspicuity architecture: m activation maps sensitive to n different scales and m different directions.

4.4.6 Motion vector model based on motion pyramid

In this subsection, we present the alternative implementation of the motion vector model (Subsection 4.3.4). Illustrated in Figure 4.22, this implementation computes the conspicuity map from the motion pyramid $\mathbf{M}_n$. The vector conspicuity operator D_{cs} (Equ. 4.35) is applied at the different levels of the pyramid to compute the intermediate conspicuity maps $C_{D_{ij}}$:

$$C_{D_{ij}}(x, y) = \|\mathbf{v}_i(x, y) - \mathbf{v}_j(x, y)\|, \tag{4.40}$$

where $\mathbf{v}_i$ is directly the vector of the center level $\mathbf{M}_i$ and $\mathbf{v}_j$ corresponds to the vector after up-sampling of the surround level $\mathbf{M}_j$ at the corresponding resolution. Notice that up-sampling is necessary to perform point-by-point substraction. Finally, all maps are up-sampled at the initial resolution and are integrated into the vector conspicuity map C_{vector} using a classical map integration scheme:

$$C_{vector} = \sum_{i,j} \mathcal{N}(C_{D_{ij}}). \tag{4.41}$$

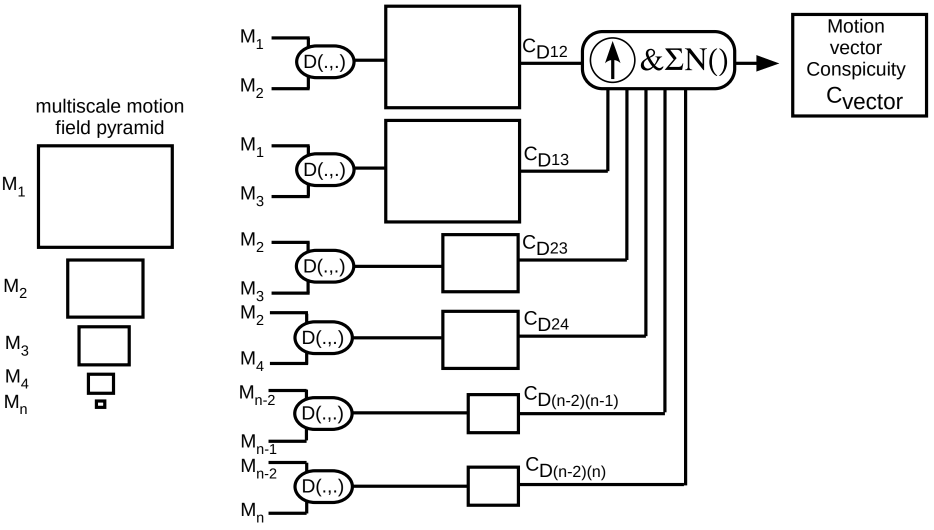


Figure 4.22: Motion map based on a vector conspicuity operator highlighting motion contrast in phase and magnitude.

4.4.7 Phase & magnitude model based on motion pyramid

In this subsection, we present the alternative implementation of the motion model proposed in Subsection 4.3.5. The proposed implementation presented in Figure 4.23 is defined in three steps. The motion phase conspicuity and the motion magnitude conspicuity are computed in parallel by applying respectively the phase conspicuity operator P_{cs} (Equ. 4.33) and the magnitude conspicuity operator A_{cs} (Equ. 4.32) to the multi-scale motion pyramid $\mathbf{M}_n$. The intermediate phase conspicuity maps P_{ij} highlight center-surround phase differences between a center level i and a surround level j. Formally, P_{ij} is defined as follow:

$$P_{ij}(x,y) = Compl\left(\left|tan^{-1}\left(\frac{v_{iy}}{v_{ix}}\right) - tan^{-1}\left(\frac{v_{jy}}{v_{jx}}\right)\right|\right), \tag{4.42}$$

$$Compl(\alpha) = \left\{ \begin{array}{rl} \alpha & if \quad \alpha < \pi \\ (2\pi - \alpha) & otherwise, \end{array} \right. \tag{4.43}$$

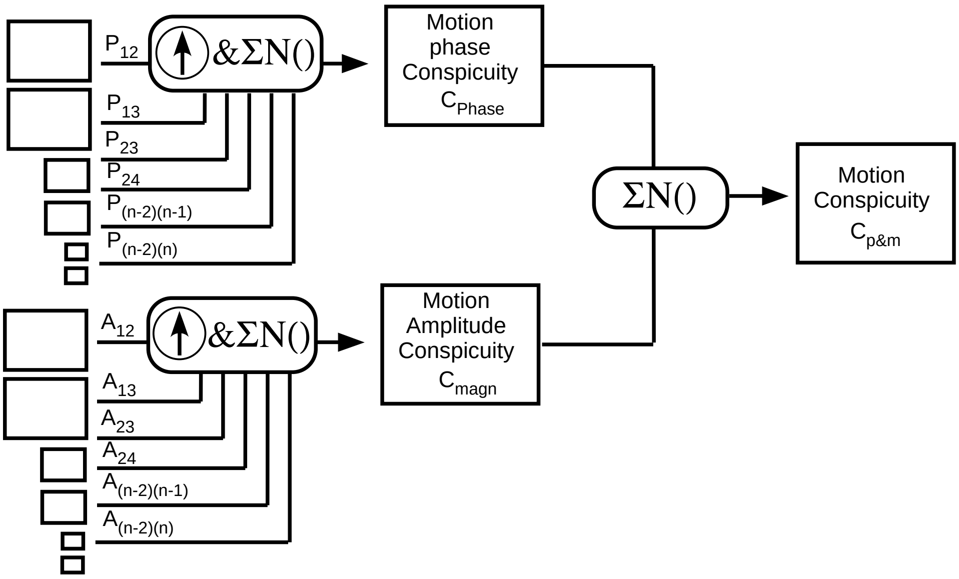


Figure 4.23: The phase & magnitude motion model based on motion pyramid: the resulting motion map is computed from decoupled phase and magnitude contrasts.

where (v_{ix}, v_{iy}) and (v_{jx}, v_{jy}) are respectively the xy-motion vector components of the center level $\mathbf{M}_i$ and up-sampled surround level $\mathbf{M}_j$. *Compl*(.) is an operator used to shift the phase difference when the phase exceed π.

The intermediate magnitude conspicuity map A_{ij} highlights magnitude differences and are defined as follow:

$$A_{ij}(x,y) = \mid \|\mathbf{v}_i(x,y)\| - \|\mathbf{v}_j(x,y)\| \mid , \tag{4.44}$$

where $\mathbf{v}_i$ is directly the vector of the center level $\mathbf{M}_i$ and $\mathbf{v}_j$ corresponds to the vector after up-sampling of the surround level $\mathbf{M}_j$ at the corresponding resolution. Finally, the motion map $C_{p\&m}$ results from the fusion of the phase and magnitude conspicuity maps using the classical competitive scheme:

$$C_{p\&m} = \mathcal{N}(C_{phase} + \mathcal{N}(C_{magn})), \tag{4.45}$$

$$where \qquad C_{phase} = \sum_{i,j} \mathcal{N}(P_{ij}), \quad C_{magn} = \sum_{i,j} \mathcal{N}(A_{ij}). \tag{4.46}$$

4.5 Chapter summary

This chapter discussed the design of the motion VA model. Modeling as well as implementation issues have been considered. Considering the neuroscience point of view, we have exposed two motion contrasts to which the human vision system is sensitive. The first is the contrast in magnitude, the second in phase. The former is discriminant in terms of speed magnitude difference between the center and surrounding motion fields. The latter is discriminant in terms of phase difference, in other word, a difference of speed direction.

The four motion models that have been considered rely on the mentioned motion contrasts. The *motion magnitude model* highlights magnitude contrasts, while the three other models highlight motion contrasts both in phase and magnitude.

As alternative to the *motion model based on direction decomposition*, two novel approaches, namely the *the motion vector model*, and *the phase* & *magnitude motion model* have been introduced. Both models are more optimal in term of computation costs and resources.

All the models are based on difference-of-gaussian filtering (DoG). This approach, previously proposed in [4, 26] to model the center-surround differences, is a plausible way to simulate motion contrasts of the human visual system. However, its inconvenience is the heavy computational complexity. Indeed, large kernels are required to compute the center-surround contrasts by spatial convolution. For this reason, we have introduced an alternative motion pyramid approach, which approximates center-surround filtering by using motion pyramid and cross-scale difference.

An evaluation of the four models will be performed in Chapter 8, by means of psycho-physical experiments. For reason of computation time, we will consider the implementation based on the motion pyramid approach.

The next chapter will discuss the integration of both static and motion model to define the resulting dynamic computer VA model.

Chapter 5

Dynamic model: motion integration schemes

5.1 Chapter introduction

Dynamic visual attention model refers to a computer model capable to highlight the most relevant parts of the scene in the context of video sequences. We have seen previously that both motion and static information are required to design a suitable computer model. While the previous chapters define the static model and several motion model variations, this chapter investigates the integration of both motion and static information.

The classical map integration scheme is based on a weighting scheme, in which each map competes for saliency. An alternative way to integrate motion consists in providing the priority to the motion. In [7], we define several map integration strategies, which are classified in two distinctive schemes: The competitive and the motion priority scheme. We will describe both strategies in this chapter.

The remainder of the chapter is defined as follow. The motion integration based on the competitive scheme is presented in Section 5.2, while the motion priority scheme is presented in Section 5.3.

5.2 Competitive scheme

A straightforward way to integrate motion and static features is to combine all features in a competitive scheme. The saliency map contains a contribution of each feature, which depends on the feature competition. This is the classical way used to integrate features in the static model.

Given a set of conspicuity maps $\mathcal{C}$ to be integrated, the competitive scheme combines all the maps additively into the resulting saliency map S. Formally, the competitive scheme is defined as:

$$S = \sum_{i=1}^{n} \mathcal{N}(\mathcal{C}_i), \tag{5.1}$$

where $\mathcal{C}_i$ refers to one of the n conspicuity maps and $\mathcal{N}()$ is a map integration strategy that is used to simulate intra-map and inter-map competition. We have introduced previously several map integration strategies (Section 3.3), which can be used to integrate conspicuity maps derived from any kind of feature.

The competitive scheme having been defined, we will introduce two possible ways to integrate motion into the static model, which depend on the level of integration. The first one considers motion as an additional cue and the integration is performed at the cue level. All the cues (color, intensity, orientation and motion) are integrated into the saliency map in a competitive way [6, 30, 59]. This strategy, named as *the cue competition scheme* is thus defined as:

$$S_{cuecomp} = \mathcal{N}(C_{color}) + \mathcal{N}(C_{int}) + \mathcal{N}(C_{orient}) + \mathcal{N}(C_{motion}). \tag{5.2}$$

The second alternative integrates motion at a higher level [8]. The motion map is directly combined to the static map according to the competitive scheme. Formally, *the static and motion competition scheme*, is defined as:

$$S_{static\&motion} = \mathcal{N}(S_{static}) + \mathcal{N}(S_{motion}). \tag{5.3}$$

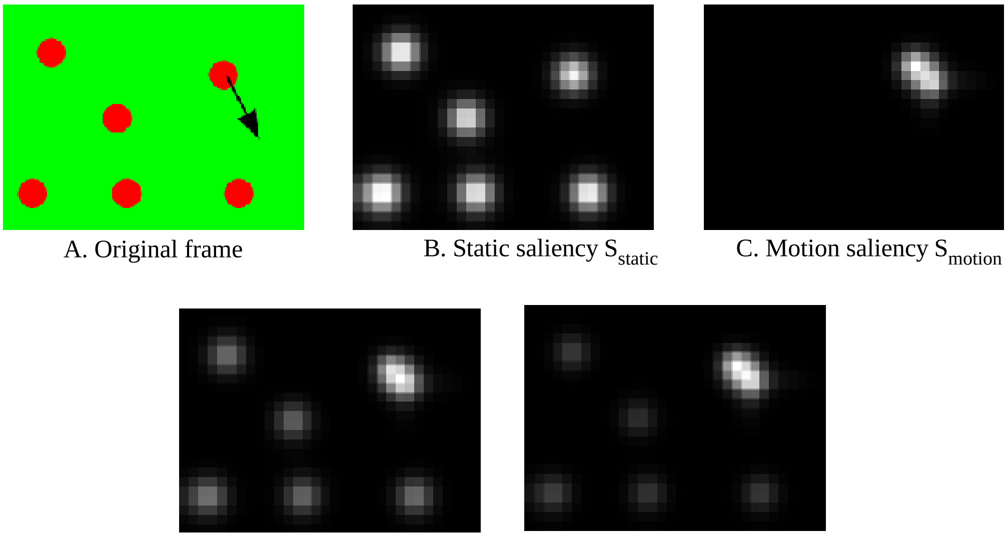
To illustrate the difference, we show the saliency maps issued from both map integration schemes (Figure 5.1). We use a synthetic video sequence (A.) composed of several salient static circles supposed to be distractors, and one moving circle supposed to be salient.

The saliency maps are presented in (D) and (E). In both cases, the most salient region corresponds to the moving stimulus. We can see the difference between both strategies. The static contribution is lower in (E) in comparison to (D). Indeed, the saliency value of the static stimuli is lower. Therefore, this illustrates the higher motion contribution in *the static and motion competition scheme*, which competes for 50% in the saliency map, while motion contribution competes for 25% in *the cue competition scheme.*

5.3 Motion priority scheme

The motion priority scheme combines the static and motion maps by prioritizing motion: in presence of strong salient motion feature (condition A, below), the saliency map $S_{priority}$ is computed by suppressing the static channel. In other words, motion has the priority. In absence of salient motion features (condition C), $S_{priority}$ is computed as the static saliency map. In case of intermediate salient motion features (condition B), $S_{priority}$ is computed with the competitive scheme. Formally, *The motion priority model* is defined as:

$$S_{priority} = \begin{cases} S_{motion} & if \quad (A) \qquad\qquad\quad \Phi(S_{motion}) > T_{motion} \\ \mathcal{N}(S_{static}) + \mathcal{N}(S_{motion}) & if \quad (B) \quad T_{static} < \ \Phi(S_{motion}) < T_{motion} \\ S_{static} & if \quad (C) \qquad\qquad\quad \Phi(S_{motion}) < T_{static}. \end{cases} \tag{5.4}$$



A. Original frame

B. Static saliency S_{static}

C. Motion saliency S_{motion}

D. Cue competition scheme: saliency $S_{cue\ comp}$

E. Static&dynamic scheme: saliency $S_{static\&motion}$

Figure 5.1: Motion integration in a competitive way: (D.) *the cue competition scheme* and (E.) *the static and motion competition scheme.*

where $\Phi(.)$ is a function, which quantifies the rate of saliency in the motion map. A low rate indicates an homogeneous distribution (numerous peaks of activities), while a high rate indicates a sparse distribution, i.e. a few salient locations. $\Phi(.)$ can be estimated as follow:

$$\Phi(S_{motion}) = \frac{1}{n_{spot}}, \tag{5.5}$$

where n_{spot} is the number of salient spot locations superior to a given saliency value. The spot selection is performed by winner-take-all mechanisms (WTA) and inhibition of return (IOR) (Section 3.2). Therefore, a high rate Φ indicates strong conspicuous motion while a low rate low conspicuous motion. We note that the weight function $w(.)$ used in the map integration strategies (Subsection 3.3.3) is another alternative to measure the rate of saliency.

Figure 5.2 illustrates the different modes of the priority model, using three different situations (A) to (C). The original frames 1., the static map 2., the motion map 3., and finally, the dynamic saliency map 4. are presented for the three different situations. In (A), the sequence content is composed of one salient moving circles among a set of static circles (distractors). Motion is therefore highly conspicious (high saliency rate Φ) and the integration scheme provide the priority to motion. In (B), half of the circles are moving, the others stand still. The motion map has an intermediate rate Φ and the competitive scheme is used. Finally in (C), a set of dots are moving uniformly in the same direction. One dot in the center is more salient according to the color. The rate Φ is low and the integration scheme switches to the static map. Therefore, the

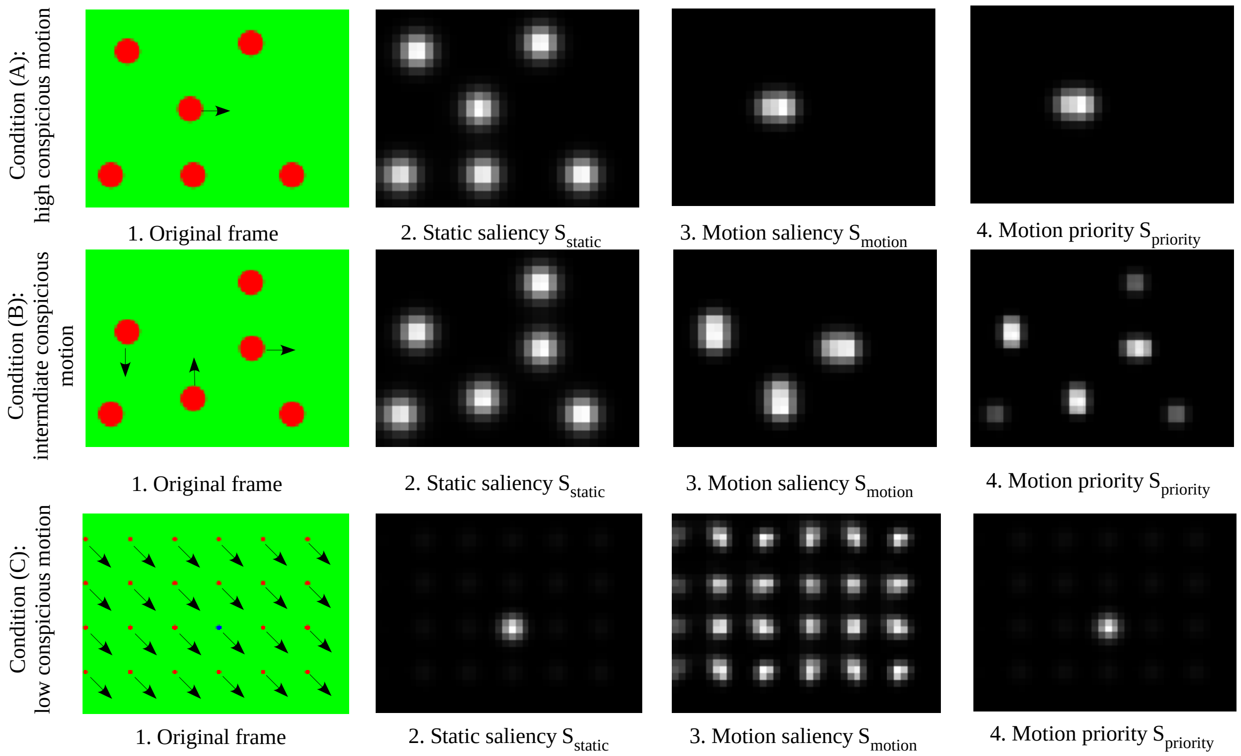

Condition (A): high conspicious motion
1. Original frame
2. Static saliency S_static
3. Motion saliency S_motion
4. Motion priority S_priority
Condition (B): intermdiate conspicious motion
1. Original frame
2. Static saliency S_static
3. Motion saliency S_motion
4. Motion priority S_priority
Condition (C): low conspicious motion
1. Original frame
2. Static saliency S_static
3. Motion saliency S_motion
4. Motion priority S_priority

Figure 5.2: Illustration of the different modes of the priority model: (A) high conspicious motion, (B) intermediate conspicious motion and (C) low conspicious motion.

motion priority scheme acts like a switch between the static and motion map according to the rate of activity Φ, with an transitory regime in-between (competitive scheme).

To summarize, this chapter presented three different ways to integrate both motion and static channels. The *cue competition* and *static & motion competition schemes* apply the classical strategy to integrate the different channels. The alternative *motion priority scheme* provides the priority to motion in case of high conspicuous motion. An evaluation of the three strategies will be performed in Chapter 9, by means of psycho-physical experiments.

Chapter 6

Model validation methodology

6.1 Chapter introduction

This chapter presents the methodology used to evaluate and compare computer models using psycho-physical experiments. All the experiments have been performed in collaboration with the Perception and Eye Movement Laboratory, University of Bern [60].

Since visual attention is tightly linked to the eye and since fixation points correspond to salient locations in images [61], eye movements recording is a suitable means for comparing computer models experimentally. The experimental frame consists in showing a set of images (or video sequences) to a population of human subjects, while an eye tracker system is recording their eye movement patterns. Then resulting fixation points (smooth pursuit is also taken into account in case of video scene) are compared to the prediction of the computer models for the purpose of qualitative and quantitative evaluations.

The current chapter is composed of two main parts. Section 6.2 provides an overview of the methodology used for comparing computer models by using psycho-physical experiments. Section 6.2.1 discusses how to extract eye movement patterns (saccades, fixations, smooth pursuit, blink) from the eye tracker system, while Section 6.2.2 discusses how to construct the human saliency resulting from the experimental data. Finally, Section 6.3 proposes several metrics quantifying similarity between the experimental data and the predictive computer models.

6.2 Overview of the methodology

Psycho-physical experiments involve a population of human subjects that are viewing a set of images or a set of video sequences while an eye-tracker system (Figure 6.1) is recording their eye movement patterns. In the experiment, an infrared-video-based eye tracker (HiSpeedTM, SensoMotoric Instruments GmbH, Teltow, Germany, 240Hz) is used to record eye movements [62]. The system is composed of two fixed infrared camera to track the eye pupil, a screen to display images, and a computer to synchronize, acquire and compute the data. Images and videos are displayed full screen on a 20" color monitor with a refresh rate of 60Hz. The subject sits in front of the screen and the head leans against a fixed structure to prevent from any head movement. The viewing distance is 71.5 cm, resulting to a visual angle of approximately 32° by 24°. The system computes the gaze position at 240Hz by triangulation, knowing the pupil and corneal reflexion positions and other geometric parameters. Gaze position accuracy is approximately 0.5 to 1°, which is strongly dependant on the calibration performed just before the experiment.

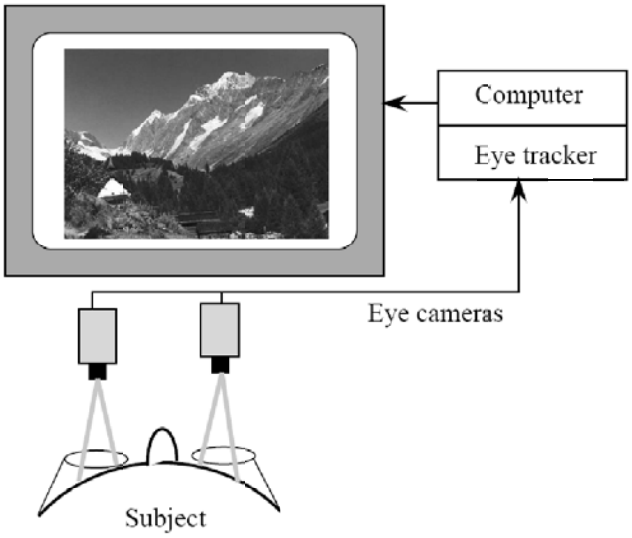


Figure 6.1: Eye tracker system used for recording eye movement patterns.

Typically, eye movement recording of one subject for a video sequence of 10s. duration results in a set of 2400 measurements $M = \{m(i)\} = \{t_i, x_i, y_i\}$ where the time $t_i = \Delta t \cdot i$ is defined by the period Δt and the index of measurement i. x_i and y_i are the spatial coordinates on the screen. The next step described below consists in segmenting the set of measurements M into fixations, smooth pursuits, saccades and blinks in order to discard saccades and blinks in the evaluation.

6.2.1 Eye movement patterns classification

Scan path of the eyes can be represented by three different eye movement patterns: fixations, saccades and smooth pursuits. A visual fixation is defined as the focus of the visual gaze on a specific fixed location. To change from one fixated location to another, the human vision system performs saccades, an involuntary, abrupt, rapid movement of both eyes in order to change the point of fixation. Finally, smooth pursuit is the ability of the vision system to smoothly follow a moving object.

It is admitted that VA is intimately linked to the eye movements and that fixation points correspond to salient location in a still image [61]. In dynamic scene, motion is clearly linked to VA. In [30], the authors confirm experimentally that motion contrast is much more relevant than any other features for predicting human attentional behavior. Therefore, in order to compare a computer VA model with respect to the eye movement patterns, both fixation and smooth pursuit must be taken into account.

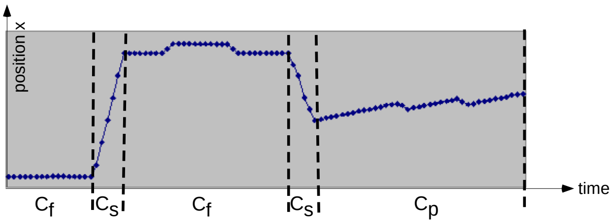


Figure 6.2: An example of classification for a set of data supposing the spatial coordinate y constant over the time. The measurements is classified into subsets C ϵ $\{C_f, C_p, C_s, C_b\}$ corresponding respectively to fixation, smooth pursuit, saccade and blink.

In this subsection, we present a method to extract fixation and smooth pursuit from the eye movements recordings. The classification consists in segmenting the set of measurements M (Figure 6.2) in successive subsets S_C, characterized by its initial index of time i_S, its duration T_S and its class C:

$$S_C(i) = \{i_S, T_S, C\} \qquad with \qquad C \ \epsilon \ \{C_f, C_p, C_s, C_b, \overline{C}\}, \tag{6.1}$$

where C_f, C_p, C_s, C_b refer to fixation, smooth pursuit, saccade and blink periods respectively. Blink corresponds to a period when the eyes are closed or focusing out of the screen. $\overline{C}$ is a class that is used to discard any subset that does not fit to the criteria of the other classes.

Basically, the classifier considers a saccade as a subset of measurements with high local velocity. The subset is considered as a saccade if the velocity $\|v\|$ of the subset is above a given threshold $V_{saccade}$ (typically $25[°/s]$). Formally:

$$\|v\| = \sqrt{v_x^2 + v_y^2}, \tag{6.2}$$

where v_x and v_y are computed independently using linear regression. We note that $\|v\|$ represents the velocity of the whole subset and not the velocity of two successive measurement.

Fixation and smooth pursuit, are defined according to its respective velocity (Eq. 6.2) and duration T_S. In case of fixation, velocity is supposed to be nearly to zero and duration is supposed to be superior to a certain time. Therefore, a subset is considered as fixation if $\|v\|$ is below a speed threshold $V_{fixation}$ (closed to zero) and T_S above a threshold duration $T_{fixation}$.

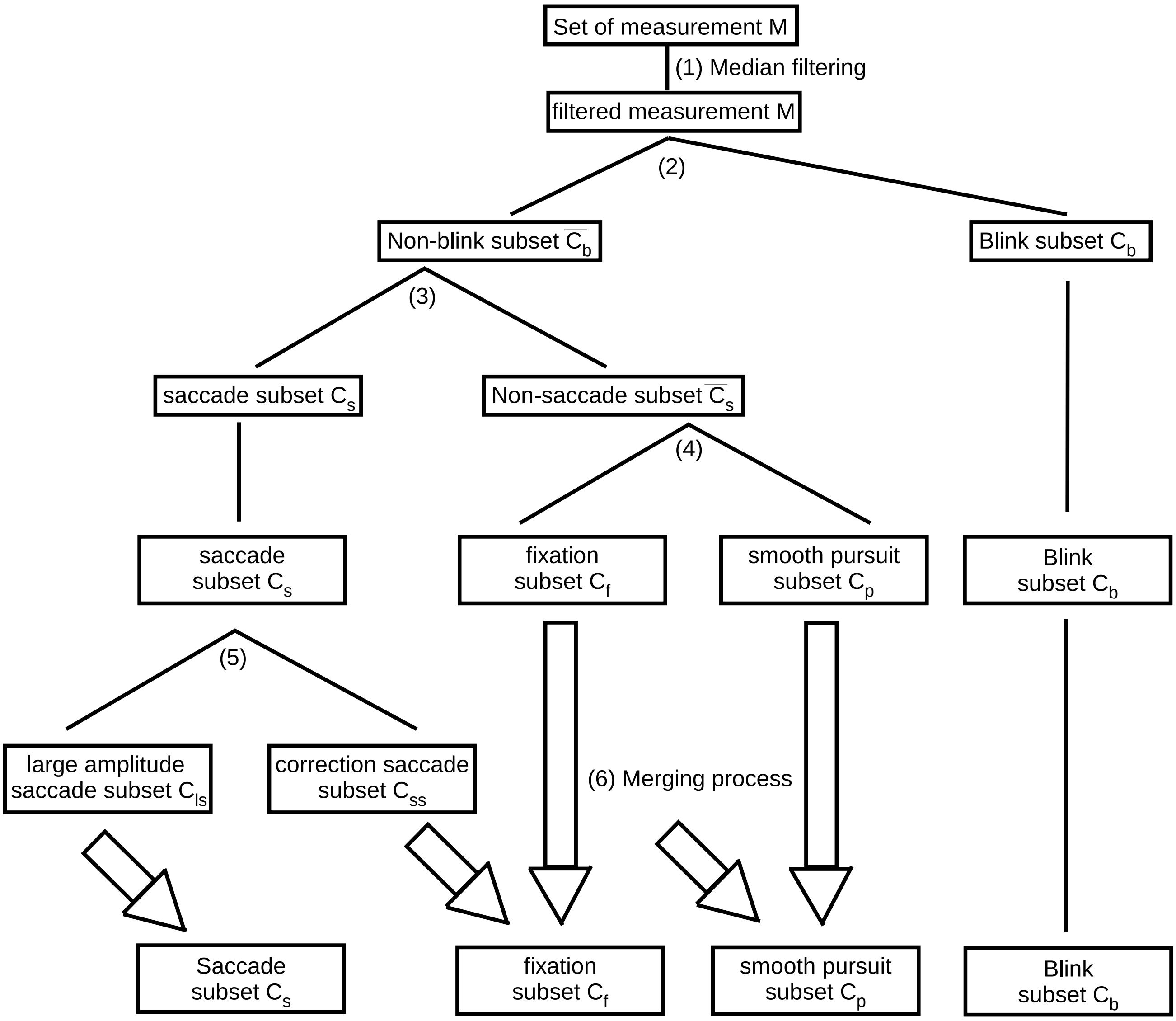


Figure 6.3: Implementation of the classification.

In case of smooth pursuit, velocity depends on the moving object. The velocity is between a minimum speed value $V_{fixation}$ and a maximum speed value $V_{saccade}$. In addition, a minimum duration is required to consider a smooth pursuit.

Regarding the last category, blink subsets are classified automatically by the eye-tracker system, indicating undefined coordinates for the considered blink.

Therefore, saccade, fixation, smooth pursuit and blink are defined as:

$$\begin{cases} saccade: & C = C_s \quad if \quad \|v\| > V_{saccade} & \\ fixation: & C = C_f \quad if \quad \|v\| < V_{fixation} & and \quad T_S > T_{fixation} \\ pursuit: & C = C_p \quad if \quad V_{fixation} < \|v\| < V_{saccade} & and \quad T_S > T_{pursuit} \\ blink: & C = C_b \quad if \quad (x,y)\ undefined. & \end{cases} \tag{6.3}$$

The definition of the four classes having been defined, we briefly describe the implementation. The basic idea is to partition the set of measurements M. The classification can be described in several steps (Figure 6.3):

- Step 1: a median filter is applied to the coordinates of each measurement $m(i)$ to reduce the signal noise (Figure 6.3 (1)).

- Step 2: the set of measurements M is divided in two categories: blink C_b and non-blink $C_{\overline{b}}$ subsets (2). At this point, blink are classified.

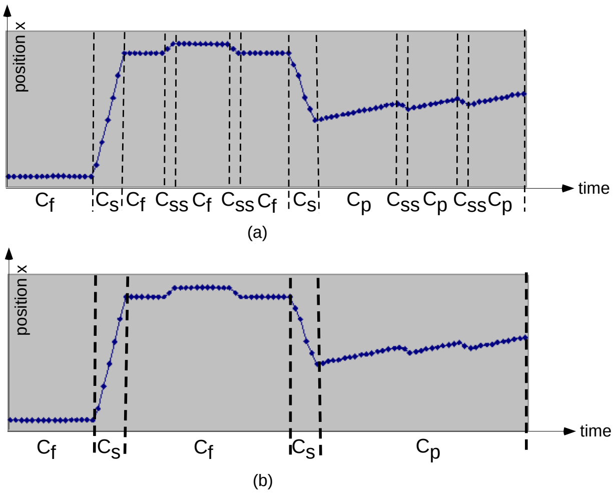
- Step 3: the remaining non-blink $C_{\overline{b}}$ are then classified between two other categories: saccades C_s or non-saccades $C_{\overline{s}}$ subset (3). Saccades and blink are classified.

- Step 4: non-saccades $C_{\overline{s}}$ are classified between fixations C_f or smooth pursuit C_p subset (4). The subsets are now classified in blink, fixation, smooth pursuit and saccade.

  At this point of the classification, we have observed that fixation and smooth pursuit periods are sometime interrupted by correction-saccade (saccades of short amplitude). For this reason, we include two additional steps. The idea is to detect correction saccades (step (5)) and perform a merging process (steps (6)), which filter them and merge interrupted fixations or smooth pursuit (Figure 6.4).

- Step 5: saccades C_s are then classified into correction saccades C_{ss} (short amplitude) and saccades of large amplitude C_{ls}.

- Step 6: finally, the classification in four classes C_f, C_p, C_s, C_b results in a merging process: fixations (resp. smooth pursuit) which are interrupted by correction saccades in-between are merge into a unique fixation (resp. smooth pursuit).



(a)

(b)

Figure 6.4: An illustration of the merging process: correction saccades (C_{ss}) interrupt fixation and smooth pursuit (a). The merging process reconstructs the fixation and smooth pursuit by filtering the correction saccades (b).

Once the classification is performed, blinks and saccades are discarded. Only fixations and smooth pursuits are used in the evaluation. Finally, fixations and smooth pursuits are characterized as follows:

$$fixation : C_f(t_i, D_f, (x_f, y_f)), \quad pursuit : C_p(t_i, D_p, (x_p, y_p), (vx_p, vy_p)), \tag{6.4}$$

where t_i, D_f and (x_f, y_f) refer respectively to the index time, the duration of the fixation and the coordinates, while D_p, (x_p, y_p) and (vx_p, vy_p) refer to the duration of the pursuit, the initial location and the average speed vector.

6.2.2 Human saliency map

This section presents a solution to compute a human saliency map from the experimental data. It is computed under the assumption that it is an integral of gaussian point spread functions $h(x_{k,l})$ at the locations where subjects focus their attention. It is assumed that each location x_k gives rise to a gaussian distributed activity. The width of the gaussian is chosen to approximate the size of the fovea. Formally, the human saliency map is defined as:

$$S_h(\mathbf{x}) = \frac{1}{KL} \sum_{k=1}^{K} \sum_{l=1}^{L} h(\mathbf{x}_{k,l}), \tag{6.5}$$

where k and l refer respectively to the index of fixation and index of human subject while K and L to the total number of fixations and human subjects.

We note that a visual scan path recorded for a still image is significantly different than a scan path for a video sequence. The scan path is mainly composed of fixations with saccades in-between for the former, while it is composed of both fixations and smooth pursuits with saccades in-between for the latter. Therefore, it is plausible to keep only fixations and discard smooth pursuits to compute the human saliency map in case of still images, while both fixations and smooth pursuits are used in case of video sequences. Next we present the computation of the human saliency map for the case of still image, followed by the computation for the case of video sequence.

For the case of still image, a unique human saliency map is computed, representing the average visual behavior of the set of human subjects. In our approach, the set of points x_k corresponds to the n first fixation points of each human subject and the human saliency is computed according to Equ. 6.5.

For the case of video sequence, the scene is changing over the time. To compare experimental and computer data, we compute a human saliency map for each frame of the video sequence, that will be compared with its corresponding computer saliency map. This method requires a large number of human subjects in order to generate an coherent human saliency distribution. The human saliency map $H(\mathbf{x}, t)$ is computed for each frame t:

$$S_h(\mathbf{x}, t) = \frac{1}{KL} \sum_{k=1}^{K} \sum_{l=1}^{L} h(\mathbf{x}_{k,l}(t)), \tag{6.6}$$

where the points $\mathbf{x}_{k,l}(t)$ correspond to the locations of fixation and pursuit that occur at the frame t (Figure 6.5). The locations are computed from Equ. 6.4.

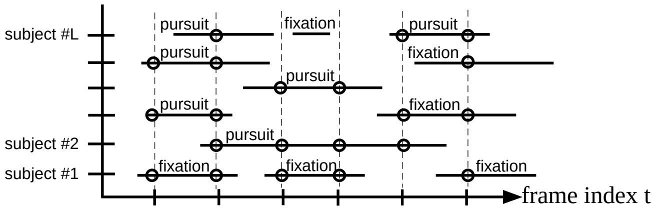
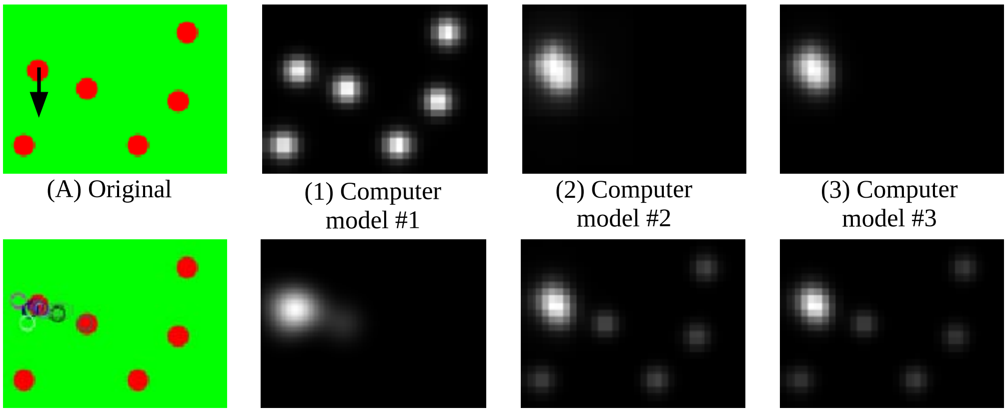


Figure 6.5: Human saliency computation for video sequence: the set of points x_k are sampled at the locations of fixations and pursuits that occur at the time t.

For the purpose of a qualitative comparison, the human saliency map, as well as eye movement patterns mapped onto the image, are compared visually to one or several computer saliency maps according to several computer models. An example comparing five computer models is illustrated in Figure 6.6.

For the purpose of a quantitative comparison, we present in the next section the definition of different metrics that provide a quantitative measure of the similarity between the experimental and computer-based data.



(A) Original

(1) Computer model #1

(2) Computer model #2

(3) Computer model #3

(B) Human Observation

(C) S_{human}

(4) Computer model #4

(5) Computer model #5

Figure 6.6: An example of qualitative comparison: (A) represents the original frame; (B) the human observations; (C) the human saliency map issued from the fixation and smooth pursuit periods; (1) to (5) the saliency maps issued from the five computer models.

6.3 Evaluation Metrics

For quantifying the correspondence of human eye movement patterns with a given computer saliency map, several metrics which are commonly used in the literature are considered. They rely on two distinct approaches. The first one compares two distributions of saliency values. One distribution corresponds to the saliency values sampled at the fixation points $S(\mathbf{x}_{k,l})$, while the other one to the saliency values sampled randomly. From both distributions, a distance measure is computed. We consider one method based on this approach, namely the fixation-to-chance distance [63].

Regarding the second approach, the idea is to transform the experimental data into the form of an experimental saliency map (human saliency map). Then, a metric quantifies the similarity between the experimental map $S_h(\mathbf{x})$ and computer saliency map $S(\mathbf{x})$. The metrics using this approach are the correlation coefficient [36], the Kullback Leibler divergence and ROC curve analysis [31].

6.3.1 Fixation-to-chance distance

The fixation-to-chance distance s_{ftc} quantifies the similarity of a given saliency map S with respect to a set of fixation points (and pursuit points for the case of video sequence). The idea is to define the score as the difference of two contributions. The former is the average saliency $\overline{s}_{fix}$ obtained by sampling the saliency map S at the fixation and pursuit points. The later is the average saliency $\overline{s}$ obtained by a random sampling of S. When the experimental and computer-based data correlate, the computational saliency values at human fixation locations are higher than saliency values sampled randomly.

Compared to [63], the score is normalized so that the integral of the saliency map is constant and thus independent of the scale of the saliency map. Formally, the score s_{ftc} is thus defined as:

$$s_{ftc} = \frac{\overline{s}_{fix} - \overline{s}}{\overline{s}}, \quad where \quad \overline{s}_{fix} = \frac{1}{K}\sum_{k=1}^{K} S(\mathbf{x}_k) \quad and \quad \overline{s} = mean(S). \tag{6.7}$$

This score represents simply the ratio $\frac{\overline{s}_{fix}}{\overline{s}}$ shifted with an offset of -1.

6.3.2 Correlation coefficient

The correlation coefficient s_{cc} measures the degree of linearity between the human saliency map S_h and the computer saliency map S, according to the following equation:

$$s_{cc} = \frac{cov(S_h, S)}{\sigma_{S_h}\sigma_S}, \tag{6.8}$$

where $cov(S_h, S)$ refers to the covariance value between the human saliency S_h and computer saliency S. σ_{S_h} and σ_S are the standard deviations of S_h and S. High similarity provides a score s_{cc} close to 1 while low similarity provides a score s_{cc} close to 0.

6.3.3 Kullback-Leibler divergence

The Kullback-Leibler divergence, denoted s_{KL}, estimates the dissimilarity between two probability density functions. Formally, the score s_{KL} is defined as:

$$s_{KL} = \sum_{\mathbf{x}} p(\mathbf{x}) \cdot ln\left(\frac{p(\mathbf{x})}{p_h(\mathbf{x})}\right), \tag{6.9}$$

where $p(\mathbf{x})$ and $p_h(\mathbf{x})$ are respectively, the probability densities deduced from the computer saliency map S and from the human saliency map S_h:

$$p(\mathbf{x}) = \frac{S(\mathbf{x})}{\sum S(\mathbf{x})} \quad and \quad p_h(\mathbf{x}) = \frac{S_h(\mathbf{x})}{\sum S_h(\mathbf{x})} \tag{6.10}$$

As it is a measure of dissimilarity, a score s_{KL} close to zero indicates that the map S is almost identical to the map S_h.

6.3.4 ROC curve analysis: AUC value

ROC curve analysis [31] estimates the true positive rate (TPR) and false positive rate (FPR), by labeling each pixel location of the computer saliency S and the human saliency maps in two classes (fixated, non-fixated). The computer data (S) is compared to the experimental data (S_h), which is considered as the ground truth.

First, both maps are labeled in two classes $\{f, \overline{f}\}$ by thresholding. Each pixel location is considered as fixated (f) when the saliency value is superior to a given threshold, otherwise it is labeled as non-fixated ($\overline{f}$). Formally, the human saliency map is labeled as follows:

$$L_h(\mathbf{x}) = \begin{cases} f & if \;\; S_h(\mathbf{x}) \geq T_h \\ \overline{f} & otherwise. \end{cases} \tag{6.11}$$

The computer saliency map is labeled in the same way:

$$L(\mathbf{x}) = \begin{cases} f & if \;\; S(\mathbf{x}) \geq T \\ \overline{f} & otherwise. \end{cases} \tag{6.12}$$

Then TPR and FPR are estimated, using L_h as ground truth. A pixel location is true positive if L_h and L are both labeled as fixated, while a location is false positive if L_h is non-fixated and L is fixated. Therefore, a pair of values (TPR,FPR) is computed for a given threshold T. Finally, to estimate the ROC curves, the procedure is applied by varying the threshold value T_h.

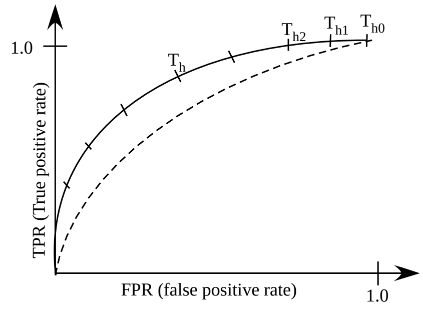


Figure 6.7: ROC curve analysis: the curve is computed by varying the threshold values T_h.

Figure 6.7 shows an example of ROC curves. The more the top left-hand corner the curve approaches, the better the detection is. The ideal discrimination is obtained by a false positive rate equal to 0 and a true positive rate equal to 1. In the example, the model indicated in plain correlates better compared to the model in dash. The more the area under the curve is, the higher the similarity is.

Therefore, in order to assign a quantitative value representing the similarity between the experimental and computer data, the AUC value is computed as the area under the curve.

Four methods have been presented to evaluate quantitatively the correspondences between the experimental and computer data. The quantitative evaluation is performed as follows: for each model, for each sequence and for each time t corresponding to a frame, the considered scores are computed.

Chapter 7

Static model evaluation

7.1 Chapter introduction

This chapter deals with the static model evaluation. Several models are considered according to different map integration strategies. As reminder, six static models ($\mathcal{M}_1$ to $\mathcal{M}_6$) result from the combination of three map transforms $\mathcal{N}_{lin}$, $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$, with one of the normalization schemes $\mathcal{N}_{PP}$ and $\mathcal{N}_{LT}$ (Table 7.1). Psycho-physical experiments are used to assess the model performances.

Table 7.1: Six map integration strategies issued from the combination of the three map transforms ($\mathcal{N}_{lin}$, $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$) with one of the normalization schemes ($\mathcal{N}_{PP}$ and $\mathcal{N}_{LT}$).

	$\mathcal{N}_{PP}$	$\mathcal{N}_{LT}$
$\mathcal{N}_{lin}$	$\mathcal{M}_1$	$\mathcal{M}_4$
$\mathcal{N}_{iter}$	$\mathcal{M}_2$	$\mathcal{M}_5$
$\mathcal{N}_{exp}$	$\mathcal{M}_3$	$\mathcal{M}_6$

Regarding the experimental design, 20 human subjects were viewing the image set, while an eye tracker system was recording their eye movement patterns. All of them have normal or corrected-to-normal acuity, as well as normal color vision. Each image was presented to the subject for a duration of 5 seconds, resulting in an average of 290 fixations per image. The experimental image set consists of 16 color images of various natural scenes.

For each image, the evaluation consists in comparing experimental data (human saliency maps) to computer data (computer saliency maps). On one hand, the experimental data are transformed into an experimental saliency map, named as human saliency map (Subsection 6.2.2). On the other hand, the computer saliency maps are computed according to the considered models. Finally, the human saliency map is compared to the computer saliency maps. Qualitative and quantitative evaluations are performed to define the most suitable model.

This chapter is divided in three parts. First, the qualitative evaluation is presented in Section 7.2. Second, the quantitative evaluation is exposed in Section 7.3. Finally, the third part provides a brief summary of the model evaluation, and concludes by defining the static model used in the dynamic computer VA model.

7.2 Qualitative evaluation

For the purpose of a qualitative comparison, the human saliency map is compared visually to the computer saliency maps issued from the considered models. Subsection 7.2.1 deals with the normalization schemes comparison, while Subsection 7.2.2 deals with the map transforms comparison.

7.2.1 Normalization schemes: $\mathcal{N}_{PP}$ and $\mathcal{N}_{LT}$

This subsection presents the qualitative evaluation comparing the peak-to-peak normalization $\mathcal{N}_{PP}$ to the long-term $\mathcal{N}_{LT}$. Both normalization schemes are compared for each map transform $\mathcal{N}_{lin}$, $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$.

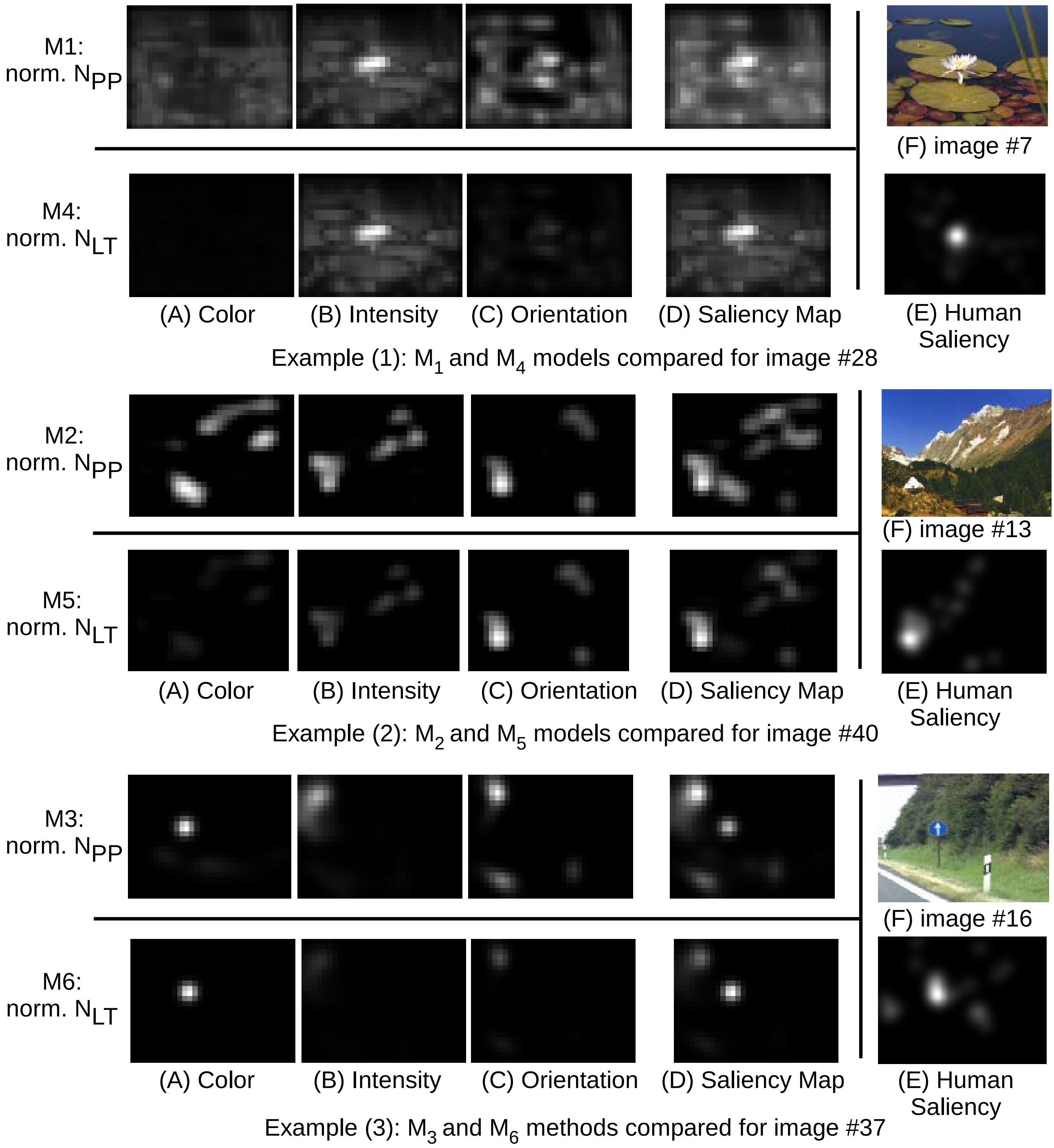
Figure 7.1 illustrates the qualitative evaluation for different images. Three examples are presented. Examples (1), (2) and (3) compare respectively $\mathcal{M}_1$ to $\mathcal{M}_4$, $\mathcal{M}_2$ to $\mathcal{M}_5$ and $\mathcal{M}_3$ to $\mathcal{M}_6$. (A), (B) and (C) illustrate the feature contribution, while (D) shows the computer saliency map. (F) and (E) present the original image and the corresponding human saliency map.

Before comparing human and computer saliency maps, we first observe the experimental data. We can observe that each image contains a dominant cue. Indeed, the dominant cue in example (1) is intensity (contrast of the flower), while it is orientation in (2) (contrast of the white roof) and color in (3) (contrast of the blue traffic sign). As expected, most human fixations (represented in (E)) correlate and are located on the most salient region of the image.

By comparing visually human (E) and computer saliency maps (D), we conclude that all models using the long-term $\mathcal{N}_{LT}$ ($\mathcal{M}_4$,$\mathcal{M}_5$,$\mathcal{M}_6$) are closer to predict the human saliency map, compared to the three models using the peak-to-peak $\mathcal{N}_{PP}$ ($\mathcal{M}_1$,$\mathcal{M}_2$,$\mathcal{M}_3$).

The higher suitability of the long-term $\mathcal{N}_{LT}$ is explained by a more representative cue contribution. Indeed, the long-term $\mathcal{N}_{LT}$ has the advantage of taking into account the relative contribution of the cues. Therefore, the cue contribution of the image are conserved. For example, in (1), where intensity is the dominant cue, color (A) and orientation (C) are strongly attenuated, while intensity (B) is strongly promoted.

Conversely, the peak-to-peak $\mathcal{N}_{PP}$ scales each cue to the same value range, regardless of the effective map amplitude. We can see in (1) the approximative equal cue contribution ((A),(B) and (C)) in the resulting saliency map. Therefore, in the peak-to-peak normalization, all cues contribute in a similar way to the saliency map.



Example (1): M_1 and M_4 models compared for image #28

Example (2): M_2 and M_5 models compared for image #40

Example (3): M_3 and M_6 methods compared for image #37

Figure 7.1: Peak-to-peak $\mathcal{N}_{PP}$ versus long-term $\mathcal{N}_{LT}$ normalizations.

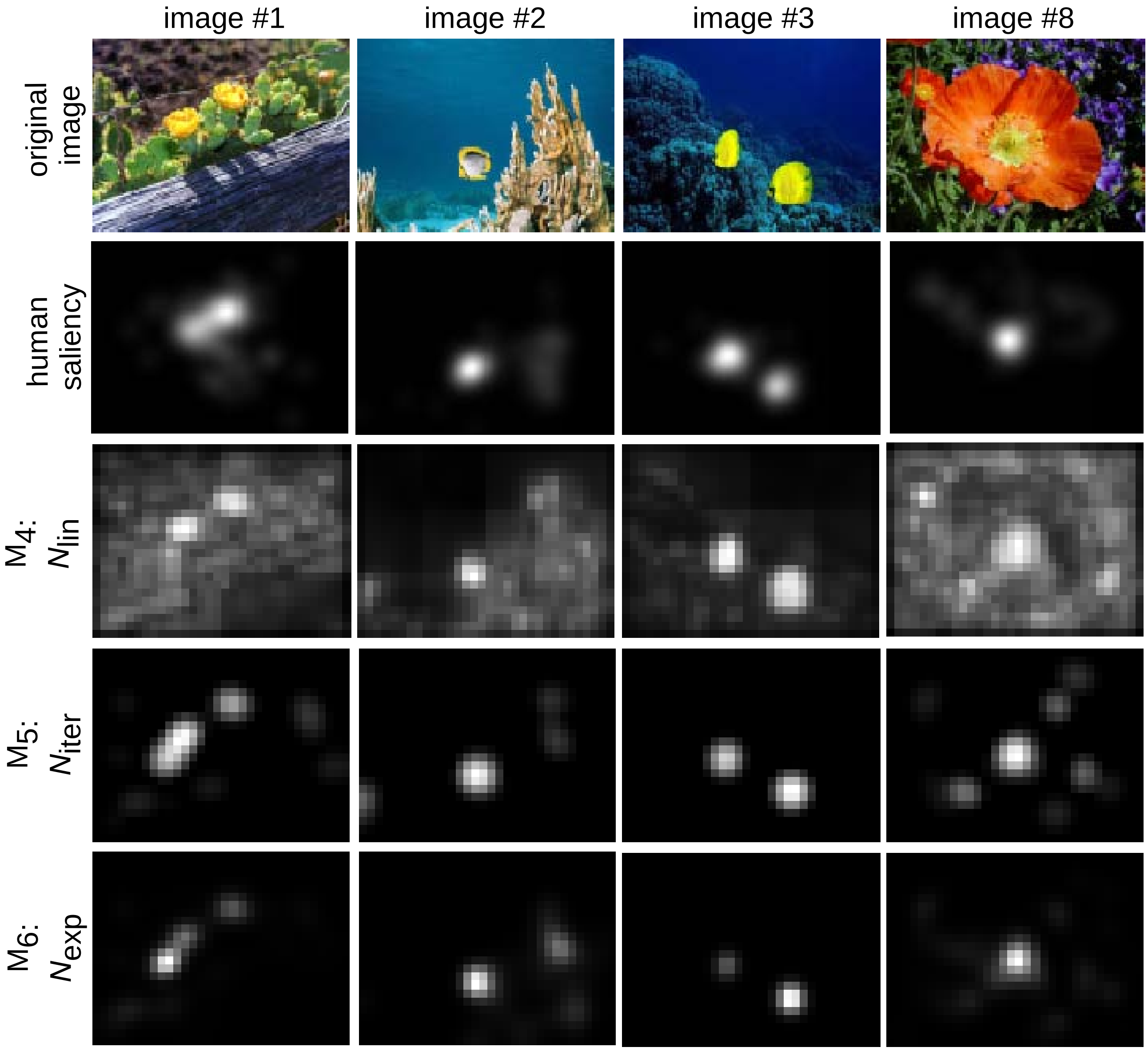


Figure 7.2: Qualitative evaluation of the map transforms $\mathcal{N}_{lin}$, $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$, by comparing the saliency map issued from $\mathcal{M}_4$, $\mathcal{M}_5$ and $\mathcal{M}_6$ with the human saliency map.

The examples presented in this subsection are representative of the higher suitability of the long-term normalization than the peak-to-peak normalization. It will be confirmed by the quantitative evaluation in Section 7.3.

7.2.2 Map transforms: $\mathcal{N}_{lin}$, $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$

This subsection presents the qualitative evaluation of the map transforms $\mathcal{N}_{lin}$, $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$. For each image, the computer saliency maps issued from $\mathcal{M}_4$, $\mathcal{M}_5$ and $\mathcal{M}_6$ are visually compared with the human saliency map. We note that the long-term normalization $\mathcal{N}_{LT}$ is used.

Figure 7.2 illustrates the qualitative evaluation. For each examples, we can see that both models ($\mathcal{M}_5$, $\mathcal{M}_6$) using non-linear map transforms ($\mathcal{N}_{iter}$, $\mathcal{N}_{exp}$), are closer for predicting the human saliency map, compared to the model ($\mathcal{M}_4$) using the linear map transform $\mathcal{N}_{lin}$.

We can see that the non-linear nature of both map transforms ($\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$) tends to promote the higher peak values and demotes the lower ones. Both models therefore tend to suppress the low-level values formed by the background. Conversely, we can see that the linear nature of the map transform $\mathcal{N}_{lin}$ tends to include in the resulting saliency map, irrelevant background noise around the salient regions.

7.3 Quantitative evaluation

In order to measure the similarity between the experimental and computer data, four metrics, issued from the state of the art and described in Section 6.3, are used in the evaluation. We present an overview of the score repartition for the image set. In addition, a statistical test is applied to compare the normalization schemes $\mathcal{N}_{PP}$ and $\mathcal{N}_{LT}$, as well as the map transforms $\mathcal{N}_{lin}$, $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$.

Figure 7.3 presents the score repartition of the six static models ($\mathcal{M}_1$ to $\mathcal{M}_6$) for the fixation-to-chance distance and correlation coefficient, while Figure 7.4 for the Kullback-Leibler divergence and AUC value.

A non-parametric paired t-test is applied to compare both normalization schemes $\mathcal{N}_{PP}$ and $\mathcal{N}_{LT}$ for each map transform. Therefore, $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$ are respectively compared to $\mathcal{M}_4$, $\mathcal{M}_5$ and $\mathcal{M}_6$. Table 7.2 shows the results of the paired t-test. The t-value and respective level of significance p-value are presented. μ_D refers to the average of the difference computed from the image set.

Regarding the comparison of the normalization schemes, the averages μ_D and high p-values illustrate the higher suitability of the long-term $\mathcal{N}_{LT}$. As example, μ_D ranges between 0.36 to 2.01 for the fixation-to-chance distance, and between 0.059 and 0.08 for the coefficient correlation. We note the negative values of μ_D for the Kullback-Leibler divergence, which measures the dissimilarity, while the other metrics measure the similarity.

Regarding the map transform comparison, the statistical analysis presented in Table 7.3 shows higher suitability of the non-linear $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$ compared to the linear $\mathcal{N}_{lin}$ (e.g.

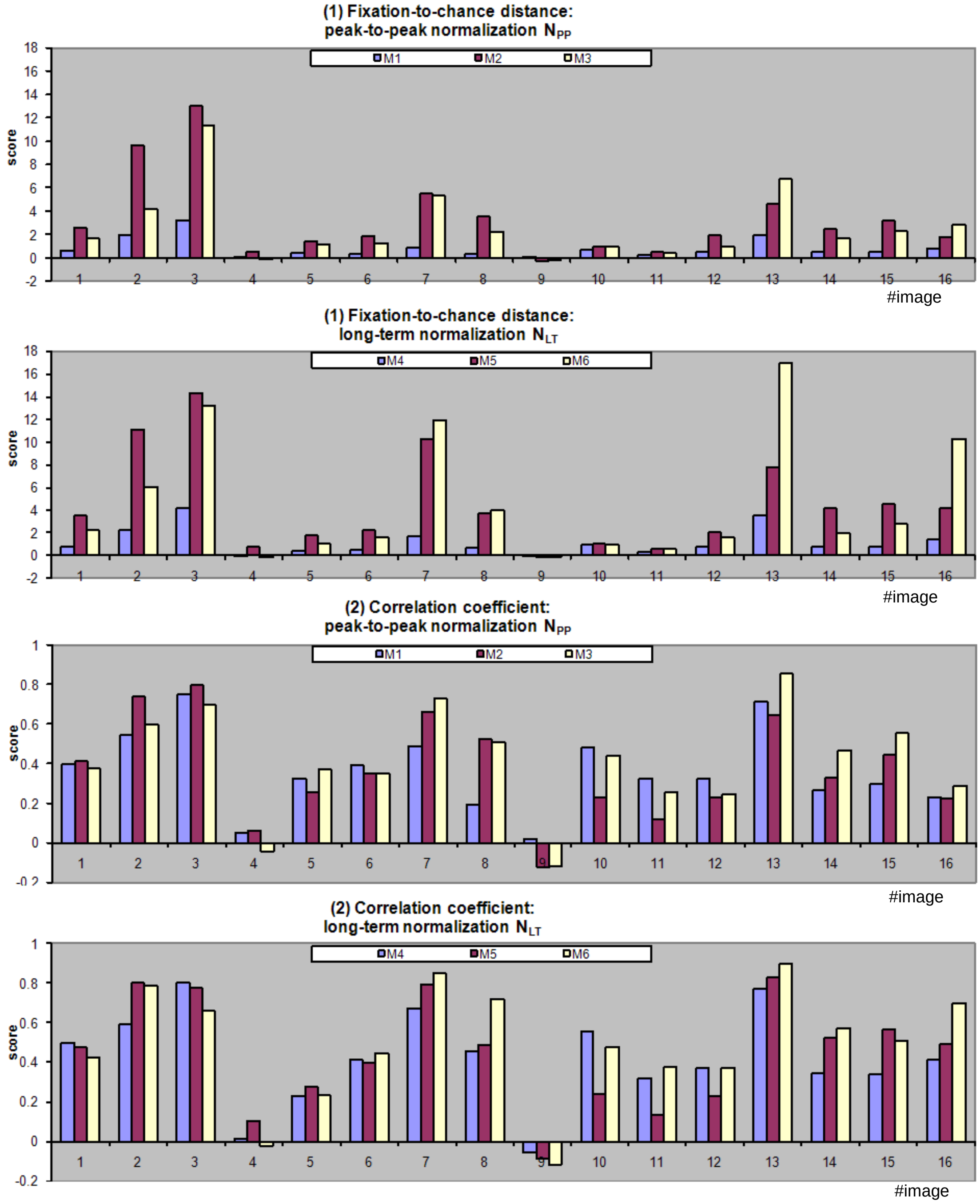
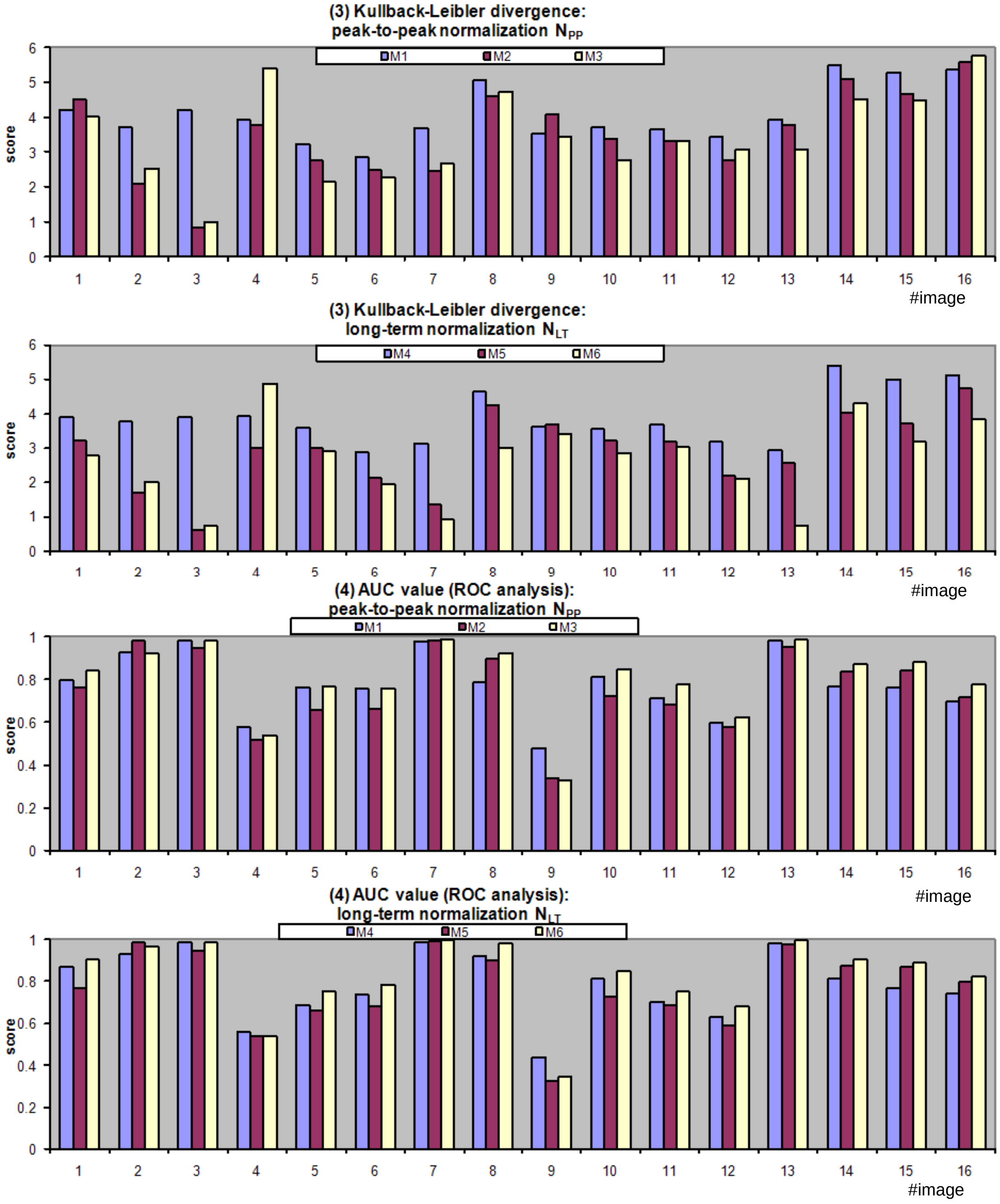


Figure 7.3: Quantitative evaluation: an overview of the score repartition for the fixation-to-chance distance and correlation coefficient.



(3) Kullback-Leibler divergence: peak-to-peak normalization N_{PP}

(3) Kullback-Leibler divergence: long-term normalization N_{LT}

(4) AUC value (ROC analysis): peak-to-peak normalization N_{PP}

(4) AUC value (ROC analysis): long-term normalization N_{LT}

Figure 7.4: Quantitative evaluation: an overview of the score repartition for the Kullback-Leibler divergence and AUC value.

Table 7.2: Quantitative evaluation: a non-parametric paired t-test is used to compare the normalization schemes $\mathcal{N}_{PP}$ and $\mathcal{N}_{LT}$. The statistical test is computed from a set of 16 images (n=16) and parameter μ_0 is set to 0.

		Fixation-to-chance distance	correlation coefficient	Kullback-Leibler divergence	AUC value (ROC analysis)
$\mathcal{M}_1$ vs $\mathcal{M}_4$	t	3.27 (p<0.0025)	2.52 (p<0.025)	2.39 (p<0.025)	1.04 (-)
	μ_D	0.36	0.059	-0.18	0.012
$\mathcal{M}_2$ vs $\mathcal{M}_5$	t	3.49 (p<0.0025)	3.26 (p<0.0025)	5.41 (p<0.0005)	2.94 (p<0.005)
	μ_D	1.17	0.07	-0.6	0.015
$\mathcal{M}_3$ vs $\mathcal{M}_6$	t	2.53 (p<0.025)	2.56 (p<0.01)	3.69 (p<0.001)	2.99 (p<0.005)
	μ_D	2.01	0.08	-0.78	0.021

Table 7.3: Quantitative evaluation: a non-parametric paired t-test is used to compare the three map transforms $\mathcal{N}_{lin}$, $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$. The statistical test is computed from a set of 16 images (n=16) and parameter μ_0 is set to 0.

		Fixation-to-chance distance	correlation coefficient	Kullback-Leibler divergence	AUC value (ROC analysis)
$\mathcal{M}_4$ vs $\mathcal{M}_5$	t	4.09 (p<0.0005)	0.53 (-)	4.7 (p<0.0005)	-1.05 (-)
	μ_D	3.32	0.019	-0.98	-0.016
$\mathcal{M}_4$ vs $\mathcal{M}_6$	t	3.19 (p<0.005)	2.08 (p<0.05)	5.21 (p<0.0005)	2.92 (p<0.005)
	μ_D	3.5	0.07	-1.23	0.036
$\mathcal{M}_5$ vs $\mathcal{M}_6$	t	0.22 (-)	1.63 (-)	1.25 (-)	4.36 (p<0.0005)
	μ_D	0.18	0.051	-0.25	0.052

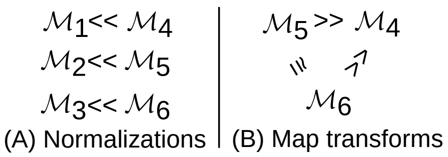

$\mathcal{M}_1$<< $\mathcal{M}_4$
$\mathcal{M}_2$<< $\mathcal{M}_5$
$\mathcal{M}_3$<< $\mathcal{M}_6$
(A) Normalizations
$\mathcal{M}_5$ >> $\mathcal{M}_4$
$\mathcal{M}_6$
(B) Map transforms

Figure 7.5: Performance summary of the six map integration strategies: (A) normalization study and (B) map transform study.

fixation-to-chance distance, $\mathcal{M}_4$ versus $\mathcal{M}_5$: μ_D=3.32 (p<0.0005), $\mathcal{M}_4$ versus $\mathcal{M}_6$: μ_D=3.5 (p<0.005)). Moreover, both non-linear $\mathcal{N}_{iter}$ and $\mathcal{N}_{exp}$ map transforms perform similarly. Indeed, low μ_D and p-values are not significative for the fixation-to-chance distance, correlation coefficient and Kullback-Leibler divergence.

Therefore, the quantitative evaluation confirms the observations of the qualitative one, showing higher performances of the long-term normalization $\mathcal{N}_{LT}$ compared to the peak-to-peak normalization. Regarding the map transforms, both non-linear iterative $\mathcal{N}_{iter}$ and exponential $\mathcal{N}_{exp}$ perform equally well and are more suitable than the linear $\mathcal{N}_{lin}$.

7.4 Chapter summary

First, the study concludes to the higher suitability of the long-term normalization $\mathcal{N}_{LT}$ compared to the peak-to-peak normalization $\mathcal{N}_{PP}$ (Figure 7.5 (A)). Indeed, $\mathcal{N}_{LT}$ has the advantage to take into account the relative contribution of the cues, while the inconvenient $\mathcal{N}_{PP}$ scales each cue to the same value range, regardless of the effective map amplitude.

Second, regarding the map transforms, both non-linear iterative $\mathcal{N}_{iter}$ and exponential $\mathcal{N}_{exp}$ perform equally well and are more suitable than the linear $\mathcal{N}_{lin}$ (Figure 7.5 (B)). While the linear one tends to include in the saliency map irrelevant background noise around salient regions, both non-linear map transforms have the advantage to suppress low-level values formed by the background.

From this study, two optimal configurations showed higher performances compared to the others. The first one combining the map transform $\mathcal{N}_{iter}$ with the normalization scheme $\mathcal{N}_{LT}$ and the second one the map transform $\mathcal{N}_{exp}$ with $\mathcal{N}_{LT}$:

$$Configuration\ \mathcal{M}_5 : \begin{cases} \mathcal{N}_{iter/ID}(c) = & \mathcal{N}_{iter}(c) & (multiscale\ level) \\ \mathcal{N}_{iter/ID}(c) = & \mathcal{N}_{iter}(c) & (feature\ conspicuity\ level) \\ \mathcal{N}_{iter/LT}(c) = & \mathcal{N}_{iter}(\mathcal{N}_{LT}(C)) & (cue\ conspicuity\ level), \end{cases} \tag{7.1}$$

$$Configuration\ \mathcal{M}_6 : \begin{cases} \mathcal{N}_{exp/ID}(c) = & \mathcal{N}_{exp}(c) & (multiscale\ level) \\ \mathcal{N}_{exp/ID}(c) = & \mathcal{N}_{exp}(c) & (feature\ conspicuity\ level) \\ \mathcal{N}_{exp/LT}(c) = & \mathcal{N}_{exp}(\mathcal{N}_{LT}(c)) & (cue\ conspicuity\ level). \end{cases} \tag{7.2}$$

We use the second configuration to design the static model for its lower computation costs.

Chapter 8

Motion model evaluation

8.1 Chapter introduction

This section presents the motion model evaluation. As mentioned previously, both static and motion features are required in the design of a dynamic model. Indeed, it does not make real sense to design an attentional system purely driven by the motion channel. Besides, several studies [30, 31, 6] show that a dynamic model including all features is more suitable to predict human visual attention than a model comprising a single motion feature.

Therefore, several dynamic models are defined to assess the motion model suitability. The models include a static contribution, defined by the static model (Chapter 3), and a motion contribution defined by one of the four motion models (Chapter 4). Both contributions refer to the static cue C_{static} and the motion cue C_{motion}. The resulting dynamic saliency map is obtained by fusion of both maps using the competitive map integration scheme (Section 5.2).

Therefore, the four dynamic models are defined as follows:

$$\begin{array}{llcccc}
(1)\ \mathcal{M}_{magn}\ Magnitude\ dynamic\ model: & S_{magn} & = & \mathcal{N}(C_{magn}) & + & \mathcal{N}(C_{static}), \\
(2)\ \mathcal{M}_{vector}\ Vector\ dynamic\ model: & S_{vector} & = & \mathcal{N}(C_{vector}) & + & \mathcal{N}(C_{static}), \\
(3)\ \mathcal{M}_{p\&m}\ Phase\&magnitude\ dynamic\ model: & S_{p\&m} & = & \mathcal{N}(C_{p\&m}) & + & \mathcal{N}(C_{static}), \\
(4)\ \mathcal{M}_{dir}\ Direction\ dynamic\ model: & S_{dir} & = & \mathcal{N}(C_{dir}) & + & \mathcal{N}(C_{static}).
\end{array} \tag{8.1}$$

The model evaluation is performed by using psycho-physical experiments (Chapter 6). These experiments include video sequences of different nature (synthetic and natural real scenes, acquired with fixed and moving background), showing advantages and inconveniences of the different models.

We have mentioned previously that the considered motion models highlight specific motion contrasts. Before entering into the main scope of this chapter (model evaluation), we will first analyze the experimental data. An analysis of the human saliency map will bring to light the motion contrasts, to which the human vision system is sensitive. Such an analysis is an experimental illustration of the requirements of a suitable computer motion model. Then, we will present the model evaluation. First, a qualitative evaluation is provided. The computer saliency maps of the four models are compared visually to the human saliency map. Through typical examples, we will therefore illustrate the differences between the models and show how the models performs in two distinctive video categories (fixed and moving background). Finally, a quantitative evaluation is performed.

The chapter is structured as follow. Section 8.2 presents the experiments, including a description of the video sequence set and the design of the experiment. Then, Section 8.3 presents the human saliency analysis. Section 8.4 and 8.5 describe the qualitative and quantitative evaluation. Finally, Section 8.6 provides additional comments, including a discussion on the influence of top-down attention in the results.

8.2 Experiments

The psycho-physical experiments were conducted with 20 human subjects. All of them have normal or corrected-to-normal visual acuity as well as normal color vision. Each subject was viewing a set of video sequences, while the eye tracker is recording their eye movement patterns for a duration of 15 minutes. A preliminary calibration was performed for each subject just before the experiment.

Synthetic and real video sequences were displayed alternatively and randomly in order to keep a close attention of the subject throughout the viewing session. Each video sequence was preceded by a central fixation cross for 2 seconds. The subjects were instructed to freely look at the screen with no specific task.

In the design of the experiment, we include synthetic and natural real video sequences for three reasons. First, using synthetic video provides the advantage to create simple situations with specific motion contrasts. Second, synthetic scenes reduce top-down influence that may occur for the case of natural real scenes. Third, for evident reasons, natural real scenes are used in the evaluation for prospective computer vision applications.

8.2.1 Set of video sequences

The set of video clips consists of 84 short sequences (10 sec. duration), representing four categories. The first and second categories include synthetic scenes, while the third and fourth natural real scenes. Some examples are illustrated in Figure 8.1. The first category contains 15 videos with fixed background, combining static, moving, high color-contrasted and low-color-contrasted targets on a uniform background. The second category contains 19 videos with moving background. The considered motion pattern is pure-translation in the camera plane.

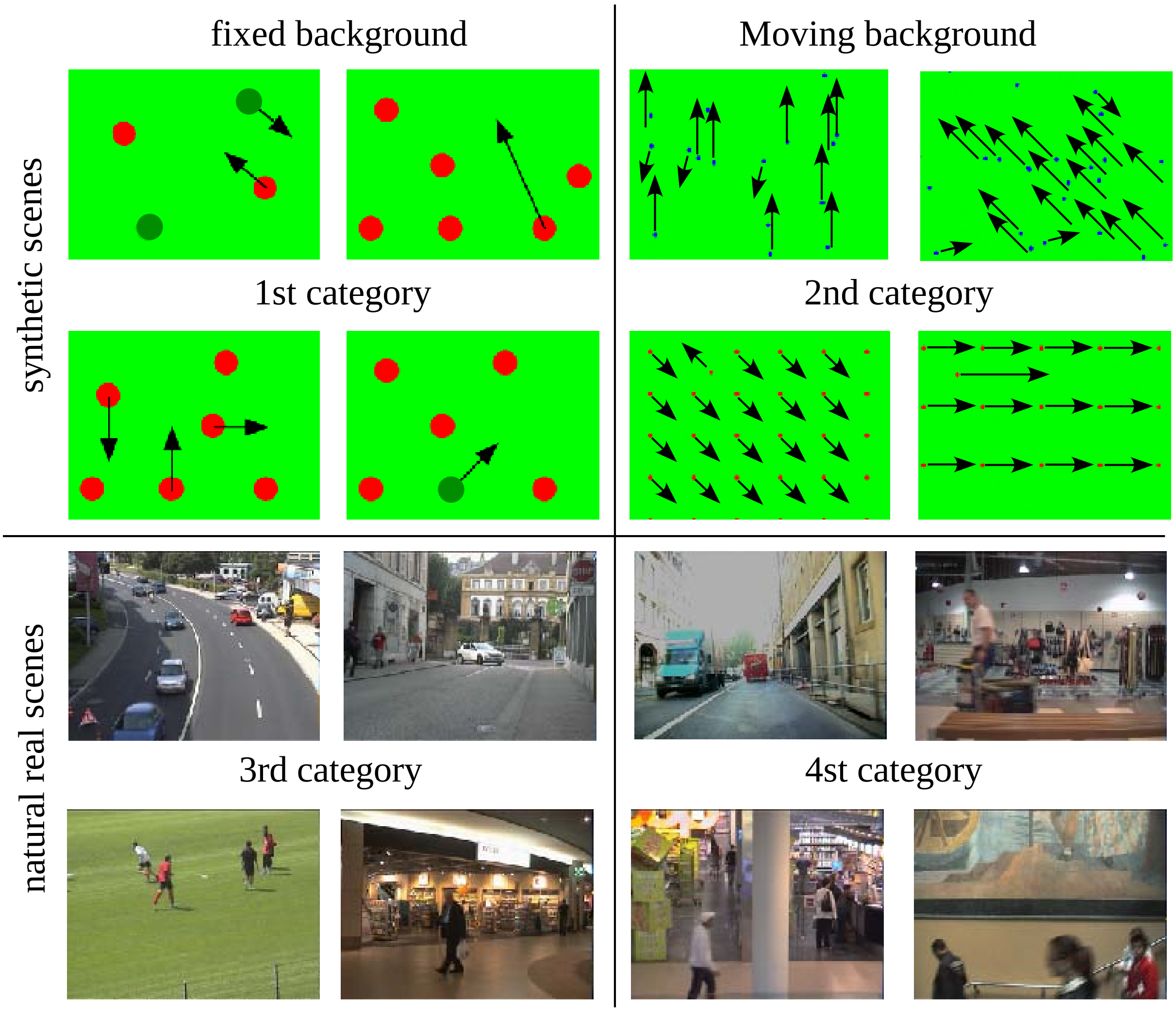


Figure 8.1: Some examples of the video clips used in the experiments. 84 sequences of synthetic and natural real scenes are considered, classified in four categories: synthetic scenes with fixed background, synthetic scenes with moving background, real scenes with fixed background and real scenes with moving background. The arrows indicate the direction of motion for the synthetic scenes. The first category contains sequences combining static, moving, high-color-contrasted and low-color-contrasted targets on a uniform background. The second category contains videos composed of one or several moving dots (targets supposed to be salient) among a background of moving dots (distractors). Various configurations of motion contrast are considered: target dot(s) moving either faster or slower and in the same direction relatively to the background; target dot(s) moving slower, at the same speed, or faster and in another direction relatively to the background. The third and fourth categories include videos of natural real scenes in outdoor and indoor environment (city, traffic road, street, football field, train station, stores).

The video content consists of one or several moving dots (targets supposed to be salient) among a background of moving dots (distractors). The background is either composed of random dots or either of a grid of dots. Various configurations of motion contrast are considered: target dot(s) moving either faster or slower in the same direction relatively to the background; target dot(s) moving slower, at the same speed, or faster in another direction than the background. The third (30 videos with fixed background) and fourth categories (20 videos with moving background) include videos of natural real scenes in outdoor and indoor environment (city, traffic road, street, football field, train station, stores).

8.3 Human saliency analysis

In order to illustrate the sensitivity of human attention to specific motion contrasts, namely contrasts in phase and magnitude, a visual analysis of the experimental measurements is performed. The idea is to analyze visually wether the most fixated regions (represented by the human saliency map) are located on motion contrasts.

As reminder, two types of motion contrasts are salient in the motion representation, the former in magnitude, the latter in phase (Figure 4.1). The former is discriminant in terms of magnitude. Such a contrast is defined as a difference of speed magnitude between the center and surrounding motion. The latter is discriminant in terms of phase, in other word, a difference of speed direction. Video sequences containing both types of contrast have been used in the experiments to investigate wether such contrasts are visually salient.

The results are presented in Figure 8.2 for synthetic scenes and in Figure 8.3 for natural real scenes. A description of motion contrasts of different nature (in magnitude, in phase, both in phase and magnitude) is depicted in column (1). Each motion contrast is defined by $\mathbf{v}_c$ and $\mathbf{v}_s$, which refers respectively to the motion vector of the center and surround region. We mention that the video sequences include most configuration of motion contrasts, but not an exhaustive list of all possible cases. The original frames, with arrows indicating motion, are presented in column (2). Arrows in red indicate motion of the center region expected to be salient, while arrows in blue motion of the surrounding region. Column (3) shows the eye movement patterns, and finally, column (4) the corresponding human saliency maps. Globally, over all video sequences, a majority of eye movement patterns concentrate on specific regions, containing one of the mentioned motion contrasts.

This analysis illustrates experimentally that human attention is sensitive to motion contrasts in phase and magnitude. It is an interesting observation, which confirms the importance of motion contrasts in the VA modeling. Therefore, a suitable motion computer model requires to highlight motion contrasts both in phase and magnitude.

This section illustrated the motion contrasts to which the human vision is system is sensitive. The next section will present the qualitative evaluation of the computer models.

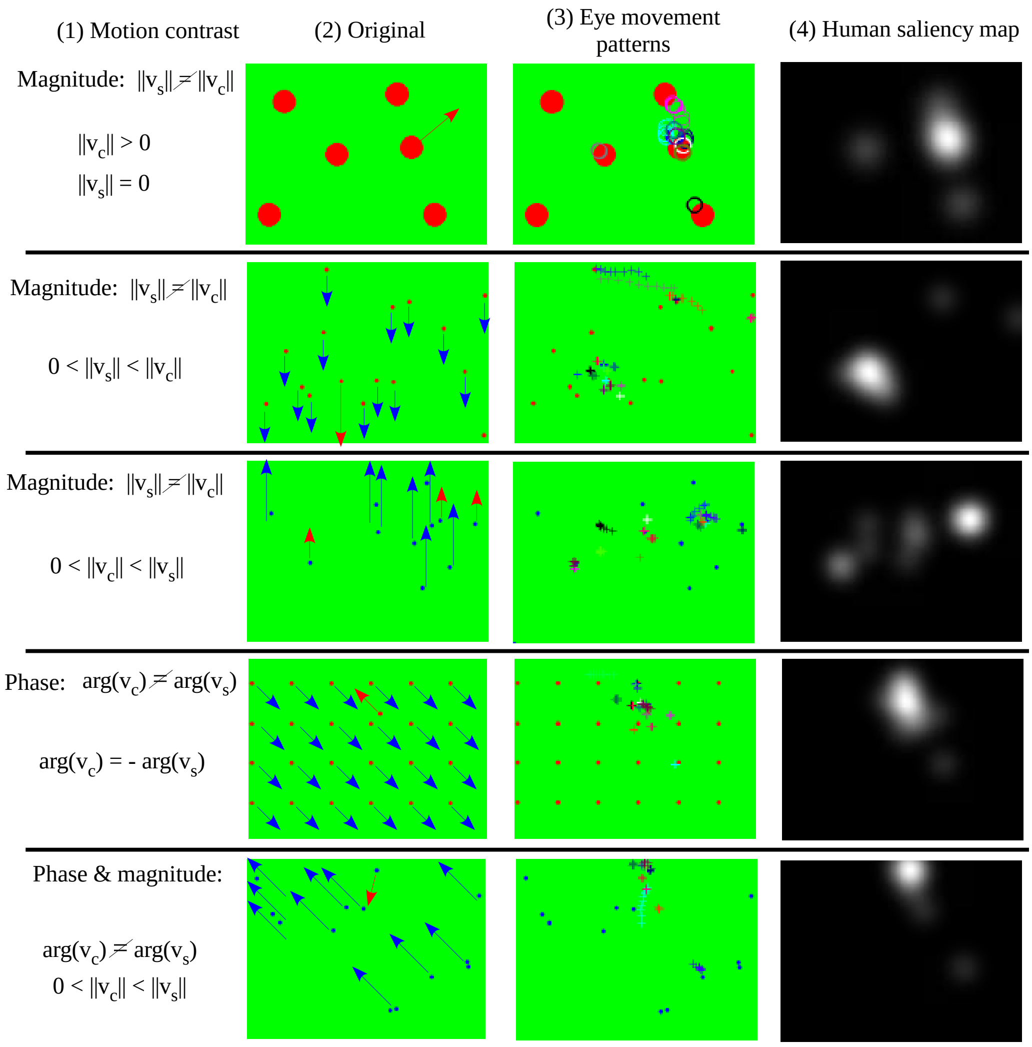


Figure 8.2: Psycho-physical experiments using synthetic scenes: different motion contrasts are investigated, using video sequences composed of one or several moving dots (targets supposed to be salient) among a background of moving dots (distractors).

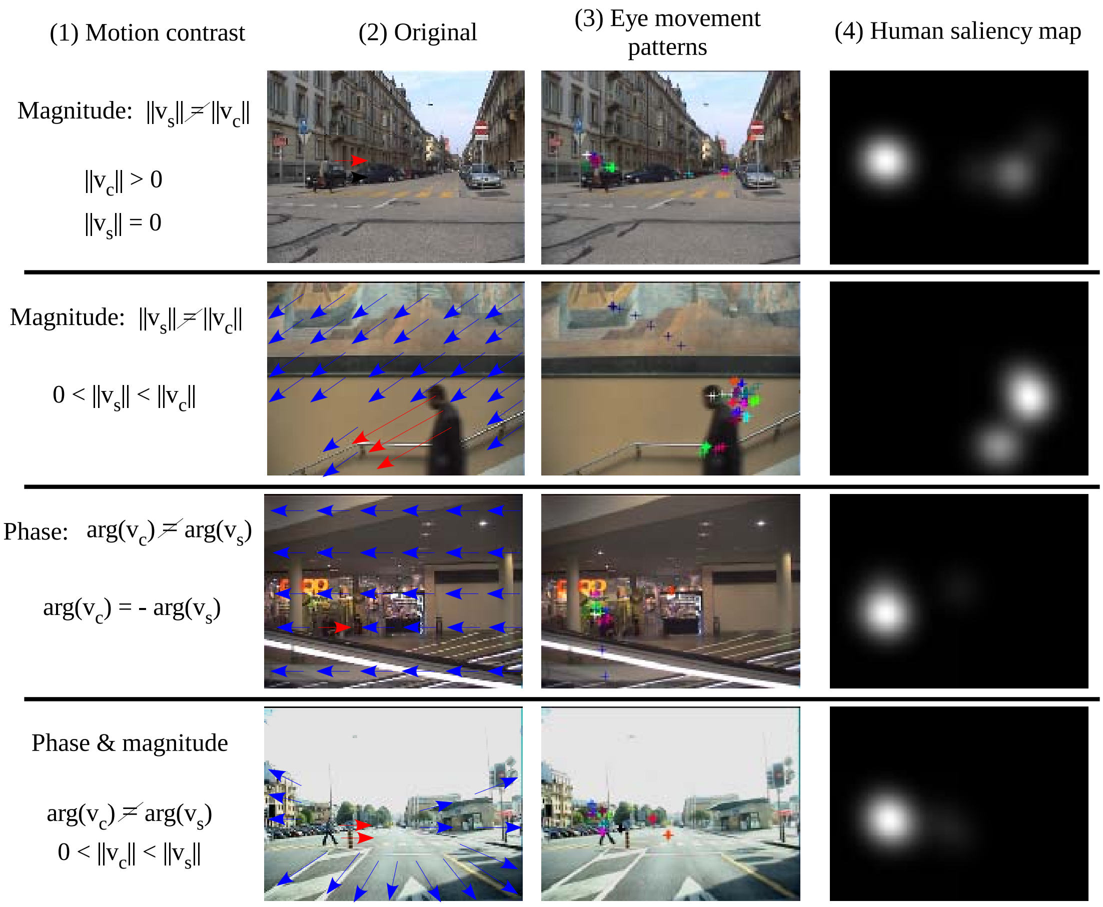

(1) Motion contrast
(2) Original
(3) Eye movement patterns
(4) Human saliency map
Magnitude: $\|v_s\| \neq \|v_c\|$
$\|v_c\| > 0$
$\|v_s\| = 0$
Magnitude: $\|v_s\| \neq \|v_c\|$
$0 < \|v_s\| < \|v_c\|$
Phase: $\arg(v_c) \neq \arg(v_s)$
$\arg(v_c) = - \arg(v_s)$
Phase & magnitude
$\arg(v_c) \neq \arg(v_s)$
$0 < \|v_c\| < \|v_s\|$

Figure 8.3: Psycho-physical experiments using natural real scenes: eye movement patterns are located on motion contrasts.

8.4 Qualitative model evaluation

The qualitative evaluation is divided in two parts: fixed background (Subsection 8.4.1) and moving background (Subsection 8.4.2). Such a division is motivated by two reasons: first, motion contrast sensitivity is different according to the models and performances are expected to be dependant on the nature of the video sequence. Indeed, for sequences with fixed background, the magnitude model $\mathcal{M}_{magn}$ is expected to perform accurately, and similarly to the three remaining models. Conversely, for sequences with moving background, the magnitude model is expected to provide lower performances, while the three models, namely the vector $\mathcal{M}_{vector}$, phase & magnitude $\mathcal{M}_{p\&m}$ and direction $\mathcal{M}_{dir}$ models are expected to provide higher performances.

Second, dividing the evaluation in this way provides insight regarding the choice of one model for prospective computer vision applications. Indeed, one model may be preferred according to the specificity of the application.

8.4.1 Fixed background

In this subsection, we visually compare the human saliency map to the computer saliency maps issued from the four models (Eq. 8.1). We expose several representative examples of the model evaluation for sequences with fixed background.

Figure 8.4 presents the evaluation of the first category, synthetic scenes with fixed background. The examples have been chosen in order to provide a representative overview of the results. (A) shows the original frame. Each one is annotated by arrows indicating motion. The experimental data correspond to the human observations (B) and the human saliency map (C). The computer saliency maps (1) to (4) represent the four computer models.

We point out three observations. First, we can see the high correspondences between the human and computer saliency maps, illustrating the high accuracy of the models. Each subject focusses on salient moving target(s) for a long period, then alternatively briefly looks elsewhere for a short period, and focusses again on the moving target. Globally, the distribution of human observations strongly focusses on motion contrast. This illustrates that motion is one of the most important visual feature for predicting human attentional behavior. Therefore, the computer models correlates to the experimental data.

Second, by comparing the four computer models, we can see no significant differences between the magnitude model and the three other models. As illustrated in example 1 and 4, the four saliency maps highlight the salient moving target(s) similarly.

Third, we can see the presence of artifacts in the saliency map of the direction model. As illustrated in example 2 and 3, artifacts are located in the neighborhood of the salient moving target(s). This defects arise from the computation of the motion field. As reminder, the direction model operates on a unique motion field obtained by recombination of several intermediate motion fields. The coarse-to-fine motion architecture and the absence of texture (uniform background) induce motion artifacts. We note that those defects are more due to the motion field computation rather than the model itself.

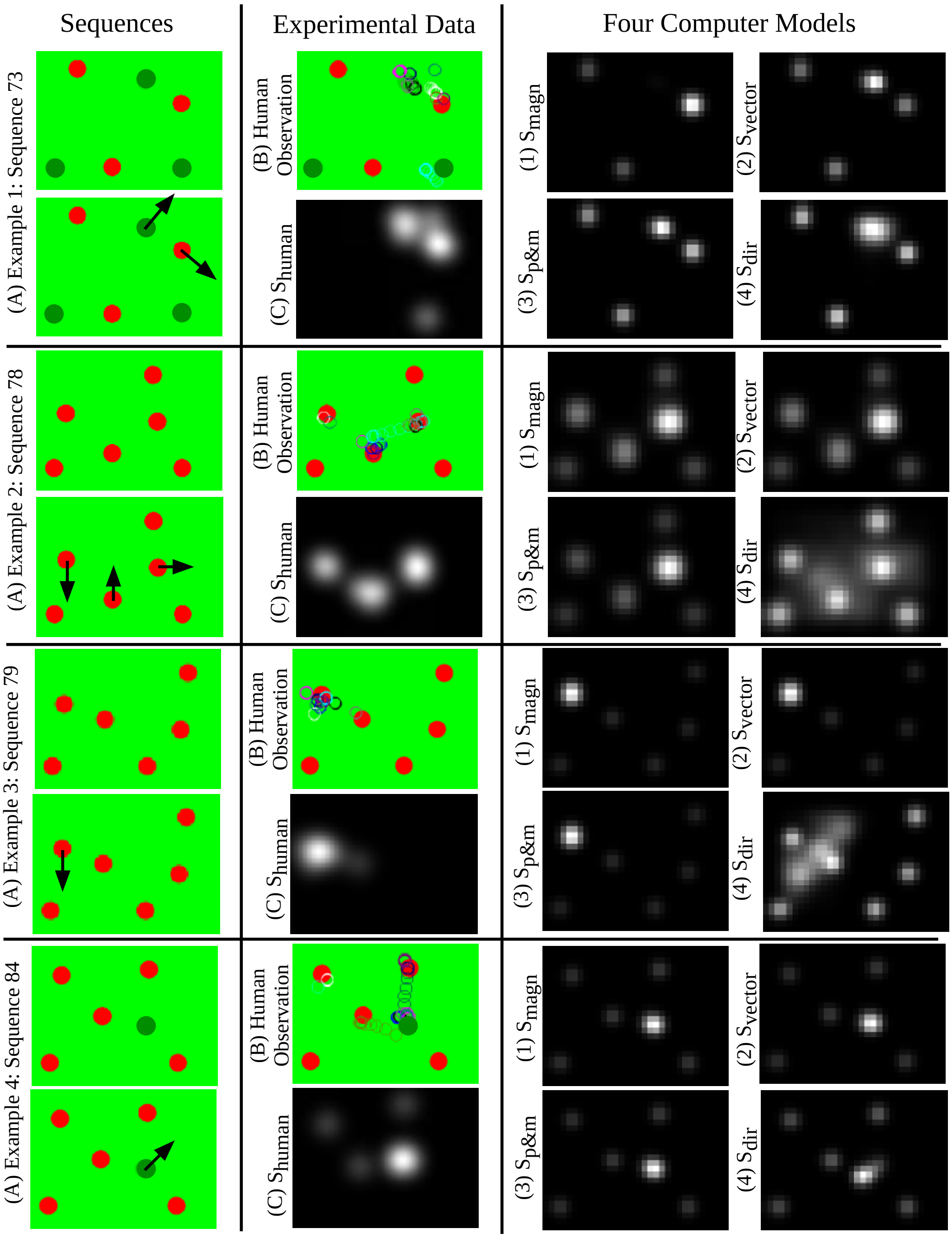


Figure 8.4: Qualitative evaluation of the first category: synthetic scenes with fixed background. The human saliency map issued from the eye movement recordings is compared to the computer saliency maps issued from the four computer models. (A) represents the original frame; (B) the human observations; (C) the human saliency map issued from the fixation and smooth pursuit periods; (1) to (4) the saliency maps issued from the four models.

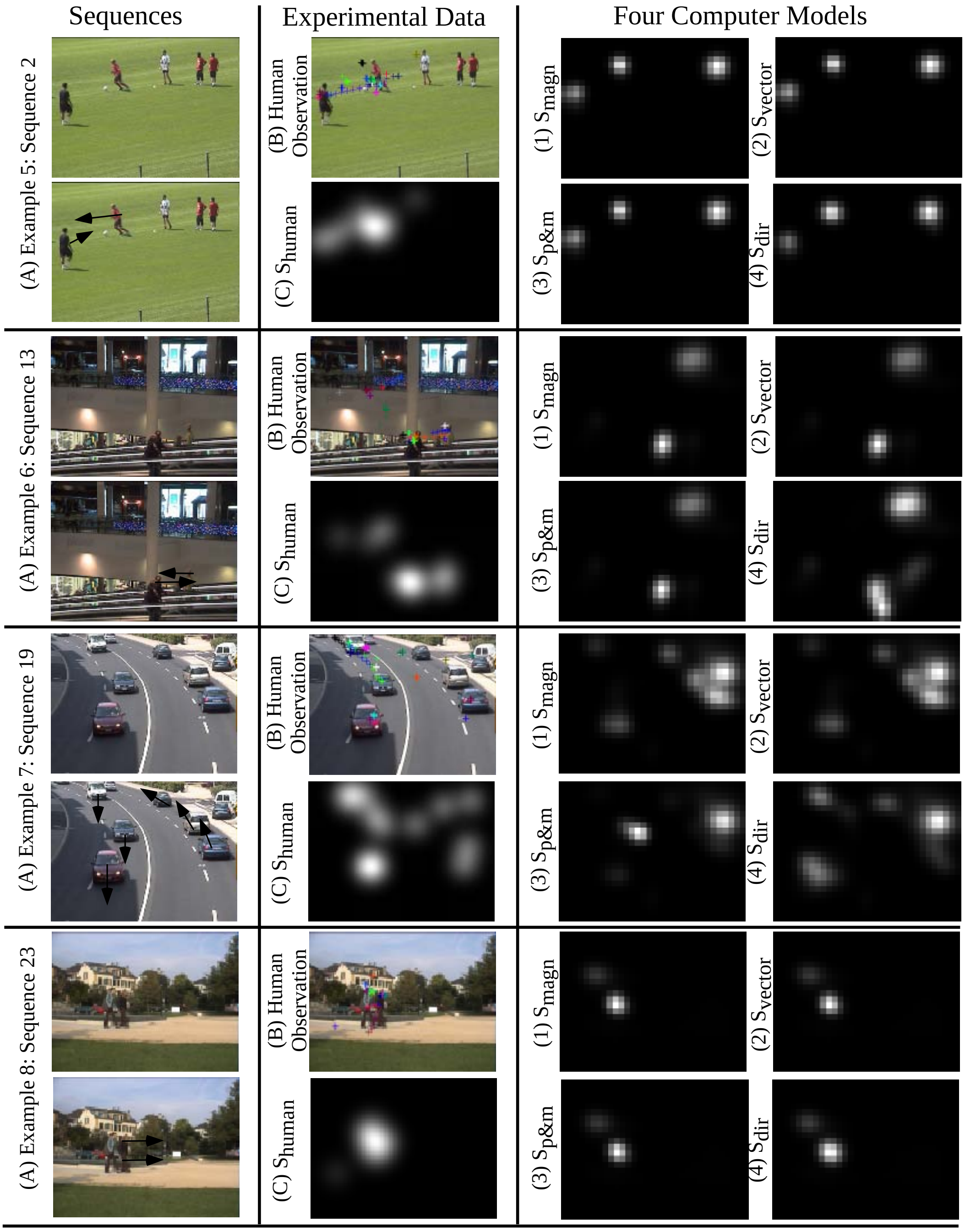


Figure 8.5: Qualitative evaluation of the third category: natural real scenes with fixed background.

Figure 8.5 presents the evaluation for natural real scenes. These examples illustrate the same observations mentioned above, except the third one. Indeed, no artifact are visible in the saliency maps of the direction model. This is explained by the high-contrasted texture of the background, which provides a suitable motion field.

Therefore, regarding the evaluation in fixed background, the four computer models show high correspondences with the experimental data. In addition, for this type of sequences, the magnitude model show similar performances compared to the remaining models. This result is expected, since video sequences with fixed background are restricted to motion contrast in magnitude.

The next subsection presents the qualitative evaluation for video sequences with moving background.

8.4.2 Moving background

In this subsection, we compare visually the human saliency maps to the four computer saliency maps for video sequences with moving background. Figure 8.6 and 8.7 present the evaluation of the second category (synthetic scenes) and fourth category (natural real scenes). Several representative examples are discussed. Some of them show situations in which the four models perform similarly. Alternatively, we will see specific situations in which the magnitude model fails, while the other models are suitable.

Examples 9 and 13 (Figure 8.6 and 8.7) show two typical situations in which the four models perform similarly. The four models highlight the salient motion contrast. In example 13, motion contrast is in magnitude, while it is both in phase and magnitude in example 9. As expected, both situations illustrate the capability of the magnitude model to highlight salient moving stimuli when the speed magnitude of the later is either slower (example 9) or faster (example 13) than the moving background.

We will now see examples in which the magnitude model is not suitable, while the three other models perform accurately. Examples 10, 11, 12 in Figure 8.6 (Synthetic scenes) and 14, 15, 16 in Figure 8.7 (natural real scenes) show several situations in presence of pure motion contrast in phase. Most human observations are located on the phase contrast. We can see that the saliency map of the magnitude model does not correspond to the human saliency map. The magnitude model does not highlight motion contrast in phase and the salient moving regions are therefore not detected. While the magnitude model is not suitable in presence of pure phase contrast, the three models $\mathcal{M}_{vector}$, $\mathcal{M}_{p\&m}$ and $\mathcal{M}_{dir}$ are conversely more suitable to predict the human saliency. It is particularly striking in examples 15 and 16. Indeed, the saliency map of (1) $\mathcal{M}_{magn}$ is dominated by static contrasts, while the saliency maps of (2) $\mathcal{M}_{vector}$, (3) $\mathcal{M}_{p\&m}$ and (4)$\mathcal{M}_{dir}$ are dominated by the phase contrasts. Therefore, in presence of pure phase contrast, the three models predict more accurately the average human visual behavior than the magnitude model.

We note the presence of artifacts in the saliency maps of the direction model, which reduces the model performance (Example 11 and 12). As mentioned previously, these defects are due to the coarse-to-fine architecture and the absence of texture (uniform background). Besides, the

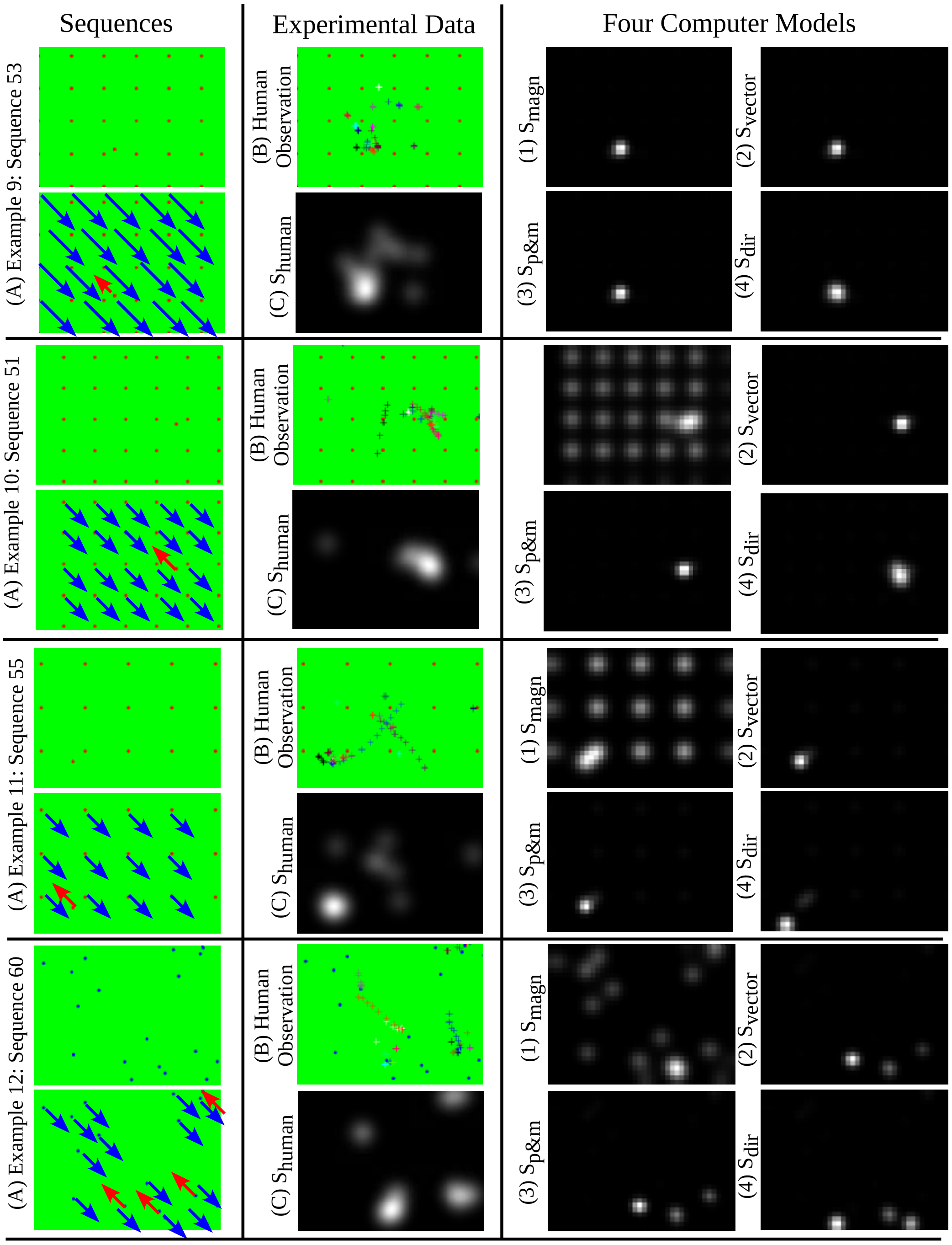


Figure 8.6: Qualitative evaluation of the second category: synthetic scenes with moving background.

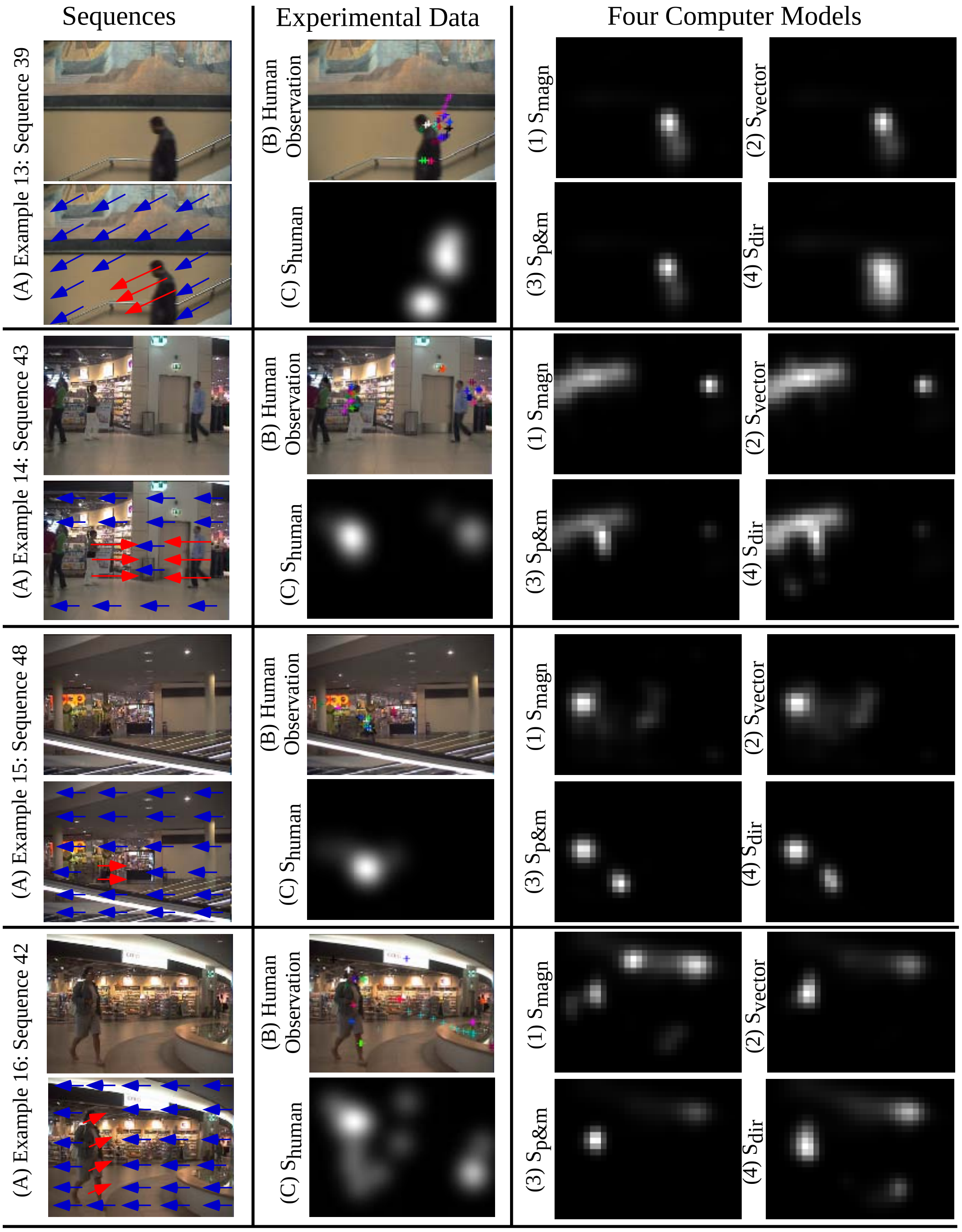
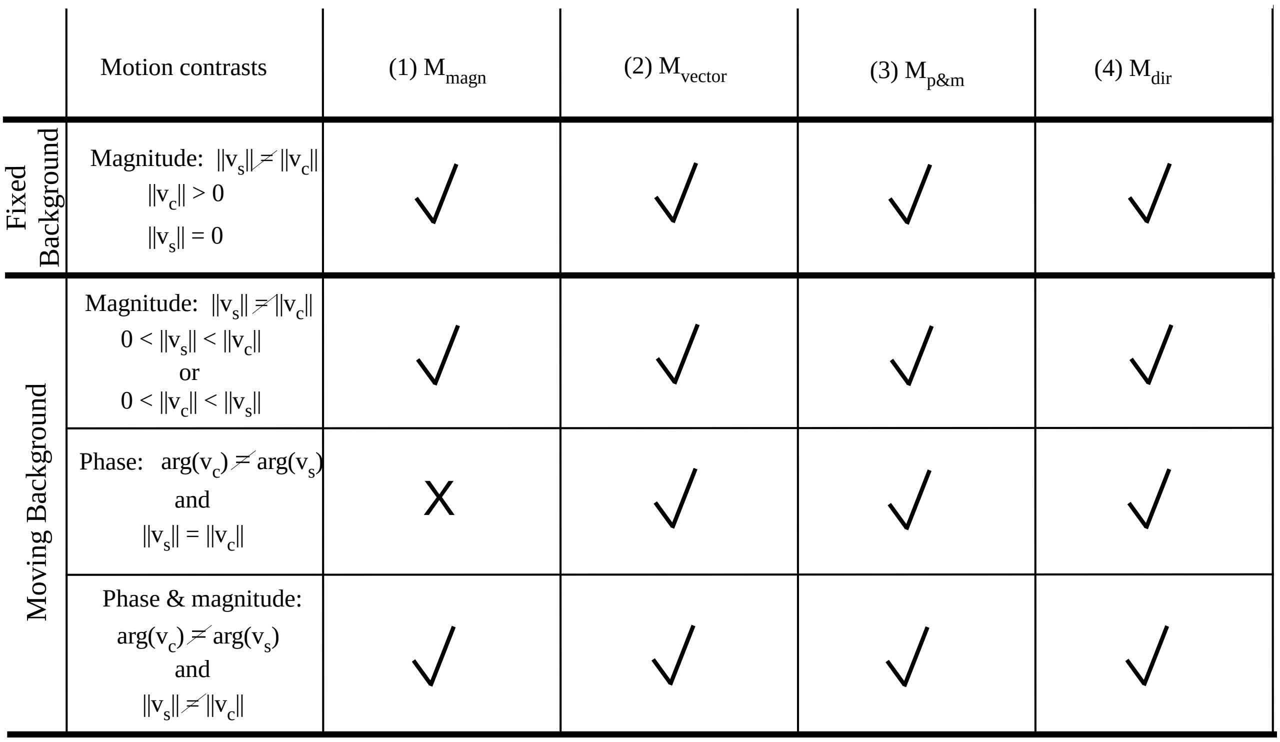


Figure 8.7: Qualitative evaluation of the fourth category: natural real scenes with moving background.

Table 8.1: An overview of the model performances according to the different types of motion contrasts.



	Motion contrasts	(1) M_{magn}	(2) M_{vector}	(3) $M_{p\&m}$	(4) M_{dir}
Fixed Background	Magnitude: $\|v_s\| \neq \|v_c\|$ $\|v_c\| > 0$ $\|v_s\| = 0$	✓	✓	✓	✓
Moving Background	Magnitude: $\|v_s\| \neq \|v_c\|$ $0 < \|v_s\| < \|v_c\|$ or $0 < \|v_c\| < \|v_s\|$	✓	✓	✓	✓
	Phase: $\arg(v_c) \neq \arg(v_s)$ and $\|v_s\| = \|v_c\|$	X	✓	✓	✓
	Phase & magnitude: $\arg(v_c) \neq \arg(v_s)$ and $\|v_s\| \neq \|v_c\|$	✓	✓	✓	✓

artifacts are only visible in synthetic scenes and not in natural real scenes, the latter having high-contrasted background texture.

To conclude the qualitative evaluation, we present in Table 8.1 an overview of the model performances according to the different types of motion contrasts. In fixed background, motion contrast is in magnitude. The four models are suitable and performances are similar.

In moving background, the suitability of the magnitude model depends on the nature of motion contrasts. In presence of pure phase contrast, $\mathcal{M}_{magn}$ is not suitable, while the three other models performs accurately. In case of magnitude contrast, or both phase and magnitude contrasts, the four models performs similarly.

8.5 Quantitative model evaluation

In this section, we present the quantitative model evaluation. In order to measure the similarity between the experimental and computer data, four metrics, issued from the state of the art and described in Section 6.3, are used in the evaluation. We will present an overview of the average scores for each video sequence. The examples shown in the qualitative evaluation, as well as the whole set of video sequences will be analyzed in order to confirm wether both evaluations conduct to similar conclusions.

This section is divided in two parts. Subsection 8.5.1 and 8.5.2 present respectively the model evaluation for sequences with fixed and moving background.

8.5.1 Fixed background

The quantitative evaluation consists in measuring the correspondences between the experimental data, represented by the human saliency map, and the computer data, represented by the computer saliency map. In the evaluation, we use four metrics. Three of them operate on the human and the computer saliency maps (correlation coefficient, Kullback Leibler divergence and AUC value). The remaining metric is the fixation-to-chance distance. It operates differently by comparing two average saliency values, the former computed by sampling the computer saliency map at the fixation and pursuit points and the latter by random sampling. We note that the Kullback-Leibler divergence is a measure of dissimilarity, while the others correspond to a measure of similarity.

Figure 8.8 presents the quantitative evaluation of the first category (synthetic scenes). For each sequence, an average score is computed according to the metric. Graph (1) to (4) show respectively the distribution of the fixation-to-chance distance, the correlation coefficient, the Kullback Leibler divergence and the AUC value.

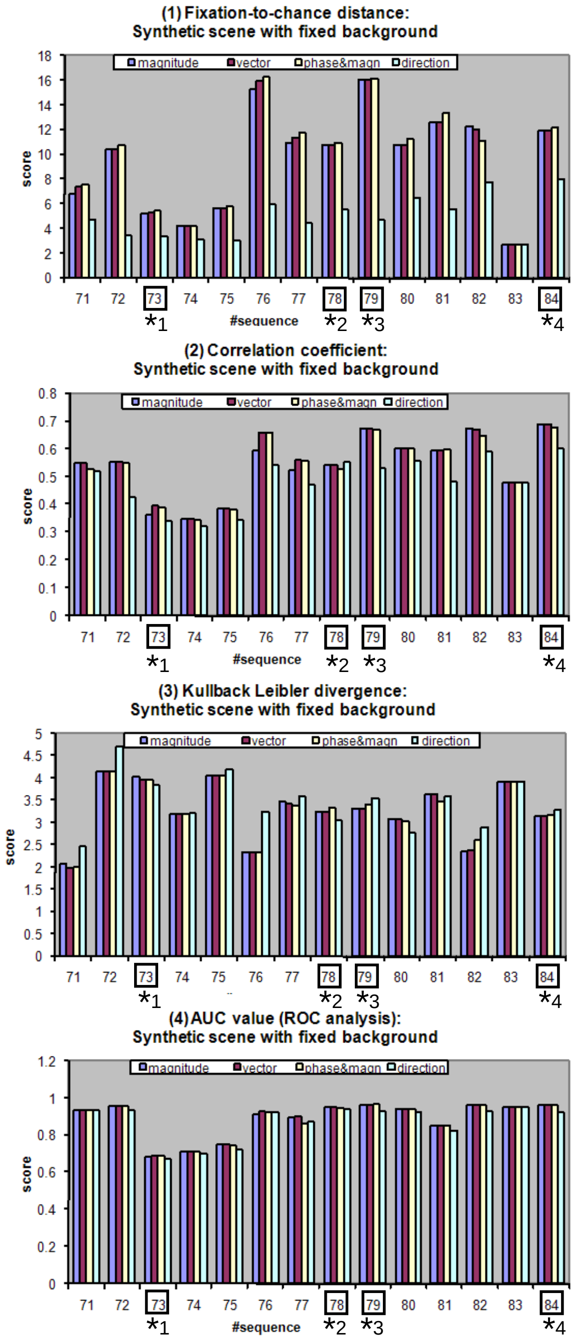
The examples presented in the qualitative evaluation are indicated by an asterisk. By analyzing the four examples, we can see that the score values reflect the observations mentioned in the qualitative evaluation.

Over the whole set of video sequences, we can first notice high correspondences between the experimental and computer data. As illustration, the fixation-to-chance distance indicates that the saliency value at the fixation points is in average ten time higher than the saliency value sampled randomly. In addition, the correlation coefficient ranges from 0.35 to 0.65 and have an average value superior to 0.5.

Second, the magnitude model $\mathcal{M}_{magn}$ shows approximatively identical performances compared to the vector model $\mathcal{M}_{vector}$ and the phase & magnitude $\mathcal{M}_{p\&m}$, while the score values of the direction model $\mathcal{M}_{dir}$ indicate lower performances, due to presence of motion artifacts in the saliency map.

Figure 8.9 presents the evaluation of the third category (natural real scenes). Globally, the four models perform similarly. Regarding the direction model, we note that the lower performances observed in the synthetic scene evaluation are not visible for the natural real scenes.

In order to compare the model performances, a statistical analysis is performed. It consists in applying a non-parametric paired t-test. Figure 8.10 presents an overview of the results for



(1) Fixation-to-chance distance: Synthetic scene with fixed background

(2) Correlation coefficient: Synthetic scene with fixed background

(3) Kullback Leibler divergence: Synthetic scene with fixed background

(4) AUC value (ROC analysis): Synthetic scene with fixed background

Figure 8.8: Quantitative evaluation of the first category (synthetic scenes with fixed background): Graph (1) to (4) show respectively the repartition of the fixation-to-chance distance, the correlation coefficient, the Kullback Leibler divergence and the AUC value. The examples presented in the qualitative evaluation are indicated by an asterisk.

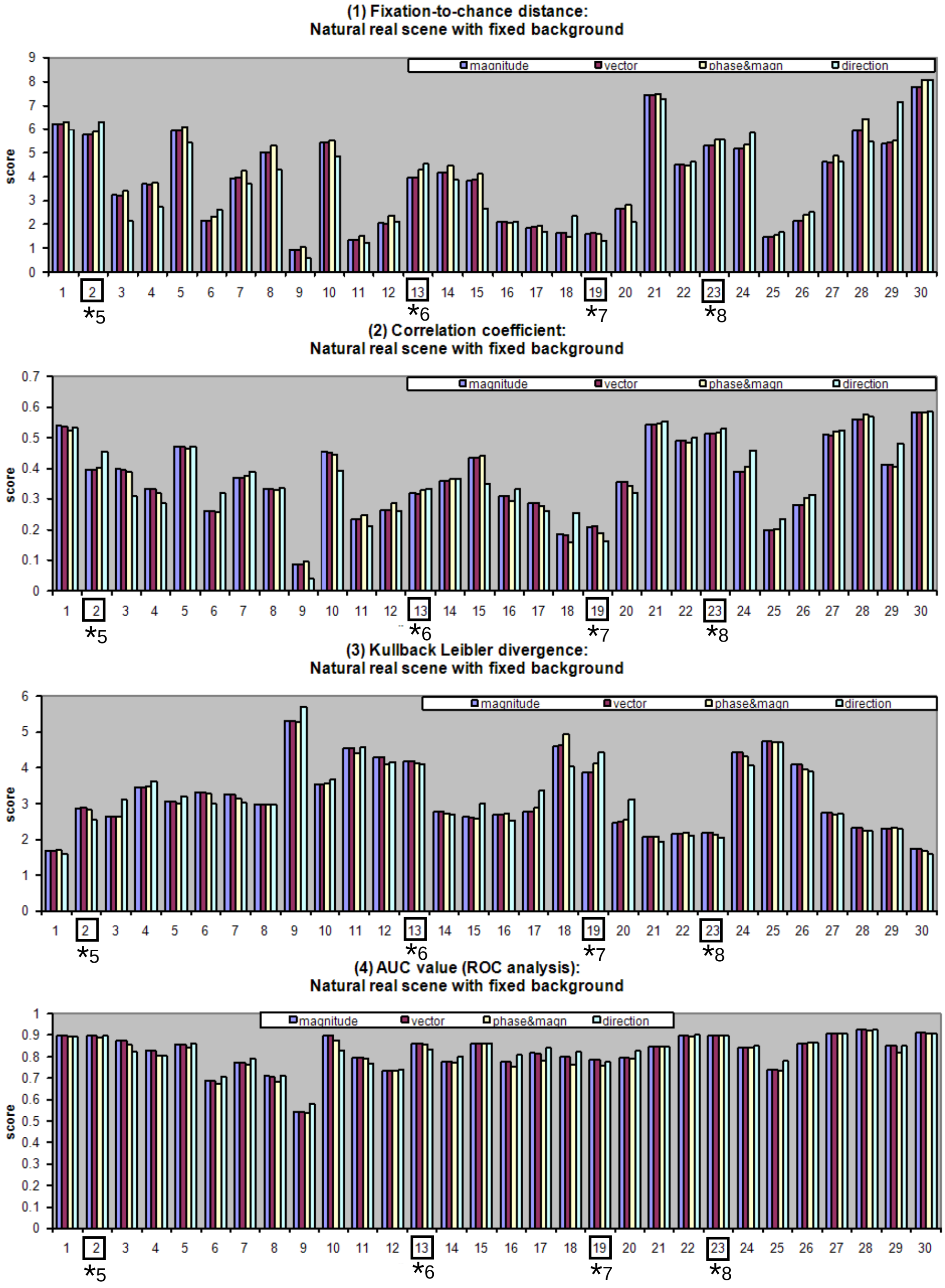


Figure 8.9: Quantitative evaluation of the third category (natural real scenes with fixed background): the examples presented in the qualitative evaluation are indicated by an asterisk.

(1) Fixation-to-chance distance

	(1) M_{magn}	(2) M_{vector}	(3) $M_{p\&m}$	(4) M_{dir}
(1) M_{magn}	▨			
(2) M_{vector}	t = 0 - μ_D= 0.118 μ_0= 0.11	▨		
(3) $M_{p\&m}$	t = 0 - μ_D= 0.274 μ_0= 0.27	t=0 - μ_D= 0.156 μ_0= 0.15	▨	
(4) M_{dir}	t=-3.2 p<0.005 μ_D= -4.75 μ_0= -2.0	t=-3.2 p<0.005 μ_D= -4.87 μ_0= -2.0	t=-3.3 p<0.0025 μ_D= -5.031 μ_0= -2.0	▨

(2) Correlation coefficient

	(1) M_{magn}	(2) M_{vector}	(3) $M_{p\&m}$	(4) M_{dir}
(1) M_{magn}	▨			
(2) M_{vector}	t = 0.8 - μ_D= 0.0095 μ_0= 0.005	▨		
(3) $M_{p\&m}$	t = 0 - μ_D= 0.0024 μ_0= 0.002	t=1.02 - μ_D= -0.007 μ_0= –0.005	▨	
(4) M_{dir}	t=-2.3 p<0.025 μ_D= -0.058 μ_0= -0.03	t=-2.9 p<0.01 μ_D= -0.067 μ_0= -0.03	t=-2.3 p<0.025 μ_D= -0.061 μ_0= -0.03	▨

(3) Kullback Leibler divergence

(1) M_{magn}	▨			
(2) M_{vector}	t = 0.31 - μ_D= -0.0129 μ_0= 0.01	▨		
(3) $M_{p\&m}$	t = 0 - μ_D= -0.00009 μ_0= -0.0001	t=-0.1 - μ_D= 0.0128 μ_0= -0.01	▨	
(4) M_{dir}	t=-1.8 p<0.05 μ_D= 0.167 μ_0= 0.0	t=-2.0 p<0.05 μ_D= 0.18 μ_0= 0.0	t=-1.94 p<0.05 μ_D= 0.167 μ_0= 0.0	▨

(4) AUC value (ROC analysis)

(1) M_{magn}	▨			
(2) M_{vector}	t = 0.6 - μ_D= 0.0017 μ_0= 0.001	▨		
(3) $M_{p\&m}$	t = -0.7 - μ_D= -0.0025 μ_0= -0.001	t = -0.9 - μ_D= -0.0043 μ_0= -0.002	▨	
(4) M_{dir}	t=-2.3 p<0.025 μ_D= -0.018 μ_0= -0.01	t=-3.2 p<0.005 μ_D= -0.0198 μ_0= -0.01	t=-1.4 p<0.1 μ_D= -0.0154 μ_0= -0.01	▨

---

(1) Fixation-to-chance distance

	(1) M_{magn}	(2) M_{vector}	(3) $M_{p\&m}$	(4) M_{dir}
(1) M_{magn}	▨			
(2) M_{vector}	t = -0.2 - μ_D= -0.0005 μ_0= 0.0	▨		
(3) $M_{p\&m}$	t = 0.6 - μ_D= 0.165 μ_0= 0.16	t=0.2 - μ_D= 0.165 μ_0= 0.16	▨	
(4) M_{dir}	t = -0.6 - μ_D= -0.06 μ_0= 0.0	t = -0.6 μ_D= -0.06 μ_0= 0.0	t= 0.1 - μ_D= -0.228 μ_0= -0.22	▨

(2) Correlation coefficient

	(1) M_{magn}	(2) M_{vector}	(3) $M_{p\&m}$	(4) M_{dir}
(1) M_{magn}	▨			
(2) M_{vector}	t = 0.0 - μ_D= -0.0004 μ_0= -0.0004	▨		
(3) $M_{p\&m}$	t = -0.2 - μ_D= -0.0004 μ_0= 0.0	t=-0.04 - μ_D= -0.0009 μ_0= 0.0	▨	
(4) M_{dir}	t=0.2 - μ_D= 0.0018 μ_0= 0.0	t=0.3 - μ_D= 0.002 μ_0= 0.0	t=0.3 - μ_D= 0.002 μ_0= 0.0	▨

(3) Kullback Leibler divergence

(1) M_{magn}	▨			
(2) M_{vector}	t = -0.3 - μ_D= -0.0022 μ_0= 0.002	▨		
(3) $M_{p\&m}$	t = 0.5 - μ_D= -0.01 μ_0= 0.0	t=0.7 - μ_D= -0.012 μ_0= 0.0	▨	
(4) M_{dir}	t=-0.2 - μ_D= 0.012 μ_0= 0.0	t=-0.2 - μ_D= 0.009 μ_0= 0.0	t=-0.4 - μ_D= 0.022 μ_0= 0.0	▨

(4) AUC value (ROC analysis)

(1) M_{magn}	▨			
(2) M_{vector}	t = -0.1 - μ_D= –0.00041 μ_0= –0.0004	▨		
(3) $M_{p\&m}$	t = -0.4 - μ_D= -0.011 μ_0= -0.01	t = -0.2 - μ_D= -0.0105 μ_0= -0.01	▨	
(4) M_{dir}	t=0.5 - μ_D= 0.0021 μ_0= 0.0	t=0.6 - μ_D= 0.0025 μ_0= 0.0	t=0.6 - μ_D= 0.013 μ_0= 0.01	▨

Figure 8.10: Comparison of the model performances using paired t-test: evaluation of the first category (synthetic scenes) at the top, evaluation of the third category (natural real scenes) at the bottom).

each metric. The evaluation for synthetic scenes is presented at the top, while it is at the bottom for natural real scenes.

For each comparison, the t-value is presented, as well as the level of significance (p-value). μ_D refers to the average of the difference and the parameter μ_0 is used in order to test wether the average μ_D is significantly different than μ_0.

By analyzing the results, we note that the four models are not significantly different for natural real scenes. Regarding synthetic scenes, μ_D and p-values indicate lower performances of the direction model $\mathcal{M}_{dir}$ in comparison to the others.

Therefore, the quantitative evaluation confirms the conclusions of the qualitative evaluation. The score values of the different metrics show high correspondences between the human and computer saliency maps of the four models. In addition, for video sequences with fixed background, the magnitude model show globally similar performances compared to the three other models. This result is expected, since video sequences with fixed background is restricted to motion contrast in magnitude. Finally, we note the lower suitability of the direction model for synthetic scenes, which is due to the motion field computation rather than the model itself.

8.5.2 Moving background

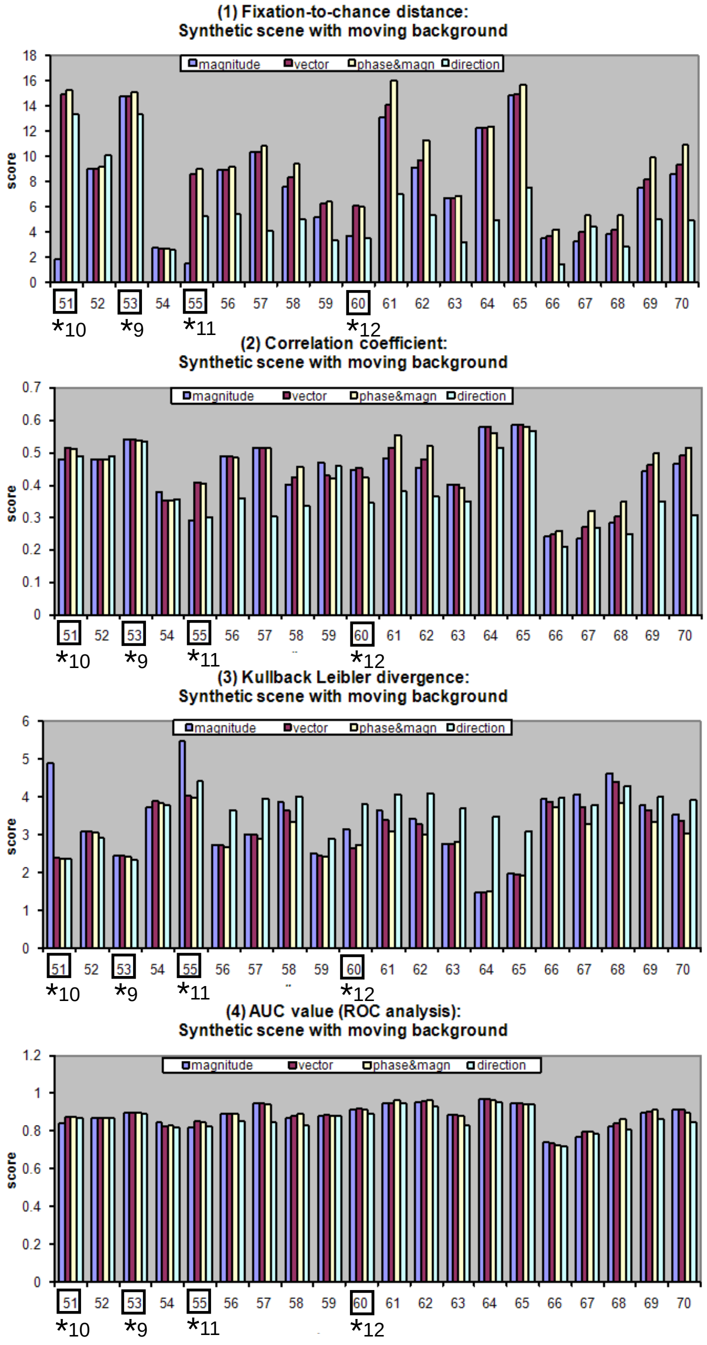
Figure 8.11 and 8.12 show the score distribution for synthetic and natural real scenes. First, we analyze the examples presented in the qualitative evaluation and investigate wether both evaluations conduct to the same conclusions.

In example 9 (Figure 8.11), motion contrast is in magnitude, while it is both in phase and magnitude in example 13 (Figure 8.12). In both cases, the scores have approximatively the same values and the four models perform similarly. This illustrates the capability of the magnitude model to highlight salient moving stimuli when the speed magnitude of the later is either slower (example 9) or faster (example 13) than the moving background.

In examples 10, 11, 12 (synthetic scenes) and 14, 15, 16 (natural real scenes), motion contrast is in phase. The scores values highlight the low performances of the magnitude model, while the three other models show higher performances. Therefore, this confirms the observations mentioned in the qualitative evaluation.

Next we present statistical results in order to perform a global model evaluation on the whole video sequence set. As previously, a non-parametric paired t-test is used. Figure 8.13 presents the statistical analysis.

We point out two observations. First, we compare the magnitude model to the models highlighting both phase and magnitude contrasts. The results illustrate globally higher performances of both vector model $\mathcal{M}_{vector}$ and phase & magnitude model $\mathcal{M}_{p\&m}$ compared to the magnitude model $\mathcal{M}_{magn}$. Moreover, we note that the differences in terms of performances are higher for synthetic scenes (tables at the top) than for natural real scenes (tables at the bottom). These differences between both categories are explained by the experimental design. Indeed, we included more video sequences containing pure phase contrasts in the synthetic category than in the natural real category.


(1) Fixation-to-chance distance: Synthetic scene with moving background

(2) Correlation coefficient: Synthetic scene with moving background

(3) Kullback Leibler divergence: Synthetic scene with moving background

(4) AUC value (ROC analysis): Synthetic scene with moving background

Figure 8.11: Quantitative evaluation of the second category (synthetic scenes with moving background): the examples presented in the qualitative evaluation are indicated by an asterisk.

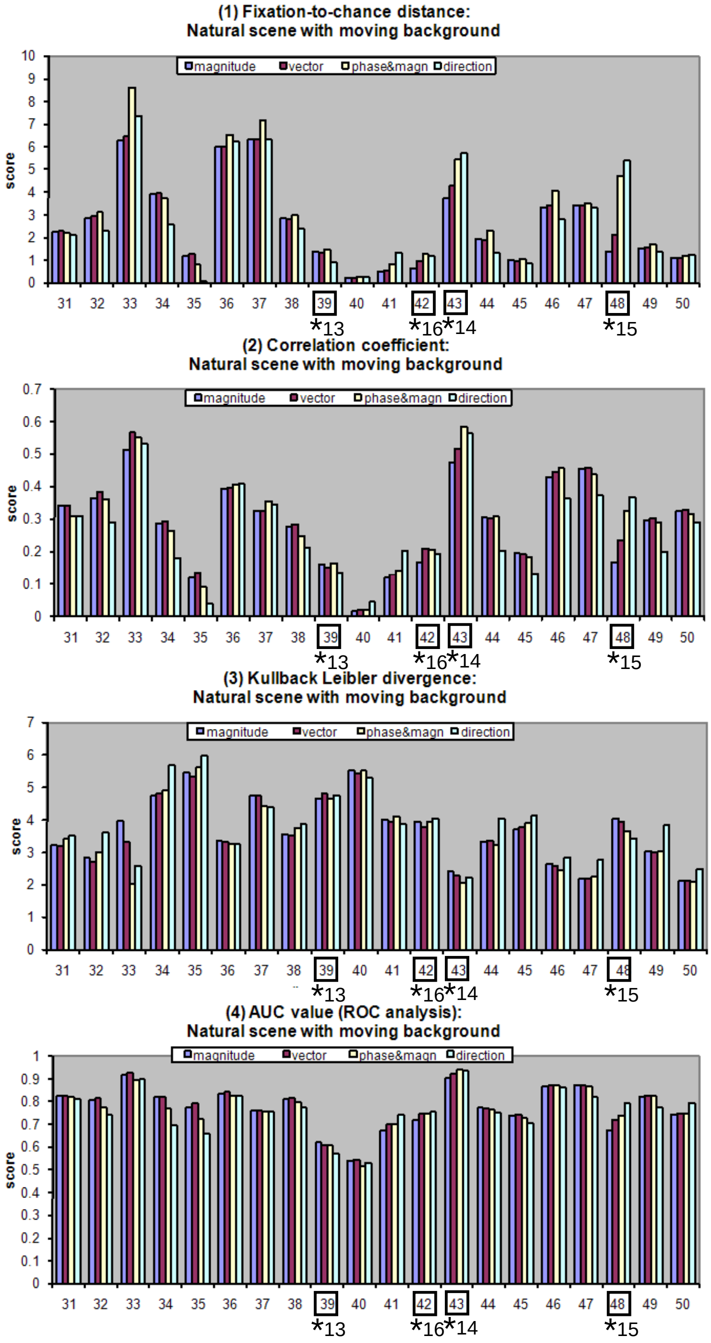


Figure 8.12: Quantitative evaluation of the fourth category (natural real scenes with moving background): the examples presented in the qualitative evaluation are indicated by an asterisk.

(1) Fixation-to-chance distance

	(1) M_{magn}	(2) M_{vector}	(3) $M_{p\&m}$	(4) M_{dir}
(1) M_{magn}	▨			
(2) M_{vector}	$t = 2.0$ $p<0.05$ μ_D= 1.44 μ_0= 0.0	▨		
(3) $M_{p\&m}$	$t = 3.1$ $p<0.005$ μ_D= 2.14 μ_0= 0.0	$t=4.9$ $p<0.0005$ μ_D= 0.70 μ_0= 0.0	▨	
(4) M_{dir}	$t=-1.9$ $p<0.05$ μ_D= -1.75 μ_0= 0.0	$t=-2.2$ $p<0.025$ μ_D= -3.21 μ_0= -2.0	$t=-3.2$ $p<0.0025$ μ_D= -3.91 μ_0= -2.0	▨

(2) Correlation coefficient

	(1) M_{magn}	(2) M_{vector}	(3) $M_{p\&m}$	(4) M_{dir}
(1) M_{magn}	▨			
(2) M_{vector}	$t = 2.0$ $p<0.05$ μ_D= 0.014 μ_0= 0.0	▨		
(3) $M_{p\&m}$	$t = 2.4$ $p<0.025$ μ_D= 0.024 μ_0= 0.0	$t=1.8$ $p<0.05$ μ_D= 0.01 μ_0= 0.0	▨	
(4) M_{dir}	$t=-2.6$ $p<0.01$ μ_D= -0.056 μ_0= -0.02	$t=-2.7$ $p<0.01$ μ_D= -0.07 μ_0= -0.03	$t=-3.0$ $p<0.005$ μ_D= -0.08 μ_0= -0.03	▨

(3) Kullback Leibler divergence

(1) M_{magn}	▨			
(2) M_{vector}	$t = 2.2$ $p<0.025$ μ_D= -0.298 μ_0= 0.0	▨		
(3) $M_{p\&m}$	$t =3.2$ $p<0.0025$ μ_D= -0.443 μ_0= 0.0	$t=3.6$ $p<0.001$ μ_D= -0.144 μ_0= 0.0	▨	
(4) M_{dir}	$t=-1.1$ - μ_D= 0.216 μ_0= 0.0	$t=-2.1$ $p<0.025$ μ_D= 0.515 μ_0= 0.25	$t=-3.5$ $p<0.001$ μ_D= 0.66 μ_0= 0.25	▨

(4) AUC value (ROC analysis)

(1) M_{magn}	▨			
(2) M_{vector}	$t = 2.1$ $p<0.025$ μ_D= 0.006 μ_0= 0.0	▨		
(3) $M_{p\&m}$	$t = 1.7$ $p<0.05$ μ_D= 0.006 μ_0= 0.0	$t = 0.2$ - μ_D= 0.0004 μ_0= 0.0	▨	
(4) M_{dir}	$t=-3.0$ $p<0.005$ μ_D= -0.02 μ_0= 0.0	$t=-2.8$ $p<0.005$ μ_D= -0.027 μ_0= -0.01	$t=-2.9$ $p<0.005$ μ_D= -0.027 μ_0= -0.01	▨

---

(1) Fixation-to-chance distance

	(1) M_{magn}	(2) M_{vector}	(3) $M_{p\&m}$	(4) M_{dir}
(1) M_{magn}	▨			
(2) M_{vector}	$t = 2.1$ $p<0.025$ μ_D= 0.098 μ_0= 0.0	▨		
(3) $M_{p\&m}$	$t = 2.7$ $p<0.01$ μ_D= 0.55 μ_0= 0.0	$t=2.7$ $p<0.01$ μ_D= 0.451 μ_0= 0.0	▨	
(4) M_{dir}	$t = 0.6$ - μ_D= 0.164 μ_0= 0.0	$t = 0.3$ - μ_D= 0.065 μ_0= 0.0	$t = -3.1$ $p<0.005$ μ_D= -0.385 μ_0= 0.0	▨

(2) Correlation coefficient

	(1) M_{magn}	(2) M_{vector}	(3) $M_{p\&m}$	(4) M_{dir}
(1) M_{magn}	▨			
(2) M_{vector}	$t = 3.0$ $p<0.005$ μ_D= 0.014 μ_0= 0.0	▨		
(3) $M_{p\&m}$	$t = 1.4$ $p<0.1$ μ_D= 0.014 μ_0= 0.0	$t=0.04$ - μ_D= 0.0003 μ_0= 0.0	▨	
(4) M_{dir}	$t=-1.0$ - μ_D= -0.017 μ_0= 0.0	$t=-2.1$ $p<0.025$ μ_D= -0.031 μ_0= 0.0	$t=-3.1$ $p<0.005$ μ_D= -0.031 μ_0= 0.0	▨

(3) Kullback Leibler divergence

(1) M_{magn}	▨			
(2) M_{vector}	$t = 2.0$ $p<0.05$ μ_D= -0.066 μ_0= 0.0	▨		
(3) $M_{p\&m}$	$t = 1.0$ - μ_D= -0.106 μ_0= 0.0	$t=0.5$ - μ_D= -0.039 μ_0= 0.0	▨	
(4) M_{dir}	$t=-1.3$ - μ_D= 0.158 μ_0= 0.0	$t=-2.2$ $p<0.025$ μ_D= 0.225 μ_0= 0.0	$t=-3.5$ $p<0.001$ μ_D= 0.264 μ_0= 0.0	▨

(4) AUC value (ROC analysis)

(1) M_{magn}	▨			
(2) M_{vector}	$t = 2.9$ $p<0.005$ μ_D= 0.009 μ_0= 0.0	▨		
(3) $M_{p\&m}$	$t = -0.5$ - μ_D= -0.003 μ_0= 0.0	$t =-2.5$ $p<0.025$ μ_D= -0.012 μ_0= 0.0	▨	
(4) M_{dir}	$t=-1.1$ - μ_D= -0.015 μ_0= 0.0	$t=-2.1$ $p<0.025$ μ_D= -0.023 μ_0= 0.0	$t=-1.4$ $p<0.1$ μ_D= -0.011 μ_0= 0.0	▨

Figure 8.13: Comparison of the model performances using paired t-test: evaluation of the second category (synthetic scenes) at the top, evaluation of the fourth category (natural real scenes) at the bottom).

Regarding the direction model $\mathcal{M}_{dir}$, surprisingly, the statistical analysis for the synthetic category shows lower performances compared to the magnitude model $\mathcal{M}_{magn}$. This is explained by the presence of motion artifacts in the motion field computation, which reduces the model performances. Besides, we already mentioned this problem in the qualitative evaluation. This problem is due to a non-optimal recombination of intermediate motion fields. Therefore, using an alternative motion recombination approach, we believe that the direction model $\mathcal{M}_{dir}$ would probably outperforms the magnitude model $\mathcal{M}_{magn}$.

The second observation discusses the overall model performances. Over the four models, the analysis suggests that the phase & magnitude model $\mathcal{M}_{p\&m}$ is the most suitable one. For example, for synthetic scenes, μ_D ranges from 0.7 to 3.91 for the fixation-to-chance distance and from -0.14 to -0.66 for the Kullback-Leibler divergence, both with significant p-values ($0.0005 < p < 0.0025$). We note the positive differences for the fixation-to-chance distance and the negative one for the Kullback value, which are respectively a measure of similarity and dissimilarity.

Over the three models highlighting both phase and magnitude contrasts, the evaluation indicates slight higher performances of the phase & magnitude model $\mathcal{M}_{p\&m}$. The higher suitability may be induced by the decoupling of phase and magnitude contrasts. Indeed, the motion saliency computation results in the fusion of two independent contrast maps, both having an equal contribution. Alternatively, the vector model is based on a unique contribution highlighting relative motion contrast, which is computed by vector difference. While both models perform similarly in simple situations, typically the synthetic sequences containing one unique salient stimulus (Figure 8.6: example 10 and 11, Figure 8.7: example 13), the quantitative evaluation shows higher performances of the $\mathcal{M}_{p\&m}$ model in more complex situations (example 14, 15 and 16, Figure 8.7), which include several salient moving targets. Actually, the vector model tends to promote high relative magnitude contrast and demote low relative phase contrast, while the $\mathcal{M}_{p\&m}$ model tends to equilibrate both contributions.

8.6 Additional discussions

8.6.1 Pop-out effect

A pop-out stimulus refers to a unique region in the image, which differs from the rest of the image according to a single feature. Typically, a red flower lying on a uniform background of grass will strongly attract human attention. In visual search experiments, it has been shown that such stimuli are easily found by human subjects [64]. In this subsection, we illustrate experimentally the existence of pop-out stimuli induced by the motion feature, as well as the capability of the computer model to highlight them.

Figure 8.14 shows several video examples containing salient motion stimuli. Strong motion pop-out effects are visible. Indeed, we can see in the human saliency map that most human subjects focus on the moving target. Regarding the computer model, most of the time, the pop-out stimulus constitutes the most salient location in the saliency map. Therefore, in presence of

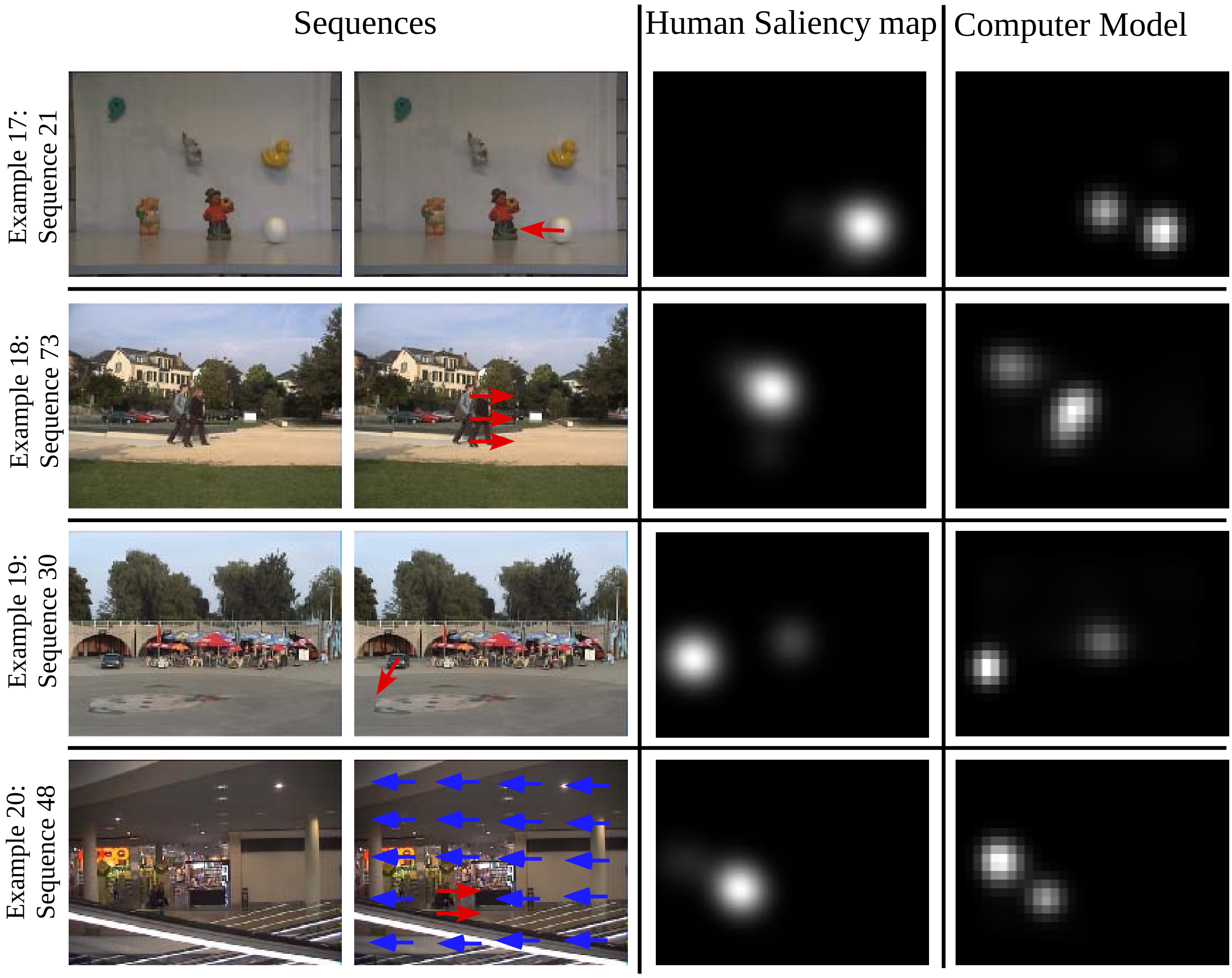


Figure 8.14: Motion pop-out effect: salient moving stimuli tend to catch globally the human visual attention.

Table 8.2: Quantitative evaluation: correlation coefficient at the top, AUC value at the bottom.

1st category: synthetic scenes with fixed background

	Fixation-to-chance distance	correlation coefficient	Kullback-Leibler divergence	AUC value (ROC analysis)
$\mathcal{M}_{magn}$	9.65 ± 4.09	0.539 ± 0.113	3.28 ± 0.67	0.885 ± 0.099
$\mathcal{M}_{vector}$	9.77 ± 4.13	0.548 ± 0.112	3.26 ± 0.67	0.887 ± 0.1
$\mathcal{M}_{p\&m}$	9,92 ± 4.21	0.541 ± 0.111	3.28 ± 0.64	0.883 ± 0.098
$\mathcal{M}_{dir}$	4.89 ± 1.69	0.481 ± 0.092	3.44 ± 0.59	0.867 ± 0.099
$\mathcal{M}_{static}$	5.88 ± 2.72	0.284 ± 0.105	5.68 ± 1.16	0.769 ± 0.158

2nd category: natural real scenes with fixed background

	Fixation-to-chance distance	correlation coefficient	Kullback-Leibler divergence	AUC value (ROC analysis)
$\mathcal{M}_{magn}$	3.91 ± 1.91	0.37 ± 0.125	3.19 ± 0.97	0.821 ± 0.082
$\mathcal{M}_{vector}$	3.91 ± 1.91	0.37 ± 0.125	3.19 ± 0.97	0.82 ± 0.082
$\mathcal{M}_{p\&m}$	4.07 ± 1.95	0.369 ± 0.126	3.17 ± 0.97	0.81 ± 0.085
$\mathcal{M}_{dir}$	3.84 ± 2.03	0.372 ± 0.131	3.2 ± 0.99	0.823 ± 0.074
$\mathcal{M}_{static}$	1.1 ± 1.77	0.143 ± 0.152	4.52 ± 1.02	0.683 ± 0.124

3rd category: synthetic scenes with moving background

	Fixation-to-chance distance	correlation coefficient	Kullback-Leibler divergence	AUC value (ROC analysis)
$\mathcal{M}_{magn}$	7.39 ± 4.2	0.434 ± 0.102	3.4 ± 0.97	0.878 ± 0.061
$\mathcal{M}_{vector}$	8.83 ± 3.85	0.448 ± 0.094	3.1 ± 0.75	0.884 ± 0.057
$\mathcal{M}_{p\&m}$	9.53 ± 3.96	0.457 ± 0.088	2.96 ± 0.64	0.885 ± 0.06
$\mathcal{M}_{dir}$	5.63 ± 3.24	0.378 ± 0.099	3.62 ± 0.6	0.857 ± 0.058
$\mathcal{M}_{static}$	4.38 ± 3.12	0.308 ± 0.103	4.6 ± 1.09	0.863 ± 0.058

4th category: natural real scenes with moving background

	Fixation-to-chance distance	correlation coefficient	Kullback-Leibler divergence	AUC value (ROC analysis)
$\mathcal{M}_{magn}$	2.6 ± 1.89	0.287 ± 0.134	3.67 ± 1.0	0.774 ± 0.095
$\mathcal{M}_{vector}$	2.7 ± 1.9	0.301 ± 0.138	3.6 ± 1.0	0.783 ± 0.094
$\mathcal{M}_{p\&m}$	3.15 ± 2.34	0.301 ± 0.145	3.56 ± 1.1	0.771 ± 0.096
$\mathcal{M}_{dir}$	2.76 ± 2.21	0.27 ± 0.142	3.83 ± 1.03	0.759 ± 0.097
$\mathcal{M}_{static}$	1.34 ± 2.09	0.201 ± 0.155	4.17 ± 1.0	0.723 ± 0.105

motion pop-out effect, the computer model reproduces efficiently the human visual behavior.

Finally, we investigate the human attention behavior when static and moving pop-out stimuli compete. As we can see, each example contains a static salient stimulus, which competes against a motion one. Moreover, the computer model highlights both stimuli, using the classical map integration strategy (competitive scheme). According to the sequence content, the computer saliency map highlights the static stimulus as the most salient target (example 20), or the motion one (examples 17, 18 and 19).

By analyzing the experimental data, the human saliency maps tend to show that, in presence of motion pop-out effect, human visual attention focuses entirely on the moving stimuli. The static contribution seems to be strongly inhibited. Therefore, for computer modeling issues, the competitive scheme does not seem to be the most appropriate strategy to integrate static and motion contributions. The motion priority scheme is expected to be more suitable to predict the average human visual behavior. In presence of salient moving stimuli, the motion priority scheme provides the priority to the motion cue by suppressing the static contribution. We will discuss the motion integration schemes evaluation more in details in Chapter 9.

8.6.2 Top-down and bottom-up influence

We have previously mentioned in the introduction that attention can be controlled either in an unconscious manner or in a voluntary manner. The former is called bottom-up attention, the later top-down attention. While the scope of the thesis deals with bottom-up computer modeling, a bias induced by top-down factors is expected in the model evaluation. In this section, we briefly discuss the influence of top-down attention over the experiments.

The design of the experiments is defined in order to reduce as much as possible top-down influences. We ask human subjects to freely look at the screen without specific task. Short sequence duration (10 sec.) is set. Indeed, recent studies evaluate experimentally computer models using short viewing durations [63, 65], supposing that bottom-up influence is varying over the viewing time and greatest just after stimulus onset. We mention another recent study [31], which conducts to opposite conclusion. The authors conclude that bottom-up influence still remains important over the time and in a free-viewing task, attentional allocation is continuously and strongly driven by low-level visual features.

Regarding our experimental design, we include both synthetic and real scenes, with the idea to compare top-down influence between both categories. Table 8.2 presents the average score values for the different motion models according to the synthetic and natural real categories. By comparing the first to the second category, and the third to the fourth category, we can see that the scores computed for synthetic scenes are globally higher than those computed for the natural real scenes. This suggests that the nature of the scene influences the model performances. The lower performances for natural real scenes is probably induced by stronger top-down influences. Besides, the content of the real scenes is much more complex. By analyzing the locations of the human eye movement patterns, we have noticed evident top-down influences. Typically, human subjects focus on lettering or human faces, which are not necessarily salient according to bottom-up features. In addition, we notice a shift between experimental and computer data when human subjects look at moving persons. As illustrated in example 18 (Figure 8.14), the average visual behavior tends to focus frequently on the face of moving persons while the computer model highlights the center of the body.

In conclusion, these experiments illustrate stronger top-down influence over natural real scenes than synthetic scenes.

8.7 Chapter summary

This chapter presents the dynamic model evaluation. Four computer models are considered, each one combining a static contribution defined by the static model, and a motion contribution defined by one of the four motion models: (1) the magnitude $\mathcal{M}_{magn}$, (2) the vector $\mathcal{M}_{vector}$, (3) the phase & magnitude $\mathcal{M}_{p\&m}$ and (4) the direction $\mathcal{M}_{dir}$ models.

An experimental frame using psycho-physical experiments is used to assess the model suitability. The experiments include video sequences of different nature (synthetic and natural real scenes, acquired with fixed and moving background), showing advantages and inconveniences of the different models.

A qualitative evaluation is performed by comparing visually the experimental and computer data, respectively represented by human and computer saliency maps. In addition, a quantitative evaluation is performed by measuring the correspondences between the experimental and computer data. Four metrics used in the state of the art are considered: the fixation-to-chance distance, the correlation coefficient, the Kullback-Leibler divergence and the ROC analysis.

The different investigations conduct to the following conclusions:

- First, a preliminary experiment, consisting in analyzing the experimental human saliency maps, investigates wether the most fixated regions are located on motion contrasts. The analysis illustrates the sensitivity of human attention to specific motion contrasts, namely the magnitude and phase contrasts. It confirms the importance of motion in VA modeling and defines the requirements of a suitable dynamic computer model, i.e. the ability of highlighting such motion contrasts.

- Second, the model evaluation is divided in two parts, video sequences with fixed and moving background. Both qualitative and quantitative evaluations conduct to the same conclusions.

  With fixed background, the magnitude model is suitable and performs similarly compared to the three other models. This result is expected, since video with fixed background is restricted to motion contrast in magnitude.

  With moving background, the suitability of the magnitude model depends on the nature of motion contrasts. In presence of magnitude contrast or both phase and magnitude contrasts, the four models perform similarly. This illustrates the capability of the magnitude model to highlight salient moving stimuli when the speed magnitude of the later is either slower or faster than the moving background. In presence of pure phase contrast, the magnitude model is not suitable, while the vector, phase & magnitude, and direction models are more suitable to predict the human saliency. $\mathcal{M}_{magn}$ highlights magnitude contrasts only, while the three models $\mathcal{M}_{vector}$, $\mathcal{M}_{p\&m}$ and $\mathcal{M}_{dir}$ highlight both phase and magnitude contrasts.

- Third, regarding the models which highlight both phase and magnitude contrast, the evaluation indicates slight higher performances of the phase & magnitude model. This model decouples phase and magnitude contrasts, which tend to equilibrate both contributions in the computation of the motion saliency map.
- Fourth, these experiments illustrate stronger top-down influence for natural real scenes compared to synthetic scenes.
- Finally, we illustrate the existence of pop-out moving stimuli, which tend to catch entirely human visual attention, even in presence of static pop-out stimuli. For computer modeling issues, the motion priority scheme seems to be more appropriate to predict the average human visual behavior than the competitive scheme. This will be the scope of the next chapter.

Chapter 9

Motion integration scheme evaluation

9.1 Chapter introduction

As mentioned previously, both static and motion contributions are required in the design of a dynamic model. While the previous chapter discussed the dynamic model evaluation including the four motion models, this chapter presents the motion integration schemes evaluation. Three dynamic models are considered, by combining the static and motion cues using one of the three motion integration schemes defined in Chapter 5:

$$\begin{gathered}(1)\ \textit{Cue competition} : S_{cuecomp} = \mathcal{N}(C_{color}) + \mathcal{N}(C_{int}) + \mathcal{N}(C_{orient}) + \mathcal{N}(C_{motion}),\\ (2)\ \textit{Static\&motion competition} : S_{static\&motion} = \mathcal{N}(S_{static}) + \mathcal{N}(S_{motion}),\\ (3)\ \textit{Motion priority} : S_{priority} = \begin{cases} S_{motion} & if \qquad \Phi(S_{motion}) > T_m \\ \mathcal{N}(S_{static}) + \mathcal{N}(S_{motion}) & if \quad T_s < \ \Phi(S_{motion}) < T_m \\ S_{static} & if \qquad \Phi(S_{motion}) < T_s \end{cases}\end{gathered} \tag{9.1}$$

The performances of the motion integration schemes are evaluated using the same experimental frame as in the previous chapter. In addition, the same set of 84 video sequences is used.

The chapter is structured in two parts. Section 9.2 and 9.3 describe respectively the qualitative and quantitative evaluation.

9.2 Qualitative model evaluation

This section presents the qualitative evaluation. In order to assess the correspondences between the experimental and computer data, the human saliency maps are compared visually to the computer saliency maps issued from the different motion integration schemes.

Several representative examples are shown in Figure 9.1, illustrating the differences between the integration schemes. (A) shows the original frame. Each one is annotated by arrows indicating motion. The experimental data correspond to the human observations (B) and the human saliency map (C). (1) to (3) represent respectively the computer saliency maps issued from the cue competition, static & motion competition and the motion priority schemes. In addition, the static saliency map (4) is also presented in order to compare both static and motion contributions in the saliency maps.

Before going into the details of the model evaluation, we discuss the computer models and the experimental human saliency maps separately. First, as we can see, there are differences between the models. The static and motion contributions in the saliency map depend on the integration scheme. Motion competes for approximatively 25% in the cue competitive scheme, while for 50% in the static & motion scheme. In the priority scheme, the static contribution is inhibited and the saliency map is entirely dominated by the motion cue. Second, by analyzing the human saliency maps, we can see in each example the presence of one salient moving stimulus (indicated by red arrows), which tends to catch the global attention. It is particularly striking in example 21, most eye movement patterns are located on the salient moving target and rarely on the static salient stimulus. In other words, motion cue tends to catch globally human attention, independently of the static cue.

By comparing the human and computer saliency maps, the priority scheme provides the highest level of correspondences over all schemes and is therefore the most suitable one. Globally, the qualitative evaluation suggests the following performance ranking: #1 the motion priority scheme, #2 the static & motion competition scheme and #3 the cue competition scheme. The performance variations simply illustrate the influence of the motion contribution in the resulting saliency map. The more motion contributes to the saliency map, the more the model correlates to the average human visual behavior. Therefore, the most suitable scheme is the motion priority scheme which suppresses the static contribution in presence of salient moving stimuli, while the cue competition and static & motion competition schemes are less suitable due to static contribution.

9.3 Quantitative model evaluation

While the qualitative evaluation conducted to the highest suitability of the motion priority scheme, we investigate in this section wether the quantitative evaluation conducts to identical conclusions. In a similar way to the evaluation presented in Chapter 8, four metrics are used to measure the similarity between the human and computer saliency maps. In order to compare the performances of the three motion integration schemes, each of the four motion cues (C_{magn}, C_{vector}, $C_{p\&m}$ and C_{dir}) is integrated to the static cue using each of the three motion integration schemes. Therefore, twelve model variations are considered.

A paired t-test is applied on the whole sequence set and the motion integration schemes

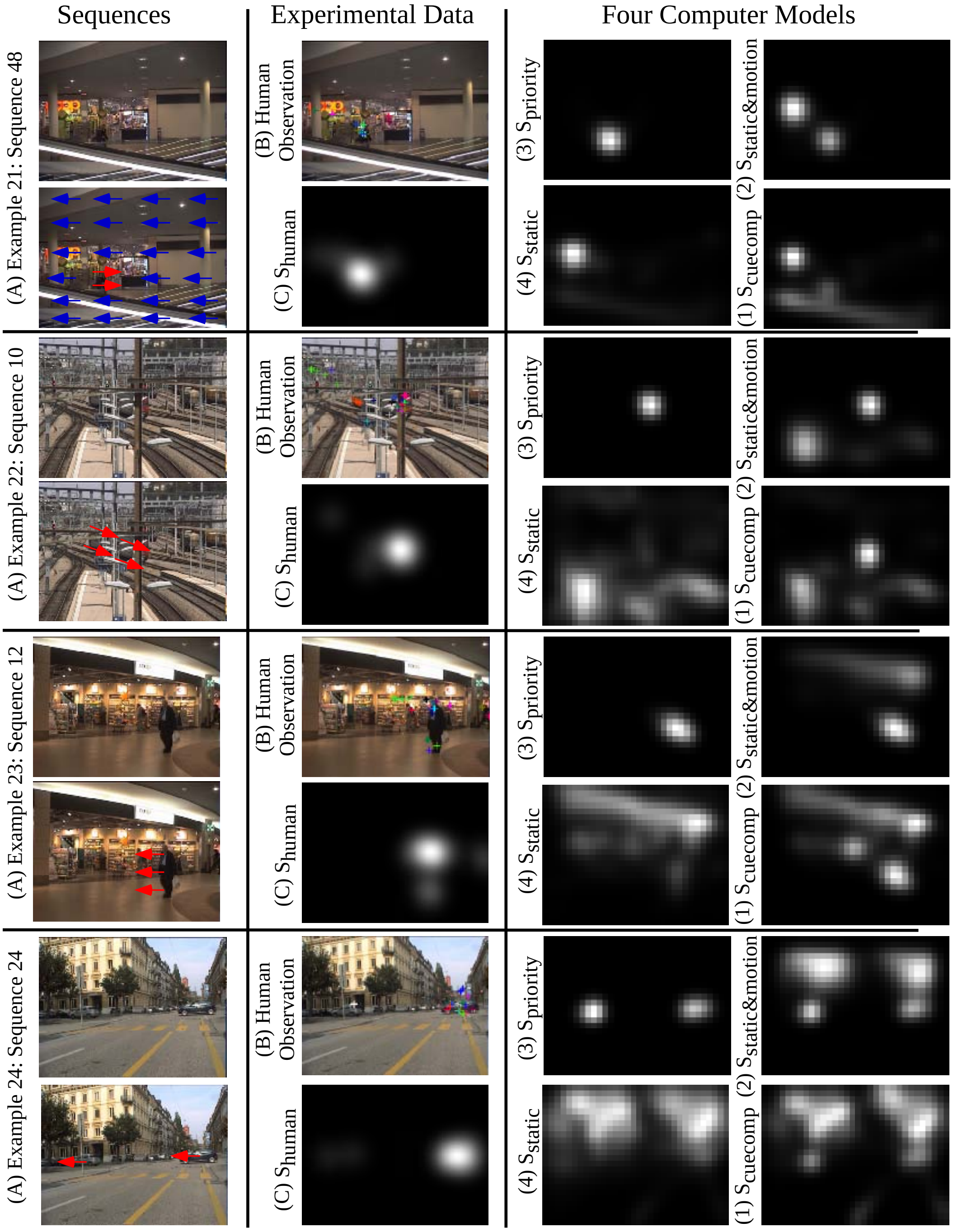


Figure 9.1: Qualitative evaluation of the motion integration schemes: (A) the original frame; (B) the human observations; (C) the human saliency map issued from the fixation and smooth pursuit periods; (1) to (3) are respectively the saliency maps resulting from the cue competition, static & motion competition and motion priority schemes; (4) the static saliency map.

are compared pair by pair: (1) cue competition versus (2) static & motion competition, (1) cue competition versus (3) motion priority, and finally, (2) static & motion competition versus (3) motion priority. The results are presented for the fixation-to-chance distance and Kullback-Leibler divergence in Table 9.1, and for the correlation coefficient and AUC value in Table 9.2. For each pair, the t-value is presented, as well as the level of significance (p-value). μ_D refers to the average of the difference and the parameter μ_0 is used in order to test wether the average of the difference μ_D is significantly different than μ_0.

Regarding the first pair evaluation, static & motion competition versus cue competition, the statistics show higher performances of the static & motion scheme. All metrics have significant p-values with high average difference μ_D. As example, for the fixation-to-chance distance, μ_D value ranges from 0.63 to 1.66 with p-value between $0.0005 < p < 0.01$. For the Kullback-Leibler divergence, μ_D value ranges from -0.81 to -1.07 with p-value between $0.0005 < p < 0.0025$. We mention the negative value of the average difference, since the metric measures the dissimilarity.

Regarding the second pair evaluation, motion priority versus cue competition, the statistics show higher performances of the static & motion scheme. As example, μ_D value ranges from 2.43 to 5.06 for the fixation-to-chance distance, and from 0.051 to 0.114 for the correlation coefficient.

Finally, the third pair evaluation show higher performances of the motion priority scheme compared to the static & motion scheme. μ_D value ranges from 1.793 to 3.17 for the fixation-to-chance distance. Regarding the Kullback-Leibler divergence, the μ_D value for $\mathcal{M}_{magn}$, $\mathcal{M}_{p\&m}$ and $\mathcal{M}_{dir}$ range from -0.47 to -0.56.

Therefore, the statistic analysis confirms the qualitative evaluation. The motion priority scheme is the most suitable motion integration scheme. In presence of salient motion stimuli, this strategy provides the priority to the motion cue by suppressing entirely the static contribution in the saliency map. Statistically, this motion integration strategy turns out to be the most efficient way to predict the average human visual behavior. Human subjects tend to focuss their attention more frequently on salient moving stimuli and rarely on static salient stimuli.

Table 9.1: Quantitative evaluation: a non-parametric paired t-test is used to compare the different motion integration schemes. Fixation-to-chance distance at the top, Kullback-Leibler divergence at the bottom.

Fixation-to-chance distance:

		(2) $S_{static\&motion}$	(3) $S_{priority}$	(3) $S_{priority}$
		vs	vs	vs
		(1) $S_{cuecomp}$	(1) $S_{cuecomp}$	(2) $S_{static\&motion}$
$\mathcal{M}_{magn}$:	t and p-value	3.6 $p<0.0005$	4.68 $p<0.0005$	3.01 $p<0.001$
	μ_D	1.36	4.22	2.86
	μ_0	0.5	2.0	2.0
$\mathcal{M}_{vector}$:	t and p-value	2.77 $p<0.005$	5.8 $p<0.0005$	3.77 $p<0.0005$
	μ_D	1.66	4.74	3.08
	μ_0	1.0	2.0	2.0
$\mathcal{M}_{p\&m}$:	t and p-value	3.63 $p<0.0005$	6.32 $p<0.0005$	3.9 $p<0.0005$
	μ_D	1.89	5.06	3.17
	μ_0	1.0	2.0	2.0
$\mathcal{M}_{dir}$:	t and p-value	2.4 $p<0.01$	3.36 $p<0.001$	3.22 $p<0.001$
	μ_D	0.63	2.43	1.79
	μ_0	0.0	1.0	1.0

Kullback-Leibler divergence:

		(2) $S_{static\&motion}$	(3) $S_{priority}$	(3) $S_{priority}$
		vs	vs	vs
		(1) $S_{cuecomp}$	(1) $S_{cuecomp}$	(2) $S_{static\&motion}$
$\mathcal{M}_{magn}$:	t and p-value	5.3 $p<0.0005$	3.51 $p<0.0005$	2.9 $p<0.0025$
	μ_D	-1.0	-1.52	-0.51
	μ_0	0.5	1.0	0.25
$\mathcal{M}_{vector}$:	t and p-value	5.84 $p<0.0005$	4.31 $p<0.0005$	3.57 $p<0.0005$
	μ_D	-1.06	-1.62	-0.56
	μ_0	0.5	1.0	0.25
$\mathcal{M}_{p\&m}$:	t and p-value	5.86 $p<0.0005$	3.53 $p<0.0005$	2.35 $p<0.025$
	μ_D	-1.07	-1.55	-0.47
	μ_0	0.5	1.0	0.25
$\mathcal{M}_{dir}$:	t and p-value	3.07 $p<0.0025$	2.31 $p<0.025$	0.14 -
	μ_D	-0.81	-0.79	0.01
	μ_0	0.5	0.4	0.0

Table 9.2: Quantitative evaluation: correlation coefficient at the top, AUC value at the bottom.

Correlation coefficient:

		(2) $S_{static\&motion}$	(3) $S_{priority}$	(3) $S_{priority}$
		vs	vs	vs
		(1) $S_{cuecomp}$	(1) $S_{cuecomp}$	(2) $S_{static\&motion}$
$\mathcal{M}_{magn}$:	t and p-value	5.51 p<0.0005	4.92 p<0.0005	0.81 -
	μ_D	0.104	0.11	0.0054
	μ_0	0.05	0.05	0.0
$\mathcal{M}_{vector}$:	t and p-value	5.76 p<0.0005	5.26 p<0.0005	0.99 -
	μ_D	0.107	0.114	0.0064
	μ_0	0.05	0.05	0.0
$\mathcal{M}_{p\&m}$:	t and p-value	5.45 p<0.0005	3.26 p<0.001	-0.99 -
	μ_D	0.102	0.094	-0.0076
	μ_0	0.05	0.05	0.0
$\mathcal{M}_{dir}$:	t and p-value	2.68 p<0.005	2.23 p<0.025	-3.21 p<0.001
	μ_D	0.076	0.051	-0.025
	μ_0	0.05	0.02	0.0

AUC value (ROC analysis):

		(2) $S_{static\&motion}$	(3) $S_{priority}$	(3) $S_{priority}$
		vs	vs	vs
		(1) $S_{cuecomp}$	(1) $S_{cuecomp}$	(2) $S_{static\&motion}$
$\mathcal{M}_{magn}$:	t and p-value	4.45 p<0.0005	1.02 -	-3.02 p<0.005
	μ_D	0.02	0.007	-0.013
	μ_0	0.0	0.0	0.0
$\mathcal{M}_{vector}$:	t and p-value	4.67 p<0.0005	0.86 -	-3.35 p<0.001
	μ_D	0.021	0.006	-0.015
	μ_0	0.0	0.0	0.0
$\mathcal{M}_{p\&m}$:	t and p-value	3.21 p<0.001	-1.78 p<0.005	-4.81 p<0.0005
	μ_D	0.015	-0.015	-0.029
	μ_0	0.0	0.0	0.0
$\mathcal{M}_{dir}$:	t and p-value	2.08 p<0.025	-3.39 p<0.001	-5.9 p<0.0005
	μ_D	0.011	-0.034	-0.045
	μ_0	0.0	0.0	0.0

Chapter 10

Conclusions and perspectives

This thesis investigates the design of bottom-up VA models dedicated to video sequences. Named as dynamic visual attention model, such a model provides an automatic selection of potential regions of interest all over the sequence duration. The selection process relies on motion as well as on static feature contrasts. All this information is extracted from the video sequence and integrated in a competitive way to generate the resulting saliency map.

The design of a dynamic model can therefore be divided according to three main axes: (i) static model design, (ii) motion model design and (iii) integration of both models.

In this thesis, modeling and implementation issues are examined, with a main focus on the motion model and its integration. Moreover, the methodology used to evaluate experimentally the model performances is described. Psycho-physics experiments are used to assess the model suitability in comparison with human visual attention. The main contributions, limitations and perspectives are presented below according to the three axes.

10.1 Static model

The presented static model relies extensively on the classical saliency-based model of VA [5]. Compared to the classical one, we propose a static model that shares similar concepts, with several differences regarding the map integration schemes.

As first contribution, the non-linear exponential map transform $\mathcal{N}_{exp}$ is proposed as alternative to the non-linear DoG iterative $\mathcal{N}_{iter}$. As second contribution, the long-term normalization $\mathcal{N}_{LT}$ is proposed as alternative to the peak-to-peak normalization $\mathcal{N}_{PP}$.

Psycho-physical experiments are used to evaluate the performances of the map integration schemes. Regarding the normalization schemes, the evaluation concludes first to the higher suitability of the long-term normalization compared to the peak-to-peak normalization. Indeed, the former has the advantage to take into account the relative contribution of the cues, while the latter scales each cue to the same value range, regardless of the effective map amplitude.

Regarding the map transforms, both non-linear iterative $\mathcal{N}_{iter}$ and exponential $\mathcal{N}_{exp}$ perform equally well and also better than the linear $\mathcal{N}_{lin}$. While the linear map transform tends to include in the saliency map irrelevant background noise around salient regions, both non-linear map transforms have the advantage to suppress low-level values formed by the background.

From this study, we can state that the optimal map integration strategy for computing a saliency close to a collective human visual attention is the long-term normalization scheme combined with one of the non-linear map transforms $\mathcal{N}_{iter}$ or $\mathcal{N}_{exp}$, with a possible preference for the later method for its lesser computation costs. Indeed, $\mathcal{N}_{iter}$ applies an iterative process based on repetitive and time consuming DoG convolution. Therefore, for the dynamic model design, we use the static model configuration combining the non-linear exponential map transform with the long-term normalization scheme.

10.2 Motion model

From the neuroscience point of view, we have exposed two motion contrasts, to which the human vision system is sensitive: the magnitude and phase contrasts. It constitutes the core of the computer model. The former is discriminant in term of magnitude. Such contrast is defined as a difference of speed magnitude between the center and surrounding motion. The latter is discriminant in term of phase, i.e. a difference of speed direction.

Regarding the modeling issue, four motion models relying on the mentioned contrasts have been considered. (i) The motion magnitude model, which computes one scalar motion feature, highlights magnitude contrasts. The three other models highlight motion contrasts both in phase and magnitude. (ii) The motion direction model uses several scalar features sensitive to specific directions. This approach has been proposed previously in [30]. We note that the authors include four oriented motion energy maps in the model, while the proposed one includes eight scalar direction-based features. (iii) The motion vector model, which is based on vectorial convolution, highlights relative motion contrast. (iv) Finally, the phase & magnitude motion model is based on two scalar motion features which decouple phase and magnitude contrasts.

The two last models, namely the motion vector and the phase & magnitude motion models represent novel approaches as alternative to the biologically-plausible motion direction model, which has the inconvenience to be heavy in term of resources and computation costs.

Regarding contrast computation, all the models are based on difference-of-gaussian filtering (DoG). This approach, previously proposed in [4, 26] to model the center-surround differences, is a plausible way to simulate motion contrasts of the human visual system. However, an inconvenience is its heavy computational complexity. Indeed, large kernels are required to compute the center-surround contrasts based on spatial convolution. For this reason, we have presented an alternative motion pyramid approach, which approximates center-surround filtering by using motion pyramid and cross-scale difference.

An experimental frame using psycho-physical experiments is used to assess the model suitability. Four dynamic computer models are considered in the evaluation, each one combining a static contribution defined by the static model, and a motion contribution defined by one of the four motion models: (1) the magnitude $\mathcal{M}_{magn}$, (2) direction $\mathcal{M}_{dir}$ (3) vector $\mathcal{M}_{vector}$, and (4) the phase & magnitude $\mathcal{M}_{p\&m}$ models. The experiments include video sequences of different nature (synthetic and natural real scenes, acquired with fixed and moving background), showing advantages and inconveniences of the different models.

The evaluation is divided in two parts. First, regarding video sequences with fixed background, the magnitude model is suitable to highlight the magnitude contrasts, to which human attention is sensitive. In addition, this model performs similarly compared to the three other models. Therefore, in computer vision applications operating on fixed background video scene, $\mathcal{M}_{magn}$ will be preferred for computational issues.

Second, regarding video sequences with moving background, the magnitude model is not suitable in presence of pure motion phase contrast. Conversely, the vector, phase & magnitude, and direction models, are more suitable to predict the human saliency. While $\mathcal{M}_{magn}$ highlights magnitude contrasts only, the three models $\mathcal{M}_{dir}$, $\mathcal{M}_{vector}$ and $\mathcal{M}_{p\&m}$ highlight both phase and magnitude contrasts.

Globally, the quantitative evaluation shows the higher suitability of the $\mathcal{M}_{vector}$, $\mathcal{M}_{p\&m}$ and $\mathcal{M}_{dir}$ compared to the $\mathcal{M}_{magn}$. Over the three models, the evaluation indicates slight higher performances of the phase & magnitude model. This model decouples phase and magnitude contrasts, which tends to equilibrate both contributions in the computation of the motion saliency map.

For prospective computer vision applications, model suitability and computation costs have to be considered as criterion of selection. Over the four models, the direction model is clearly the most heavy in term of computation costs. Indeed, the detection of center-surround contrasts applies on each direction-based feature. In addition, for an hardware implementation, storage resources are much more important. Therefore, in computer vision applications operating on moving background video scene, the phase & magnitude or the vector models will be preferred for computation issue.

In conclusion, we have proposed two novel approaches, namely the vector and phase & magnitude models as alternative to the biologically plausible direction model. The proposed models, which perform at least as well as the direction model according to the evaluation, have the advantage to be more optimal in term of resources and computation costs.

Limitations and perspectives

The proposed computer models rely on a selection process based on static and motion features, which are computed instantaneously, by only considering data from two successive frames. In other word, such an approach does not take into account temporal influences. However, human VA is changing over the time according to the motion persistency and the presence of several moving stimuli. Therefore, including temporal influences in the computer model constitutes an attractive perspective of development.

The proposed models are suitable to highlight relative motion contrasts in the camera plane. Specifically, the models operate accurately on video sequences acquired by a camera moving in the sensor plane. In this configuration, the geometric transformations in the image are dominated by translations and region-based matching technique is an appropriate motion estimation method to detect translations in the camera plane. For a camera moving out of the sensor plane, other geometric transformations appear in the image, typically expansion and contraction. Also, it is admitted that there exists an area in the brain (MST area) that is specialized in the processing of such transformations. Therefore, including additional motion features in the model, such as expansion and contraction, constitutes a possible extension of the proposed model.

10.3 Motion integration schemes

For the purpose of integrating static and motion contributions in the dynamic model, the novel motion priority scheme is proposed as alternative to the classical competitive scheme. This approach is motivated by the existence of pop-out moving stimuli, which tend to catch entirely human visual attention, even in presence of static pop-out stimuli. In other words, the psychophysical experiments suggests that the influence of the static contribution over visual attention may be strongly inhibited in presence of salient motion contrast. Therefore, for computer modeling issues, the competitive scheme does not seem to be the most appropriate strategy to integrate static and motion contributions. In presence of salient moving stimuli, the motion priority scheme provides the priority to the motion cue by suppressing entirely the static contribution in the saliency map.

Three integration schemes are considered in the evaluation: (i) the cue competition scheme integrates motion at the cue level, by applying the classical competitive strategy; (ii) the static & motion scheme integrates motion at a higher level. Color, intensity and orientation contributions are first integrated in the static cue, which is then combined to the motion cue; (iii) the motion priority scheme provides the priority to the motion cue.

The evaluation concludes to the following performance ranking: #1 motion priority scheme, #2 static & motion competition scheme and #3 cue competition scheme. The performance variations simply illustrate the influence of the motion contribution in the resulting saliency map. The more motion contributes to the saliency map, the more the model correlates to the average human visual behavior. Therefore, the most suitable scheme is the motion priority scheme which suppresses the static contribution in presence of salient moving stimuli, while the cue competition and static & motion competition schemes are less suitable due to static contribution.

Limitations and perspectives

Statistically, the motion priority strategy turns out to be the most efficient way to predict the average human visual behavior. Humans tend to focuss their attention more frequently on salient moving stimuli and rarely on static salient stimuli. However, for application perspectives, such a strategy can lead to limitations. For example, prioritizing motion by discarding static information is not realistic for an advanced system operating with its environment. Indeed, such a system is expected to interact both with static and moving items. Therefore, an interesting perspective is the development of an adaptive attentive system operating dynamically in a competitive or priority mode, according to the motion persistency.

Appendix A

Application 1: robot navigation

Robot Navigation by Panoramic Vision and Attention Guided Features

Abstract

In visual-based robot navigation, panoramic vision emerges as a very attractive candidate for solving the localization task. Unfortunately, current systems rely on specific feature selection processes that do not cover the requirements of general purpose robots. In order to fulfill new requirements of robot versatility and robustness to environmental changes, we propose in this paper to perform the feature selection of a panoramic vision system by means of the saliency-based model of visual attention, a model known for its universality. The first part of the paper describes a localization system combining panoramic vision and visual attention. The second part presents a series of indoor localization experiments using panoramic vision and attention guided feature detection. The results show the feasibility of the approach and illustrate some of its capabilities.

1. Introduction

Vision is an interesting and attractive choice of sensory input, in the context of robot navigation. Specifically, panoramic vision is becoming very popular because it provides a wide field of view in a single image and the visual information obtained is independent of the robot orientation. Many robot navigation methods based on panoramic vision have been developed in literature. For instance, a model in [9] was designed to perform topological navigation and visual path-following. The method has been tested on a real robot equipped with an omnidirectional camera. Another model for robot navigation using panoramic vision is described in [1]. Vertex and line features are extracted from the omnidirectional image and tracked so that to determine the robot's position and orientation. In [8], the authors present an appearance-based system for topological localization. An omnidirectional camera was used. The resulting images were classified in real-time based on nearest-neighbor learning, image histogram matching and a simple voting scheme. Tapus et al. [7] have conceived a multi-modal, feature-based representation of the environment called a fingerprint of a place for localization and mapping. The multi-modal system is composed of an omnidirectional vision system and a 360 degrees laser rangefinder.

In these systems, the feature selection process is usually quite specific. In order to fulfill new requirements of versatility and robustness imposed to general purpose robot operating in wide varying environments, adaptive multi modal feature detection is required. Inspired from human vision, the saliency-based model of visual attention [3] is able to automatically select the most salient features in different environments. In [5], the authors presented a feature-based robot localization method relying on visual attention applied on conventional images and also showed its robustness. Applying the saliency-based model for feature detection provides automatic adaptation to different environments, like indoor and outdoor environments (Figure 1).

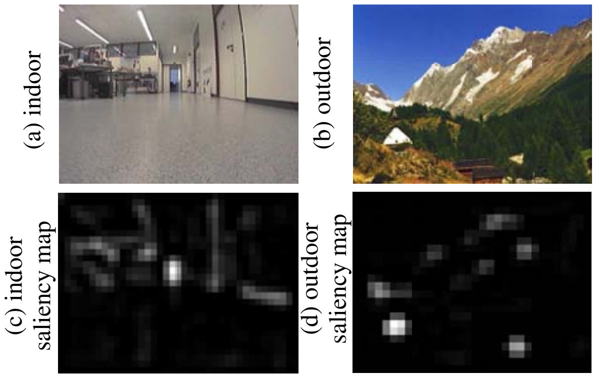


Figure 1. Adaptive behavior of the visual attention model for different environments

The purpose of this work is to get benefit of two main aspects: a) the omnidirectional vision for its independence of robot orientation and b) the visual attention-based feature extraction for its ability to cope with a wide varying environment.

The rest of the paper is structured as follows. Section 2 shows how visual attention applies to panoramic vision and how orientation independent robot localization is performed. Section 3 presents robot localization experiments and section 4 provides conclusions.

2. Visual Attention-based Navigation Using Panoramic Vision

2.1. Saliency-based Model of Visual Attention

The saliency-based model of visual attention, used for selecting the features of a scene, is composed of four main steps [3, 4], described as follows:
1) A number of cues are extracted from the scene by computing the so called feature maps F_j.
2) Each feature map F_j is transformed in its conspicuity map C_j. Each conspicuity map highlights the parts of the scene that strongly differ, according to a specific feature, from their surrounding.
3) The conspicuity maps are integrated together, in a competitive way, into a *saliency map* $\mathcal{S}$ in accordance with:

$$\mathcal{S} = \sum_{j=1}^{J} \mathcal{N}(\mathcal{C}_j) \tag{1}$$

where $\mathcal{N}()$ is the weighting operator responsible for map promotion [3].
4) The features are derived from the peaks of the saliency map (Figure 1 c and d).

2.2. Visual Feature Detection in Panoramic Images

The saliency computation must be tuned to the specificities of panoramic images. As the features should also be detected in the full range of $360°$, saliency computation algorithm must be adapted to the circularity of the input image. The circularity of the panoramic images allows to define the neighborhood on the borders, so that features on the image borders are also detected. Thus, the feature detection is obtained in the full panoramic range (Figure 2 b and c). In this paper, the saliency map is based on four different cues: image intensity, two opponent color components red/green and yellow/blue, and a corner-based cue according to the Harris approach [2].

Feature Characterization and Landmark Selection

Once detected, each feature O_n is characterized by its spatial position in the image $\mathbf{x}_{O_n} = (x_{O_n}, y_{O_n})$ and a visual descriptor vector $\mathbf{f}_{O_n}$, in which each component f_j holds the value of a cue at that location:

$$\mathbf{f}_{O_n} = (f_1, ..., f_j, ..., f_J)^T \quad with \quad f_j = F_j(\mathbf{x}_{O_n}) \tag{2}$$

In order to take into account the spatial information of the features, an appropriate spatial representation is used: each set of features is represented on an horizontal one-dimensional space, by projection (Figure 2d).

Finally, an observation catched by a panoramic image is described by the set of features S_t (Figure 2c):

$$S_t = \{O_n\} \;\; with \;\; O_n = (\mathbf{x}_{O_n}, n_{x_{O_n}}, \mathbf{f}_{O_n}) \tag{3}$$

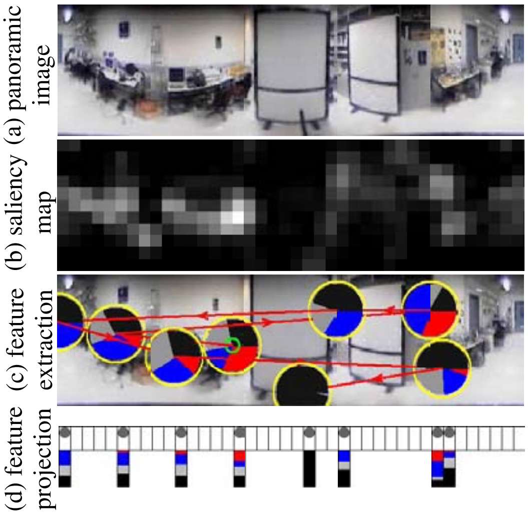


Figure 2. From the panoramic image to the horizontal feature projection

where $n_{x_{O_n}}$ is the index corresponding to the rank of the features spatially ordered in the x direction.

2.3. Map Building

Basically, the features detected during a learning phase are used as landmarks for localization during the navigation phase. In this work, a topological approach is used. The path is divided into equidistant portions E_q, each described by a configuration of landmarks named key-frame K_q.

Intrinsically, saliency provides a powerful adaptation to the robot environment. To provide a further adaptation, detected features are then chosen accordingly to their robustness. The step consists in tracking features along the environment [6] and to select as landmarks, the most persistent features, i.e. the ones with the longest tracking paths. A landmark is thus the representation of a robust feature that is persistent along the same portion E_q.

A key-frame K_q is a set of robust features named landmarks L_m, where each landmark is defined by the mean characteristics of the considered feature along the same portion: its mean spatial position in the image $\bar{\mathbf{x}}_{L_m} = (\bar{x}_{L_m}, \bar{y}_{L_m})$, its index $n_{x_{L_m}}$, its mean descriptor vector $\bar{\mathbf{f}}_{L_m}$ and its standard deviation vector $\mathbf{f}_{\sigma_{L_m}}$:

$$K_q = \{L_m\} \;\; with \;\; L_m = (\bar{\mathbf{x}}_{L_m}, n_{x_{L_m}}, \bar{\mathbf{f}}_{L_m}, \mathbf{f}_{\sigma_{L_m}}) \tag{4}$$

2.4. Navigation Phase

As soon as the navigation map is available, the robot is able to localize itself by determining which key-frame K_q

matches the best the robot's observation S_t at its current location.

A. Localization by Key-frame

The purpose is to match a set S_t of visual features with a set K_q of landmarks. Our matching method takes into account two criteria: visual and spatial similarity.

The visual landmark similarity: A landmark L_m and a feature O_n are said similar in terms of visual characterization if their Mahalanobis distance is inferior to a given threshold α:

$$\Delta\mathbf{f} = (\frac{f_{1_{L_m}} - f_{1_{O_n}}}{f_{1_{\sigma_{L_m}}}}, ..., \frac{f_{J_{L_m}} - f_{J_{O_n}}}{f_{J_{\sigma_{L_m}}}})^T \; and \; \|\Delta\mathbf{f}\| < \alpha \tag{5}$$

where $f_{J_{L_m}}$, $f_{J_{O_n}}$ and $f_{J_{\sigma_{L_m}}}$ are the J components of respectively $\mathbf{f}_{L_m}$, $\mathbf{f}_{O_n}$ and $\mathbf{f}_{\sigma_{L_m}}$.

The spatial similarity of landmark triplet: In this work, a comparison "feature group to landmark group" is used and the spatial similarity is measured by comparing the relative distances between each element of the group. Such a group matching strategy has the advantage to take into account the spatial relationships of each element of the group, which improves the matching quality. In this work, the groups contain three elements (triplet).

Formally, let $o = \{O_1, O_2, O_3\}$ be a set of three features compared with a set of three landmarks $l = \{L_1, L_2, L_3\}$. A triplet *o* is spatially similar to a triplet *l* if:

- the pairings (O_1;L_1), (O_2;L_2) and (O_3;L_3) satisfy Eq.5.

- both sets are ordered with respect to their index $n_{x_{L_m}}$, $n_{x_{O_n}}$ under the principle of circularity.

- the absolute difference distances δ_{12} and δ_{23} are inferior to a threshold T_d:

$$\delta_{12}, \delta_{23} < T_d \tag{6}$$

where

$$\delta_{12} = |\ (x_{O_2} - x_{O_1}) - (\overline{x}_{L_2} - \overline{x}_{L_1})\ | \tag{7}$$

$$\delta_{23} = |\ (x_{O_3} - x_{O_2}) - (\overline{x}_{L_3} - \overline{x}_{L_2})\ | \tag{8}$$

Given two spatial similar triplets, a function s_{c_i} not further defined here quantifies the overall similarity:

$$s_{c_i}(\Delta\mathbf{f}_1, \Delta\mathbf{f}_2, \Delta\mathbf{f}_3, \delta_{12}, \delta_{23}) \tag{9}$$

where $\Delta\mathbf{f}_i$ holds for the visual similarity of the pairing (O_i, L_i) and δ_{12}, δ_{23} for the spatial similarity.

Observation likelihood: Let n_{K_q} be the number of observation triplets that satisfy the landmark triplet similarity for the key-frame K_q. In order to define which key-frame K_q matches the best the observation, $S_{C_{(K_q)}}$ is computed as the sum of the similarity contribution of the n_{K_q} triplets:

$$S_{C_{(K_q)}} = \sum_{i}^{n_{K_q}} s_{c_i} \tag{10}$$

Thus, each key-frame receives several contributions, depending on the observation triplets that match the landmarks triplets. The measurement is then normalized in order to represent a probability distribution, called visual observation likelihood and formalized as $P(S_t|K_q)$:

$$P(S_t|K_q) = \frac{S_{C_{(K_q)}}}{\sum_n S_{C_{(K_n)}}} \tag{11}$$

$P(S_t|K_q)$ quantifies the likelihood of the observation S_t at time t given the associated key-frame K_q. Thus, simple localization is performed according to the maximum likelihood criterion:

$$q^* = arg\ max_q P(S_t|K_q) \tag{12}$$

B. Contextual Localization

To improve the robustness of the localization, the contextual information of the environment is taken into account. Thus, the visual observation likelihood $P(S_t|K_q)$ is integrated into a Markov localization framework. In this work, the states of the Markov model correspond to the portions E_q represented by its key-frame K_q and the state transition model is defined by $P(K_i, K_j)$, corresponding to the probability of the state transition from E_j to E_i.

Let $P(K_t)$ be the probabilistic estimation of its location at time t. $P(K_t)$ is computed in Eq.13 by fusing the prediction $P_{pred}(K_t = K_i)$ with the visual observation likelihood $P(S_t|K_q)$:

$$P(K_t = K_i) = \frac{1}{\alpha_t} P(S_t|K_i) \cdot P_{pred}(K_t = K_i) \tag{13}$$

$$P_{pred}(K_t = K_i) = \frac{1}{\beta_t} \sum_{K_j \in K^\star} P(K_i, K_j) \cdot P(K_{t-1} = K_j) \tag{14}$$

Note that α_t and β_t are normalization factors used to keep $P(K_t)$ a probability distribution.

3. Experiments

In the experiments, the robot acquires a sequence of panoramic images obtained from an equiangular omnidirectional camera, while moving along a path in a lab environment (Figure 2). The path of about 10 meters long gives rise to a sequence of 64 panoramic images. From this sequence, the navigation map is built in three different configurations: (A) the map segmenting the path in 8 equidistant portions, (B) in 10 portions and (C) in 13 portions.

To quantify the localization, an approximate success rate *R* is defined. *R* corresponds to the percentage of approximate correct localization, which is considered as correct if the location with the maximum likelihood q^* corresponds to $q_e \pm 1$, where q_e represents the exact location.

During the localization experiment, the visual observation S_t of each frame of the navigation sequence is computed and compared with the key-frames of the map.

The value $R_{\overline{c}}$ measures the success rate of the simple context-free localization. The value R_c holds for the contextual localization with the Markov framework, where the initial estimation $P(K_{t=0})$ is set to 80% at the exact location and the other are uniformly distributed at the other locations. The state transitions $P(K_i, K_j)$ are modelled by a Gaussian distribution, i.e. transition to the neighboring portions is more likely than transition to distant portions.

The first experiment (Exp.1) tends to evaluate the quality of the visual landmarks. It uses the same sequence for map building and navigation.

The second experiment (Exp.2) verifies the orientation independence of the proposed process. It uses three test sequences corresponding to rotated views of the original sequence by $90°$, $180°$ and $270°$ respectively to be matched with the original map.

Exp.1	8 KF Map	11 KF Map	13 KF Map	mean
$R_{\overline{c}}$	87.5%	82.8%	79.7%	83.3%
R_c	98.4%	96.9%	96.9%	97.4%

Exp.2	8 KF Map	11 KF Map	13 KF Map	mean
$R_{\overline{c}}$	80.2%	80.2%	78.1%	79.5%
R_c	94.8%	98.4%	98.9%	97.4%

Table 1. Localization Results

The results are presented in Table 1. For simple key-frame localization, the success rate $R_{\overline{c}}$ decreases as expected when the number of portions increases and experiment 1 provides an average rate of 83%. Contextual localization improves the performance further and provides an average rate R_c of 97%. Given the fact that the sequence of panoramic images provides only small changes, with key-frames representing small portions of about one meter length, the localization performance is considered as quite good.

In Exp.2, the results are similar to Exp.1 and show the orientation independence of the localization method.

These results confirm the feasibility of the proposed approach and show the capacity of the system to catch robust discriminant features.

The next step will be to evaluate the robustness of the proposed method in the presence of condition changes (luminosity, different robot navigation trajectories).

4. Conclusions

An original robot localization system was presented, that encompasses panoramic vision and attention guided feature detection. First, the multi-cue saliency-based model of visual attention was adapted to panoramic image sequences; a description for a feature set, as well as a suited feature set matching method were also proposed. Then, localization experiments were conducted using two simple methods. In a sequence of panoramic images showing only small changes, the rate of successful localization is typically 83% and 97% with the context-free and contextual methods respectively. Another experiment shows the orientation independence of the proposed processing. These results confirm the feasibility of the proposed approach and show the capacity of the system to catch robust discriminant features.

References

[1] M. Fiala and A. Basu. Robot navigation using panoramic landmark tracking. *Society of Manufacturing Engineers (SME). Article RPOS-100. USA, NRC 47136*, 2003.

[2] C. Harris and M. Stephens. A combined corner and edge detector. *Fourth Alvey Vision Conference, pp. 147-151*, 1988.

[3] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 20, No. 11, pp. 1254-1259*, 1998.

[4] C. Koch and S. Ullman. Shifts in selective visual attention: Towards the underlying neural circuitry. *Human Neurobiology, Vol. 4, pp. 219-227*, 1985.

[5] N. Ouerhani, A. Bur, and H. Hugli. Visual attention-based robot self-localization. *ECMR 2005, in Proc. of European Conference on Mobile Robotics, Italy, pp. 8-13*, 2005.

[6] N. Ouerhani and H. Hugli. A visual attention-based approach for automatic landmark selection and recognition. *WAPCV 04, in Lecture Notes in Computer Science, Springer Verlag, LNCS 3368, pp. 183-195*, 2005.

[7] A. Tapus and R. Siegwart. Incremental robot mapping with fingerprints of places. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Canada*, 2005.

[8] I. Ulrich and I. R. Nourbakhsh. Appearance-based place recognition for topological localization. *IEEE International Conference on Robotics and Automation (ICRA), USA*, 2000.

[9] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor. Omnidirectional vision for robot navigation. *Proceedings of IEEE Workshop on Omnidirectional Vision (Omnivis00)*, 2000.

Appendix B

Application 2: omnidirectional visual attention

Visual Attention on the Sphere

Iva Bogdanova, Alexandre Bur and Heinz Hügli

Abstract

Human visual system makes an extensive use of visual attention in order to select the most relevant informations and speed-up the vision process. Inspired by visual attention, several computer models have been developped and many computer vision applications rely today on such models. But the actual algorithms are not suitable to omnidirectional images, which contain a significant amount of geometrical distorsion. In this paper, we present a novel computational approach that performs in spherical geometry and thus is suitable for omnidirectional images. Following one of the actual models of visual attention, the spherical saliency map is obtained by fusing together intensity, chromatic and orientation spherical cue conspicuity maps that are themselves obtained through multi-scale analysis on the sphere. Finally, the consecutive maxima in the spherical saliency map represent the spots of attention on the sphere. In the experimental part, the proposed method is then compared to the standard one using a synthetic image. Also, we provide examples of spots detection in real omnidirectional scenes which show its advantages. Finally, an experiment illustrates the homogeneity of the detected visual attention in omnidirectional images.

I. Introduction

A. State of the Art on Visual Attention

It is generally admitted today that the human visual system makes extensive use of visual attention (VA) in order to select relevant visual information and speed up the vision process. Visual attention represents also a fundamental mechanism for computer vision where similar speed up of the processing can be envisaged. Thus the paradigm of computational VA has been widely investigated over the past two decades and possible fields of application include image and video compression [1], [2], object recognition [3], [4], image segmentation [5] and robot localization [6], [7].

The authors are with the Institute of Microtechnology, University of Neuchâtel, Rue A.-L. Breguet 2, CH-2000 Neuchâtel, Switzerland. E-mail: {iva.bogdanova, alexandre.bur, heinz.hugli}@unine.ch

While some computational VA models concentrate on psychophysical and biologically plausible aspects [8], [9], [10], other models focus more on efficient computation for related computer vision applications. Proposed in [11], the first computational architecture of VA includes the main concepts such as feature integration, saliency map, winner-take-all network (WTA), center-surround difference and inhibition of return (IOR). Several models are based on these concepts. In [12], the authors develop one of the first implementation including center-surround difference based on classical filtering and a relaxation process is used for the map integration, resulting to high computational cost. One of the most actual and used model is presented in [13] and [14]. An efficient approximation of center-surround difference is performed with gaussian image pyramid and the relaxation process is replaced by a weighting scheme for the map integration, resulting to faster computation.

There are several VA models that use classical filtering approach [14], [12], [4], [1], [15], [10], and other models that are based on neural network [16], [17]. The mentionned computer models are mainly bottom-up, i.e. attention driven by a reflexive behavior, due to strong feature-related contrasts. Other approaches include top-down informations, in which attention is driven by prior knowledges, expectations or tasks. In [18] and [19], the saliency map results from the fusion of bottom-up and top-down cues.

B. Omnidirectional Vision: the Sphere of View

While conventional imaging systems (like photographic or video) are severely limited in their field of view, omnidirectional imaging systems were developed so that they are capable of viewing the world in all directions from a center of projection, i.e. the entire sphere of view around a single point. An ideal omnidirectional image thus provides a full spherical field of view (4π steradian) of a given scene. The scene, as viewed by an omnidirectional imaging system, can be represented by the ideal plenoptic function which completely describes the light field [20]. In real omnidirectional systems, the field of view is somehow restricted: typical omnidirectional images will thus exhibit a full azimuthal (360^0) range but a restricted zenithal ($< 180^0$) range. Such images will be considered here.

Various kinds of omnidirectional imaging systems exist. For instance, a *rotating camera* at 360^0 can produce a sequence of images that covers the whole scene. However, this kind of imaging system cannot simultaneously cover actions in all directions of a dynamic scene. To cope with this, either a *multi-camera* or a *catadioptric sensor* can be used instead. The first one simultaneously captures images that cover about 75% of the visible sphere. Actually, in this case, all individual images (from each of the cameras in the set) are stitched together in order to form a spherical image. The second one is an imaging system based on combination of a curved mirror and a lens to form a projection onto the image plane of a (video) camera [21], [22]. Such an omnidirectional imaging system offers the potential for simultaneously capturing an image with a high resolution on a target as well as a wide field-of-view periphery. Of interest are catadioptric imaging systems with a single effective viewpoint called central catadioptric sensors, as the hyperbolic or the parabolic ones. It is important to note that there is an equivalence between the catadioptric projection and two-steps mapping onto the sphere [23].

It is clear that the omnidirectional images can be defined as non-Euclidean, i.e. spherical, hyperbolic, parabolic. The mapping between points in the 3-D world and points in the image of a given omnidirectional imaging system is non-linear. In the catadioptric systems, the curved mirror produces inevitable radial distortions, proportional to the radial curvature of the mirror. In fact, these mirrors, produce a ”fish-eye” effects, i.e. they magnify the objects reflected in the center (typically the camera used in the given system) which is of minimal interest, while in the same time they shrink the region around the horizon, thereby reducing the available spatial resolution in the area which is of interest. It is of great importance to note, that the linear calculation methods cannot appropriately cope with these non-linear deformations on the projected image (i.e. the resulting omnidirectional image).

C. A Word of Motivation: Why Visual Attention on the Sphere?

The current VA algorithms are performing in Euclidean geometry and thus are limited to Euclidean images. Therefore, the so known visual attention implementations suits only the conventional images. In the same time, the omnidirectional sensors are more and more used nowadays because of their advantages over the conventional imaging sensors, namely their larger field of view. One particular example of application is in surveillance [24], [25], where one would be able to track persons in heavily cluttered environments. They are also required in robotics, where an autonomous robot may benefit from omnidirectional vision for robot navigation and situational awareness [26], [27].

The intuitive approach for defining a VA on omnidirectional image is to first map the image to a panoramic image (i.e. unwrap it) and then to apply the conventional VA algorithm. In fact, it was proposed in [6] an adaptation to omnidirectional (panoramic) images. The proposed representation basically uses a cylindrical source image and this algorithm solves the circularity problem by introducing a modulo operation on the i-indexes in order to handle the cyclic nature of that coordinate. This approach however is bound by the cylindrical projection (which still contains distortions) and therefore is limited to panoramic images with a specific and limited zenithal extension.

The images obtained by omnidirectional sensors suffer under significant deformations. Specific mappings, like panoramic or log-polar mappings, attempt to reduce somehow the distortions but do not succeed completely. In fact, no Euclidean mapping represents omnidirectional views homogeneously. A natural choice of a non-deformed domain for the full sphere of view, where there are no limitations on the zenithal range, is the sphere $S^2 \in \mathbb{R}^3$. Therefore, we argue that detection of spots of attention in omnidirectional images has to take place in spherical geometry. Developing a VA algorithm for an omnidirectional image, that provide the full sphere of view of a scene, is equivalent to detecting the spots of attention homogeneously on the sphere, i.e. in all directions.

The possibility to compute VA on the sphere opens new perspectives for large-field-of-view imaging applications, where the Euclidean geometry does not hold anymore. Computing VA in spherical coordinates provides a homogeneous behavior, i.e. invariant to its location and orientation on the sphere and which is therefore applicable to any omnidirectional image that can be mapped on the sphere.

In this paper, we present a derivation of a new algorithm for computing the VA of images obtained by omnidirectional imaging systems. Inspired by a classical visual attention model applied on Euclidean images [13], the proposed model operates in spherical geometry and is thus applicable for any omnidirectional image that can be mapped on the sphere. Such are not only images obtained with multi-camera sensors but as well those obtained with hyperbolic or parabolic catadioptric imaging sensors. By computing in spherical coordinates, the attention mechanism remedies the distortions introduced by the computations as performed in the Euclidean case. It must be noted that we have chosen to use a pyramidal architecture in this approach but other VA computational models can be considered. For instance, a filter-based VA model [19] can be easily defined on the sphere.

This paper is organized as follows. Section II presents a classical model of visual attention and its algorithm. Then the paper proceeds with pointing out data processing on the sphere and more particularly, with presenting in Section III the multiscale analysis of spherical data. Section IV then presents the VA algorithm on the sphere which is then applied to omnidirectional images, first to a synthetic spherical image in Section V and to a real omnidirectional image from a multi-camera system in Section VI and VII. In Section VIII we study the limitations of both spherical and Euclidean VA while in Section IX we test the homogeneity of the spherical VA.

II. Visual Attention: the Euclidean Case

This section presents the classical model of VA and the related equations which hold for its computation in the 2D Euclidean space.

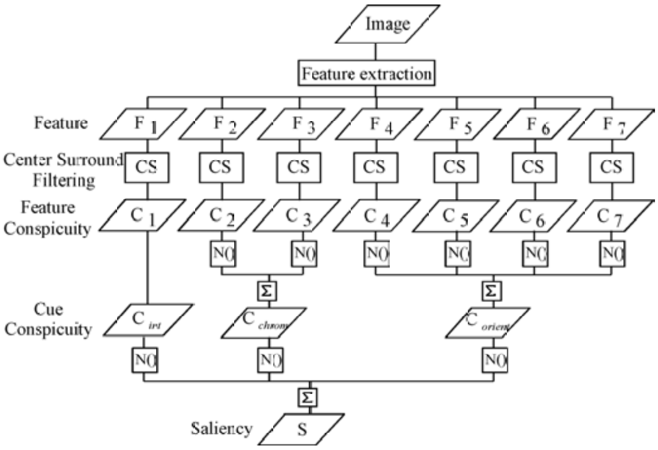


Fig. 1. The saliency-based model of VA.

A. The Saliency-Based Model of VA

The saliency-based model of VA, originally proposed by Koch and Ullman in [11] is widely used nowadays. Several works have dealt with the realization of this model e.g. [13]. It is based on three major principles: VA acts on a multi-featured input; saliency of locations is influenced by the surrounding context; the saliency of locations is represented on a scalar saliency map.

The different steps are detailed in the model illustrated in Figure 1 which, for a simpler notation, has a specific number of features (7) and cues (3) although any number can be considered in general.

First, seven features $(1..j..7)$ are extracted from the scene by computing the so-called feature maps from an RGB color image. The features are:

- intensity feature F_1;
- two chromatic features based on the two color opponency filters: red-green F_2 and blue-yellow F_3;
- four local orientation features $F_{4..7}$.

In a second step, each feature map is transformed into its conspicuity map C_j, which highlights the parts of the scene that strongly differ, according to a specific feature, from their surrounding. This process, relies on a multiscale center-surround filtering which is later described in Section II-C.

In the third step, the seven $(j = 1...7)$ features are then grouped using a competitive map integration scheme and according to their nature into the three cues: intensity, chromaticity and orientation. The ***cue***

conspicuity maps are thus:

$$\begin{aligned} C_{int} &= C_1; & (1)\\ C_{chrom} &= \sum_{j\epsilon\{2,3\}} \mathcal{N}(C_j); & (2)\\ C_{orient} &= \sum_{j\epsilon\{4,5,6,7\}} \mathcal{N}(C_j), & (3) \end{aligned}$$

where C_{int} is the intensity conspicuity map, C_{chrom} is the chromaticity conspicuity map and C_{orient} is the orientation conspicuity map. $\mathcal{N}(.)$ refers to a normalization function defined below in Section II-B.

In the final step of the attention model, the cue conspicuity maps are integrated, into a ***saliency map*** S, defined as:

$$S = \sum_{cue\epsilon\{int,chrom,orient\}} \mathcal{N}(C_{cue}). \tag{4}$$

Given a saliency map, the "winner-take-all" (WTA) mechanism starts with selecting the location with the maximum value of the map. This selected region is considered as the most salient part of the image (winner). The ***spot of attention*** is then shifted to this location. Local inhibition is then applied on the saliency map, in an area around the actual spot. This yields dynamical shifts of the spot of attention by allowing the next most salient location to subsequently become a winner. Besides, the inhibition mechanism prevents the spots of attention from returning to previously attended locations.

B. Normalization $\mathcal{N}()$ for Map Fusion

The normalization function adjusts the range of maps of different nature and simulates the competition between the different maps to be integrated. Several methods were proposed, which are reviewed in [28]. Although any method would be applicable here, for simplicity, a linear weighting scheme is considered in this paper. Given the conspicuity map C(**x**), it defines $\mathcal{N}(C(\mathbf{x}))$ as:

$$\mathcal{N}(C(\mathbf{x})) = w \cdot C(\mathbf{x}) \quad \text{with} \quad w = \frac{Max(C(\mathbf{x}))}{Mean(C(\mathbf{x}))}. \tag{5}$$

C. Multiscale Center-Surround Filtering

Each feature map F_j is transformed independently into a feature conspicuity map C_j by a biologically inspired center-surround mechanism that tends to highlight the parts of the feature map that strongly differ from their surrounding. In order to highlight regions of different sizes, the mechanism applied at various map scales and thus the transform consists in a so called multiscale center-surround filtering, illustrated in Figure 2 and described as follows.

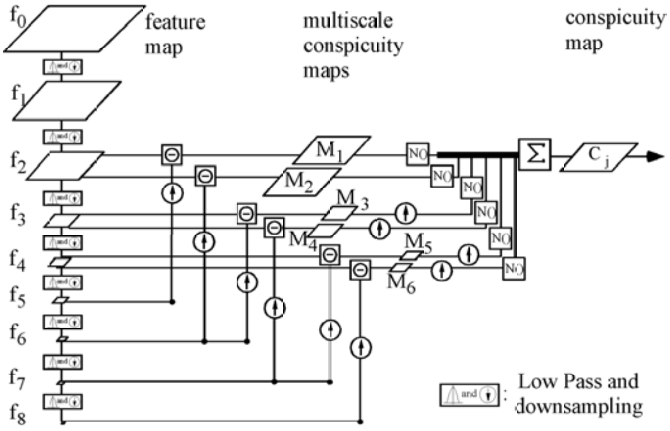


Fig. 2. Multiscale center-surround mechanism with an $n = 8$ level pyramid and six scale maps

First, for a feature F_j, a gaussian pyramid is created by successively lowpass filtering the signal with a gaussian filter g and down-sampling the result by factor 2. Formally, the successive images f_k at the pyramid output are:

$$f_0 = F_j, \quad f_1 = \downarrow 2(f_0 * g), \;\; ... \;\; , f_k = \downarrow 2(f_{k-1} * g).$$

Then, the effective center-surround mechanism, which necessitates subtracting a surround region from a center region, is implemented by simply subtracting pairwise output images from the gaussian pyramid. According to the VA model that forsees a ratio of 8 and 16 in the relative sizes of center and surround, the scale difference of the images to be subtracted is thus 3 and 4. Accordingly, from the $f_k, k = 1..n$ maps a number of multiscale maps $\mathcal{M}_k$ are thus computed as follows:

$$\begin{aligned} \mathcal{M}_1 &= |f_2 \ominus f_5|, \mathcal{M}_2 = |f_2 \ominus f_6|, \mathcal{M}_3 = |f_3 \ominus f_6|, \\ \mathcal{M}_4 &= |f_3 \ominus f_7|, \mathcal{M}_5 = |f_4 \ominus f_7|, \mathcal{M}_6 = |f_4 \ominus f_8|, \end{aligned} \tag{6}$$

where $\ominus$ refers to a cross-scale difference operator that interpolates the coarser scale to the finer one and then performs a point-by-point substraction.

Finally, the ***feature conspicuity map*** C_j is computed by combining in a competitive way the set of multi-scale maps $\mathcal{M}_k$ using the normalization function defined in Equation (5):

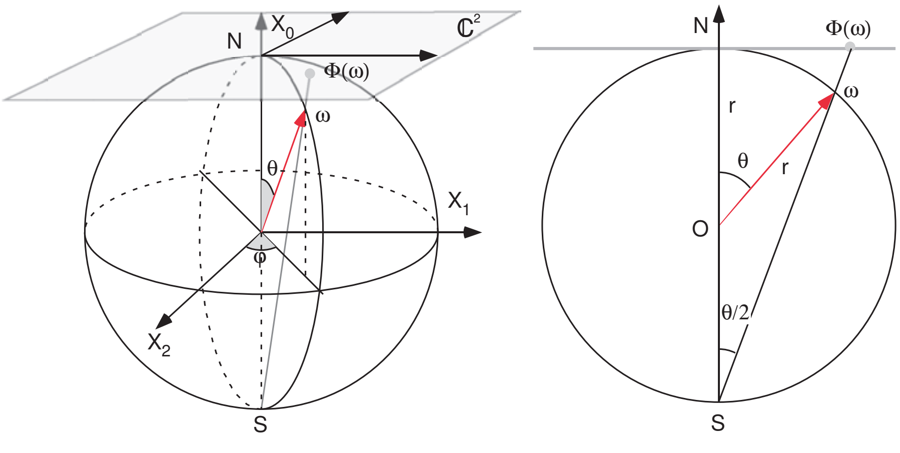
$$C_j = \sum_{k=1}^{n-3} \mathcal{N}(\mathcal{M}_k). \tag{7}$$

III. Data Processing on the Sphere

This section presents data processing methods in spherical geometry and proposes a means for computing a conspicuity map of a single feature in the spherical domain.

A. Spherical Geometry

The 2-sphere ($S^2 \in \mathbb{R}^3$) is a compact manifold of constant positive curvature. In polar coordinates, each point on the sphere is a three-dimensional vector $\omega = (x_0, x_1, x_2) \equiv (r\cos\theta, r\sin\theta\sin\varphi, r\sin\theta\cos\varphi)$, with $r \in (0,\infty), \theta \in [0,\pi]$ and $\varphi \in (0, 2\pi]$ as illustated in Figure 3(a). Figure 3(b) also illustrates the so called stereographic projection from the South Pole, a projection that maps any point of the sphere onto a point of the tangent plane at the North Pole. If we take the sphere S^2 as the Riemannian sphere $(r = 1)$ and the tangent plane as the complex plane $\mathbb{C}^2$, then the stereographic projection is a bijection given by



(a) (b)

Fig. 3. Spherical geometry: (a) spherical coordinates, (b) cross-section of a stereographic projection through the South Pole.

$$\Phi(\omega) = 2\tan\frac{\theta}{2}(\cos\varphi, \sin\varphi), \tag{8}$$

where $\omega \equiv (\theta, \varphi),\ \theta \in [0,\pi], \varphi \in [0, 2\pi)$.

B. Multiscale Analysis on the Sphere S^2

The central step towards defining VA onto the sphere concerns the multiscale center-surround filtering. As shown in Section II-C, this step requires to set up a Gaussian pyramid and to compute differences between two maps of the pyramid which are each time three or four scales apart. This contrasts the implementation of another spherical pyramid, namely the spherical Laplacian Pyramid, worked out in [29]. As both problems are very similar, the procedure for VA is widely inspired from it. In fact, a Laplacian pyramid also relies on differences of maps from a Gaussian pyramid. It differs however in the scale difference of the two maps to be subtracted. In the following we present in details the filtering on the sphere that is needed for building the Spherical Gaussian Pyramid.

C. Filtering on the Sphere

In general, the convolution on the (Euclidean) plane is defined in terms of the inner product between two functions translated relative to each other, and is parameterized by the amount of translation. On the sphere, it is more natural to use relative rotations. For a given spherical signal f and a filter g, their correlation $(g \star f)_{\alpha_2,\alpha_1,\alpha_0} \in L^2(SO(3))$ reads

$$(g \star f)_{\alpha_2,\alpha_1,\alpha_0} = \int_{S^2} \left[R_{\alpha_2,\alpha_1,\alpha_0} g\right](\theta,\varphi) f(\theta,\varphi) d\cos\theta d\varphi, \tag{9}$$

where $R_{\alpha_2,\alpha_1,\alpha_0}$ is the rotation operator that first rotates the function by α_0 about the x_0-axis, then by α_1 about the x_1-axis and finally by α_2 about x_0-axis again. These are the three (Euler) angles which define an element of $SO(3)$-group and they provide a natural parameterization of the correlation on the sphere. Actually, the correlation $(g \star f)_{\alpha_2,\alpha_1,\alpha_0}$ is the inner product of the rotated version of the filter g with the signal f, or the projection coefficient of f onto $[R_{\alpha_2,\alpha_1,\alpha_0}]$. If the filter is an axisymmetric function, i.e. $g(\theta,\varphi) = g(\theta)$, the rotation by α_0 about the x_0-axis has no effect. In other words this reads

$$(g \star f)_{\alpha_2,\alpha_1,\alpha_0} = (g \star f)_{\alpha_2,\alpha_1},$$

and is a spherical signal parameterized by $\theta \equiv \alpha_1$ and $\varphi \equiv \alpha_2$.

It is obvious that in practice we need the discrete form of the spherical correlation. For an axisymmetric filter g we can discretize Equation (9) and thus obtain

$$(g \star f)_{\alpha_2,\alpha_1} = \sum_{\theta=0}^{\pi} \sum_{\varphi=0}^{2\pi} \left[R_{\alpha_2,\alpha_1} g(\theta,\varphi)\right] f(\theta,\varphi).$$

First, we must note that the $(\theta,\varphi)-$grid is not invariant under rotation, therefore it is not usually possible to evaluate $[R_{\alpha_2,\alpha_1} g(\theta,\varphi)]$ from the samples of $g(\theta,\varphi)$ on the grid. That is why this kind of discrete implementation of correlation on the sphere is not efficient.

Let us have a spherical signal defined at scale k on a $2\beta_k \times 2\beta_k$, $(\beta_k \in \mathbb{N})$ square grid of respectively equi-angular resolution in θ and φ:

$$\omega \in \mathcal{G}_k := \left\{ \omega_{kpq} = (\theta_{kp}, \varphi_{kq}) : \theta_{kp} = \frac{(2p+1)\pi}{4\beta_k},\ \varphi_{kq} = \frac{q\pi}{\beta_k},\ p,q \in \mathbb{Z}\left[2\beta_k\right] \right\}. \tag{10}$$

This grid allows us to perfectly sample any band-limited function $f \in L^2(S^2)$ of bandwidth β_k, i.e. such that the Fourier coefficients $\hat{f}(l, m) = 0, \forall l > \beta_k$. Moreover, this class of sampling grids is associated to a Fast Spherical Transform [30].

Another method to perform spherical convolution is to first project the discretized spherical function and filter onto the span of spherical harmonics and perform the convolution in Fourier domain via simple multiplication. Of great importance is the Spherical Convolution Theorem, as derived in [31] and which we remind here for convenience: for functions $f, g \in L^2(S^2)$, the transform of the convolution is a point-wise product of the transforms:

$$\widehat{g \star f}(l, m) = \sqrt{\frac{4\pi}{2l+1}}\hat{g}(l, 0)\hat{f}(l, m), \tag{11}$$

where $\hat{f}(l, m)$ denotes the (l, m)-Fourier coefficient, which in discrete form reads

$$\hat{f}(l, m) = \frac{\sqrt{2\pi}}{2\beta_k} \sum_{p=0}^{2\beta_k - 1} \sum_{q=0}^{2\beta_k - 1} \alpha_{kp}^{\beta_k} \; f(\theta_{kp}, \varphi_{kq}) \; e^{-im\varphi_{kq}} \; P_l^m(\cos\theta_{kp}), \tag{12}$$

with P_l^m -the associated Legendre function of degree l and order m and $\alpha_{kp}^{\beta_k}$ is a weight.

It must be noted that the convolution theorem is independent of sampling. Hence, as long as we can project our samples onto the span of spherical harmonics accurately, we can perform convolution via Fourier domain accurately, regardless of the sampling grid. These methods are implemented in *SpharmonicKit*[1] and used together with MATLAB YAWtb toolbox[33].

D. Spherical Gaussian Pyramid

Regarding the VA mechanism as defined in Section II-A, we first need to define a Gaussian pyramid on the sphere. The filter used for smoothing the spherical data is a spherical axisymmetric Gaussian filter, defined by its Fourier coefficients:

$$\hat{g}_{\sigma_k}(l) = e^{-(\sigma_k l)^2}. \tag{13}$$

The parameter σ_k is chosen so that the filter is numerically close to a perfect half-band filter $|\hat{g}_{\sigma_k}(l)| << 1, \forall l > \beta_k$.

Let us look at a spherical signal defined on a grid of size 1024×1024 and which represents the scale level $k = 0$. The Fourier transform of this signal results in (512×512)-matrix, and thus we have $\beta_0 = 512$ and $l \in [0, \beta_0]$. Then, for designing the Gaussian filter at this level, we need to define the bandwidth parameter. The choice $\sigma_0 = \frac{\sqrt{2}}{\beta_0} = 0.0028$ satisfies our requirement $|\hat{g}_{\sigma_0}(l)| \approx 0, \forall l > 512$.

[1] under GPL license (General Public License)[32]

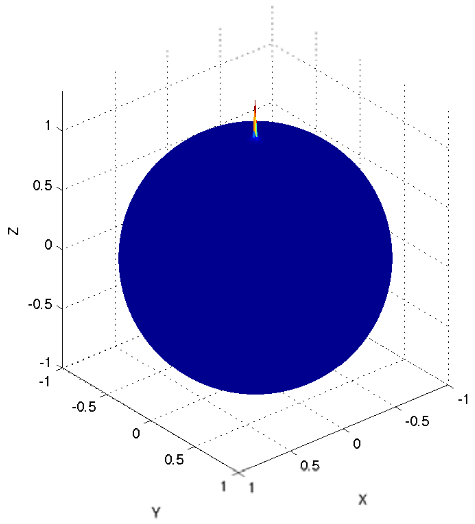


Fig. 4. A spherical Gaussian filter in spatial domain.

For purpose of visualization and for this particular value of σ_0 we can easily find the corresponding filter in spatial domain which is depicted on Figure 4.

Finally, we can define the procedure for obtaining one level in the *spherical Gaussian pyramid* that transforms f_k into f_{k+1} as follows:

1) Apply the spherical Fourier transform Equation (12) on the signal f_k and thus obtain $\hat{f}_k(l,m)$;
2) Multiply in Fourier domain with a Gaussian filter as defined in Equation (11) and thus obtain $g \cdot \widehat{f_k(l,m)}$;
3) Apply the inverse spherical Fourier transform on it and thus obtain the filtered signal $g \star f_k$;
4) Subsample this signal $g \star f_k$ by a factor of 2 and thus obtain f_{k+1}.

Subsampling on the sphere consists of reducing by a factor of 2 the spherical grid $\mathcal{G}_k$.

The full spherical Gaussian pyramid is now obtained by iteratively applying previous procedure to each scale level $k = 1 \cdots n$, thus producing the pyramid of spherical signals $f_1, f_2, f_3, \cdots f_n$.

A schematic representation of the spherical Gaussian pyramid is illustrated in Figure 5 for the case of an initial spherical signal defined on a grid of size 1024×1024. Note that the grid size is reduced by a factor of 2 at each level and that a grid size of 4×4 characterizes the last 8th level. This repeated size reduction also clearly speaks in favor of an initial grid size which is sufficiently large and expressed as a power of 2: $2\beta_0 \times 2\beta_0 = 2^{n+2} \times 2^{n+2}$.

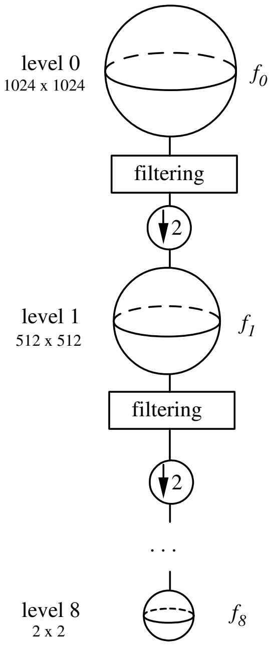


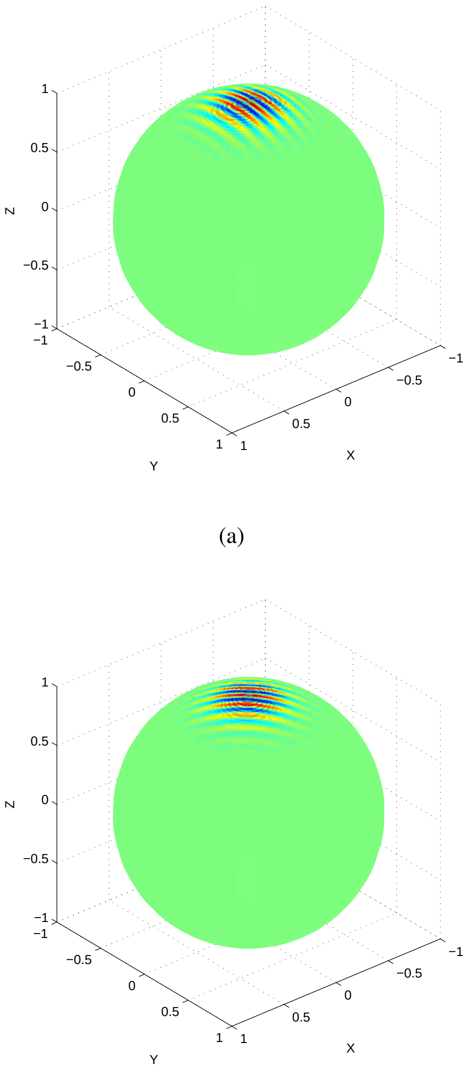
Fig. 5. Schematic diagram for the gaussian pyramid on S^2.

E. Spherical Gabor Pyramid

In order to deal with the orientations in spherical coordinates we proceed toward defining a Gabor Pyramid on the sphere. In fact, there are oriented spherical filters that have anisotropic nature, i.e. they are not axisymmetric. Let us remind, that we have defined in Equation (9) a rotation operator $R_{\alpha_2,\alpha_1,\alpha_0}$ that first rotates the function by α_0 about the x_0-axis, then by α_1 about the x_1-axis and finally by α_2 about x_0-axis again. If a rotation by α_0 takes place, the function is anisotropic, i.e the filter is oriented at, let say, angle $\phi \in [0, 2\pi)$ and thus depends on three rotation angles: $g(\theta, \varphi, \phi)$. We can interprete $g(\theta, \varphi, \phi)$ as $g(\omega, \phi)$, where $\omega \in S^2$ is a position on the sphere and ϕ is an angle of rotation. In fact, on the sphere one can define orientation with respect to meridians and parallels. In other words, directions can be referred to as cardinal points: $\phi = 0^o$ corresponds to North-South direction, i.e. meridians, and $\phi = 90^0$ to East-West directions, i.e. parallels.

The intuition about what a directional filter in spherical domain looks like, is clear: if centered at North Pole, the spherical directional filter has a stereographic projection on the tangent plane that is directional in the Euclidean sense. A natural candidates for such is the Gabor spherical filter:

$$G_{gabor}(\theta, \varphi) = \frac{e^{i|k_0|\tan(\theta/2)\cos(\varphi_0-\varphi)}e^{-(1/2)\tan^2(\theta/2)}}{1+\cos\theta}, \tag{14}$$



(a)

(b)

Fig. 6. Real part of the spherical Gabor wavelet: (a) $a = 0.1$ and orientation 0^0 , (b) $a = 0.1$ and orientation $\frac{\pi}{4}$.

where φ_0 is the argument of $|k_0|$. This filter is (numerically) admissible only for $|k_0|$ large enough, usually $|k_0| \geqslant 6$. Moreover, a spherical Gabor wavelet reads [34], [35]:

$$\psi_{gabor}(\theta, \varphi, \phi) = \lambda^{1/2}(\theta, a) e^{ik_0 \frac{2}{a} \tan \frac{\theta}{2}} \cos{(\phi - \varphi)}\, e^{-\frac{2}{a^2} \tan^2 \frac{\theta}{2}} \big(1 + \frac{1}{a^2} \tan^2 \frac{\theta}{2}\big), \tag{15}$$

where $\lambda(\theta, a)$ is a normalization factor and the scale parameter $a > 0$. A particular example of Gabor spherical wavelet for two different orientations is shown in Figure 6.

The directional correlation on the sphere is expressed in terms of the Wigner D-function coefficients [36] which read:

$$\widehat{g_{gabor} \star f}(l,m,n) = \frac{8\pi^2}{2l+1}\hat{g}^*_{gabor}(l,n)\hat{f}(l,m), \tag{16}$$

where g^* denotes the complex conjugate. In other words, the Wigner D-function coefficients of the directional correlation are given as the pointwise product of the scalar spherical harmonics coefficients $\hat{f}(l,m)$ and $\hat{g}^*(l,n)$. This method is implemented in *SOFT* ($SO(3)$ Fourier Transform) [37].

Now, we can define the procedure for obtaining one level r_k of the *Spherical Gabor Pyramid* :

1) Apply the spherical Fourier transform Equation (12) on the signal f_k and thus obtain $\hat{f}_k(l,m)$;
2) Multiply in Fourier domain with a Gaussian filter as defined in Equation (11) and thus obtain $\widehat{g \cdot f_k(l,m)}$;
3) Apply the inverse spherical Fourier transform on it and thus obtain the filtered signal $g \star f_k$;
4) Subsample this signal $g \star f_k$ by a factor of 2 and thus obtain f_{k+1};
5) Apply again the spherical Fourier transform (Equation (12)) on f_{k+1} and obtain $\hat{f}_{k+1}(l,m)$;
6) Multiply in Wigner domain with a Gabor filter as defined in Equation (16) and obtain $\widehat{g_{gabor} \cdot f}_{k+1}(l,m,n)$;
7) Apply the inverse Wigner transform on it and obtain $r_{k+1} = g_{gabor} \star f_{k+1}$.

The full spherical Gabor pyramid is obtained by iteratively applying the previous procedure to each scale level $k = 1 \cdots n$ and producing the pyramid of spherical signals $r_1, r_2, \cdots r_n$. The schematic representation of the spherical Gabor pyramid is illustrated in Figure 7, where $f_0 \equiv r_0$.

F. Up-Sampling on the Sphere

Up-sampling on the sphere $S^2 \in \mathbb{R}^3$ is the process of increasing the sampling rate of a spherical signal $f(\theta_k, \varphi_k)$. The sampling integer factor U multiplies the sampling rate. This process consists of two steps:

1) Add $(U-1)$ zeros between each sample in $f(\theta_k, \varphi_k)$ defined on the grid $\mathcal{G}_k$ (the spherical grid is defined in Equation (10));
2) Filter with a low-pass spherical filter;

The filtering is once again performed in the Fourier domain and a particular example is the Gaussian spherical filter as defined in Equation (13). For instance, at the lowest level of the spherical Gaussian pyramid ($k = 8$), the signal is defined on a grid of size 4×4. After introducing zeros into it, we obtain a signal defined on (8×8)-spherical grid whose Fourier transform results in 4×4-matrix, i.e. we have $\beta_8 = 4$. Consequently, $\sigma_8 = \frac{\sqrt{2}}{\beta_8} = 0.3536$, satisfies our requirement.

Finally, for completing the set of analyzing tools on the sphere we need to define the basic notion of normalization on the sphere.

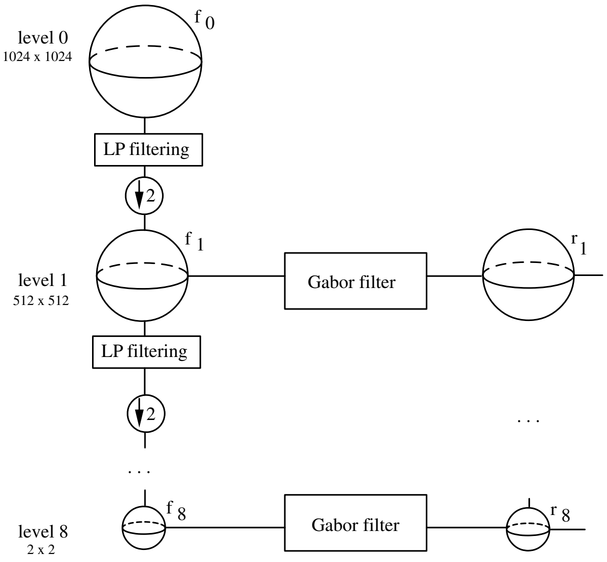


Fig. 7. Schematic diagram for the Gabor pyramid on S^2.

G. *Normalization $\mathcal{N}_{S^2}()$ for Spherical Map Fusion*

The mean value of a function over the surface of a sphere, i.e. $f(\theta, \varphi) \in L^2(S^2)$, in discrete form is given by the double sum:

$$f_{mean_{S^2}} = \frac{1}{4\pi} \sum_{\theta} \sum_{\varphi} f(\theta, \varphi) \sin\theta,$$

where $\theta \in [0, \pi]$ and $\varphi \in (0, 2\pi]$.

Consequently, the normalization function for map fusion on the sphere, takes the following form:

$$\mathcal{N}_{S^2}(C(\theta, \varphi)) = w_{S^2} \cdot C(\theta, \varphi), \tag{17}$$

with

$$w_{S^2} = \frac{4\pi Max(C(\theta, \varphi))}{\sum_{\theta} \sum_{\varphi} C(\theta, \varphi) \sin\theta},$$

where $C(\theta, \varphi)$ is the corresponding spherical conspicuity map.

IV. Visual Attention Algorithm on the Sphere S^2

Now, let us define the procedures for obtaining the different cue conspicuity maps on the sphere so that the spherical saliency map can be obtained. The input signal is a color image defined on the sphere. First, each of the spherical image features are extracted. Then, a conspicuity map for each feature is built. Finally, the spherical saliency map is obtained by fusing together all the spherical cue conspicuity maps. The spots of attention on the sphere are defined using the spherical saliency map.

A. Computing Several Features on the Sphere

First, we need to define each of the features of the spherical image, f_j, for $j = 1 \cdots 7$ as follows:

1) intensity: $f_{int} = 0.3r + 0.59g + 0.11b$;
2) yellow-blue: $f_{BY} = \frac{(b-r/2-b/2)}{f_{int}}$;
3) red-green: $f_{RG} = \frac{(r-g)}{f_{int}}$;
4) four orientations: $f_{0^0}, f_{45^0}, f_{90^0}, f_{135^0}$. They are obtained applying a Gabor pyramid on f_{int} where the Gabor filter is defined in Equation (15) with $\phi \in \{0^0, 45^0, 90^0, 135^0\}$, respectively and $k_0 = 30, a = 0.03$;

B. Spherical Conspicuity Map for Each Feature

Let us have a spherical signal (image) f_0 defined on a grid of size $2^{n+2} \times 2^{n+2}$. The procedure for computing the spherical feature conspicuity map C_j relies on the spherical Gaussian pyramid and the center-surround mechanism. It includes the following steps:

1) construct the n-level spherical gaussian pyramid as described in Section III-D;
2) compute the multiscale maps as defined in Equation (6);
3) compute the weight coefficients w_{S^2} and normalize the maps as defined in Equation (17);
4) compute the final *spherical feature conspicuity map* C_j using Equation (7).

This procedure is applied to each of the features in order to compute seven spherical conspicuity maps $C_j, j = 1 \cdots 7$.

C. Spherical Cue Conspicuity Maps

Using the seven feature conspicuity maps as obtained in Section IV-B, three cue conspicuity maps are computed as follows:

1) $C_{int} = C_1$;
2) $C_{chrom} = \frac{\mathcal{N}_{S^2}(C_2)+\mathcal{N}_{S^2}(C_3)}{2}$, where C_2 is the red-green spherical conspicuity map and C_3 is the yellow-blue conspicuity map. They are normalized according Equation (17) ;
3) $C_{orient} = \frac{\mathcal{N}_{S^2}(C_4)+\mathcal{N}_{S^2}(C_5)+\mathcal{N}_{S^2}(C_6)+\mathcal{N}_{S^2}(C_7)}{4}$, where C_4, C_5, C_6, C_7 are obtained after applying the procedure in Section IV-B on the four orientation features from Section IV-A-4, and normalized according Equation (17).

D. Spherical Saliency Map

The spherical saliency map is computed by fusing together all cue conspicuity maps obtained in Section IV-C:

$$S_{S^2} = \sum_{cue \in int, chrom, orient} \mathcal{N}(C_{cue}), \tag{18}$$

where $\mathcal{N}()$ is the normalization step according to Equation (17). Due to the different nature of the spherical cue conspicuity maps, the conspicuity cues are previously scaled at the same range values by applying a peak-to-peak normalization.

E. Spots of Attention on the Sphere

The consecutive maxima in the spherical saliency map represent the most salient locations on the sphere, which actually define the spots of attention. The spots are detected successively using the "winner-take-all" mechanism, as previously discussed in Section II-A. A local inhibition on the sphere takes place. The inhibition function used to attenuate the consecutive spots of attention is defined by

$$F_{inh}(\omega) = 1 - G_{S^2}(\omega) \in L^2(S^2), \quad \omega \equiv (\theta, \varphi) \in S^2 \tag{19}$$

where $\theta \in [0, \pi], \varphi \in [0, 2\pi)$ and the spherical Gaussian reads

$$G_{S^2}(\omega) = e^{-\eta^2 \tan^2 \frac{\theta}{2}}, \quad \eta \in \mathbb{R}_+, \tag{20}$$

which is the inverse stereographic projection of the Gaussian in the tangent plane at the North pole of the sphere. The size of the filter depends on the parameter η.

The process of local inhibition consists of multiplying the spherical saliency map by the function defined in (19) which is placed at the considered maximum ω_{max}:

$$S_{inh} = S_{S^2} \cdot F_{inh}(\omega - \omega_{max}). \tag{21}$$

Finally, the number of detected locations can be either set by the user or automatically determined through the activities of the saliency map using a given threshold value.

It is clear, that by now we have used a pyramidal approach to define the VA model on the sphere. However, other approaches can be considered as well.

F. Toward Another Approach for VA on the Sphere

A filter-based approach can be applied for computing each spherical conspicuity map. This approach does not consider a pyramidal architecture but keeps the same size of the input signal while the filter size is varying. The intensity and chromatic feature can be extracted applying a bank of difference of Gaussian (DOG) spherical filters. For extracting the orientation features, a bank of spherical Gabor filters can be used. In both cases, the multiresolution analysis on the sphere is achieved through wavelet approach instead of building a pyramid. This approach is supposed to be more precise but it is expected to be more expensive in terms of computation time.

V. Saliency Map on a Spherical Synthetic Signal

In this section we compare the spherical saliency-based model with Euclidean one, in an experiment with a spherical synthetic signal. On one hand, we compute the saliency map according to the spherical model and on the other hand, according to the Euclidean one by applying the classical implementation on the unwrapped synthetic signal.

A given spherical synthetic signal f_0 consists of twelve white disks distributed along a meridian of the sphere. The 12 disks are represented in four quadrant, each one containing a group of three disks of a same given size. This results in four disk groups of four different sizes. It is illustrated in Figure 8(a), where four views of the sphere in 3-D space are provided, each view representing one of the four groups. The signal is defined on a 1024×1024 equi-angular spherical grid (θ, φ). Its unwrapped version is shown on Figure 8(b), where $\varphi \in [0, 2\pi)$ is at the horizontal axis and $\theta \in [0, \pi]$ is at the vertical axis. The beginning and the end of the vertical axis correspond to the South and North Poles of the sphere, respectively. The four groups of disks are easily distinguished but it is clear that each of the disks is deformed when the spherical signal is unwrapped as illustrated in Figure 8(b). Actually, in this and in the following examples, the unwrapped version is provided for purpose of visualization.

According to the algorithm defined in Section IV-B, we create first the 8 levels of the spherical gaussian pyramid, then the six corresponding multiscale conspicuity maps and obtain finally, after normalization and summation, the overall saliency map.

By analysing the spherical saliency map, (Figure 8(c) and (d)), one can easily see that each disk of the same size (disk in the same quadrant) provides the same type of response. This suggests that the performed VA computation is independent of the spatial location on the sphere and that the proposed algorithm operates thus homogeneously as expected.

TABLE I

MEAN VALUE AND STANDARD DEVIATION FOR THE SPHERICAL AND EUCLIDEAN SALIENCY MAPS

	spherical			Euclidean		
group	μ	σ	$\frac{\sigma}{\mu}$	μ	σ	$\frac{\sigma}{\mu}$
I	242	7	**0.02**	159	59	**0.37**
II	243	12	**0.04**	228	32	**0.14**
III	64	5	**0.08**	129	59	**0.45**
IV	165	5	**0.03**	223	23	**0.1**

For better understanding the differences between the saliency map computed in the spherical geometry and the corresponding one in the Euclidean geometry, we apply the Euclidean VA algorithm on the unwrapped spherical signal f_0 (i.e. f_0 is considered to be an Euclidean signal (as if it were defined on a cartesian grid (x, y))), and then compare both results. Figure 8 provides such a comparison by displaying the saliency maps obtained by the Euclidian (Figure 8(e) and (f)) and spherical (Figure 8(c) and (d)) approaches. For a given group of three disks of a specific size, the Euclidean model provides non-homogeneous saliency response while we expect the model to detect three identical saliency response for disks of same size. In spherical geometry, the type of response is identical for disks of the same size and this illustrates the homogeneous saliency response everywhere on the sphere.

Moreover, considering the obtained saliency maps in Figure 8(d) and (f) we proceed toward a quantitative comparison. Namely, for each group of three identical disks, we compute the mean saliency value μ and standard deviation σ at the center of the disk. Table I summarizes the obtained values measured for each of the four groups (In Figure 8(b), group I is located at up-left, group II at down-left, group III at up-right and group IV at down-right). By comparing the ratio $\frac{\sigma}{\mu}$, we observe a clear higher variability for the Euclidean case with respect to the spherical one. An example, for group III, the Euclidean case have a ratio five times higher than the spherical case. Since the disks of a given group are identical, the variability is expected to be as low as possible. Thus, this quantitative measure clearly confirms the qualitative evaluation. We notice that the ratios in the spherical case indicate a non-null variability (between 2% and 8%), suggesting that the saliency response is only approximately independent of the location on the sphere. Actually, this is explained by the fact that the saliency computation results from center-surround difference. Indeed, the surroundings of the disks are different, which explains the low variability. In order to examine the rotation invariance of the saliency response, we perform in Section IX a comparison of the spots of attention in a spherical image that has been rotated by different angles.

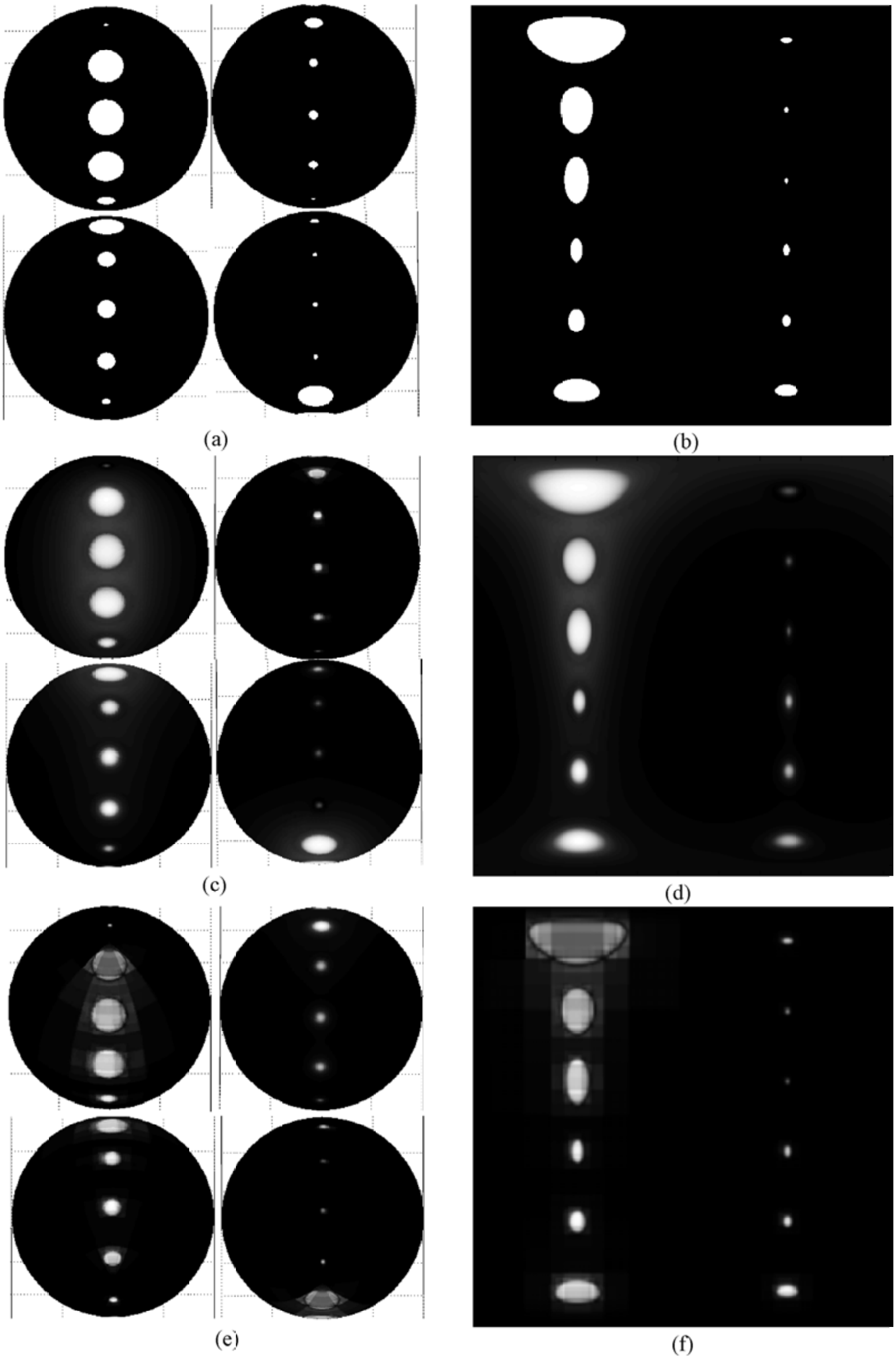


Fig. 8. Euclidean vs. spherical saliency map: (a) synthetic signal - four views of the sphere, (b) unwrapped spherical synthetic image, (c) spherical saliency map represented on the sphere, (d) unwrapped spherical saliency map, (e) Euclidean saliency map represented on the sphere, (f) saliency map obtained after applying the Euclidean VA on the signal given in (b).

To summurize, the experiment with a synthetic spherical signal provided in this section illustrates the capacity of the proposed VA algorithm to process the features and, consequently, saliency map in a way which is independent of the spatial location of on the sphere. It also shows that this property is missing when the signal is processed using the Euclidean VA.

We will see in Section VIII that distortion leads to inaccurate spot detection, due to non-homogeneous response in the Euclidean model, especially when salient objects are located at the sphere's poles.

VI. Visual Attention on a Real Omnidirectional Image

We have seen how the spherical saliency map differs from the Euclidean one while performing the experiment in the previous section. Now, we proceed toward applying our algoritm on a real omnidirectional image, which represents the entire sphere of view. We first compute the spherical saliency map and then define the spots of attention.

A. Spherical Saliency Map

Let us start with an omnidirectional (r, g, b) spherical image of size 1024×1024 which is obtained by a multi-camera sensor [38]. The input spherical image is shown in Figure 9(a), where two views of the sphere in 3-D are represented. Seven features are considered and derived as in Section IV-A. The obtained feature conspicuity maps for the different features, intensity, red-green, and yellow-blue, are shown on Figure 9(b), (c) and (d), and the orientation features are shown in (e), (f), (g) and (h) respectively. Consequently, three spherical cue conspicuity maps are calculated: C_{int}, C_{chrom} and C_{orient} as described in Section IV-C.

For the computation of the spherical saliency map, the procedure of Section IV-D applies. It is shown in Figure 10(c). For better visualization, we provide its unwrapped version as well, in Figure 10(d).

B. Spots of Attention

The spots of attention are defined as described in Section IV-E. The local inhibition is performed using Equation 19. Twelve spots of attention are detected and shown in Figure 10(e) on the sphere. The unwrapped spherical image with the corresponding spots is illustrated as well in Figure 10(f).

VII. VA in Different Omnidirectional Scenes

We have experimented the spherical VA on 20 omnidirectional images, representing different scenes. In this section we present three of them illustrated in Figure 11: (a) and (b) represent the omnidirectional

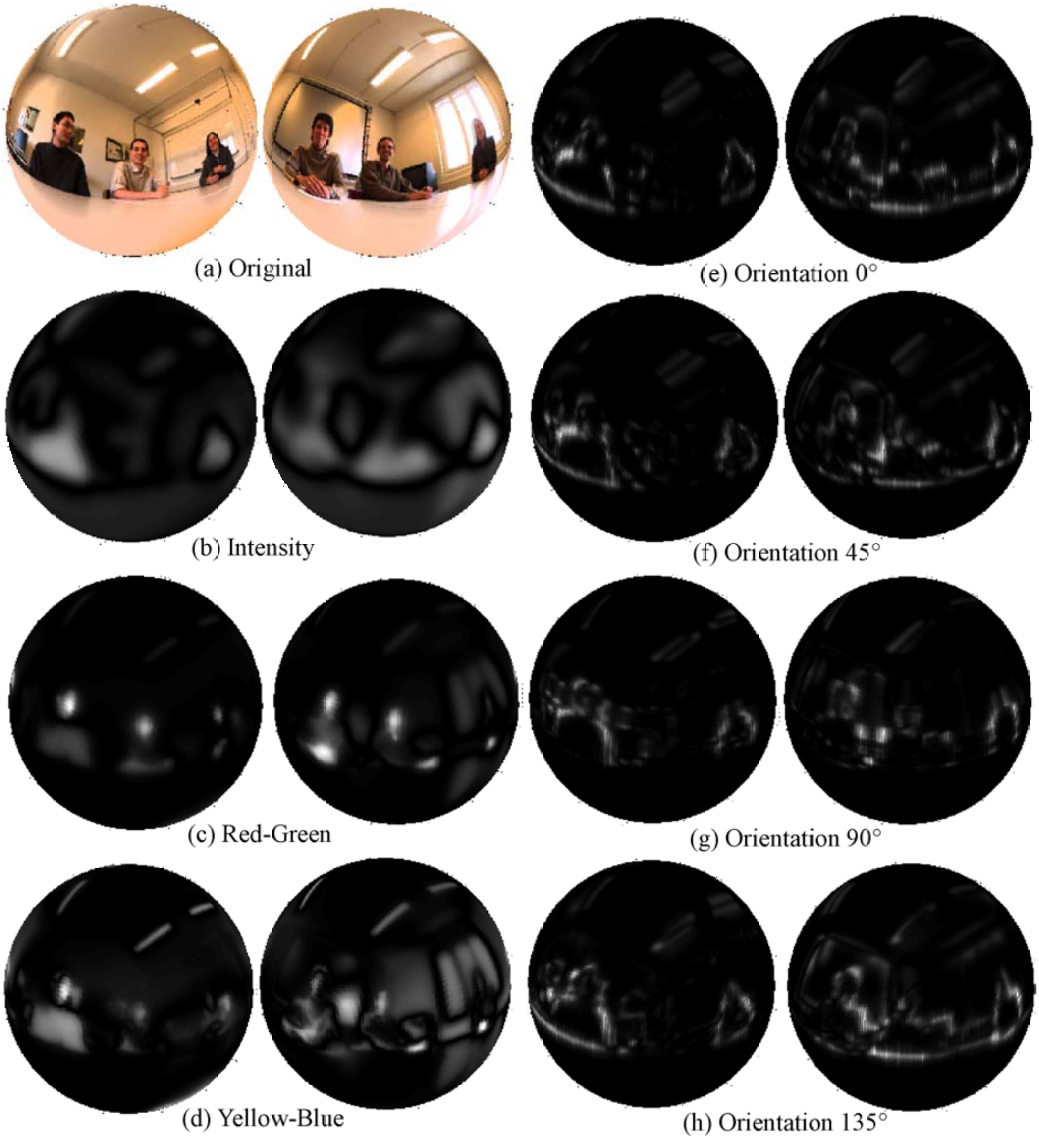


Fig. 9. Spherical feature conspicuity maps of a real omnidirectional image (3D views of the sphere): (a) original spherical image; (b) intensity conspicuity map C_{int}; (c) red-green conspicuity map C_{RG}; (d) yellow-blue conspicuity map C_{YB}; (e) orientation conspicuity map C_{0^0}, (f) orientation conspicuity map C_{45^0}, (g) orientation conspicuity map C_{90^0}, (h) orientation conspicuity map C_{135^0}.

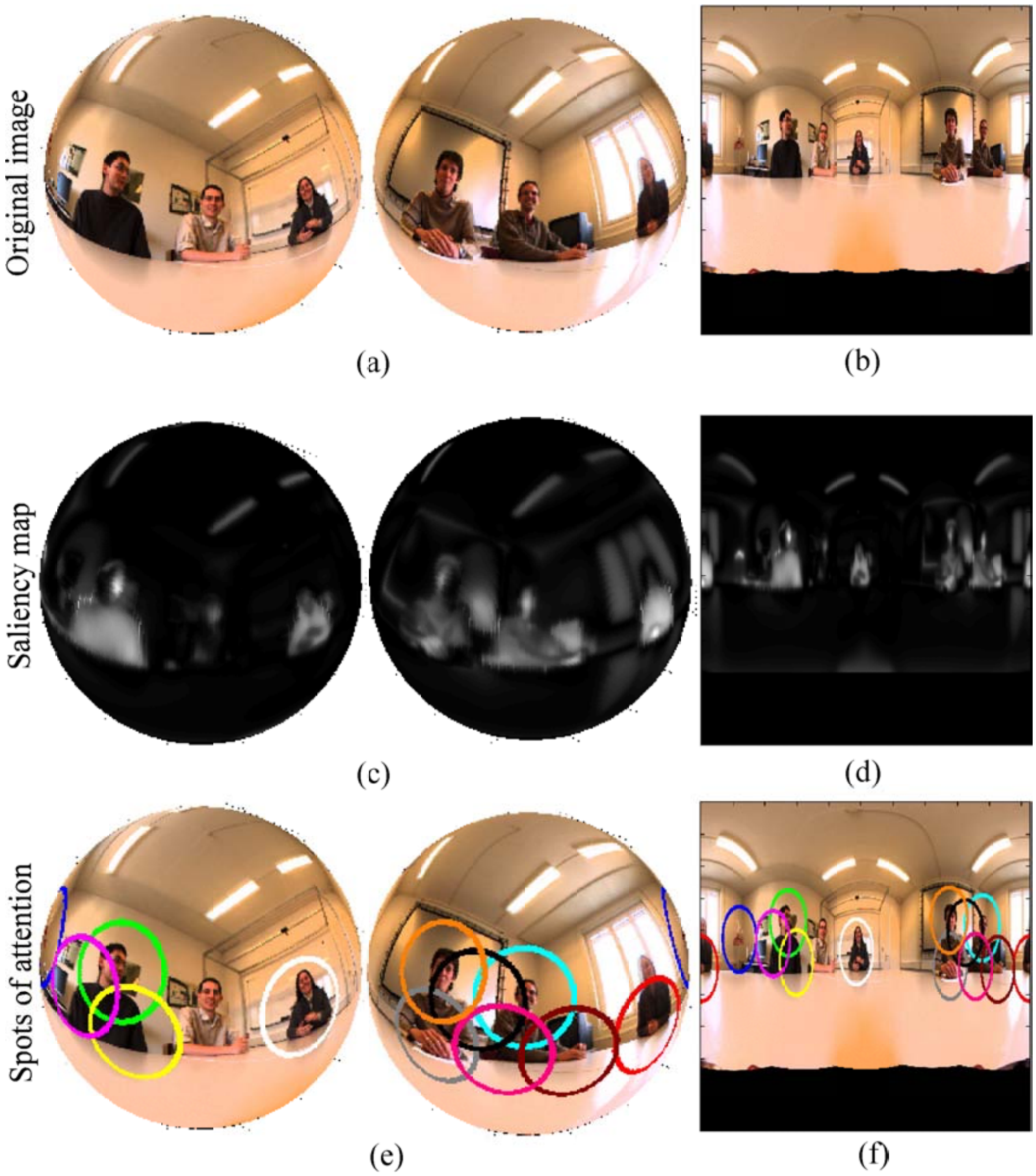


Fig. 10. Spots of attention on a real spherical image: (a) original spherical image, (b) unwrapped spherical image; (c) spherical saliency map; (d) unwrapped spherical saliency map; (e) spots of attention on the sphere; (f) unwrapped sphere with its spots of attention.

image and the spherical image, respectively, while (c) and (d) show the spots of attention and the spherical saliency map.

In the first image, the camera is placed on the table in an office. In the second and third image the camera is fixed at the ceiling of a meeting room and is pointing down the table. The detected spots illustrated in these examples are located everywhere on the sphere (including the poles) at locations containing strong feature contrasts (i.e. salient objects on the table).

VIII. Euclidean vs. Spherical VA

For better illustrating the advantages of the VA on the sphere, we provide a particular example of an omnidirectional image, on which both Euclidean and spherical visual attention are applied.

The input spherical image is shown on Figure 12 (a), where both its 3-D views on the sphere and its unwrapped version are illustrated. The scene represents a meeting room and is captured by a multi-camera omnidirectional system placed in the center and pointing down the table. There are two red salient objects on the table. One of them, which is located in the middle of the table and is bigger in size, appears at the South Pole on the sphere. It is identified among the whole bottom of the unwrapped spherical image. After applying the spherical VA we obtain the spherical saliency map which is illustrated in Figure 12(b), again in both 3D and unwrapped versions. As shown in Figure 12(c), twelve spots of attention are detected based on ”winner-take-all” mechanism. The first three among them are ranked.

Then we consider the unwrapped spherical image as if it were an Euclidean (flat) image and consequently, we apply the Euclidean VA. The Euclidean spots of attention are shown as well on Figure 12 (c). It detects one of the red objects as most salient spot (rank 1), but does not detect the other at all. In fact, when unwrapped, the bigger red object is completely distorted at the bottom of the Euclidean version of the omnidirectional image. Here, we refer to a distortion, as any deformation in the scene resulting from unwrapping of the omnidirectional image. If the omnidirectional image was projected to a panoramic one, the same red object would be distorted in the same way.

From this comparison, it is clear, that the salient object situated on the pole is detected only by the spherical VA. This particular omnidirectional image is, actually, a typical example where the Euclidean VA fails in correctly detecting spots of attention in omnidirectional images. In fact, the more we approach the poles, the more the distortions are important regarding the Euclidean VA. Thus the correct way to deal with distortions in omnidirectional images is to work on the sphere, because this is the natural domain for any omnidirectional scene.

In this section, we have demonstrated that the spherical VA performs better in omnidirectional images

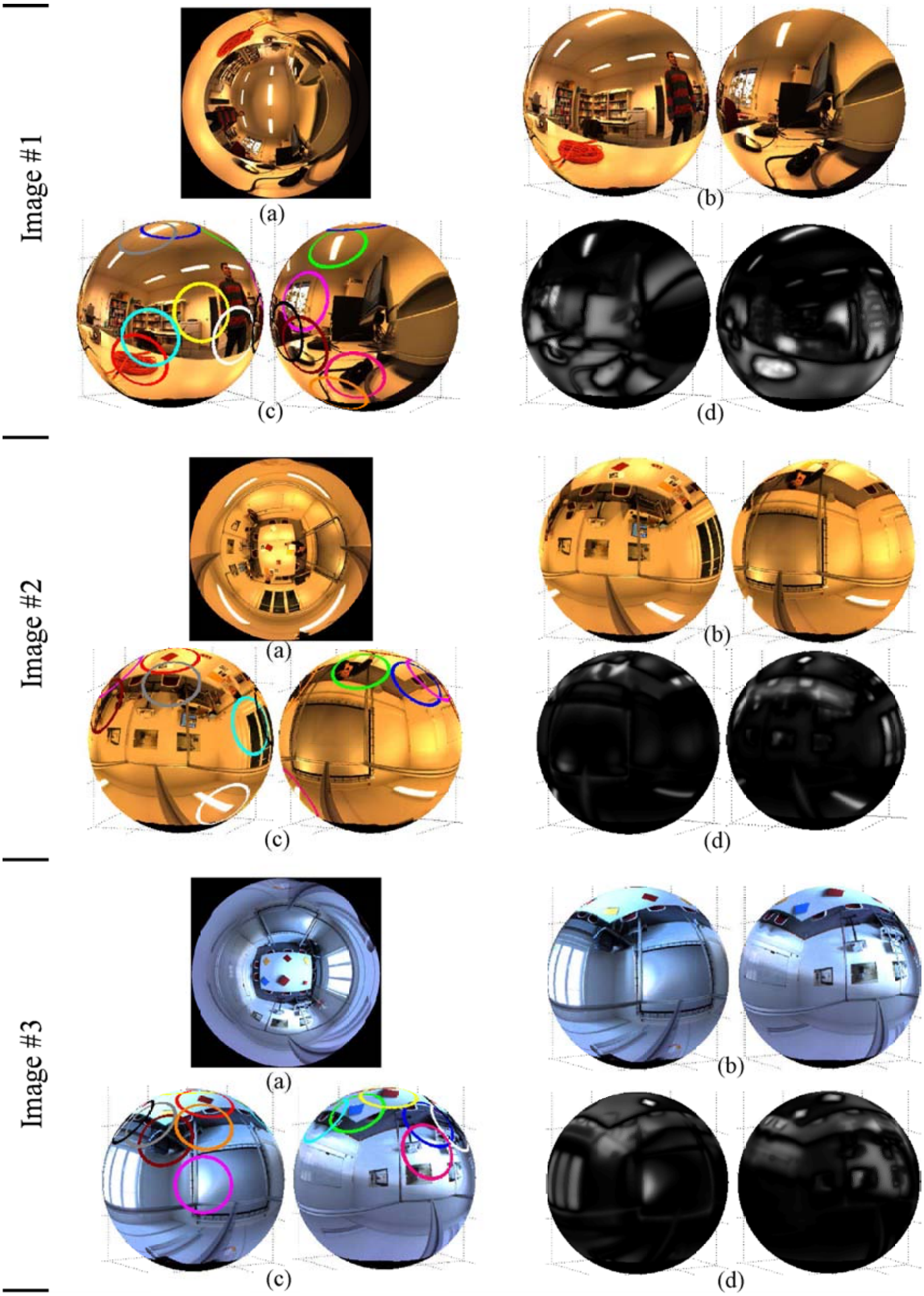


Fig. 11. VA in omnidirectional images: (a) omnidirectional image, (b) omnidirectional image mapped on the sphere, (c) spots of attention on the sphere, (d) spherical saliency map.

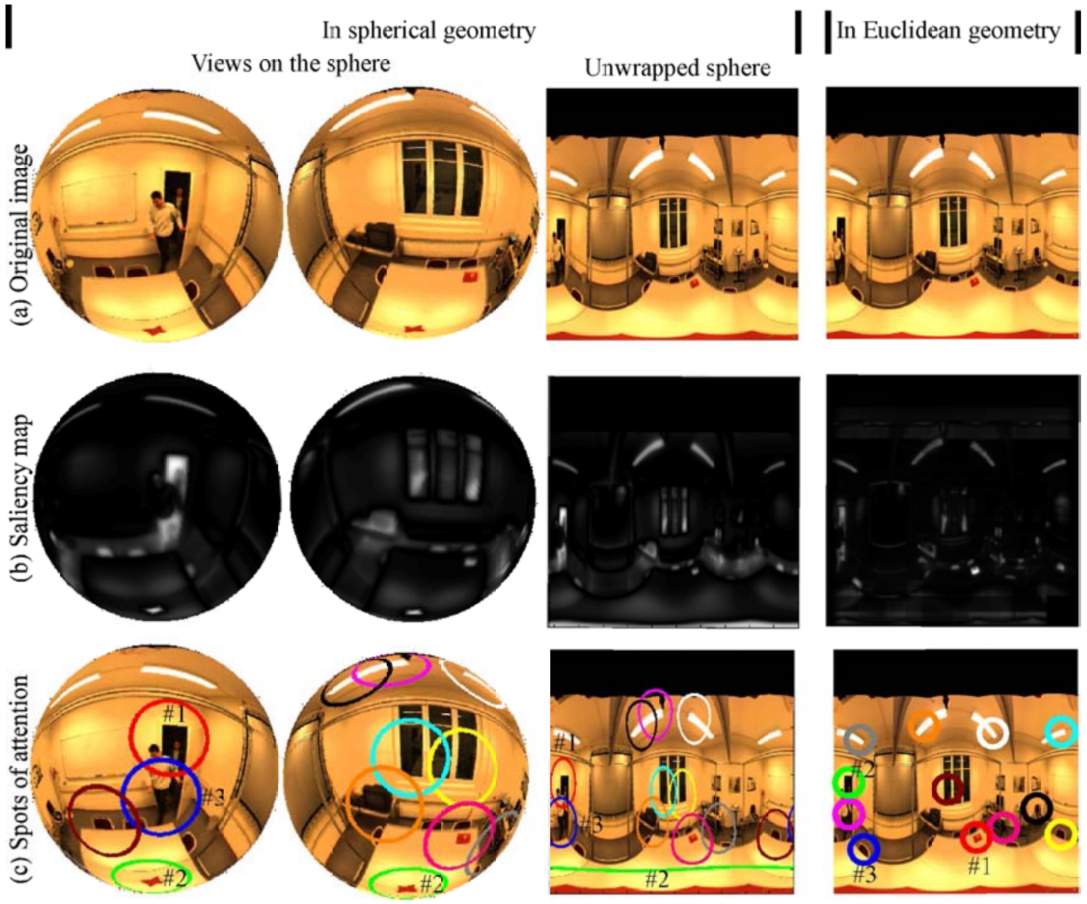


Fig. 12. Euclidean vs. spherical VA: (a) input signal; (b) saliency map; (c) spots of attention

than the Euclidean VA. Another way to validate a VA model, is to perform psycho-physical experiments. Since a human observing the whole omnidirectional scene from a sphere's center is an impossible set-up, one might consider a set-up where the omnidirectional images in their disk-like form (Figure 11(a)) is used. An eye tracker system is recording the eye movement patterns while the human observer is looking at the omnidirectional image. Then, the experimental data is compared to the spherical saliency map projected back to its disk-like form.

IX. Homogeneous VA detection

Now, we examine the homogeneity of the spherical VA model and consequently compare it with the Euclidean case. For this purpose, a rotation by (θ, φ) on the spherical image is applied. The original image is assumed to have no rotation, i.e $\theta = 0^0, \varphi = 0^0$ as shown on Figure 13(a). Then we consider three more possibilities: rotation only by the angle φ, i.e. $\theta = 0^0, \varphi = 90^0$ as shown in Figure 13(c); rotation by angle θ, i.e. $\theta = 90^0, \varphi = 0^0$ as illustrated in Figure 13(e); and rotation by both angles, i.e. $\theta = 90^0, \varphi = 90^0$. The spots of attention according to the spherical model are represented in Figure 13((a), (c), (e), (g)) while the spots detected with the Euclidean model are represented on the same Figure in (b), (d), (f) and (h). In all of the cases, three spots of attention are considered and their rank is shown as well.

Under a closer observation, we can easily see that the spherical model always detects the same objects with the same ranking independently of the rotation. Concerning the Euclidean VA, the detected spots of attention are not the same (as in (d), (f) and (h)) and this illustrates a rotation dependance of this model.

In conclusion, the experiment performed in this section illustrates that the VA on the sphere is rotation-invariant while it is clearly not the case for the Euclidean VA.

Conclusions

In this paper we have defined a new visual attention algorithm for images defined on the sphere. This new algorithm operates in the spherical domain. First, the multiresolution analysis on the sphere is used to determining the spherical feature conspicuity maps and consequently, the spherical cue conspicuity maps and the final spherical saliency map. Then, the consecutive maxima in the spherical saliency map represent the spots of attention. Operating on the sphere, the new algorithm provides a homogeneous saliency response and spot detection, and thus best suited for omnidirectional images. The property to perform homogeneously was illustrated in a comparison of saliency maps obtained by processing a synthetic spherical signal with the Euclidean VA algorithm on one hand, and with the new algorithm, on the other hand. Another comparison was performed on a real omnidirectional image obtained with a multi-camera sensor. In both comparisons it is clear that the spherical VA remedies the problem of the distortion, persistent in omnidirectional imaging, which is not the case of the Euclidean VA. While the Euclidean model provides a rough approximation of the spherical one, it is not able to detect salient regions on the poles. This is due to distortions increasing with latitude and becoming critical for VA detection on the poles. In contrast, operating VA on the sphere allows an accurate detection in the full omnidirectional scene. As a demonstration, the algorithm was applied with success on different omnidirectional images. Finally, the application perspectives are quite universal, as the method is basically applicable to any omnidirectional image that can be mapped onto the sphere. Such are the images obtained with, for instance, a hyperbolic or parabolic catadioptric sensor.

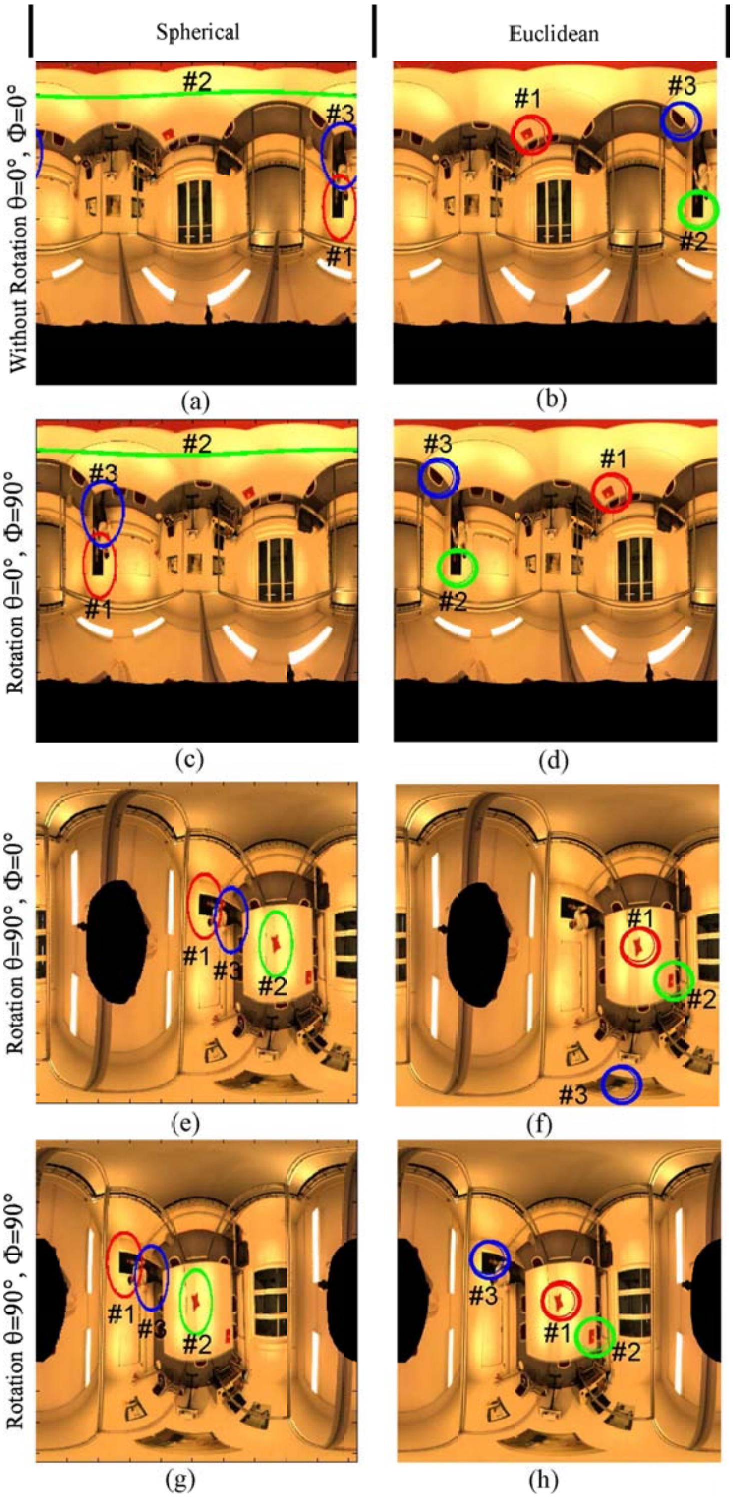


Fig. 13. Spherical vs. Euclidean spots detection: (a), (c), (e), (g) spots of attention detected by the spherical VA applied on an omnidirectional image after it has been rotated; (b), (d), (f), (h) spots of attention detected by the Euclidean VA on the same image. The spherical VA detects the same spots even after the image rotation.

In a future work, we intend to explore the dynamic visual attention on the sphere, where the motion will be considered and integrated.

ACKNOWLEDGMENTS

The authors acknowledge the support of the Swiss National Center for Competence in Research IM2 and Swiss National Science Foundation under contract FN-108060. The authors would also like to thank the four unknown referees for their remarks and comments, which considerably improved the quality of the paper.

REFERENCES

[1] N. Ouerhani, *Visual Attention: from bio-inspired Modeling to Real-Time Implementation (PhD Thesis pp.42-52)*, http://www-imt.unine.ch/parlab/, 2004.

[2] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE Transaction on Image Processing*, vol. 13, pp. 1304–1318, 2004.

[3] D. Walther, U. Rutishauser, C. Koch, and P. Perona, "Selective visual attention enables learning and recognition of multiple objects in cluttered scenes," in *Computer Vision and Image Understanding*, vol. 100, 2005, pp. 41–63.

[4] S. Frintrop, *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*, ser. Lecture Notes in Artificial Intelligence (LNAI). Springer, 2005.

[5] N. Ouerhani and H. Hügli, "MAPS: Multiscale attention-based presegmentation of color images," in *4th International Conference on Scale-Space theories in Computer Vision*, ser. LNCS, vol. 2695, 2003, pp. 537–549.

[6] A. Bur, A. Tapus, N. Ouerhani, R. Siegwart, and H. Hügli, "Robot navigation by panoramic vision and attention guided fetaures," in *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, Washington, DC, USA, 2006, pp. 695–698.

[7] C. Siagian and L. Itti, "Biologically-inspired robotics vision monte-carlo localization in the outdoor environment," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[8] A. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, pp. 97–136, 1980.

[9] J. Wolfe, "Guided search 2.0: A revised model of visual search," *Psychonomic Bulletin and Rewiew*, pp. 1(2):202–238, 1994.

[10] F. H. Hamker, "Modeling attention: from computational neuroscience to computer vision," in *Proc. of the 2nd International Workshop on Attention and Performance in Computational Vision (WAPCV04)*, 2004, pp. 59–66.

[11] C. Koch and S. Ullman, "Shifts in selective visual attention: Towards the underlying neural circuitry," *Human Neurobiology*, vol. 4, pp. 219–227, 1985.

[12] R. Milanese, *Detecting salient regions in an image: from biological evidence to computer implementation*, PhD thesis, University of Geneva, Switzerland, 1994.

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

[14] L. Itti and C. Koch, "Computational modeling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[15] O. Le Meur, and P. Le Callet, and D. Barba, and D. Thoreau, "A coherent computational approach to model bottom-up visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 802–817, 2006.

[16] J. Tsotsos, S. M. Culhane, W. Wai, Y. Lai, N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning," *Artificial Intelligence*, vol. 78, no. 1-2.

[17] B. Olshausen, C. Anderson, and D. van Essen, "A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing information," *Journal of Neuroscience*, vol. 13, no. 11.

[18] V. Navalpakkam and L. Itti, "Modeling the influence of task on attention," *Vision Research*, vol. 45, no. 2, pp. 205–231, 2005.

[19] R. Milanese, H. Wechsler, S. Gil, J. Bost, and T. Pun, "Integration of bottom-up and top-down cues for visual attention using non-linear relaxation," in *Computer Vision and Pattern Recognition CVPR 94*, 1994, pp. 781–785.

[20] E. H. Adelson and J. Bergen, "The plenoptic function and the elements of early vision," *Annals of Physics*, vol. 1, pp. 1–37, 1993.

[21] K. Yamazawa, Y. Yagi, and M. Yachida, "Omnidirectional imaging with hyperboloidal projection," in *Proceedings of IROS*, Jokohama, Japan, 1993.

[22] Y. Yagi, "Omnidirectionl sensing and its applications," in *IEICE Trans. Inf. Syst.*, vol. 82(3), 1999, pp. 2568–578.

[23] C. Geyer and K. Daniilidis, "Catadioptric projective geometry," *International Journal of Computer Vision*, vol. 45(3), pp. 223–243, 2001.

[24] T. E. Boult, X. Gao, R. J. Micheals, and M. Eckmann, "Omni-directional visual surveillance." *Image Vision Comput.*, vol. 22, no. 7, pp. 515–534, 2004.

[25] H. Watanube, H. Tanabashi, Y. Satoh, Y. Niwa, and K. Yamamoto, "Event detection for a visual surveillance system using stereo omni-directional system," *Knowledge-based Intelligent Information and Engineering Systems*, 2003.

[26] Y. Yagi and M. Yachida, "Real-time omnidirectional image sensors," *International Journal of Computer Vision*, vol. 58(3), pp. 173–207, 2004.

[27] E. Menegati, T. Maeda, and H. Ishiguro, "Image-based memory for robot navigation using properties of omnidirectional images," *Robotics and Autonomous Systems*, vol. 47, no. 4, pp. 251–267, 2004.

[28] A. Bur and H. Hügli, "Optimal cue combination for saliency computation: A comparison with human vision," in *Second International Work-Conference on the Interplay Between Natural and Artificial Computation (IWINAC 2007)*, ser. Lecture Notes in Computer Science, vol. LNCS 4528, 2007, pp. 109–118.

[29] I. Bogdanova, "Wavelets on non-euclidean manifolds," Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2005.

[30] D. M. Healy, D. Rockmore, P. J. Kostelec, and S. S. B. Moore, "FFTs for the 2-Sphere - implementations and variations," vol. 9, no. 4, 2003, pp. 341–385.

[31] J. R. Driscoll and D. M. Healy", "Computing Fourier transforms and convolutions on the 2-sphere," *Adv. in Appl. Math.*, vol. 15, pp. 202–250, 1994.

[32] D.Rockmore, S. Moore, D. Healy, and P. Kostelec, *SpharmonicKit is freely available collection of C programs for doing Legendre and scalar spherical transforms developed at Dartmouth College; available at http://www.cs.dartmouth.edu/ geelong/sphere/.*

[33] YAWtb, *http://rhea.tele.ucl.ac.be/yawtb/.*

[34] J.-P. Antoine, L. Demanet, L. Jacques, and P. Vandergheynst, "Wavelets on the sphere: implementations and approximations," *Applied and Computational Harmonic Analysis*, vol. 13, pp. 177–200, 2001.

[35] L. Demanet and P. Vandergheynst, “Gabor wavelets on the sphere,” in *Proc. SPIE Wavelets X conf.*, San Diego, USA.

[36] Y. Wiaux, L. Jacques, P. Vielva, and P. Vandergheynst, “Fast directional correlation on the sphere with steerable filters,” *The Astrophisical Journal*, vol. 652, pp. 820–832, 2006.

[37] P. J.Kostelec and D. Rockmore., “FFTs on the Rotation Group,” *Santa Fe Institute Working Papers Series, http://www.cs.dartmouth.edu/geelong/*, vol. 03-11-060, 2003.

[38] LADYBUG, *http://www.ptgrey.com/products/spherical.asp*.

Bibliography

[1] L. Itti. *Models of Bottom-Up and Top-Down Visual Attention*. Phd thesis, Pasadena, California, Jan 2000.

[2] R. Desimone and J. Duncan. Neural mechanisms of selective visual-attention. *Annual Review Of Neuroscience*, 18:193–222, 1995.

[3] M. Corbetta and G. L. Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews Neuroscience*, 3(3):201–215, 2002.

[4] C. Koch and S. Ullman. Shifts in selective visual-attention - towards the underlying neural circuitry. *Human Neurobiology*, 4(4):219–227, 1985.

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

[6] A. Bur, P. Wurtz, R. M. Muri, and H. Hugli. Motion integration in visual attention models for predicting simple dynamic scenes. *Human Vision and Electronic Imaging XII*, 6492, 2007.

[7] A. Bur, P. Wurtz, R. M. Muri, and H. Hugli. Dynamic visual attention: competitive versus motion priority scheme. *Proc. Int. Conf. Computer Vision Systems (ICVS), Bielefeld, Germany*, 2007.

[8] A. Bur, P. Wurtz, R. M. Muri, and H. Hugli. Dynamic visual attention: Motion direction versus motion magnitude. *Human Vision And Electronic Imaging XIII*, 6806, 2008.

[9] A. Bur and H. Hugli. Optimal cue combination for saliency computation: A comparison with human vision. *Nature Inspired Problem-Solving Methods in Knowledge Engineering, Pt 2, Proceedings*, 4528:109–118, 2007.

[10] S. Frintrop. Vocus: a visual attention system for object detection and goal-directed search. *VOCUS: a visual attention system for object detection and goal-directed search*, pages xii+216, 2006.

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

[12] D. Walther, U. Rutishauser, C. Koch, and P. Perona. Selective visual attention enables learning and recognition of multiple objects in cluttered scenes. *Computer Vision And Image Understanding*, 100(1-2):41–63, October 2005.

[13] N. Ouerhani and H. Hugli. Maps: Multiscale attention-based presegmentation of color images. *Scale Space Methods In Computer Vision, Proceedings*, 2695:537–549, 2003.

[14] A. Bur, A. Tapus, N. Ouerhani, R. Siegwart, and H. Huegli. Robot navigation by panoramic vision and attention guided features. *18th International Conference on Pattern Recognition, Vol 1, Proceedings*, pages 695–698, 2006.

[15] C. Siagian and L. Itti. Biologically-inspired robotics vision monte-carlo localization in the outdoor environment. *2007 Ieee/Rsj International Conference On Intelligent Robots And Systems, Vols 1-9*, pages 1729–1736, 2007.

[16] S. Frintrop, P. Jensfelt, and H. Christensen. Simultaneous robot localization and mapping based on a visual attention system. *Attention In Cognitive Systems - Theories And Systems From An Interdisciplinary Viewpoint*, 4840:417–430, 2007.

[17] R. Milanese, H. Wechsler, S. Gil, J. M. Bost, and T. Pun. Integration of bottom-up and top-down cues for visual-attention using nonlinear relaxation. *1994 Ieee Computer Society Conference On Computer Vision And Pattern Recognition, Proceedings*, pages 781–785, 1994.

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

[19] B. A. Olshausen, C. H. Anderson, and D. C. Vanessen. A neurobiological model of visual-attention and invariant pattern-recognition based on dynamic routing of information. *Journal Of Neuroscience*, 13(11):4700–4719, November 1993.

[20] J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. H. Lai, N. Davis, and F. Nuflo. Modeling visual-attention via selective tuning. *Artificial Intelligence*, 78(1-2):507–545, October 1995.

[21] L. Itti and C. Koch. A comparison of feature combination strategies for saliency-based visual attention systems. *Human Vision And Electronic Imaging IV*, 3644:473–482, 1999.

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

[23] F. H. Hamker. Modeling attention: From computational neuroscience to computer vision. *Attention And Performance In Computational Vision*, 3368:118–132, 2005.

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

[25] J. M. Wolfe. Guided search 2.0 - a revised model of visual-search. *Psychonomic Bulletin & Review*, 1(2):202–238, June 1994.

[26] R. Milanese. *Detecting salient regions in an image: From biological evidence to computer implementation*. Phd thesis, Geneva, Switzerland, 1993.

[27] T. Watanabe, Y. Sasaki, S. Miyauchi, B. Putz, N. Fujimaki, M. Nielsen, R. Takino, and S. Miyakawa. Attention-regulated activity in human primary visual cortex. *Journal Of Neurophysiology*, 79(4):2218–2221, 1998.

[28] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cereb Cortex*, 1(1):1–47, 1991.

[29] J. K. Tsotsos, Y. J. Liu, J. C. Martinez-Trujillo, M. Pomplun, E. Simine, and K. H. Zhou. Attending to visual motion. *Computer Vision And Image Understanding*, 100(1-2):3–40, 2005.

[30] L. Itti. Quantifying the contribution of low-level saliency to human eye movements in dynamic scenes. *Visual Cognition*, 12(6):1093–1123, 2005.

[31] O. Le Meur, P. Le Callet, and D. Barba. Predicting visual fixations on video based on low-level visual features. *Vision Research*, 47:2483–2498, 2007.

[32] M. Guironnet, N. Guyader, D. Pellerin, and P. Ladret. Spatio-temporal attention model for video content analysis. *2005 International Conference on Image Processing (ICIP), Vols 1-5*, pages 2989–2992, 2005.

[33] J.M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models in complex image sequences. *Signal Processing VII, Theories and Applications. Proceedings of EUSIPCO-94. Seventh European Signal Processing Conference—Signal Processing VII, Theories and Applications. Proceedings of EUSIPCO-94. Seventh European Signal Processing Conference*, pages 411–14, 1994.

[34] Yun Zhai and Mubarak Shah. Visual attention detection in video sequences using spatiotemporal cues. pages 815–824, 2006.

[35] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal Of Computer Vision*, 60(2):91–110, 2004.

[36] N. Ouerhani. *Visual Attention: From Bio-Inspired Modeling to Real-Time Implementation.* PhD thesis, University of Neuchatel, Switzerland, 2003.

[37] L. Itti. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(11):194–203, 2001.

[38] T. Jost N. Ouerhani, A. Bur and H. Hugli. Cue normalization schemes in saliency-based visual attention models. *International Cognitive Vision Workshop, Graz, Austria*, 2006.

[39] N. Ouerhani, A. Bur, and H. Hugli. Linear vs. nonlinear feature combination for saliency computation: A comparison with human vision. *Pattern Recognition, Proceedings*, 4174:314–323, 2006.

[40] F. Dufaux and M. Kunt. Multigrid block matching motion estimation with an adaptive local mesh refinement. *Visual Communications And Image Processing 92, Pts 1-3*, 1818:97–109, 1992.

[41] P. Anandan. A computational framework and an algorithm for the measurement of visual-motion. *International Journal Of Computer Vision*, 2(3):283–310, 1989.

[42] A. Singh. An estimation-theoretic framework for image-flow computation. *Proceedings. Third International Conference on Computer Vision (Cat. No.90CH2934-8)—Proceedings. Third International Conference on Computer Vision (Cat. No.90CH2934-8)*, pages 168–77—xv+759, 1990.

[43] M. H. Chan, Y. B. YU, and A. G. Constantinides. Variable size block matching motion compensation with applications to video coding. *Iee Proceedings-I Communications Speech And Vision*, 137(4):205–212, 1990.

[44] B. K. P. Horn and B. G. Schunk. Determining optical-flow - a retrospective. *Artificial Intelligence*, 59(1-2):81–87, 1993.

[45] H. H. Nagel. On the estimation of optical-flow - relations between different approaches and some new results. *Artificial Intelligence*, 33(3):299–324, 1987.

[46] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics*, 60(2):79–87, 1988.

[47] R. Eagleson and T. Caelli. Group-theoretic analysis of local flow characteristics while visually tracking a textured surface. *Proceedings of the 5th International Conference on Image Analysis and Processing. Progress in Image Analysis and Processing—Proceedings of the 5th International Conference on Image Analysis and Processing. Progress in Image Analysis and Processing*, pages 443–50—xiv+787, 1990.

[48] D. J. Fleet and A. D. Jepson. Computation of normal velocity from local phase information. *Proceedings CVPR '89 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.89CH2752-4)—Proceedings CVPR '89 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.89CH2752-4)*, pages 379–86—xvii+693, 1989.

[49] D. J. Heeger. Optical-flow using spatiotemporal filters. *International Journal Of Computer Vision*, 1(4):279–302, 1987.

[50] B. G. Haskell. Frame-to-frame coding of television pictures using 2-dimensional fourier-transforms. *Ieee Transactions On Information Theory*, 20(1):119–120, 1974.

[51] Y. C. Lin and S. C. Tai. Fast full-search block-matching algorithm for motion-compensated video compression. *Ieee Transactions On Communications*, 45(5):527–531, 1997.

[52] P. Moulin, R. Krishnamurthy, and J. W. Woods. Multiscale modeling and estimation of motion fields for video coding. *Ieee Transactions On Image Processing*, 6(12):1606–1620, 1997.

[53] H. G. Musmann, P. Pirsch, and H. J. Grallert. Advances in picture coding. *Proceedings Of The Ieee*, 73(4):523–548, 1985.

[54] M. Mattavelli, A. Nicoulin, and G. Fernandez. Overlapped motion compensation for sub-band coding of video sequences. *Signal Processing-Image Communication*, 8(2):149–160, 1996.

[55] S. Nogaki and M. Ohta. An overlapped block motion compensation for high-quality motion-picture coding. *1992 Ieee International Symposium On Circuits And Systems, Vols 1-6*, pages 184–187, 1992.

[56] K. Minoo and T. Nguyen. Reciprocal subpixel motion estimation: Video coding with limited hardware resources. *Ieee Transactions On Circuits And Systems For Video Technology*, 17(6):707–718, June 2007.

[57] R. Srinivasan and K. R. Rao. Predictive coding based on efficient motion estimation. *Ieee Transactions On Communications*, 33(8):888–896, 1985.

[58] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro. Motion-compensated interframe coding for video conferencing. *NTC '81. IEEE 1981 National Telecommunications Conference. Innovative Telecommunications - Key to the Future—NTC '81. IEEE 1981 National Telecommunications Conference. Innovative Telecommunications - Key to the Future*, pages G5.3/1–5, 1981.

[59] Trent J. Williams and Bruce A. Draper. An evaluation of motion in arti.cial selective attention. page 85, 2005.

[60] Perception and eye movement laboratory, departments of neurology and clinical research, university of bern, bern, switzerland, http://www.eyelab.dkf.unibe.ch/.

[61] A. Yarbus. Eye movements and vision. *Plenum Press*, 1967.

[62] http://www.smivision.com/en/eye-gaze-tracking-systems/products/iview-x-hi-speed.html.

[63] D. Parkhurst, K. Law, and E. Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision Research*, 42(1):107–123, January 2002.

[64] A. Treisman and S. Gormican. Feature analysis in early vision - evidence from search asymmetries. *Psychological Review*, 95(1):15–48, 1988.

[65] R. Carmi and L. Itti. The role of memory in guiding attention during natural vision. *Journal Of Vision*, 6(9):898–914, 2006.