# CONTINUOUS REGION-BASED PROCESSING OF SPATIOTEMPORAL SALIENCY

Jan Tünnermann and Bärbel Mertsching
*GET Lab, University of Paderborn, Pohlweg 47-49, 33098 Paderborn, Germany*

Keywords: Spatiotemporal Saliency, Motion Saliency, Visual Attention.

Abstract: This paper describes a region-based attention approach on motion saliency, which is important for systems that perceive and interact with dynamic environments. Frames are collected to create volumes, which are sliced into stacks of spatiotemporal images. Color segmentation is applied to these images. The orientations of the resulting regions are used to calculate their prominence in a spatiotemporal context. Saliency is projected back into image space. Tests with different inputs produced results comparable with other state-of-the-art methods. We also demonstrate how top-down influence can affect the processing in order to attend objects that move in a particular direction. The model constitutes a framework for later integration of spatiotemporal and spatial saliency as independent streams, which respect different requirements in resolution and timing.



## 1 INTRODUCTION

The biologically inspired concept of visual attention is used in artificial vision to filter relevant from irrelevant information at early stages of processing. As in biological systems, only parts of the scene that are in the focus of attention (FOA) are forwarded to higher processing levels, such as object recognition or scene learning. Classic computational models create saliency maps from local contrasts in the input image regarding different feature dimensions, including color, intensity and orientation (see (Itti et al., 1998), for example). Additionally, besides this bottom-up process, top-down processes have been identified in biological attention which influence the FOA with respect to the current task. Recent technical models have incorporated such influences; (Wischnewski et al., 2010), for example, propose a method based on a psychological model. In mobile robotics and other systems that perceive — and possibly act in — dynamic environments it is not sufficient to process static saliency. Often robots observe a great deal of motion due to self motion and events in the environment. Local contrasts in direction or speed usually relate to something interesting, such as another moving entity that should be avoided (or approached, depending on the task). Salient motion is also a strong feature in biological attention (Mahapatra et al., 2008) and computational models are being updated to respect motion cues. Obtaining motion information from successive frames constitutes a coherence problem, the problem of describing the displacement of image parts from one frame to the next. This implies that raw image data must be grouped and described to match occurrences in subsequent frames. These are computationally expensive processes, which require solving high level problems such as object recognition and representation. Attention was considered to speed them up at an early stage, but extracting motion information in that manner requires these high level problems to be solved before the attentional stages. To resolve this chicken-egg situation, the coherence problem can be bypassed by creating spatiotemporal slices from the input. Successive $X-Y$ frames are collected for a certain duration of time and then converted into stacks of spatiotemporal $X-T$ and $Y-T$ slices. The incremental displacements in the $X-Y$ frames result in 'traces' or *motion signatures* in some of the slices, and their angles are related to the motions. Figure 1 illustrates this relation: The horizontal motion of the person produces a tilted region in the $X-T$ slice. If the person would be standing still, the trace would be parallel to the temporal axis. In the case where horizontal as well as vertical motion is present, tilted motion signatures would exist in $X-T$ and $Y-T$ slices. In short, the orientation of a motion signature describes the motion of the corresponding object, a fact that can be exploited to process motion saliency.

*Spatial* orientation is a classic feature of at-


Tünnermann J. and Mertsching B..
CONTINUOUS REGION-BASED PROCESSING OF SPATIOTEMPORAL SALIENCY.
DOI: 10.5220/0003823402300239
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2012), pages 230-239
ISBN: 978-989-8565-03-7

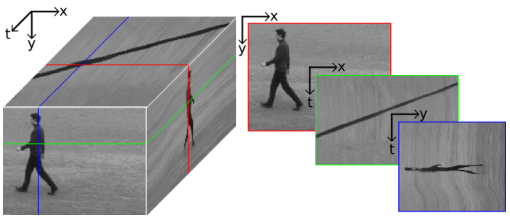


Figure 1: $X-Y-T$ volume visualization of a clip from (Schüldt et al., 2004). The images on the faces of the cuboid show slices from the inside. The same are shown separately without perspective distortion. The marker frames indicate where the slices are located in the volume.

tention models, so existing methods for calculating spatial orientation saliency may be used to calculate motion saliency on spatiotemporal slices. Pixel-based models emulate receptive fields and center-surround mechanisms (Itti et al., 1998); these have also been applied in the spatiotemporal domain (Belardinelli et al., 2008). An approach that constructs spatial saliency from singularities in frequency domain, *Spectral Residual* (Hou and Zhang, 2007) can be applied to spatiotemporal slices (Cui et al., 2009). A region-based approach on spatial attention is described in (Aziz and Mertsching, 2008a) where a color-segmentation is performed as an initial step. This model is capable of including top-down influences utilizing templates (Aziz and Mertsching, 2008b) which is more specific than tuning weights when combining different feature saliencies, a common method in classic models. First tests to expand this idea to the spatiotemporal domain have been conducted in (Tünnermann, 2010) where the same method was used for obtaining spatial and spatiotemporal orientations. Due to resulting constraints regarding the temporal range of the collected volume (it must be long), this system is incapable of processing continuous streams of visual data in (near) real-time, undoubtedly an important ability for the use in robotics. In this paper, we use a method for calculating region-based spatiotemporal saliency that is not subject to these restrictions but still preserves the ability of template-based top-down interfacing.

The following section discusses related work, while section 3 describes the proposed method. Evaluation of the results is shown in section 4. Section 5 discusses the proposed architecture with an eye on future work.

## 2 RELATED WORK

In the previous section we discussed different approaches that make use of spatiotemporal slices to detect motion saliency and we suggested a region-based method realizing this concept. However, other recent work approaches the problem in different ways. (Seo and Milanfar, 2009) and (Mahadevan and Vasconcelos, 2010), for example, apply center-surround windows and image statistical descriptions of the stimuli. (Mahadevan and Vasconcelos, 2010) apply their algorithm in a foreground-background-classification scenario and report robust results, even when a complicated dynamic background is present. In (K. Rapantzikos S. Kollias, 2009) a method is proposed that processes spatiotemporal volumes to make use of saliency information for video classification. The classic center-surround mechanism is applied in 3D to differently scaled sub-volumes. (Guo et al., 2008) extend the concept of the Spectral Residual. In contrast to (Cui et al., 2009), who applied it to spatiotemporal slices, they use a Quaternion Fourier Transformation that allows coding color, intensity and spatiotemporal change to obtain saliency by analyzing the phase spectrum. The mentioned models do not provide mechanisms to integrate top-down motion information. This is not only important when knowledge should influence the FOA, but also to retain the focus in consecutive frames. In other situations, refocusing must be suppressed, for example, when the object has been analyzed sufficiently (inhibition of return). These models also calculate spatiotemporal and spatial saliency in one common process. In contrast, separate pathways allow adjustments to different requirements. The processing of motion information must be fast but might not require resolutions as high as for spatial information. In human perception, magnocellular and parvocellular pathways transport visual information with large, fast low-resolution cells and slower high-resolution cells, respectively. The magnocellular pathway mainly contributes to motion and depth perception, while the parvocellular is involved with color and form (Livingstone and Hubel, 1987; Goodale and Milner, 1992). The information is combined at later stages to generate a coherent, conscious percept. Information from the pathways can be accessed before combination happens. We automatically dodge an object thrown at us even without seeing how it looks. This concept should be transferred to artificial systems. When something of possible interest quickly passes through a robot's vision and it is too fast to retrieve detailed information it may be useful to turn the camera in attempt to follow the object (or dodge, if it moves towards the robot). The method we suggest separates spatiotemporal from spatial information by processing spatiotemporal slices independently from the spatial frames. The region-based approach allows the use of top-down templates, which

in future work can be used to establish feedback-loops for saliency-based tracking and basic attentional controls, as inhibition of return. The initial color segmentation of each slice results in a list of regions for which basic features are calculated. By a voting mechanism regions collect saliency based on how different they are from their neighbors. Additional to this bottom-up path, top-down templates can be used to assign saliency with regard to similarity of each region to a template. All mentioned operations as well as the final combination of the different saliency channels can be done efficiently by looping through the region lists. Pixel-based saliency maps (usually only required for visualization) can be generated by information from the region lists and a label image which is created during the segmentation process and which maps pixel positions to the regions. In addition to spatiotemporal saliency processing with a good interface to top-down information that is separated from — and can be later integrated with — spatial attention, we see the need for a system that allows continuous processing of a visual input stream. Most of the other models work on closed volumes or finite numbers of frames and only some are stated to be able to process input in (or near) real time. To our knowledge there is no system for which it has been demonstrated how processing volumes are collected online from continuous input, say a camera, and calculate the FOA based on spatiotemporal saliency.

# 3 PROPOSED METHOD

The concept of region-based processing of spatiotemporal slices to obtain motion saliency can be illustrated as a flow of processing volumes (containing region lists for each slice). In figure 2 an overview of the processing done for each volume (initially a stack of input frames) is depicted. In the following explanations, numbers in round brackets refer to the circled numbers in the figure. Processing volumes are stacks of frames which can be collected from continuous input (see section 4.2). The volume is sliced into stacks of $X-T$ and $Y-T$ slices (1a). To enable a re-transformation to the spatial domain later, spatial color segmentation (see section 3.1) is performed for each frame (1b). For the spatiotemporal slices feature magnitudes are determined (2). This is done by performing color segmentation on each slice and then calculating the spatiotemporal angle (section 3.2). This information is then forwarded to the bottom-up module (3a) where spatiotemporal saliency is computed as described in section 3.3. With consideration of top-down influences, top-down saliency is processed, also based on the feature magnitudes (3b), as described in section 3.4. The resulting bottom-up and top-down feature maps are combined to form spatiotemporal master saliency maps (4). This combination is done by forming weighted averages of bottom-up and top-down saliency, where the weights should depend on the task. They can be used to completely switch off one of the pathways. This was done for the experiments reported in section 4 to evaluate bottom-up and top-down attention separately. The spatiotemporal master saliency maps from the $X-T$ and $Y-T$ pathways are now combined to produce a volume of spatiotemporal saliency which is then sliced back into common $X-Y$ orientation (5). This crucial step makes the spatiotemporal saliency accessible from the spatial context and is described in section 3.5. The results of this step are pixel-based intensity maps. The coherence problem which was bypassed by working on spatiotemporal slices must now be solved in a different form: Spatiotemporal saliency, projected back into the spatial domain, must now be assigned to the entities that were responsible for the motion. This is done by adding the values from the pixel-based intensity maps to corresponding regions from the spatial segmentation (1b) and normalizing them by the region size (section 3.5). Once this is done, the FOA is determined by selecting the region with the most projected spatiotemporal saliency for each $X-Y$ frame.

## 3.1 Segmentation of Spatial Frames and Spatiotemporal Slices

Initially, spatial as well as spatiotemporal images must be segmented. Please note that the used color segmentation cannot be described in detail in the scope of this paper. Conceptually, any method can be used that turns pixel images into coherent regions. The algorithm we used in our implementation is basically the same as described by (Aziz, 2009). Seed pixels are selected and regions are grown iteratively. Six thresholds decide whether a pixel is added to a region or not. Thresholds $\Gamma^h$, $\Gamma^i$ and $\Gamma^s$ denote how much the pixels may vary in *hue*, *intensity* and *saturation* regarding the region's seed pixel. Thresholds $\tau^h$, $\tau^i$, $\tau^s$ denote the tolerance of variance between the neighboring pixels (all in range $[0..255]$). In contrast to (Aziz, 2009) we use the same thresholds for the complete hue range. For gray-scale input we consider intensity only. The minimum region size $\mu$ (in pixels) is used to filter out all regions smaller in size. It is crucial because on short spatiotemporal slices motion signatures will be small but must be kept while on spatial slices small regions can appear due to noise

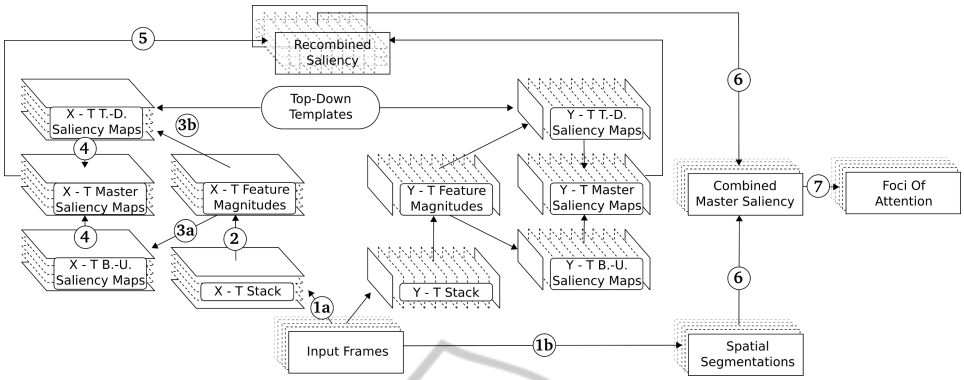


Figure 2: Architecture overview of the proposed method. This figure depicts the flow of information for one processing volume. The circled numbers relate to the order of steps performed and refer to explanations in the text. Please note that only the $X-T$ pathway and the shared path is numbered and explained, whereas the $Y-T$ processing is done in an analog manner.

and may be dropped. The existence of these parameters does not imply that the model needs a great deal of supervision to adjust them for different images. Quite the opposite is the case; a set of parameters is sufficient for various kinds of input scenes. All clips in our evaluation were processed with the same parameter set (see section 4 for concrete parametrization).

## 3.2 Feature Magnitudes of Spatiotemporal Orientation

The proposed model is based on the fact that the angle of motion signatures on spatiotemporal slices (*spatiotemporal orientation*) is related to the motion of the object that produced the signature. Assuming no other transformations, such as scaling or rotating, a motion signature is a spatiotemporal area defined by a parallelogram. Non-moving background objects, assuming that they are not disturbed by moving objects, produce rectangular signatures. The sides that are in parallel to the time axis indicate that the object did not move during the time represented by the slice. When an object starts to move, the signature shears and the sides that were parallel to the time axis have a specific angle. In general, the orientation of edges produced by a moving object in spatiotemporal slices is related to the object's velocity $\mathbf{u}$ with

$$\mathbf{u} = -\begin{bmatrix} \tan\varphi_x \\ \tan\varphi_y \end{bmatrix} \tag{1}$$

where $\varphi_x$ is the angle between the temporal axis and the edge on the $X-T$ slice and $\varphi_y$ the corresponding edge's angle on the $Y-T$ slice.

A common method to obtain a region's orientation is to use second order central moments. However, this results in special requirements for the processing volume dimensions. A rather short volume can cause false spatiotemporal orientations. This can be seen considering region two in figure 3 (b). As the angle produced by that approach is relative to the major axis of the region, it would describe a region parallel to the spatial axis. The region was produced by a non-moving background object so its spatiotemporal orientation should be 90° to the spatial axis. The major axis would in fact be 90° to the spatial axis, if more frames would have been recorded (elongating the slice temporally) or if the input image resolution would have been smaller (compressing the slice spatially). Both options have drawbacks. Decreasing resolution means loosing detail and increasing the frame number leads to requiring more time to collect the frames, delaying the result.

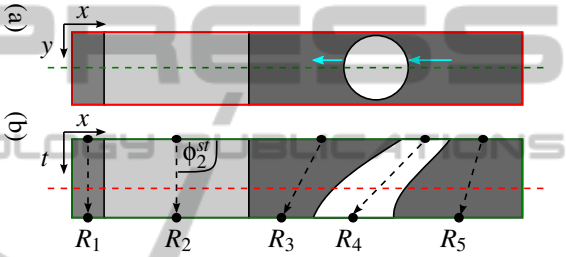


Figure 3: (a) A white disk moving in front of gray rectangles. The horizontal dashed line marks the $Y$ level of the $X-T$ slice shown in (b). The $X-T$ slice shows spatiotemporal signatures of background and disk. Dashed vectors illustrate how spatiotemporal orientation $\phi_i^{st}$ is determined (the angle is marked explicitly for region $R_2$, only).

As we are interested in the "tiltiness" of a region rather than the orientation of its major axis, we use a simple method to obtain spatiotemporal orientation in this sense. For each region $R_i$ on a spatiotemporal slice, we determine the first row $L_i^{first}$ and the last row $L_i^{last}$ of pixels that belong to the region with

$$L_i^{first} = \{p_{xt} \mid t \leq a \wedge p_{xt}, p_{xa} \in R_i\} \quad (2)$$

$$L_i^{last} = \{p_{xt} \mid t \geq a \wedge p_{xt}, p_{xa} \in R_i\} \quad (3)$$

The region membership of pixels $p_{xt}$ is obtained from the label image, which was produced in the segmentation step. The centers of these rows (corresponding to the black dots in figure 3) are determined as

$$c_i^{first} = \frac{1}{|L_i^{first}|} \sum_{p_{xt} \in L_i^{first}} x \quad (4)$$

$$c_i^{last} = \frac{1}{|L_i^{last}|} \sum_{p_{xt} \in L_i^{last}} x \quad (5)$$

and the spatiotemporal orientation $\phi_i^{st}$ is obtained by

$$\phi_i^{st} = \text{atan2}(h, c_i^{first} - c_i^{last}) \quad (6)$$

where $\text{atan2}(t,x)$ calculates $\arctan(tx^{-1})$ and adjusts the angles to give the angle between $(x,t)$ and the positive $x$ axis. The $\phi_i^{st}$ correspond to the angles of the vectors drawn in figure 3. An angle of 90° relates to the motion signature of a static object. Motion towards the left in image space produces angles between 90° and 0°, while motion towards the right results in values between 90° and 180°. We can use short processing volumes, so an object in motion will usually be present in a number of them and have straight first and last rows of pixels. Region four in figure 3 is not a perfect parallelogram. It rather has a bent edge and the last row is larger than the first row. This means there was some acceleration and also the object (or its projection) was scaled. Given the processing volume is short and such changes not to heavy, the vector is a good approximation of the motion. Similarly, regions three and five are motion signatures that are not rectangular and have spatiotemporal angles different from zero, even though they represent a static background. These errors are produced by the moving object passing in front of the background. In theory these are not errors, as there is spatiotemporal change. However, we do not want to highlight non-moving objects and luckily these signatures usually do not carry much weight. Their spatiotemporal angles are often still different from the moving objects, so they contribute to its saliency but won't acquire as much as the moving object does.

### 3.3 Bottom-up Saliency of Spatiotemporal Orientation

With spatiotemporal orientation angles $\phi_i^{st}$ being determined for each region of a slice, the bottom-up saliency $\uparrow S^i_{\phi^{st}}$ with values between 0 and 1 is computed straight forward by summing up the normalized difference of the angles (180° is the maximum difference between two regions' angles).

$$\uparrow S^i_{\phi^{st}} = \sum_{j=1}^{|R|} \frac{|\phi_i^{st} - \phi_j^{st}|}{180^\circ} w_{ij}^{\Delta} \quad (7)$$

where $w_{ij}^{\Delta}$ with values between 0 and 1 denotes a weight that depends on the distance between the centers of $R_i$ and $R_j$. Due to the fact that slices are usually short in the temporal dimension, the distance weight can be approximated by taking the difference of the $x$-components and normalizing it by the slice width.

### 3.4 Top-down Saliency of Spatiotemporal Orientation

The top-down saliency $\downarrow S^i_{\phi^{st}}$ is obtained with regard to the spatiotemporal orientation $\phi_T^{st}$ of a template region. Spatiotemporal top-down saliency $\downarrow S^i_{\phi^{st}}$ (ranging from 0 to 1) is then determined by the similarity between each region and the template.

$$\downarrow S^i_{\phi^{st}} = 1 - \frac{|\phi_i^{st} - \phi_T^{st}|}{180^\circ} \quad (8)$$

Here, we only make use of what spatiotemporal orientation contributes to saliency. However, a segmented motion signature carries more information. Color, size or other features from region-based spatial saliency computation can be used to create more specific top-down templates. The mechanisms can easily be transfered from the spatial context, in which they are used in (Aziz and Mertsching, 2008b). In this context we restrict the template mechanism to spatiotemporal orientation as it is directly induced by motion.

### 3.5 Transforming Saliency from Spatiotemporal to Spatial Domain

The purpose of saliency processing is to obtain a FOA of attention, which is a region in image space that is important in the current situation. The previous steps described how to calculate the saliency of motion signatures on spatiotemporal slices. These saliency values must be transformed back into image space and integrated with the spatial segmentation to make them accessible for FOA selection.

For each slice of the $X-T$ and $Y-T$ stack pixel-based master saliency maps are produced as a weighted combination of top-down and bottom-up saliency. The pixel activity for a $X-T$ slice can be

described as

$$a_{xt} = \sum_{i=1}^{|R|} (\uparrow S^i_{\phi^{st}} w_{\text{bu}} + \downarrow S^i_{\phi^{st}} w_{\text{td}}) \frac{1}{w_{\text{bu}} + w_{\text{td}}} \rho^i_{xt} \quad (9)$$

where

$$\rho^i_{xt} = \begin{cases} 1 & \text{if } (x,t) \text{ belongs to region } R_i \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

which is determined by a look-up in the label image that has been created when the slices of spatiotemporal stacks were segmented before the calculation of the feature magnitudes. Weights $w_{\text{bu}}$ and $w_{\text{td}}$ can be used to control the ratio of bottom-up and top-down influence. With $a^y_{xt}$ we denote the pixel activities $a_{xt}$ on slice $y$ of the $X-T$-stack and with $a^x_{yt}$ activities $a_{yt}$ on slice $x$ of the $Y-T$-stack, $a_{yt}$ is defined analogous to $a_{xt}$. A saliency volume $A_{xyt}$ is constructed by blending the pixel activities from both stacks.

$$A_{xyt} = (a^y_{xt} w_{xt} + a^x_{yt} w_{yt}) \frac{1}{w_{xt} + w_{yt}} \quad (11)$$

$X-Y$ slices extracted from $A_{xyt}$ already constitute pixel-based saliency maps, which are the final results of many models. To obtain a more differentiated focus that is a region (or a proto-object), the pixel activities are grouped with regard to the spatial regions $Q^t_i$ of frame $t$, which have been created initially in the spatial segmentation step (1b, in figure 2). So the final motion saliency map $M^t$ for frame $t$ in spatial domain is constructed by iterating over the image (of size $X \times Y$), summing and averaging the intensities that belong to each region.

$$M^t_i = \frac{1}{|Q^t_i|} \sum_{x=1}^{X} \sum_{y=1}^{Y} A_{xyt} \rho^{t,i}_{xy} \quad (12)$$

with $\rho^{t,i}_{xy}$ indicating the region membership check and yielding 1 if the pixel at $(x,y)$ is a member of region $Q^t_i$ and 0 else. Region size in pixels is denoted as $|Q^t_i|$.

For the experiments in the scope of this paper, for each frame $t$ the region $Q^t_i$ with the highest corresponding motion saliency $M^t_i$ is selected to constitute the FOA. As $Q^t$ represents region lists of the same kind as in (Aziz and Mertsching, 2008a), the spatial saliency processing there can be performed on them and the saliencies of spatial and spatiotemporal processing can easily be merged.

# 4 EXPERIMENTS

## 4.1 Saliency Results

It is virtually impossible to obtain objective measures for correctness of the output of attention models. Ground truth from human subjects, obtained by eye-tracking or manual marking of salient spots, is heavily influenced by top-down processes, which cannot be fully modeled with today's technical systems. Covert attention shifts are not accounted for in eye-tracking experiments and manual marking relies on introspection. Observers report full objects even if only some salient part of it drew their attention. Despite these difficulties, ground truth obtained in such ways has been used to evaluate the output of attention models. (Mahadevan and Vasconcelos, 2010) use manually created masks that isolate moving foreground objects and evaluate their model in a foreground-background-classification task. They use receiver operator characteristics (ROC) curves to quantify the models' classification success. A similar ROC-based analysis based on eye-tracking results, originally from (Itti and Baldi, 2006), was done by (Seo and Milanfar, 2009) for their model. These techniques cannot be used to evaluate our model. One reason is that, due to its region-based nature, the output consists of maps of salient regions which are not necessarily full objects. The inner regions of a moving object may not be salient, because their neighbors show the same motion. In an ROC analysis based on object masks this is reflected in a high false negative rate. Similarly, fixations from eye-tracker data will often hit non-salient regions of a moving object. Additionally, the test sequences and eye-tracker results from (Itti and Baldi, 2006) are not useful to evaluate a purely *spatiotemporal* model, as under the free-viewing conditions, *spatial* conspicuities (and top-down influences) are also likely to influence the gaze.

To enable quantitative evaluation, at least to some degree, we report simple hit-counts for the clips processed. We award one hit for each frame, where the FOA was assigned to a location displaying target motion. Fixations were also counted as hits when an element adjacent to the object in motion was highlighted because it was disturbed by the moving objects. These fixations are still very close to the moving target. However, as this is a region-based approach, it is our aspiration to highlight regions that belong to the target. Strict hit rates, where only such hits are considered, are given in round brackets behind the lenient values (see table 1).

To enable a qualitative visual comparison, we visualize our model output for sequences that have been used for the evaluation of other state-of-the-art models. The clips that we processed are from (Belardinelli et al., 2008), the *KTH* data set (Schüldt et al., 2004), the Weizmann data set (Gorelick et al., 2007), (Mahadevan and Vasconcelos, 2010) and our own sequences (*GET*). The prefixes of the clip names in ta-

Table 1: Clip properties and bottom-up hit rates (and strict hit rates) for all evaluated clips.

| Clip name | Frames (evaluated) | Hits in % (strict %) |
|---|---|---|
| BELA_DOTS | 40 (40) | 75 (75) |
| BELA_WALK | 40 (40) | 77.5 (60) |
| BELA_SHAKE | 30 (30) | 96.7 (90) todo |
| BELA_FLICKR1 | 30 (30) | 96.7 (53.3) |
| KTH_P01_WALKING | 80 (70) | 95.7 (94.3) |
| KTH_P02_BOXING | 100 (100) | 95 (90) |
| KTH_P03_HANDCLAP | 100 (100) | 71 (69) |
| KTH_P04_JOGGING | 50 (44) | 86.7 (81.8) |
| KTH_P05_RUNNING | 30 (25) | 88 (88) |
| KTH_P06_HANDWAVE | 100 (100) | 99 (85) |
| WEIZ_DARIA_JUMP | 60 (60) | 100 (91.7) |
| WEIZ_DENIS_SIDE | 50 (50) | 98 (98) |
| WEIZ_ELI_BEND | 60 (60) | 71.7 (45) |
| WEIZ_IDO_SKIP | 30 (30) | 83.3 (76.7) |
| WEIZ_IRA_JACK | 70 (70) | 95.7 (81.4) |
| WEIZ_LENA_PJUMP | 40 (40) | 95 (90) |
| MAHA_SKIING | 110 (110) | 47.3 (47.3) |
| MAHA_TRAFFIC | 190 (190) | 98.9 (98.9) |
| MAHA_LAND | 50 (50) | 0 (0) |
| GET_LTR1 | 20 (20) | 95 (95) |
| GET_LTR2 | 20 (20) | 100 (100) |
| GET_LTR3 | 20 (10) | 100 (100) |
| GET_RTL1 | 20 (20) | 95 (95) |
| GET_RTL2 | 20 (20) | 100 (100) |
| GET_RTL3 | 20 (20) | 100 (100) |
| GET_EGO_MOTION | 80 (80) | 88.8 (78.8) |
| GET_TD | 30 (30) | 100 (96.7) |

ble 1 refer to the source. Input frames were scaled to $240 \times 180$, except clip *BELA_DOTS*, which was processed at its original resolution of $256 \times 256$ to avoid that the artificial stimuli (dots) become too small for the parameter set. Segmentation parameters $\Gamma^h$, $\Gamma^i$, $\Gamma^s$, $\tau^h$, $\tau^i$ and $\tau^s$ were all set to 8. Minimum region size for spatial frames is $\mu^s = 20$ and for spatiotemporal slices $\mu^{st} = 10$ (see section 3.1). Except for our *GET* clips all input was processed in gray-scale as they were processed by the respective models.

Results of our system processing clips from (Belardinelli et al., 2008) are depicted in figure 4. Their approach is also based on spatiotemporal slices but is pixel-based. Clip *BELA_DOTS* shows circulating dots, with one dot being faster than the rest (only the motion of the target is indicated in the figure). Our model was able to select the correct dot in almost all frames (see table 1 for hit rates) and its spiraling trace can be seen in the saliency volume visualization. *BELA_SHAKE* shows a surveillance scenario (originally from (CAVIAR, 2001)), where two people approach each other and shake hands. Saliency peaks can be seen where the motion is taking place. The remaining two clips shown in the figure have also reasonable hit rates (the focus is on a plausible target the most time) but show a lot of activity in the

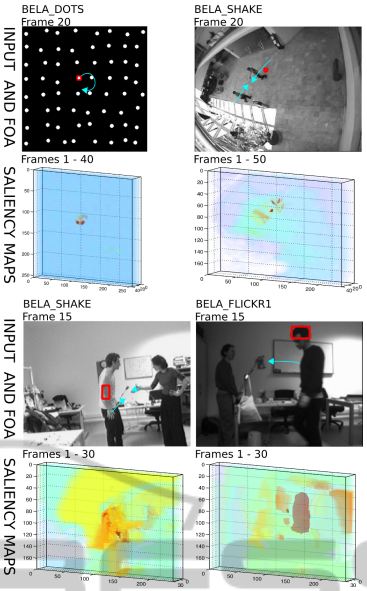


Figure 4: Results for clips from (Belardinelli et al., 2008). Exemplary frames with FOA (red rectangle). Manually added arrows indicate motion. Volumes show saliency maps over time.

background, too. A medium amount of saliency is assigned to background regions when they are crossed by the moving persons.

Figure 5 shows results from our model for clips of people performing different actions. They were also processed and depicted by (Seo and Milanfar, 2009). As the arrows in the figure indicate, different kinds of motion are contained. Some clips, such as *KTH_P01_WALKING* feature persons moving through the complete image space, while others such as *KTH_P02_BOXING* show motion only in some part of the person. The exemplary frames depicted for *WEIZ_ELI_BEND* and *WEIZ_IRA_JACK* are examples for relaxed hits as they strictly do not hit the moving person, but they are clearly induced by the motion.

Next we look at figure 6, which shows our results for scenes that also have been processed — and their results depicted — by (Mahadevan and Vasconcelos, 2010). *MAHA_SKIING* is heavily affected by noise induced by snow and poor sight. This is reflected to some degree by the hit rates. Overall, the saliency distribution looks reasonable. *MAHA_TRAFFIC* shows a traffic surveillance scene, of rather low contrast. Our system performs good hitting targets in almost every frame. For *MAHA_LAND* we show a very poor performance, the target is never hit. The scene contains slight relative motion and ego motion as the camera follows the target. Theoretically, the region-based spatiotemporal approach can deal with this sort of scenes, as moving background objects and a target

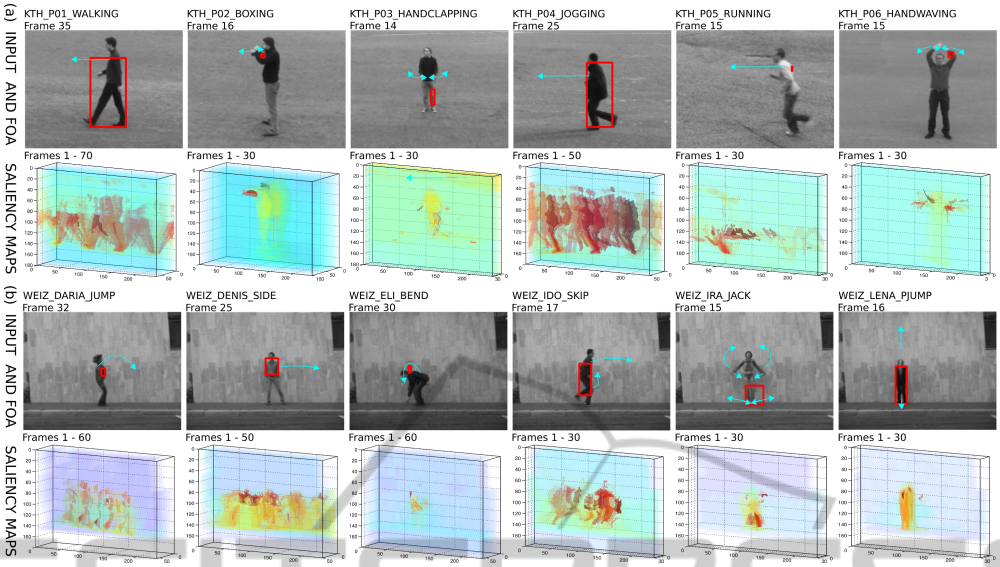


Figure 5: (a) Results for the KTH set; (b) the Weizmann set. Both data sets were also used by (Seo and Milanfar, 2009).

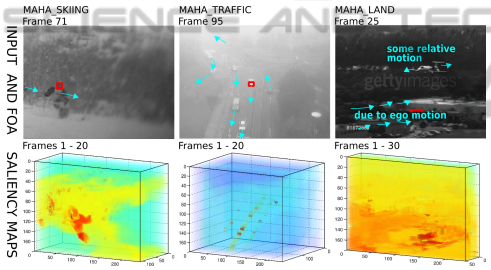


Figure 6: Results for clips from (Mahadevan and Vasconcelos, 2010).

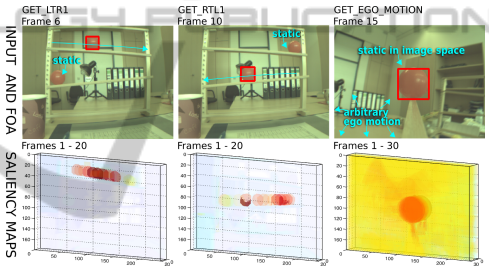


Figure 7: Results of bottom-up processing with our own stimulus material.

(almost) static in image space should produce motion signatures that contrast in orientation. However, in this scene the background has a low contrast and so orientation in spatiotemporal slices is more likely to differ over time due to changes in the segmentations. That the model is not principally unable to deal with ego motion is demonstrated in the following, when looking at *GET_EGO_MOTION*.

We now turn to results obtained from experiments with our own stimulus material. Figure 7 shows clips with a red ball in motion. Clips *GET_LTR1* and *GET_RTL1* are exemplary for a series of three clips each, where the red ball is rolling on tracks from left to right (*LTR*) and right to left (*RTL*), respectively. This simple motion is reliably highlighted (see Table 1). The scenes also feature an identical but static red ball, as a control, which demonstrates that motion and not a static attribute is the relevant feature. The control ball is never selected by the model. In *GET_EGO_MOTION* the ball was connected to the camera to stay in a position which is static in image space, while the camera was panned and tilted randomly. The model can handle this kind of ego motion quite well. The saliency volume visualization shows relatively high saliency values in the environment, but highest saliency is still assigned to the target.

In figure 8 we show results of a top-down experiment. The input clip contains a scene with two red balls on tracks rolling in opposite directions. Additionally, a horizontally flipped version of this clip was used. A bottom-up processing of the clip (a control) is shown in the figure as *GET_TD (Bottom-Up)*, where high saliency values occur for both balls. For *GET_TD (Top-Down LTR)*, a template was used to bias the model to prefer the left-to-right moving ball by the top-down mechanism described in section 3.4. The template with $\phi_{\mathrm{T}}^{sf_X} = 172^\circ$ and $\phi_{\mathrm{T}}^{sf_Y} = 160^\circ$ was obtained experimentally by roughly observing the angles of motion signatures that led to the selection of the ball during bottom-up processing of clip *GET_LTR1*. The saliency volume visualization shows the trace of the upper ball (it is only slightly visible

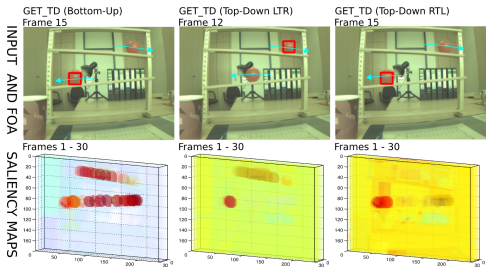


Figure 8: Results of our top-down experiments. The left row shows a bottom-up processing as a control.

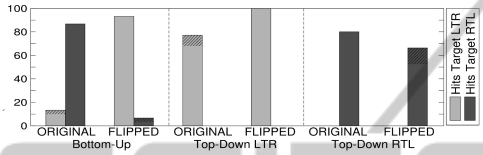


Figure 9: Hit rates from the top-down experiment. Hatched tips indicate hits that did not comply with the strict hit criterion.

as it is occluded by the later frames). Additionally, a saliency peak of the other ball (the prominent red dot) can be seen in *GET_TD (Top-Down LTR)*, which occurs due to the fact that the ball bounces off the barrier, changing its direction to target direction towards the end of the clip. To create the *RTL* template, the previous template is horizontally flipped (subtracted from 180°), $\phi_T^{stX} = 8°$ while the $Y$-component is kept as is. Results of the bottom-up control as well as processing with templates for each direction are shown in figure 9. All top-down processings eliminated selections of the ball moving in the opposite direction and selected the target ball in a large portion of frames. Due to perspective and physically different balls, the projected motion differs in all variations. Interestingly, the best result was observed for the experimentally determined template used on the flipped clip. This shows that template creation (which in future work will be done automatically by learning or feedback-loops) is crucial. The selection frequency can also be optimized by incorporating further features, which is discussed in section 3.4.

### 4.2 Performance and Continuous Processing

The system was implemented as nodes for ROS, Robot Operating System (Quigley et al., 2009), which run in parallel (or interleaved, depending on the machine). In most of our tests, input data was streamed in from image sequences and the system produced output on the fly. Alternatively, it can be connected to a camera, which was done for clip *GET_EGO_MOTION*. The system was divided into two main nodes that run in parallel. One continuously collects frames to form processing volumes and sends them to the second node, where all other processing is done. This division allows the frame collector to continue during the saliency processing and avoids gaps in the output. The input frame rate is adjusted so that a new volume is prepared when the saliency processing of previous one completed. With the configuration used in the previously described experiments the system was in a balanced state at approximately ten frames per second.

As volumes must be collected first, a certain lag between the input and output is unavoidable. Volumes of ten frames each were used in our experiments, so at ten frames per second, the oldest frame is about a second old when the volume enters processing. Additional to this conceptual lag (reducible when higher frame rates become possible) the processing itself adds to the delay. Depending on the complexity of the scene, it adds up to one or two seconds. A camera processing lagged behind even more (up to 5 seconds), as the camera driver and image downscaling produced additional load. These tests were performed on a dual core system (2.4 GHz) without explicit optimization. There is some margin, as $X-T$ and $Y-T$ stacks are processed sequentially now. Their processing, as well as the initial segmentation, could be separate parallel processes to better exploit multi-core machines.

We made the interesting observation that ignoring the $Y-T$ stack, which reduces processing time to the half, has only little effect on the output quality for most natural scenes. For some scenes, we collected hit rates (relaxed and strict) for such a "half processing". Table 2 contains hit rate differences to the full processing. Most of the scenes contain mainly horizontal motion (as natural scenes usually do) and it is no surprise that they were little affected. *MAHA_TRAFFIC*, however, has mainly vertical motion and still only five out of the 190 frames were missed due to the reduction.

## 5 CONCLUSIONS

We demonstrated that motion saliency can be processed in a region-based way. The results are brought back into a spatial image space segmentation, which consists of region-lists of the same form as used by (Aziz and Mertsching, 2008a), so integration with spatial saliency is possible and subject of future work. The proposed method processes spatiotemporal and

Table 2: Differences between a full processing and processing only $X-T$. Strict values are given in round brackets.

| Clip name | Hit diff. (frames) | Hit rate diff. (Percentage Points) |
|---|---|---|
| BELA_WALK | 3 (1) | 7.5 (2.5) |
| KTH_P06_HANDWAVE | 4 (4) | 4 (4) |
| DARIA_JUMP | 0 (6) | 0 (10) |
| MAHA_TRAFFIC | 5 (6) | 2.6 (3.2) |
| GET_LTR1 | 0 (0) | 0 (0) |
| GET_RTL1 | -1 (-1) | -5 (-5) |
| GET_TD | 2 (2) | 6.7 (6.7) |

spatial saliency independently, which enables different spatial and temporal resolutions for this integration. The visualized output from the experiments shows reasonable saliency deployment and the hit counts reflect good results for most of the test clips from a heterogeneous set. Top-down experiments were conducted to show how the model can be influenced to prefer a direction of motion, the mechanism can be extended to include further features. Our system is able to perform online on continuous input. The result lags behind up to a few seconds which is due to the concept (collecting a volume first) and computation time. We demonstrated that by using only $X-T$ slices, the lag can be reduced with only little influence on the quality of the outcome. In future work we will integrate spatiotemporal with spatial saliency processing and focus on grouping the regions to from "real" objects based on the attentional results to enable a quantitative comparison with manually marked test clips or eye-tracker data.

## ACKNOWLEDGEMENTS

This work was supported by the German Research Foundation (DFG) under grant Me 1289/12-1(AVRAM). The authors also wish to thank Konstantin Werkner for improvements suggested for the algorithms and Dr. Zaheer Aziz for his useful comments on the manuscript.

## REFERENCES

Aziz, M. Z. (2009). *Behavior adaptive and real-time model of integrated bottom-up and top-down visual attention*. Dissertation, Universität Paderborn.

Aziz, M. Z. and Mertsching, B. (2008a). Fast and robust generation of feature maps for region-based visual attention. In *IEEE Transactions on Image Processing*, volume 17, pages 633–644.

Aziz, M. Z. and Mertsching, B. (2008b). Visual search in static and dynamic scenes using fine-grain top-down visual attention. In *ICVS*, volume 5008, pages 3–12.

Belardinelli, A., Pirri, F., and Carbone, A. (2008). Motion saliency maps from spatiotemporal filtering. In *WAPCV*, pages 112–123.

CAVIAR (2001). EC funded caviar project/IST 2001 37540; http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/. [Online; accessed 5-September-2011].

Cui, X., Liu, Q., and Metaxas, D. (2009). Temporal spectral residual: Fast motion saliency detection. In *Proc. ACM Multimedia*, pages 617–620. ACM.

Goodale, M. A. and Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20–25.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. (2007). Actions as space-time shapes. In *IEEE PAMI*, volume 29, pages 2247–2253.

Guo, C., Ma, Q., and Zhang, L. (2008). Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In *IEEE CVPR*, pages 1–8.

Hou, X. and Zhang, L. (2007). Saliency detection: A spectral residual approach. In *IEEE CVPR*, pages 1–8.

Itti, L. and Baldi, P. F. (2006). Bayesian surprise attracts human attention. In *NIPS*, pages 547–554.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. In *IEEE PAMI*, volume 20, pages 1254–1259.

K. Rapantzikos S. Kollias, T. A. (2009). Spatiotemporal saliency for video classification. *Signal Processing: Image Communication*, 24:557–571.

Livingstone, M. and Hubel, D. (1987). Psychophysical evidence for separate channels for the perception of form, color, movement, and depth. *The Journal of Neuroscience*, 7(11):3416–3468.

Mahadevan, V. and Vasconcelos, N. (2010). Spatiotemporal saliency in dynamic scenes. In *IEEE PAMI*, volume 32, pages 171–177.

Mahapatra, D., Winkler, S., and Yen, S.-C. (2008). Motion saliency outweighs other low-level features while watching videos. In *SPIE*, volume 6806.

Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). ROS: An open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Schüldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: A local svm approach. In *ICPR*, pages 32–36.

Seo, H. J. and Milanfar, P. (2009). Static and space-time visual saliency detection by self-resemblance. *Journal of Vision*, 9(12).

Tünnermann, J. (2010). *Biologically inspired spatiotemporal saliency processing to enhance a computational attention model*. Master's thesis, Universität Paderborn.

Wischnewski, M., Belardinelli, A., Schneider, W. X., and Steil, J. J. (2010). Where to look next? Combining static and dynamic proto-objects in a TVA-based model of visual attention. *Cognitive Computation*, 2(4):326–343.