Slovak University of Technology in Bratislava

Faculty of Informatics and Information Technologies

FIIT-5208-5737

Bc. Patrik Polatsek

SPATIOTEMPORAL SALIENCY MODEL OF HUMAN ATTENTION IN VIDEO SEQUENCES

Master thesis

Degree Course:	Information systems
Field of study:	9.2.6 Information systems
Place of development:	Institute of Informatics and Software Engineering, FIIT STU Bratislava
Supervisor:	Ing. Vanda Benešová, PhD.

May 2015

ANOTÁCIA

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
Študijný odbor: INFORMAČNÉ SYSTÉMY

Autor:	Bc. Patrik Polatsek
Diplomová práca:	Časovopriestorový model významných čŕt ľudskej pozornosti vo videosekvenciách
Vedúci diplomovej práce:	Ing. Vanda Benešová, PhD.

máj, 2015

Jedným z najdôležitejších zmyslov je náš zrak. Ľudské oči prijímajú každú sekundu obrovské množstvo vizuálnych informácií. Spracovanie takéhoto množstva dát je veľmi náročné, preto vizuálna pozornosť poskytuje nášmu mozgu schopnosť selekcie najdôležitejších aspektov okolia. Modely, ktoré predpovedajú vizuálnu pozornosť, vytvárajú mapu významných čŕt. Štandardné hierarchické metódy neberú do úvahy tvary objektov a modelujú významné črty ako rozdiel medzi stredom a jeho okolím po pixeloch.

Cieľom tejto diplomovej práce je zdokonaliť predikciu významných čŕt s využitím superpixelov. Ich hlavnou výhodou je, že ich hranice by mali odpovedať kontúram objektov. Náš navrhnutý model významných čŕt kombinuje hierarchické spracovanie vizuálnych príznakov so superpixelovou segmentáciou.

Naše vnímanie ovplyvňujú aj dynamické stimuly. Preto sme našu metódu rozšírili s použitím máp optického toku o dynamické vplyvy pozornosti, aby sme mohli predikovať vizuálnu pozornosť aj na videosekvenciách.

ANNOTATION

Slovak University of Technology in Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: INFORMATION SYSTEMS

Author:	Bc. Patrik Polatsek
Master Thesis:	Spatiotemporal saliency model of human attention in video sequences
Supervisor:	Ing. Vanda Benešová, PhD.
2015, May	

One of the most important senses is our vision. Human eyes receive a huge amount of visual information every second. Processing such amount of data is very demanding, thus visual attention provides our brain the ability to select the most important aspects of a scene. Models that predict visual attention create a saliency map. Standard hierarchical saliency methods do not respect the shape of objects and model saliency as the pixel-by-pixel difference between the centre and its surround.

The aim of this master thesis is to improve the saliency prediction using a superpixel-based approach. Their key benefit is that their boundaries should correspond to object's contours. Our proposed saliency model combines a hierarchical processing of visual features and a superpixel-based segmentation.

Our perception is influenced by dynamic stimuli too. Thus, we have extended our method to consider dynamic impacts of attention using optical flow maps to predict visual attention also in video sequences.

Declaration of Honour

I honestly declare, that I wrote this thesis independently under professional supervision of Ing. Vanda Benešová, PhD. with citated bibliography.

May, 2015 in Bratislava signature

Acknowledgement

I would like to thank my supervisor Ing. Vanda Benešová, PhD. for her willingness, continual support and helpful advice during the work on this thesis. I also thank my family and friends for their unceasing encouragement and support.

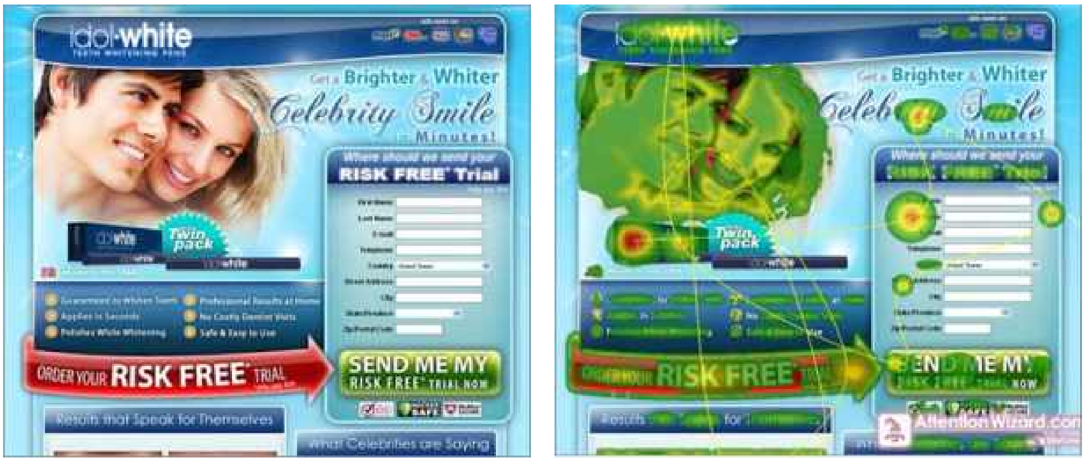
Contents

Chapter 1

Introduction

Our environment contains many objects which provide us huge amounts of visual information. The human brain, analogous to a computer, has limited computational capacities, due to which it cannot process all incoming visual data. Thus *attention* provides mechanisms of reducing and selecting important information (Mancas, 2007; Borji – Itti, 2013). In other words, attention optimises computations in the brain by selective concentrating on a single aspect of a scene whereas others are ignored[1].

Various elements of the environment compete for our attention. Visual attention helps us decide where to move and fix the eyes by detecting salient regions (Goldstein, 2010). Eye *fixations* of various subjects for a given image can be visualised by a *heat map*, where the most attentive regions are denoted with red colour (Figure 1.1).


idol white
Get a Brighter & Whiter
Celebrity Smile
in Minutes!
RISK FREE Trial
Twin pack
ORDER YOUR RISK FREE TRIAL
SEND ME MY RISK FREE TRIAL NOW
Results that Speak for Themselves
What Celebrities are Saying
AttentionWizard.com

(a) Original image.

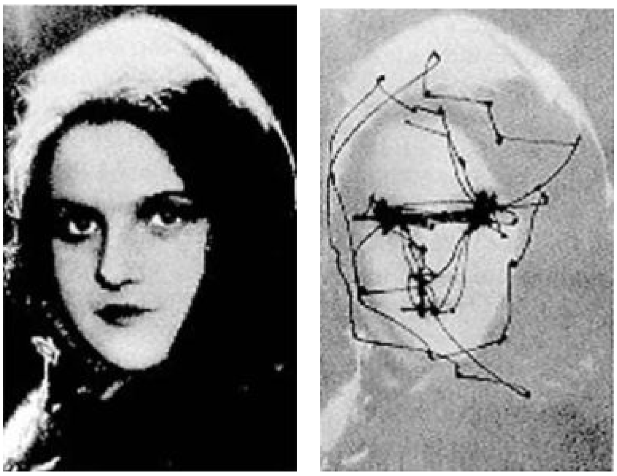
(b) Heat map.

Figure 1.1: Heat map is a visualisation of subjects' eye fixations. Red colour represents the most attentive regions of a given image[2].

Between the fixations our eyes perform very quick *saccadic* movements (about 3 movements per second) (Goldstein, 2010). Figure 1.2 represents an example of a saccadic trace.

[1] WANG, W. Visual Attention: What Attract You?, Institute of Digital Media, Peking University. Available from: `http://www.math.pku.edu.cn/teachers/yaoy/math112230/Lecture22_WangW_VisualAttention.pptx`.

[2] `https://www.attentionwizard.com/`



(a) Original image. (b) Trace of saccades.

Figure 1.2: Trace of saccades of the human eye during an eye-tracking experiment (Yarbus, 1967).

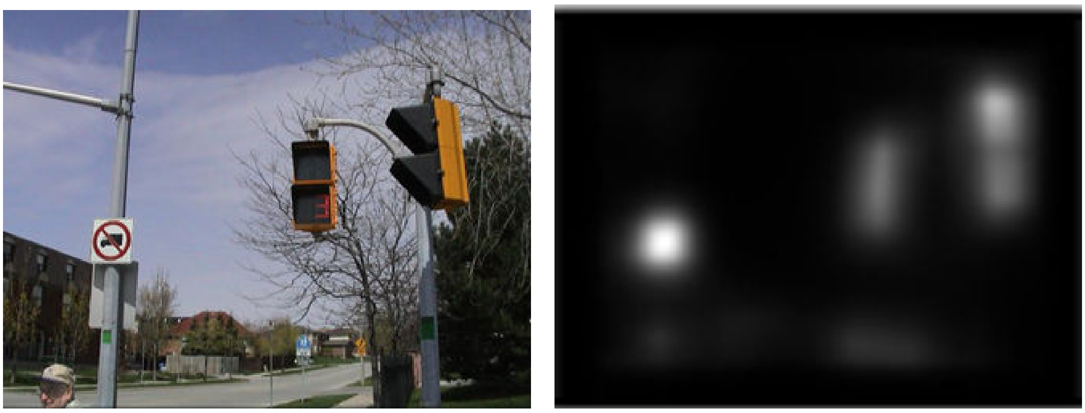
Figure 1.3: William James, the father of psychology (1842 – 1910)[3].

Despite what the father of psychology, *William James* (Figure 1.3), said in his book *Principles of Psychology* in 1890 (James, 1918): "Everyone knows what attention is", attention modelling is very difficult. Scientists have tried to create models of visual attention almost 25 years. The result of the models is a probability map of visual conspicuousness, so-called *saliency map* (Borji – Itti, 2013) which predicts the position of regions that visually stand out from their neighbourhood (Figure 1.4).

1.1 Motivation

A majority of computational models of visual attention have been estimated on static imagery. However, visual information obtained from the environment changes constantly, due to which the modelling based just on static stimuli such as colour, contrast and orientation, is not sufficient. Our aim is to extend a spatial attentional model with temporal information. Resulting *spatiotemporal model* will include both, static as well as dynamic aspects of atten-

[3] `http://psychology.about.com/od/profilesofmajorthinkers/p/jamesbio.htm`



(a) Input image.

(b) Saliency map.

Figure 1.4: Saliency map determines salient regions of a given image. Light parts of the map represent the regions which probably attract our attention, whereas dark ones are unimportant[1].

tion. Using the information from previous images it will model visual attention with better results.

Visual attention includes the research in the following areas of study (Borji – Itti, 2013):

- psychology,
- neurophysiology,
- computer vision (computational modelling of visual attention).

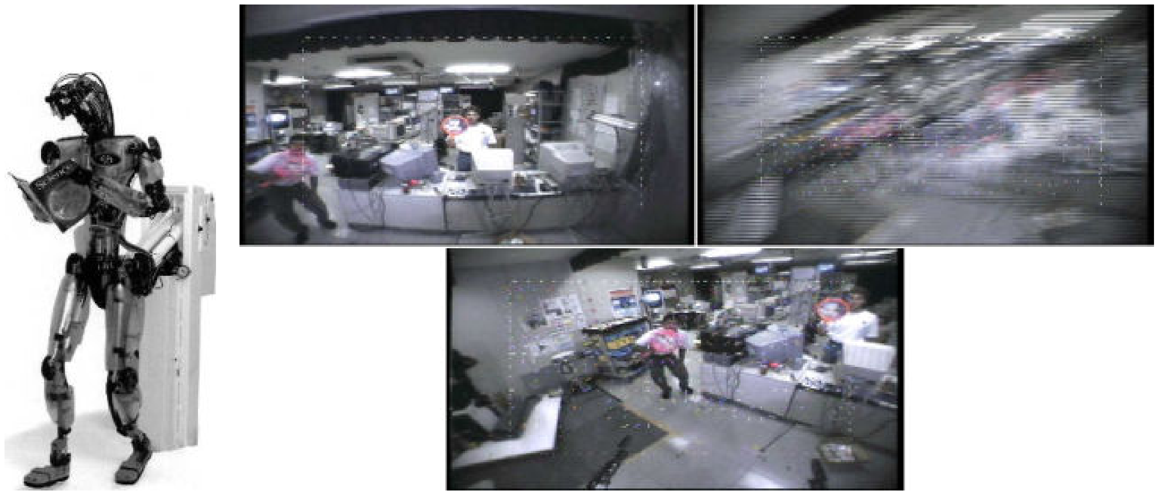
Visual attention modelling has a wide range of applications.

Detecting salient objects can be applied in **robotics**. Performing tasks in an environment requires the ability of selective *focusing on relevant objects* and separation the background. Due to that the saliency extraction module is implemented in robots. An example of such application is presented in (Scheier – Egner, 1997). Using saliency map robots may obtain an active vision and shift their view on important parts of an environment (Vijayakumar et al., 2001) (Figure 1.5). Salient objects can be used in robots as *localisation cues* in order to orient and navigate themselves in the space in cases when GPS navigation may not be applicable (Siagian – Itti, 2009).

Another field of usage is represented by **surveillance systems**. The SEARISE project called *Smart Eyes* (Figure 1.6) is an active camera system that is able to track and zoom in on salient objects with active binocular cameras (Endres et al., 2011). Searching for the security relevant events is reduced to search salient objects only. Thus the camera system may analyse an input stream in real-time[4].

Saliency maps have also many different applications in computer vision and graphics in **image and video processing**. (Stentiford, 2007) proposes a method of auto *image cropping* using salient region detection (Figure 1.7(a)). (Marchesotti et al., 2009) uses saliency detection for automatic *image thumbnailing* (Figure 1.7(b)). Visual saliency is applied in *context-aware image resizing* (Figure 1.7(c)) in (Achanta – Susstrunk, 2009). Resizing image ratios may deform objects. Saliency map shows the prominent regions which ratios should be preserved. Detecting salient parts of an image is also used in *image retargeting* of large size

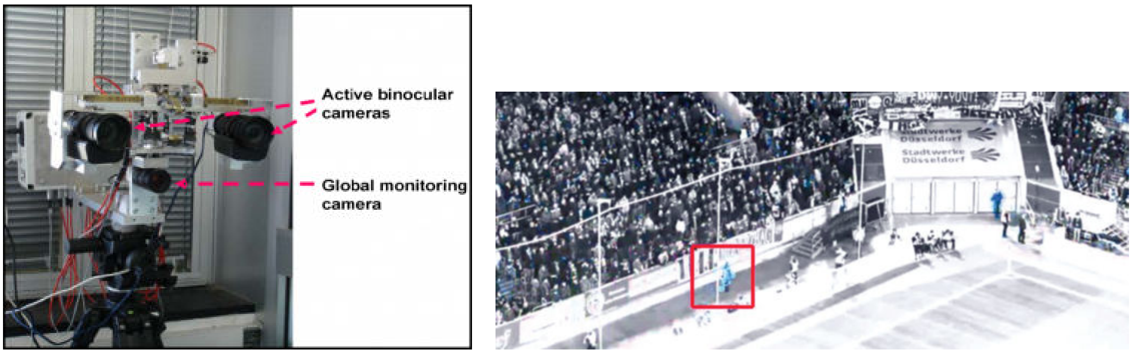
[4] http://www.fit.fraunhofer.de/en/fb/life/projects/searise.html



(a) Humanoid robot.

(b) Peripheral view before, after and during a saccade.

Figure 1.5: A peripheral view of a humanoid robot. Red circles represent the robot's actual focusing (Vijayakumar et al., 2001).



(a) Smart Eyes.

(b) Camera system is focused on a salient object in a red box.

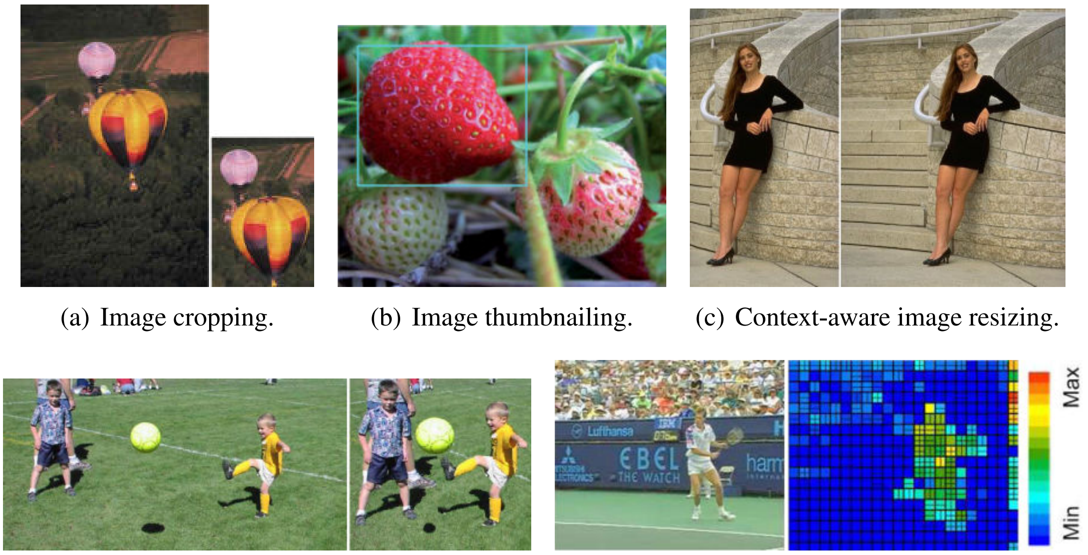
Figure 1.6: Smart Eyes[5], an active cognitive visual system, consists of a fixed camera for global monitoring and two active binocular cameras that focus on salient objects[6].

images to small size (Setlur et al., 2005). The proposed method preserves important objects by eliminating gaps among them (Figure 1.7(d)). *Images* and *videos* can be effectively *compressed* according to visual saliency (Figure 1.7(e)). Salient regions are stored at higher resolution than unimportant parts (Itti, 2004; Le Meur et al.). (Jacobson – Nguyen, 2011) presents a saliency application in *frame rate-up conversion* (FRUC). Presented implementation of FRUC is useful in low frame rate videos. Using the algorithm, motion blur is reduced for salient regions.

Visual attention models can be useful in **medical imaging** (Le Callet – Niebur, 2013). Understanding visual attention by reading medical images can automate pathology detection and localisation process.

[5] BRUCE, N. Saliency: Applications in Vision, Image, Processing and Computer Graphics. CVPR 2013 Tutorial: A Crash Course on Visual Saliency Modeling: Behavioral Findings and Computational Models. Available from: `http://ilab.usc.edu/borji/cvpr2013/SalApplicationsF.pdf`.

[6] Smart Eyes: Detection of Salient Events, Fraunhofer Institute for Applied Information Technology. Available from: `http://psychology.about.com/od/profilesofmajorthinkers/p/jamesbio.htm`.

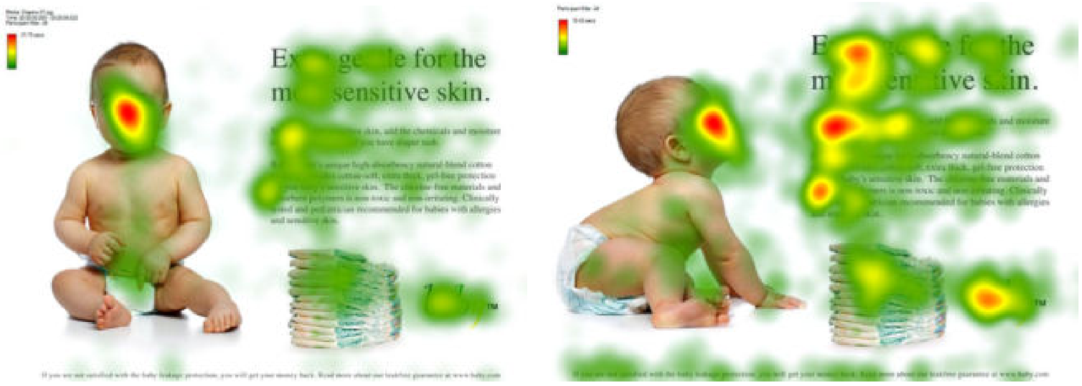


(a) Image cropping. (b) Image thumbnailing. (c) Context-aware image resizing.

(d) Image retargeting. (e) Image and video compression.

Figure 1.7: Applications of visual attention in image and video processing (Stentiford, 2007; Marchesotti et al., 2009; Achanta – Susstrunk, 2009; Setlur et al., 2005; Le Meur et al.).

Advertisement and **design** can better conform to customers with the help of visual attention model as shown in Figure 1.8[7].



(a) The most attentive part is a baby's face. (b) The most attentive part is the place where a baby is looking at.

Figure 1.8: Changing the baby's gaze influences reader's visual attention for the advertisement[7].

1.2 Requirements

We decided to create a spatiotemporal model of human visual attention. The model predicts visual attention for input images or video sequences. It creates saliency maps using static and temporal stimuli of attention and estimates the most probable candidates of eye fixations. Our model compares eye fixations data with our prediction.

[7] http://www.o-psani.cz/2011/06/co-upouta-pozornost-ctenare.html

Our attentional model has to satisfy the following requirements:

- identify static stimuli of visual attention of a scene,
- identify static and dynamic effects of visual attention for a video sequence,
- create a spatiotemporal saliency map,
- predict salient regions and positions of eye fixations for image and video datasets.

Chapter 2

Attention and Scene Perception

The human brain receives from the environment the large amount of sensorial data every second. It is physically impossible to pay attention to all stimuli at once. *Attention* is a set of selection mechanisms in the brain which enables to select specific aspects of the environment. The primary aim of attention of all living beings is to alert of impending danger and help survive (Mancas, 2007; Wolfe, 2009).

The main features of attention are:

- *Selection*: We focus on several aspects of the environment while ignoring others.
- *Limitation*: The rate of sensorial information processing of the brain is limited.

In this chapter, we focus on the basic principles of human *visual attention* that selects incoming visual data.

2.1 Visual System

Vision is our most important sense. *Human visual system* (HVS), which processes visual information, consists of various subsystems for identifying contrast, shape, depth, colour and other visual properties (Dobeš, 2005). Information obtained from left and right visual field is led into the visual cortex (Figure 2.1).

Visual data processing starts when light enters the eye through the small hole in the *iris* called the *pupil* (Feldman, 2012; Goldstein, 2010; Ciccarelli – White, 2008). The iris can adjust the amount of incoming light by changing the pupil size. The eyeball is covered by the *cornea* and reflects the light.

The lens behind the pupil focuses the light into a single point on the *retina*, so-called *fovea*. The retina contains light-sensitive photoreceptors called *cones* and *rods*. Cones concentrated mainly on fovea are responsible for colour vision and sensitive for details, whereas rods are more sensitive to light and enable us to see under low light conditions.

Figure 2.2 represents the structure of the human eye.

When light reaches the retina, it is converted into electrical signals within the *ganglion neurons* and sent through the *optic nerve* into the brain.

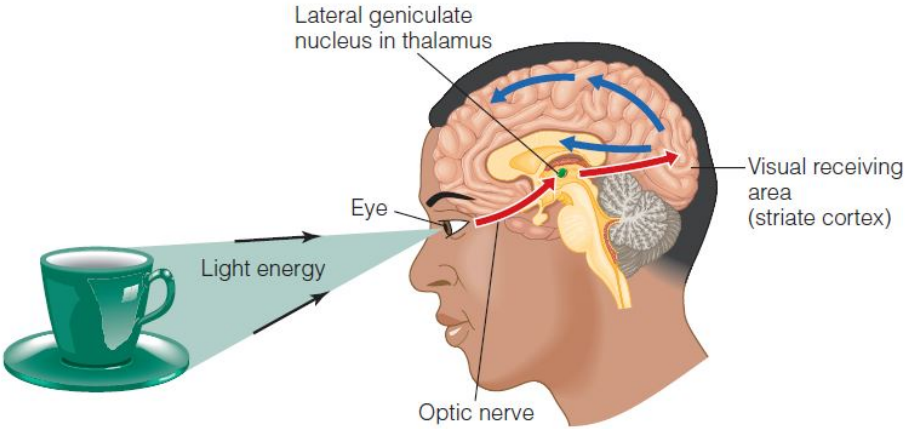

Lateral geniculate nucleus in thalamus
Eye
Light energy
Visual receiving area (striate cortex)
Optic nerve

Figure 2.1: Overview of human visual system (Goldstein, 2010).

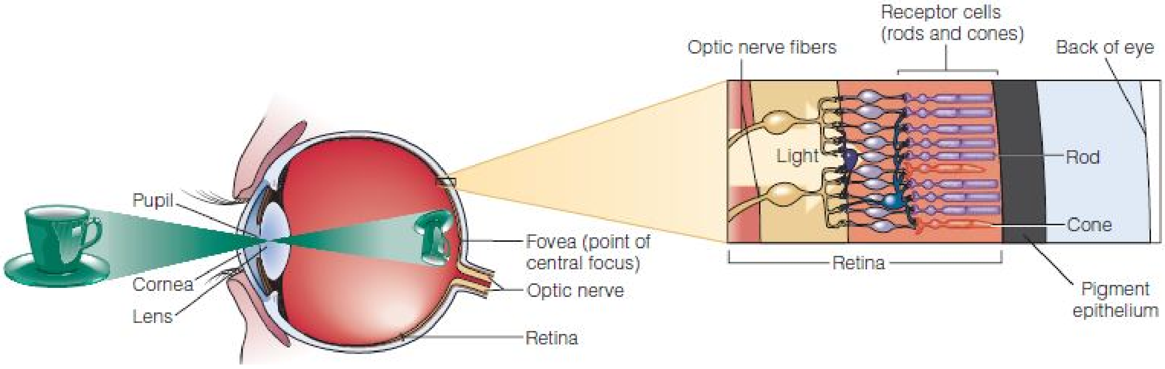


Figure 2.2: Structure of the human eye (Goldstein, 2010).

The receptive field is an area that influences of a neuron. The structure of retinal ganglion receptive fields is called *centre-surround*. A group of ganglions which are excited when light hits their centre and inhibited when it hits the surroundings are called *on-centre* cells. On the other hand, *off-centre* cells have the excitatory and inhibitory in reverse order (Dobeš, 2005).

Left and right optic nerve cross at a place called the *optic chiasm* (Figure 2.3). The signal from both optic nerves is further transmitted to the opposite side of the brain.

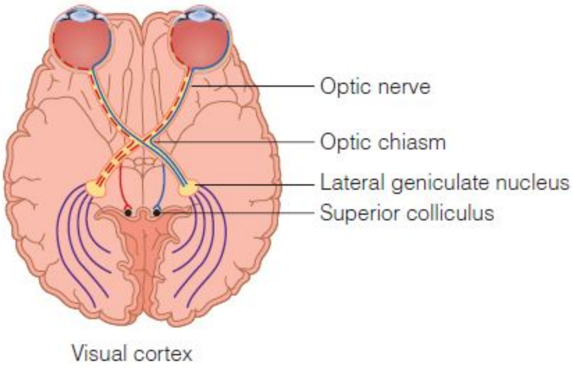


Figure 2.3: Structure of the human brain (Goldstein, 2010).

About 10% of the fibres of the optic nerve are received by a brain area called the *superior colliculus* (SC), which uses visual information to control eye movements. The primary aim of SC is to direct the eyes onto the important parts of the surroundings (Mancas, 2007).

Visual information passes through the *lateral geniculate nucleus* (LGN) in the thalamus. The LGN structure contains *M (magno)* and *P (parvo) cells.* Bigger M cells are sensitive to big objects and a quick change in stimulus, whereas smaller P cells are sensitive to colour and details (Dobeš, 2005).

The information travels to the *visual (striate) cortex*, mainly to the *primary visual cortex* called *V1* (Figure 2.4). Further visual processing is performed for example, in *V2* responsible for contour detection, *V4* sensitive to colour and responsible for shape detection and finally *MT (V5)* responsible for motion processing.

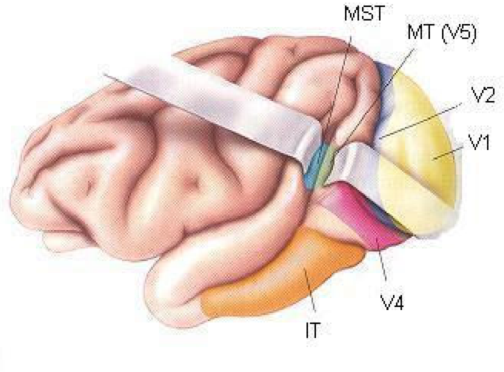


Figure 2.4: Structure of the macaque visual cortex (Dobeš, 2005).

2.2 Visual Attention

Visual attention is a process of selection visual information from the environment (Goldstein, 2008).

Attention may be *overt* and *covert.* Overt attention refers to the attention focus when the fovea is directed toward a stimulus. Covert attention is scanning a scene in the peripheral vision without any eye movement (Borji – Itti, 2013; Mancas, 2007).

Attention helps us to decide where to move our eyes and which parts of a scene should be deeper processed.

We distinguish different eye behaviours (Holmqvist et al., 2011; Mancas, 2007):

1. **Fixation**: The pause in a movement when eyes are fixated to the specific position and remain still is called fixation. During the fixation, visual information is taken from the environment. It lasts 200 – 300 msec. However, eyes are not completely still, but they perform micro-movements such as *tremor*, *microsaccades* and *drifts*.
2. **Saccade**: Saccade is a quick movement from one fixation to another. It is the fastest movement the body can produce. We make approximately 3 saccades per second that last only 30 – 80 msec. An example of a sequence of saccadic movements is represented by Figure 2.5.

3. **Smooth pursuit**: Our eyes perform a movement called smooth pursuit when we follow a moving object voluntary.

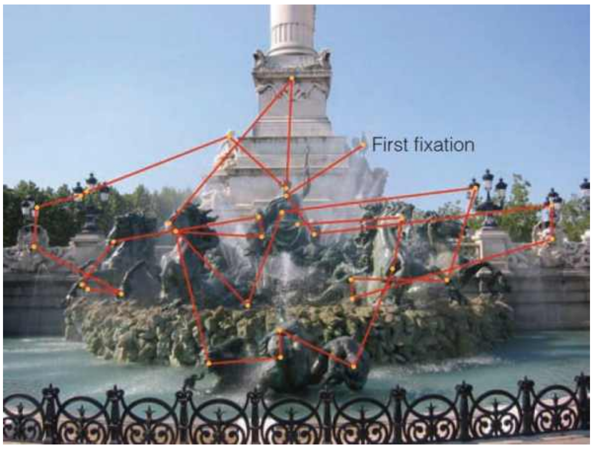


Figure 2.5: Sequence of saccadic movements. Yellow dots denote fixations and red lines represent saccadic paths (Goldstein, 2010).

There are several types of visual attention[1]. *Location-based* attention selects stimulus according their location. *Feature-based* attention is based on directing the gaze to a specific visual feature such as colour or movement. *Object-based* attention is focused on an object that is defined by multiple visual features at a specific location.

There are various factors that influence our attention. We can divide them into two main categories (Borji – Itti, 2013):

- stimulus-driven *bottom-up* factors
- and goal-driven *top-down* factors.

Perception is a process of assigning the meaning to the incoming information which occurs after bottom-up and top-down processing (Feldman, 2012). These two processing mechanisms do not work separately, but they interact with each other.

2.2.1 Bottom-up Attention

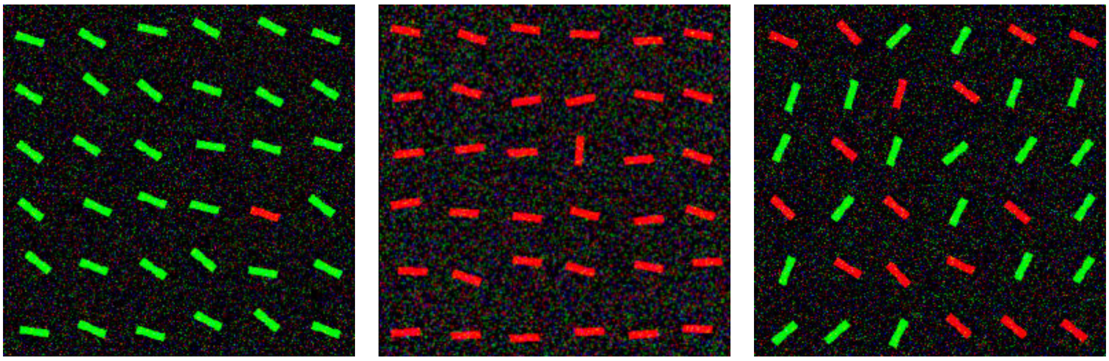
Attention driven *exogenously* is called *bottom-up* (stimulus-driven) attention. It is involuntary, rapid and unconscious attention based on visual characteristics of a scene which automatically draw our attention (Goldstein, 2008)[2].

Bottom-up attention is related to the term *saliency*. Saliency is the vividness of a stimulus which stands out relatively from its neighbours. Typical bottom-up features involve colour, contrast, orientation, texture and movement (Wolfe, 2009).

[1] WANG, W. Visual Attention: What Attract You?, Institute of Digital Media, Peking University. Available from: `http://www.math.pku.edu.cn/teachers/yaoy/math112230/Lecture22_WangW_VisualAttention.pptx`.

[2] LE MEUR, O. Selective visual attention: from experiments to computational models, Psychologie de la cognition, University of Paris. Available from: `http://people.irisa.fr/Olivier.Le_Meur/teaching/visualattention.pdf`.

Figure 2.6 presents examples of visual saliency[3]. The most salient item in Figure 2.6(a) is the red bar due to its unique colour. Different orientation determines the vertical bar in Figure 2.6(b) as the most salient object. In contrast, an object with the highest saliency in Figure 2.6(c) does not stand out due to its single unique visual feature. Its saliency is much lower than previous salient objects, due to which it is searched after scanning the image. The most salient object, the only red and vertical bar, is unique by the combination of two features.



(a) Unique colour.

(b) Unique orientation.

(c) Unique combination of colour and orientation.

Figure 2.6: Examples of visual saliency[3].

Perception starts with bottom-up processing based on incoming visual data which are sent to the brain for interpretation (Goldstein, 2010). The mechanism analyses individual stimuli and combines them to objects to build up a complete perception (King, 2010; Ciccarelli – White, 2008).

2.2.2 Top-down Attention

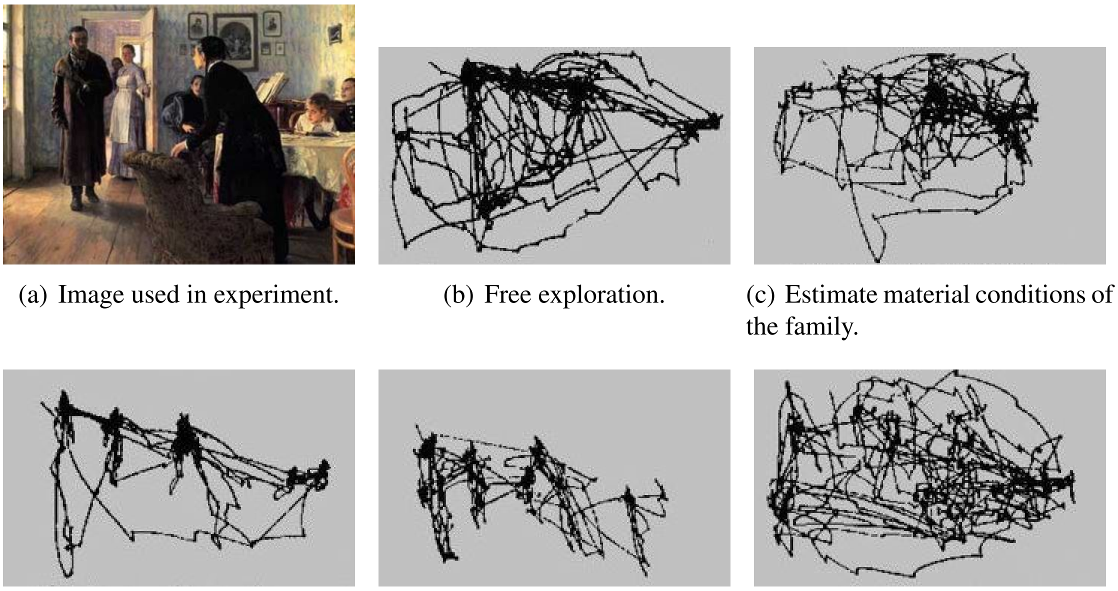
Top-down (goal-driven) attention driven *endogenously* is guided by prior knowledge, experiences, expectations, tasks or goals. Contrary to the bottom-up approach, top-down attention related to our memory is much slower, voluntary and conscious (Goldstein, 2008)[2].

Top-down factors influence attention and modify scanning of a scene. A Russian psychologist *Alfred Lukyanovich Yarbus* did a research how tasks and questions about an image change eye motion pattern (Yarbus, 1967). The results of the research are shown in Figure 2.7. First, observers view an image (Figure 2.7(a)) in task-independent way (Figure 2.7(b)). Figures 2.7(c), 2.7(d), 2.7(e) and 2.7(f) present the dependency between attention and a given task – each task strongly influences a trace of observer's saccades.

Top-down processing modifies the perception using prior knowledge, expectations or our needs and adds sense to incoming information. It organises individual features processed by bottom-up attention to a coherent whole and fill in missing information from our memory (Goldstein, 2010; King, 2010; Ciccarelli – White, 2008).

Cooperation of bottom-up and top-down processing is shown in Figure 2.8. First, in bottom-up processing basic visual features of incoming data are analysed and then after top-down

[3] ITTI, L. Visual salience. Scholarpedia, 2(9):3327, 2007. Available from: `http://www.scholarpedia.org/article/Visual_salience`.



(a) Image used in experiment.

(b) Free exploration.

(c) Estimate material conditions of the family.

(d) Estimate family age.

(e) Remember the family members' clothes.

(f) Remember the positions of family members and objects.

Figure 2.7: Eye tracking experiment. Observers view a given image within various tasks (Yarbus, 1967).

processing using a previous knowledge a moth on a tree is identified.

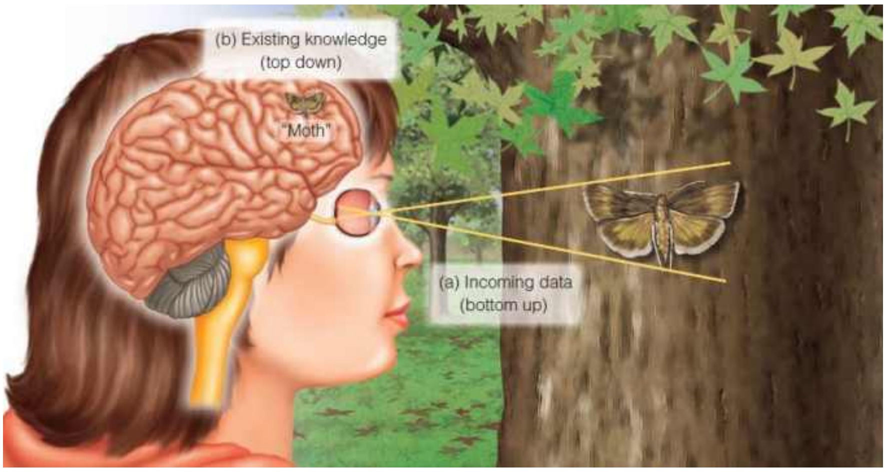


Figure 2.8: Perception involves bottom-up and top-down processing. Within bottom-up processing basic features of incoming data are analysed. After top-down processing a moth on a tree is recognised using a prior knowledge (Goldstein, 2010).

2.3 Motion Perception

Since we live in a dynamic world, objects and observers can move, motion processing plays a key role in visual attention. The ability of motion processing is essential for survival of many living organisms. Motion analysis helps us to understand complex actions that constantly happen around us.

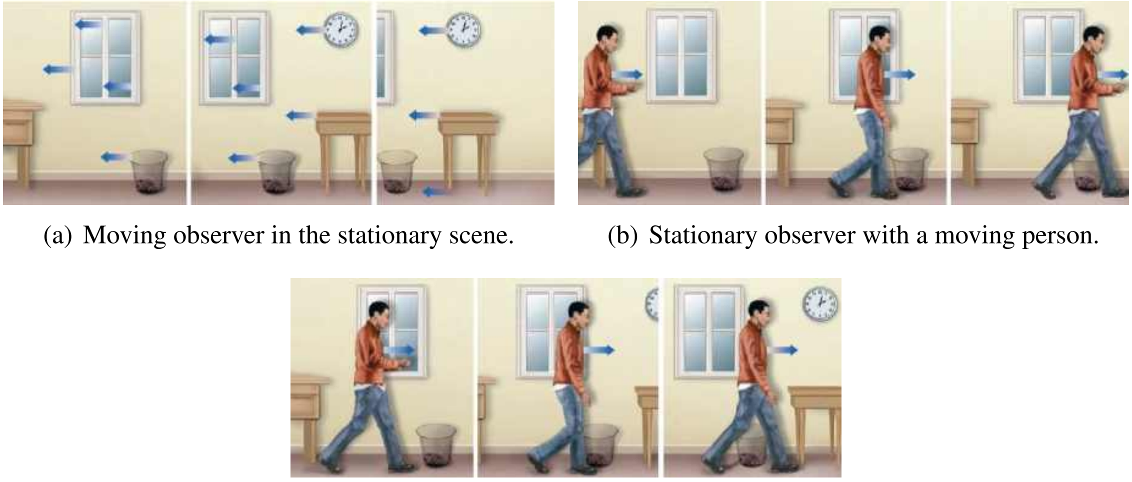
Stimuli with motion effects can be divided into the following groups:

1. **real motion** of moving objects such as walking people and driving cars,
2. **illusory motion** of objects creating the illusion of motion that do not really move:
   (a) *apparent motion* of stationary objects flashing at slightly different locations that create an illusion of a single moving object, for example in cinema projection,
   (b) *induced motion* of stationary objects in the surroundings of a moving object,
   (c) *motion aftereffect* of a stationary image after viewing a moving object for a time, for example the spiral aftereffect or the waterfall illusion.

According to Gibson's approach described in (Gibson, 1950), information about the movement in the environment is provided by the *optic array*. It is the structured pattern of light formed from textures, surfaces and contours of objects. A movement in a scene results in a disturbance of the optic array called *optical flow*. Optical flow may be defined as the movement of elements relative to an observer.

A movement that an observer perceives may occur in the following situations (Goldstein, 2010):

1. an observer is moving (Figure 2.9(a)),
2. an observer is stationary and an object in a scene is moving (Figure 2.9(b)),
3. a moving object is followed by observer's eyes (Figure 2.9(c)).



(a) Moving observer in the stationary scene.

(b) Stationary observer with a moving person.

(c) Moving person followed by observer's eyes.

Figure 2.9: Motion perception through the changes of optic array. A moving observer causes a global optical flow and a moving object in a scene results in a local optical flow. (Goldstein, 2010)

The presence of a *global optical flow* indicates that an observer is moving (Figure 2.9(a)). The position where an observer directs the gaze is the only place without the flow called the *focus of extension*. Optical flow is *gradual* – the magnitude of the flow increases from the focus towards the observer. The direction of optical flow is *opposite* to the observer's movement. If an observer moves to a target point, the flow comes from the target. On the other hand, if an observer moves away from the target, the flow directs to the point (Figure 2.10) (Goldstein, 2010).

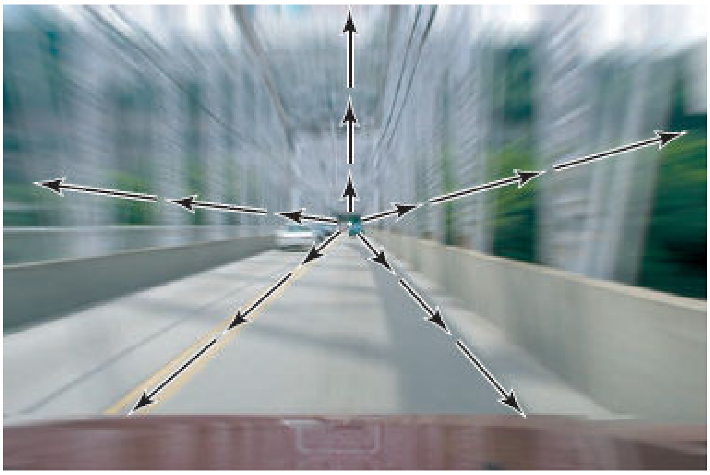
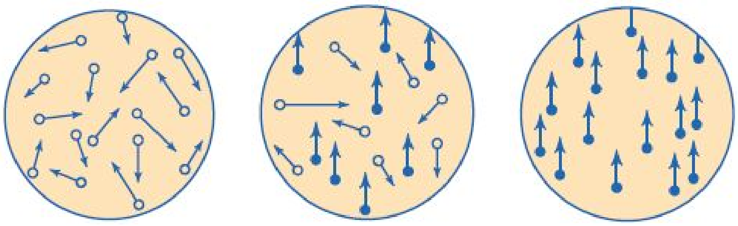


Figure 2.10: Optic array of a scene seen through a front window of a moving car. Global optical flow indicates that an observer is moving. The flow directs and increases towards the observer. A white dot represents the focus of extension (Goldstein, 2010).

Figure 2.9(b) and Figure 2.9(c) represent situations when a moving object is observed. The movement relative to the scene produces a *local optical flow*.

(Newsome et al., 1989) presents a research of motion perception conducted on monkeys. Trained monkeys reported the direction of motion in a moving dots display. In the display, a coherence of dots, representing the degree of movement in the same direction, was varying (Figure 2.11). The goal of this study was to determine the relationship between the report of the direction of motion and the response of MT neurons in the cortex. As the correlation of moving dots increased, the correctness of judgment about the direction increased and the neurons fired more rapidly.

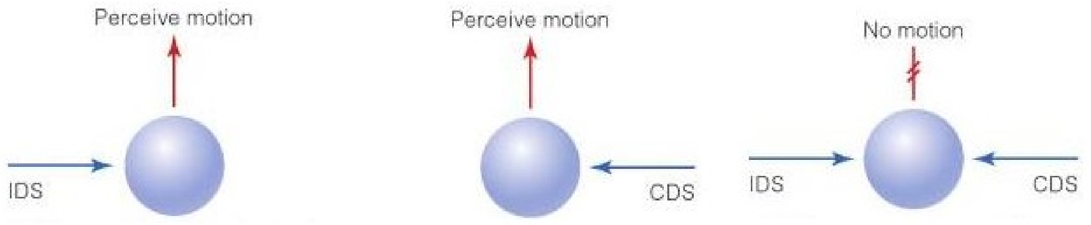


(a) 0% coherence. (b) 50% coherence. (c) 100% coherence.

Figure 2.11: Moving dots displays. In higher correlated displays MT neurons fired more quickly and the report of the direction of motion was more accurately (Goldstein, 2010).

An explanation of motion perception during an eye movement provides the **corollary discharge theory** described in (Von Holst, 1954). Eye muscles are controlled by *motor signals* (MS) sent from the brain to move the eyes. Subsequently, the command is copied to the *corollary discharge signal* (CDS) and transmitted to the brain for the further comparison with another neural signal called the *image displacement signal* (IDS). IDS is generated by the retinal receptors when an image is moved across the retina. Motion perception occurs if just one type of neural signal is received.

In case of stationary eyes (Figure 2.9(b)), an object moved across the retina activates only an IDS (Figure 2.12(a)). If eyes follow a moving stimulus (Figure 2.9(c)), only a CDS



(a) Stationary eyes and a moving object.

(b) Moving eyes and a moving object.

(c) Moving eyes and a stationary scene.

Figure 2.12: IDS and CDS are compared to perceive motion according to the corollary discharge theory. Motion perception occurs when only one type is present (Goldstein, 2010).

is received as a copy of motor signals (Figure 2.12(b)). CDS and IDS are simultaneously transmitted to the brain (Figure 2.12(c)), when a stationary scene is scanned by an observer (Figure 2.9(a)). In this situation, signals cancel each other and no motion is perceived.

Chapter 3

Related Work

In recent decades scientists have studied mechanisms of human attention to determine regions of interest from the huge amount of visual information. In general, there are two ways of selecting the regions which attract visual attention (Mancas, 2007):

1. *Measuring attention*: track eye movements (Figure 3.1), investigate brain activity and study human behaviour.
2. *Computing attention*: create an algorithm that predicts salient regions in images or video sequences.

Figure 3.1: Mobile eye tracker by Tobii[1].

In this chapter we focus on the state-of-the-art in visual attention modelling.

There are many factors for division of attention models (Borji – Itti, 2013).

According to the type of processing, we divide the models on *bottom-up*, *top-down* and those that combine both processes. The majority of them models bottom-up attention. The result of such models is a *saliency map* (Figure 3.2) which is a topographic representation of visual saliency of a scene.

Most attention models use only *spatial* visual information to create a saliency map. Dynamics and constant changes of a real-world environment indicate the requirement to model *spatiotemporal* effects of attention. In order to append temporal information to attention models and predict the attention from videos, we can use dynamic stimuli such as motion contrast or implement learning processes in attention.

According to the attention type, models may be *feature-based*, *space-based* or *object-based*.

[1] http://www.tobii.com/en/eye-tracking-research/global/products/hardware/tobii-glasses-eye-tracker

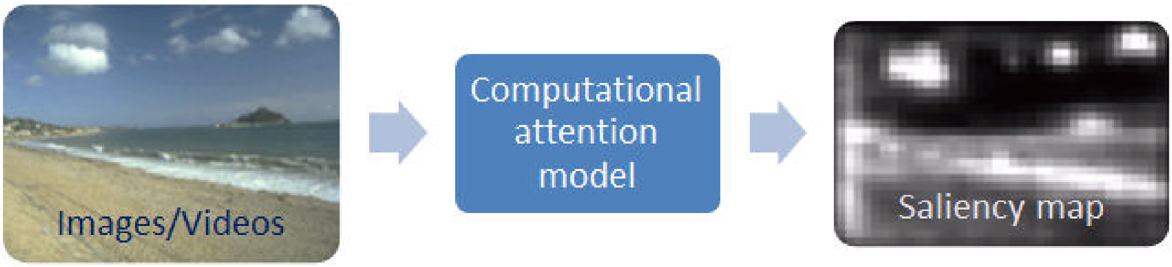


Figure 3.2: General process of bottom-up attention model[2].

The major categories of attention models together with their essential description are listed in Table 3.1 (Borji – Itti, 2013; Filipe – Alexandre, 2013).

Table 3.1: Overview of attention models.

Model	**Description**
hierarchical (cognitive)	hierarchical decomposing of features
Bayesian	combination of saliency and prior knowledge
decision theoretic	discriminant saliency theory
information theoretic	maximisation of information from a given environment
graphical	graph-based computations of saliency
spectral analysis	saliency computation in the frequency domain
pattern classification	machine learning from salient patterns
reinforcement learning	maximisation of a gained cumulative reward

3.1 Hierarchical Models

Hierarchical (cognitive) *models* are biologically inspired models based on *hierarchical decomposition* of visual features using Gaussian, Fourier or wavelet decomposition (Le Meur – Le Callet, 2009).

These models are inspired by *Feature Integration Theory* (FIT) (Treisman – Gelade, 1980) which presents visual information as a set of individual feature maps. In early and parallel pre-attentive processing a scene is analysed to identify individual features. Within the second, focused attention phase various features are combined and integrated to perceive whole objects. According to the theory, attention is responsible for the object perception instead of the perception of individual features.

FIT became a basis of the first theoretical biologically based attention model presented in (Koch – Ullman, 1985) in 1985 (Figure 3.3). First, elementary features such as colour, orientation and direction are extracted in parallel in order to create multidimensional topographical *feature maps* at different scales, called the *early representation*. Locations that differ mostly from their neighbourhoods are considered as the most conspicuous. Feature maps are finally combined and fused into a single *saliency map*. In order to determine the most salient location in a visual scene, a so-called *Winner-Take-All* (WTA) neural network is used. The properties of the winning location are transferred into the *central representation*.

[2] NIEBUR, E. Saliency map. Scholarpedia, 2(8):2675, 2007. Available from: `http://www.scholarpedia.org/article/Saliency_map`.

The WTA mechanism transforms input units of the saliency map by the following equation:

$$y_i = \begin{cases} 0 & \text{if } x_i < \max_j(x_j), \\ f(x_i) & \text{if } x_i = \max_j(x_j), \end{cases} \tag{3.1}$$

where x_i is an input unit of the saliency map at the location i, y_i is an output unit and f is the function which maps an input to a non-zero output.

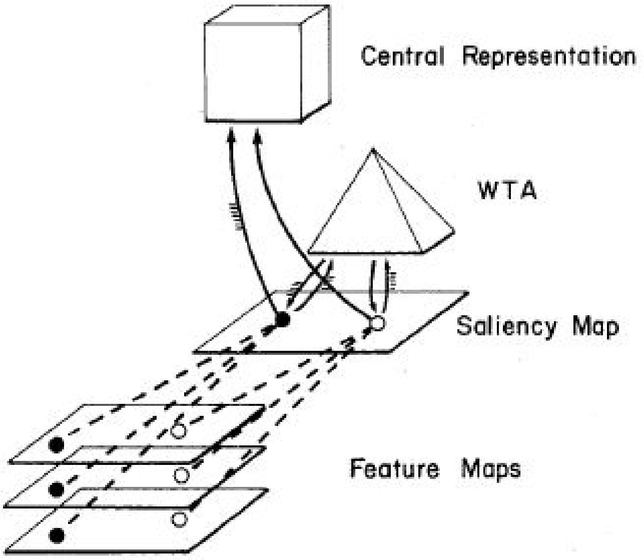


Figure 3.3: General overview of the attention model (Koch – Ullman, 1985).

The WTA mechanism shifts to the next most salient location by inhibiting the winning location. This process is called the *inhibition of return*. Shifting to the next location is influenced by the *proximity* and the *similarity* to the previous location.

One of the most known bottom-up saliency model based on the previous model is presented in (Itti et al., 1998). Figure 3.4 illustrates the architecture of the proposed model.

This model extracts the following visual features: *colour*, *intensity* and *orientation*. After the creation of the intensity image defined as $I = (r + g + b)/3$, red (r), green (g) and blue (b) colour channels are normalised by I.

The model creates *Gaussian pyramids* for *red* $R(\sigma)$, *green* $G(\sigma)$, *blue* $B(\sigma)$, *yellow* $Y(\sigma)$, *intensity* $I(\sigma)$ channel and *local orientations* $O(\sigma, \theta)$, where a pyramid level σ ranges from 0 to 8 and an orientation $\theta \in \{0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}\}$.

Colour channels of each pixel in the pyramid are defined by the following forms:

$$R = r - (g + b)/2, \tag{3.2}$$

$$G = g - (r + b)/2, \tag{3.3}$$

$$B = b - (r + g)/2, \tag{3.4}$$

$$Y = (r + g)/2 - |r - g|/2 - b. \tag{3.5}$$

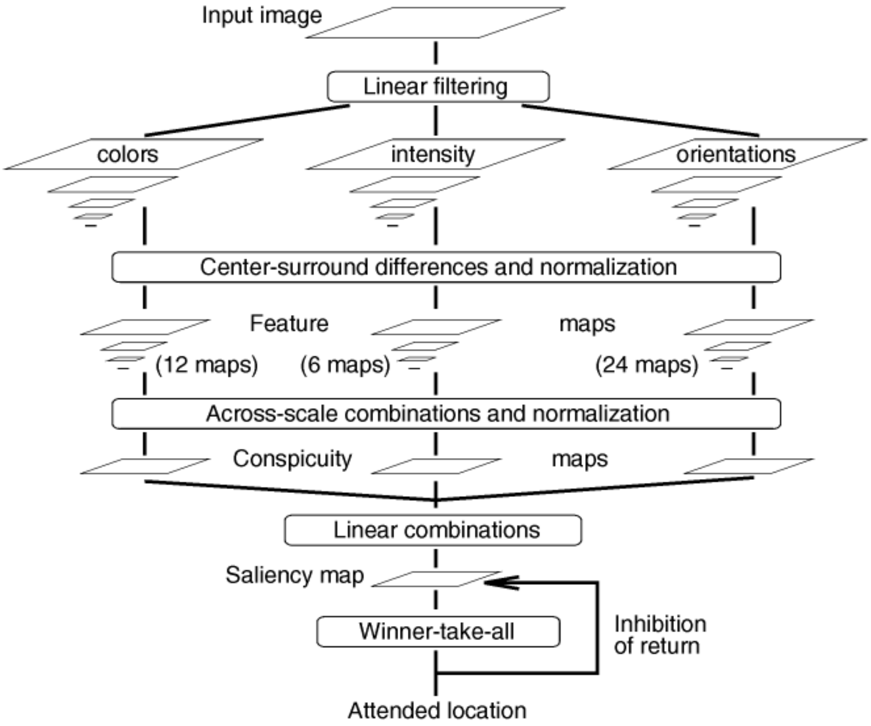


Figure 3.4: The architecture of Itti's attention model (Itti et al., 1998).

As mentioned in Section 2.1 the structure of ganglion neurons is characterised by *center-surround*. This model achieves center-surround operations as the difference between finer scales of *Gaussian pyramid* representing the center and coarser scales representing the surroundings. Gaussian pyramid is a set of images at multiple scales, where an input image is iteratively downsampled and smoothed by Gaussian filter (Bradski – Kaehler, 2008). Scales at the center c ranges from value 2 to 4. Scales in the surround are defined as $s = c + \delta$, where $\delta \in \{3, 4\}$. Difference of Gaussian is implemented by interpolation and point-to-point subtraction.

Using center-surround operations denoted as Θ, visual features are computed. The model creates totally 42 different *feature maps*: 6 maps for intensity, 12 for colours and 24 for orientation.

In order to determine the intensity contrast, intensity maps are computed by the equation:

$$I(c, s) = I(c) \ominus I(s). \tag{3.6}$$

According to the *Opponent-Process Theory of Colour Vision* (Hering, 1920), human colour vision is a response of opponent colour channels. Colour stimuli are recombined and the colour perception is a result of two opponent-colour mechanisms: *red-green* (RG) and *blue-yellow* (BY). This algorithm models the colour opponency by the following colour maps:

$$RG(c, s) = |(R(c) - G(c)) \ominus (G(s) - R(s))|, \tag{3.7}$$

$$BY(c, s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))|. \tag{3.8}$$

In order to obtain the characteristics of texture and local orientation, the image is filtered using a linear 2D *Gabor kernel*. The filter is created by multiplying Gaussian function with sinusoid at different orientations θ. Image convolved with this kernel is a base of a pyramid O which is used to obtain a last conspicuous map:

$$O(c, s, \theta) = |O(c, \theta) \ominus O(s, \theta)|. \tag{3.9}$$

Each computed map is normalised and multiplied by $(M - \overline{m})^2$, where M is a global maximum and $\overline{m}$ is an average of local maxima.

The next step consists of across-scale combination of all feature maps into 3 *conspicuous maps* for intensity $\overline{I}$, colour $\overline{C}$ and orientation $\overline{O}$ which are normalised again.

Finally, these conspicuous maps are fused into a single *saliency map* S:

$$S = \frac{1}{3}\left(N(\overline{I}) + N(\overline{C}) + N(\overline{O})\right), \tag{3.10}$$

where N represents the map normalisation.

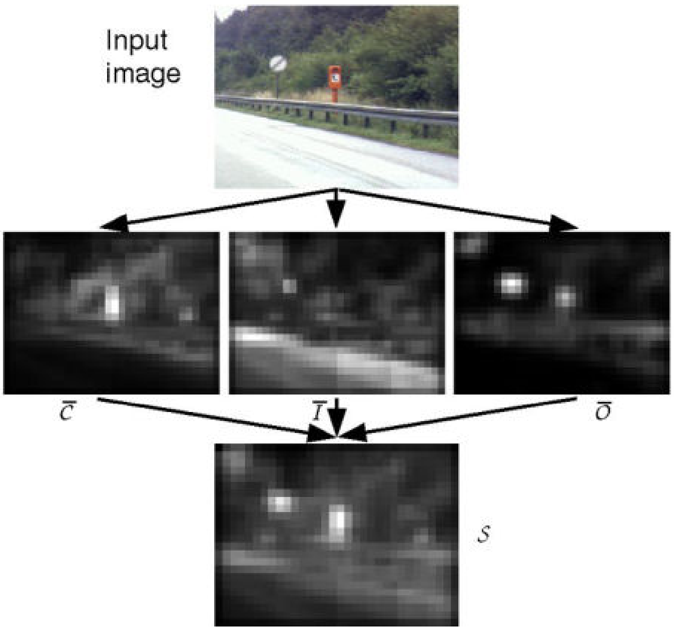


Figure 3.5: Input image is decomposed into several feature maps fused into 3 conspicuous maps for intensity, colour and orientation. The maps are finally combined into a single saliency map (Itti et al., 1998).

The most salient location is determined by the WTA network and shifts to the next salient locations are performed using the inhibition of return described in (Koch – Ullman, 1985).

A spatiotemporal extension of (Itti et al., 1998) is defined in (Itti et al., 2004). Dynamic changes between consequent video frames are expressed in two additional types of Gaussian pyramids.

The first dynamic pyramid called *flicker* (F_t) is the absolute difference between the intensity of the current frame and the previous frame.

Next, the *motion* Gaussian pyramid is computed from spatially-shifted differences of orientation pyramid layers between consequent frames $t-1$ and t as follows:

$$R_t(\sigma,\theta) = |O_t(\sigma,\theta) * S_{t-1}(\sigma,\theta) - O_{t-1}(\sigma,\theta) * S_t(\sigma,\theta)|\,, \tag{3.11}$$

where $S_t(\sigma,\theta)$ is $O_t(\sigma,\theta)$ shifted by one pixel in the direction orthogonal to the Gabor orientation θ and $*$ denotes an element-wise multiplication.

Dynamic feature maps are produced equivalently to the static maps:

$$F_t(c,s) = |F_t(c) \ominus F_t(s)|\,, \tag{3.12}$$

$$R_t(c,s,\theta) = |R_t(c,\theta) \ominus R_t(s,\theta)|\,. \tag{3.13}$$

In conclusion, static and dynamic feature maps are fused into a spatiotemporal saliency map.

Another spatiotemporal model mentioned in (Rudoy et al., 2013) searches for motion candidates of eye fixations using *optical flow*. The model represents video frames by the optical flow magnitude. Regions with local motion are then detected through the difference in a Gaussian pyramid.

(Borji et al., 2011) introduced a top-down hierarchical model working with the same visual features as in (Itti et al., 1998). Bottom-up attention is modelled with center-surround differences in each scale of Gaussian pyramids separately instead of DoG. The surround inhibition compares the similarity of a center pixel with the average of its local surround window.

Top-down extension of this model modulates a fusion of features maps into a saliency map by learned weights associated to scales, dimensions and feature channels. Top-down weights are learned in the evolutionary process using a training set of images with objects of interest. The goal of this optimisation problem is to find a weight vector with the maximum detection rate of objects of interest and the minimum processing cost.

3.2 Bayesian Models

Bayesian models are probabilistic frameworks which combine bottom-up saliency with the effects of *prior visual experience* (Le Meur – Le Callet, 2009). When we search target features, top-down factors influences the attention. Impact of top-down attention is modelled using *Bayes' rule*.

Bayes' theorem is a probability function which converts the prior probability into the posterior probability by the following form (Murty – Devi, 2011):

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)}, \tag{3.14}$$

where $P(A \mid B)$ is the *conditional probability*, $P(A)$ and $P(B)$ denote the *prior probabilities* and $P(B \mid A)$ represents the *posterior probability*.

Bayesian framework defined in (Oliva et al., 2003) characterises a local bottom-up saliency as:

$$S(x) = \frac{1}{p(v_l(x))}, \tag{3.15}$$

where v_l is a feature vector with 48 dimensions describing the local structure created by hierarchical center-surround differences. Saliency of a given location increases with decreasing likelihood of finding a set of visual features in an image.

However, Equation 3.15 is not sufficient in case of searching of a target object based on our past experiences. In order to model a top-down mechanism of the visual searching the probability of object o at the location x has to be defined using Bayes' rule as the followings:

$$p(o, x \mid v_l, v_c) = \frac{p(v_l \mid o, x, v_c)}{p(v_l \mid v_c)} p(o, x \mid v_c), \tag{3.16}$$

where $p(v_l \mid o, x, v_c)$ is the *object likelihood*, $p(v_l \mid v_c)$ is the *local saliency* and $p(o, x \mid v_c)$ is the *contextual prior probability*. v_c is a *conceptual vector* which describes the structure of a whole image. Global context is represented by dimension reduction of local features using *Principal Component Analysis* (PCA) algorithm.

Saliency influenced by past knowledge can be defined by the following form:

$$S_c(x) = \frac{p(o, x \mid v_c)}{p(v_l \mid v_c)} = S(x) p(o, x \mid v_c). \tag{3.17}$$

Attention integrates a local saliency with the contextual prior probability as shown in Figure 3.6.

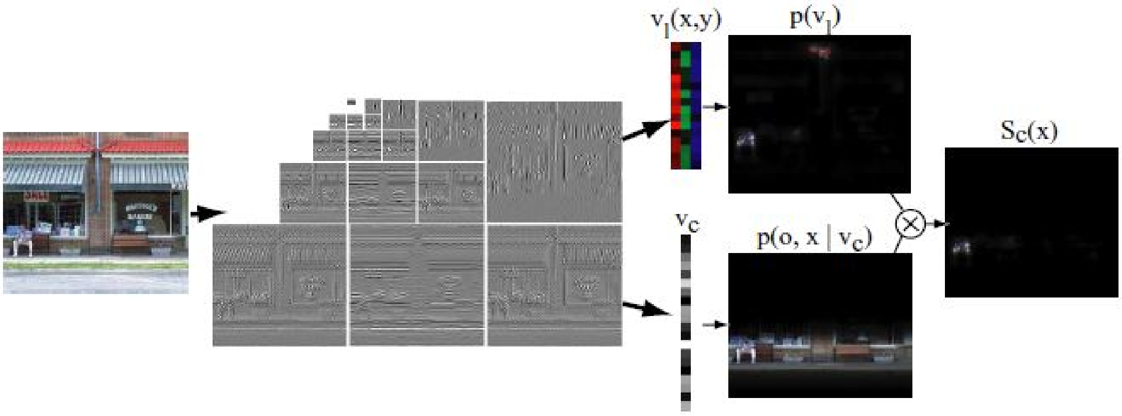


Figure 3.6: Attention as a combination of a local saliency with the context information (Oliva et al., 2003).

(Zhang et al., 2008) proposed a Bayesian framework called *SUN* (Saliency Using Natural statistics) which takes into account searching of target's features. Let C denote whether or not a point belongs to a target class, Z an image pixel, L the pixel location and F the visual features of a given point. SUN defines the saliency of a point z with the features f_z at the location l_z as:

$$s_z = p(C = 1 | F = f_z, L = l_z). \tag{3.18}$$

If features and locations are independent and conditionally independent for $C = 1$, Equation 3.18 can be rewritten in the following form:

$$\log s_z = -\log p(F = f_z) + \log p(F = f_z \mid C = 1) + log(C = 1 \mid L = l_z), \tag{3.19}$$

where $-\log p(F = f_z)$ is called the *self-information* and it is independent of target's features. It describes bottom-up saliency and measures the rareness of a feature. The rest of the equation depends on a target (top-down saliency). The term $\log p(F = f_z \mid C = 1)$ is a *log-likelihood* which increases with the consistence of features with prior knowledge of the target. The last term, $log(C = 1 \mid L = l_z)$, denotes the *location prior* depending on prior knowledge of the target's location.

A spatiotemporal Bayesian model in (Itti – Baldi, 2005) defines saliency as surprise that significantly changes prior beliefs of an observer. Surprise is measured by the *Kullback-Leibler* (KL) *divergence* between prior and posterior beliefs:

$$d_{KL} = KL(p(M \mid X), p(M)) = \int_M p(M \mid X) \log \frac{p(M \mid X)}{p(M)} dM, \tag{3.20}$$

where M denotes a prior knowledge and X is a new data observation. The surprise metric is applied both over space and time.

3.3 Decision Theoretic Models

Decision-theoretic models are based on the theory known as *discriminant saliency* that all saliency decisions are optimal in a decision-theoretic sense. Saliency is considered as the selection of optimal attributes that most distinguish a visual class of interest from the other classes.

The theory was first proposed for top-down saliency processing in (Gao – Vasconcelos, 2004). The discriminant feature selection is modelled by maximising the mutual information between a set of features and class labels.

The theory defines two different classes of stimuli (Gao – Vasconcelos, 2009):

1. a *null hypothesis* composed of non-salient background stimuli,
2. *stimuli of interest* composed of visual features distinguishing the foreground from the null hypothesis.

Decision-theoretic models classify the locations of stimuli of interest as salient with the *lowest expected misclassification error probability*.

(Gao et al., 2008) extends the model and combines the discriminant theory with center-surround differences of hierarchical bottom-up saliency for intensity, colour and orientation presented in (Itti et al., 1998).

Let $\mathbf{X} = \{X1, ..., X_d\}$ be a feature vector and Y a class label. $Y = 1$ represents the center and $Y = 0$ is its surround region. Then the saliency detection at location l corresponds to the computation of the mutual information I between the center and the surround defined as (Gao – Vasconcelos, 2009; Gao et al., 2008):

$$S(l) = I_l(\mathbf{X}, Y) = \sum_{i=1}^{d} I_l(X_i, Y). \tag{3.21}$$

The general process of the discriminant center-surround theory based saliency model is included in Figure 3.7.

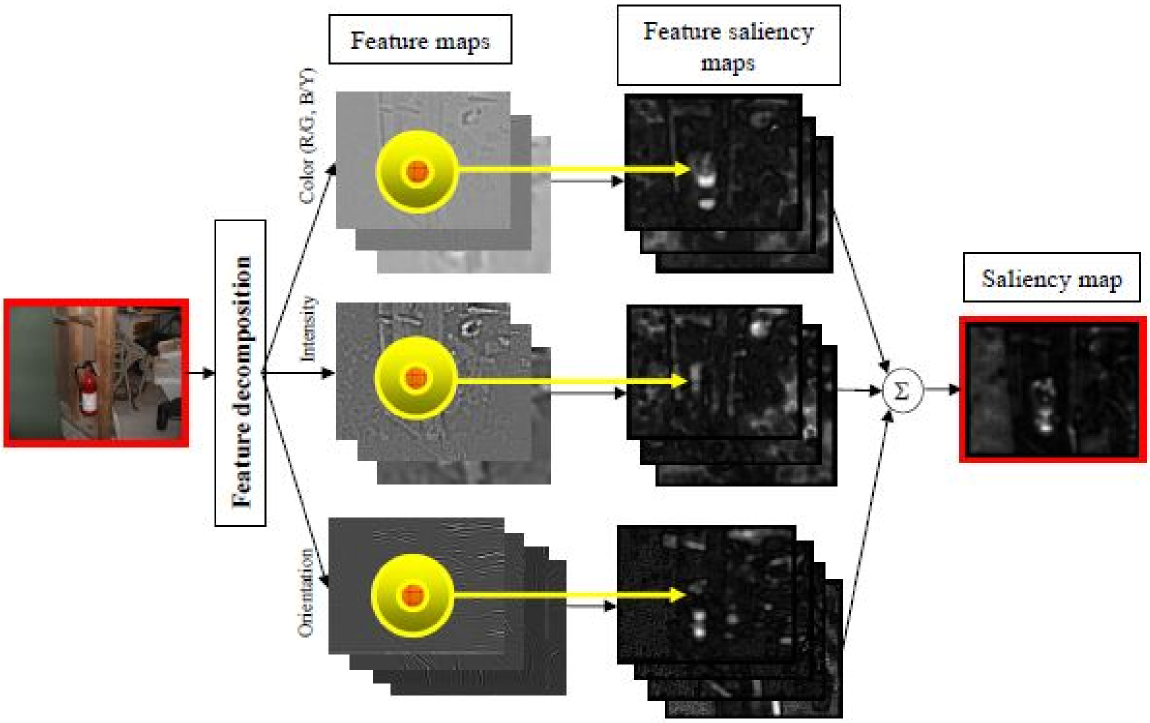


Figure 3.7: Discriminant saliency model (Gao et al., 2008).

3.4 Information Theoretic Models

Information-theoretic models are based on the theory which assumes that saliency results in the maximum information sampled from a given environment (Bruce, 2008).

(Bruce – Tsotsos, 2005) introduced the *AIM* (Attention based on Information Maximization) model measuring the information content of each image patch using a self-information defined as $-\log p(\mathbf{X})$, where $\mathbf{X}$ is a feature vector.

Let a patch size be $W \times H$, the probability density function (pdf) $p(\mathbf{X})$ of a RGB patch then requires an estimate in $W \times H \times 3$ dimensional space. In order to reduce the dimensionality of this problem, *Independent Component Analysis* (ICA) algorithm is implemented. ICA projects features representing a local neighbourhood into a space whose dimensions are as independent as possible. Instead of a high-dimensional pdf, ICA allows to estimate only $W \times H \times 3$ 1D probability density functions.

ICA is performed on a large sample of image patches from natural scenes.

It considers a patch as a linear combination of basis filters. If the mutual independence of ICA coefficients w_i is assumed, the pdf of an n-dimensional vector $\mathbf{w}$ can be computed as:

$$p(w_1 = v_1, w_2 = v_2, ..., w_n = v_n) = \prod_{i=1}^{n} p(w_i = v_i). \tag{3.22}$$

The product of all probability density functions of a local image region leads to a joint likelihood that is easily converted to the self-information (Figure 3.8) (Bruce, 2008).

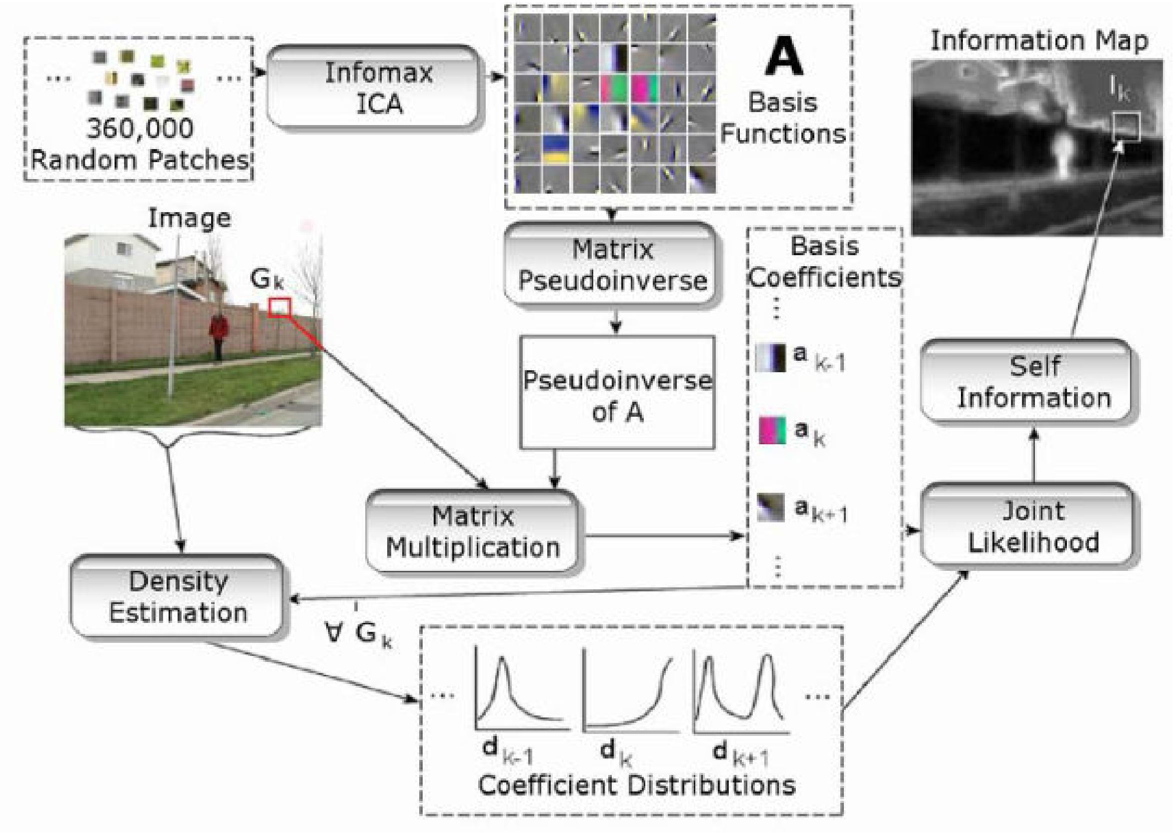


Figure 3.8: AIM model (Bruce – Tsotsos, 2007).

3.5 Graphical Models

Graphical models use graph-based computations to create a saliency map. Such models are probabilistic frameworks represented by a graph whose nodes present a set of variables and edges their probabilistic dependencies. An eye movement sequence treated as a time-ordered sequence is modelled using various methods such as *Markov Models*, *Conditional Random Fields* and *Dynamic Bayesian Networks* (Murty – Devi, 2011; Borji – Itti, 2013).

(Salah et al., 2002) designed an attention graphical model for handwritten digit and face recognition. The model consists of three basic steps.

In the first step called *Attentive Level* it constructs a bottom-up saliency map as a product of multiple feature maps.

The next step, *Intermediate Level*, simulates a sequence of saccades which connect eye fixations. The information about the sequence is extracted by dividing the image space into uniform regions and supervised training single-layer perceptrons over each region.

In the last *Associative Level* the quantised information is combined with a discrete *observable Markov model* (OMM) whose states denote regions with eye fixations.

Let S is a set of n distinct states and q_t a state at time t. The probability of an observation sequence $O = \{q_1 q_2 ... q_T\}$ depending on previous states is defined as (Alpaydin, 2010):

$$P(q_{t+1} = S_j \mid q_t = S_i, q_{t-1} = S_k, ...). \tag{3.23}$$

In the *first-order Markov model*, where the state at time $t+1$ depends only on the state at t, the probability of O is computed by the following form (Murty – Devi, 2011):

$$P(O, S \mid \lambda) = \pi_{s_1} b_{s_1}(O_1) \Pi_{i=2}^{n} a_{s_{i-1}s_i} b_{s_i}(O_i), \tag{3.24}$$

where O_i is the i^{th} observation symbol of the sequence O and $\lambda = \{\pi_i, a_{ij}, b_j(k)\}$ defines the parameters of the model:

- the *initial state probability* π_i: the probability of starting in state i,
- the *state transition probability* a_{ij}: the probability of transition from state i to state j,
- the *observation probability* $b_j(k)$: the probability of observing symbol k in the state j.

An observation sequence is classified to the class with the highest observation probability.

Shifting the focus to the next location is performed in (Salah et al., 2002) using the inhibition of return.

(Harel et al., 2006) introduced a bottom-up saliency model – *Graph-Based Visual Saliency* (GBVS). The extraction of intensity, colour and orientation feature maps is based on the hierarchical pyramid approach similar to (Itti et al., 1998).

Then a graph-based representation of a feature map is formed. The dissimilarity between two positions of a feature map (i, j) and (p, q) in a logarithmic scale is defined as:

$$d((i,j)\|(p,q)) = \left|\log \frac{M(i,j)}{M(p,q)}\right|, \tag{3.25}$$

where $M(x, y)$ is a feature value at the location (x, y).

The directed edge weight from node (i, j) to (p, q) is proportional to their mutual dissimilarity and their distance in M:

$$w((i,j),(p,q)) = d((i,j)\|(p,q))F(i-p, j-q), \tag{3.26}$$

where the distance $F(a, b) = \exp\left(-\frac{a^2+b^2}{2\sigma^2}\right)$.

After the normalisation of the outbound edges to 1 for each graph node, a *Markov chain* is defined on the graph whose nodes are treated as states and edge weights as transition probabilities.

The *equilibrium distribution*, reflecting the fraction of time spending at each state of this chain, is treated as activation and saliency map values. Normalised activation maps are finally combined into a single saliency map.

3.6 Spectral Analysis Models

Spectral analysis models process images in the frequency domain for example using *Fast Fourier Transform* instead of the spatial domain.

A spectral analysis model based on the *spectral residual* was proposed in (Hou – Zhang, 2007). The model adopts the idea that the visual information is the summation of two parts:

$$H(image) = H(innovation) + H(prior\ knowledge), \tag{3.27}$$

where $H(innovation)$ denotes the novelty part and $H(prior\ knowledge)$ is the redundant part of the information.

In order to express the novelty part of the information, a down-sampled input image $I(x)$ (width of $64px$) is transformed into the spectrum by the Fourier Transform $\mathfrak{F}$ and its amplitude $\mathcal{A}(f)$ and phase $\mathcal{P}(f)$ are derived:

$$\mathcal{A}(f) = abs(\mathfrak{F}(x)), \tag{3.28}$$

$$\mathcal{P}(f) = angle(\mathfrak{F}(x)). \tag{3.29}$$

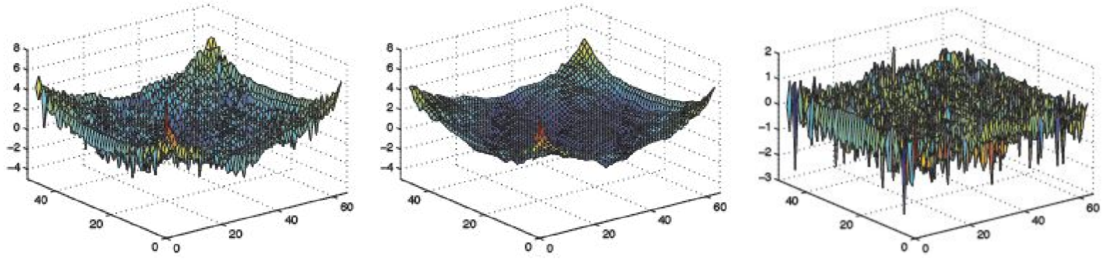
Consequently, the *log spectrum* representation $\mathcal{L}(f)$ is computed:

$$\mathcal{L}(f) = \log(\mathcal{A}(f)). \tag{3.30}$$

The *spectral residual* $\mathcal{R}(f)$, containing the innovation of an image, is obtained as the difference between the original and smoothed version of the log spectrum (Figure 3.9):

$$\mathcal{R}(f) = \mathcal{L}(f) - h_n(f) * \mathcal{L}(f), \tag{3.31}$$

where $h_n(f)$ is an $n \times n$ average filter.



(a) Log spectrum. (b) Smoothed log spectrum. (c) Spectral residual.

Figure 3.9: Spectral residual, the difference between the original and smoothed log spectrum (Loy et al., 2012).

The final saliency map is built in the spatial domain using the Inverse Fourier Transform $\mathfrak{F}^{-1}$ and smoothed with a Gaussian filter $g(x)$:

$$\mathcal{S}(x) = g(x) * \mathfrak{F}^{-1}[\exp(\mathcal{R}(f) + \mathcal{P}(f))]^2. \tag{3.32}$$

(Loy et al., 2012) uses the similar spectral residual approach to detect salient motion. Using the *optical flow* algorithm, an image is represented by flow magnitude and phase fields. Magnitude and orientation are separately processed to obtain spectral residual based saliency maps that are finally combined into a single motion saliency map as shown in Figure 3.10.

Another spectral model called *HFT* is proposed by (Li et al., 2013). Instead of a single feature, it computes saliency using three features, similarly to (Itti et al., 1998): intensity,

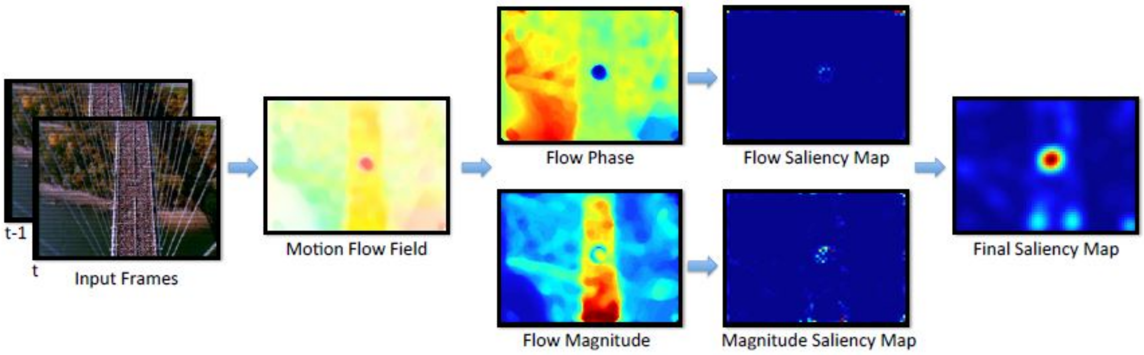


Figure 3.10: Motion saliency model (Loy et al., 2012).

RG and BY colour pairs. Hence, it analyses the spectrum using *Hypercomplex Fourier Transform* (HFT) whose input is a hypercomplex matrix of vectors instead of a real matrix. Each element of the matrix is defined as follows:

$$f(m,n) = w_1 f_M + w_2 f_I i + w_3 f_{RG} j + w_4 f_{BY} k, \tag{3.33}$$

where f_I, f_{RG} and f_{BY} are static feature maps and f_M is an optional motion map. Equation 3.33 is derived from a *quaternion* definition $q = a + bi + cj + dk$, where a, b, c, $d \in \mathbb{R}$ and $i^2 = j^2 = k^2 = ijk = -1$. After the performing of HFT the amplitude spectrum is smoothed with series of Gaussian kernels. Performing the inverse HFT we obtain saliency maps at multiple scales.

3.7 Pattern Classification Models

Pattern classification models use *supervised machine learning* algorithms to learn visual attention from eye-tracking data or labelled salient regions. These models may cover bottom-up and top-down attention too (Borji – Itti, 2013).

(Kienzle et al., 2009) proposed a learning saliency model that is trained on recorded eye tracking data. In order to classify the saliency of image patches, the model uses the *Support Vector Machine* (SVM) algorithm. SVM is learned on a training data set of patches described by the local intensities. The patches corresponding to eye fixations are labelled as target and the patches produced from the same eye tracking data but applied on different images as non-target.

A model in (Judd et al., 2009) also uses the SVM classifier for attention prediction. A training dataset consists of feature vectors from fixations and random locations. The model combines low-level features such as center-surround based intensity, colour and orientation maps, mid-level features including a horizon line detector, high-level features including people and face detectors and the distance to the center.

3.8 Reinforcement Learning Models

Reinforcement learning models predict visual saliency using the *reinforcement learning* algorithm (Filipe – Alexandre, 2013).

In the reinforcement learning (RL) inspired by behaviourist psychology, an agent takes actions in an environment that change its actual state and receives reward or penalty. The aim of the algorithm is to learn the best sequence of agent's actions that maximises the *cumulative reward* (Alpaydin, 2010).

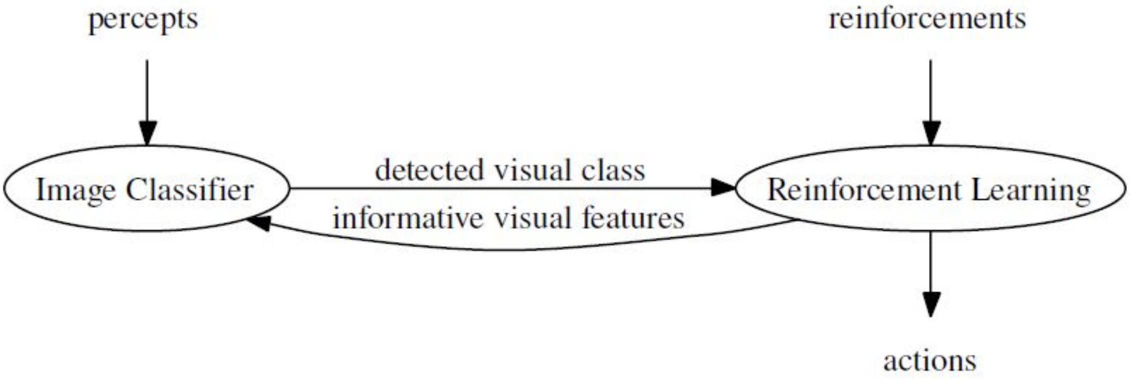


Figure 3.11: Reinforcement learning (Jodogne – Piater, 2007).

(Jodogne – Piater, 2007) introduced a learning model based on the RL known as *Reinforcement Learning of Visual Classes* (RLVC). RLVC consists of two processes. RL unit learns an optimal mapping from visual classes to actions and an image classifier iteratively learns to partition the feature space of an image into a set of distinct visual classes according the presence or the absence of informative visual features (Figure 3.11).

At the beginning of splitting the feature space, there is only one visual class. The agent selects iteratively for each detected class a new distinctive feature and the classifier is refined using the descriptor.

3.9 Other Models

This section contains examples of saliency models that do not fit into the previous categories.

The authors in (Goferman et al., 2012) define a *context-aware saliency* (CAS) based on four principles of visual attention:

1. *local low-level considerations* including distinctive colour and contrast,
2. *global rarity considerations* which suppress frequent visual features,
3. *visual organisational rules* according to which salient pixels are grouped together,
4. and *high-level factors*, such as recognised objects and human faces.

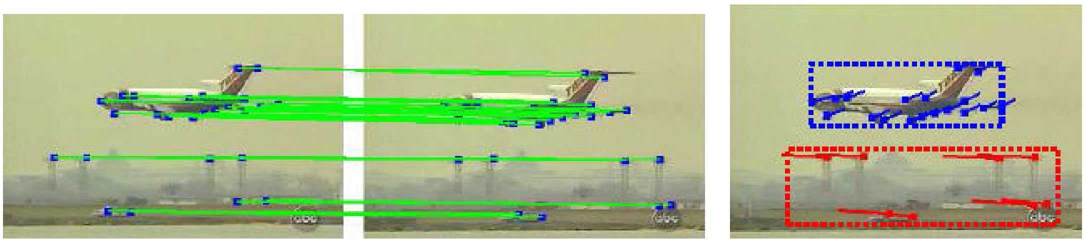
A temporal attention model introduced in (Zhai – Shah, 2006) is based on geometric transformations between consequent frames of a video sequence. First, the *Scale Invariant Feature Transformation* (SIFT) is accomplished to extract keypoints and describe their local neighbourhood in images. This descriptor computes a scale- and rotation-invariant 128-dimensional vector formed from a local histogram of oriented gradients (Lowe, 2004). After that, the model matches corresponding points between consequent frames.

Temporal saliency is computed based on a *homography* (Figure 3.12). It is a 3-by-3 transformation matrix that projects points from one plane to another, such that $x' = Hx$, where x is a point in homogenous coordinates. The model applies an iterative method called *RANSAC* (Random Sample Consensus) to compute multiple homographies in order to detect different moving objects in videos.

The temporal saliency value of a point $\mathbf{p}$ is then defined as:

$$SalT(\mathbf{p}) = \sum_j \alpha_j \times \epsilon(\mathbf{p}, \mathbf{H}_j), \tag{3.34}$$

where $\epsilon(\mathbf{p}, \mathbf{H}) = \|\hat{\mathbf{p}}' - \mathbf{p}'\|$, $\mathbf{p}'$ is a corresponding point of $\mathbf{p}$ and $\hat{\mathbf{p}}' = \mathbf{Hp}$. α_j defines the spanning area of a homography $\mathbf{H}_j$ proportional to the spatial distributions of the inlier set.



(a) Correspondences between consequent frames.

(b) Motion regions.

Figure 3.12: Motion saliency based on geometric transformations between video frames (Zhai – Shah, 2006).

A generic spatiotemporal attention model in (Ma et al., 2005) combines visual, aural and linguistic attention models for video summarisation. Visual attention modelling incorporates static contrast-based attention, motion attention, face detection and camera motions.

Dynamic saliency is estimated from motion vectors between blocks of pixels called *macro blocks* representing the smallest independent units of MPEG streams. Motion information of each macro block is specified by three characteristics:

1. *motion intensity* derived from the magnitude of motion vector,
2. *spatial coherence* of motion vectors,
3. and *temporal coherence* of motion vectors.

Chapter 4

Analysis of Used Principles

The main goal of this master thesis is the creation of a novel bottom-up saliency model.

In order to create such a model, we have chosen one of the most popular models presented in (Itti et al., 1998), closely described in Section 3.1. We have implemented a saliency detection method inspired by this model and analysed its advantages, disadvantages and possible improvements.

The model relies on the following three features - *intensity*, *colour* and *orientation* which are used for creating a *Gaussian pyramid*. Each feature is represented by a *conspicuous map* computed by combining the results of *difference of Gaussians* algorithm (Figure 4.1(b), 4.1(c) and 4.1(d)).

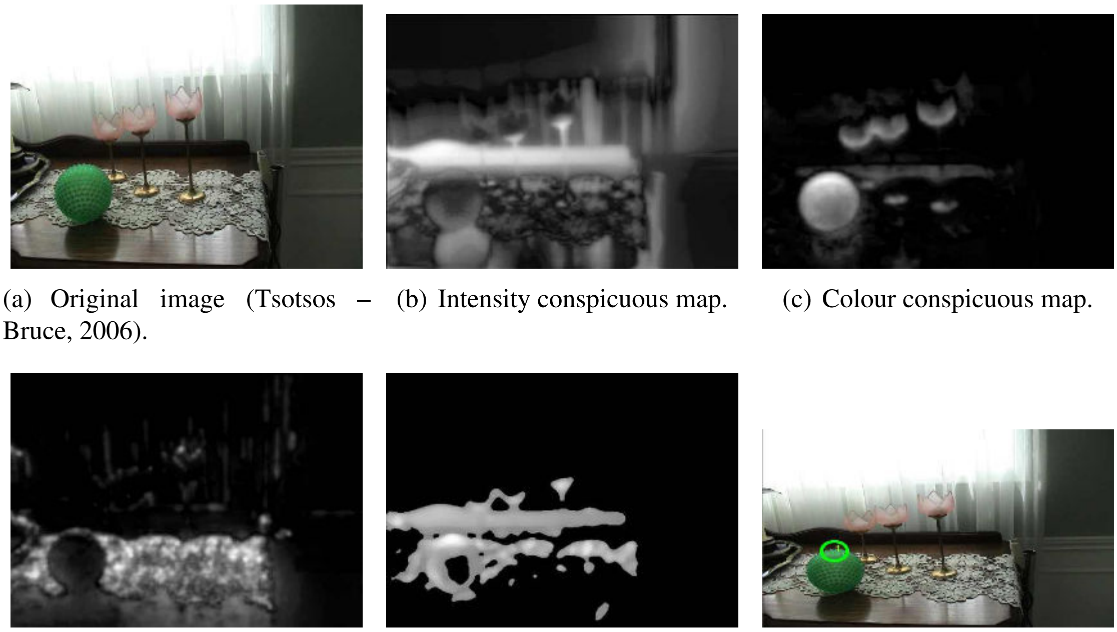
The final *saliency map* is a product of combination of these 3 conspicuous maps (Figure 4.1(e)). Values of saliency map pixels denote conspicuousness of the location. Thus, the maximum value pixel refers to the most salient location of the scene (Figure 4.1(f)). Shift to the next salient location is done by suppressing the nearest surroundings of the previously searched most salient position whose pixels are adjusted to zero. Then the model finds the next salient region as the surroundings of a new maximum value pixel. Figure 4.2 represents an example of this process.

Generally, visual information processing consists of two main parts (Goldstein, 2010). In the *pre-attentive* phase, individual features are searched and within the second *focused attentive* phase detected features are combined to perceive objects. However, the focused attention is omitted in the method proposed in (Itti et al., 1998).

In order to at least partially cover this attention, we implement a *superpixel-based saliency* in our model instead of a simple pixel-by-pixel-based difference of Gaussian pyramid layers. Our model is also a hierarchical saliency model based on FIT. A *superpixel* represents a visually coherent region (Lim – Han, 2014) which can better correspond to object contours than a rigid structure of pixels. The usage of superpixels is the primary difference between our model and the standard hierarchical model in (Itti et al., 1998).

Our superpixel-based model differs in the orientation detection as well as in the computation of an output saliency map from conspicuous maps too.

In order to apply saliency maps on video sequences, we append motion processing to the model. According to Gibson's theory (Gibson, 1950) of optical flow patterns, a moving object in a scene is perceived through a local disturbance of an optic array. Using the as-



(a) Original image (Tsotsos – Bruce, 2006).

(b) Intensity conspicuous map.

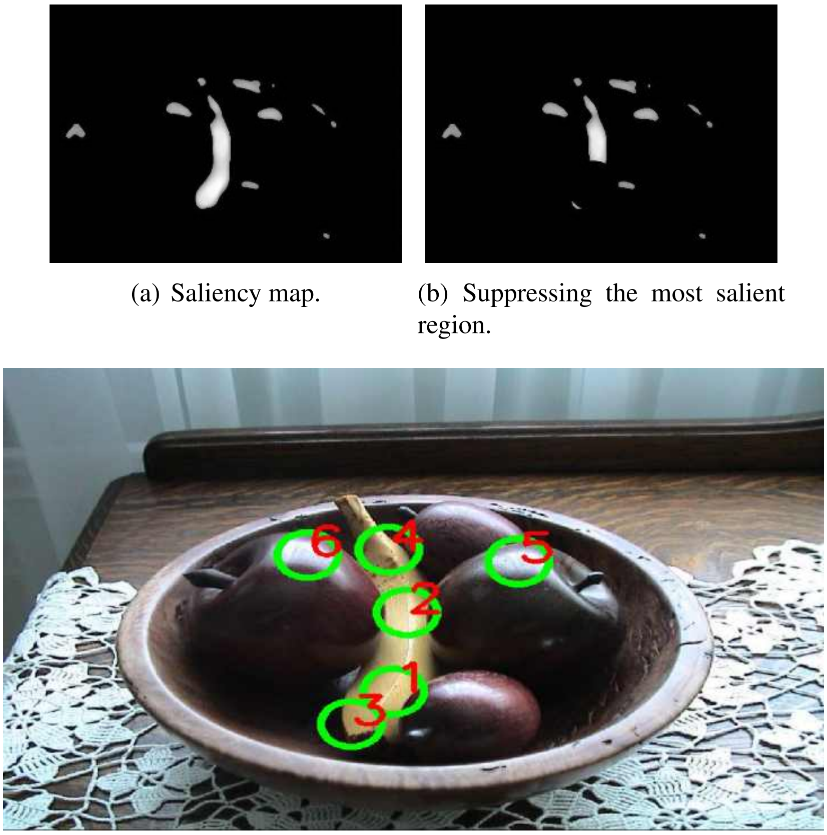
(c) Colour conspicuous map.

(d) Orientation conspicuous map.

(e) Thresholded saliency map.

(f) The most salient location according the model.

Figure 4.1: Hierarchical saliency model based on (Itti et al., 1998).



(a) Saliency map.

(b) Suppressing the most salient region.

(c) The most salient regions denoted with green circles (numbers represent the order of saliency).

Figure 4.2: The process of salient regions detection. The next salient location is searched by suppressing the previous maximum salient location. Repeating this process the model finds all salient areas.

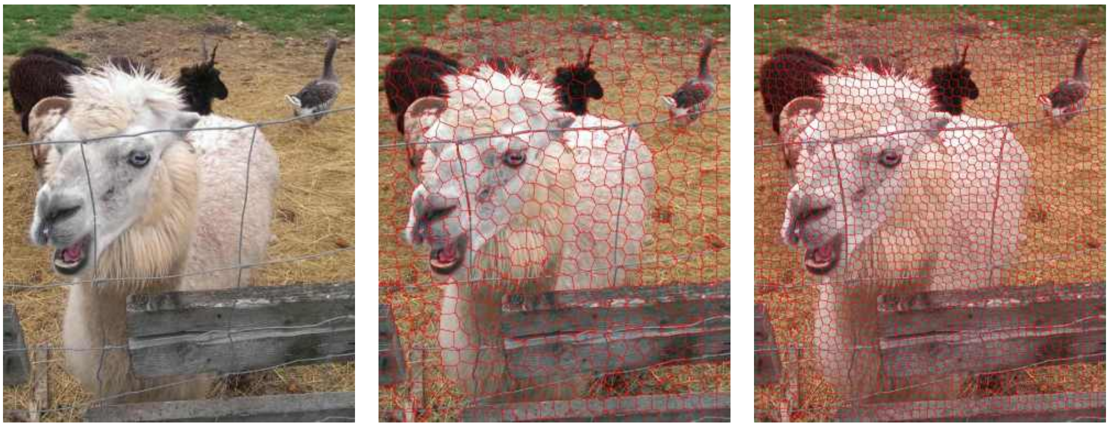
sumption of motion coherency within a superpixel we implement the theory using the same hierarchical approach used in a static form of our novel model. Resulting *spatiotemporal superpixel-based model* combines a static saliency map with a motion saliency map as well as a *motion innovation map* characterising dynamic changes in a scene.

The further details of our model and other differences with a standard hierarchical saliency map are mentioned in Chapter 5.

4.1 Superpixels

Superpixel segmentation, a special case of an image over-segmentation, splits an image into perceptually meaningful atomic units called *superpixels* (Figure 4.3). It is a more natural representation of an image than a pixel grid because it is a group of nearly uniform pixels which should align well with image edges (Neubert – Protzel, 2012).

The characteristics of superpixels provide an opportunity to reduce the complexity of many computer vision tasks. Superpixels may be applicable in various areas such as object localisation, segmentation, skeletonisation or depth estimation (Achanta et al., 2010).



(a) Image. (b) Superpixels of size 30. (c) Superpixels of size 15.

Figure 4.3: Superpixel segmentation using SLIC algorithm.

There are two main approaches to generate superpixels (Achanta et al., 2012):

1. **Graph-based** algorithms use a graph representation of superpixels. Graph nodes represent pixels and edge weights between them depend on the neighbouring pixels similarities. Grouping pixels into superpixels is performed via minimum cut algorithm. Typical segmentation methods are *Normalised Cuts* (Shi – Malik, 2000) and *Felzenszwalb-Huttenlocher Segmentation* (Felzenszwalb – Huttenlocher, 2004).

2. **Gradient-ascend-based** algorithms iteratively refine clusters of pixels, for example *Mean-shift* (Comaniciu – Meer, 2002), *Quickshift* (Vedaldi – Soatto, 2008), *Watersheds* (Vincent – Soille, 1991) and *Turbopixels* (Levinshtein et al., 2009).

4.1.1 Simple Linear Iterative Clustering

Simple linear iterative clustering (SLIC) (Achanta et al., 2012) groups pixels according their colour similarity and spatial proximity. This state-of-the-art approach is based on *k-means algorithm*. SLIC uses a 5-dimensional feature space $[labxy]^T$, where l, a and b are colour channels of CIELab colour space, x and y are image coordinates.

It initialises k cluster means regularly on a grid with the step S in the 5D feature space. k means are moved to the location with the lowest gradient in 3×3 neighbourhood to avoid a cluster centre on an image edge.

Afterwards, k-means algorithm is performed to assign all pixels to clusters. Using a standard k-means algorithm, a pixel is associated to the nearest cluster after the computation of distances to all clusters means. To speed up the whole process, the searched region is reduced to the size $2S \times 2S$ around the centre, due to the approximate superpixel size $S \times S$. The assignment process is iteratively repeated until convergence. Equation 4.3 is a distance measure in the 5D space, where d_c is a colour proximity, d_s is a spatial proximity and m denotes a weight between the proximities:

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}, \tag{4.1}$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \tag{4.2}$$

$$D = \sqrt{{d_c}^2 + \left(\frac{d_S}{S}\right)^2 m^2}. \tag{4.3}$$

In a post-processing phase, separated pixels are reassigned to neighbouring superpixels by a connected components algorithm.

4.2 Histogram and Histogram Comparison

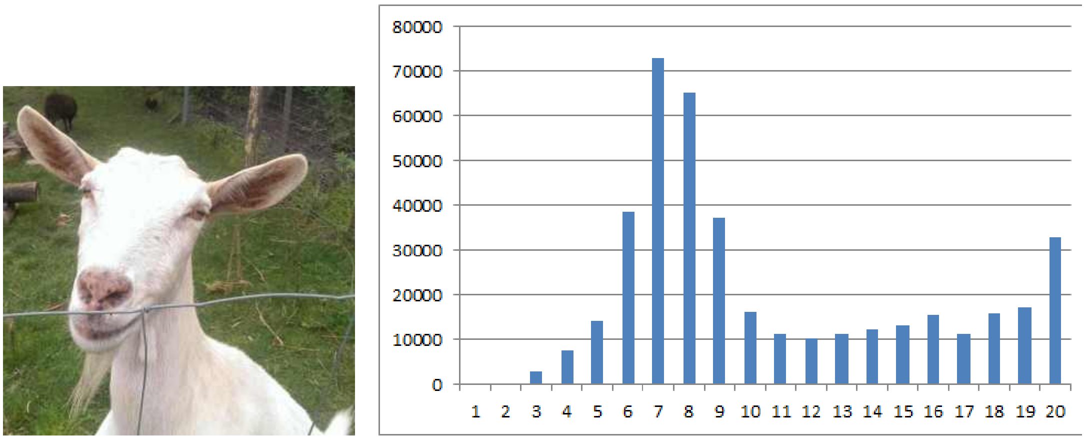
Histograms provide the frequency of values in a given image organised into a set of intervals called *bins*. In other words, it is a probabilistic description of an image feature, such as brightness and colour (Sonka et al., 2007). Figure 4.4 illustrates an example of a 1D grayscale histogram with 20 bins.

Histograms may be applied in measuring the similarity between images. Let H_1 be a histogram obtained from the first image and H_2 a histogram from the other one, both with N bins. There are various ways how to compare histograms. Among the most common measures belong (Bradski – Kaehler, 2008):

1. *Correlation*:

$$d_{correl}(H_1, H_2) = \frac{\sum_i H_1'(i) \cdot H_2'(i)}{\sqrt{\sum_i H_1'^2(i) \cdot H_2'^2(i)}}, \tag{4.4}$$

where $H_k' = H_k(i) - \frac{1}{N}\sum_j H_k(j)$. The correlation coefficient takes values within $\langle -1, 1 \rangle$. Values $+1$ and -1 denote a perfect positive and negative correlation. If the coefficient equals zero, histograms are totally independent.



(a) Image. (b) 1D grayscale histogram (20 bins).

Figure 4.4: Histogram representation of an image.

2. *Chi-square*:

$$d_{chi-square}(H_1, H_2) = \sum_i \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)}, \tag{4.5}$$

where the chi-square coefficient is greater than or equal to 0. Zero value represents the perfect match.

3. *Intersection*:

$$d_{intersection}(H_1, H_2) = \sum_i \min(H_1(i), H_2(i)), \tag{4.6}$$

where the higher intersection coefficient indicates the higher similarity. If compared histograms are normalised to 1 at first, a value of 1 represents the match.

4. *Bhattacharyya distance*:

$$d_{Bhattacharyya}(H_1, H_2) = \sqrt{1 - \sum_i \frac{\sqrt{H_1(i) \cdot H_2(i)}}{\sqrt{\sum_i H_1(i) \cdot \sum_i H_2(i)}}}, \tag{4.7}$$

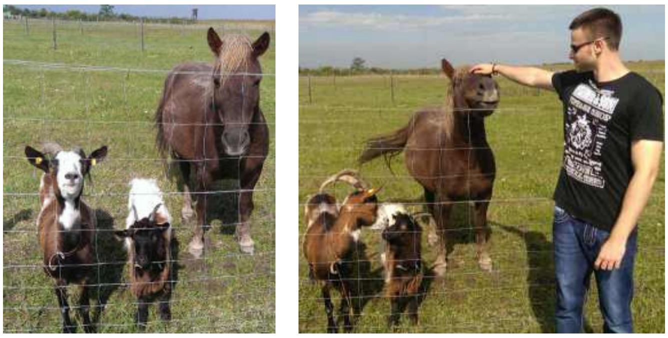
where the Bhattacharyya coefficient ranges from 0 to 1 and the match is a value of 0.

An example of histogram matching represents Figure 4.5. Histograms of two images converted to grayscale have been compared using various methods and the results are listed in Figure 4.5(c).

4.3 Optical Flow

Optical flow is defined as the change in *optic array* – the structured pattern of light reaching a viewer from all directions (Figure 4.6). The disturbance of optic array is generated by *apparent motion* of similar brightness patterns (Sonka et al., 2007)[1].

[1] MAJUMDER, A. Optical Information, Department of Computer Science, University of California. Available from: `http://www.ics.uci.edu/~majumder/vispercep/chap2notes.pdf`



Method	Measurement
Correlation	0.571769
Chi-square	29.201021
Intersection	28.556991
Bhattacharyya	0.232051

(a) Source image 1. (b) Source image 2. (c) The results of histogram matching.

Figure 4.5: Grayscale histograms normalised into the interval $\langle 0, 1\rangle$ have been compared using four comparison algorithms: correlation, chi-square, intersection and Bhattacharyya distance.

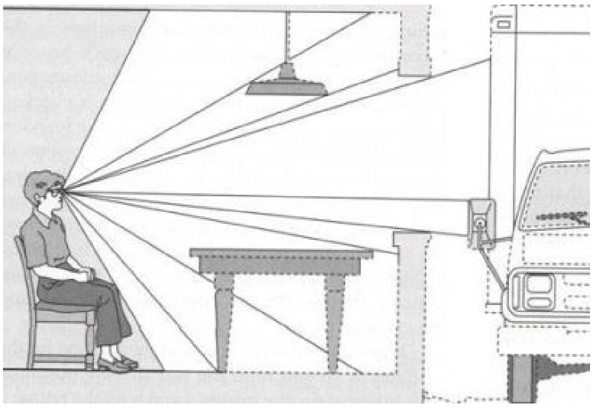


Figure 4.6: Optic array formed by surfaces, textures and contours[1].

A typical usage of dynamic scene analysis of a sequence of image frames includes motion detection and segmentation, object tracking and derivation of 3D object properties from motion. Optical flow is one of the most common techniques for motion analysis. It is an approximation to the *motion field*, which is the perspective projection of 3D motion onto the image plane. A *velocity vector* is assigned to each point of the motion field corresponding to the motion direction and velocity magnitude as shown in Figure 4.7 (Sonka et al., 2007; Wu, 2001):

$$\mathbf{p} = Z\widehat{m}, \tag{4.8}$$

where $\mathbf{p} = [X, Y, Z]^T$ is a 3D point and $\widehat{m}$ is the homogenous coordinate of its 2D representation $m = [x, y]^T$. Differentiating the previous expression with respect to time gives:

$$\frac{d\mathbf{p}}{dt} = \frac{dZ}{dt}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + Z\begin{bmatrix} dx/dt \\ dy/dt \\ 0 \end{bmatrix}, \tag{4.9}$$

which results in:

$$\mathbf{V_p} = \mathbf{V_{p_z}}\widehat{m} + Z\mathbf{v_m}. \tag{4.10}$$

2D motion field $\mathbf{v_m}$ is then defined as:

$$\mathbf{v_m} = \frac{\mathbf{V_p} - \mathbf{V_{p_z}}\widehat{m}}{Z}. \tag{4.11}$$

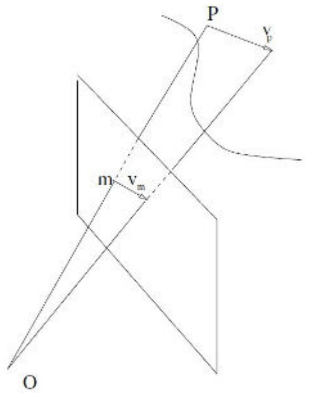


Figure 4.7: Motion field.

Optical flow computation is based on three assumptions (Sonka et al., 2007)[2]:

1. *Brightness constancy*: The brightness of any object point remains constant over time.
2. *Spatial coherence*: Neighbouring points are likely to belong to the same surface and hence have similar motions.
3. *Temporal persistence* (small movements): Motion changes gradually over time.

The brightness constancy assumption can be expressed as:

$$I_t(x,y) = I_{t+1}(x+u, y+v), \tag{4.12}$$

where I_t is the brightness at time t and (u,v) is the displacement. Using Taylor series and assuming the movement is very small, we get the following approximation:

$$I_{t+1}(x+u,y+v) \approx I_t(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t}. \tag{4.13}$$

It leads to Equation 4.14 known as the *optical flow constraint equation*.

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v = -\frac{\partial I}{\partial t} \tag{4.14}$$

The goal is to compute the velocity vector:

$$\mathbf{c} = \left(\frac{dx}{dt}, \frac{dy}{dt}\right) = (u,v). \tag{4.15}$$

Methods for optical flow computation can be classified into two major categories (Sonka et al., 2007):

1. **Local (sparse) approach** such as the *Lucas-Kanade* (LK) method (Lucas et al., 1981): LK tracker assumes that motion is constant in a small window (Figure 4.8).
2. **Global (dense) approach** such as *Horn-Schunck* (HS) method (Horn – Schunck, 1981): HS technique combines the brightness constancy constraint with the smooth-

[2] POCK, T. Image Processing and Pattern Recognition: Optical Flow, Institute for Computer Graphics and Vision, Graz University of Technology. Available from: `http://www.icg.tugraz.at/courses/lv710.080/info/motion_estimation.pdf`

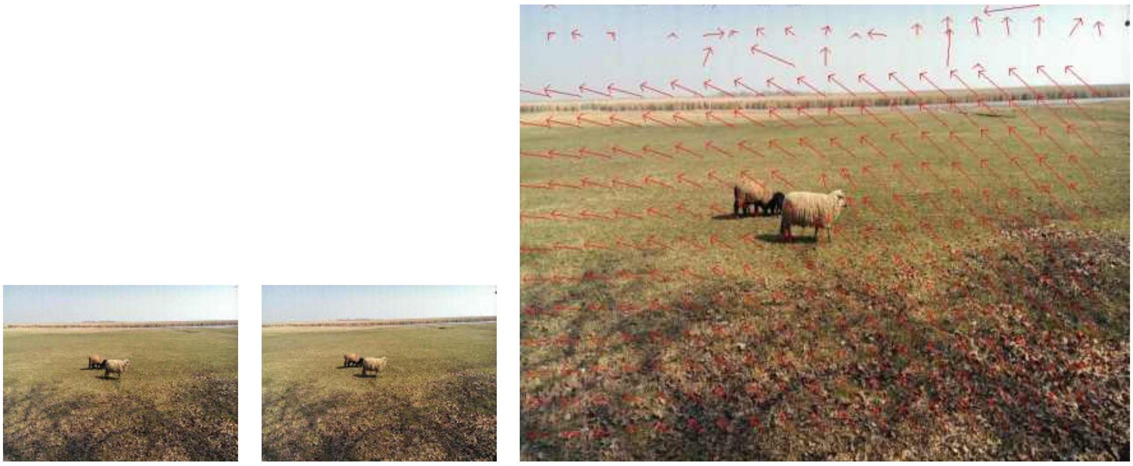
ness constraint according to which optical flow varies smoothly. HS approach minimises a global energy represented by the squared error quantity:

$$E_{HS} = \|\nabla u\|_2^2 + \|\nabla v\|_2^2 + \lambda \|I_t + I_x(u - u_0) + I_y(v - v_0)\|_2^2, \tag{4.16}$$

where ∇ is the gradient operator, λ is the *regularisation parameter* adjusting the smoothness of vectors and $\|\mathbf{x}\|_2$ is the ℓ_2 norm. This method does not allow for sharp discontinuities in the flow field. In contrast to the HS method, the *TV-L1 approach* (Zach et al., 2007) replaces the quadratic functions by ℓ_1 norms which better preserve discontinuities:

$$E_{TV-L1} = \|\nabla(u, v)\|_{2,1} + \lambda \|I_t + I_x(u - u_0) + I_y(v - v_0)\|_1, \tag{4.17}$$

where $\|\nabla(u, v)\|_{2,1} = \sum_{i,j} \sqrt{|(\nabla u)_{i,j}|_2^2 + |(\nabla v)_{i,j}|_2^2}$. The first term is the total variation (TV) of the flow field and the second term is referred to as the ℓ_1 norm of the optical flow constraint[2].



(a) Previous frame. (b) Next frame. (c) Optical flow.

Figure 4.8: Optical flow computed by LK tracker. A global flow is generated by motion of an observer and a local flow by motion of objects.

Chapter 5

Proposed Algorithm

We introduce a novel *Hierarchical Superpixel-Based Saliency Model* for the detection of bottom-up saliency. The model segments input images converted to grayscale into superpixels using *SLIC* algorithm. SLIC is performed twice in the model with two different region sizes of 15 and 30.

Each superpixel is represented by a 1D histogram. Superpixel representation can partially include the integration of visual features to objects in the human visual attention processing.

A spatial form of the model based on *FIT* integrates the following three features:

1. *intensity*,
2. *colour*,
3. *orientation*.

The algorithm hierarchically processes all features using *Gaussian pyramids* with 6 layers. Center-surround organisation of human ganglion cells is modelled as a difference between finer and coarser levels of the pyramid. The center is represented at scales $c \in \{0, 1, 2\}$ and the surround scales are $s = c + \delta$, where $\delta \in \{1, 2, 3\}$.

We extend this novel model with a temporal feature – *motion* to detect salient locations in videos. Our spatiotemporal model extracts motion information from optical flow maps representing the direction and the velocity magnitude of the associated flow vector. Motion feature is used to localise salient moving objects and motion innovation parts of a video sequence.

5.1 Superpixel Gaussian Pyramid

Due to the usage of superpixels, the standard algorithm for Gaussian pyramid is replaced by our superpixel version. Each pyramid layer consists of a *superpixel map* representing the locations of all superpixels and a *set of superpixel histograms*.

Within the first pyramid layer multiple 1D superpixel histograms are constructed using three visual features.

In order to create the next layers, we have to downsample the superpixel map to the half of its size.

Then we search neighbours to all superpixels in this resized superpixel map. The neighbour assignment procedure processes superpixel borders per pixel. Each border pixel is classified into one of the following categories based on its location to the analysed superpixel: *left* (L), *right* (R), *upper* (U), *bottom* (B), *upper-left* (UL), *bottom-left* (BL), *upper-right* (UR) and *bottom-right* (BR). Within each category, neighbours are characterised by a weight which depends on the boundary length with the superpixel. Longer the mutual boundary is, higher weight is assigned to the neighbour.

Listing 5.1 describes the searching process of superpixel neighbourhood.

Listing 5.1: Searching for superpixel neighbourhood.

```
Input:
spx_map - superpixel map
spx_id - number of the analysed superpixel

Initialisation:
neighbourhood_weight[category,id] - list of neighbouring superpixel
   weights assigned to their location categories

Superpixel neighbouhood processing:
FOR EACH border pixel [x,y] of superpixel spx_id in spx_map
    IF id(x,y) <> id(x-1,y) THEN
        ADD 1 TO neighbourhood_weight[LEFT,id(x-1,y)]
    ENDIF

    IF id(x,y) <> id(x+1,y) THEN
        ADD 1 TO neighbourhood_weight[RIGHT,id(x+1,y)]
    ENDIF

    IF id(x,y) <> id(x,y-1) THEN
        ADD 1 TO neighbourhood_weight[UPPER,id(x,y-1)]
    ENDIF

    IF id(x,y) <> id(x,y+1) THEN
        ADD 1 TO neighbourhood_weight[BOTTOM,id(x,y+1)]
    ENDIF

    IF id(x,y) <> id(x-1,y-1) AND id(x,y) <> id(x-1,y) THEN
        ADD 1 TO neighbourhood_weight[UPPER-LEFT,id(x-1,y-1)]
    ENDIF

    IF id(x,y) <> id(x-1,y+1) AND id(x,y) <> id(x-1,y) THEN
        ADD 1 TO neighbourhood_weight[BOTTOM-LEFT,id(x-1,y+1)]
    ENDIF

    IF id(x,y) <> id(x+1,y-1) AND id(x,y) <> id(x+1,y) THEN
        ADD 1 TO neighbourhood_weight[UPPER-RIGHT,id(x+1,y-1)]
    ENDIF

    IF id(x,y) <> id(x+1,y+1) AND id(x,y) <> id(x+1,y) THEN
        ADD 1 TO neighbourhood_weight[BOTTOM-RIGHT,id(x+1,y+1)]
    ENDIF
ENDFOR

FOR EACH weight IN neighbourhood_weight
    normalise weight within each location category
ENDFOR
```

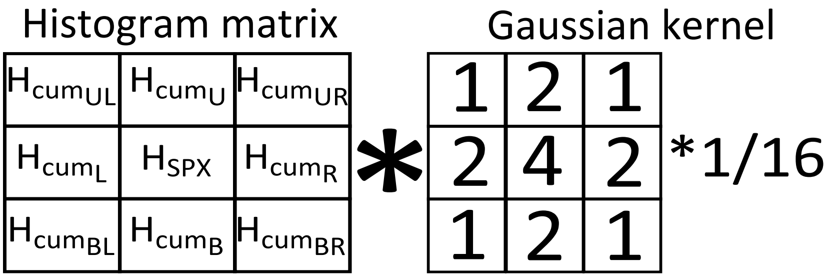
```
Output:
neighbourhood_weight - weight of all superpixels in the neighbourhood
```

After the processing of the whole superpixel neighbourhood, we can build a *histogram matrix* of size 3×3. The center element corresponds to the analysed superpixel histogram H_{SPX}. All 8 location categories are presented with a *cumulated histogram* defined as the weighted sum of all neighbour histograms connected to the category:

$$H_{cum_i} = \sum_j H_j \cdot w_j, \tag{5.1}$$

where H_j is a neighbour histogram and w_j is a weight of the neighbour.

The rest of the histogram matrix is build using the 8 cumulated histograms of all location categories. Each cumulated histogram is assigned to the position in the matrix depending on the category name, for example the upper-left cumulated histogram takes place in the first (upper-left) position of the matrix. The histogram matrix is finally convolved with a discrete 3×3 Gaussian kernel (Figure 5.1). In case of convolution at image borders where empty location categories without any neighbours may occur, the cumulated histogram of such categories equals to the histogram of analysed superpixel H_{SPX}.



$$\begin{array}{|c|c|c|} \hline H_{cum_{UL}} & H_{cum_{U}} & H_{cum_{UR}} \\ \hline H_{cum_{L}} & H_{SPX} & H_{cum_{R}} \\ \hline H_{cum_{BL}} & H_{cum_{B}} & H_{cum_{BR}} \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} * 1/16$$

Figure 5.1: Superpixel Gaussian convolution. Histogram matrix created from the analysed histogram H_{SPX} and 8 cumulated histograms H_{cum_i} is convolved with a Gaussian kernel.

In order to create a pyramid layer, this procedure is repeated for all superpixels.

To produce the rest of Gaussian pyramid layers, the whole process is iteratively performed with the half size of an input superpixel map (Figure 5.2). The general description of superpixel Gaussian pyramid algorithm is listed in Listing 5.2.

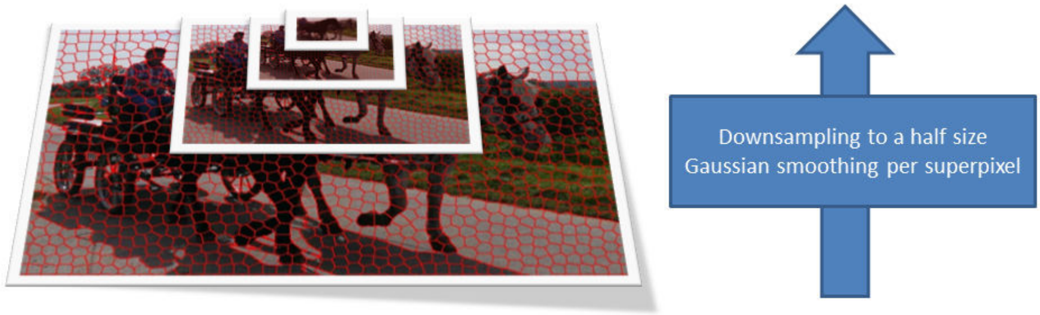

Downsampling to a half size
Gaussian smoothing per superpixel

Figure 5.2: Iterative process of superpixel Gaussian pyramid.

Listing 5.2: Superpixel Gaussian pyramid.

```
Input:
spx_map - superpixel map obtained from SLIC algorithm
hist_set - histogram representations of all superpixels
levels - number of pyramid layers

Initialisation:
spx_map_pyr[0] ← spx_map (pyramid of superpixel maps)
hist_set_pyr[0] ← hist_set (pyramid of superpixel histogram sets)

Gaussian pyramid computing:
FOR l = 1 to levels
    spx_map_pyr[l] ← resize spx_map_pyr[l-1] to a half size
    spx_neighbourhoods ← extract superpixel neighbourhoods with
        spx_map_pyr[l]
    hist_set_pyr[l] ← apply superpixel Gaussian smoothing to
        hist_set_pyr[l-1] with spx_map_pyr[l] and spx_neighbourhoods
ENDFOR

Output:
spx_map_pyr
hist_set_pyr
```

5.2 Superpixel Feature Processing

After the segmentation of an input image into superpixels using *SLIC* algorithm, our saliency model processes subsequently all features. For each feature it represents individual superpixels by a histogram. The superpixel map and the histogram set enter in the iterative process of the *superpixel Gaussian pyramid* as the first layer.

After the creation of all levels of the Gaussian pyramid, our model compares center and surround layers of the pyramid per pixel. In order to achieve the center-surround differences, we find superpixels which contain compared pixels at the center and surround scale. Then we measure the similarity between the histograms belonging to these selected superpixels using a *histogram matching algorithm*. Such a difference between the layers produces a *feature map* FM (Figure 5.3).

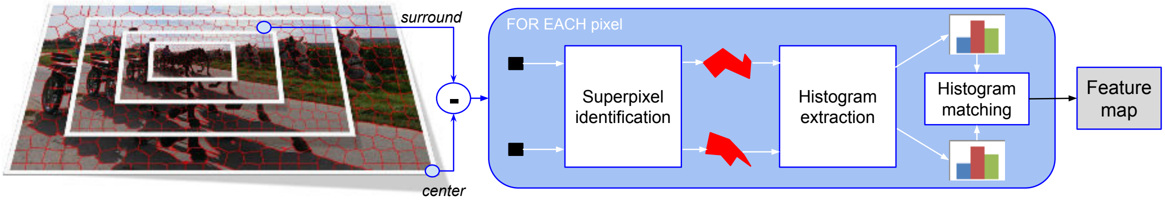


Figure 5.3: Difference between center and surround scales of superpixel Gaussian pyramid.

In general, the procedure to create a feature map consists of 4 steps:

1. superpixel segmentation,
2. representation of superpixels with histograms,
3. creating a superpixel Gaussian pyramid,

4. difference between center and surround pyramid layers.

5.2.1 Intensity

In order to analyse the intensity feature, an image is simply converted to *grayscale*. Superpixels are described by 1D histograms with 128 bins. Histogram comparison of Gaussian pyramid layers is based on the *correlation* method described by Equation 4.4. A value of each pixel in the resulting feature map is computed as follows:

$$FM_I(x,y) = 1 - abs(d_{correl}(H_c(x,y), H_s(x,y)), \tag{5.2}$$

where d_{correl} is a correlation coefficient that ranges from -1 to 1, $H_i(x,y)$ is a histogram of a superpixel at the scale i which contains a pixel at position $[x,y]$ and subscripts c and s denote a center and a surround scales.

5.2.2 Colour

An input image is at first converted to 4-channel *RGBY* (Red Green Blue Yellow) colour space for colour feature processing. Colour conversion is the same as in (Itti et al., 1998):

$$R = r - (g+b)/2, \tag{5.3}$$

$$G = g - (r+b)/2, \tag{5.4}$$

$$B = b - (r+g)/2, \tag{5.5}$$

$$Y = (r+g)/2 - |r-g|/2 - b, \tag{5.6}$$

where r, g and b are red, green and blue colour channels of the origin image. Each colour channel is defined by its 1D histogram with 128 bins.

The process also implements the *Opponent-Process Theory of Colour Vision*, due to which it works with two opponent colour pairs - RG and BY. The center-surround is expressed as the difference between *mean colour values* of compared superpixels for both opponent pairs:

$$FM_{RG}(x,y) = norm(abs(mean(H_{diff_R}(x,y)) - mean(H_{diff_G}(x,y)))), \tag{5.7}$$

$$FM_{BY}(x,y) = norm(abs(mean(H_{diff_B}(x,y)) - mean(H_{diff_Y}(x,y)))), \tag{5.8}$$

where $norm$ normalises values within the interval $\langle 0,1 \rangle$, $mean$ is the mean colour and $H_{diff_{COL}}$ is a difference of histograms of colour channel COL at a center c and a surround s scale defined as:

$$H_{diff_{COL}}(x,y) = abs(H_{COL_c}(x,y) - H_{COL_s}(x,y)). \tag{5.9}$$

5.2.3 Orientation

The processing of orientation feature starts with the image conversion to *grayscale* colour space. Each superpixel is characterised with a *histogram of oriented gradients* with 9 bins. Single channel input image I is filtered by the *Sobel* kernel in both directions (Bradski – Kaehler, 2008):

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I, \tag{5.10}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I, \tag{5.11}$$

where $*$ denotes the convolution, G_x and G_y capture horizontal and vertical changes.

Image gradients are described by:

- the gradient *magnitude*:

$$\mathbf{G} = \sqrt{G_x^2 + G_y^2}, \tag{5.12}$$

- the gradient *direction*:

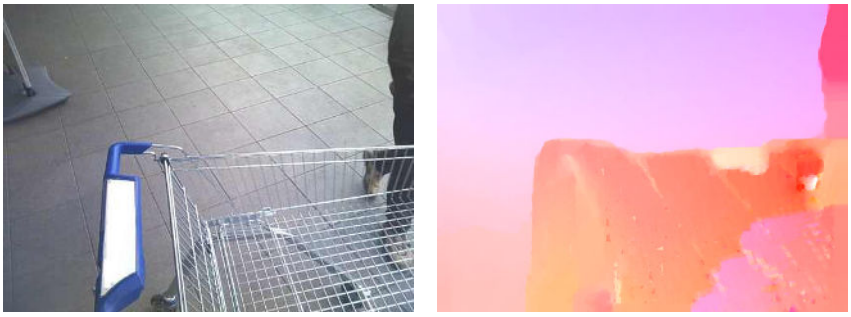
$$\Theta = \arctan(G_y, G_x). \tag{5.13}$$

Gradient directions of each superpixel are split into 9 angle intervals. Each gradient pixel in the histogram of oriented gradient has a weight proportional to its gradient magnitude. Orientation differences are computed with the same *correlation*-based method as in intensity feature maps:

$$FM_O(x,y) = 1 - abs(d_{correl}(H_c(x,y), H_s(x,y))). \tag{5.14}$$

5.2.4 Motion

Motion processing requires 2-channel dense *optical flow maps* characterising the angle and the magnitude of flow vectors (Figure 5.4). Hue and saturation components of the flow map encode a flow vector with a *colour wheel* in Figure 5.5.



(a) Video frame.

(b) Optical flow map (hue = orientation, saturation = magnitude).

Figure 5.4: Optical flow map characterises the angle and the magnitude of optical flow vectors.

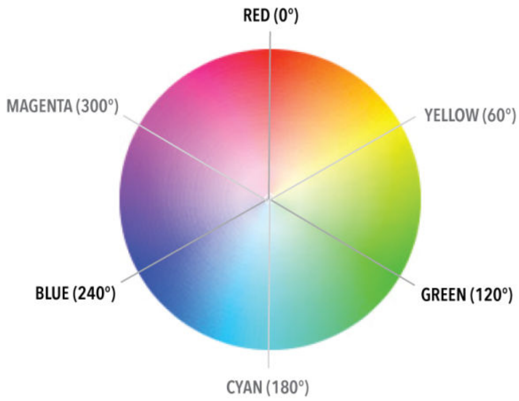


Figure 5.5: Colour wheel decodes a hue-saturation value into a flow vector[1].

Using the flow maps each superpixel is represented by a *histogram of flow orientations* H_o and a *histogram of flow magnitudes* H_m, both with 90 bins. In order to compare superpixels on different pyramid layers, each superpixel is characterised by a flow vector with 2 parameters – *orientation* φ and *magnitude* r:

$$\mathbf{v} = [\overline{\varphi}_{H_o}, \overline{r}_{H_m}], \tag{5.15}$$

where $\overline{x}_H$ denotes the mean value of a histogram H. Let $\mathbf{v}_c(x, y)$ and $\mathbf{v}_s(x, y)$ be flow vectors of superpixels on a center and surround pyramid layer at location (x, y), a value of a motion feature map can be expressed as the magnitude of the vector difference:

$$FM_M(x, y) = \|\mathbf{v}_s(x, y) - \mathbf{v}_c(x, y)\|. \tag{5.16}$$

Using the *law of cosines*[2], the following equation is obtained:

$$FM_M(x, y) = \sqrt{r_s^2 + r_c^2 - 2r_s r_c \cos\gamma}, \tag{5.17}$$

where r_i is the magnitude of the flow vector v_i at (x, y) scaled into the range $\langle 0, 0.5\rangle$ and γ indicates the angle between the corresponding vectors, as shown in Figure 5.6. The highest value of a feature map occurs when vectors at the same location have the opposite directions and the maximum velocity magnitudes.

5.3 Spatial Saliency Map

In the next phase, all extracted static feature maps are combined into 3 *conspicuous maps* CM of intensity, colour and orientation for each used region size in SLIC algorithm:

$$CM_i = \sum_j \mathcal{N}(FM_{i_j}). \tag{5.18}$$

[1] `http://demosthenes.info/assets/images/hsl-color-wheel.png.pagespeed.ce.IF6-EXzipy.png`

[2] WEISSTEIN, E. W. Law of Cosines. Available from MathWorld – A Wolfram Web Resource: `http://mathworld.wolfram.com/LawofCosines.html`

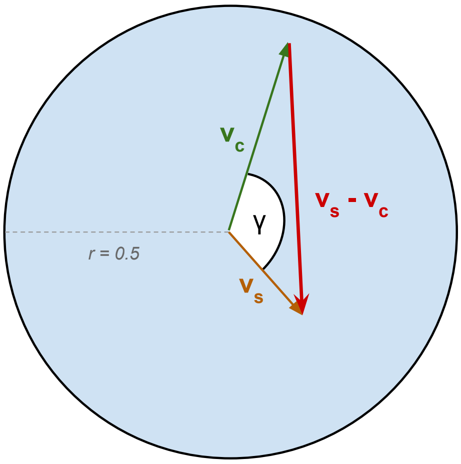


Figure 5.6: Motion difference between center c and surround s scales of a pyramid.

First, small values of all feature maps are set to zero. Such maps are modified using a *normalisation operator* $\mathcal{N}$. The operator searches for local maxima in rectangular image regions. An input is then multiplied by $\mathcal{N} = (M - m)^2$, where M is the global maximum and m is the average of all local maxima in image blocks.

Finally, three conspicuous maps are linearly combined to create the resulting topography representation of image saliency called a *saliency map* SM:

$$SM = \sum_{reg}^{\{15;30\}} \left(0,45 * \mathcal{N}(CM_{I_{reg}}) + 0,3 * \mathcal{N}(CM_{C_{reg}}) + 0,25 * \mathcal{N}(CM_{O_{reg}})\right), \quad (5.19)$$

where reg is a region size used in SLIC algorithm, $\mathcal{N}$ is a normalisation operator, I is an intensity, C is a colour and O is an orientation feature. The position with the highest pixel value in the saliency map is the location with the highest bottom-up saliency. Weighting factors resulted from experiments as the best choice.

Figure 5.7 is an example of combining all conspicuous maps of intensity, colour and orientation into a single saliency map (Figure 5.7(h)). The most salient location is marked with a green circle in Figure 5.7(i).

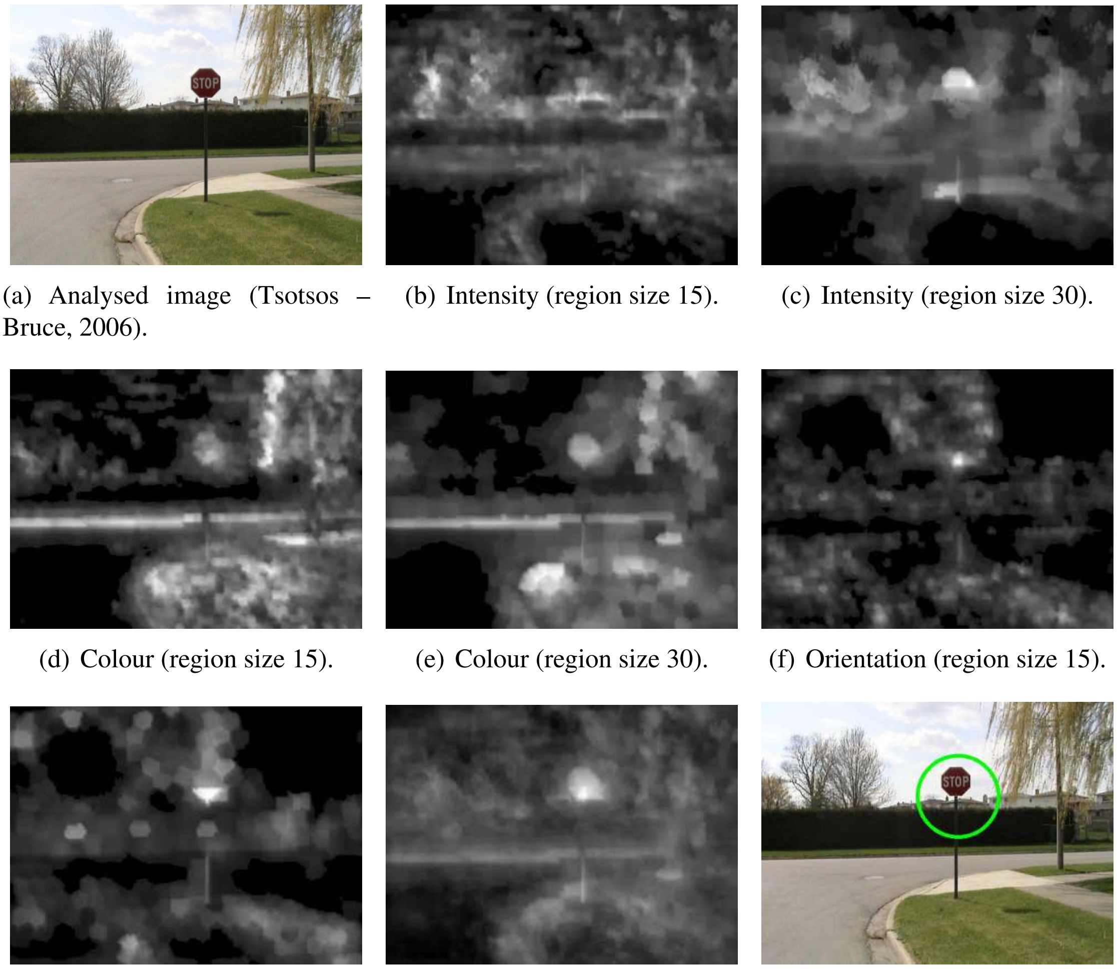
The whole hierarchical superpixel-based saliency model is briefly summarised in Figure 5.8.

WTA algorithm of our model is performed by the suppressing of the circular neighbourhood of the most salient location in the final saliency map and the searching for a new location with the maximum saliency.

5.4 Motion Innovation Map

Motion feature maps represent motion saliency of a current video frame. To determine dynamic changes in a scene, we build a *motion innovation map* considering not only a single optical flow map but also a subsequence of several previous flow maps.

Temporal changes in motion are detected using a *motion memory*, which is updated with



(a) Analysed image (Tsotsos – Bruce, 2006).

(b) Intensity (region size 15).

(c) Intensity (region size 30).

(d) Colour (region size 15).

(e) Colour (region size 30).

(f) Orientation (region size 15).

(g) Orientation (region size 30).

(h) Saliency map.

(i) The most salient location.

Figure 5.7: Combining conspicuous maps into a saliency map.

each incoming optical flow map. The update can be defined as:

$$MEM_{t+1} = (1-\eta)MEM_t + \eta o_{t+1}, \tag{5.20}$$

where MEM_t represents a motion memory at time t, o_t is an optical flow map and a *learning rate* $\eta = 0.05$.

Before the memory update, an actual flow map is compared with the motion memory (Figure 5.9).

Each pixel of both compared images represents a vector $v = [\varphi, r]$, where φ and r define magnitude and orientation at the pixel position.

A pixel-by-pixel comparison between the memory and the flow map is analogous to the motion difference in a motion feature map (Subsection 5.2.4). Motion innovation is then defined by the following form:

$$IM_M(x,y) = \|\mathbf{v}_{MAP_t}(x,y) - \mathbf{v}_{m_t}(x,y)\|. \tag{5.21}$$

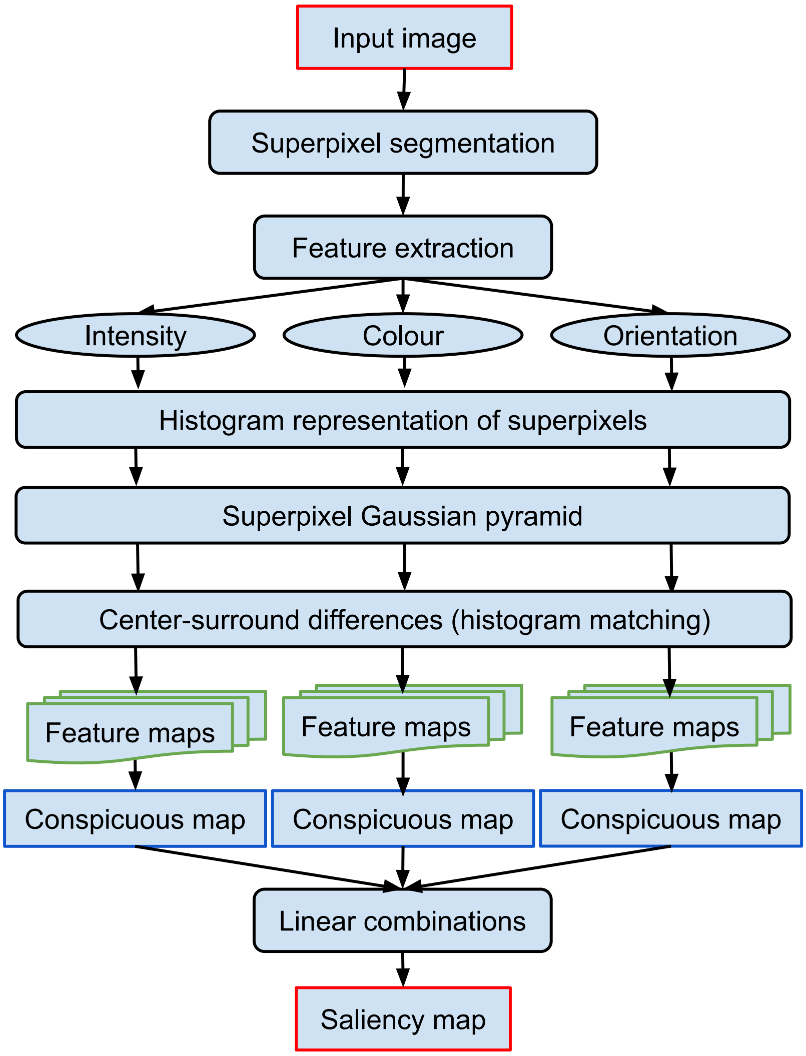


Figure 5.8: Scheme of the static form of our hierarchical superpixel-based saliency model.

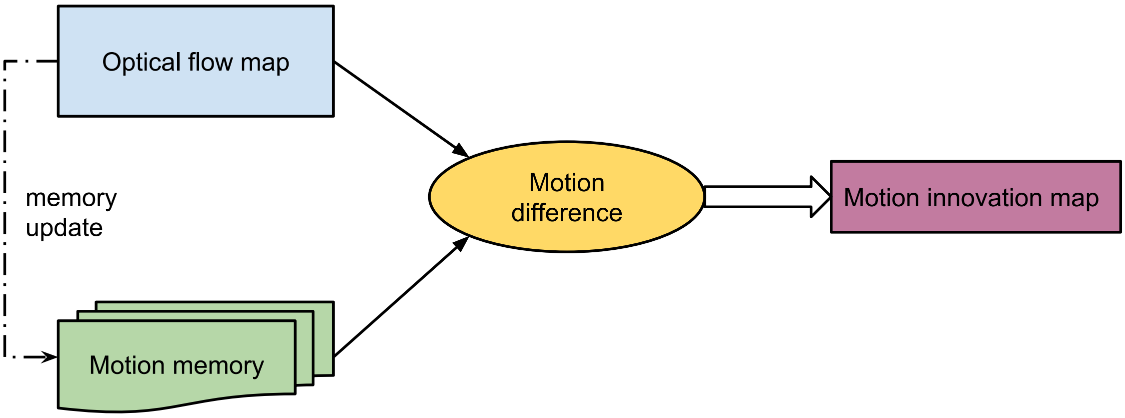


Figure 5.9: Motion innovation.

5.5 Spatiotemporal Saliency Map

Fusion of a spatial saliency map SM_S described in Section 5.3 and a temporal saliency map SM_T created from motion feature maps results in a spatiotemporal map:

$$SM = (1 - \lambda)SM_S + \lambda SM_T, \tag{5.22}$$

where $SM_T = \sum_{reg}^{\{15;30\}} \mathcal{N}(CM_{I_{reg}})$, $CM_{I_{reg}}$ is a motion conspicuous map and λ denotes a motion saliency rate. Considering temporal changes in a video sequence obtained from a motion innovation map IM_M, the equation for a final saliency map has the following form:

$$SM = (1 - \lambda)SM_S + \frac{\lambda}{2}SM_T + \frac{\lambda}{2}IM_M. \tag{5.23}$$

This spatiotemporal fusion is illustrated in Figure 5.10.

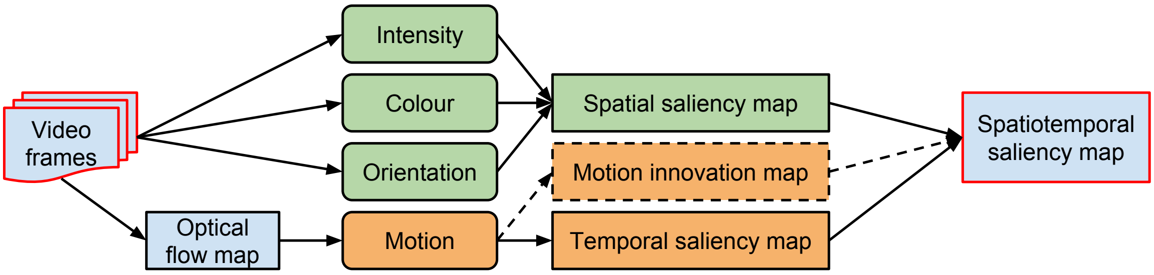
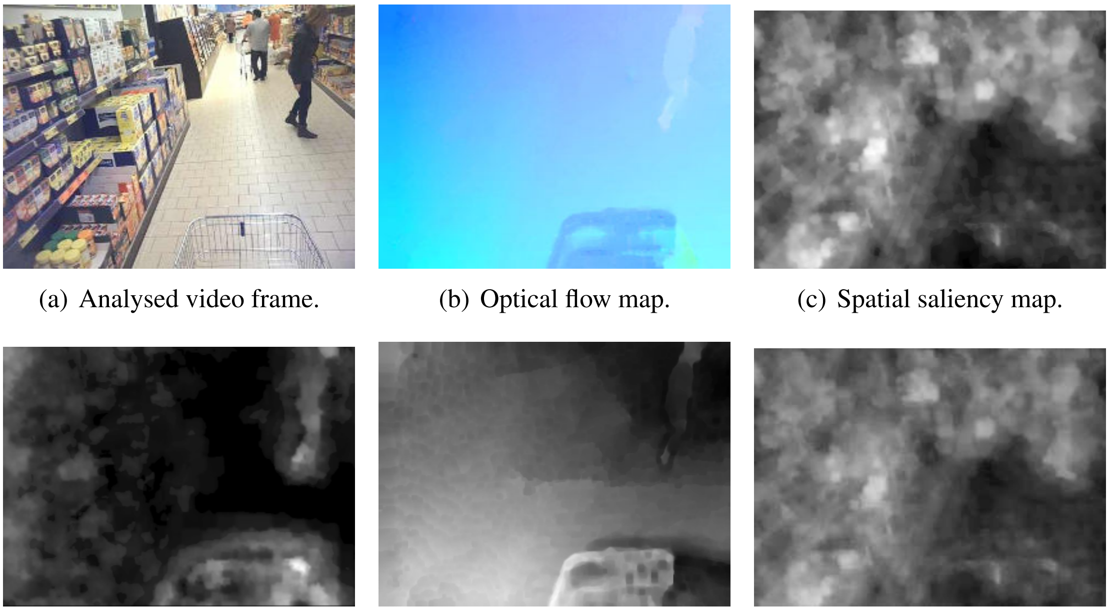


Figure 5.10: General scheme of spatiotemporal saliency model.

An example of a spatiotemporal saliency map with innovation is included in Figure 5.11.



(a) Analysed video frame.

(b) Optical flow map.

(c) Spatial saliency map.

(d) Temporal saliency map.

(e) Motion innovation map.

(f) Spatiotemporal saliency map.

Figure 5.11: Fusion of saliency maps into a spatiotemporal saliency map with motion innovation using a motion rate $\lambda = 0.25$.

5.6 Implementation Details

The algorithms of our spatial and spatiotemporal saliency model are implemented in C++ language using *OpenCV*[3] library. For superpixel segmentation with SLIC the implementation from *VlFeat*[4] library has been used.

The architecture of the model inspired by (Itti et al., 1998) as well as our superpixel-based solution is presented by the class diagram in Figure 5.12:

- `IttiHierarchicalModel`: saliency model based on (Itti et al., 1998),
- `SuperpixelHierarchicalModel`: hierarchical superpixel-based saliency model.

[3] `http://opencv.org/`

[4] `http://www.vlfeat.org/index.html`

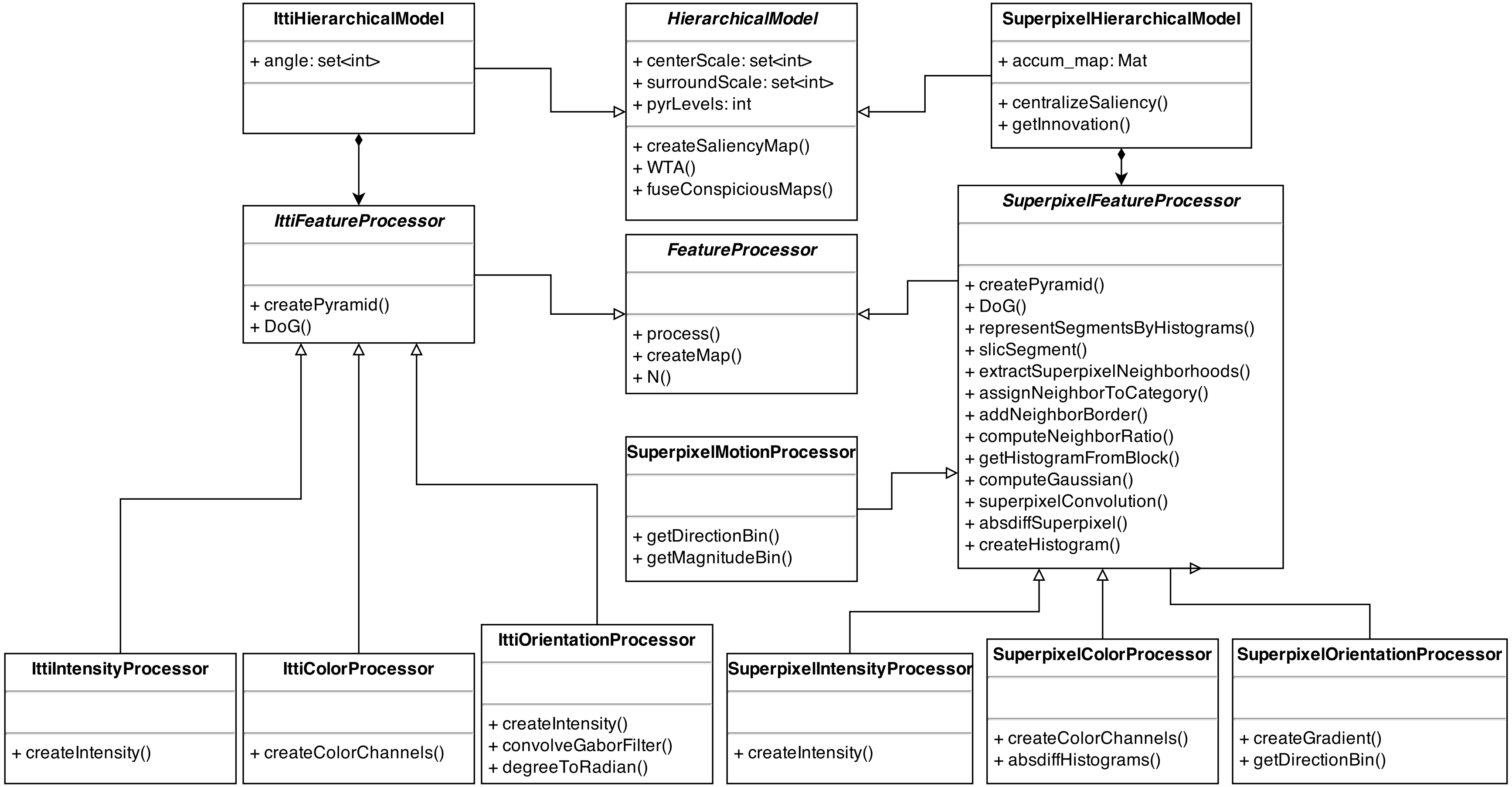


Figure 5.12: Architecture of the saliency model. The model based on (Itti et al., 1998) consists of classes with prefix `Itti-` and our superpixel-based classes of the saliency model start with prefix `Superpixel-`.

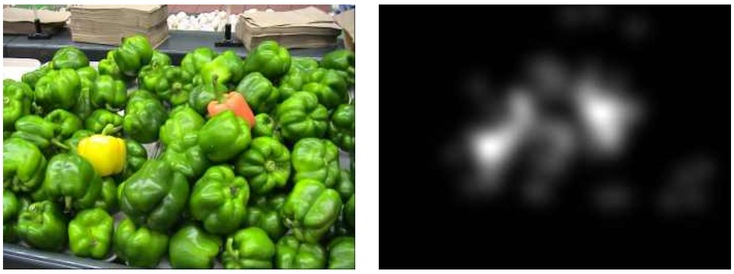
Chapter 6

Evaluation and Discussion

A spatial and a spatiotemporal type of our superpixel-based saliency method described in Chapter 4 are evaluated on an image dataset and a video dataset with eye tracking data. The performance is compared with the standard hierarchical model based on (Itti et al., 1998).

6.1 Spatial Saliency Model

Our spatial saliency model was tested on a publicly available dataset[1] called *Toronto* (Tsotsos – Bruce, 2006). It contains 120 colour images of 681×511 size obtained from natural indoor and outdoor scenes. Images were freely observed by 20 subjects for 4 seconds. Each image is accompanied with eye tracking data as well as a density fixation map produced from the tracking data (Figure 6.1).



(a) Source image.

(b) Density map produced from eye tracking data.

Figure 6.1: Toronto dataset (Tsotsos – Bruce, 2006).

In order to measure the accuracy of a superpixel-based and a standard saliency model, we use four different evaluation methods. Our model is based on principles of (Itti et al., 1998), due to which we do not expect a marked increase in performance.

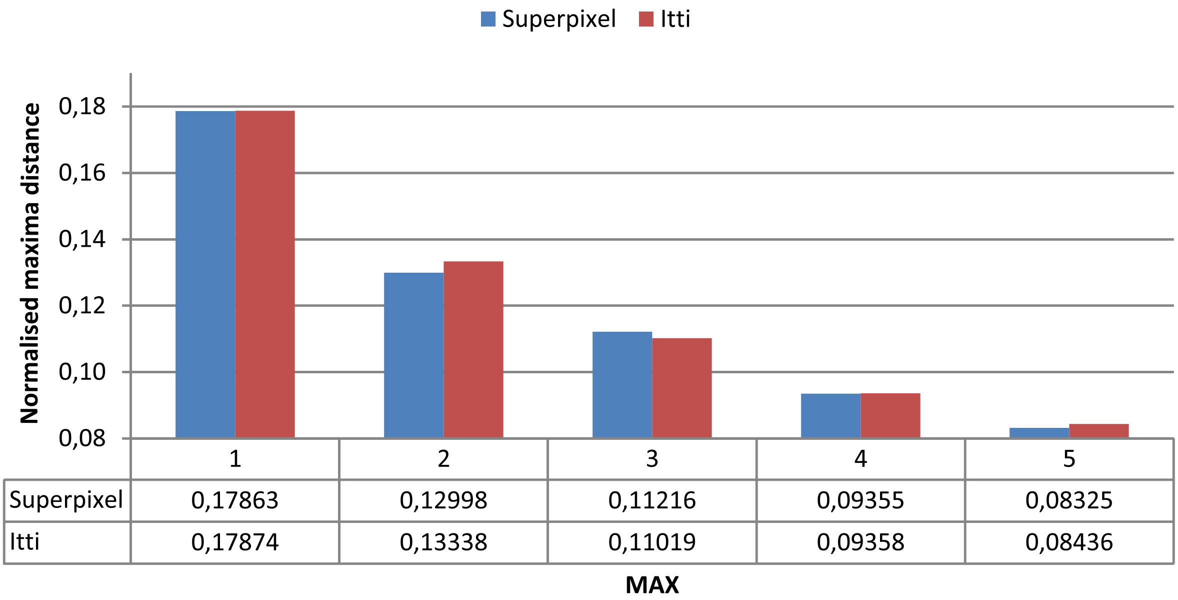
The first metric is our own method called the *maxima distance*. It measures the Euclidean distance between the most salient location on a saliency map and the location with the highest density on a corresponding fixation map. In order to compute the distance, we take iteratively

[1] http://www-sop.inria.fr/members/Neil.Bruce/

into account together with the first maximum value of the fixation map also the second, the third, the fourth and finally the fifth highest value as follows:

$$d_{MAX_i} = \min_i \Big(d(\max_i(SM), \max_i(FM)) \Big), \tag{6.1}$$

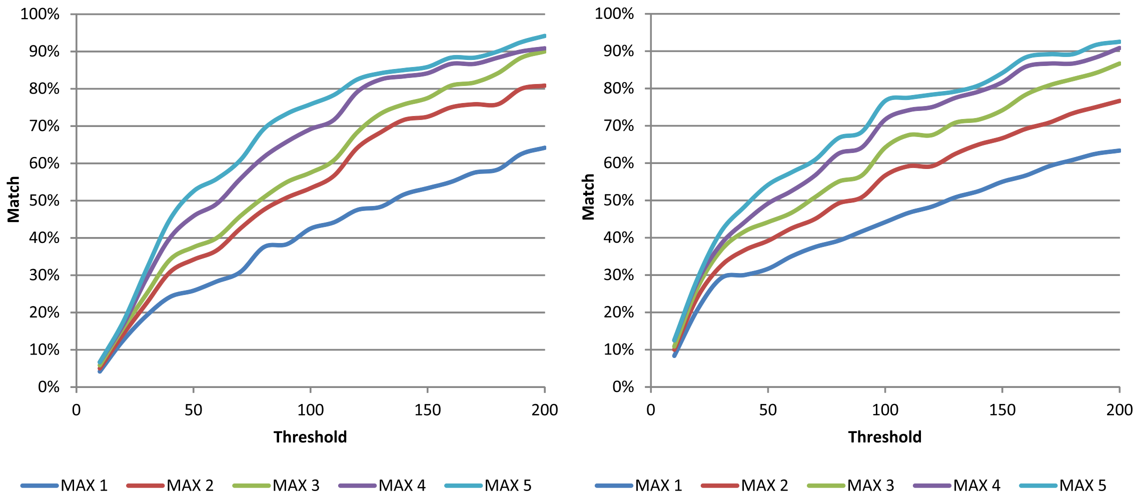
where SM is a saliency map, FM is a fixation map and $\max_i$ is the i^{th} maximum value. The results are visualised in Figure 6.2.



	1	2	3	4	5
Superpixel	0,17863	0,12998	0,11216	0,09355	0,08325
Itti	0,17874	0,13338	0,11019	0,09358	0,08436

Figure 6.2: Normalised maxima distance.

If the maxima distance is lower than a given threshold, it is considered as a match. Figure 6.3 represents the tradeoff between the number of matches and a threshold for both saliency models.



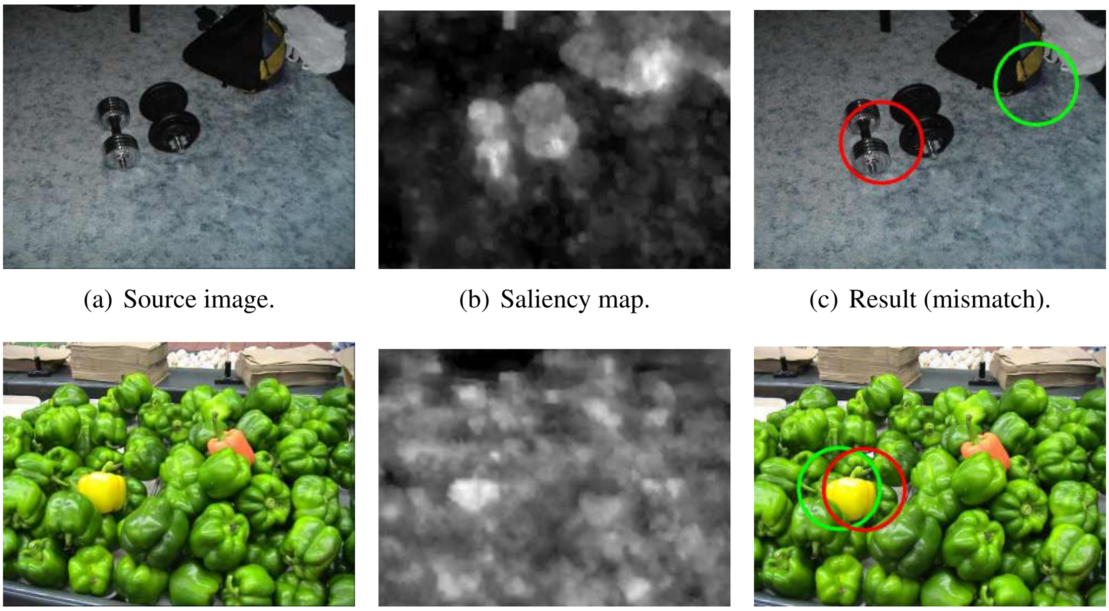
(a) Superpixel-based method.

(b) Standard method based on (Itti et al., 1998).

Figure 6.3: Maxima distance matches. Distances below a threshold are considered as matches.

The results of the maxima distance are summarised in Table 6.1. If the highest values of saliency and fixation maps are only considered and the threshold is adjusted to 100 px, a novel superpixel-based model achieves 42.5% matches on the dataset. The total Euclidean distance produced by our model is 18250 px what is very similar to the results of a standard hierarchical model inspired by (Itti et al., 1998) – 44.17% accuracy and the total distance of 18261 px. Using two maxima of a fixation map and a threshold of 120 px the superpixel-based method produces slightly better results.

For visualisation purposes we label the most salient location in the resulting saliency map with a green circle and the location with highest value in a fixation map with a red circle as shown in Figure 6.4.



(a) Source image. (b) Saliency map. (c) Result (mismatch).

(d) Source image. (e) Saliency map. (f) Result (match).

Figure 6.4: Superpixel-based saliency map of the superpixel model. A green circle represents the most salient location and a red circle denotes the most viewed location. If the distance between the maximum value pixels is lower than a threshold, it is considered as a match.

The superpixel-based model is nearly always successful in case of simple images with a single dominant object. Examples of such scenes are included in Figure 6.5. However, some images may be classified due to the bigger object size as a mismatch even if a saliency map finds properly the most viewed object, for example Figure 6.5(i).

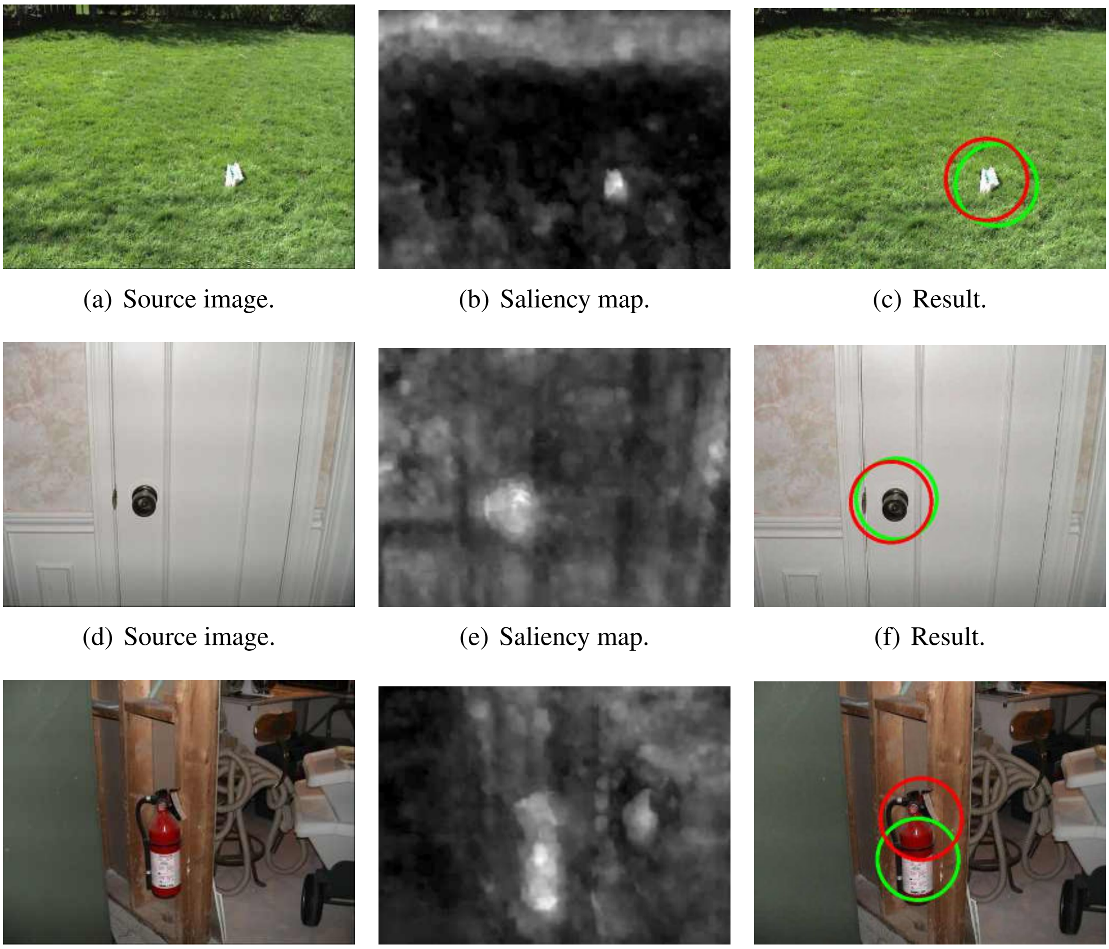
The accuracy of the model decreases with the complexity of a given image as shown in Figure 6.6. Combining visual features of images with multiple objects into a saliency map may cause some mismatches.

Wrong classification of the most salient location is often the result of absenting top-down attention in this hierarchical model. The top-down part of our saliency can affect the conspicuousness of objects more significantly in such complex scenes than in simple ones. Human faces, texts or traffic signs may not be salient in terms of our model (Figure 6.6(i)). Despite of that fact, top-down attention is focused on these objects in most cases.

Some images have the most attentive region in their centre even without any unique visual characteristics instead of the focus on more dominant objects. An example of such a scene is represented by Figure 6.7.

The adjustment of a superpixel region size, a minimum region size and a regularisation coefficient used in SLIC is crucial in the model. If the selected parameters are insufficient, the resulting saliency map may not properly detect conspicuous objects with a tiny size (Figure 6.8).

Despite of that, there are scenes where the superpixel-based saliency model outperforms the standard one. The main reason is the fact that the superpixel segmentation of an input which can correspond to object edges. Examples of a superpixel-based saliency map achieving better results than a standard one are presented in Figure 6.9.



(a) Source image. (b) Saliency map. (c) Result.

(d) Source image. (e) Saliency map. (f) Result.

(g) Source image. (h) Saliency map. (i) Result.

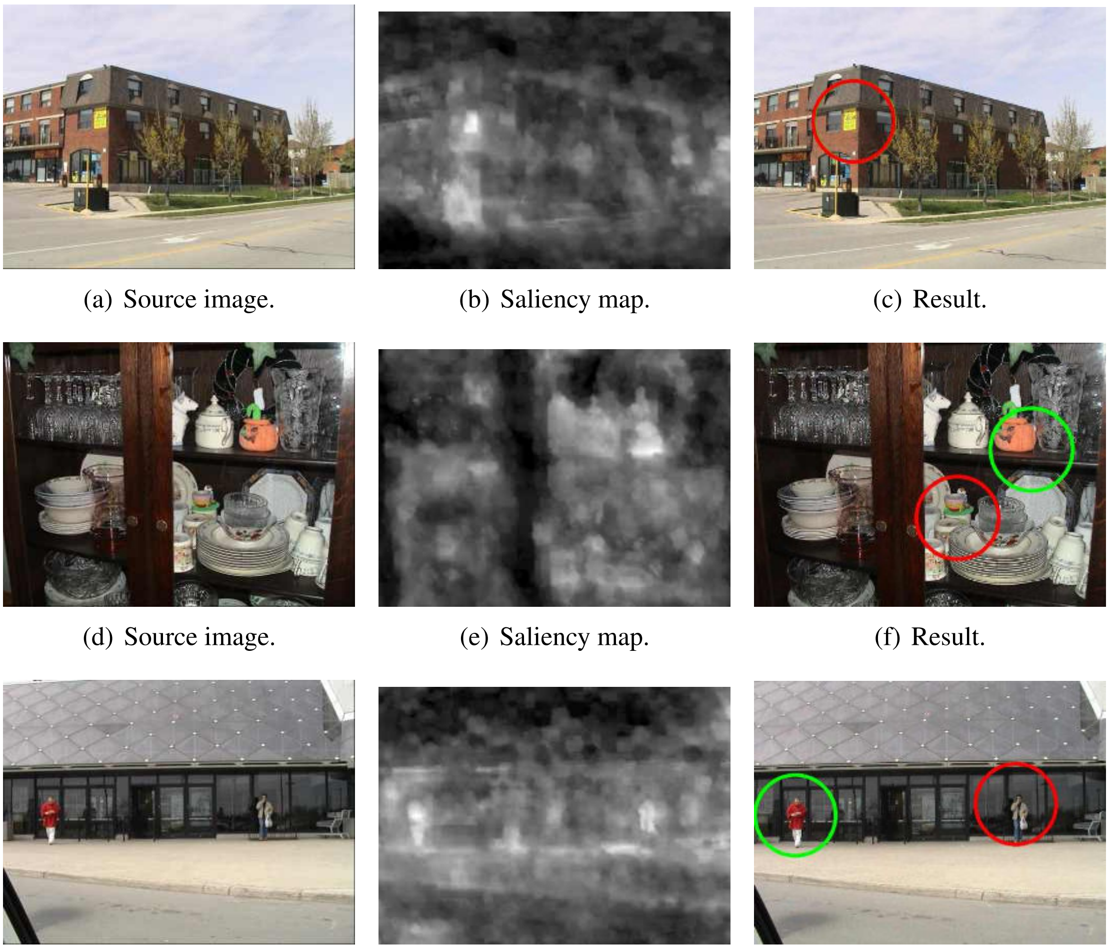
Figure 6.5: Saliency maps created from simple scenes with a single conspicuous object. The most salient and the most viewed locations are marked by green and red circles.

We have also evaluated both models using standard methods. The *similarity metric* (Riche et al., 2013) represents the degree of similarity between the normalised distributions of a saliency map SM and a fixation map FM:

$$S = \sum_{x} \min(SM(x), FM(x)), \tag{6.2}$$

where $\sum_x SM(x) = \sum_x FM(x) = 1$. A similarity score ranges from 0 to 1. If the distributions do not overlap, the score equals to zero.

To compare the models, we have also used the *Kullback-Leibler (KL) divergence* (Riche

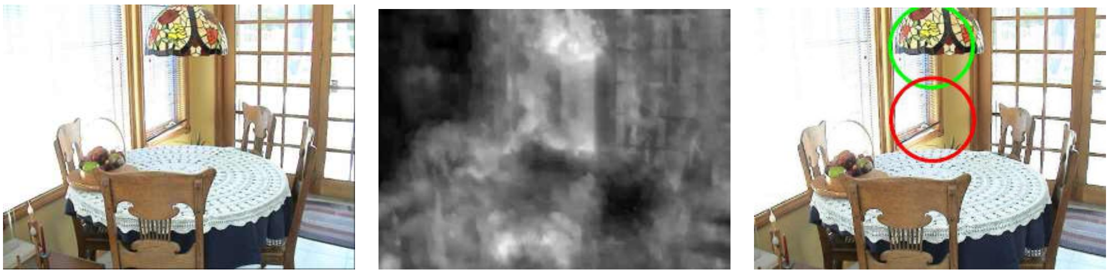


(a) Source image. (b) Saliency map. (c) Result.

(d) Source image. (e) Saliency map. (f) Result.

(g) Source image. (h) Saliency map. (i) Result.

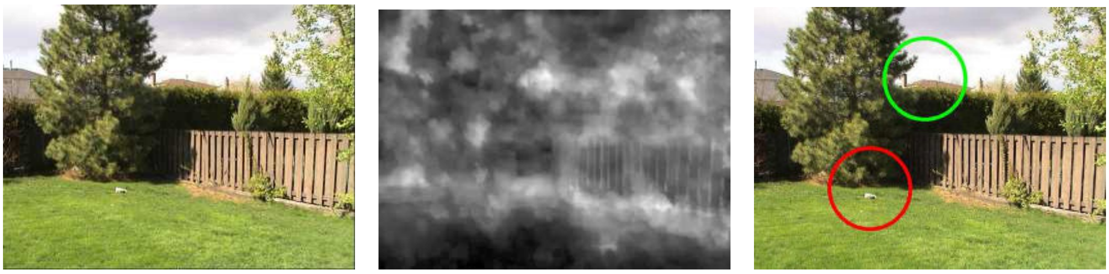
Figure 6.6: Saliency maps created from complex scenes with multiple objects. The most salient and the most viewed locations are marked by green and red circles.



(a) Source image. (b) Saliency map. (c) Result.

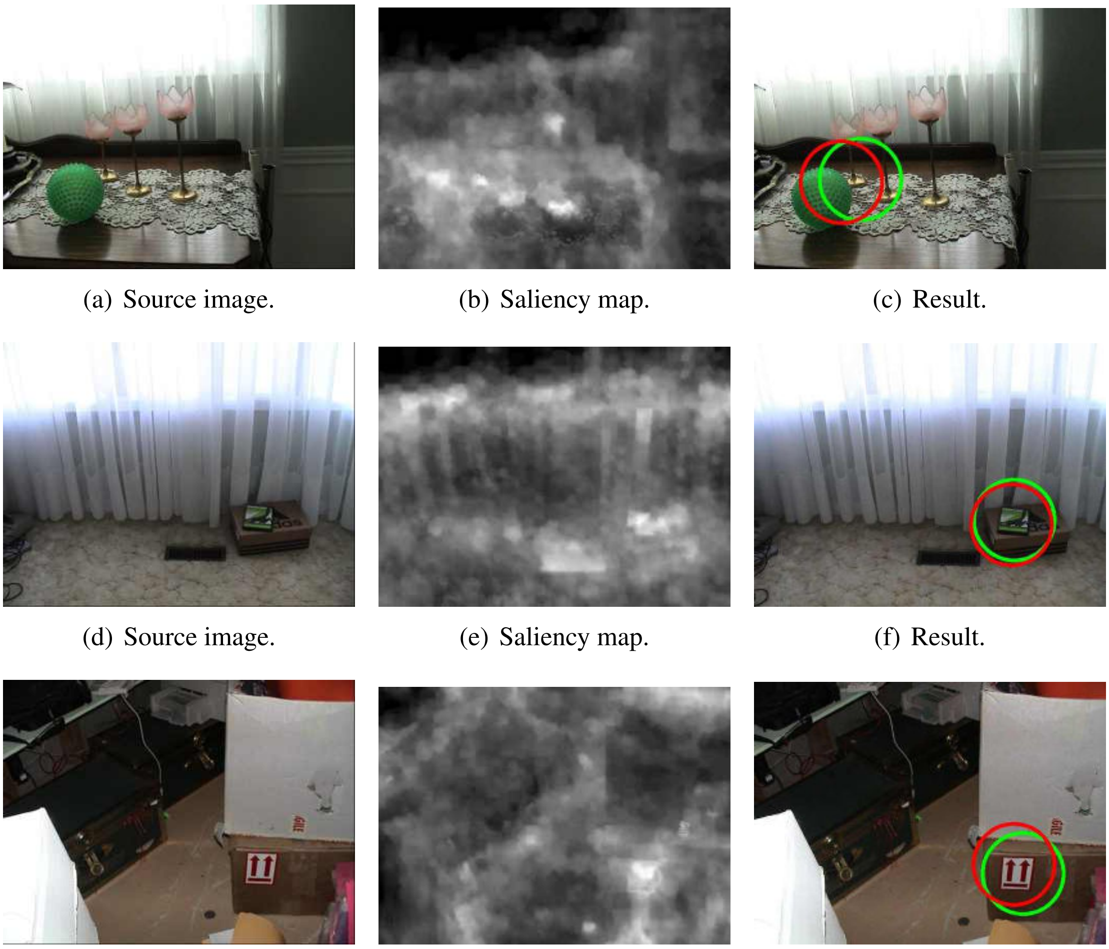
Figure 6.7: Saliency maps from scenes where attention is focused on the centre of a given image. The most salient and the most viewed locations are marked by green and red circles.

et al., 2013; Le Meur – Baccino, 2013). The KL divergence is a nonnegative value that indicates the information lost when a fixation map FM distribution is approximated by a saliency map SM. The zero KL divergence occurs when the probability distributions are



(a) Source image. (b) Saliency map. (c) Result.

Figure 6.8: Saliency maps from scenes where parameters used in SLIC do not conform to a dominant object in an input image. The most salient and the most viewed locations are marked by green and red circles.



(a) Source image. (b) Saliency map. (c) Result.

(d) Source image. (e) Saliency map. (f) Result.

(g) Source image. (h) Saliency map. (i) Result.

Figure 6.9: Saliency maps from scenes where the superpixel-based model outperforms the standard one. The most salient and the most viewed locations are marked by green and red circles.

equal. It is computed by the following form:

$$KL_{div} = \sum_{x} FM_{norm}(x) * \log\left(\frac{FM_{norm}(x)}{SM_{norm}(x) + \epsilon} + \epsilon\right), \tag{6.3}$$

where $SM_{norm}(x) = \frac{SM(x)}{\sum_x SM(x)+\epsilon}$, $FM_{norm}(x) = \frac{FM(x)}{\sum_x FM(x)+\epsilon}$ and ϵ is a small constant to avoid logarithm and division by zero.

The results of the similarity metric as well as the KL divergence of the superpixel-based model are close to the results of the standard model (Figure 6.10 and 6.11). The median and the average of both models for Toronto dataset are shown in Table 6.1.

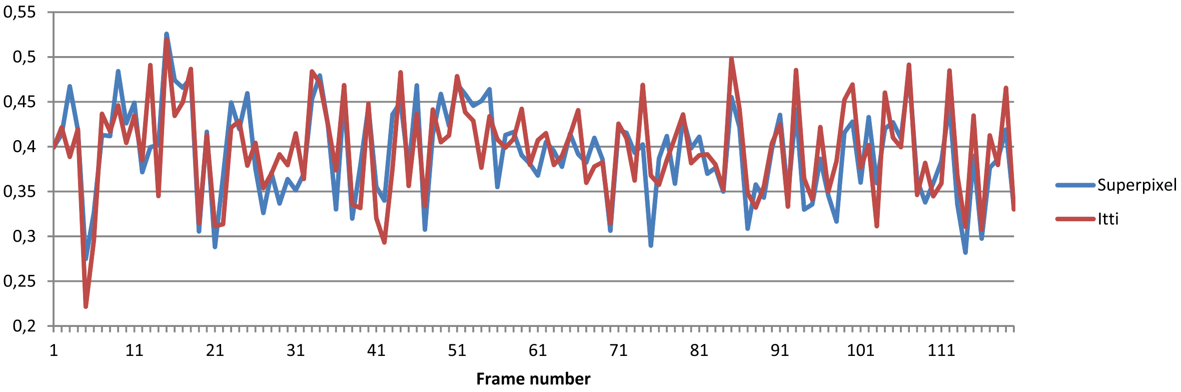


Figure 6.10: Similarity metric.

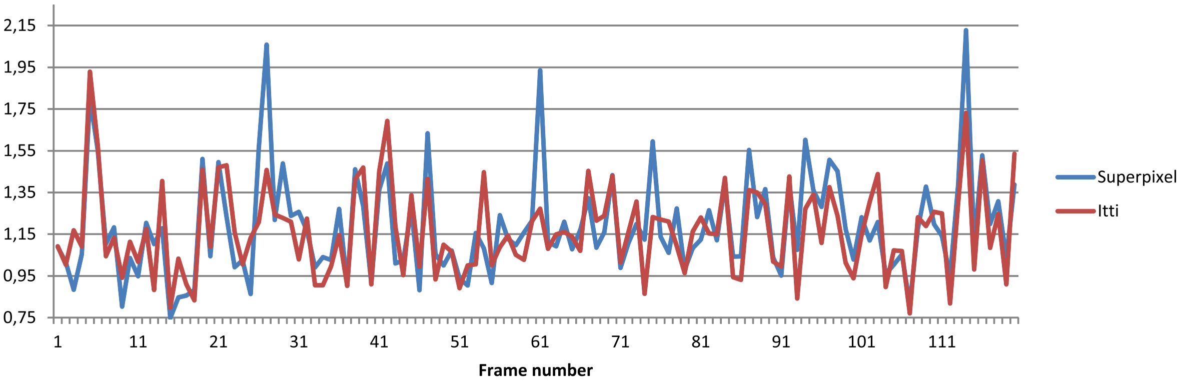


Figure 6.11: Kullback-Leibler divergence.

Table 6.1: Comparison of a *standard* hierarchical saliency model based on (Itti et al., 1998) and a novel *superpixel*-based model on Toronto dataset (Tsotsos – Bruce, 2006). The saliency models are evaluated using the maxima distance, the similarity metric and the KL divergence.

Model	**Maxima distance**			**Similarity**		**KL-div.**	
	MAX_1	MAX_1, *th=100 px*	MAX_2, *th=120 px*	*Avg.*	*Med.*	*Avg.*	*Med.*
Standard	18261 px	44.17 %	59.17 %	0.3969	0.3992	1.1649	1.1484
Superpixel	18250 px	42.50 %	64.17 %	0.3939	0.3979	1.1870	1.1334

The performance of saliency models are also compared by plotting a graph called the *receiver operating characteristic (ROC) curve*. The ROC curve (Le Meur – Baccino, 2013) represents the tradeoff between the *true positive rate* and the *false positive rate*.

The true positive rate also called the *sensitivity* is computed as $TPR = \frac{TP}{TP+FN}$ and the false positive rate FPR also known as the *fall-out* is defined as $FPR = \frac{FP}{FP+TN} = 1 - SPC$, where TP is true positive, FP is false positive, TN is true negative, FN is false negative and SPC is specificity.

A saliency map is thresholded by a gradually increasing threshold, a fixation map by a constant threshold and the true positive and false positive rates are recorded. Using the rates, the ROC curve is plotted that specifies the sensitivity as a function of fall-out.

The ROC curve represented by Figure 6.12 shows that the superpixel-based method is slightly more sensitive than the standard hierarchical saliency model.

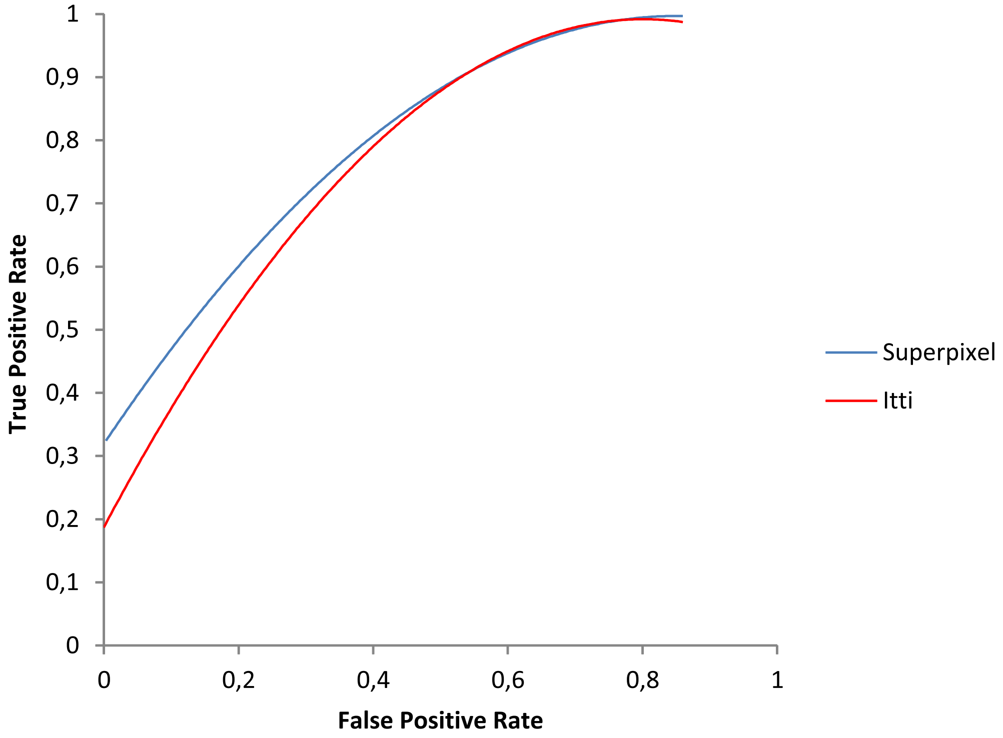


Figure 6.12: ROC curve.

The last metric called *shuffled AUC* (sAUC) is area under the ROC curve, which treats fixations as a positive set and fixations over other images as a negative set (Borji et al., 2013). An AUC score of 0.5 indicates prediction equivalent to chance while a perfect model will score an AUC of 1.

In order to compare the results of our superpixel-based method with state-of-the-art saliency models we use publicly available pre-computed saliency maps of Toronto dataset (Zhang – Sclaroff, 2013)[2]. Saliency maps are smoothed with a Gaussian kernel with variable σ from 0.01 to 0.13 in image width (in steps of 0.01).

The evaluation of saliency models is shown in Table 6.2. Numbers of each model represent sAUC scores and optimal σ values of a Gaussian kernel where it takes its maximum score.

Section 3 contains a short description of all compared saliency models.

The original version of Itti's model (Itti et al., 1998) and Graph-Based Visual Saliency by (Harel et al., 2006) produce worse sAUC results than our novel superpixel-based method with a sAUC score of 0.6515.

[2] http://cs-people.bu.edu/jmzhang/BMS/BMS.html

Table 6.2: Shuffled AUC of saliency models using optimal σ values of a Gaussian kernel (range of $0.01 : 0.01 : 0.13$ in image width). Saliency models were evaluated on Toronto dataset (Tsotsos – Bruce, 2006) and compared with our novel method denoted as SPX. Models are assigned to one of these categories: hierarchical (H), graphical (G), information-theoretic (I), pattern classification (P), spectral analysis (S) and others (O).

Model	Itti (Itti et al., 1998)[3]	Itti2 (Itti et al., 1998)[4]	GBVS (Harel et al., 2006)	AIM (Bruce – Tsotsos, 2005)	Judd (Judd et al., 2009)	HFT (Li et al., 2013)	CAS (Goferman et al., 2012)	SPX
Category	H	H	G	I	P	S	O	H
sAUC	0.6091	0.6575	0.641	0.6805	0.6862	0.6833	0.694	0.6515
σ	0.10	0.01	0.02	0.02	0.03	0.03	0.03	0.05

6.2 Spatiotemporal Saliency Model

Our dynamic extension of a spatial saliency model was evaluated on a private video dataset. This video sequence is provided by the research institute in Austria, *Joanneum Research*[5]. The dataset of 410 video frames was recorded using eye tracking glasses at a shopping mall (mostly one fixation per frame). A viewer was asked to find two particular products in a store. Eye tracking data are supplemented by dense optical flow maps (640×480 size), as shown in Figure 5.4.

Optical flow maps are calculated using a global energy minimising method based on the TV-L1 approach (Zach et al., 2007) technique. The traditional assumption of optical flow that pixel intensities are constant over time may be violated due to the illumination changes in videos. Thus, an extension of the optical flow formula is used for the appropriate calculation (Chambolle – Pock, 2011):

$$\min_{u,v} \int_{\Omega} |Du| \int_{\Omega} |Dv| + \lambda \|\rho(u,v)\|_1, \tag{6.4}$$

where Ω is the image domain, v is the motion field, $\|\mathbf{x}\|$ is the ℓ_1 norm of $\mathbf{x}$, Dv is the distributional derivative and λ is the regularisation coefficient. The optical flow constraint is defined as $\rho(u,v) = I_t + (\nabla I)^T(v - v^0) + \beta u$. I_t is the time derivative, v^0 is the initial field and ∇I denotes the image gradient. The parameter β controls the influence of the illumination term which models the varying illumination by means of an additive function u.

To test this video dataset, a superpixel-based and a standard hierarchical saliency map based

[3] original version of Itti's model

[4] improved version of Itti's model by (Harel et al., 2006)

[5] PALETTA, L. – FERKO, R. Joanneum Research, Graz, Austria. `http://www.joanneum.at/`

on (Itti et al., 1998) are fused with a dynamic superpixel-based saliency map. The fusion into a spatiotemporal map is expressed in Equation 5.22, eventually in Equation 5.23 when a motion innovation map is considered.

As an evaluation metric, the average of saliency values at fixation locations has been used. We have been investigated the effect of changing a motion rate λ in a saliency map (from 0.05 to 0.50 in steps of 0.05). The results compared with spatial saliency modes are visualised in Figure 6.13. Table 6.3 shows optimal motion rates where models take their maximum saliency values at fixation locations.

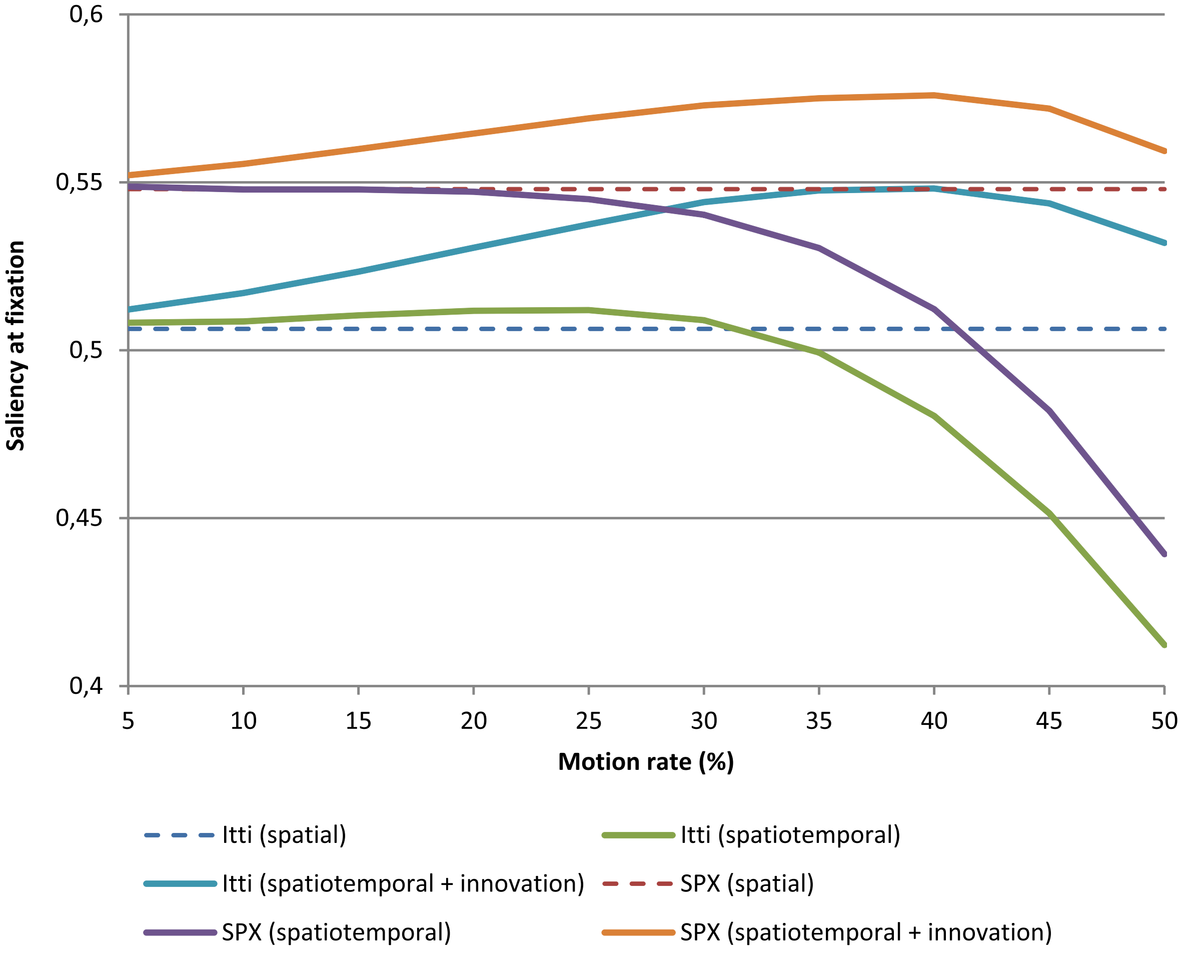


Figure 6.13: Mean saliency value at a fixation location of a private video dataset using different motion rate values λ (from 0.05 to 0.50 in steps of 0.05). Dashed lines represent the performance of spatial saliency models without motion processing. A superpixel-based motion saliency map and a motion innovation map have been added into a novel superpixel-based as well as a standard spatial saliency model inspired by (Itti et al., 1998). When motion innovation is considered, the maximum saliency value of both saliency models is achieved using $\lambda = 0.40$.

Adding motion processing into a saliency model results in saliency increase at fixation locations. A standard hierarchical model with motion saliency maps achieves the maximum saliency value when λ is set to 0.25. When motion innovation is considered, the maximum saliency value of both saliency models is achieved using $\lambda = 0.40$, where the mean saliency value increases from 0.506 up to 0.548. Combining a spatial superpixel-based saliency map with a motion saliency map and a motion innovation map produces the highest increase in

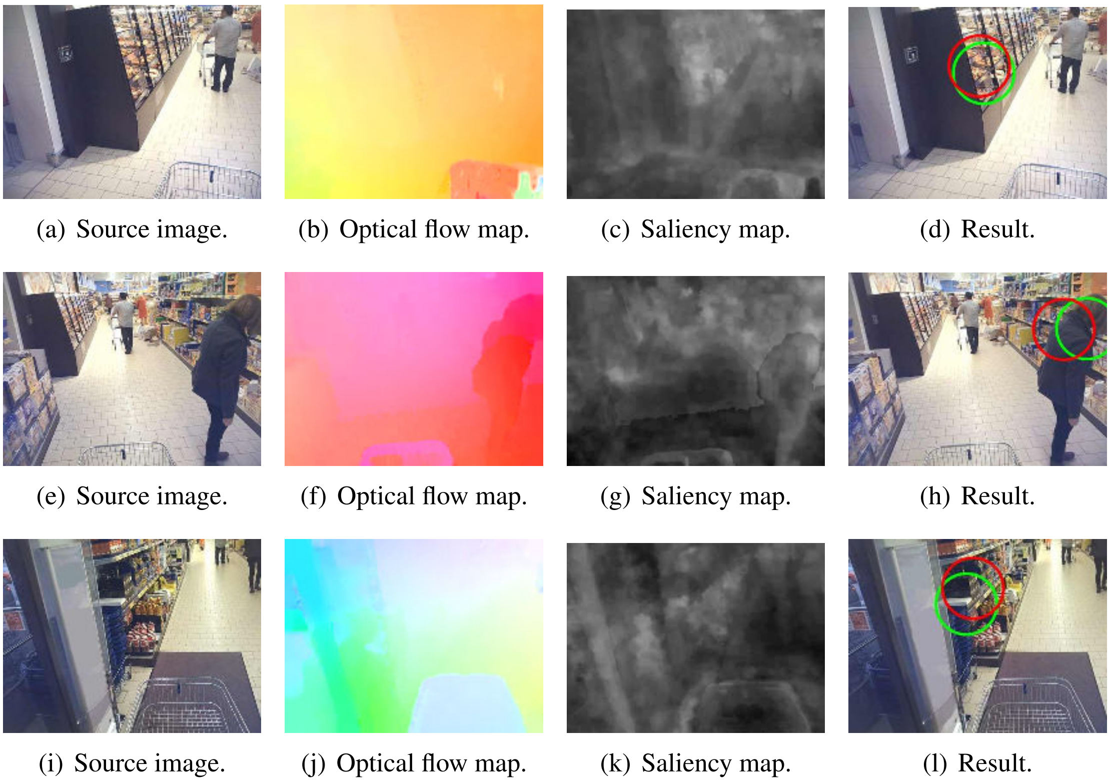
Table 6.3: Maximum saliency values at fixation locations when temporal saliency maps and motion innovation maps are added to a spatial superpixel-based and a standard model inspired by (Itti et al., 1998). Number in brackets shows an optimal motion rate λ where a model takes its maximum value.

Model	**Itti** (λ)	**Superpixel** (λ)
Spatial	0.5063	0.5479
Spatiotemporal	0.5117 (0.25)	0.5487 (0.05)
Spatiotemporal + innovation	0.5481 (0.40)	0.5759 (0.40)

saliency – from 0.548 to 0.576 using the same motion rate value as in the standard model. The novel method achieves better results than the standard method using this metric .

For visualisation purposes we label the most salient location in the resulting saliency map with a green circle and a fixation location with a red circle. Figures included in this section represent saliency maps obtained from our spatiotemporal superpixel-based saliency model with innovation.

Figure 6.14 contains examples of superpixel-based saliency maps with motion saliency and innovation where either static saliency or motion saliency is more dominant. In these examples the most salient locations equal to eye fixations.

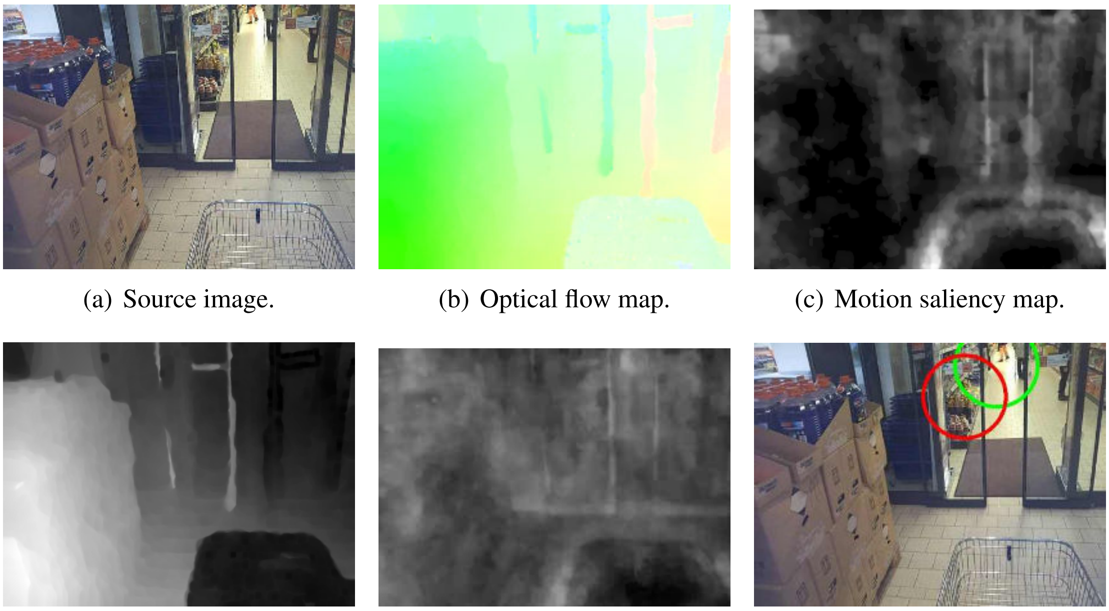

(a) Source image. (b) Optical flow map. (c) Saliency map. (d) Result.

(e) Source image. (f) Optical flow map. (g) Saliency map. (h) Result.

(i) Source image. (j) Optical flow map. (k) Saliency map. (l) Result.

Figure 6.14: Examples where a saliency map correctly predicts fixation locations. The most salient location is marked by a green circle and a fixation by a red circle ($\lambda = 0.5$).

Even though the best performance of our novel method occurs when $\lambda = 0.40$, moving objects do not attract the attention constantly. Directing the gaze to objects in motion occurs mostly at the beginning of object recognition, afterwards the same objects are usually not tracked anymore. Using motion innovation, saliency of a constantly moving trolley at the same location can be reduced, as shown in Figure 6.15. However, our motion innovation map does not track objects, it just learns about the motion direction and magnitude at a given location. Hence, our innovation map considers objects moved to another location or objects that change motion as novelty.

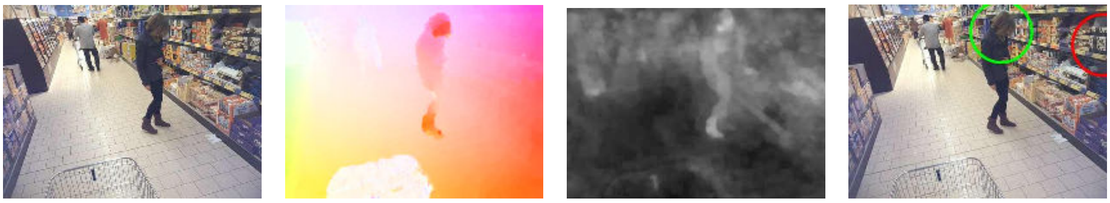


(a) Source image. (b) Optical flow map. (c) Motion saliency map.

(d) Motion innovation map. (e) Saliency map. (f) Result.

Figure 6.15: Motion innovation reduces a final saliency value at locations where motion has not been changed. The most salient location is marked by a green circle and a fixation by a red circle ($\lambda = 0.5$).

Due to the task of looking for some products in a store, top-down attention suppresses motion saliency and directs the gaze to static parts of a scene – shelves. Because of that, motion in this dataset is less important than static saliency (Figure 6.16).

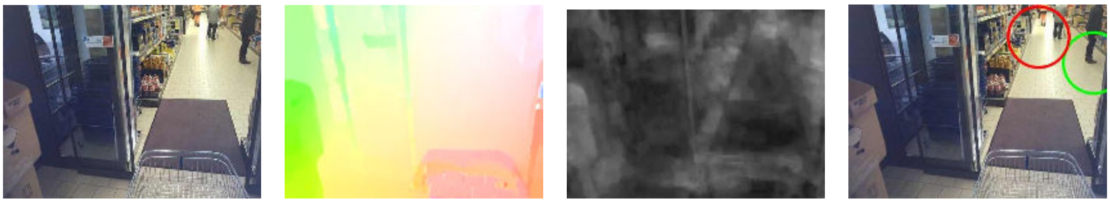


(a) Source image. (b) Optical flow map. (c) Saliency map. (d) Result.

Figure 6.16: Top-down attention directs the gaze to shelves. The most salient location is marked by a green circle and a fixation by a red circle ($\lambda = 0.5$).

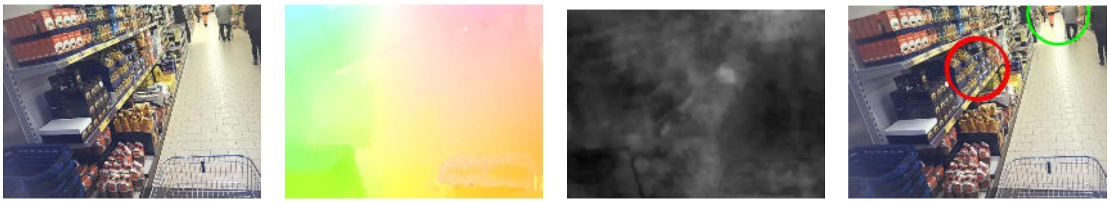
Figure 6.17 represents a situation when there is no distinct maximum in a saliency map and attention remains in a central position.

Using an optical flow map in Figure 6.17 we cannot detect distant movements of person at the top of a frame. Optical flow is an approximate 2D representation of 3D motion. Using the depth information motion processing could scale the flow magnitude according the distance from a viewer and attention prediction would be more precise. Objects nearer to a viewer are more attentive than distant parts of a scene, as shown in Figure 6.18.



(a) Source image. (b) Optical flow map. (c) Saliency map. (d) Result.

Figure 6.17: In case of no dominant, visual stimuli attention remains in a central position. The most salient location is marked by a green circle and a fixation by a red circle ($\lambda = 0.5$).



(a) Source image. (b) Optical flow map. (c) Saliency map. (d) Result.

Figure 6.18: Distance plays an important role in visual attention, due to which objects nearer to a viewer are more attentive. The most salient location is marked by a green circle and a fixation by a red circle ($\lambda = 0.5$).

Chapter 7

Conclusion

The goal of this thesis is a novel spatiotemporal bottom-up saliency model.

We have focused on characteristics of human attention and some applications of saliency models in this master thesis. We have also described essential features of saliency models and focused on hierarchical saliency models.

We have analysed one of the most popular hierarchical models presented in (Itti et al., 1998) and discussed its advantages, disadvantages and possible improvements.

Consequently, we have proposed a novel hierarchical superpixel-based saliency model which is a combination of a standard hierarchical and a superpixel approach in Chapter 5. Our saliency model based on Feature Integration Theory processes 3 visual features – intensity, colour and orientation. Superpixel segmentation allows us to partially involve object-based attention in our model which absents in standard hierarchical methods.

We have implemented the model in C++ language using OpenCV and VlFeat library.

The novel superpixel-based model as well as our implementation of the model inspired by (Itti et al., 1998) were evaluated on Toronto dataset (Tsotsos – Bruce, 2006) with a set of natural scene images. The main benefit of superpixels in our model is respecting the shape of objects in the visual attention processing. Our model is nearly always successful in the prediction of the most viewed location in case of simple images. In case of natural scenes, our attention is influenced by top-down stimuli too. Since our model is bottom-up only, we cannot exactly measure its accuracy on this dataset. Despite of that, a significant improvement can be achieved using our proposed superpixel-based saliency model compared with the standard model (Itti et al., 1998). The improvement is observable as higher precision of saliency in those cases where the scene is well suited for a bottom-up saliency modelling without top-down factors at all. The main reason is the superpixel segmentation of an input which can correspond to object edges.

However, a saliency map obtained from the novel method models only static effects of our attention. In order to include also temporal aspects of human attention, we have designed and implemented our own algorithm for detecting salient regions in video sequences. The algorithm joins visual saliency of static as well as dynamic stimuli. Our spatiotemporal saliency model covers a dynamic impact of visual attention such as speed or changes in a scene.

The superpixel-based saliency model was extended by the fourth feature – *motion*. This dynamic feature is extracted from an optical flow map characterising the motion direction and the magnitude of each video frame. Combining static and dynamic attentional factors we have created a complex saliency model that may better predict human attention than static methods.

The spatiotemporal model was evaluated on a private video dataset captured at a shopping mall. It is supplemented by eye tracking data and optical flow maps. Saliency maps created using motion feature have higher saliency values at fixation locations than spatial maps.

Bibliography

ACHANTA, R. – SUSSTRUNK, S. Saliency detection for content-aware image resizing. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 1005–1008, 2009. doi: 10.1109/ICIP.2009.5413815. Available from: `http://infoscience.epfl.ch/record/135218/files/ICIP2009.pdf`.

ACHANTA, R. et al. SLIC Superpixels. Technical report, EPFL, 2010.

ACHANTA, R. et al. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Trans. Pattern Anal. Mach. Intell.* November 2012, 34, 11, pages 2274–2282. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.120. Available from: `http://dx.doi.org/10.1109/TPAMI.2012.120`.

ALPAYDIN, E. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010. ISBN 026201243X, 9780262012430.

BORJI, A. – ITTI, L. State-of-the-Art in Visual Attention Modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2013, 35, 1, pages 185–207. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2012.89. Available from: `http://ilab.usc.edu/borji/papers/06180177.pdf`.

BORJI, A. – AHMADABADI, M. N. – ARAABI, B. N. Cost-sensitive learning of top-down modulation for attentional control. *Machine Vision and Applications*. 2011, 22, 1, pages 61–76.

BORJI, A. et al. Analysis of scores, datasets, and models in visual saliency prediction. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 921–928. IEEE, 2013.

BRADSKI, G. – KAEHLER, A. *Learning OpenCV: Computer Vision with OpenCV Library*. O'Reilly Media, 1. ed. edition, 2008. ISBN 0-596-51613-4.

BRUCE, N. D. B. *Saliency, Attention and Visual Search: An Information Theoretic Approach*. PhD thesis, Canada, 2008. AAINR45988.

BRUCE, N. D. B. – TSOTSOS, J. K. Saliency Based on Information Maximization. In *NIPS*, 2005. Available from: `http://dblp.uni-trier.de/db/conf/nips/nips2005.html#BruceT05`.

BRUCE, N. – TSOTSOS, J. An Information Theoretic Model of Saliency and Visual Search. In PALETTA, L. – ROME, E. (Ed.) *Attention in Cognitive Systems. Theories and Systems from an Interdisciplinary Viewpoint*, volume 4840 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2007. pages 171–183. doi: 10.1007/978-3-540-77343-6_

11. Available from: http://dx.doi.org/10.1007/978-3-540-77343-6_11. ISBN 978-3-540-77342-9.

CHAMBOLLE, A. – POCK, T. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*. 2011, 40, 1, pages 120–145.

CICCARELLI, S. – WHITE, J. *Psychology*. MyPsychLab Series. Prentice Hall Higher Education, 2008. Available from: http://books.google.sk/books?id=BAVpHwAACAAJ. ISBN 9780136004288.

COMANICIU, D. – MEER, P. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* May 2002, 24, 5, pages 603–619. ISSN 0162-8828. doi: 10.1109/34.1000236. Available from: http://dx.doi.org/10.1109/34.1000236.

DOBEŠ, M. *Základy neuropsychológie*. Spoločenskovedný ústav SAV, 2005. Available from: http://books.google.sk/books?id=A20jGQAACAAJ. ISBN 9788096718245.

ENDRES, D. et al. Hooligan Detection: the Effects of Saliency and Expert Knowledge. *4th International Conference on Imaging for Crime Detection and Prevention, ICDP-11. IET.ISBN-978-1-84919-565-2*. 2011, pages 1–6. Available from: http://www.compsens.uni-tuebingen.de/joomla/index.php?option=com_jresearch&view=project&task=show&id=3&Itemid=67&lang=en.

FELDMAN, R. *Essentials of Understanding Psychology: Tenth Edition*. McGraw-Hill Higher Education, 2012. Available from: http://books.google.sk/books?id=nT40AAAAQBAJ. ISBN 9780077434465.

FELZENSZWALB, P. F. – HUTTENLOCHER, D. P. Efficient Graph-Based Image Segmentation. *Int. J. Comput. Vision*. September 2004, 59, 2, pages 167–181. ISSN 0920-5691. doi: 10.1023/B:VISI.0000022288.19776.77. Available from: http://dx.doi.org/10.1023/B:VISI.0000022288.19776.77.

FILIPE, S. – ALEXANDRE, L. From the human visual system to the computational models of visual attention: a survey. *Artificial Intelligence Review*. 2013. ISSN 0269-2821. doi: 10.1007/s10462-012-9385-4. Available from: http://dx.doi.org/10.1007/s10462-012-9385-4.

GAO, D. – VASCONCELOS, N. Decision-theoretic Saliency: Computational Principles, Biological Plausibility, and Implications for Neurophysiology and Psychophysics. *Neural Comput.* January 2009, 21, 1, pages 239–271. ISSN 0899-7667. doi: 10.1162/neco.2009.11-06-391. Available from: http://dx.doi.org/10.1162/neco.2009.11-06-391.

GAO, D. – VASCONCELOS, N. Discriminant Saliency for Visual Recognition from Cluttered Scenes. In *NIPS*, 2004. Available from: http://dblp.uni-trier.de/db/conf/nips/nips2004.html#GaoV04.

GAO, D. – MAHADEVAN, V. – VASCONCELOS, N. On the plausibility of the discriminant centersurround hypothesis for visual saliency. *Journal of Vision*. 2008, pages 1–18.

GIBSON, J. J. The perception of the visual world. 1950.

GOFERMAN, S. – ZELNIK-MANOR, L. – TAL, A. Context-Aware Saliency Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. Oct 2012, 34, 10, pages 1915–1926. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.272.

GOLDSTEIN, E. *The Blackwell Handbook of Sensation and Perception*. Blackwell Handbooks of Experimental Psychology. Wiley, 2008. Available from: `http://books.google.sk/books?id=Fs-5McBOqU4C`. ISBN 9780470751992.

GOLDSTEIN, E. *Sensation and Perception*. PSY 385 Perception Series. Wadsworth Cengage Learning, 2010. ISBN 9780495601494.

HAREL, J. – KOCH, C. – PERONA, P. Graph-Based Visual Saliency. In *NIPS*, pages 545–552. MIT Press, 2006. Available from: `http://dblp.uni-trier.de/db/conf/nips/nips2006.html#HarelKP06`. ISBN 0-262-19568-2.

HERING, E. *Grundzüge der Lehre vom Lichtsinn*. Handbuch der gesamten Augenheilkunde. J. Springer, 1920.

HOLMQVIST, K. et al. *Eye Tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011. Available from: `http://books.google.sk/books?id=5rIDPV1EoLUC`. ISBN 9780191625428.

HORN, B. K. – SCHUNCK, B. G. Determining optical flow. In *1981 Technical Symposium East*, pages 319–331. International Society for Optics and Photonics, 1981.

HOU, X. – ZHANG, L. Saliency Detection: A Spectral Residual Approach. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383267.

ITTI, L. Automatic Foveation for Video Compression Using a Neurobiological Model of Visual Attention. *Trans. Img. Proc.* October 2004, 13, 10, pages 1304–1318. ISSN 1057-7149. doi: 10.1109/TIP.2004.834657. Available from: `http://dx.doi.org/10.1109/TIP.2004.834657`.

ITTI, L. – KOCH, C. – NIEBUR, E. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. Nov 1998, 20, 11, pages 1254–1259. ISSN 0162-8828. doi: 10.1109/34.730558. Available from: `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=730558`.

ITTI, L. – BALDI, P. F. Bayesian surprise attracts human attention. In *Advances in neural information processing systems*, pages 547–554, 2005.

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

JACOBSON, N. – NGUYEN, T. Video processing with scale-aware saliency: Application to Frame Rate Up-Conversion. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 1313–1316, 2011. doi: 10.1109/ICASSP.2011.5946653. Available from: `http://videoprocessing.ucsd.edu/publications/Year_2011/13_Jacobson.pdf`.

JAMES, W. *The Principles of Psychology*. Number zv. 1 in American science series: Advanced course. H. Holt, 1918.

JODOGNE, S. – PIATER, J. H. Closed-loop Learning of Visual Control Policies. *J. Artif. Int. Res.* March 2007, 28, 1, pages 349–391. ISSN 1076-9757. Available from: `http://dl.acm.org/citation.cfm?id=1622591.1622601`.

JUDD, T. et al. Learning to predict where humans look. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2106–2113, Sept 2009. doi: 10.1109/ICCV.2009.5459462.

KIENZLE, W. et al. Center-surround patterns emerge as optimal predictors for human saccade targets. *Journal of Vision*. 2009, 9, 5, pages 7.

KING, L. *The Science of Psychology: An Appreciative View*. McGraw-Hill Education, 2010. Available from: `http://books.google.sk/books?id=CtvynlfJ06IC`. ISBN 9780073532066.

KOCH, C. – ULLMAN, S. Shifts in selective attention: Towards the underlying neural circuitry. *Human Neurobiology*. 1985, 4, pages 219–227.

LE CALLET, P. – NIEBUR, E. Visual Attention and Applications in Multimedia Technologies. *Proceedings of the IEEE*. 2013, 101, 9, pages 2058–2067. ISSN 0018-9219. doi: 10.1109/JPROC.2013.2265801. Available from: `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=06547645`.

LE MEUR, O. – LE CALLET, P. What we see is most likely to be what matters: Visual attention and applications. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 3085–3088, Nov 2009. doi: 10.1109/ICIP.2009.5414481.

LE MEUR, O. – LE CALLET, P. – BARBA, D. Selective H.264 video coding based on a saliency map. *Not Published*. Available from: `http://people.irisa.fr/Olivier.Le_Meur/publi/LeMeur_Coding_NotPublished.pdf`.

LE MEUR, O. – BACCINO, T. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods*. 2013, 45, 1, pages 251–266.

LEVINSHTEIN, A. et al. TurboPixels: Fast Superpixels Using Geometric Flows. *IEEE Trans. Pattern Anal. Mach. Intell.* December 2009, 31, 12, pages 2290–2297. ISSN 0162-8828. doi: 10.1109/TPAMI.2009.96. Available from: `http://dx.doi.org/10.1109/TPAMI.2009.96`.

LI, J. et al. Visual Saliency Based on Scale-Space Analysis in the Frequency Domain. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. April 2013, 35, 4, pages 996–1010. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.147.

LIM, J. – HAN, B. *Generalized Background Subtraction Using Superpixels with Label Integrated Motion Estimation*. volume 8693 of *Lecture Notes in Computer Science*. Springer International Publishing, 2014. Available from: `http://dx.doi.org/10.1007/978-3-319-10602-1_12`. ISBN 978-3-319-10601-4.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*. 2004, 60, 2, pages 91–110.

LOY, C. – XIANG, T. – GONG, S. Salient motion detection in crowded scenes. In *Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on*, pages 1–4, May 2012. doi: 10.1109/ISCCSP.2012.6217836.

LUCAS, B. D. – KANADE, T. et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

MA, Y.-F. et al. A generic framework of user attention model and its application in video summarization. *Multimedia, IEEE Transactions on*. 2005, 7, 5, pages 907–919.

MANCAS, M. *Computational Attention Towards Attentive Computers*. Presses univ. de Louvain, 2007. Available from: `http://tcts.fpms.ac.be/attention/Attention_Thesis_v2.5.pdf`. ISBN 9782874630996.

MARCHESOTTI, L. – CIFARELLI, C. – CSURKA, G. A framework for visual saliency detection with applications to image thumbnailing. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2232–2239, 2009. doi: 10.1109/ICCV.2009.5459467. Available from: `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5459467`.

MURTY, N. – DEVI, V. *Pattern Recognition: An Algorithmic Approach*. Undergraduate Topics in Computer Science. Springer, 2011. Available from: `http://books.google.sk/books?id=uBWD3HnzYFUC`. ISBN 9780857294951.

NEUBERT, P. – PROTZEL, P. Superpixel Benchmark and Comparison. In *Proc. of Forum Bildverarbeitung*, Regensburg, Germany, 2012. Available from: `https://www.tu-chemnitz.de/etit/proaut/forschung/superpixel.html`.

NEWSOME, W. T. – BRITTEN, K. H. – MOVSHON, J. A. Neuronal correlates of a perceptual decision. *Nature*. 1989.

OLIVA, A. et al. Top-down control of visual attention in object detection. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I–253–6 vol.1, Sept 2003. doi: 10.1109/ICIP.2003.1246946.

RICHE, N. et al. Saliency and Human Fixations: State-of-the-Art and Study of Comparison Metrics. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1153–1160, Dec 2013. doi: 10.1109/ICCV.2013.147.

RUDOY, D. et al. Learning video saliency from human gaze using candidate selection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1147–1154. IEEE, 2013.

SALAH, A. – ALPAYDIN, E. – AKARUN, L. A selective attention-based method for visual pattern recognition with application to handwritten digit recognition and face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. Mar 2002, 24, 3, pages 420–425. ISSN 0162-8828. doi: 10.1109/34.990146.

SCHEIER, C. – EGNER, S. Visual attention in a mobile robot. In *Industrial Electronics, 1997. ISIE '97., Proceedings of the IEEE International Symposium on*, volume 1, pages SS48–SS52 vol.1, 1997. doi: 10.1109/ISIE.1997.651734. Available from: `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=651734`.

SETLUR, V. et al. Automatic Image Retargeting. In *Proceedings of the 4th International Conference on Mobile and Ubiquitous Multimedia*, MUM '05, pages 59–68, New York, NY, USA, 2005. ACM. doi: 10.1145/1149488.1149499. Available from: http://doi.acm.org/10.1145/1149488.1149499. ISBN 0-473-10658-2.

SHI, J. – MALIK, J. Normalized Cuts and Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* August 2000, 22, 8, pages 888–905. ISSN 0162-8828. doi: 10.1109/34.868688. Available from: http://dx.doi.org/10.1109/34.868688.

SIAGIAN, C. – ITTI, L. Biologically Inspired Mobile Robot Vision Localization. *Robotics, IEEE Transactions on*. 2009, 25, 4, pages 861–873. ISSN 1552-3098. doi: 10.1109/TRO.2009.2022424. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5071253.

SONKA, M. – HLAVAC, V. – BOYLE, R. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007. ISBN 049508252X.

STENTIFORD, F. Attention Based Auto Image Cropping. *Workshop on Computational Attention and Applications, ICVS*. 2007. Available from: http://www.ee.ucl.ac.uk/~fstentif/WCAA2007.pdf.

TREISMAN, A. M. – GELADE, G. A Feature-Integration Theory of Attention. *Cognitive Psychology*. 1980, 12, pages 97–136.

TSOTSOS, J. K. – BRUCE, N. D. B. Saliency based on information maximization. In *Advances in Neural Information Processing Systems 18*, pages 155–162, MIT Press, 2006. MIT Press. Available from: http://www.cs.umanitoba.ca/~bruce/datacode.html.

VEDALDI, A. – SOATTO, S. Quick Shift and Kernel Methods for Mode Seeking. In *Computer Vision – ECCV 2008*, volume 5305 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008. pages 705–718. doi: 10.1007/978-3-540-88693-8_52. Available from: http://dx.doi.org/10.1007/978-3-540-88693-8_52. ISBN 978-3-540-88692-1.

VIJAYAKUMAR, S. et al. Overt visual attention for a humanoid robot. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 4, pages 2332–2337 vol.4, 2001. doi: 10.1109/IROS.2001.976418. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.63.2352&rep=rep1&type=pdf.

VINCENT, L. – SOILLE, P. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Trans. Pattern Anal. Mach. Intell.* June 1991, 13, 6, pages 583–598. ISSN 0162-8828. doi: 10.1109/34.87344. Available from: http://dx.doi.org/10.1109/34.87344.

VON HOLST, E. Relations between the central nervous system and the peripheral organs. *The British Journal of Animal Behaviour*. 1954, 2, 3, pages 89–94.

WOLFE, J. *Sensation and Perception*. Sinauer Associates, Incorporated, 2009. ISBN 9780878939534.

WU, Y. Optical flow and motion analysis. *ECE510-Computer Vision Course Notes*. 2001.

YARBUS, A. *Eye movements and vision*. New York : Plenum Press, 1967.

ZACH, C. – POCK, T. – BISCHOF, H. A duality based approach for realtime TV-L 1 optical flow. In *Pattern Recognition*. Springer, 2007. pages 214–223.

ZHAI, Y. – SHAH, M. Visual attention detection in video sequences using spatiotemporal cues. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 815–824. ACM, 2006.

ZHANG, J. – SCLAROFF, S. Saliency detection: A boolean map approach. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 153–160. IEEE, 2013.

ZHANG, L. et al. Sun: A Bayesian framework for saliency using natural statistics. *Journal of Vision*. 2008.

Appendix A

Technical Documentation

Implementation of our saliency model is done in C++ language using the OpenCV library and the Vlfeat library. Source codes of our solution are attached on DVD.

Here are some interesting parts of our novel superpixel-based saliency model:

- Listing A.1: difference between superpixel histograms,
- Listing A.2: difference between Gaussian pyramid layers,
- Listing A.3: superpixel Gaussian smoothing,
- Listing A.4: superpixel Gaussian pyramid,
- Listing A.5: spatial form of the superpixel-based saliency model,
- Listing A.6: intensity conspicuous map.

Listing A.1: Difference between superpixel histograms.

```
void SuperpixelFeatureProcessor::absdiffSuperpixel(const Mat &
   centerLabelsMap, const Mat &surroundLabelsMap, superpixel_map &
   centerSuperpixelsMap, superpixel_map &surroundSuperpixelsMap, Mat &
   output, const int compareMethod)
{
        output = Mat::zeros( centerLabelsMap.size(), CV_64F ); //
           difference between pyramid layers

        for( int i = 0; i < centerLabelsMap.rows; i++ )
        {
                for( int j = 0; j < centerLabelsMap.cols; j++ )
                {
                        Mat centerHistogram, surroundHistogram;

                        centerHistogram = centerSuperpixelsMap[
                           centerLabelsMap.at<vl_uint32>(i, j) ]; //
                           superpixel histogram at center scale
                        surroundHistogram = surroundSuperpixelsMap[
                           surroundLabelsMap.at<vl_uint32>(i, j) ]; //
                           superpixel histogram at surround scale

            // compare histograms
                        if( compareMethod == COMP_MEAN_COL )
                                output.at<double>(i, j) = abs(
                                   getMeanColor( centerHistogram ) -
                                   getMeanColor( surroundHistogram ) );
                        else if( compareMethod == COMP_MEAN_VECTORS )
                                output.at<double>(i, j) =
                                   getMotionDifference( centerHistogram,
                                   surroundHistogram );
                        else if( compareMethod == CV_COMP_CORREL ||
                           compareMethod == CV_COMP_INTERSECT )
                                output.at<double>(i, j) = 1 - abs(
                                   compareHist( centerHistogram,
                                   surroundHistogram, compareMethod ) );
                        else
                                output.at<double>(i, j) = compareHist(
                                   centerHistogram, surroundHistogram,
                                   compareMethod );
                }
        }
}
```

Listing A.2: Difference of Gaussians.

```
void SuperpixelIntensityProcessor::DoG(const vector<vector<Mat>> &
   labelsPyramids, vector<vector<superpixel_map>> &superpixelsPyramids,
   const set<int> &centerScale, const set<int> &surroundScale, vector<Mat
   > &differences)
{
        Mat pyr_c, pyr_s;

        const vector<Mat> &labelsPyramid = labelsPyramids[0];
        vector<superpixel_map> &superpixelsPyramid = superpixelsPyramids
           [0];

        for( int i : centerScale )
        {
                labelsPyramid[i].copyTo( pyr_c );

                for( int j : surroundScale )
                {
                        Mat diff;
                        pyr_s = labelsPyramid[i+j];
                        resize( pyr_c, pyr_c, pyr_s.size(), 0., 0.,
                           INTER_NEAREST );
                        absdiffSuperpixel( pyr_c, pyr_s,
                           superpixelsPyramid[ i ], superpixelsPyramid[ i
                           +j ], diff, CV_COMP_CORREL ); // difference
                           between SPXs
                        normalize( diff, diff, 0, 1, NORM_MINMAX );
                        differences.push_back( diff );
                }
        }
}
```

Listing A.3: Superpixel Gaussian smoothing.

```
void SuperpixelFeatureProcessor::computeGaussian(superpixel_map &
    superpixels, std::map<short, Neighborhood> &neighborhood, const Mat &
    labels, superpixel_map &prev_superpixels, int norm_type)
{
        const int kernel_size = 3;
        std::map<short, Neighborhood>::iterator it;

        MatND histogram[ kernel_size*kernel_size ];
        Mat gaussian;

        gaussian = getGaussianKernel( kernel_size, -1, CV_32F ) *
            getGaussianKernel( kernel_size, -1, CV_32F ).t();

        for( it = neighborhood.begin(); it != neighborhood.end(); it++ )
            // konvolve with Gaussian filter (3x3)
        {
                Mat finalHistogram, doubleHist;
                prev_superpixels[ it->first ].convertTo( doubleHist,
                    CV_64F );

                histogram[0] = getHistogramFromBlock( it->first, it->
                    second.neighbors_UL, prev_superpixels );
                histogram[1] = getHistogramFromBlock( it->first, it->
                    second.neighbors_U, prev_superpixels );
                histogram[2] = getHistogramFromBlock( it->first, it->
                    second.neighbors_UR, prev_superpixels );
                histogram[3] = getHistogramFromBlock( it->first, it->
                    second.neighbors_L, prev_superpixels );
                histogram[4] = doubleHist;
                histogram[5] = getHistogramFromBlock( it->first, it->
                    second.neighbors_R, prev_superpixels );
                histogram[6] = getHistogramFromBlock( it->first, it->
                    second.neighbors_BL, prev_superpixels );
                histogram[7] = getHistogramFromBlock( it->first, it->
                    second.neighbors_B, prev_superpixels );
                histogram[8] = getHistogramFromBlock( it->first, it->
                    second.neighbors_BR, prev_superpixels );

                superpixelConvolution( histogram, gaussian, kernel_size,
                    finalHistogram );

                finalHistogram.convertTo( finalHistogram, CV_32F );

                superpixels.insert( make_pair( it->first, finalHistogram
                    ) );
        }
}
```

Listing A.4: Superpixel Gaussian pyramid.

```
void SuperpixelFeatureProcessor::createPyramid(const int levels, Mat &
  labels, superpixel_map superpixels[], vector<Mat> &labelsPyramid,
  vector<superpixel_map> superpixelsPyramid[], int norm_type, const int
  n)
{
     neighborMaps.clear();
     labelsPyramid.push_back( labels );

     for( int j = 0; j < n; j++ )
          superpixelsPyramid[j].push_back( superpixels[j] );

     for( int i = 1; i <= levels; i++ ) // build the next layer ->
       downsample the label map and convolve it by Gaussian
     {
          Mat labelsLayer;
          neighbor_map neighborMap;

          resize( labelsPyramid[ i-1 ], labelsLayer, Size(
            labelsPyramid[ i-1 ].size().width / 2, labelsPyramid[
            i-1 ].size().height / 2  ), 0, 0, INTER_NEAREST );
          labelsPyramid.push_back( labelsLayer );

          // extract neighbourhoods of superpixels
          extractSuperpixelNeighborhoods( labelsLayer, neighborMap
            );
          neighborMaps.push_back( neighborMap );

          // apply Superpixel Gaussian smoothing for all dimensions
          for( int j = 0; j < n; j++ )
          {
               superpixel_map superpixelsLayer;
               computeGaussian( superpixelsLayer, neighborMap,
                 labelsLayer, superpixelsPyramid[j][ i-1 ],
                 norm_type ); // compute Gaussian of a new
                 layer
               superpixelsPyramid[j].push_back( superpixelsLayer
                 );
          }
     }
}
```

Listing A.5: Spatial superpixel-based saliency model.

```
Mat SuperpixelModel::createSaliencyMap(const Mat &input)
{
        vector<Mat> conspicuousMaps;
        vector<double> weights;
        SuperpixelFeatureProcessor *featureProcessor;

        vl_size regionSize[2] = { 15, 30 }; // superpixel size
        vl_size minRegionSize[2] = { 5, 15 }; // minimum superpixel size
        float regularization = 0.1f;
        int steps = 2;

        for( int i = 0; i < steps; i++ )
        {
                Mat mapI, mapC, mapO;

                // intensity conspicuous map
                featureProcessor = new SuperpixelIntensityProcessor(
                   pyrLevels, centerScale, surroundScale, regionSize[i],
                   regularization, minRegionSize[i], true );
                featureProcessor->process( input, mapI );
                conspicuousMaps.push_back( mapI );
                weights.push_back( 0.45 / steps );
                delete featureProcessor;

                // colour conspicuous map
                featureProcessor = new SuperpixelColorProcessor(
                   pyrLevels, centerScale, surroundScale,
                   zeroThresholdRatio, regionSize[i], regularization,
                   minRegionSize[i], false );
                featureProcessor->process( input, mapC );
                conspicuousMaps.push_back( mapC );
                weights.push_back( 0.3 / steps );
                delete featureProcessor;

                // orientation conspicuous map
                featureProcessor = new SuperpixelOrientationProcessor(
                   pyrLevels, centerScale, surroundScale, regionSize[i],
                   regularization, minRegionSize[i], false );
                featureProcessor->process( input, mapO );
                conspicuousMaps.push_back( mapO );
                weights.push_back( 0.25 / steps );
                delete featureProcessor;
        }

        // combination of maps into a saliency map
        return fuseConspiciousMaps( conspicuousMaps, weights );
}
```

Listing A.6: Intensity conspicuous map.

```
void SuperpixelIntensityProcessor::process(const Mat &input, Mat &map)
{
        Mat intensity, labels;
        vector<Mat> labelsPyr, differences;
        vector<vector<Mat>> labelsPyramids;
        vector<superpixel_map> superpixelsPyr;
        vector<vector<superpixel_map>> superpixelsPyramids;
        superpixel_map superpixels;

        createIntensity( input, intensity ); /// convert to grayscale

        computeIntensityLabels( input, labels, regionSize, regularization
           , minRegionSize, recomputeSuperpixels );

        representSegmentsByHistograms( &intensity, labels, &superpixels,
           NORM_MINMAX );

        createPyramid( levels, labels, &superpixels, labelsPyr, &
           superpixelsPyr, NORM_MINMAX, recomputeSuperpixels ); ///
           create intensity pyramid

        labelsPyramids.push_back( labelsPyr );
        superpixelsPyramids.push_back( superpixelsPyr );

        DoG( labelsPyramids, superpixelsPyramids, centerScale,
           surroundScale, differences ); /// create feature intensity
           maps (difference of gaussians)

        createMap( differences, input.size(), map ); /// create
           conspicuous intensity map
        resize( map, map, input.size() );
}
```

Appendix B

User Guide

The following minimum software requirements must be met to run our saliency model:

- Microsoft Visual Studio 2012,
- OpenCV 2.4.6[1],
- Vlfeat 0.9.18[2].

Our implementation offers three options:

1. `standard hierarchical model`: It runs the standard model based on (Itti et al., 1998) and creates saliency maps for an input image dataset.
2. `spatial superpixel-based model`: It runs our spatial superpixel-based model and creates saliency maps for an input image dataset.
3. `spatiotemporal superpixel-based model`: It runs our spatiotemporal superpixel-based model and creates saliency maps for an input video dataset.

All saliency models attached on DVD compute saliency maps and visualise human fixations, eventually the most viewed location with red circles and the most salient position with green circles. The C++ program displays stepwise all conspicuous maps used in the fusion into a single saliency map.

Standard Hierarchical Model

The first choice creates saliency maps for images using the standard hierarchical model (Itti et al., 1998). This option requires paths to the folder with images and the folder with fixation maps produced from eye tracking data as input. These folders must contain images named as `X.jpg` and gray-scaled fixation maps named as `dX.jpg`, where *X* is the ordinal number (starting from 1).

Spatial Superpixel-Based Model

The second option is our novel spatial superpixel-based model described in Chapter 5. It processes the same input as in the standard hierarchical saliency model.

[1] `http://opencv.org/`

[2] `http://www.vlfeat.org/`

Spatiotemporal Superpixel-Based Model

The last choice creates saliency maps using our spatiotemporal version of the superpixel-based algorithm. In contrast to the previous options, it analyses video sequences. Five parameters are required to run this model. The first parameter is the folder path with video frames. Each frame must be named as `rgb_XXXXXX.jpeg`, where *X* is the ordinal number (starting from 000000). The next input is the folder with images, where the second half of the image `XXXXXX.png` corresponds to a 2-channel optical flow map with the half size of the frame. The first channel characterises the flow orientation and the second one specifies the flow magnitude. The third parameter of this model is the file path to eye tracking data. Each line of this text files consists of the video frame ID and the fixation position. The fourth parameter enables or disables motion innovation in the model and the last input adjusts the motion rate in resulting saliency maps.

Appendix C

IIT.SRC 2015 paper

Our full paper was accepted for IIT.SRC 2015. It is the student research conference in Informatics and Information Technologies organised by the Faculty of Informatics and Information Technologies of the Slovak University of Technology in Bratislava.

Hierarchical Superpixel-Based Saliency Model

Patrik POLATSEK[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
patrik.polatsek@gmail.com

Abstract. Prediction of human visual attention is more and more frequently applicable in computer graphics, image processing, human-computer interaction and computer vision. Saliency models implement bottom-up factors of visual attention and represent the conspicuousness of a given environment using a saliency map. Standard hierarchical saliency methods do not respect the shape of objects and model the saliency as the pixel-by-pixel difference between the centre and its surround. The aim of our work is to improve the saliency prediction using a superpixel-based approach whose regions should correspond to object's borders. In this paper we propose a novel saliency method that combines a hierarchical processing of visual features and a superpixel-based segmentation. The proposed method is compared with existing saliency models and evaluated on a publicly available dataset.

1 Introduction

Our environment contains many objects which provide us a huge amount of visual information. The human brain has limited computational capacities, due to which it cannot process all incoming visual data. Thus *attention* provides mechanisms of reducing and selecting important information. Visual attention helps us to decide where to move our eyes and which parts of a scene should be deeper processed [2].

There are various factors that influence our attention. We can divide them into two main categories [5]:

1. stimulus-driven *bottom-up* factors such as colour, contrast, orientation, texture and movement,

2. and goal-driven *top-down* factors involving prior knowledge, experiences, expectations, tasks or goals.

Bottom-up attention is based on visual characteristics of a scene which automatically draw the attention. It is related to the term *saliency*. Saliency is the vividness of a stimulus which stands out relatively from its neighbour.

[*] Master study programme in field: Information Systems
Supervisor: Dr. Vanda Benešová, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

In recent decades scientists have studied mechanisms of human attention to determine regions of interest. Visual attention modelling has a wide range of applications such as computer graphics, image processing, human-computer interaction, psychology, neurophysiology and computer vision.

2 Related Work

According to the type of processing, we divide attention models on bottom-up, top-down and those that combine both processes. The majority of them models bottom-up attention. The result of such models is a *saliency map* which is a topographic representation of visual saliency of a scene [2].

Hierarchical (cognitive) models are biologically inspired models based on hierarchical decomposition of visual features inspired by *Feature Integration Theory* (FIT) [11]. According to the theory, in early and parallel *pre-attentive* processing a scene is analysed to identify individual features. Within the second, *focused* attention phase various features are combined and integrated to perceive whole objects.

One of the most known bottom-up hierarchical model is presented in [7]. It extracts three visual features – colour, intensity and orientation. The model uses the Opponent-Process Theory of Colour Vision based on two opponent-colour mechanisms: red-green and blue-yellow. The characteristics of texture and local orientation are obtained using the Gabor kernel. This model creates *Gaussian pyramids* for red, green, blue, yellow, intensity channel and local orientations. The structure of retinal ganglion receptive fields is characterised by *center-surround* organisation. This model achieves center-surround operations as the point-to-point difference between finer scales of Gaussian pyramid representing the center and coarser scales representing the surroundings that leads to multiple *feature maps*. Normalised feature maps are combined into three *conspicuous maps* for intensity, colour and orientation and finally into a single saliency map.

Bayesian models [13] are probabilistic frameworks which combine a bottom-up saliency with the effects of prior visual experience.

Decision-theoretic models [4] are based on the theory known as *discriminant saliency* selecting optimal attributes that most distinguish a visual class of interest from the other classes.

Information-theoretic models [3] are based on the theory which assumes that saliency results in the maximum information sampled from a given environment.

Graphical models use graph-based computations to create a saliency map. Nodes of a graph present a set of variables and edges their probabilistic dependencies. An eye movement sequence treated as a time-ordered sequence is modelled in [10] using a *Markov model*.

Spectral analysis models process images in the frequency domain instead of the spatial domain. The model mentioned in [6] is based on the *spectral residual* obtained as the difference between the original and smoothed version of the log spectrum.

Pattern classification [9] models use supervised machine learning algorithms to learn the visual attention from eye-tracking data or labelled salient regions.

Reinforcement learning models [8] predict the saliency using the reinforcement learning algorithm.

3 Proposed Algorithm

We introduce a novel method called *Hierarchical Superpixel-Based Saliency Model* for the detection of bottom-up saliency.

In order to at least partially cover the focused phase of attention that includes the integration of visual features to objects, we implement a superpixel-based saliency in our model instead of a simple pixel-by-pixel-based difference of Gaussian pyramid layers proposed in [7]. Our model is also a hierarchical saliency model based on FIT that integrates intensity, colour and orientation.

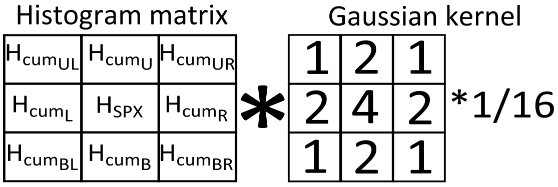

Histogram matrix
Gaussian kernel
$H_{cum_{UL}}$ H_{cum_U} $H_{cum_{UR}}$
H_{cum_L} H_{SPX} H_{cum_R}
$H_{cum_{BL}}$ H_{cum_B} $H_{cum_{BR}}$
*
1 2 1
2 4 2
1 2 1
*1/16

Figure 1. Superpixel Gaussian convolution.

A *superpixel* represents a visually coherent region which can better correspond to object contours than a rigid structure of pixels. The usage of superpixels is the primary difference between our model and the standard hierarchical model in [7].

The proposed solution segments input images using *Simple linear iterative clustering* (SLIC) [1] algorithm. SLIC is performed twice in the model with two different region sizes of 15 and 30.

3.1 Superpixel Gaussian Pyramid

Due to the usage of superpixels, the standard algorithm for Gaussian pyramid is replaced by our superpixel version. Each pyramid layer consists of a superpixel map representing the locations of all superpixels and a set of superpixel histograms.

Within the first pyramid layer 1D superpixel histograms are constructed using one of three visual features.

In order to create the next layers, we have to downsample the superpixel map to the half size.

Then we search neighbours to all superpixels in this resized superpixel map. The neighbour assignment procedure processes superpixel borders per pixel. Each border pixel is classified into one of the following categories based on its location to the analysed superpixel: left (L), right (R), upper (U), bottom (B), upper-left (UL), bottom-left (BL), upper-right (UR) and bottom-right (BR). Within each category, neighbours are characterised by a weight which depends on the boundary length with the superpixel. Longer the mutual boundary is, higher weight is assigned to the neighbour.

After the processing of the whole superpixel neighbourhood, we can build a histogram matrix of size 3 × 3. The center element corresponds to the analysed superpixel histogram H_{SPX}.

All 8 location categories are presented with a cumulated histogram H_{cum_i} defined as the weighted sum of all neighbour histograms connected to the category.

The rest of the histogram matrix is build using the 8 cumulated histograms of all location categories. Each cumulated histogram is assigned to the position in the matrix depending on the category name, for example the upper-left cumulated histogram takes place in the first (upper-left) position of the matrix. The histogram matrix is finally convolved with the discrete Gaussian kernel (Figure 1). In case of convolution at image borders where empty location categories without any neighbours may occur, the cumulated histogram of such categories equals to the histogram of analysed superpixel H_{SPX}.

In order to create a pyramid layer this procedure is repeated for all superpixels.

To produce the rest of Gaussian pyramid layers, the whole process is iteratively performed with the half size of an input superpixel map.

3.2 Superpixel Feature Processing

After the segmentation of an input image into superpixels using *SLIC* algorithm, our saliency model processes subsequently all features. For each feature it represents individual superpixels by a histogram. The superpixel map and the histogram set enter in the iterative process of the *superpixel Gaussian pyramid* as the first layer.

After the creation of the whole Gaussian pyramid, our model compares center and surround layers of the pyramid per pixel. In order to achieve the center-surround differences, we find

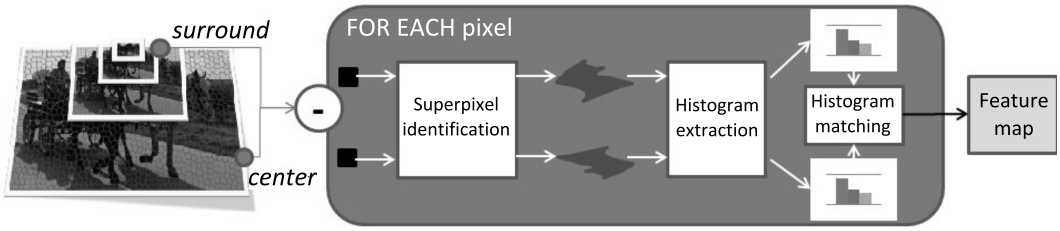


Figure 2. Difference between center and surround scales of superpixel Gaussian pyramid.

superpixels which contain compared pixels at the center and surround scale. Then we measure the similarity between the histograms belonging to these selected superpixels using a *histogram matching algorithm*. Such a difference between the layers produces a *feature map* (Figure 2).

3.2.1 Intensity

In order to analyse the intensity feature, an image is simply converted to *grayscale*. Superpixels are described by 1D histograms with 128 bins. A value of each pixel in the resulting feature map is computed using the *correlation* method as the follows:

$$FM_I(x,y) = 1 - abs(d_{correl}(H_c(x,y), H_s(x,y)), \tag{1}$$

where d_{correl} is a correlation coefficient, $H_i(x,y)$ is a histogram of a superpixel at the scale i which contains a pixel at position $[x,y]$ and subscripts c and s denote a center and a surround scales.

3.2.2 Colour

An input image is at first converted to 4-channel RGBY colour space for colour feature processing. Each colour channel is defined by its 1D histogram with 128 bins. The process implements the *Opponent-Process Theory*. The center-surround is expressed as the difference between *mean colour values* of compared superpixels for both opponent pairs:

$$FM_{RG}(x,y) = norm(abs(\max(H_{diff_R}) - \max(H_{diff_G}))), \tag{2}$$

$$FM_{BY}(x,y) = norm(abs(\max(H_{diff_B}) - \max(H_{diff_Y}))), \tag{3}$$

where $norm$ normalises values within the interval $\langle 0,1 \rangle$, max is the most frequently colour and $H_{diff_{COL}}$ is a difference of histograms of colour channel COL at a center c and a surround s scale defined as $H_{diff_{COL}}(x,y) = abs(H_{COL_c}(x,y) - H_{COL_s}(x,y))$.

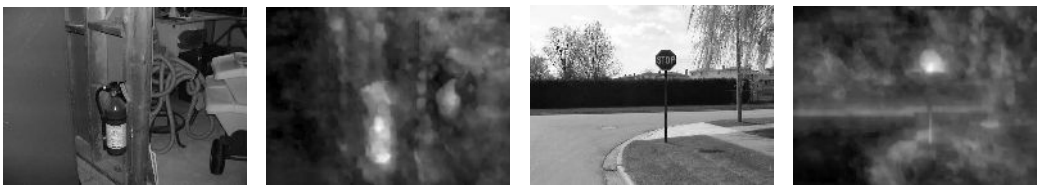
3.2.3 Orientation

The processing of orientation feature starts with the image conversion to *grayscale* colour space. Each superpixel is characterised with a *histogram of oriented gradients* with 9 bins. Orientation differences are computed with the same *correlation*-based method as in intensity feature maps:

$$FM_O(x,y) = 1 - abs(d_{correl}(H_c(x,y), H_s(x,y)). \tag{4}$$

3.3 Saliency Map

Extracted feature maps are combined into 3 *conspicuous maps* of intensity, colour and orientation for each used region size in SLIC algorithm. After their normalisation, they are linearly combined to create a single *saliency map* (Figure 3).



(a) Source image. (b) Saliency map. (c) Source image. (d) Saliency map.

Figure 3. Hierarchical superpixel-based saliency map.

4 Evaluation

In order to evaluate the novel method we have implemented our superpixel-based model as well as a standard hierarchical model inspired by [7] in C++ using OpenCV library. Both models were tested on a publicly available Toronto dataset [12] (120 images, 20 subjects).

In order to measure the accuracy we use our own method – *maxima distance*. It measures the distance between the most salient location on a saliency map and the location with the highest density on a fixation map. In order to compute the distance, we take iteratively into account together with the first maximum value of the fixation map also the next highest value as follows: $d_{MAX_i} = \min_i\left(d(\max(SM), \max_i(FM))\right)$, where SM is a saliency map, FM is a fixation map and $\max_i$ is the i^{th} maximum value. The results are visualised in Figure 4.

If the highest values of saliency and fixation maps are only considered, the total Euclidean distance produced by our model is 18250 px what is very similar to the results of a standard hierarchical model – the total distance of 18261 px.

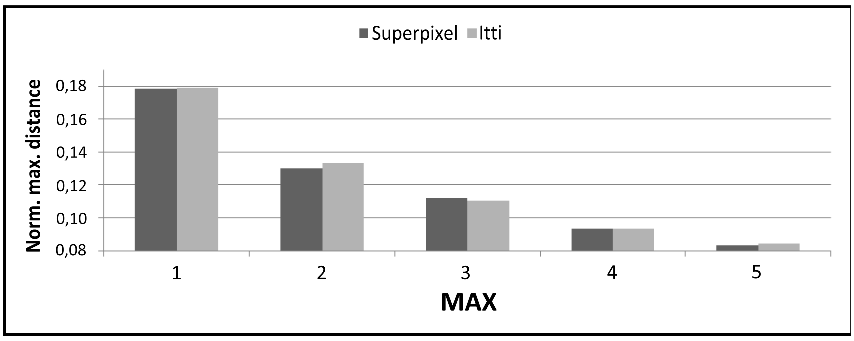


Figure 4. Normalised maxima distance.

To compare the models, we have also used standard methods such as the *similarity metric* defined as $S = \sum_x \min(SM(x), FM(x))$ and the *Kullback-Leibler (KL) divergence* computed as $KL_{div} = \sum_x FM(x) * \log\left(\frac{FM(x)}{SM(x)+\epsilon} + \epsilon\right)$, where ϵ is a small constant.

The results of all evaluation methods produced by our model are very close to the results of a standard hierarchical model as shown in Table 1.

Wrong classification of the most salient location is often the result of the absenting top-down attention in this hierarchical model. The top-down part of our saliency can affect the conspicuousness of objects more significantly in such complex scenes than in simple ones. It is focused in most cases on objects such as texts, human faces or traffic signs that may not be salient in terms of our model.

Despite of that, there are scenes where our model outperforms the standard one. The main reason is the superpixel segmentation of an input which can correspond to object edges.

Table 1. Comparison of a standard hierarchical model based on [7] and a novel model on Toronto [12].

Model	**Max.dist.**	**Similarity**		**KL-div.**	
	MAX_1	*Avg.*	*Med.*	*Avg.*	*Med.*
Standard	18261 px	0.3969	0.3992	1.1649	1.1484
Superpixel	18250 px	0.3939	0.3979	1.1870	1.1334

5 Conclusion and Future Work

In this paper we have presented a novel saliency model that integrates a hierarchical and a superpixel-based approach. The main benefit of superpixels in our model is respecting the shape of objects in the visual attention processing. Our model achieves on a publicly available dataset very similar results to a standard hierarchical model based on [7].

We will further focus on dynamic stimuli such as a motion contrast that significantly influence our attention. Combining static and dynamic attentional factors we create a complex spatiotemporal model that may better predict human attention.

References

[1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, vol. 34, no. 11, pp. 2274–2282.

[2] Borji, A., Itti, L.: State-of-the-Art in Visual Attention Modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, vol. 35, no. 1, pp. 185–207.

[3] Bruce, N.D.B.: *Saliency, Attention and Visual Search: An Information Theoretic Approach*. PhD thesis, Canada, 2008, AAINR45988.

[4] Gao, D., Vasconcelos, N.: Discriminant Saliency for Visual Recognition from Cluttered Scenes. In: *NIPS*, 2004.

[5] Goldstein, E.: *The Blackwell Handbook of Sensation and Perception*. Blackwell Handbooks of Experimental Psychology. Wiley, 2008.

[6] Hou, X., Zhang, L.: Saliency Detection: A Spectral Residual Approach. In: *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, pp. 1–8.

[7] Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1998, vol. 20, no. 11, pp. 1254–1259.

[8] Jodogne, S., Piater, J.H.: Closed-loop Learning of Visual Control Policies. *J. Artif. Int. Res.*, 2007, vol. 28, no. 1, pp. 349–391.

[9] Kienzle, W., Franz, M.O., Schölkopf, B., Wichmann, F.A.: Center-surround patterns emerge as optimal predictors for human saccade targets. *Journal of Vision*, 2009, vol. 9, no. 5, p. 7.

[10] Salah, A., Alpaydin, E., Akarun, L.: A selective attention-based method for visual pattern recognition with application to handwritten digit recognition and face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2002, vol. 24, no. 3, pp. 420–425.

[11] Treisman, A.M., Gelade, G.: A Feature-Integration Theory of Attention. *Cognitive Psychology*, 1980, vol. 12, pp. 97–136.

[12] Tsotsos, J.K., Bruce, N.D.B.: Saliency based on information maximization. In: *Advances in Neural Information Processing Systems 18*, MIT Press, MIT Press, 2006, pp. 155–162.

[13] Zhang, L., Tong, M.H., Marks, T.K., Shan, H., Cottrell, G.W.: Sun: A Bayesian framework for saliency using natural statistics. *Journal of Vision*, 2008.

Appendix D

SCCG 2015 paper

The Spring Conference on Computer Graphics (SCCG) is the international conference in the field of computer graphics, image processing and applications held in Smolenice castle. Our paper was accepted for SCCG 2015.

Bottom-up saliency model generation using superpixels

Patrik Polatsek[*]
Slovak University of Technology, Bratislava

Wanda Benesova[†]
Slovak University of Technology, Bratislava

Abstract

Prediction of human visual attention is more and more frequently applicable in computer graphics, image processing, human-computer interaction and computer vision. Human attention is influenced by various bottom-up stimuli such as colour, intensity and orientation as well as top-down stimuli related to our memory. Saliency models implement bottom-up factors of visual attention and represent the conspicuousness of a given environment using a saliency map. In general, visual attention processing consists of identification of individual features and their subsequent combination to perceive whole objects. Standard hierarchical saliency methods do not respect the shape of objects and model the saliency as the pixel-by-pixel difference between the centre and its surround. The aim of our work is to improve the saliency prediction using a superpixel-based approach whose regions should correspond to objects borders. In this paper we propose a novel saliency method that combines a hierarchical processing of visual features and a superpixel-based segmentation. The proposed method is compared with existing saliency models and evaluated on a publicly available dataset.

CR Categories: I.4.8 [IMAGE PROCESSING AND COMPUTER VISION]: Scene Analysis—Scene Analysis; I.2.10 [ARTIFICIAL INTELLIGENCE]: Vision and Scene Understanding—Perceptual reasoning;

Keywords: saliency, visual attention, saliency map model, superpixel

1 Introduction

The human brain, analogous to a computer, has limited computational capacities, due to which it cannot process all incoming visual data. Thus an attention provides mechanisms of reducing and selecting important information [Mancas 2007] [Borji and Itti 2013]. Visual attention modelling has a wide range of applications. In recent decades scientists have studied mechanisms of human attention to determine regions of interest from the huge amount of visual information. In general, there are two ways of selecting the regions which attract the attention [Mancas 2007]:

1. *Measuring the attention*: track eye movements, investigate brain activity and study human behaviour.
2. *Computing the attention*: create an algorithm that predicts salient regions on images or video sequences.

There are many factors for division of attention models [Borji and Itti 2013].

According to the type of processing, we divide the models on *bottom-up*, *top-down* and those that combine both processes. The majority of them models bottom-up attention. The result of such models is a *saliency map* which is a topographic representation of visual saliency of a scene.

Most attention models use only *spatial* visual information to create a saliency map. Dynamics and constant changes of a real-world environment indicate the requirement to model *spatiotemporal* effects of attention. In order to append temporal information to attention models and predict the attention from videos, we can use dynamic stimuli such as motion contrast or implement learning processes in attention.

Superpixels are regions in an image which can be used as basic units in the next image processing like segmentation, visual salience mapping or object detection [Benesova and Kottman 2014]. Superpixels typically cover the whole image, they are distributed regularly with respect to the nature of the input image, the desirable variation of the size of superpixels is preferably small and the boundary of superpixels has to be corresponding with the natural boundary of objects presented on the image. The goal of this work was to create a saliency model using superpixels as basic segmentation units. Our expectation is an increase in local accuracy. Superpixels follow naturally edges and hence they are more favorable compared to the commonly used regular regions as for example rectangle or circle.

Applications Detecting salient objects can be applied in robotics. An example of such application is presented in [Scheier and Egner 1997]. Using saliency map robots may obtain an active vision and shift their view on important parts of an environment [Vijayakumar et al. 2001]. Salient objects can be used in robots as localisation cues in order to orient and navigate themselves in the space in cases when GPS navigation may not be applicable [Siagian and Itti 2009].

Another field of usage represents surveillance systems. The SEARISE project called Smart Eyes is an active camera system that is able to track and zoom in on salient objects with active binocular cameras [Endres et al. 2011].

Saliency maps also have many different applications in computer vision and graphics in image and video processing. Marchesotti et al. [Marchesotti et al. 2009] use saliency detection for automatic image thumbnailing. Visual saliency is applied in contextaware image resizing in [Achanta and Susstrunk 2009]. Resizing image ratios may deform objects. Saliency map shows the prominent regions which ratios should be preserved. Detecting salient parts of an image is also used in image retargeting large size images to small size [Setlur et al. 2005]. The proposed method preserves important objects by eliminating gaps among them. Images and videos can be effectively compressed according to visual saliency. Salient regions are stored at higher resolution than unimportant parts [Itti 2004]. Jacobson and Nguyen present in [Jacobson and Nguyen 2011] a saliency application in frame rate-up conversion (FRUC).

Visual attention models can be useful in medical imaging [Le Callet and Niebur 2013]. Understanding the visual attention by reading medical images can automate pathology detection and localisation process.

2 Related Work

In this chapter we focus on the state-of-the-art in visual attention modelling. According to the attention type, models may be *feature-*

[*] e-mail:patrik.polatsek@gmail.com
[†] e-mail:benesova@fiit.stuba.sk

based, *space-based* or *object-based*.

2.1 Hierarchical Models

Hierarchical (cognitive) models are biologically inspired models based on hierarchical decomposition of visual features using Gaussian, Fourier or wavelet decomposition [Le Meur and Le Callet 2009].

These models are inspired by *Feature Integration Theory* (FIT) [Treisman and Gelade 1980] which presents visual information as a set of individual feature maps. In early and parallel pre-attentive processing a scene is analysed to identify individual features. Within the second, focused attention phase various features are combined and integrated to perceive whole objects. According to the theory, attention is responsible for the object perception instead of the perception of individual features.

FIT became a basis of the first theoretical biologically based attention model presented in [Koch and Ullman 1985]. First, elementary features such as colour, orientation and direction are extracted in parallel in order to create multidimensional topographical *feature maps* at different scales, called the *early representation*. Locations that differ mostly from their neighbourhoods are considered as the most conspicuous. Feature maps are finally combined and fused into a single *saliency map*. In order to determine the most salient location in a visual scene, a so-called *Winner-Take-All* (WTA) neural network is used. The properties of the winning location are transferred into the *central representation*.

One of the most known bottom-up saliency model based on the previous model is presented in [Itti et al. 1998] (Example: Figure 1). This model extracts the following visual features: *colour*, *intensity* and *orientation*.

The model creates *Gaussian pyramids* for *red* $R(\sigma)$, *green* $G(\sigma)$, *blue* $B(\sigma)$, *yellow* $Y(\sigma)$, *intensity* $I(\sigma)$ channel and *local orientations* $O(\sigma,\theta)$, where a pyramid level σ ranges from 0 to 8 and an orientation $\theta \in \{0,45,90,135\}$.

Colour channels of each pixel in the pyramid are defined by the following forms:
$R = r-(g+b)/2$,
$G = g-(r+b)/2$,
$B = b-(r+g)/2$,
$Y = (r+g)/2-|r-g|/2-b$.

where r,g,b are red, green and blue color coordinates of the pixel in the rgb color space. (Exact definition of the rgb color space depends on the image acquisition device.)

The structure of ganglion neurons is characterised by *center-surround*. The model achieves center-surround operations as the difference between finer scales of *Gaussian pyramid* representing the center and coarser scales representing the surroundings. Using center-surround operations denoted as Θ, visual features are computed. The model creates totally 42 different *feature maps*: 6 maps for intensity, 12 for colours and 24 for orientation.

In order to determine the intensity contrast, intensity maps are computed by the equation:

$$I(c,s) = I(c)\ \Theta\ I(s). \tag{1}$$

where c is center layer and s i surround layer.

According to the *Opponent-Process Theory of Colour Vision* [Hering 1920], human colour vision is a response of opponent colour channels. Colour stimuli are recombined and the colour perception is a result of two opponent-colour mechanisms: *red-green* (RG) and *blue-yellow* (BY). This algorithm models the colour opponency by the following colour maps:

$$RG(c,s) = |(R(c)-G(c))\ \Theta\ (G(s)-R(s))|, \tag{2}$$

$$BY(c,s) = |(B(c)-Y(c))\ \Theta\ (Y(s)-B(s))|. \tag{3}$$

In order to obtain the characteristics of texture and local orientation, the image is filtered using linear 2D *Gabor kernel*. The filter is created by multiplying Gaussian function with sinusoid at different orientations θ. The image convolved with this kernel is a base of a pyramid O which is used to obtain a last conspicuous map:

$$O(c,s,\theta) = |O(c,\theta)\ \Theta\ O(s,\theta)|. \tag{4}$$

Each computed map is normalised and multiplied by $(M-\overline{m})^2$, where M is a global maximum and $\overline{m}$ is an average of local maxima.

The next step consists of across-scale combination of all feature maps into 3 *conspicuous maps* for intensity $\overline{I}$, colour $\overline{C}$ and orientation $\overline{O}$ which are normalised again.

Finally, these conspicuous maps are fused into a single *saliency map* S:

$$S = \frac{1}{3}\left(N(\overline{I})+N(\overline{C})+N(\overline{O})\right), \tag{5}$$

where N represents the map normalisation.

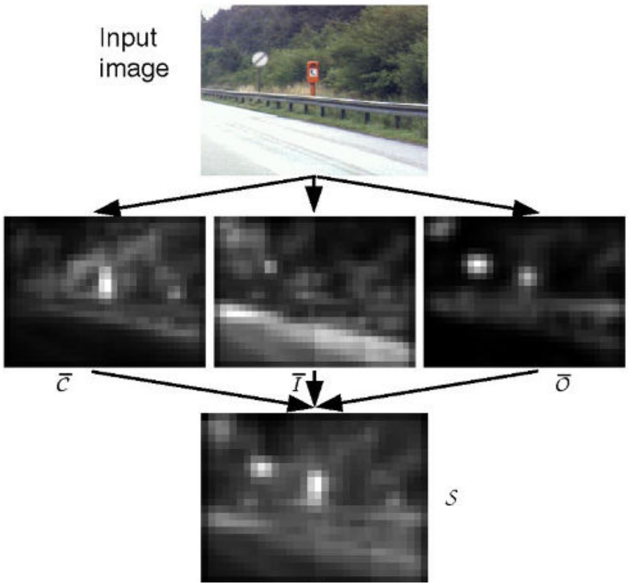


Figure 1: Input image is decomposed into several feature maps fused into 3 conspicuous maps for intensity, colour and orientation. The maps are finally combined into a single saliency map [Itti et al. 1998].

The most salient location is determined by the WTA network and shifts to the next salient locations are performed using the inhibition of return described in [Koch and Ullman 1985].

2.2 Bayesian Models

Bayesian models are probabilistic frameworks which combine a bottom-up saliency with the effects of *prior visual experience* [Le Meur and Le Callet 2009]. Impact of top-down attention could be modelled using *Bayes' rule*. Zhang et al. [Zhang et al. 2008] proposed a Bayesian framework called *SUN* (Saliency Using Natural statistics) which takes into account searching of target's features.

2.3 Decision Theoretic Models

Decision-theoretic models are based on the theory known as *discriminant saliency* that all saliency decisions as optimal in a decision-theoretic sense. Saliency is considered as the selection of optimal attributes that most distinguish a visual class of interest from the other classes.

The theory was first proposed for top-down saliency processing in [Gao and Vasconcelos 2004]. Gao et al. [Gao and Vasconcelos 2009] define two different classes of stimuli:

1. a *null hypothesis* composed of non-salient background stimuli,
2. *stimuli of interest* composed of visual features distinguishing the foreground from the null hypothesis.

Decision-theoretic models classify the locations of stimuli of interest as salient with the *lowest expected misclassification error probability*.

Gao et al. in [Gao et al. 2008] extend the model and combines the discriminant theory with center-surround differences of hierarchical bottom-up saliency for intensity, colour and orientation presented in [Itti et al. 1998].

2.4 Information Theoretic Models

Information-theoretic models are based on the theory which assumes that saliency results in the maximum information sampled from a given environment [Bruce 2008].

Bruce et al. in [Bruce and Tsotsos 2005] introduced the *AIM* (Attention based on Information Maximization) model measuring the information content of each image patch using a self-information defined as $-\log p(\mathbf{X})$, where $\mathbf{X}$ is a feature vector. The product of all probability density functions of a local image region leads to a joint likelihood that is easily converted to the self-information [Bruce 2008].

2.5 Graphical Models

Graphical models use graph-based computations to create a saliency map. Such models are probabilistic frameworks represented by a graph whose nodes present a set of variables and edges their probabilistic dependencies. An eye movement sequence treated as a time-ordered sequence is modelled using various methods such as *Markov Models*, *Conditional Random Fields* and *Dynamic Bayesian Networks* [Murty and Devi 2011; **?**]. Salah et al. [Salah et al. 2002] designed an attention graphical model for handwritten digit and face recognition.

2.6 Spectral Analysis Models

Spectral analysis models process images in the frequency domain for example using *Fast Fourier Transform* instead of the spatial domain.

A spectral analysis model based on the *spectral residual* was proposed in [Hou and Zhang 2007]. The model adopts the idea that the visual information is the summation of two parts:

$$H(image) = H(innovation) + H(prior\ knowledge), \qquad (6)$$

where $H(innovation)$ denotes the novelty part and
$H(prior\ knowledge)$ is the redundant part of the information.

In order to express the novelty part of the information, a down-sampled input image $I(x)$ (width of $64px$) is transformed into the spectrum by the Fourier Transform $\mathfrak{F}$ and its amplitude $\mathscr{A}(f)$ and phase $\mathscr{P}(f)$ are derived:

$$\mathscr{A}(f) = abs(\mathfrak{F}(x)), \mathscr{P}(f) = angle(\mathfrak{F}(x)). \qquad (7)$$

Consequently, the *log spectrum* representation $\mathscr{L}(f)$ is computed:

$$\mathscr{L}(f) = \log(\mathscr{A}(f)). \qquad (8)$$

The *spectral residual* $\mathscr{R}(f)$, containing the innovation of an image, is obtained as the difference between the original and smoothed version of the log spectrum:

$$\mathscr{R}(f) = \mathscr{L}(f) - h_n(f) * \mathscr{L}(f), \qquad (9)$$

where $h_n(f)$ is an $n \times n$ average filter.

The final saliency map is built in the spatial domain using the Inverse Fourier Transform $\mathfrak{F}^{-1}$ and smoothed with a Gaussian filter $g(x)$:

$$\mathscr{S}(x) = g(x) * \mathfrak{F}^{-1}[\exp(\mathscr{R}(f) + \mathscr{P}(f))]^2. \qquad (10)$$

Loy et al. in [Loy et al. 2012] use the similar spectral residual approach to detect salient motion.

2.7 Pattern Classification Models

Pattern classification models use *supervised machine learning* algorithms to learn the visual attention from eye-tracking data or labelled salient regions. These models may cover bottom-up and top-down attention too [Borji and Itti 2013].

Kienzle et al. [Kienzle et al. 2009] proposed a learning saliency model that is trained on recorded eye tracking data. In order to classify the saliency of image patches, the model uses the *Support Vector Machine* (SVM) algorithm.

A model in [Judd et al. 2009] also uses the SVM classifier for attention prediction. A training data set consists of feature vectors from fixation and random locations.

2.8 Reinforcement Learning Models

Reinforcement learning models predict the saliency using the *reinforcement learning* algorithm [Filipe and Alexandre 2013].

In the reinforcement learning (RL) inspired by behaviourist psychology, an agent takes actions in an environment that change its actual state and receives reward or penalty. The aim of the algorithm is to learn the best sequence of agent's actions that maximises the *cumulative reward* [Alpaydin 2010].

Jodogne et al. [Jodogne and Piater 2007] introduced a learning model based on the RL known as *Reinforcement Learning of Visual Classes* (RLVC). RLVC consists of two processes.

2.9 Model based on local low-level considerations, global considerations, visual organizational rules, and high-level factors

The authors [Goferman 2012] define a new type of saliency: context-aware saliency and propose an algorithm for the context-aware saliency detection. One interesting observation has indicated a gap between qualitative and quantitative evaluation. This gap has been caused by the ground-truth saliency maps which are extremely sparse and only the sparse points have been considered. Therefore the visual assessment was somewhat biased in the presented results.

3 Proposed Algorithm

We introduce a novel model of bottom-up saliency using superpixels. The model segments input images converted to grayscale into superpixels using SLIC algorithm [Achanta et al. 2012]. SLIC is performed twice in the model with two different sizes of superpixels: 15 and 30. Each superpixel is represented by a 1D histogram. Superpixel representation can partially include the integration of visual features to objects in the human visual attention processing. The model based on FIT integrates the following three features: 1. intensity, 2. colour, 3. orientation. The algorithm hierarchically processes all features using Gaussian pyramids with 6 layers. Center-surround organisation of human ganglion cells is modelled as a difference between finer and coarser levels of the pyramid. The center is represented at scales $c \in \{0,1,2\}$ and the surround scales are $s = c + \delta$, where $\delta \in \{0,1,2\}$.

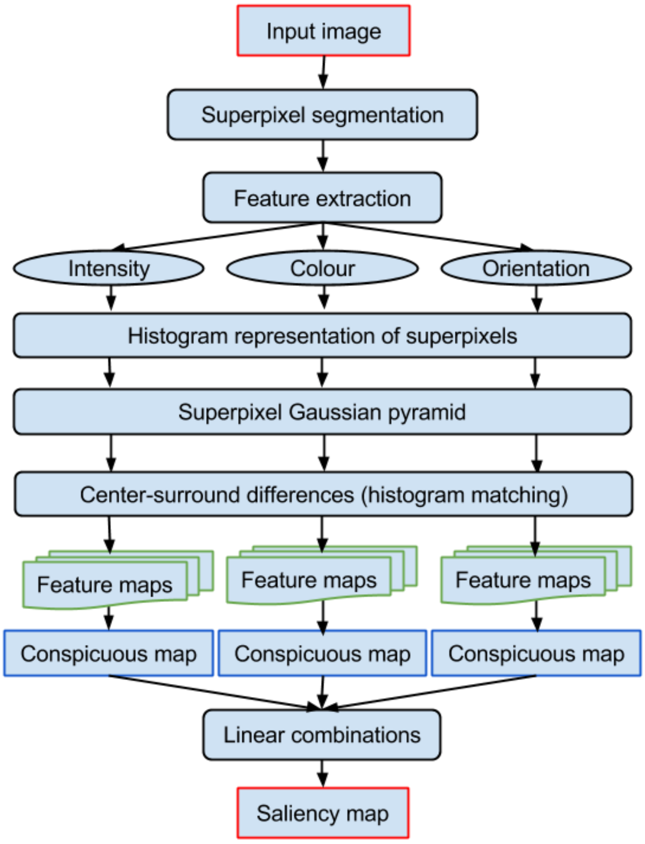


Figure 2: Scheme of hierarchical superpixel-based saliency model.

3.1 Superpixel Gaussian Pyramid

Due to the usage of superpixels, the standard algorithm for Gaussian pyramid is replaced by our superpixel version (Figure 2). Each pyramid layer consists of a superpixel map representing the locations of all superpixels and a set of superpixel histograms. Within the first pyramid layer 1D superpixel histograms are constructed using one of three visual features. In order to create the next layers, we have to downsample the superpixel map to the half of its size. Then we search neighbours to all superpixels in this resized superpixel map. The neighbour assignment procedure processes superpixel borders per pixel. Each border pixel is classified into one of the following categories based on its location to the analysed superpixel: left (L), right (R), upper (U), bottom (B), upper-left (UL), bottom-left (BL), upper-right (UR) and bottom-right (BR). Within each category, neighbours are characterised by a weight which depends on the boundary length with the superpixel. Longer the mutual boundary is, higher weight is assigned to the neighbour.

After the processing of the whole superpixel neighbourhood, we can build a histogram matrix of size 3x3. The center element corresponds to the analysed superpixel histogram HSPX. All 8 location categories are presented with a cumulated histogram defined as the weighted sum of all neighbour histograms connected to the category: $H_{cum_i} = \sum_j H_j * w_j$, where Hj is a neighbour histogram and wj is a weight of the neighbour. The rest of the histogram matrix is build using the 8 cumulated histograms of all location categories. Each cumulated histogram is assigned to the position in the matrix depending on the category name, for example the upper-left cumulated histogram takes place in the first (upper-left) position of the matrix. The histogram matrix is finally convolved with the discrete 3x3 Gaussian kernel (Figure 3). In case of convolution at image borders where empty location categories without any neighbours may occur, the cumulated histogram of such categories equals to the histogram of analysed superpixel HSPX. In order to create a pyramid layer this procedure is repeated for all superpixels. To produce the rest of Gaussian pyramid layers, the whole process is iteratively performed with the half size of an input superpixel map (Figure 4).

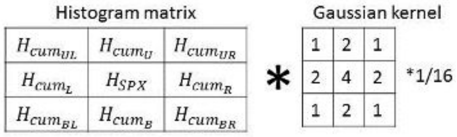


Figure 3: Superpixel Gaussian convolution

After the segmentation of an input image into superpixels using SLIC algorithm, our saliency model processes subsequently all features. For each feature it represents individual superpixels by a histogram. The superpixel map and the histogram set enter in the iterative process of the superpixel Gaussian pyramid as the first layer. After the creation of all levels of the Gaussian pyramid, our model compares center and surround layers of the pyramid per pixel. In order to achieve the center-surround differences, we find superpixels which contain compared pixels at the center and surround scale. Then we measure the similarity between the histograms belonging to these selected superpixels using a histogram matching algorithm. Such a difference between the layers produces a feature map FM (Figure 5).

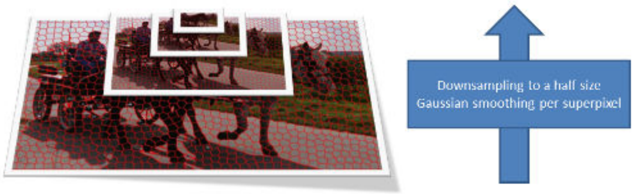

Downsampling to a half size
Gaussian smoothing per superpixel

Figure 4: Iterative process of superpixel Gaussian pyramid.

In general, the procedure to create a feature map consists of 4 steps:

1. superpixel segmentation,
2. representation of superpixels using histograms,
3. creating a superpixel Gaussian pyramid,
4. difference between center and surround pyramid layers.

3.2 Intensity

In order to analyse the intensity feature, an image is simply converted to grayscale. Superpixels are described by 1D histograms with 128 bins. Histogram comparison of Gaussian pyramid layers is based on the correlation method described by Equation 11. A value of each pixel in the resulting feature map is computed as the follows:

$$FM_I(x,y) = 1 - abs(d_{correl}(H_c(x,y), H_s(x,y)) \qquad (11)$$

where d_{correl} correlation coefficient ranges from -1 to 1, $H_i(x,y)$ is a histogram of a superpixel at the scale i which contains a pixel at position $[x,y]$ and subscripts c and s denote a center and a surround scales.

3.3 Colour

An input image is at first converted to 4-channel RGBY (Red Green Blue Yellow) colour space for colour feature processing. Each colour channel is defined by its 1D histogram with 128 bins. The process also implements the Opponent-Process Theory of Colour Vision, due to which it works with two opponent colour pairs - RG and BY . The center-surround is expressed as the difference between mean colour values of compared superpixels for both opponent pairs:

$$FM_{RG}(x,y) = norm(abs(max(H_{diffR}) - max(H_{diffG}))). \qquad (12)$$

$$FM_{BY}(x,y) = norm(abs(max(H_{diffB}) - max(H_{diffY}))). \qquad (13)$$

where *norm* normalises values within the interval$< 0,1 >$, max is the most frequently colour and $H_{diffCOL}$ is a difference of histograms of colour channel COL at a center c and a surround s scale defined as:

$$H_{diffCOL}(x,y) = abs(H_{COLc}(x,y) - H_{COLs}(x,y)). \qquad (14)$$

3.4 Superpixel Feature Processing

3.5 Orientation

The processing of orientation feature starts with the image conversion to grayscale colour space. Each superpixel is characterised with a histogram of oriented gradients with 9 bins. Single channel input image I is filtered by the Sobel kernel (size 3x3)in both directions. Image gradients are described by the gradient magnitude: $G = \sqrt{G_x^2 G_y^2}$ and the gradient direction : $\Theta = arctan(G_y/G_x)$ Gradient directions of each superpixel are split into 9 angle intervals. Each gradient pixel in the histogram of oriented gradient has a weight proportional to its gradient magnitude. Orientation differences are computed with the same correlation-based method as in intensity feature maps:

$$FM_O(x,y) = 1 - abs(d_{correl}(H_c(x,y), H_s(x,y)). \qquad (15)$$

3.6 Saliency Map

In the next phase all extracted feature maps are combined into 3 conspicuous maps CM of intensity, colour and orientation for each used region size in SLIC algorithm:

$$CM_i = \sum_j N(FM_{ij}). \qquad (16)$$

First, small values of all feature maps are set to zero. Such maps are modified using a normalisation operator N. The operator searches for local maxima in rectangular image regions. An input is then multiplied by $N = (M-m)^2$, where M is the global maximum and m is the average of all local maxima in image blocks. Finally, three conspicuous maps are linearly combined to create the resulting topography representation of image saliency called a saliency map SM.

$$SM = \sum_{reg}^{\{15;30\}} (0,45 * NI + 0,3 * NC + 0,25 * NO. \qquad (17)$$

where $NI = N(CM_{I_{reg}})$, $NC = N(CM_{C_{reg}})$, $NO = N(CM_{O_{reg}})$ and reg is a region size used in SLIC algorithm, N is a normalisation operator, I is an intensity, C is a colour and O is an orientation feature. The position with the highest pixel value in the saliency map is the location with the highest bottom-up saliency. Weighting factors resulted from experiments as the best choice. Figure 6 is an example of combining all conspicuous maps of intensity, colour and orientation into a single saliency map. The most salient location is marked with a green circle in result image.

WTA algorithm of our model is performed by the suppressing of the circular neighbourhood of the most salient location in the final saliency map and the searching for a new location with the maximum saliency.

4 Implementation

The algorithm of our saliency model has been implemented in C++ language using OpenCV library. Implementation of the Superpixel segmentation algorithm SLIC [Achanta et al. 2012] has been used from VlFeat library .

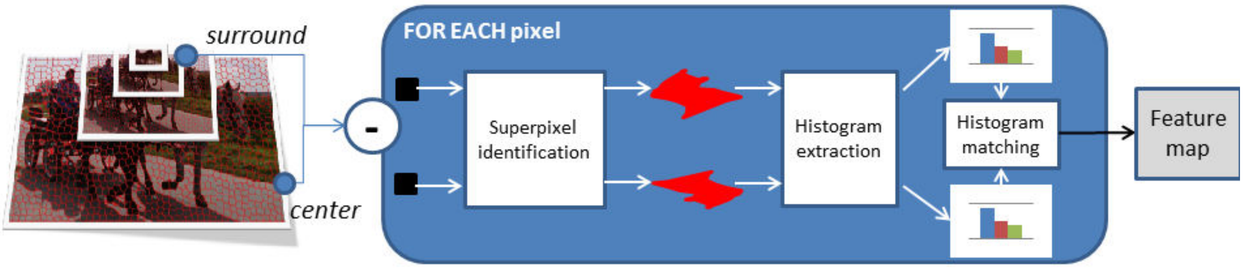


Figure 5: Difference between center and surround scales of superpixel Gaussian pyramid.

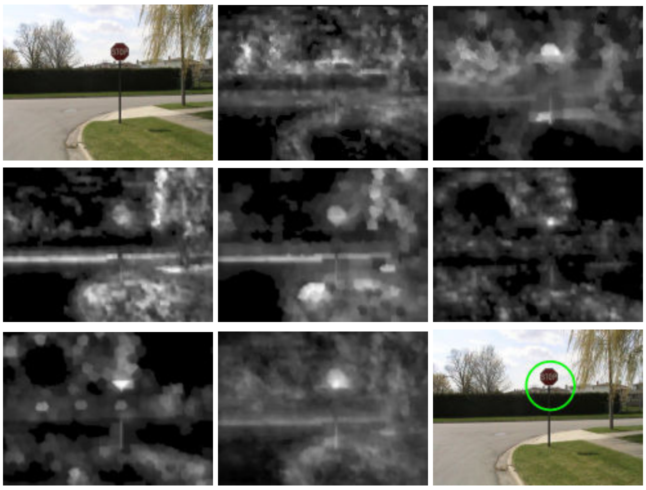


Figure 6: Combining conspicuous maps into a saliency map. Left to right, top to bottom: Analysed image, Intensity (region size 15), Intensity (region size 30), Colour (region size 15), Colour (region size 30), Orientation (region size 15), Orientation (region size 30), Saliency map, The most salient location.

5 Results

The superpixel-based saliency method has been evaluated on an image dataset which includes reference data acquired by the eye tracking and also compared with the Itti hierarchical model [Itti et al. 1998].

Saliency models were tested on a publicly available dataset[1] called *Toronto* [Tsotsos and Bruce 2006]. It contains 120 colour images of 681×511 size obtained from natural indoor and outdoor scenes. Images were freely observed by 20 subjects for 4 seconds. Each image is accompanied with eye tracking data as well as a density fixation map produced from the tracking data.

For visualisation purposes we label the most salient location in the resulting saliency map with a green circle and the location with highest value in a fixation map from eye tracker with a red circle as shown in Figure 7.

We have also evaluated both models using three methods. The *similarity metric* [Riche et al. 2013] represents the degree of similarity between the normalised distributions of a saliency map SM and a fixation map FM:

$$S = \sum_x \min(SM(x), FM(x)), \quad (18)$$

where $\sum_x SM(x) = \sum_x FM(x) = 1$. A similarity score ranges from 0 to 1. If the distributions do not overlap, the score equals to zero.

To compare the models, we have also used the *Kullback-Leibler (KL) divergence* [Riche et al. 2013; Le Meur and Baccino 2013]. The KL divergence is a nonnegative value that indicates the information lost when a fixation map FM distribution is approximated by a saliency map SM. The zero KL divergence occurs when the

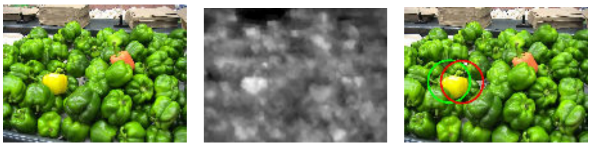


Figure 7: Example of a superpixel-based saliency map using superpixel model. From left to right: source image, saliency map, results. A green circle represents the predicted most salient location and a red circle denotes the most viewed location detected by the eye tracker.

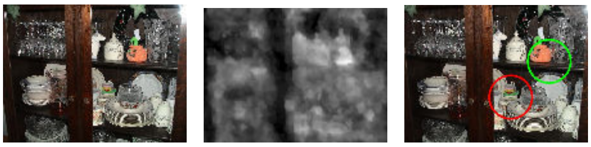


Figure 8: Example of a superpixel-based saliency map created from complex scenes with multiple objects. From left to right: source image, saliency map, results. A green circle represents the predicted most salient location and a red circle denotes the most viewed location detected by the eye tracker.

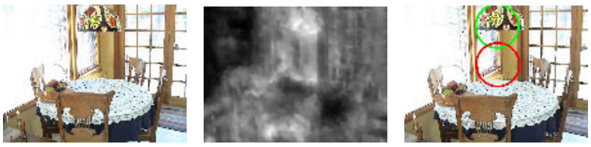


Figure 9: Example of a superpixel-based saliency map from scenes where the attention is focused on the centre of a given image. From left to right: source image, saliency map, results. A green circle represents the predicted most salient location and a red circle denotes the most viewed location detected by the eye tracker.

[1] http://www-sop.inria.fr/members/Neil.Bruce/

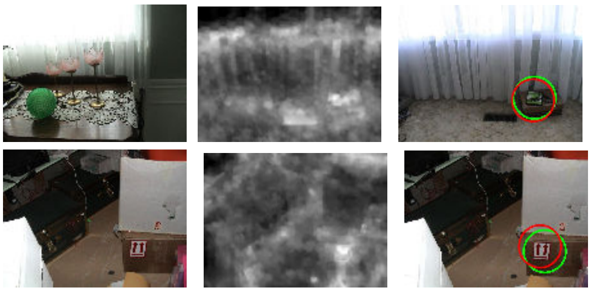


Figure 10: Example of superpixel-based saliency maps from scenes where the superpixel-based model outperforms the standard one. From left to right: source image, saliency map, results. The most salient location)calculated) and the most viewed locations (eye tracker) are marked by green and red circles.

probability distributions are equal. It is computed by the following form:

$$KL_{div} = \sum_{x} FM_{norm}(x) * \log\left(\frac{FM_{norm}(x)}{SM_{norm}(x)+\varepsilon} + \varepsilon\right), \qquad (19)$$

where $SM_{norm}(x) = \frac{SM(x)}{\sum_x SM(x)+\varepsilon}$, $FM_{norm}(x) = \frac{FM(x)}{\sum_x FM(x)+\varepsilon}$ and ε is a small constant to avoid logarithm and division by zero.

The results of the similarity metric as well as the KL divergence of the superpixel-based model are close to the results of the standard model (Figure 11 and 12). The median and the average of both models for Toronto dataset are shown in Table 1. The saliency models are evaluated using the similarity metric and the KL divergence.

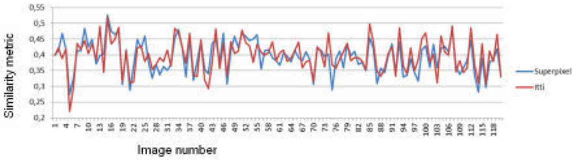


Figure 11: Similarity metric.

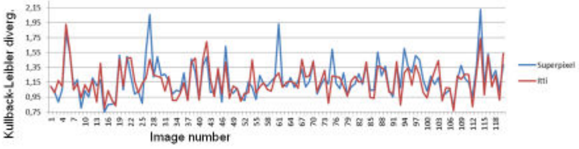


Figure 12: Kullback-Leibler divergence.

Table 1: Comparison of a *standard* hierarchical saliency model and a novel *superpixel*-based model.

Model	**Similarity**		**KL-div.**	
	Avg.	*Med.*	*Avg.*	*Med.*
Standard %	0.1969	0.1992	1.1649	1.1484
Superpixel %	0.1939	0.1979	1.1870	1.1334

The performance of saliency models are also compared by plotting a graph called the *receiver operating characteristic (ROC) curve*. The ROC curve [Le Meur and Baccino 2013] represents the tradeoff between the *true positive rate* and the *false positive rate*.

The true positive rate also called the *sensitivity* is computed as $TPR = \frac{TP}{TP+FN}$ and the false positive rate FPR also known as the *fall-out* is defined as $FPR = \frac{FP}{FP+TN} = 1 - SPC$, where TP is true positive, FP is false positive, TN is true negative, FN is false negative and SPC is specificity.

A saliency map is thresholded by a gradually increasing threshold, a fixation map by a constant threshold and the true positive and false positive rates are recorded. Using the rates the ROC curve is plotted that specifies the sensitivity as a function of fall-out.

The ROC curve represented by Figure 13 shows that the superpixel-based method is slightly more sensitive than the standard hierarchical saliency model.

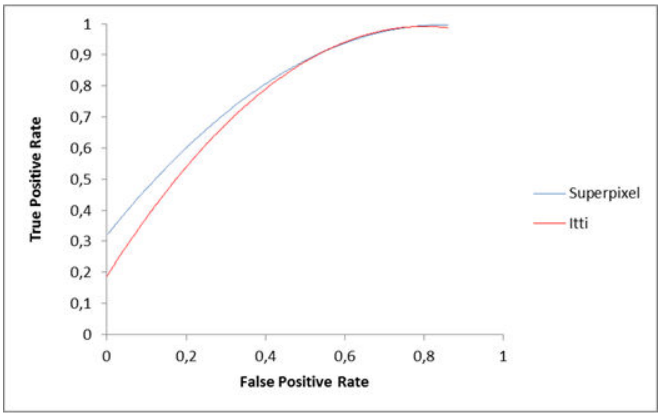


Figure 13: ROC curve.

6 Conclusion

The superpixel-based model is nearly always successful in case of simple images with a single dominant object. We have observed, that the accuracy of the proposed model decreases with the complexity of a given image as shown in Figure 8. Combining visual features of images with multiple objects into a saliency map may cause some mismatches. Wrong classification of the most salient location is often the result of the absenting top-down attention in this hierarchical model. The top-down part of our saliency can affect the conspicuousness of objects more significantly in such complex scenes than in simple ones.

The adjustment of a superpixel region size, a minimum region size and a regularisation coefficient used in SLIC is crucial in the model. If the selected parameters are insufficient, the resulting saliency map may not properly detect conspicuous objects with a tiny size. Despite of that, a significant improvement can be achieved using our proposed superpixel-based saliency model compared with the standard Itti model [Itti et al. 1998]. The improvement is observable as higher precision of the saliency in those cases where the scene is well suited for a bottom-up saliency modelling without top-down factors at all. The main reason is the superpixel segmentation of an input which can correspond to object edges.

In this paper we have presented a novel saliency model that integrates a hierarchical and a superpixelbased approach. The main benefit of superpixels in our model is respecting the shape of objects in the visual attention processing.

We will further focus on dynamic stimuli such as a motion contrast that significantly influence our attention. Combining static and dynamic attentional factors we create a complex spatiotemporal model that may better predict human attention.

7 Acknowledgments

This work was supported by the Grant VEGA 1/0625/14.

8 References

References

ACHANTA, R., AND SUSSTRUNK, S. 2009. Saliency detection for content-aware image resizing. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 1005–1008.

ACHANTA, R., SHAJI, A., SMITH, K., LUCCHI, A., FUA, P., AND SSSTRUNK, S. 2010. SLIC Superpixels. Tech. rep., EPFL.

ACHANTA, R., SHAJI, A., SMITH, K., LUCCHI, A., FUA, P., AND SUSSTRUNK, S. 2012. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell. 34*, 11 (Nov.), 2274–2282.

ALPAYDIN, E. 2010. *Introduction to Machine Learning*, 2nd ed. The MIT Press.

BENESOVA, W., AND KOTTMAN, M. 2014. Fast superpixel segmentation using morphological processing. *Proceedinks of the International Conference on Machine Vision and Machine Learning - MVML 2014.*

BORJI, A., AND ITTI, L. 2013. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 1, 185–207.

BRADSKI, G., AND KAEHLER, A. 2008. *Learning OpenCV: Computer Vision with OpenCV Library*, 1. ed. ed. O'Reilly Media.

BRUCE, N. D. B., AND TSOTSOS, J. K. 2005. Saliency based on information maximization. In *NIPS*.

BRUCE, N., AND TSOTSOS, J. 2007. An information theoretic model of saliency and visual search. In *Attention in Cognitive Systems. Theories and Systems from an Interdisciplinary Viewpoint*, L. Paletta and E. Rome, Eds., vol. 4840 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 171–183.

BRUCE, N. D. B. 2008. *Saliency, Attention and Visual Search: An Information Theoretic Approach*. PhD thesis, Canada. AAINR45988.

CICCARELLI, S., AND WHITE, J. 2008. *Psychology*. MyPsychLab Series. Prentice Hall Higher Education.

COMANICIU, D., AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 5 (May), 603–619.

ENDRES, D., H., N., M., K., AND GIESE, M. A. 2011. Hooligan detection: the effects of saliency and expert knowledge. *4th International Conference on Imaging for Crime Detection and Prevention*, 1–6.

FELDMAN, R. 2012. *Essentials of Understanding Psychology: Tenth Edition*. McGraw-Hill Higher Education.

FELZENSZWALB, P. F., AND HUTTENLOCHER, D. P. 2004. Efficient graph-based image segmentation. *Int. J. Comput. Vision 59*, 2 (Sept.), 167–181.

FILIPE, S., AND ALEXANDRE, L. A. 2013. From the human visual system to the computational models of visual attention: a survey. *Artificial Intelligence Review 39*, 1, 1–47.

GAO, D., AND VASCONCELOS, N. 2004. Discriminant saliency for visual recognition from cluttered scenes. In *NIPS*.

GAO, D., AND VASCONCELOS, N. 2009. Decision-theoretic saliency: Computational principles, biological plausibility, and implications for neurophysiology and psychophysics. *Neural Comput. 21*, 1 (Jan.), 239–271.

GAO, D., MAHADEVAN, V., AND VASCONCELOS, N. 2008. On the plausibility of the discriminant centersurround hypothesis for visual saliency. *Journal of Vision*, 1–18.

GOFERMAN, ZELNIK-MANOR, T. A. 2012. Context-aware saliency detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 34*, 10, 1915–1926.

GOLDSTEIN, E. 2010. *Sensation and Perception*. PSY 385 Perception Series. Wadsworth Cengage Learning.

HAREL, J., KOCH, C., AND PERONA, P. 2006. Graph-based visual saliency. In *NIPS*, MIT Press, 545–552.

HERING, E. 1920. *Grundzüge der Lehre vom Lichtsinn*. Handbuch der gesamten Augenheilkunde. J. Springer.

HOLMQVIST, K., NYSTRÖM, M., ANDERSSON, R., DEWHURST, R., JARODZKA, H., AND VAN DE WEIJER, J. 2011. *Eye Tracking: A comprehensive guide to methods and measures*. OUP Oxford.

HOU, X., AND ZHANG, L. 2007. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 1–8.

ITTI, L., KOCH, C., AND NIEBUR, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 20*, 11 (Nov), 1254–1259.

ITTI, L. 2004. Automatic foveation for video compression using a neurobiological model of visual attention. *Trans. Img. Proc. 13*, 10 (Oct.), 1304–1318.

JACOBSON, N., AND NGUYEN, T. 2011. Video processing with scale-aware saliency: Application to frame rate up-conversion. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 1313–1316.

JODOGNE, S., AND PIATER, J. H. 2007. Closed-loop learning of visual control policies. *J. Artif. Int. Res. 28*, 1 (Mar.), 349–391.

JUDD, T., EHINGER, K., DURAND, F., AND TORRALBA, A. 2009. Learning to predict where humans look. In *Computer Vision, 2009 IEEE 12th International Conference on*, 2106–2113.

KIENZLE, W., FRANZ, M. O., SCHÖLKOPF, B., AND WICHMANN, F. A. 2009. Center-surround patterns emerge as optimal predictors for human saccade targets. *Journal of Vision 9*, 5, 7.

KING, L. 2010. *The Science of Psychology: An Appreciative View*. McGraw-Hill Education.

KOCH, C., AND ULLMAN, S. 1985. Shifts in selective attention: Towards the underlying neural circuitry. *Human Neurobiology 4*, 219–227.

LE CALLET, P., AND NIEBUR, E. 2013. Visual attention and applications in multimedia technologies. *Proceedings of the IEEE 101*, 9, 2058–2067.

LE MEUR, O., AND BACCINO, T. 2013. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods 45*, 1, 251–266.

LE MEUR, O., AND LE CALLET, P. 2009. What we see is most likely to be what matters: Visual attention and applications. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 3085–3088.

LE MEURA, O., AND BARBAB, P. L. C. D. Selective h. 264 video coding based on a saliency map.

LEVINSHTEIN, A., STERE, A., KUTULAKOS, K. N., FLEET, D. J., DICKINSON, S. J., AND SIDDIQI, K. 2009. Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell. 31*, 12 (Dec.), 2290–2297.

LIM, J., AND HAN, B. 2014. *Generalized Background Subtraction Using Superpixels with Label Integrated Motion Estimation*, vol. 8693 of *Lecture Notes in Computer Science*. Springer International Publishing.

LOY, C., XIANG, T., AND GONG, S. 2012. Salient motion detection in crowded scenes. In *Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on*, 1–4.

MANCAS, M. 2007. *Computational Attention Towards Attentive Computers*. Presses univ. de Louvain.

MARCHESOTTI, L., CIFARELLI, C., AND CSURKA, G. 2009. A framework for visual saliency detection with applications to image thumbnailing. In *Computer Vision, 2009 IEEE 12th International Conference on*, 2232–2239.

MEUR, O. L., AND CALLET, P. L. What we see is most likely to be what matters: Visual attention and applications. In *ICIP*, IEEE, 3085–3088.

MURTY, N., AND DEVI, V. 2011. *Pattern Recognition: An Algorithmic Approach*. Undergraduate Topics in Computer Science. Springer.

NEUBERT, P., AND PROTZEL, P. 2012. Superpixel benchmark and comparison. In *Proc. of Forum Bildverarbeitung*.

OLIVA, A., TORRALBA, A., CASTELHANO, M., AND HENDERSON, J. 2003. Top-down control of visual attention in object detection. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 1, I–253–6 vol.1.

RICHE, N., DUVINAGE, M., MANCAS, M., GOSSELIN, B., AND DUTOIT, T. 2013. Saliency and human fixations: State-of-the-art and study of comparison metrics. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, 1153–1160.

SALAH, A., ALPAYDIN, E., AND AKARUN, L. 2002. A selective attention-based method for visual pattern recognition with application to handwritten digit recognition and face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 24*, 3 (Mar), 420–425.

SCHEIER, C., AND EGNER, S. 1997. Visual attention in a mobile robot. In *Industrial Electronics, 1997. ISIE '97., Proceedings of the IEEE International Symposium on*, vol. 1, SS48–SS52 vol.1.

SETLUR, V., TAKAGI, S., RASKAR, R., GLEICHER, M., AND GOOCH, B. 2005. Automatic image retargeting. In *Proceedings of the 4th International Conference on Mobile and Ubiquitous Multimedia*, ACM, New York, NY, USA, MUM '05, 59–68.

SHI, J., AND MALIK, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 8 (Aug.), 888–905.

SIAGIAN, C., AND ITTI, L. 2009. Biologically inspired mobile robot vision localization. *Robotics, IEEE Transactions on 25*, 4, 861–873.

SONKA, M., HLAVAC, V., AND BOYLE, R. 2007. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering.

STENTIFORD, F. 2007. Attention based auto image cropping. *Workshop on Computational Attention and Applications, ICVS*.

TREISMAN, A. M., AND GELADE, G. 1980. A feature-integration theory of attention. *Cognitive Psychology 12*, 97–136.

TSOTSOS, J. K., AND BRUCE, N. D. B. 2006. Saliency based on information maximization. In *Advances in Neural Information Processing Systems 18*, MIT Press, MIT Press, 155–162.

VEDALDI, A., AND SOATTO, S. 2008. Quick shift and kernel methods for mode seeking. In *Computer Vision ECCV 2008*, vol. 5305 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 705–718.

VIJAYAKUMAR, S., CONRADT, J., SHIBATA, T., AND SCHAAL, S. 2001. Overt visual attention for a humanoid robot. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 4, 2332–2337 vol.4.

VINCENT, L., AND SOILLE, P. 1991. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell. 13*, 6 (June), 583–598.

WOLFE, J. 2009. *Sensation and Perception*. Sinauer Associates, Incorporated.

YARBUS, A. 1967. *Eye movements and vision*. Plenum Press, New York.

ZHANG, L., TONG, M. H., MARKS, T. K., SHAN, H., AND COTTRELL, G. W. 2008. Sun: A bayesian framework for saliency using natural statistics. *Journal of Vision*.

Appendix E

Resumé

Resumé

1 Úvod

Naše okolie nám poskytuje obrovské množstvo vizuálnych informácií. L'udský mozog má ale podobne ako počítač obmedzené výpočtové kapacity, kvôli čomu nie je schopný spracovat' všetky prichádzajúce dáta. Preto *pozornost'* poskytuje mechanizmy na redukciu a selekciu dôležitých informácií.

1.1 Motivácia

Väčšina modelov vizuálnej pozornosti pracuje so statickými obrazmi. Vizuálna informácia, ktorú z prostredia získavame sa neustále mení. Preto je naším ciel'om rozšírit' priestorový model pozornosti o časovú informáciu. Výsledný *časovo-priestorový model* bude zahŕňat' statické ako aj dynamické vplyvy pozornosti. Pomocou informácie z predchádzajúcich obrazov môžeme lepšie modelovat' l'udskú pozornost' .

Modelovanie vizuálnej pozornosti má obrovské možnosti využitia:

- robotika,
- bezpečnostné systémy
- spracovanie obrazu a videa, napr. orezávanie obrazu, zmena vel'kosti obrazu, video kompresia,
- lekárske snímky,
- reklama a dizajn.

1.2 Požiadavky

Náš časovo-priestorový model pozornosti musí spĺňat' nasledovné podmienky:

- identifikovat' statické stimuly vizuálnej pozornosti na scéne,
- identifikovat' statické a dynamické vplyvy vizuálnej pozornosti na videosekvenciách,
- vytvorit' časovo-priestorový model významných čŕt,
- predpovedat' polohu významných oblastí a fixácie očí na obraze a videu.

2 Pozornost' a vnímanie scény

Pozornost' optimalizuje procesy v mozgu tak, že sa zameria na jediný aspekt zo scény, pričom ostatné bude ignorovat'. Hlavným ciel'om pozornosti všetkých živočíchov je upozornit' na hroziace nebezpečenstvo a pomôct' prežit'.

Hlavnými črtami pozornosti sú:

- *Selekcia*: Zameriavame sa na určité prvky prostredia na úkor ostatných.
- *Limitácia*: Úroveň spracovania senzorických informácii v mozgu je limitovaná.

2.1 Zraková sústava

Spracovanie optických dát začína vstupom svetla cez malú dieru v *dúhovke – zrenicu*. Svetlo sa koncentruje do jediného bodu na *sietnici*, ktorý sa nazýva *fovea*. Sietnica obsahuje receptory citlivé na svetlo – *tyčinky* a *čapíky*. Prichádzajúce svetlo sa premieňa do elektrických signálov v *gangliových neurónoch* a cez optický nerv sa dostane až do mozgu.

2.2 Vizuálna pozornosť

Existuje mnoho faktorov, ktoré ovplyvňujú našu vizuálnu pozornosť:

- stimulmi riadené *bottom-up* faktory,
- cieľmi riadené *top-down* faktory.

Vnímanie je proces priradenia významu prichádzajúcej informácii, ktorý nastáva po bottom-up a top-down spracovaní. Tieto dva mechanizmy nepracujú oddelene, ale navzájom spolupracujú.

2.2.1 Bottom-up pozornosť

Bottom-up pozornosť je veľmi rýchla, nevedomá a založená na vizuálnych črtách scény, ktoré automaticky pritiahnu pozornosť. Táto pozornosť súvisí s pojmom *význačnosť* (anglicky saliency). Je to nápadnosť takého stimulu, ktorý relatívne vystupuje zo svojho okolia. Typickými bottom-up faktormi sú farba, kontrast, orientácia, textúra a pohyb.

2.2.2 Top-down pozornosť

Top-down pozornosť je ovplyvňovaná našimi predchádzajúcimi vedomosťami, skúsenosťami, úlohami a cieľmi. V porovnaní s bottom-up pozornosťou, je tento typ pozornosti omnoho pomalší a vedomý. Top-down spracovanie organizuje jednotlivé vizuálne príznaky spracované bottom-up pozornosťou do logického celku, ktorý doplĺňa o chýbajúce informácie z našej pamäti.

2.3 Vnímanie pohybu

Stimuly, ktoré spôsobujú pohybové efekty môžeme rozdeliť do nasledovných kategórií:

1. **skutočný pohyb** pohybujúcich sa objektov,
2. **iluzórny pohyb** objektov, ktoré sa v skutočnosti nehýbu:
   (a) *zdanlivý pohyb* statických objektov zjavujúcich sa v mierne odlišných oblastiach a vytvárajú tak ilúziu pohybu,

(b) *indukovaný pohyb* statických objektov v okolí pohybujúceho sa objektu,

(c) *druhotný pohybový účinok* statických objektov po vzhliadnutí objektu v pohybe.

Podľa *Gibsonovej teórie* z roku 1950, informáciu o pohybe získavame z *optického poľa*. Každý pohyb na scéne potom spôsobí lokálne narušenie tohto poľa, tzv. *optický tok*. Optický tok môžeme definovať ako relatívny pohyb elementov k pozorovateľovi.

3 Analýza dostupných riešení

Algoritmy, ktoré modelujú vizuálnu pozornosť môžeme rozdeliť podľa mnohých kritérií.

Podľa typu spracovania poznáme *bottom-up*, *top-down* modely a také, ktoré kombinujú oba procesy spracovania. Väčšina z nich modeluje bottom-up pozornosť. Výsledkom týchto modelov je *mapa významných čŕt*, ktorá je topografickou reprezentáciou vizuálnej význačnosti scény.

Väčšina modelov pozornosti využíva len *priestorovú* vizuálnu informáciu. S použitím temporálnej informácie vzniká *časovo-priestorový* model pozornosti, ktorý predpovedá pozornosť z videozáznamov.

Podľa typu pozornosti ich zas delíme na modely založené na *príznakoch*, založené na *priestore* a tie, ktoré sú založené na *objektoch*.

Tabuľka 1 obsahuje najčastejšie kategórie modelov pozornosti spolu s ich základnou charakteristikou.

Tabuľka 1: Prehľad modelov pozornosti.

Model	**Popis**
hierarchické (kognitívne)	hierarchická dekompozícia príznakov
Bayesove	kombinácia význačnosti s predchádzajúcimi znalosťami
rozhodovaco-teoretické	diskriminačná teória význačnosti
informačno-teoretické	maximalizovanie informácie z daného prostredia
grafické	význačnosť založená na grafových algoritmoch
spektrálno-analytické	výpočet význačnosti vo frekvenčnej doméne
vzorovo klasifikačné	strojové učenie z význačných vzorov
s učením s odmenou a trestom	maximalizovanie získanej kumulovanej odmeny

3.1 Hierarchické modely

Hierarchické (kognitívne) modely sú biologicky inšpirované modely, ktoré využívajú hierarchickú dekompozíciu vizuálnych príznakov na základe *Teórie integrujúcej príznaky* (anglicky Feature Integration Theory). Podľa tejto teórie sa v prvej fáze spracovania analyzujú jednotlivé príznaky scény. V ďalšej fáze sú tieto príznaky kombinované tak, aby sme vnímali celé objekty.

Jedným z najznámejších hierarchických modelov je *Ittiho model*, ktorý extrahuje 3 príznaky — farbu, intenzitu a orientáciu. Tento model využíva *Oponentnú teóriu farebného videnia* založenej na dvoch oponentných farebných pároch: červená–zelená a modrá–žltá. Informácie o textúre a lokálnej orientácii sa získavajú pomocou Gaborovho filtra. Model potom vytvorí pre všetky príznaky *Gaussovu pyramídu*. Receptívne pole gangliových buniek má štruktúru typu *stred-okolie*. Algoritmus modeluje takúto štruktúru ako rozdiely medzi hrubšími a jemnejšími vrstvami Gaussovej pyramídy po pixeloch. Takéto rozdiely vytvárajú viaceré mapy príznakov, ktoré sú kombinované do máp nápadnosti pre každý príznak. Na záver spojenie týchto máp vedie k jedinej mape významných čŕt.

4 Analýza použitých princípov

Primárnym cieľom tejto diplomovej práce je vytvoriť bottom-up model významných čŕt.

Pre vytvorenie takéhoto modelu sme sa inšpirovali princípmi hierarchického Ittiho modelu významných čŕt.

Vo všeobecnosti sa spracovanie vizuálnej informácie skladá z 2 častí. Najskôr sa identifikujú jednotlivé príznaky a potom sú v druhej časti spracovania tieto príznaky spájané tak, aby sme vnímali jednotlivé objekty. Avšak táto fáza je opomínaná v Ittiho modeli.

Aby sme aspoň čiastočne pokryli aj túto pozornosť, v našom modeli implementujeme význačnosť založenú na superpixeloch namiesto jednoduchého rozdielu medzi vrstvami Gaussovej pyramídy po pixeloch. Superpixel reprezentuje vizuálne súvislú oblasť, ktorá môže korešpondovať kontúram objektov lepšie ako pevná štruktúra pixelov. Využitie superpixelov je hlavný rozdiel medzi naším a štandardným hierarchickým Ittiho modelom.

Aby sme mohli aplikovať mapy významných čŕt aj na videu, pridávame do nášho modelu aj spracovanie pohybu. S využitímv predpokladu súvislého pohybu v rámci superpixelov, implementujeme Gibsonovu teóriu optického toku rovnakým prístupom ako pri statických vizuálnych príznakoch.

5 Navrhovaný algoritmus

V tejto časti predstavíme nový *hierarchický model významných čŕt založený na superpixeloch* na detekciu bottom-up význačnosti. Model segmentuje vstupný obraz na superpixely s využitím *SLIC* (Simple linear iterative clustering) algoritmu. SLIC sa vykoná v našom programe dvakrát s dvoma odlišnými veľkosťami regiónov veľkosti 15 a 30.

Každý superpixel je reprezentovaný histogramom. Superpixelová segmentácia môže čiastočne pokryť integráciu vizuálnych príznakov do objektov vo vizuálnej pozornosti.

Náš model je založený na Teórii integrujúcej príznaky. Priestorová verzia modelu spracováva tieto príznaky:

1. *intenzita*,
2. *farba*,
3. *orientácia*.

Algoritmus hierarchicky spracováva všetky príznaky pomocou Gaussovej pyramídy so 6 vrstvami. Organizácia ľudských gangliových buniek typu stred-okolie je modelovaná cez rozdiely medzi jemnejšími a hrubšími vrstvami pyramídy. Stred je reprezentovaný škálami $c \in \{0, 1, 2\}$ a okolie ako $s = c + \delta$, kde $\delta \in \{1, 2, 3\}$.

Tento model rozširujeme aj temporálnym príznakom – *pohybom*, aby sme mohli hľadať význačné oblasti aj vo videosekvenciách. Náš časovo-priestorový model extrahuje informáciu o pohybe z máp optického toku reprezentujúce smer a veľkosť toku.

5.1 Superpixelová Gaussova pyramída

Kvôli použitiu superpixelov musel byť štandardný algoritmus na vytvorenie Gaussovej pyramídy nahradený našou superpixelovou verziou. Každá vrstva pyramídy sa skladá zo superpixelovej mapy reprezentujúcej polohu všetkých superpixelov a množiny superpixelových histogramov.

V rámci prvej pyramídovej vrstvy vytvoríme viaceré 1D superpixelové histogramy s použitím 3 vizuálnych príznakov.

Na zostrojenie ďalších vrstiev musíme zmenšiť superpixelovú mapu o polovicu.

Potom nájdeme susedov všetkým superpixelom v zmenšenej mape. Pri hľadaní susedov spracovávame hranice superpixelu po pixeloch. Každému hraničnému pixelu priradíme jednu z nasledovných kategórií na základe jeho polohy voči analyzovanému superpixelu: *vľavo*, *vpravo*, *hore*, *dole*, *vľavo hore*, *vľavo dole*, *vpravo hore* a *vľavo dole*. Susedom v každej z týchto kategórií priradíme váhu na základe dĺžky hranice so superpixelom. Čím dlhšia je ich vzájomná hranica, tým priradíme susedovi väčšiu váhu.

Po spracovaní celého okolia superpixela môžeme vytvoriť *maticu histogramov* veľkosti 3×3. Prostredný prvok matice bude zodpovedať histogramu analyzovaného superpixela. Všetkých 8 polohových kategórií reprezentujeme *kumulovaným histogramom*, ktorý vypočítame ako vážený priemer všetkých histogramov v danej kategórii.

Zvyšok matice sa zostrojí s použitím 8 kumulovaných histogramov. Každý histogram vložíme do matice na pozíciu podľa názvu jeho kategórie. Napríklad na prvú pozíciu v matici (vľavo hore) vložíme histogram kategórie vľavo hore. Nakoniec môže prebehnúť konvolúcia matice histogramov s diskrétnym Gaussovským filtrom veľkosti 3×3.

Tento postup musíme aplikovať na všetky superpixely v superpixelovej mape.

Na vytvorenie ďalších vrstiev opakujeme celý proces iteratívne so superpixelovou mapou zmenšenou o polovicu.

5.2 Superpixelové spracovanie príznakov

Po segmentácii obrazu s použitím SLIC algoritmu, spracuje náš model postupne všetky príznaky. Po reprezentovaní superpixelov ich histogramami vytvoríme iteratívne *superpixelovú Gaussovu pyramídu*.

Následne náš model porovnáva stredové a okrajové vrstvy pyramídy po pixeloch. Pre oba porovnávané pixely na centrálnej a okrajovej škále nájdeme superpixely, ktoré ich obsahujú. Potom zmeriame podobnosť medzi histogramami týchto superpixelov. Rozdielom medzi 2 vrstvami získame *mapu príznakov*.

Vo všeobecnosti môžeme proces vytvorenia mapy príznakov opísať 4 krokmi:

1. superpixelová segmentácia,
2. reprezentácia superpixelov histogramami,
3. vytvorenie superpixelovej Gaussovej pyramídy,
4. rozdiel medzi stredovými a okrajovými vrstvami pyramídy.

5.2.1 Intenzita

Na analýzu intenzity prevedieme obraz do šedotónového priestoru a Superpixely opíšeme 1D histogramami. Hodnotu pixela v mape príznakov vypočítame pomocou korelácie nasledovne:

$$FM_I(x,y) = 1 - abs(d_{correl}(H_c(x,y), H_s(x,y)), \tag{1}$$

kde d_{correl} je korelačný koeficient, $H_i(x,y)$ je histogram superpixelu na úrovni i, ktorý obsahuje pixel na pozícii $[x,y]$ a dolné indexy c a s označujú stredové a okrajové vrstvy.

5.2.2 Farba

Obraz je najskôr prevedený do 4-kanálového RGBY farebného priestoru a každý kanál reprezentujeme histogramom. Spracovanie farby je založené na Oponentnej teórii farebného videnia a porovnáva priemerné hodnoty farieb superpixelov pri oboch oponentných pároch:

$$FM_{RG}(x,y) = norm(abs(mean(H_{diff_R}(x,y)) - mean(H_{diff_G}(x,y)))), \tag{2}$$

$$FM_{BY}(x,y) = norm(abs(mean(H_{diff_B}(x,y)) - mean(H_{diff_Y}(x,y)))), \tag{3}$$

Kde $norm$ normalizuje hodnoty v intervale $\langle 0,1 \rangle$, $mean$ je priemerná farba a $H_{diff_{COL}}$ je rozdiel histogramov farebného kanálu COL na stredovej c a okrajovej s úrovni definovaný ako $H_{diff_{COL}}(x,y) = abs(H_{COL_c}(x,y) - H_{COL_s}(x,y))$.

5.2.3 Orientácia

Pre spracovanie orientácie prevedieme obraz do šedotónového priestoru. Každý superpixel charakterizujeme cez histogram orientovaných gradientov. Rozdiely v orientácii vypočítame rovnakou korelačnou metódou ako pri intenzite:

$$FM_O(x,y) = 1 - abs(d_{correl}(H_c(x,y), H_s(x,y)). \tag{4}$$

5.2.4 Pohyb

Na spracovanie pohybu používame 2-kanálové mapy optického toku charakterizujúce orientáciu a veľkosť toku. Pomocou týchto máp reprezentujeme superpixely *histogramom orientácií* a *histogramom veľkosti toku.*

Každý superpixel potom charakterizujeme jediným vektorom toku, ktorý vypočítame ako priemernú orientáciu a priemernú veľkosť z odpovedajúcich histogramov. Nech $\mathbf{v}_c(x,y)$ a

$\mathbf{v}_s(x, y)$ sú vektory toku superpixelov na stredovej a okrajovej úrovni pyramídy na pozícii (x, y), hodnota pohybovej mapy príznakov sa dá potom vyjadriť ako veľkosť rozdielu vektorov:

$$FM_M(x, y) = \|\mathbf{v}_s(x, y) - \mathbf{v}_c(x, y)\|. \tag{5}$$

5.3 Priestorová mapa významných čŕt

Získané mapy statických príznakov potom skombinujeme do 3 *máp nápadnosti* pre intenzitu, farbu a orientáciu pre každú použitú veľkosť regiónu v SLIC algoritme. Po ich normalizácii ich spojíme do jedinej *priestorovej mapy významných čŕt*.

5.4 Pohybová mapa inovácie

Pohybové mapy príznakov vyjadrujú pohybovú významnosť aktuálnej video snímky. Na určenie dynamických zmien v scéne vytvoríme *pohybovú mapu inovácie*, ktorá neberie do úvahy len aktuálnu mapu optického toku, ale aj niekoľko predchádzajúcich máp.

Temporálne zmeny sa detegujú pomocou *pohybovej pamäte*, ktorá sa aktualizuje s každou prichádzajúcou mapou:

$$MEM_{t+1} = (1 - \eta)MEM_t + \eta o_{t+1}, \tag{6}$$

kde MEM_t reprezentuje pohybovú pamäť v čase t, o_t je mapa optického toku a faktor učenia $\eta = 0.05$.

Pred aktualizáciou sa aktuálna mapa toku porovná s pamäťou po pixeloch. Každý pixel máp vyjadríme vektorom t pomocou orientácie a veľkosti toku na danej pozícii. Pohybová inovácia sa potom vypočíta ako:

$$IM_M(x, y) = \|\mathbf{v}_{MAP_t}(x, y) - \mathbf{v}_{m_t}(x, y)\|. \tag{7}$$

5.5 Časovo-priestorová mapa významných čŕt

Spojením pohybových máp príznakov získame *temporálnu mapu významných čŕt*. Kombináciou priestorovej a temporálnej mapy významných čŕt, poprípade doplnením o pohybovú mapu inovácie získame *časovo-priestorovú mapu významných čŕt*.

5.6 Implementačné detaily

Náš model významných čŕt bol implementovaný v jazyku C++ s použitím knižnice OpenCV. Na superpixelovú segmentáciu bola použitá implementácia z knižnice VlFeat.

6 Zhodnotenie a diskusia

Na vyhodnotenie nášho modelu sme použili verejne dostupnú testovaciu množinu obrazových dát Toronto a súkromnú videosekvenciu.

6.1 Priestorový model významných čŕt

Náš superpixelový priestorový model bol otestovaný na množine dát s názvom *Toronto*. Obsahuje 120 snímok z prirodzených scenérií, ktoré boli otestované na 20 subjektoch. Zo získaných dát o polohe očí bola pre každý snímok vytvorená fixačná mapa.

Na vyhodnotenie úspešnosti superpixelového modelu sme použili 3 metriky, ktoré sme porovnali s našou implementáciou Ittiho modelu.

Prvá metrika je naša vlastná metóda založená na porovnávaní vzdialenosti najvyšších hodnôt na vytvorenej mape významných čŕt a na fixačnej mape. Zároveň sme pri tejto metóde sledovali aj počet zhôd. Za zhodu sme považovali vzdialenosť medzi maximami menšiu než stanovený prah.

Ďalšia metrika spočíva v meraní *podobnosti* medzi distribúciami mapy významných čŕt a fixačnej mapy. Ak sa tieto distribúcie neprekrývajú, podobnosť je rovná nule.

Tretia metrika s názvom *Kullback-Leibler (KL) divergencia* meria informačnú stratu, ak sa na odhad fixačnej mapy použije mapa významných čŕt. V prípade, že distribúcie oboch máp sú zhodné, KL divergencia je nulová.

Výsledky všetkých 3 metrík sú uvedené v Tabuľke 2.

Tabuľka 2: Porovnanie štandardnej hierarchickej metódy založenej na Ittiho princípoch a novej superpixelovej metódy na testovacej množine dát Toronto.

Model	**Porovnávanie maxím**			**Podobnosť**		**KL-div.**	
	MAX_1	MAX_1, *prah 100 px*	MAX_2, *prah 120 px*	*Priem*	*Med*	*Priem*	*Med*
štandard	18261 px	44.17 %	59.17 %	0.3969	0.3992	1.1649	1.1484
superpixel	18250 px	42.50 %	64.17 %	0.3939	0.3979	1.1870	1.1334

6.2 Časovo-priestorový model významných čŕt

Časovo-priestorový model významných čŕt sme testovali na súkromnom videu od inštitútu v Rakúsku, *Joanneum Research*. Video pozostáva zo 410 snímok doplnených mapami optického toku.

Kvôli vyhodnocovaniu sme sledovali hodnoty mapy významných čŕt v oblastiach fixácie očí. Pridaním temporálnej mapy významných čŕt a pohybovej mapy inovácie do nášho superpixelového modelu došlo k nárastu tejto hodnoty z 0.548 na 0.576 pri faktore pohybu $\lambda = 0.40$.

7 Záver

V tejto diplomovej práci sme navrhli nový model významných čŕt, ktorý spája hierarchický a superpixelový prístup. Hlavnou výhodou superpixelov v našom modeli je rešpektovanie tvaru objektov pri spracovaní vizuálnej informácie. Navrhnutý model sme rovnako rozšírili o dynamickú informáciu, čím sme vytvorili časovo-priestorový model významných čŕt, ktorý môže predpovedať našu vizuálnu pozornosť aj vo videosekvenciách.

Appendix F

DVD Contents

/attention/	- source code of project solution
/relatedWork/	- contains most of the citated papers
/pdf/	- pdf version of this master thesis
/results/	- results of our saliency model