Discovering Salient Objects from Videos using Spatiotemporal Salient Region Detection

Rajkumar Kannan[1*], Gheorghita Ghinea[2], Sridhar Swaminathan[3]

[1] College of Computer Sciences and Information Technology,
King Faisal University, Al Ahsa 31982, Saudi Arabia,
rkaruppan@kfu.edu.sa, Tel: +966-13-5899273

[2] Brunel University, Middlesex UB8 3PH, United Kingdom
george.ghinea@brunel.ac.uk

[3] Bishop Heber College (Autonomous), Tiruchirappalli 620017, India
sridarah@gmail.com

* Corresponding author

Abstract. Detecting salient objects from images and videos has many useful applications in computer vision. In this paper, a novel spatiotemporal salient region detection approach is proposed. The proposed approach computes spatiotemporal saliency by estimating spatial and temporal saliencies separately. The spatial saliency is computed estimating of color contrast cue and color distribution cue by exploiting patch level and region level image abstractions in a unified way. The aforementioned cues are fused to compute an initial spatial saliency map, which is further refined to emphasize saliencies of objects uniformly, and to suppress saliencies of background noises. The final spatial saliency map is computed by integrating the refined saliency map with center prior map. Temporal saliency is computed based on local and global temporal saliencies estimations using patch level optical flow abstractions. Both local and global temporal saliencies are fused to compute the temporal saliency. Finally, spatial and temporal saliencies are integrated to generate a spatiotemporal saliency map. The proposed temporal and spatiotemporal salient region detection approaches are extensively experimented on challenging salient object detection video datasets. The experimental results show that the proposed approaches achieve an improved performance than several state-of-the-art saliency detection approaches. In order to compensate different needs in respect of the speed/accuracy tradeoff, faster variants of the spatial, temporal and spatiotemporal salient region detection approaches are also presented in this paper.

Keywords: Salient region detection, temporal saliency, optical flow abstraction, spatiotemporal saliency detection, saliency map.

1. Introduction

Detecting salient regions from images and videos that captures the attention of *Human Visual System* is an interesting and difficult multi-disciplinary problem. Many computational visual attention models have been proposed over the years, to resemble the mechanism of the Human Visual System's remarkable ability to fixate conspicuous/salient regions from a visual scene. The field has gained considerable attention in the recent years, and has become an active area of research in computer vision due to its applications in object detection [1], object recognition [2], adaptive image and video compression [3], image retargeting [4,5], summarization of photo collections [4] and video summarization [6], where saliency detection is considered as an essential step towards achieving a vision goal.

Existing works on visual saliency detection can be roughly categorized into *bottom-up* and *top-down* approaches. Most of the earlier bottom-up approaches model humans' instinctive, stimulus-driven attention to distinct low level visual features such as color, intensity and orientation [7]. Recent bottom-up saliency detection approaches estimate saliency based on local, regional and global contrast [8,9,10]. Saliency computed by contrast-based bottom-up approaches is considered as *low-level saliency*. Top-down saliency detection approaches [11] simulate humans' task-driven attention driven by their knowledge, expectations, and current goals, thus it is considered *high-level saliency*. Goal-driven saliency is achieved by incorporating prior knowledge on visual features statistics and object level semantic features such as text, faces, etc. Some prior work [11,12] combines both bottom-up and top-down mechanisms into a unified framework for visual saliency detection. Due to their computational efficiency, adaptability and unsupervised nature, bottom-up approaches are widely used for saliency detection.

Bottom-up saliency is often computed by pixel/patch level *contrast* estimation, which is defined as the pixel's or patch's state of being different from its surroundings. Local contrast based saliency detection approaches [7,13] estimate saliency using multi-scale low level image feature analysis based on a biologically motivated saliency theory called *center-surround difference* mechanism [7], which makes it suitable for detection of human eye fixations in images. This further motivated *global contrast* based approaches [9,10], that operate on either patch level [5,10] or region level [2,9,14,15] image abstractions. When compared to the local contrast based approaches, global contrast estimation is computationally more efficient, and accurate for salient region detection. This has established global contrast estimation as a promising mechanism for salient object detection and segmentation.

Since a salient object comprises unique visual elements which cannot be represented well by their surroundings, the *rarity* of features is also widely used for saliency detection. The rarity of an image's element can be measured using self-information [16], graphic models [17], log-spectrum [3,18,19] and feature sparsity models [20]. The rarity of an image's features is measured in a global context most of the time. Saliency maps generated by these global rarity based saliency detection methods exhibit significant amount of blurriness. Moreover, these methods often emphasize small scale globally rare image elements which most of the time might be considered as noise by humans. More recent rarity-based visual saliency detection approaches belong to RARE algorithms family - RARE2007 [21], RARE2011 [22] and RARE2012 [23] effectively predict human gazes in the images with less background noises. The aforementioned local contrast-based and rarity-based visual saliency detection methods are more suitable for predicting human eye fixations rather than detecting salient objects in images. A comprehensive survey on recent state-of-the-art approaches in visual attention modeling, and a detailed study on comparison metrics for human gaze prediction can be found in [24] and [25] respectively.

Recent salient region detection approaches [5,10,15] have begun estimating another important cue for saliency called *color distribution*. Since color components of a salient object are always spatially compact rather than widely spread around the image, a lower spatial distribution of a color component indicates its higher spatial saliency, and vice versa. Apart from these two low-level saliency cues, another widely used high-level saliency cue is *center prior* [5,8,11,12,14,26]. The center prior gives more importance to regions that are near to image center, since salient objects are placed near the image center most of the time. The spatial salient region detection approach presented in this paper is based on our previous work in [27], which computes the aforementioned color contrast and color distribution cues by exploiting patch and region abstractions in a unified way. Usually, images patches are compared with each other to compute color contrast and color distribution cues for salient region detection. In our previous work, we have shown that the color contrast and color distribution of patches can also be efficiently computed by comparing them with a relatively low number of region abstractions.

Despite the significant progress made towards modeling visual saliency detection, most of the visual attention models have only a spatial saliency detection component, thus they work only on images. Relatively few spatiotemporal saliency detection models have been proposed for saliency detection in videos. Similar to spatial saliency detection, some research efforts adopted local motion contrast for

temporal saliency detection [28,29,30]. However, similar to that of local contrast based spatial saliency detection, the saliency maps produced based on local motion contrast often tend to emphasize boundaries of salient object in videos.

For temporal saliency detection, some models [3,31] simply include motion channel into the saliency detection framework. The motion channel of a video frame is computed as a *temporal gradient* [3,31], which is the intensity difference between two successive video frames. Detecting temporal saliency in this way works well as long as the camera is static. However, for videos with the presence of high background motion and camera motion, these methods often fail to detect salient objects accurately due to noises in temporal gradient. Most of these local motion contrast-based methods for spatiotemporal saliency detection need multi-scale feature analysis, thus they suffer from high computational complexity.

Optical flow computed between two consecutive videos frames is often treated as motion channel for temporal saliency detection [8,32,33]. Some of these works [8,32] directly incorporate the motion information computed using optical flow estimation into the saliency detection framework. However, high camera and background motions in a visual scene degrade the performance of optical flow based temporal saliency detection. Hence optical flow is abstracted at patch level to compute global motion contrast present in a visual scene [33]. Nonetheless, the histogram based patch level motion abstraction [33] suffers from high computational complexity and low discriminability.

In order to solve the aforementioned problems characteristic of spatiotemporal saliency detection, this paper proposes a novel and robust approach for temporal saliency detection in videos. The proposed method makes use of an efficient optical flow based patch level motion abstraction approach for computing local and global temporal saliencies. The local temporal saliency is computed as the *center-surround difference* of patch motions, where the global temporal saliency is estimated as the global *rarity* of a patch's motion. These two cues are fused to compute the temporal saliency which is then integrated with the spatial saliency for estimating the spatiotemporal saliency of a visual scene. Applications of spatial saliency detection such as image thumbnail generation, bounding box based object extraction, image retargeting, and video summarization do not need pixel accurate saliency maps, but require high speed saliency estimation [34]. To meet this requirement, faster variants of the proposed salient region detection methods are also presented in this paper.

The main contributions of this paper are summarized as follows:

1. In contrast to most of the biologically inspired spatiotemporal saliency detection approaches which are highly suitable for human eye fixation prediction, this paper proposes a novel and unified framework for spatiotemporal salient region detection from the view point of detection and segmentation of salient objects present in videos. Unlike most of the spatiotemporal saliency detection approaches which simply include motion channel into the saliency detection framework for estimating temporal saliency, the proposed approach separately estimates spatial and temporal saliencies by exploiting different salient region detection theories in a novel and unified way.
2. A novel and unified approach for temporal salient region detection is proposed in this paper. The proposed approach computes local and global temporal saliencies using a novel optical flow based patch level motion abstraction approach which makes the temporal saliency estimation robust to dynamic camera and background motions. The proposed multi-level center-surround differencing based local temporal saliency detection approach can be extended into spatial domain for salient object detection and human eye fixation prediction.
3. For spatiotemporal saliency-based applications that need a different speed/accuracy tradeoff, faster variants of the proposed temporal and spatiotemporal salient region detection approaches are also presented in this paper. The variants of the proposed approaches present a faster and more robust salient object detection performance, which also outperform the other methods in some cases.

The rest of the paper is organized as follows. Section 2 discusses related work on spatial, temporal and spatiotemporal saliency detections. Sections 3 and 4 introduce the approaches for spatial salient region detection and temporal salient region detection respectively. Experimental results of the proposed temporal and spatiotemporal approaches compared to the state-of-the-art methods are presented in section 5. Finally, section 6 concludes the paper with future work.

2. Related Work

A comprehensive survey on salient object detection can be found in [35]. However, this section briefly reviews the previous literature on bottom-up visual saliency detection and salient object detection, which is accordingly categorized into spatial, temporal and spatiotemporal saliency detection.

2.1 Spatial Saliency Detection

A biologically inspired *early representation* model for visual saliency detection was proposed by Koch and Ulman [36]. This further inspired Itti et al. [7] to propose a highly influential computational method which performs local *center-surrounded difference* analysis of image features such as color, intensity and orientation across multiple scales. The center-surround differencing mechanism is performed using the Difference of Gaussian (DoG) approach. Saliency maps generated by biologically inspired approaches are blurry and often contain highly emphasized small local features in the image which might be considered as noises.

Several approaches were proposed to improve Itti's model [7] including a fuzzy growing model [37] that estimates pixel-level dissimilarity in an image for saliency estimation. Liu et al. [8] proposed a color histogram-based computation of region-based center-surround difference mechanism. A graph-based saliency detection method using random walk is presented by Harel et al. [17]. Saliency maps produced by these methods often tend to overemphasize saliency near the edges rather than highlighting the salient region uniformly. Most of these biologically motivated local contrast based approaches require multi-scale feature analysis which makes them computationally infeasible for applications that need faster performance.

While almost all the saliency detection approaches work in the spatial domain of an image, purely computational models that work in the frequency domain of an image were also proposed in recent years. For instance, Hou et al. [18] proposed saliency detection using spectral residual in the amplitude spectrum of Fourier transformed image. Jung et al. [19] further extended [18] for local contrast detection, and combined the global and local saliencies into a unified approach. However, Guo et al. [3] showed that the Phase spectrum of the Quaternion Fourier Transform (PQFT) can be utilized for better saliency estimation. Despite these methods being considerably faster compared to the saliency detection methods operating in the spatial domain, they generate undesirable blurry saliency maps with saliency values highlighted near edges, corners and object boundaries.

Global contrast based approaches estimate saliency of an image element by computing its contrast with respect to the rest of the image elements in a global manner. Achanta et al. [38] proposed a frequency-tuned method for pixel-level saliency estimation defined as the dissimilarity between the Gaussian blurred image and the mean color of the image. This method often suffers from cluttered and textured image background, however. Goferman et al. [4] proposed a patch based global saliency detection by combining local and global contrast estimations. Since their approach needs multi-scale analysis, it suffers from high computational cost. Similar to the local contrast based methods, this approach also produces saliency maps with overemphasized object contours. Duan et al. [39] handles the combinatorial complexity behind global contrast estimation using dimensionality reduction. They defined saliency of a patch as spatially weighted dissimilarity in a reduced dimension space, which often results in significant loss of potential saliency details in the salient maps.

Cheng et al. [9] proposed a spatially weighted region-based global contrast estimation approach for saliency detection. Their global contrast based method often highlights background clutters, and detects only some parts of the salient object. Fu et al. [14] proposed a cluster based global contrast estimation for saliency detection. Since a cluster consists of disconnected pixels spread over the image, this approach does not consider the spatial distance between clusters for contrast computation. Determining the number of clusters is also a typical problem with this approach, by which saliency detection is often affected by an insufficient/unsuitable number of clusters. Moreover, the formulation of color distribution cue for saliency is difficult these two approaches. Ren et al. [2] clustered superpixels by Gaussian Mixture Model for saliency detection. In their work, saliency of a cluster is defined as its compactness which is estimated as the inter-cluster distance between clusters. Since region-based methods compute and assign saliency at region level, imprecise segmentation of regions always leads to degraded performance.

Recent approaches [5,10,15,40] estimate both color contrast cue and color distribution cue for a unified solution to salient object detection. Perazzi et al. [10] segmented an image into superpixels and computed color contrast and color distribution cues using an efficient high dimensional Gaussian filtering mechanism. However, their method sometimes highlights only some parts of the salient object. Fu et al. [5] proposed a superpixel-based saliency detection approach which uses spatially weighted color contrast, and color distribution estimations. Sometimes, their method fails to segment the salient object from a cluttered background. Gopalakrishnan et al. [40] also proposed a color and orientation distribution based spatial salient region detection approach. The color distribution based saliency is computed as the compactness and the isolation of a color cluster using a Gaussian mixture model (GMM) in the hue-saturation (H-S) space. The orientation distribution based saliency is estimated as the spatial variance of global orientation and orientation entropy contrast using orientation histogram of a local patch. In related work, Cheng et al. [15] employed a Gaussian Mixture Model based image abstraction approach for detecting salient regions, where the abstractions were used to estimate both color contrast and color distribution cues.

The existing saliency detection approaches work on either patch level [5,10] or region level [2,9,14,15] image abstractions. Each of these abstractions has their advantages. However, saliency detection with patch level image abstractions often suffers from quadratic runtime complexity. Also, the patch level saliency estimation often fails to highlight salient objects uniformly, and labels non-salient background noises as salient. On the other hand, region-based saliency estimation methods uniformly highlight the saliency of objects and suppress background saliencies with comparatively lower runtime complexity. However, region-based saliency detection methods totally depend upon the performance of the method used for image segmentation. Thus, imprecise segmentation or insufficient number of regions often results in poor performance.

Based on our previous work in [27], the spatial salient region detection approach presented in this paper combines both patch level and region level image abstractions which were separately considered for salient region detection in many others' works. Furthermore, a computationally efficient saliency refinement approach is presented to solve the saliency assignment issues in patch level saliency detection. In addition, faster variants of the method are also presented to achieve high speed saliency estimation in images.

2.2 Temporal Saliency Detection

Several temporal saliency detection models have been proposed over the years for detecting background regions in a visual scene, which is a complementary mechanism of saliency detection. Distribution of pixel intensities is represented by probability density function to predict the probability of background pixels in newly arrived video frames. Gaussian Mixture Model (GMM) is a most widely used probabilistic models for background modeling [41]. Since these models need exquisite tuning of several parameters that are involved, Elgammal et al. [42] proposed a parameter free probabilistic model for

background detection. Heikkilä et al. [43] proposed patch level texture based background modeling using histogram of Local Binary Pattern (LBP). This method is often affected by small scale textured image noises. In order to solve this problem, Liao et al. [44] proposed a novel texture descriptor called Scale Invariant Local Ternary Pattern (SILTP) for background modeling. These probabilistic models usually need a training phase in order to learn statistics of the background features. Background probability based temporal saliency detection often fails to work on videos with dynamic background or moving camera.

Optical flow estimation is often adopted for suppressing noises induced by the aforementioned problems. Wixson et al. [45] estimated temporal saliency of a video by computing directionally-consistent optical flows over successive video frames. Bugeau et al. [46] proposed an approach for removing background pixels using camera compensation, and used the mean shift algorithm for segmenting the foreground region of a salient motion. However, sometimes the salient motion detected by these methods might belong to the background, since these methods do not consider spatial saliency for detecting salient objects from videos. Moreover, the approaches for temporal saliency detection and the background modeling do not work efficiently on videos captured with significant camera motion.

2.3 Spatiotemporal Saliency Detection

Only a few models for saliency detection comprise components for both spatial and temporal saliency detections, thus they can work in images as well as videos. Zhai et al. [47] computed correspondence between keypoints of successive video frames for estimating temporal saliency. They used histogram based pixel level global contrast estimation for spatial saliency detection. Since this method uses keypoints for estimating temporal saliency, exact localization and segmentation of salient regions from a video frame becomes difficult. Inspired by retina mechanism, Marat et al. [28] computed spatiotemporal saliency by applying spatial and temporal filters in video frame. However saliency detection considering only local context results in degraded performance. Seo et al. [29] proposed to measure saliency as the center-surround contrast based on a pixel's resemblance to its surroundings. Since their approach works on downscaled images, the method results in highly blurred saliency maps. Mahadevan et al. [30] proposed a probabilistic approach for discriminant center-surround spatiotemporal saliency detection by using patch level dynamic textures. A common problem with these local motion contrast based methods is that they often emphasize saliency near object boundaries, thus they need multiscale feature analysis to reduce this effect which is computationally expensive.

Some of the spatiotemporal saliency detection models [3,31] compute temporal saliency by including motion channel into the saliency estimation strategy, in addition to color, intensity and orientation channels. These models use temporal gradients calculated from the intensity difference of two successive video frames. Kim et al. [31] computed temporal saliency as the sum of center surround difference of temporal gradients of the patches. Similar to the aforementioned local motion contrast based models, the saliency maps produced this method also often emphasize saliency near object boundaries. Guo et al. [3] simply incorporated motion channel into the Fourier Transform based saliency detection framework. Computing temporal saliency by treating temporal gradients as motion channel works well as long as the camera is static. Otherwise, the temporal saliency map for visual scene with high camera motion and dynamic background tends to contain much noise, rather than salient objects.

To overcome these issues, optical flow estimation is often adopted for computing motion contrast in a visual scene for measuring temporal saliency. Chen et al. [32] detected space-time interest points by a spatiotemporal Harris corner detector, which are fused with optical flow for spatiotemporal saliency detection. Liu et al. [8] computed spatial saliency by estimating multi-scale contrast, center-surround histogram and color distribution in local, regional and global manner. Their method computed temporal saliency by including SIFT flow based 2D motion vectors into the saliency detection framework. However their saliency maps often contain saliency values are spread around the image than being spatially compact. In related work, Wu et al. [33] computed temporal saliency as global motion contrast

using patch level histogram based optical flow abstraction called as Histogram of Average Optical Flow (HOAOF). This motion abstraction reduces the influence of high camera and background motions in optical flow. However, this histogram based motion contrast estimation is computationally expensive, and is less discriminative to different motions since it uses a smaller number of quantized flow orientations.

In order to solve these aforementioned issues, this paper proposes a novel approach for temporal saliency detection. The proposed approach computes local and global temporal saliencies separately, and fuses them to obtain temporal saliency map. Local and global temporal saliencies proposed in this paper are inspired from the work proposed in [13]. In their work, sparse coding based patch level image abstraction is used for local and global spatial saliency detection. They exploited both local and global considerations for saliency detection, which had been regarded separately by many works. Their method is proposed for spatial saliency detection which is experimented with human eye fixation prediction. In this paper, however, the saliency of a patch in local and global context is employed for temporal salient region detection, which is experimented with standard salient object detection video datasets.

Figure 1 depicts the proposed spatiotemporal salient region detection framework[1]. Usually, most of the biologically inspired visual saliency detection approaches compute spatial saliency by operating on different image channels such as color, intensity, orientation, etc. Those models are further extended to detect temporal saliency in videos just by incorporating motion channel into the saliency detection framework. Nonetheless, the proposed approach separately computes spatial and temporal saliencies based on different salient region detection theories in spatial and temporal domain, thus it can be considered as a unified framework for spatiotemporal salient region detection.

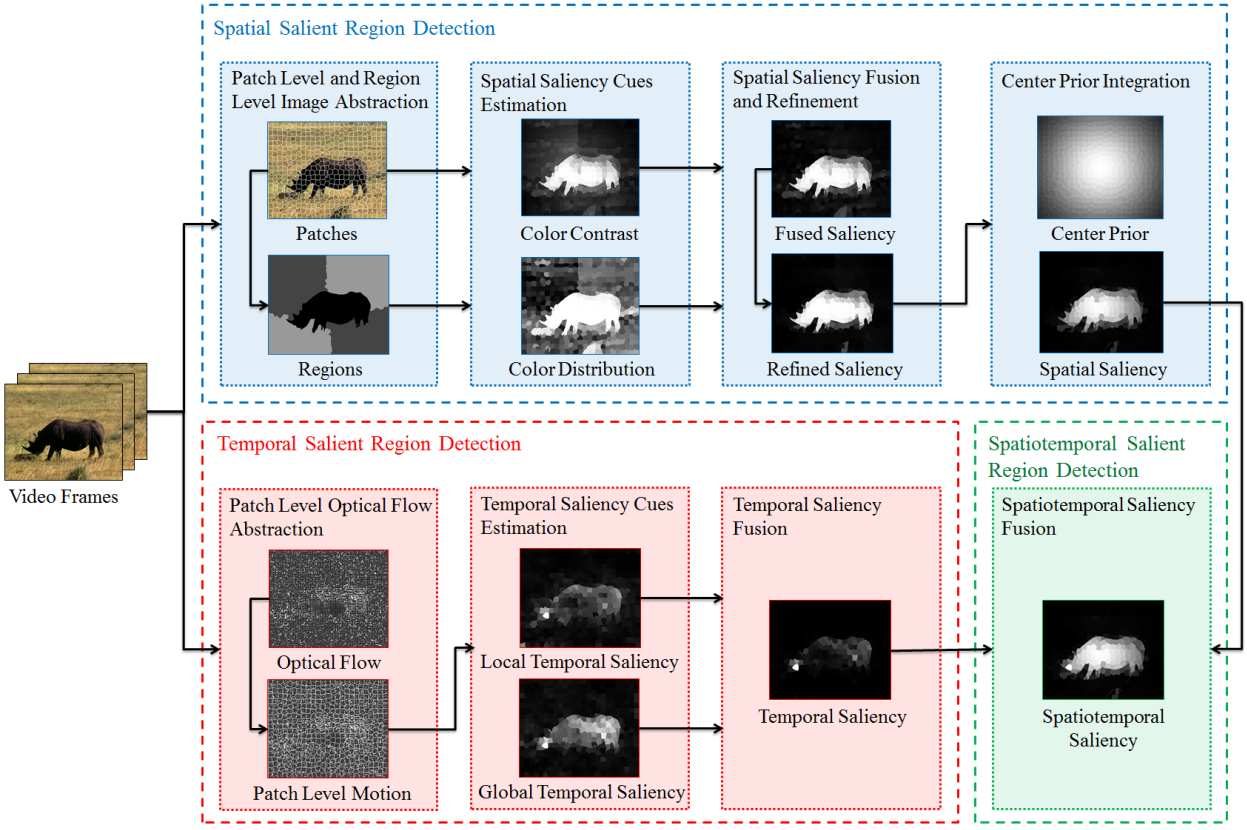
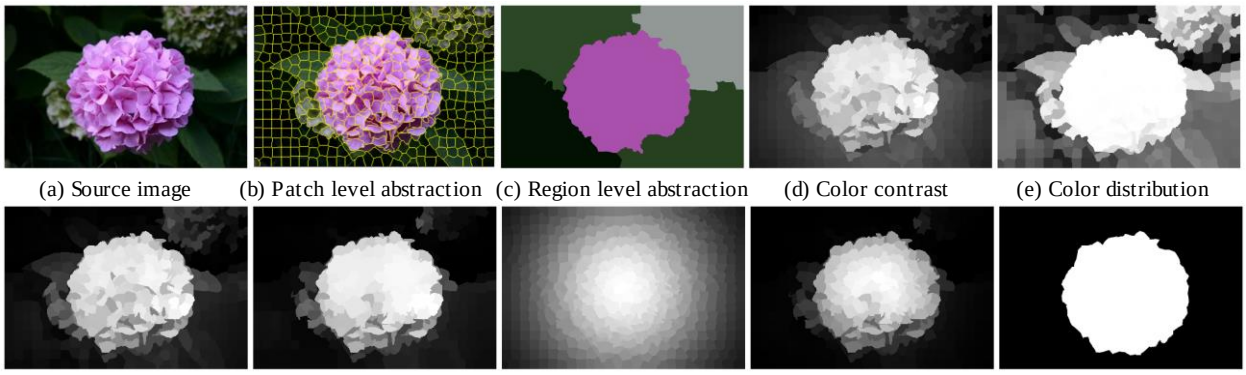


Fig.1. Spatiotemporal salient region detection architecture.

[1] The prototype software of the proposed approaches is submitted as a part of the supplementary material.

3. Spatial Salient Region Detection

The spatial salient region detection method is described in this section. Figure 2 depicts the main phases involved in the spatial salient region detection approach.



(a) Source image (b) Patch level abstraction (c) Region level abstraction (d) Color contrast (e) Color distribution

(f) Fused Saliency (g) Refined saliency (h) Center prior map (i) Final saliency map (j) Ground truth

Fig. 2. Main phases of the spatial salient region detection approach. First an image (a) is abstracted at patch level (b). The patch level abstractions are used for region level image abstraction (c). Both of these abstractions are further used for color contrast estimation (d) and color distribution estimation (e). These two cues are fused to compute saliency of an image (f) which is further refined adaptively (g). Lastly, center prior map (h) is integrated to generate the final saliency map (i).

3.1 Patch Level Image Abstraction

Saliency computation with pixel-level comparisons [47] on an image with thousands of pixels is computationally expensive. Moreover, saliency estimation on down-sampled images produce highly blurred saliency maps, thus localization or segmentation of salient objects becomes difficult. To reduce the computational complexity experienced in pixel-level saliency estimation, as in [2,5,10], the given image I is segmented into small scale edge preserving regions called *superpixels*.

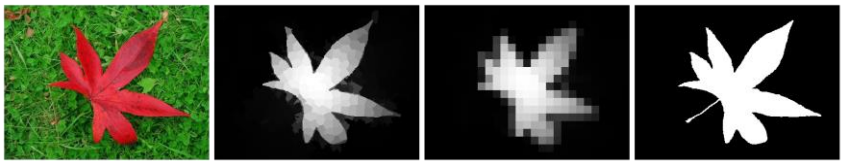
Since the computational cost of a superpixel segmentation algorithm is directly proportional to the number of pixels in an image, larger dimensions of the input image degrade the computation time of patch segmentation. Also, the video frames with higher dimensions increase the time required for optical flow estimation used in the proposed temporal saliency detection approach. In order to maintain size uniformity among different images/video frames and to reduce the computational overhead experienced with the images with large dimensions, the given image/video frame in an arbitrary size is resized with the maximum image dimension being 400 pixels. For applications such as image segmentation and image retargeting, the final saliency maps are again resized into the original image resolution. So, the input images are not resized with very lower resolution even though resizing input images with very smaller dimensions can fasten the superpixel segmentation and optical flow estimation.

The SLIC superpixel segmentation [48] is employed to achieve patch level image segmentation, which produces highly compact and edge preserving homogenous superpixels. The number of superpixels N is set to 500 for the experiments. Each superpixel s_i is represented by a mean color sc_i (in CIELab color space) and a spatial position sp_i (x and y image coordinates).

SLIC abstracts an image effectively, albeit it suffers from slow computation speed. To accomplish faster patch level image abstraction, the image is segmented into equal sized non-overlapping square patches of size $w \times w$. Since, the image dimensions might not be exactly divisible by w, an image is resized into a size that is both divisible by w and has a minimal change in the aspect ratio of the original image. Similar to N, the parameter w determines the tradeoff between speed and accuracy in saliency detection. Smaller sized patches abstract an image better than larger sized patches, because larger patches might contain pixels from both the foreground and background. However, smaller sized square patches increase

the computational complexity in saliency detection. To balance both speed and accuracy, w is empirically set to 15, which segments a 400×300 image into 540 square patches. Similar to superpixels, each square patch is also represented by a mean color and a spatial position. Figure 3 depicts the saliency maps computed with superpixel-based and uniform sampling-based patch segmentations. This shows that superpixel-based saliency maps are more accurate than square patch-based saliency ones.

3.2 Region Level Image Abstraction



(a) Source image (b) Superpixel based saliency map (c) Square patch based saliency map (d) Ground Truth

Fig. 3. (a) Source image, (b) and (c) saliency maps estimated with superpixel based patch segmentation, and uniform sampling based patch segmentation, (d) ground truth.

The segmented superpixels are grouped into regions by employing a spectral clustering algorithm [49]. Let $G = \{V, E\}$ be a weighted undirected graph, having nodes $V = \{s_1, s_2, s_3 \ldots s_N\}$ corresponding to the set of superpixels where the edges E represent the set of links that only connect adjacent superpixels in an image. An $N \times N$ affinity matrix A is constructed for the graph, where each element a_{ij} is a weight of an edge that denotes the similarity between adjacent superpixels s_i and s_j. The weight a_{ij} is defined as:

$$a_{ij} = \begin{cases} sim(s_i, s_j) & \text{if i} \neq \text{j} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The term $sim(s_i,s_j)$ is a Gaussian function that measures the color similarity between adjacent superpixels s_i and s_j, which is defined as follows,

$$sim(s_i, s_j) = \exp\left(-\frac{d(sc_i, sc_j)}{2\sigma_1^2}\right) \quad (2)$$

where $d(sc_i,sc_j)$ is the Euclidean distance between colors of the superpixels in CIELab space. This distance is normalized to [0,1] range using the min-max normalization method. The scaling parameter σ_1 is set to 0.4 as in [50]. Then, the spectral clustering algorithm [49] is applied to cluster the graph G into M clusters. Existing region-based saliency detection approaches [14] set M common for all the images. However, this often results in either over-segmentation or under-segmentation of regions. Here, the concept of *Eigen heuristics* [51] is used to automatically determine the number of clusters M. Still, M is restricted to be within a specific range $[M_{min}, M_{max}]$, where M_{min} and M_{max} are set to 5 and 10 correspondingly.

Each region r_j is then represented by a prototype that comprises of dominant color rc_j (in CIELab color space) and spatial position rp_j (x, y image coordinates). Determining the prototype of a region is also a key problem in region abstraction. Averaging superpixels' colors of a region is often affected by segmentation errors due to high feature variation. Here, the process of region prototyping is formulated as a multivariate feature mediation problem. So, geometrical mediation is used to determine the dominant color rc_j of a region r_j which is defined as,

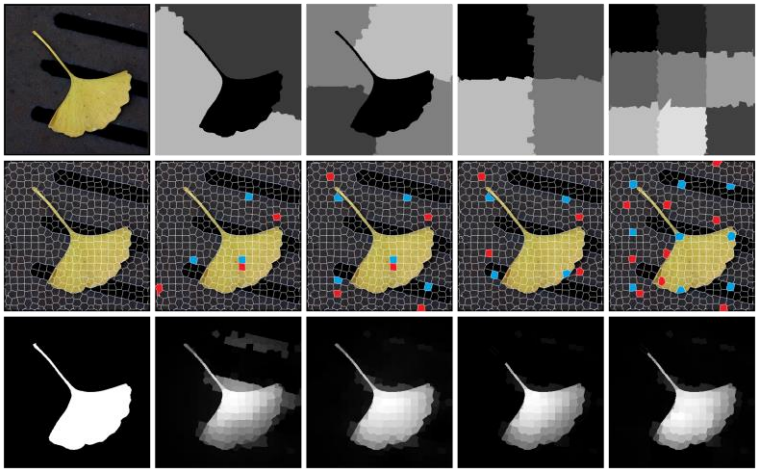
$$rc_j = \arg\min_{rc_j \in sc} \sum_{i=1}^{|r_j|} d(sc_i, rc_j) \quad (3)$$

where $|r_j|$ denotes the number of superpixels in region r_j and sc is the set of colors of superpixels in region r_j. The above objective function finds a superpixel that has the minimum color distance from the rest of the superpixels of its region, and sets its color as dominant color of the region. The spatial position

of a region rp_j (i.e. geographical midpoint) is also determined in the same manner as it is performed for dominant color estimation. The spatial position rp_j is estimated as center of minimum distance:

$$rp_j = \arg\min_{rp_j \in sp} \sum_{i=1}^{|r_j|} d(sp_i, rp_j) \tag{4}$$

The above equation finds a superpixel that has the minimum spatial distance from the rest of the superpixels of its region, and sets its position as the spatial position of the region. The term *sp* is the set of spatial positions of superpixels in region r_j.



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

Fig. 4. Top (a) Source image. Middle (a) superpixels of the image. Bottom (a) ground truth saliency map. Top (b)-(e) region segmentation with M=3, M=5, z=2 and z=3. Middle (b)-(e) red patches are the superpixels that are chosen for representing dominant colors of the corresponding regions, where blue patches are the superpixels that are chosen for representing spatial positions of the corresponding regions. Bottom (b)-(e) corresponding saliency maps produced for region segmentation in Top (b)-(e) (better viewed in color).

Even though the graph-based region segmentation divides an image into edge preserving homogenous regions, comparatively faster region segmentation is achieved by uniformly segmenting an image into $z \times z$ rectangle regions. The superpixels or the square patches that fall into a rectangular area are considered to belong to that region. The parameter z is set to 2, where the number of uniformly sampled regions becomes 4. The rectangular regions are also represented with dominant color and spatial position using the region prototyping approach. Since the region prototyping approach is robust to region segmentation errors, uniform region segmentation also achieves outperforming saliency estimation. Figure 4 depicts the selection of superpixels for region abstraction mechanism with different number of regions. It shows the robustness of the method to the number of regions and the type of region segmentation method.

3.3 Color Contrast Estimation

The color contrast of a patch s_i is measured by computing the spatially weighted color contrast to all regions of the image except the region it belongs. The color contrast estimation is formulated as:

$$con(s_i) = \sum_{j \neq i} \frac{|r_j|}{N} \cdot \exp\left(-d(sp_i, rp_j) \cdot \beta_1\right) \cdot d(sc_i, rc_j) \tag{5}$$

The term $|r_j|$ is the number of superpixels in a region r_j, which is used to favor color contrast to bigger regions to have more influence. This term is normalized to the [0,1] range by dividing it with the total number of superpixels N. The exponential function gives spatial weighting to the contrast measure, where

contrast to the spatially near regions will be given more weight than the farther regions. The parameter β_1 is a scaling factor in the exponential function that controls the spatial weight, which is set to 2 empirically. The spatial distance between a patch and a region $d(sp_i,rp_j)$ is normalized into the [0,1] range using a simple division by the maximum dimension of the image. The function $d(sc_i,rc_j)$ returns the color contrast of a patch to the region compared. Since the region abstraction method discovers the dominant color of the superpixels in a region, each superpixel is compared to all the regions except the region it belongs. Finally, the contrast cue for each superpixel is normalized to the [0,1] range using min-max normalization. When M is equal to N, the color contrast estimation becomes standard patch level contrast estimation as in [5].

3.4 Color Distribution Estimation

The color distribution of a patch is determined by computing the spatial variance of its color [10]. First, the weighted mean position of a superpixel's color sc_i is defined as,

$$msp_i = \frac{1}{M}\sum_{j\neq i}\exp\left(-d(sc_i, rc_j)\cdot\beta_2\right)\cdot rp_i \tag{6}$$

where the exponential function weights the position of each region based on its color similarity to s_i. Similar to color contrast estimation, color distribution is also computed by considering all the regions except the region it belongs. The color distribution of the superpixel s_i is defined as,

$$cdis(s_i) = \sum_{j\neq i}\frac{|r_j|}{N}\cdot d(msp_i, rp_j)\cdot\exp\left(-d(sc_i, rc_j)\cdot\beta_2\right) \tag{7}$$

The number of superpixels in a region $|r_j|$ is used to emphasize the color distribution of s_i when it is compared to bigger regions, which is normalized by the total number of superpixels N as in equation (5), because, the higher similarity to bigger regions indicates a wider distribution of a superpixel color sc_i. The function $d(msp_i,rp_j)$ returns the spatial distance between a superpixel's mean color position and a region position. This distance is normalized into the [0,1] range using a simple division by the maximum dimension of the image. The exponential function returns the color similarity between superpixel s_i and a region r_j. The scaling parameter β_2 in both equations (6) and (7) is empirically set to 8. The color distribution cue for each superpixel is then normalized to the [0,1] range using min-max normalization.

The higher color distribution indicates that the color component is widely spread over the image, which is less likely to be the color of the salient object. So, the color distribution cue for saliency is defined as:

$$dis(s_i) = 1 - cdis(s_i) \tag{8}$$

This method for color distribution estimation is similar to [10]. However, the major difference between [10] and the color distribution estimation presented here is that our method uses both patch level and region level image abstractions for color distribution estimation, whereas [10] only determines it with image patches.

3.5 Spatial Saliency Assignment and Refinement

The spatial saliency of a superpixel is computed by integrating the color contrast and color distribution cues. The color contrast and color distribution cues are considered to be independent [10] which are combined together using Multiplication fusion as in [5,10], defined as:

$$ssal(s_i) = con(s_i)\cdot dis(s_i) \tag{9}$$

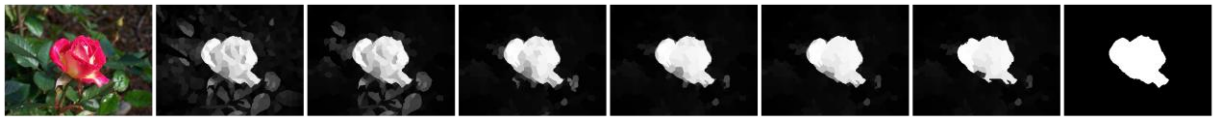
Finally, the spatial saliency value for each superpixel $ssal(s_i)$ is normalized to the [0,1] range using min-max normalization. After fusing the individual saliency cues, there may be some noises in the saliency map due to small scale textured patterns in the background. Yan et al. [26] proposed a

hierarchical model for suppressing the saliency of small-scale high-contrast patterns in the background. Their model uses multilayered local contrast estimation, which is computationally expensive. Simply averaging the surrounding superpixels' saliencies [10] cannot preserve saliency near object boundaries which will end up blurred object boundaries in saliency maps. This as a salient object will be comprised of a group of spatially connected salient superpixels. Fu et al. [5] refined the pixel level saliency map produced by superpixel based saliency detection approach. Their refinement approach first over-segments the image using mean-shift segmentation algorithm, and then computes a region's saliency as the average of the saliencies of pixels in that region. Yang et al. [52] proposed a graph regularization based saliency refinement called *smoothness prior*, to encourage adjacent superpixels to have similar saliencies. Even though these two aforementioned refinement approaches produce saliency maps with uniformly highlighted regions, the saliencies of background noises still remain the same even after refinement. Moreover, these post-segmentation based and graph-based refinement methods are computationally expensive. In this paper, a simple, effective and computationally feasible saliency refinement mechanism is presented based on the two observations regarding saliency of a superpixel. A superpixel surrounded by highly salient superpixels belongs to the salient object. Also, a superpixel surrounded by low salient superpixels belongs to the background.

The objective of saliency refinement is threefold. Firstly, to encourage adjacent superpixels to have similar saliencies. Secondly, to emphasize saliencies of the superpixels surrounded by neighbors with high saliencies. Lastly, to suppress saliencies of superpixels surrounded by neighbors with low saliencies. Accordingly, the refined saliency of a superpixel is defined as:

$$sal(s_i) = \begin{cases} \max\limits_{s_j \in ns} sal(s_j), & \text{if } \left(\frac{1}{|ns|} \sum\limits_{s_j \in ns} sal(s_j) \right) \geq 1-\mu \\ \min\limits_{s_j \in ns} sal(s_j), & \text{if } \left(\frac{1}{|ns|} \sum\limits_{s_j \in ns} sal(s_j) \right) \leq \mu \\ sal(s_i), & \text{otherwise} \end{cases} \tag{10}$$

The above equation first finds an average saliency of the adjacent superpixels of s_i. If the average neighborhood saliency exceeds 1-μ, then the maximum among neighborhood superpixels' saliencies is set as the saliency of center superpixel s_i. The parameter *ns* denote the set of neighborhood superpixels, where $|ns|$ is the number of neighborhood superpixels. If the average neighborhood saliency is lesser than μ, then the minimum among the neighborhood superpixels' saliencies is set as the saliency of center superpixel s_i. The parameters μ ranges from 0 to 0.5, which is set to 0.2 empirically. The average neighborhood saliency between 1-μ and μ denotes that superpixel s_i is adjacent to the boundary of salient object where the saliency of s_i remains the same after refinement. The influence of the parameter μ on saliency refinement is depicted in figure 5.



(a) Source image | (b) Saliency refinement with μ = 0.0 | (c) Saliency refinement with μ = 0.1 | (d) Saliency refinement with μ = 0.2 | (e) Saliency refinement with μ = 0.3 | (f) Saliency refinement with μ = 0.4 | (g) Saliency refinement with μ = 0.5 | (h) Ground Truth

Fig. 5. Illustration of the saliency refinement approach with different values for the refinement parameter μ.

A strong refinement can be achieved by setting the parameter value near 0.5, which also introduces some defects in the saliency maps. The saliency refinement method can be used in any pixel/patch level saliency detection methodology. This is due to the adaptive saliency refinement method which highlights

the salient object uniformly, and detects the background efficiently by removing the noises from the background (shown in figure 2).

3.6 Center Prior Integration

Apart from color contrast and color distribution cues, another widely used prior called *center prior* [5,8,11,12,14,26] is incorporated into the spatial saliency detection approach. Since salient objects are placed near the image center most of the time, the center prior gives more weight to the regions that are nearer the image center than the regions near the image boundaries. A center prior map is generated using a Gaussian function based on a superpixel's distance from the image center. The center prior weight for a superpixel is defined as:

$$cen(s_i) = \exp\left(-\frac{d(sp_i, c)}{2\sigma_2^2}\right) \tag{11}$$

The function $d(sp_i,c)$ returns the Euclidean distance between a superpixel and the image center c. The parameter σ_2 is set to $\min(W,H)/2.5$ as in [11], where W and H are the width and height of the image respectively. The center prior is integrated into the saliency detection framework as a post-processing step. Since the center prior is treated as a filtering mechanism, the integration of refined saliency and center prior weight is achieved using a simple multiplication [5], defined as:

$$ssal(s_i) = ssal(s_i) \cdot cen(s_i) \tag{12}$$

The spatial saliency $ssal(s_i)$ is normalized into the [0,1] range using min-max normalization. Pixel-level gray scale spatial saliency map can be produced by up-sampling patch level saliency values which are normalized into a range [0,255]. The center prior integration further suppresses background noises near image boundaries, and emphasizes saliencies of the patches that are near the image center (depicted in Figure 2).

The **P**atch-**R**egion-based spatial salient region detection is denoted by **PR**, where the faster variants of the spatial saliency detection are represented by **PFR**, **FPR** and **FPFR**. The spatial salient region detection approach **PR** uses superpixels as **P**atches and **R**egion from graph-based segmentation. The variant method **PFR** uses superpixel based **P**atch segmentation and uniform sampling based **F**aster **R**egion segmentation, where **FPR** uses uniform sampling based **F**aster **P**atch segmentation and spectral clustering based **R**egion segmentation. The variant **FPFR** uses **F**aster **P**atch segmentation and **F**aster **R**egion segmentation. Saliency maps of the variants of the spatial salient region detection method are depicted in figure 6. The figure shows that the faster variants of the method also robustly detect salient regions in images.

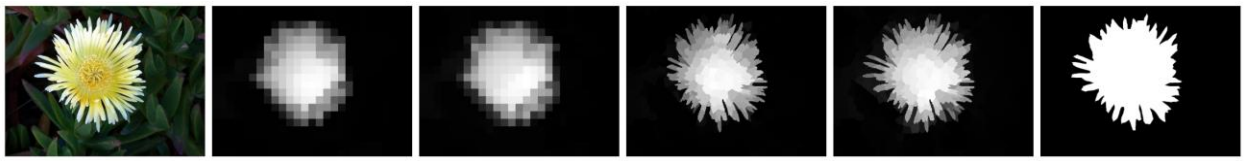
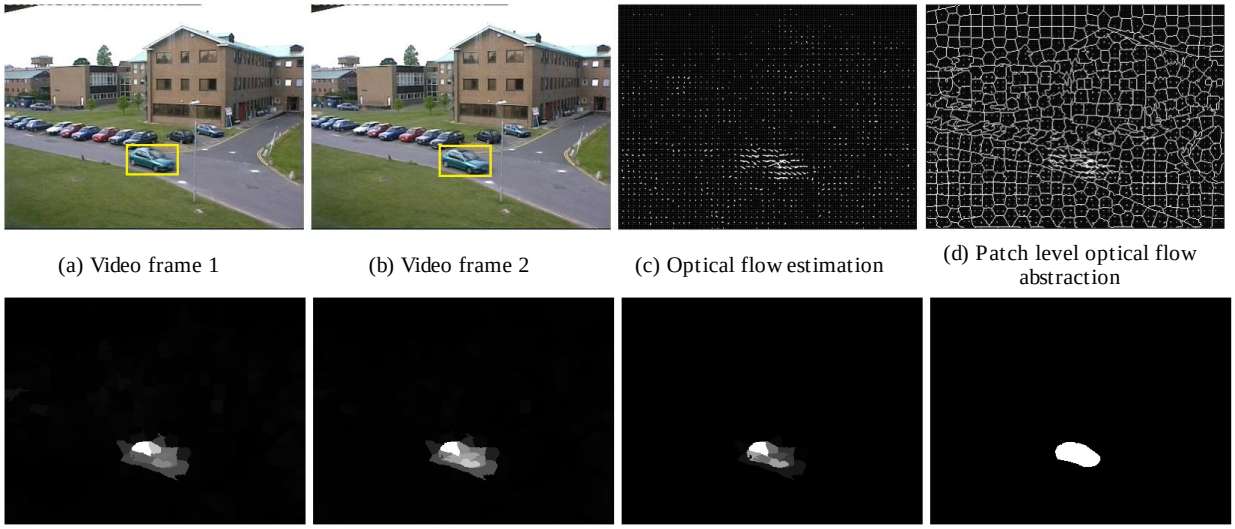


Fig. 6. Saliency maps of the variants of the spatial salient region detection method.

4. Temporal Salient Region Detection

This section proposes a unified approach for temporal salient region detection. The process of temporal saliency detection is formulated as a function of how dissimilar or contrast a patch's motion is from the rest of the patches both locally and globally. The steps involved in temporal salient region detection are depicted in figure 7.



(a) Video frame 1 (b) Video frame 2 (c) Optical flow estimation (d) Patch level optical flow abstraction

(e) Local temporal saliency (f) Global temporal saliency (g) Fused temporal saliency (h) Ground Truth

Fig. 7. Steps of the proposed temporal salient region detection approach. First, optical flow (c) is computed for a video frame (a) using consecutive video frames 1 (a) and 2 (b). The patch level optical flow abstractions (d) are used for estimation of local temporal saliency (e) and global temporal saliency (f). Both local and global temporal saliencies are fused to generate final temporal saliency map (g) of video frame 1 (a). (g) Ground truth saliency map.

4.1 Patch Level Dominant Optical Flow Abstraction

Motion in a video frame is estimated by computing the optical flow from two successive video frames. A computationally efficient global optical flow method proposed in [53] is used for estimating dense flow vectors. The optical flow estimation method proposed in [53] is a modified method of [54] and [55].

Computing temporal saliency with pixel level optical flow is computationally intensive. Also, the estimated optical flow would often contain pixel level noise caused by sudden illumination changes, and video data acquisition errors. These pixel level flow noises in background will degrade the performance of a temporal saliency detector, which can be suppressed by patch level optical flow abstraction. The patch level optical flow abstraction also makes temporal saliency detection robust to dynamic camera and background motions. A patch level histogram based optical flow approximation is proposed in [33], which quantizes optical flow vectors into 4 orientations. The cumulative magnitude of each orientation in a patch is represented as a histogram of optical flow. There are two problems with this method: 1) Patch level motion contrast estimation with histogram based approximation is computationally expensive; 2) A lower number of quantized optical flow orientations is less discriminative which cannot differentiate between different flow directions. Determining the number of flow orientations is also a crucial problem in histogram based temporal saliency estimation, because a smaller number of flow orientations suffers from less discriminability where higher number of flow orientations is computationally expensive. So, an alternative patch level optical flow abstraction approach is proposed here.

Since most of the pixels in a patch have the same optical flow orientations, instead of representing the optical flow of a patch using a statistical representation such as histogram, it can be represented by a dominant flow orientation and a cumulative magnitude of that orientation. For each pixel in a video frame, optical flow detector returns a vector of optical flow velocities (v_x,v_y) in x and y directions

correspondingly. Optical flow magnitude fm_j and flow orientation fo_j are calculated using the corresponding flow velocities (v_x,v_y) of each pixel p_j. Each flow orientation fo_j is quantized into 8 orientations fo = {0°, 45°, 90°, 135°, 180°, 215°, 270°, 315°}, which is comparatively higher discriminative than 4 orientations. Since each patch is represented only by a dominant orientation and a cumulative flow magnitude, even a very higher number of flow orientations does not affect the computational time of the proposed temporal saliency estimation. The dominant flow orientation of a superpixel s_i is calculated as:

$$dfo_i = \arg\max_{dfo_i \in fo} \sum_{j=1}^{|s_i|} fm_j \tag{13}$$

$|s_i|$ is the number of pixels in a superpixel s_i. The dominant optical flow orientation dfo_i of a superpixel is an orientation that has the maximum cumulative flow magnitude in a superpixel. The fo is the set of 8 predefined quantized orientations. The cumulative magnitude of a superpixel is the sum of magnitudes of the pixels with the dominant optical flow orientation, which is computed as:

$$dfm_i = \sum_{\forall fo_j = dof_i} fm_j \tag{14}$$

The cumulative magnitude of dominant flow dfm_i is normalized into the [0,1] range using min-max normalization. The optical flow abstraction for a 5x5 patch is illustrated in figure 8. The dominant optical flow orientation dof_i and dominant optical flow magnitude dfm_i for each superpixel s_i are further used for estimating local and global temporal saliencies.

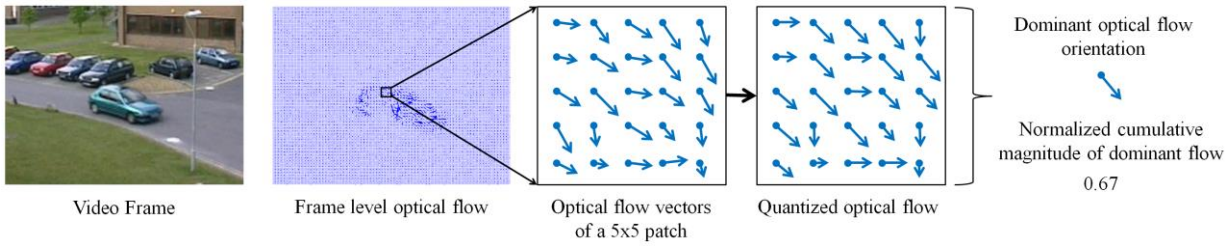


Fig. 8. Illustration of patch level optical flow abstraction.

4.2 Local Temporal Saliency Estimation

The local temporal saliency of a video frame is measured as patch level local motion contrast. The local temporal saliency of a superpixel s_i is the average motion contrast between the center superpixel and its ns adjacent neighborhood superpixels, which is weighted by the dominant optical flow magnitude, defined as:

$$lts(s_i) = dfm_i \cdot \frac{1}{|ns|} \sum_{j=1}^{|ns|} d(dof_i, dof_j) \tag{15}$$

The function $d(dof_i,dof_j)$ returns the *shortest angular difference* between dominant optical flow orientations of two superpixels s_i and s_j. This angular difference will be range in [0°,180°] which is normalized to the [0,1] range using a simple division by 180°, which is the highest possible shortest angular difference between two quantized optical flow orientations. The dominant optical flow magnitude dfm_i is used to favor patches with high motion.

A problem with this traditional center-surround difference mechanism for saliency detection is that it only highlights the borders of a salient region. When the scale of a salient region is high, this mechanism would need a multi-scale analysis of local motion contrast. To avoid the computational overhead experienced in multi-scale saliency estimation, a multi-level center-surround difference is proposed here. The proposed local temporal saliency detection approach bridges both local and global contexts for

saliency estimation. That is, instead of only comparing the center superpixels with the first level immediate neighboring superpixels, it can also be compared with next level neighboring superpixels (i.e. neighbors of neighbors).

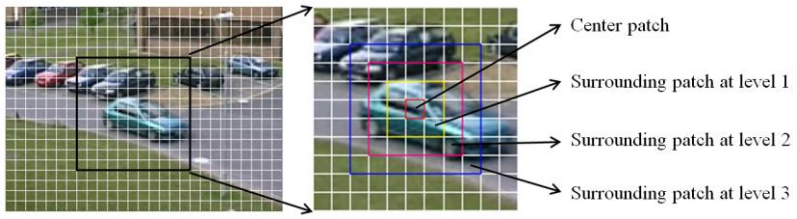


Fig. 9. Illustration of center and multi-level surrounding patches for local temporal saliency estimation.

Figure 9 shows the center and multi-level surrounding patches for local temporal saliency estimation. The idea of incorporating multi-level neighbors for local saliency detection has been already considered by [13] for spatial saliency detection. The authors have considered only two levels of neighbors for local saliency estimation. Still, the authors used multiscale analysis for more robust spatial saliency detection. They used spatial distance between patches from different surrounding levels to weight the center-surround differences. However, the proposed method weights the local temporal saliency using the number of levels rather than the spatial distances. Since the spatial distances between center and a surrounding patch is almost same for all the surrounding patches of the same level, the exact spatial distances between patches is not always required. This assumption strictly holds in the context of motion contrast estimation with uniformly sampled squared patches (figure 9). In addition, the superpixel based patch segmentation also maintains consistent regularity among superpixels. The multi-level center-surround difference-based local temporal saliency for a superpixel is defined as:

$$lts(s_i) = dfm_i \cdot \sum_{L} \frac{1}{L} \cdot \frac{1}{|ns|} \sum_{j=1}^{|ns|} d(dof_i, dof_j) \tag{16}$$

For each level L, the term ns denotes the surrounding superpixels of the corresponding level. Now the spatial weighting between center and surrounding superpixels is achieved by the number of levels L. The local temporal saliency with different surrounding levels is depicted in figure 10.

(a) Video frame (b) Local temporal saliency with L = 1 (c) Local temporal saliency with L = 2 (d) Local temporal saliency with L = 3 (e) Local temporal saliency with L = 4 (f) Local temporal saliency with L = 5 (g) Ground Truth

Fig. 10. Local temporal saliency estimation with different surrounding levels L.

When L=1, the local saliency works as the traditional center-surround difference mechanism. The figure shows that the local temporal saliency with L=1 detects only the border of salient objects and fails to detect its inner parts. This also emphasizes some background patches as salient (figure 10 (b)). With a large L, the local temporal saliency achieves spatially weighted global temporal saliency. The parameter L determines the tradeoff between speed and accuracy in the local temporal saliency estimation. The influence of this parameter in temporal saliency detection is experimentally assessed in the section 5.3. To balance both speed, and local and global scenarios of the temporal saliency, the parameter L is empirically set to 3. The local temporal saliency lts is normalized to the [0,1] range using min-max normalization. Computing global temporal saliency as the spatially weighted patch level global motion contrast needs a relatively high computation i.e. $O(N^2)$ iterations. So, an alternative solution based on patch motion statistics is proposed for global temporal saliency estimation. The proposed multi-level center-surround difference mechanism can also be extended to spatial saliency detection for detecting salient objects and

human eye fixations in an image.

4.3 Global Temporal Saliency Estimation

The motion of a patch that is locally similar to surrounding patches' motions might still be globally salient comparing to the motion of other patches in an image. Even though the proposed multi-level center-surround difference for temporal saliency can achieve balanced local and global temporal saliencies, a purely global temporal saliency is proposed here. The global temporal saliency of a superpixel is the self-information contained in its motion which is weighted by its dominant flow magnitude dfm_i. The global temporal saliency is defined as:

$$gts(s_i) = dfm_i \cdot \left(-\log P(dof_i)\right) \tag{17}$$

Similar to equation (15), the dominant optical flow magnitude dfm_i is used to favor patches with high motion. The term $-\log P(dof_i)$ is the self-information of the motion of the superpixel, where $P(dof_i)$ is the probability of the dominant flow orientation dof_i, which is computed as:

$$P(dof_i) = \frac{ndof_i}{N} \tag{18}$$

$ndof_i$ is the number of superpixels with dominant orientation dof_i in a video frame. The global temporal saliency $gts(s_i)$ is normalized to the [0,1] range using min-max normalization.

4.4 Temporal Saliency Assignment

Local and global temporal saliencies are independent saliency cues which are integrated to compute the final temporal saliency map. Among different integration schemes such as *Addition Fusion*, *Multiplication Fusion*, *Max Fusion*, and *Min Fusion* (i.e. +, ×, *max* and *min*), the integration scheme used for fusing local and global temporal saliency is empirically set to Multiplication fusion. The local and global temporal saliencies are fused using a simple multiplication defined as:

$$tsal(s_i) = lts(s_i) \cdot gts(s_i) \tag{19}$$

This takes advantage of both local and global temporal saliencies of a video frame. The temporal saliency values are normalized into the [0,1] range using min-max normalization, which can be further normalized into range [0,255] for producing gray scale temporal saliency map. The temporal saliency does not need to be refined as the spatial saliency. The rationale for avoiding temporal saliency refinement is that the observations regarding spatial saliency of objects in images (mentioned in section 3.5) might not always hold for salient objects in videos. Since the scale of salient objects or motion will be small in some visual scenarios such as surveillance video, the refinement mechanism will suppress the saliency of small scale object motions in such cases. The experimental comparison of different fusion schemes for temporal saliency fusion is presented in section 5.3.

4.5 Spatiotemporal Saliency Assignment

Figure 11 depicts the steps involved in spatiotemporal salient region estimation. The spatiotemporal saliency of a superpixel is computed by integrating its spatial and temporal saliencies. The integration scheme used for fusion of spatial and temporal saliencies is empirically set to Max fusion. The spatiotemporal saliency of a superpixel is defined as:

$$stsal(s_i) = \max\left(ssal(s_i), tsal(s_i)\right) \tag{20}$$

which is the maximum among spatial and temporal saliencies of a superpixel. This fusion scheme takes advantage of both spatial and temporal saliencies estimated for a video frame. So, even if the visual

(a) Video frame (b) Spatial saliency (c) Temporal saliency (d) Spatiotemporal saliency (e) Ground Truth

Fig. 11. Phases of computing spatiotemporal saliency detection. Spatial saliency (b) and temporal saliency (c) of a video frame (a) are fused together to obtain spatiotemporal saliency (d). (e) ground truth.

scene becomes static, this fusion scheme will highlight spatially salient regions in the spatiotemporal saliency map. Also, the temporally salient regions are emphasized in spatiotemporal saliency maps even if the spatial salient region detection approach fails to detect the salient object in a dynamic scene. The experimental evaluation of different fusion schemes for spatiotemporal saliency fusion is presented in section 5.3. Finally, the spatiotemporal saliency values are normalized into the [0,1] range which can be further normalized into the [0,255] range to produce a gray scale spatiotemporal saliency map. The spatial, temporal and spatiotemporal saliency maps are finally resized into the input image *I*'s original resolution.

5. Experimental Results

This section presents evaluation of the proposed temporal and spatiotemporal salient region detection methodologies comparing to the several state-of-the-art spatiotemporal saliency detection methods on standard video datasets.

For the comparative evaluation of our **P**atch-**R**egion-based spatial salient region detection (PR) on a salient object detection image dataset MSRA-1000 [38], the readers are recommended to refer our work in [27]. Here, the robustness of the spatial salient region detection method to different numbers of patches and different numbers of regions, and the robustness of the faster variants were presented, as also were the performance of the individual saliency cues, the effect of spatial saliency refinement parameter and time comparison with different methods were also presented in our paper [27].

Performance of the proposed **L**ocal-**G**lobal motion rarity based **T**emporal saliency approach **LGT**, and **S**patio-**T**emporal **S**aliency detection approach **STS** are extensively evaluated with the two most widely used surveillance video datasets and two challenging salient object detection video datasets. The proposed temporal and spatiotemporal saliency detection approaches LGT and STS are compared with the state-of-the-art methods, PD [56], PCAWSal [57], FastSUN [58], Simpsal [59], SeR [29], GBVS [17], PQFT [3] and SEG [60]. Methods such as PD and PCAWSal are temporal saliency detection approaches where the rest of the comparative methods are spatiotemporal saliency detection approaches. The Simpsal [59] method implements the Itti's model [7] for spatiotemporal saliency detection in videos. The optical flow method [53] that is employed for the proposed method is used for the optical flow estimation for SEG [60]. The temporal salient region detection method LGT uses superpixel based patch segmentation, where the faster variant of the temporal saliency detection FLGT use uniform sampling based patch segmentation. Spatiotemporal salient region detection approach STS uses LGT for temporal saliency detection and PR for spatial saliency detection. FSTS is the faster variant of the proposed spatiotemporal salient region detection that uses FLGT and FPFR for temporal and spatial saliency detections correspondingly.

The performance of a salient region detection approach is usually evaluated by measuring *precision* and *recall* rate. Precision is the percentage of correctly assigned salient pixels in a thresholded binary saliency map, where recall is the percentage of correctly assigned salient pixels in a binary saliency map in relation to the number of salient pixels in the ground truth map. A binary saliency map can be obtained by thresholding a saliency map using a threshold ranging from 0 to 255. Firstly, different precision and recall value pairs are computed by thresholding a saliency map using a number of fixed thresholds in the [0,1,…,255] range. These precision and recall values are then averaged over all the images/video frames for corresponding thresholds, which results in a precision-recall curve.

Since the scale of the salient moving objects that are presented in some videos is very small when compared to salient objects in images, and distribution of saliency will not be same for all the images/video frames, precision recall rates using fixed thresholding is not a good choice for evaluation of salient object detection in videos. By following previous approaches [31, 33], the performance of the proposed spatiotemporal salient region detection is assessed by precision-recall analysis using image-

dependent adaptive thresholding method. The adaptive threshold T_a is defined as twice the mean saliency of the saliency map:

$$T_a = \frac{2}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} S(x, y) \tag{21}$$

where W and H are the width and height of the saliency map correspondingly. $S(x,y)$ is the saliency of a pixel at position (x,y). Besides precision and recall, the weighted harmonic mean or F-Measure for each saliency map is computed as:

$$F_\beta = \frac{(1+\beta^2) \cdot Prcesion \cdot Recall}{\beta^2 \cdot Prcesion + Recall} \tag{22}$$

Similar to [5,9,10,38], β2 is set to 0.3 to give more weight to precision than recall. The results of comparative evaluation of different methods are visualized using bar graphs with averages of precision, recall and f-measure along with their corresponding standard deviations. Since the values of evaluation measures should be in a [0,1] range, the parts of standard deviation error bars which fall outside the [0,1] range are discarded. The rest of the results are visualized better using tables where the slight differences in the results of individual phases of the proposed approach, and of different parameter settings can be easily noticed. Visual and time comparisons of different methods for the evaluation datasets are also presented as well.

5.1 Performance Evaluation in Surveillance Video Datasets

The initial experimental evaluation is performed on two most widely used surveillance video datasets PETS2001 [61] and OTCBVS [62]. These two datasets are regarded as the difficult datasets for salient object detection since they contain dynamic illumination changes. Both of these datasets comprise outdoor surveillance video sequences containing multiple moving objects and dynamic illuminations. PETS2001 dataset (3000+ video frames) contains image sequences with gradual illumination changes, and pedestrians, vehicles moving in different directions. The OTCBVS dataset (1000+ video frames) contains image sequences with shadows of moving clouds across buildings and pedestrians moving in different directions.

For ease of ground-truth annotation and uniformity, the image sequences in both of these datasets are resized with maximum image dimension being 400 pixels. As proposed in [31], 15 video frames were randomly chosen from each surveillance video dataset (i.e. PETS2001 and OTCBVS datasets) for the experiments. For validating the performance of salient object detection, ground truth indicating the salient moving objects in the total of 30 video frames were annotated manually. Unlike the bounding box based ground truth annotation used in [31, 33], the pixel level ground truth is annotated for each video frame, which is further used for validation. Averages of precision, recall and F-measure are computed by using adaptive thresholding methods using equation (21).

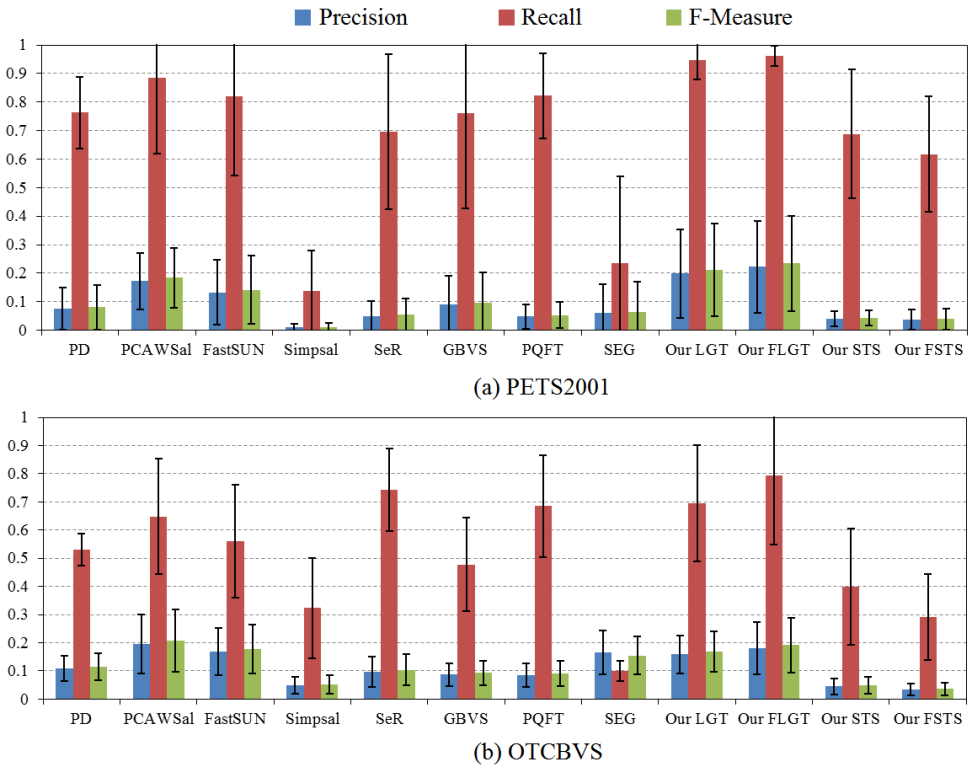


Fig. 12. Performance evaluation on (a) PETS2001 and (b) OTCBVS datasets.

Figure 12 depicts the precision, recall and f-measure of different methods on PETS2001 and OTCBVS datasets. The results show that the proposed faster temporal salient region detection method FLGT performs fairly under all the evaluation measures compared to the other methods. The FLGT outperforms other methods in PETS2001 dataset, where PCAWSal achieves better precision and F-measure than the proposed methods in the OTCBVS dataset. However, PCAWSal fails to detect all the salient objects in a visual scene and thus results in comparatively lower recall rate for OTCBVS dataset. Unlike the performances of faster variants of PR, surprisingly the faster variant of the temporal salient region detection approach FLGT outperforms the superpixel segmentation based temporal saliency detection approach LGT. The reason is that, in the case of surveillance videos, the uniform sampling based patch segmentation abstracts the patch level motion abstraction better than superpixel based patch segmentation. Superpixel based motion abstraction sometimes abstracts small scale noisy flow vectors which are removed in the uniform patch sampling based flow abstraction.

Since the proposed approach STS uses Max fusion that takes advantage of both spatial and temporal saliencies, it detects both moving objects and some salient static objects, which results in very poor precision and f-measure rate in both datasets. Unlike the case of faster variant of the temporal saliency detection approach, STS performs slightly better than FSTS. Since FSTS uses FPFR for spatial saliency detection, the performance of the FSTS is comparatively lower than that of STS. The proposed temporal saliency detection methods LGT and FLGT perform better than the proposed spatiotemporal saliency detection approaches STS and FSTS. Since STS comprises components of both spatial and temporal saliencies, some salient non-moving objects are also marked as salient in the saliency maps of STS. However, in many real world applications, spatiotemporal saliency is always desired rather than temporal saliency alone. It is concluded that the proposed temporal saliency detection approach itself is sufficient for moving object detection in videos with static camera especially the surveillance videos.

Dataset\ Method	PETS2001			OTCBVS		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Spatio Temporal Saliency (STS)	0.0400	0.6881	0.0433	0.0451	0.3996	0.0486
Spatial Saliency PR	0.0175	0.2772	0.0189	0.0259	0.2590	0.0280
Fused Temporal Saliency (LGT)	0.1992	0.9466	0.2109	0.158	0.6957	0.1682
Global Temporal Saliency	0.1202	0.9551	0.1280	0.1390	0.8374	0.1490
Local Temporal Saliency	0.1146	0.9474	0.1223	0.1328	0.8182	0.1424

Table. 1. Evaluation of individual phases of the proposed spatiotemporal saliency detection approach on PETS2001 and OTCBVS datasets.

Table 1 shows the performance of individual phases of the spatiotemporal saliency detection approach STS in PETS2001 and OTCBVS datasets. The results show that the temporal saliency LGT performs better than the individual local and global temporal saliencies. It can be seen that the global temporal saliency method presents a competitive performance against LGT. Among the individual temporal saliency cues, the global temporal saliency outperforms local temporal saliency in both datasets. The spatial saliency PR works well in terms of recall rate in PETS2001 dataset than in OTCBVS dataset, because the salient objects in PETS2001 dataset pose high contrast to the background, which makes them distinct and easily detectable.

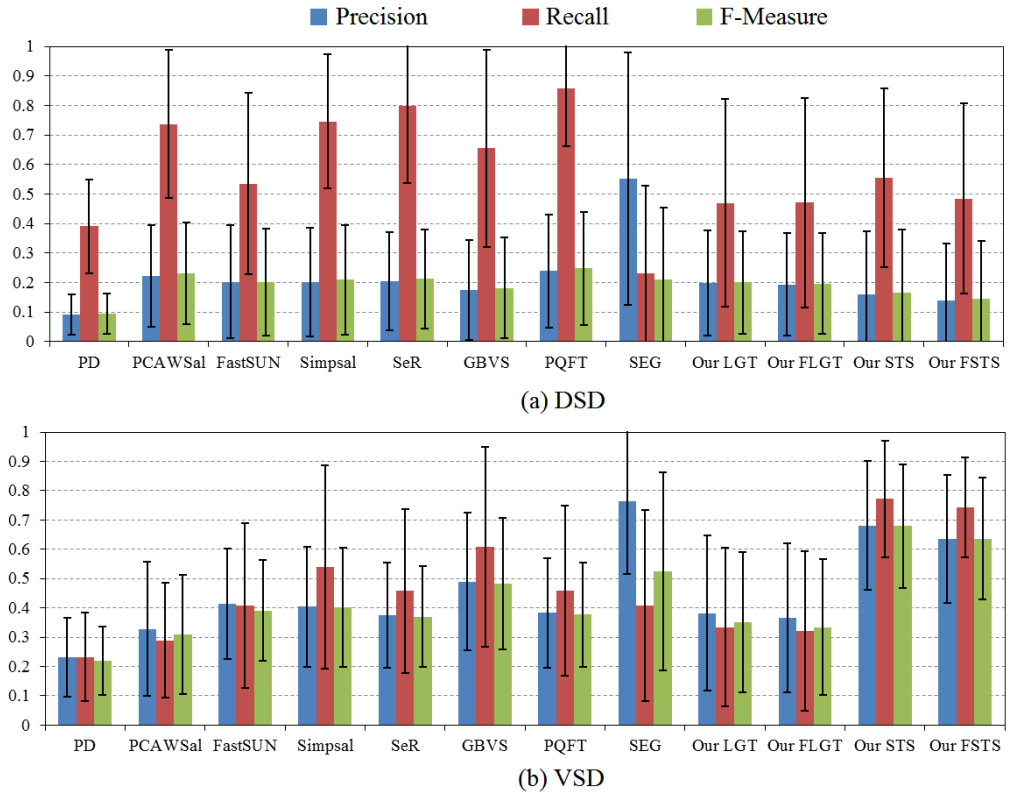


Fig. 13. Performance evaluation on (a) DSD and (b) VSD datasets.

5.2 Performance Evaluation in Challenging Salient Object Detection Video Datasets

The proposed approaches LGT and STS are further extensively evaluated with two challenging publicly available salient object detection video datasets - complex Dynamic Scenes Dataset (DSD) [30] and saliency-based Video Segmentation Dataset (VSD) [63]. The DSD dataset contain 18 video sequences (1900+ grayscale video frames) where each video has one or more moving objects. Some of

the video sequences contain surveillance footage of pedestrians or a crowded highway whereas the rest of the sequences comprise highly dynamic background (such as flock of birds, smoke, and water), high camera motion or both. The VSD dataset contain 10 video sequences (900+ video frames) where mostly one moving object is present in each video. Most of the video sequences in the VSD dataset contain smooth background with camera motion where rest of the videos present static visual scenes with highly cluttered and dynamic background. Both DSD and VSD datasets provide pixel accurate manual segmentation of the salient objects, which are served as ground truth for validation.

Figure 13 shows the performance of different methods in the two challenging datasets. Since the number of frames in individual videos of a dataset (i.e. DSD and VSD datasets) vastly differ from each other, the methods which perform well on a larger video will get benefited when the evaluation measures are calculated using average on the entire dataset. In order to avoid such evaluation bias, the precision, recall and f-measure values for the entire dataset are measured as the average of corresponding measures calculated for the individual videos. But the standard deviations of evaluation measures are calculated using average on the entire dataset. For the DSD dataset, the PQFT method presents best results in terms of recall and f-measure rates. Despite SEG boasting the highest precision for the DSD dataset, SEG presents very lower recall since it fails to detect all the salient objects that are presented in the test videos. Among the proposed methods, the temporal saliency detection approach LGT presents better results than the spatiotemporal saliency detection approach STS. It should be noted that the LGT present competitive precision and f-measure when comparing to the other methods. For the VSD dataset, the proposed approach STS outperforms other methods in terms of recall and f-measure. Even though SEG poses highest precision value, it shows comparatively lower recall value. It can be seen that the faster variant FSTS method also outperforms other methods in the VSD dataset. In both datasets, the proposed methods LGT and STS perform slightly better than their corresponding faster variants.

Dataset\ Method	DSD			VSD		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Spatio Temporal Saliency (STS)	0.1595	0.5555	0.1652	0.6810	0.7715	0.6791
Spatial Saliency PR	0.1742	0.4331	0.1786	0.7291	0.7710	0.7238
Fused Temporal Saliency (LGT)	0.1987	0.4696	0.2001	0.3823	0.3344	0.3514
Global Temporal Saliency	0.2039	0.5082	0.1957	0.4234	0.3726	0.3886
Local Temporal Saliency	0.1351	0.5774	0.1411	0.2813	0.3877	0.2747

Table. 2. Evaluation of individual phases of the proposed spatiotemporal saliency detection approach on DSD and VSD datasets.

Table 2 depicts the performance of individual steps in the proposed spatiotemporal saliency detection approach STS in DSD and VSD datasets. Comparing to the local temporal saliency cue, global temporal saliency present slightly better results in both challenging datasets. The fused temporal saliency LGT achieves better performance than the individual temporal saliency cues in the DSD dataset. Surprisingly, the global temporal saliency performs better than the LGT in the VSD dataset. However, both local and global temporal saliencies should be considered together for temporal saliency estimation. The spatial salient region detection PR does not work effectively in the DSD dataset as compared to its potential performance in the VSD dataset. The main reason for this degraded performance is that the DSD dataset provides only grayscale video frames on which the color feature based spatial saliency approach PR cannot perform well. The STS displays a better recall rate than the spatial and temporal saliencies in both datasets while also maintaining fair precision and f-measure rates. Thus the fusion of spatial and temporal saliencies increases the number of salient objects successfully detected in a visual scene.

Figure 14 shows the performance of the proposed approaches and comparative methods for the individual videos in the DSD dataset. Some of the precision, recall and f-measure values denoted in the bar graphs as 0 are not necessarily equal to zero, but are less than 10^{-2}. Almost all the methods perform well on videos such as bottle, cyclist, hockey, jump, peds, rain and traffic. Since the salient regions appear

in the videos such as landing, surf, surfers, and zodiac are single small scale objects, most of the methods present very high recall rates, but only poor precision rates. Among the two best performing methods of the DSD dataset, SEG achieves higher precision by highlighting only the most confident regions as salient regions where PQFT achieves higher recall by labeling most of the confident regions as salient regions. Thus, SEG and PQFT fail to maintain fair recall and precision correspondingly. Since both high precision and recall values are always desired in almost all the applications, a saliency detection method should always maintain high precision and recall rates.

Similar to the results in the PETS2001 and OTCBVS datasets, the proposed temporal saliency detection methods LGT and FLGT perform better than the spatiotemporal saliency detection approaches STS and FSTS for videos captured with static camera, such as boat, freeway, ocean, peds, rain and traffic. The LGT and FLGT also show better results for some videos with camera motion such as cyclists, hockey and skiing. For the rest of the videos, the spatiotemporal saliency detection approaches STS and FSTS outperform LGT and FLGT. For video sequences such as freeway, ocean, peds, rain, and surf, faster variants FLGT and FSTS present better results than LGT and STS correspondingly. It should be noted that the STS method outperforms other comparative methods for videos such as chopper, flock and jump.

Figure 15 depicts the results of the proposed approaches and other methods for the videos in the VSD dataset. Almost all the videos present distinctive salient objects except the videos bird 1 and bird 2. Since the aforementioned two videos comprise salient object with highly cluttered background, almost all the other methods give poor performance. However, the proposed approach STS outperforms other methods and presents fair precision and recall rates for these two videos. Most of the methods present appreciable performance for the rest of the videos in VSD dataset. The proposed approach STS outperform all the other methods for all the videos in VSD dataset except for the videos such as skiing 1 and airplane. The LGT approach outperforms all the comparative methods for the video horses. The proposed methods LGT and STS perform better than their corresponding faster variants in most of the videos.

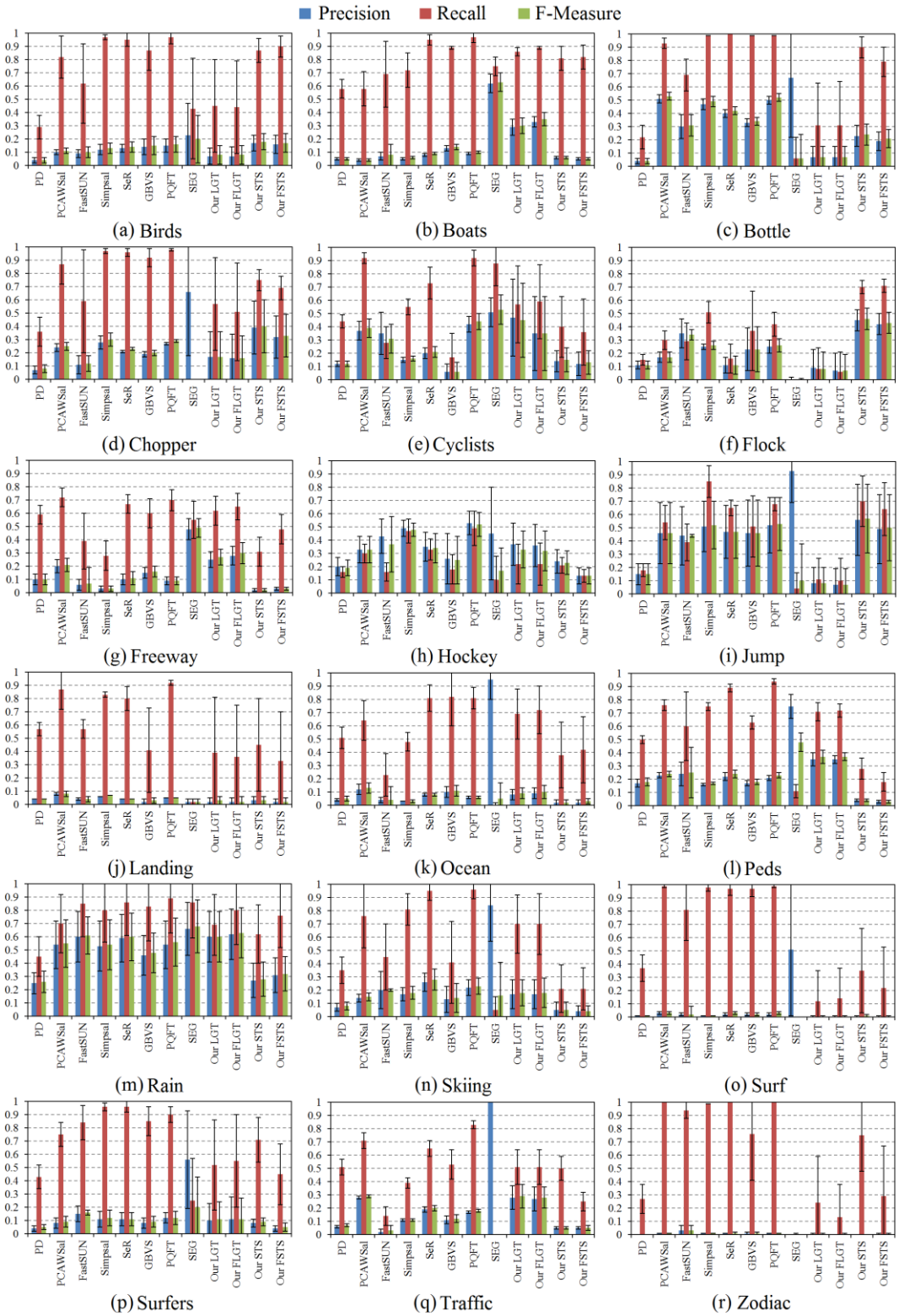


Fig. 14. Performance of different methods for individual videos in DSD dataset. The eighteen graphics (a)-(r) show the comparative results for videos - birds, boats, bottle, chopper, cyclists, flock, freeway, hockey, jump, landing, ocean, peds, rain, skiing, surf, surfers, traffic, zodiac respectively.

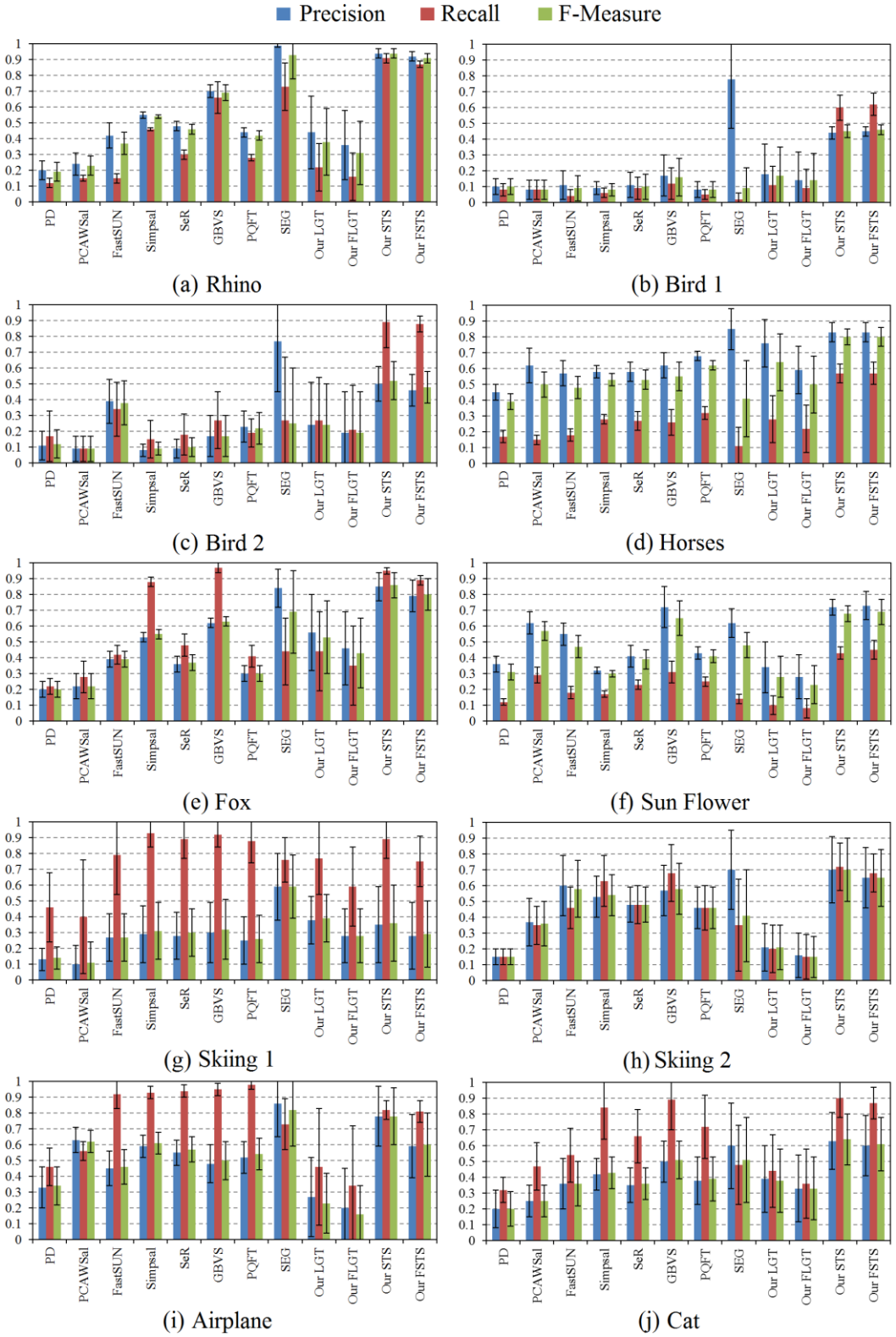


Fig. 15. Performance of different methods for individual videos in VSD dataset. The ten graphics (a)-(j) show comparative results for videos - rhino, bird 1, bird 2, horses, fox, sun flower, skiing 1, skiing 2, airplane, cat respectively.

5.3 Evaluation of Inner Parameters and Fusion Approaches

The influence of the surrounding levels parameter *L* in temporal saliency detection LGT is shown in table 3. As the parameter *L* increases, the performance of LGT is improved in all the datasets. It can be noticed that LGT achieves the best performance in almost test the datasets when *L* = 6. However, the higher *L* results in a slower computation speed. The parameter *L* is chosen to be 3, since there is no abrupt/significant change in the performance of LGT after *L* increases from 3. Since the scale of the salient objects in real world visual scenarios range from lower to higher, the parameter *L* might needed to be set higher in order to achieve effective single scale local temporal saliency estimation. However, the proposed parameter-free global temporal saliency detection can robustly achieve appreciable temporal saliency estimation regardless of the scale of the salient objects and of the performance of local temporal saliency detection.

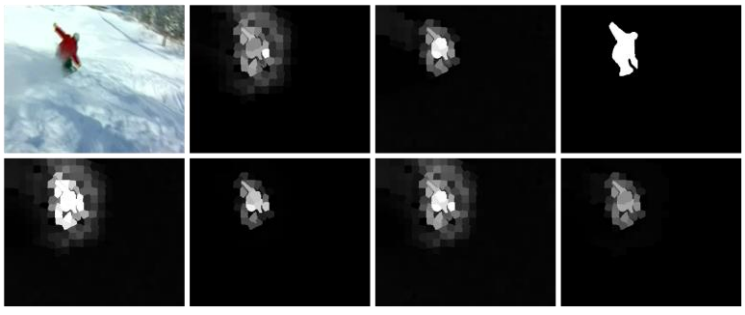
Dataset\ Method	PETS2001			OTCBVS			DSD			VSD		
	P	R	F	P	R	F	P	R	F	P	R	F
LGT - *L* = 1	0.12	0.69	0.12	0.15	0.66	0.16	0.17	0.37	0.17	0.34	0.27	0.30
LGT - *L* = 2	0.17	0.93	0.18	0.15	0.68	0.16	0.19	0.45	0.19	0.36	0.31	0.33
LGT - *L* = 3	0.19	0.94	0.21	0.15	0.69	0.16	0.19	0.46	0.20	0.38	0.33	0.35
LGT - *L* = 4	0.20	0.94	0.22	0.16	0.70	0.17	0.20	0.47	0.20	0.39	0.34	0.36
LGT - *L* = 5	0.21	0.95	0.22	0.16	0.70	0.17	0.20	0.47	0.20	0.39	0.34	0.36
LGT - *L* = 6	0.21	0.95	0.22	0.15	0.69	0.16	0.20	0.47	0.20	0.39	0.34	0.36

Table. 3. Performance of temporal saliency detection LGT with different surrounding levels *L*.

Dataset\ Fusion Method	PETS2001			OTCBVS			DSD			VSD		
	P	R	F	P	R	F	P	R	F	P	R	F
Addition	0.13	0.96	0.14	0.13	0.85	0.14	0.14	0.55	0.15	0.34	0.38	0.32
Multiplication (LGT)	0.19	0.94	0.21	0.15	0.69	0.16	0.19	0.46	0.20	0.38	0.33	0.35
Max	0.12	0.93	0.13	0.13	0.84	0.14	0.15	0.54	0.15	0.33	0.37	0.32
Min	0.13	0.96	0.14	0.13	0.83	0.14	0.16	0.54	0.17	0.36	0.39	0.34

Table. 4. Performance of different fusion approaches for temporal saliency detection LGT.

Table 4 depicts the performance of different fusion approaches for fusing local and global temporal saliencies for temporal saliency estimation. The Multiplication fusion strategy performs better than other fusion schemes in all the datasets. Usually, local and global temporal saliency estimations highlight moving objects as well as some noises. When fusing the two temporal saliency cues using integration approaches such as Addition and Max, final temporal saliency will include both salient objects and noises. The Min fusion approach does not consider the maximum saliency values of local and global temporal saliencies; thus, it excludes both highly salient objects and noises most of the time. Since the Multiplication fusion approach considers only the patches that are highlighted commonly in both temporal cues, it highlights the moving objects and reduces noises in the temporal saliency map.



(a) (b) (c) (d)

Fig. 16. Top (a) video frame, (b) local temporal saliency (c) global temporal saliency (d) ground truth. Bottom, temporal saliency computed using (a) Addition fusion, (b) Multiplication fusion (c) Maximum fusion (d) Minimum fusion.

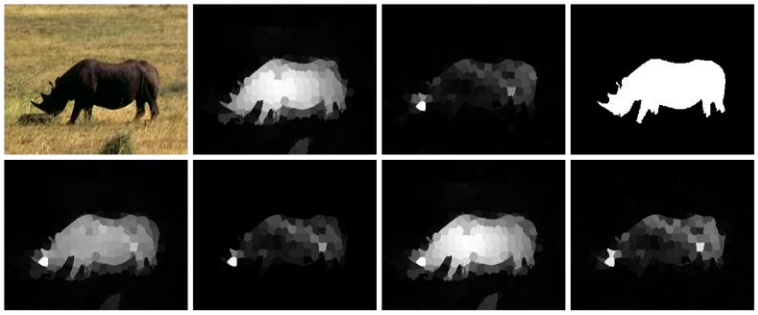
Visual comparison of different fusion approaches for temporal saliency fusion is depicted in figure 16. The figure shows that both Addition fusion and Max fusion perform similarly. But, the temporal map generated by Addition fusion approach looks slightly better than that of Max fusion. Since the Min fusion approach misses out highly salient patches of local and global temproal saliencies, its temproal saliency map look darker than maps of other fusion approaches. Moreover, the Multiplication fusion presents better visual experience than the rest of the fusion approaches.

Dataset\ Fusion Method	PETS2001			OTCBVS			DSD			VSD		
	P	R	F	P	R	F	P	R	F	P	R	F
Addition	0.04	0.76	0.05	0.05	0.48	0.05	0.16	0.57	0.17	0.68	0.77	0.68
Multiplication	0.23	0.87	0.24	0.16	0.62	0.17	0.28	0.45	0.27	0.69	0.45	0.62
Max (STS)	0.04	0.68	0.04	0.04	0.39	0.04	0.15	0.55	0.16	0.68	0.77	0.67
Min	0.14	0.83	0.15	0.14	0.78	0.15	0.19	0.49	0.19	0.51	0.42	0.47

Table. 5. Performance of different fusion approaches for spatiotemporal saliency detection STS.

The performance of different fusion approaches for fusing spatial and temporal saliencies for spatiotemporal saliency estimation is shown in table 5. As shown in table 2, the spatial saliency detection approach PR efficiently detects salient objects most of the time. The temporal saliency detection LGT usually contains partially or entirely detected salient objects. When the visual scene becomes static or the temporal saliency approach fails to detect the salient moving objects sometimes, integration approaches such as Multiplication fusion and Min fusion miss to include salient regions in spatiotemporal saliency maps. Since integration approaches such as Addition and Max consider saliencies of both spatial and temporal saliencies, the salient objects are efficiently detected when using these two fusion techniques. By considering their performances in the challenging datasets DSD and VSD, the integration approach used for fusing spatial and temporal saliencies in STS is empirically set to Max fusion. Even though the performance of Addition fusion is slightly better than that of Max fusion in the challenging datasets, the latter gives a better visual experience than the former. Moreover, the raw continuous saliency maps are highly desired than the binary saliency maps in some application scenarios [9].

The qualitative comparison of different integration approaches for spatiotemporal saliency fusion is depicted in figure 17. Since the temporal saliency map only sometimes partially highlights the salient object, the Multiplication and Min fusion approaches fail to highlight the entire salient object uniformly. This also results Addition fusion approach to present slightly suppressed salient regions. The figure shows that Max fusion approach present better saliency map than other fusion approaches.



(a) (b) (c) (d)

Fig. 17. Top (a) video frame, (b) spatial saliency (c) temporal saliency (d) ground truth. Bottom, spatiotemporal saliency computed using (a) Addition fusion, (b) Multiplication fusion (c) Max fusion (d) Min fusion.

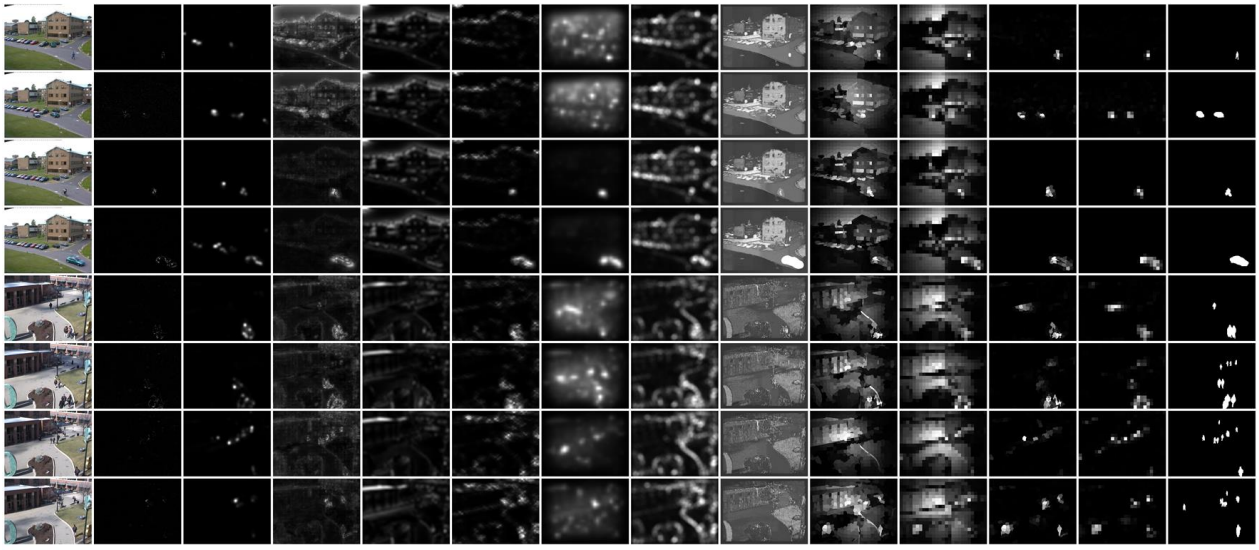


Fig. 18. Visual comparison of the saliency maps of state-of-the-art temporal and spatiotemporal saliency detection methods and the proposed temporal and spatiotemporal saliency detection methods. The top four rows and bottom four rows are the saliency maps computed by different methods for video frames from PETS2001 dataset and OTCBVS dataset correspondingly.

5.4 Visual Comparison

Figure 18 depicts a visual comparison of saliency maps[2] of the state-of-the-art methods and the proposed methods for the PETS2001 and OTCBVS datasets. Since the two datasets comprise scenes taken from static cameras, the proposed temporal saliency detection methods LGT and FLGT, and other temporal saliency detection methods PD and PCAWSal detect salient moving objects with less background noises most of the time. The proposed approaches STS and FSTS, and the rest of the other methods highlight the foreground as well as background regions.

The saliency maps of the proposed methods and different methods for the DSD and VSD datasets are depicted in figure 19. The proposed spatiotemporal saliency detection approaches STS and FSTS present better visual experience than the proposed temporal saliency detection approaches and other approaches. Almost all the comparative methods except GBVS and SEG highlight only object boundaries. In almost all the saliency maps, only a slight visual difference present is between the proposed approaches and their faster variants.

5.5 Time Comparison

In table 6, the performance of different methods is compared using the running time taken to process a video frame with resolution 400×300. The computation time are taken in a computer with Intel i5 2.50 GHz CPU and 4 GB RAM. In the computation time of the proposed STS, temporal saliency detection LGT takes 6.43s (69%) and spatial saliency detection PR takes 2.79s (30%) of the overall time, while spatiotemporal saliency assignment takes only about 0.09s (0.01%).

[2] Supplemental material containing visual comparison of saliency maps of different methods for all the datasets is available at https://drive.google.com/file/d/0B_iCUhDh1LZvSklzUjhjTXk2NFE/view?usp=sharing

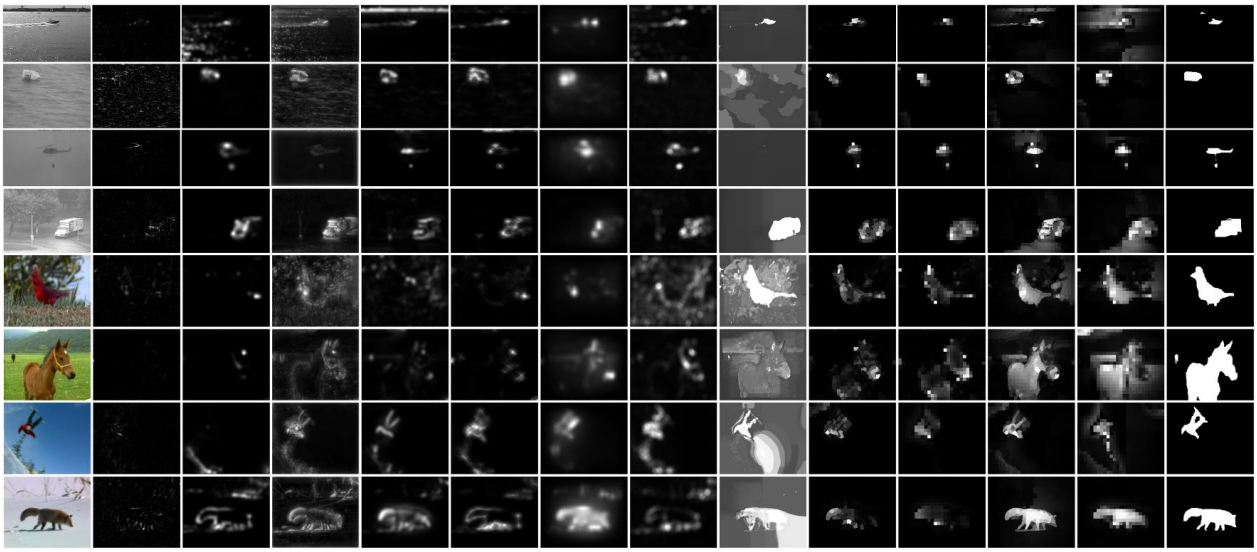


Fig. 19. Visual comparison of the saliency maps of state-of-the-art temporal and spatiotemporal saliency detection methods and the proposed temporal and spatiotemporal saliency detection methods. The top four rows and bottom four rows are the saliency maps computed by different methods for videos in DSD dataset and VSD dataset correspondingly.

Method	SEG	SeR	FastSUN	Simpsal	GBVS	PQFT	PD	PCAWSal	Our LGT	Our FLGT	Our STS	Our FSTS
Time(s)	22.38	0.99	0.53	0.46	0.26	0.19	0.11	0.08	6.43	5.96	9.23	8.19
Code	Matlab	Matlab	C++	Matlab	Matlab	Matlab	Matlab	Matlab	Matlab	Matlab	Matlab	Matlab

Table. 6. Running time taken for different methods to process a 400×300 video frame.

Optical flow estimation takes 44% of the time of the temporal saliency approach LGT. The most time consuming process in temporal saliency estimation is optical flow estimation and patch level flow abstraction, which take 45% of the time of LGT. The local and global temporal saliency estimations take 4% and 0.02% of the time of overall time, where fusion of local and global saliencies demand only about 6% of the time. The uniform sampling based patch segmentation in FLGT reduces the computation time of LGT by 8%. Furthermore, the faster patch segmentation and region segmentation in FSTS reduce the computation time of STS by 12%.

Without resizing the image with maximum dimension being 400 pixels, the times taken for LGT and STS to process a video frame with resolution 600×800 (4 times larger than the resized image) are 21. 98s and 35.25s correspondingly. So, the image resizing reduces the computation time of LGT by 71%, where the running time is reduced by 74% for STS. Without any resizing, the times taken for LGT and STS to process a video frame with resolution 150×200 (4 times smaller than the resized image) are 2.84s and 1.65s, respectively. Thus, the resizing increases the computing time for video frames with smaller dimensions. So, image resizing can be avoided for spatiotemporal saliency estimation in videos with maximum pixel resolution lower than 400. Due to the technical advancements in smartphones and digital cameras, images and videos captured using these devices usually come with higher pixel resolution. Therefore, image resizing is highly desired for faster estimation of spatiotemporal saliency in many real world applications.

The running time of the proposed spatiotemporal saliency detection approach is slower since the implementation of the proposed method is an unoptimized Matlab code. So, faster saliency estimation can be achieved by an optimized C++ implementation for real-time performance.

6. Conclusion and Future Work

A novel spatiotemporal salient region detection approach is presented in this paper. The spatial salient region detection method integrates both patch level and region level abstractions in a unified way. The spatial salient region detection method robustly detects salient regions irrespective of the number of regions and type of region segmentation method. The simple and computationally efficient adaptive saliency refinement approach uniformly highlights the salient regions and inhibits saliencies of the background noises. Moreover, the proposed patch level motion abstraction approach efficiently abstracts optical flow of a video frame, which assists the temporal saliency detection to achieve robust and effective salient object detection performance. The multi-level center surround difference based local temporal saliency and the parameter free global temporal saliency were combined in a unified way for temporal salient region detection. By considering the wider applications of saliency detection and their different need in speed and accuracy, faster variants of the proposed salient region detection methodologies were also presented in this paper.

The experiments were conducted on two widely used surveillance video datasets and two challenging salient object detection video datasets. Experimental results of the proposed temporal and spatiotemporal salient region detection methodologies have shown their potential and robust performance in salient object detection in comparison to several state-of-the-art methods. It is worthwhile to note that the faster variants of the proposed temporal and spatiotemporal salient region approaches also outperformed peer approaches in some cases. The experimental results on the performance of the individual steps in the proposed approaches, as well as the influence of inner parameter in temporal saliency detection were also presented. The performance of different integration schemes for temporal and spatiotemporal saliency fusions were analyzed using both quantitative and qualitative evaluations. In addition to quantitative comparison with the other methods, visual and time comparisons of the proposed approaches and other methods were also presented in this paper.

The performance of the spatial salient region detection can be improved by incorporating other spatial saliency cues such as *border prior*, *semantic prior*, *color prior* etc. Similar to the spatial saliency refinement approach, a suitable temporal saliency refinement approach will also be proposed for reducing temporal saliency assignment to background regions in future. The proposed spatiotemporal salient region approach has many potential applications, such as saliency based video retargeting, video compression, video summarization, etc., all of which will form the focus of our future efforts.

References

[1] J. Han, K. Ngan, M. Li, H. Zhang, Unsupervised extraction of visual attention objects in color images, IEEE Transactions on Circuit Systems and Video Technology 16 (1) (2006) 141-145.

[2] Z. Ren, S. Gao, L. T. Chia, I. Tsang, Region-based saliency detection and its application in object recognition, IEEE Transactions on Circuit Systems and Video Technology 24 (5) (2014) 769-779.

[3] C. Guo, L. Zhang, A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression, IEEE Transactions on Image Processing 19 (1) (2010) 185-198.

[4] S. Goferman, L. Zelnik-Manor, A. Tal, Context-aware saliency detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (10) (2012) 1915-1926.

[5] K. Fu, C. Gong, J. Yang, Y. Zhou, I. Yu-Hua Gu, Superpixel based color contrast and color distribution driven salient object detection, Signal Processing: Image Communication 28 (10) (2013) 1448-1463.

[6] S. Marat, M. Guironnet, D. Pellerin, Video summarization using a visual attention model, in: 15th European

Signal Processing Conference, 2007, pp. 3-7.
[7] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (11) (1998) 1254-1259.
[8] T. Liu, Z. Yuan, J. Sun, J.Wang, N. Zheng, X. Tang, H.Y. Shum, Learning to detect a salient object. IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2) (2011) 353-367.
[9] M. M. Cheng, G. X. Zhang, N. J. Mitra, X. Huang, S. M. Hu, Global contrast based salient region detection, in: Computer Vision and Pattern Recognition, 2011, pp. 409-416.
[10] F. Perazzi, P. Krahenbuhl, Y. Pritch, A. Hornung, Saliency filters: Contrast based filtering for salient region detection, in: Computer Vision and Pattern Recognition, 2012, pp. 733-740.
[11] X. Shen, Y. Wu, A unified approach to salient object detection via low rank matrix recovery, in: Computer Vision and Pattern Recognition, 2012, pp. 853-860.
[12] Z. Jiang, L. S. Davis, Submodular salient region detection, in: Computer Vision and Pattern Recognition, 2013, pp. 2043-2050.
[13] A. Borji, L. Itti, Exploiting local and global patch rarities for saliency detection, in: Computer Vision and Pattern Recognition, 2012, pp. 478-485.
[14] H. Fu, X. Cao, Z. Tu, Cluster-based co-saliency detection, IEEE Transactions on Image Processing 20 (10) (2013) 3766-3778.
[15] M. M. Cheng, J. Warrell, W. Y. Lin, S. Zheng, V. Vineet, N. Crook, Efficient salient region detection with soft image abstraction, in: International Conference on Computer Vision, 2013, pp. 1529-1536.
[16] N. Bruce, J. Tsotsos, Saliency based on information maximization, in: Advances in Neural Information Processing Systems, 2005, pp. 155-162.
[17] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, in: Advances in Neural Information Processing Systems, 2006, pp. 545-552.
[18] X. Hou, L. Zhang, Saliency detection: A spectral residual approach, in: Computer Vision and Pattern Recognition, 2007, pp. 1-8.
[19] C. Jung, C. Kim, A unified spectral-domain approach for saliency detection and its application to automatic object segmentation, IEEE Transactions on Image Processing 21 (3) (2012) pp.1272-1283.
[20] Z. Ren, S. Gao, L. T. Chia, D. Rajan, Regularized feature reconstruction for spatio-temporal saliency detection, IEEE Transactions on Image Processing 22 (8) (2013) 3120-3132.
[21] M. Mancas, Relative influence of bottom-up and top-down attention, in: Proceedings of International Workshop on Attention in Cognitive Systems, 2009, pp. 212-226.
[22] N. Riche, M. Mancas, B. Gosselin, T. Dutoit, Rare: a new bottom-up saliency model, in: Proceedings of IEEE International Conference of Image Processing, 2012, pp. 641-644.
[23] N. Riche, M. Mancas, M. Duvinage, M. Mibulumukini, B. Gosselin, T. Dutoit, RARE2012: a multi-scale rarity-based saliency detection with its comparative statistical analysis, Signal Processing: Image Communications, 28(6) (2013) 642-658.
[24] A. Borji, L. Itti, State-of-the-art in visual attention modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (1) (2013), 185-207.
[25] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, T. Dutoit, Saliency and human fixations: State-of-the-art and study of comparison metrics, in: Proceedings of IEEE International Conference on Computer Vision, 2013, pp. 1153 - 1160.
[26] Q. Yan, L. Xu, J. Shi, J. Jia, Hierarchical Saliency Detection, in: Conference on Computer Vision and Pattern Recognition, 2013, pp. 1155-1162.
[27] R. Kannan, G. Ghinea, and S. Swaminathan, Salient region detection using patch level and region level image abstractions, IEEE Signal Processing Letters 22 (6) (2015) 686-690.
[28] S. Marat, T. H. Phuoc, L. Granjon, N. Guyader, D. Pellerin, A. Guerin Dugue, Modelling spatio-temporal saliency to predict gaze direction for short videos, International Journal on Computer Vision 82 (3) (2009) 231-243.
[29] J. Seo, P. Milanfar, Static and space-time visual saliency detection by self-resemblance, Journal of Vision 9 (12) (2009) 1-17.
[30] V. Mahadevan, N. Vasconcelos, Spatiotemporal saliency in dynamic scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010) 171-177.
[31] W. Kim, C. Jung, C. Kim, Spatiotemporal saliency detection and its applications in static and dynamic scenes, IEEE Transactions on Circuits and Systems for Video Technology 21 (4) (2011) 446-456.

[32] D.Y. Chen, H. R. Tyan, D. Y. Hsiao, S. W. Shih, H. Y. Liao, Dynamic visual saliency modeling based on spatiotemporal analysis, in: International Conference on Multimedia and Expo, 2008, pp. 1085-1088.
[33] B. Wu, L. Xu, L. Zeng, Z. Wang, Y. Wang, A unified framework for spatiotemporal salient region detection, EURASIP Journal on Image and Video Processing 2013 (16) (2013) 1-12.
[34] Y. Wei, F. Wen, W. Zhu, J. Sun, Geodesic saliency using background priors, in: European Conference on Computer Vision, 2012, pp. 29-42.
[35] A. Borji, M.M. Cheng, H. Jiang, J. Li, Salient object detection: a survey, arXiv preprint arXiv:1411.5878, (2014).
[36] C. Koch, S. Ullman, Shifts in selective visual attention: towards the underlying neural circuitry, Human Neurobiology 4 (4) (1985) 219-227.
[37] Y.F. Ma, H. Zhang, Contrast-based image attention analysis by using fuzzy growing, in: ACM Multimedia, 2003, pp. 374-381.
[38] R. Achanta, S. S. Hemami, F. J. Estrada, S. Susstrunk, Frequency-tuned salient region detection, in: Computer Vision and Pattern Recognition, 2009, pp. 1597-1604.
[39] L. Duan, C. Wu, J. Miao, L. Qing, Y. Fu, Visual saliency detection by spatially weighted dissimilarity, in: Computer Vision and Pattern Recognition, 2011, pp. 473-480.
[40] V. Gopalakrishnan, Y. Hu, D. Rajan, Salient region detection by modeling distributions of color and orientation, IEEE Transactions on Multimedia 11(5) (2009) 892-905.
[41] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: Conference on Computer Vision and Pattern Recognition 2 (1999) pp. 1063–6919.
[42] M. Elgammal, D. Harwood, L. S. Davis, Non-parametric model for background subtraction, in: European Conference on Computer Vision, 2000, pp. 751-767.
[43] M. Heikkilä, M. Pietikäinen, A texture-based method for modeling the background and detecting moving objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (4) (2006) 657-662.
[44] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, S. Li, Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes, in: Computer Vision and Pattern Recognition, 2010, pp. 1301-1306.
[45] L. Wixson, Detecting salient motion by accumulating directionally consistent flow. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (80) (2000) 774-780.
[46] A. Bugeau, P. Perez, Detection and segmentation of moving objects in highly dynamic scenes, in: Computer Vision and Pattern Recognition, 2007, pp. 1-8.
[47] Y. Zhai, M. Shah, Visual attention detection in video sequences using spatiotemporal cues, in: ACM Multimedia, 2006, pp. 815-824.
[48] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, SLIC superpixels compared to state-of-the-art superpixel methods, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (11) (2012), 2274-2282.
[49] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Advances in Neural Information Processing Systems, 2002, pp. 849-856.
[50] M. Filippone, F. Camastra, F Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern recognition 41 (1) (2008) 176-190.
[51] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in: Advances in Neural Information Processing Systems, 2004, pp. 1601-1608.
[52] C. Yang, L. Zhang, H. Lu, Graph-regularized saliency detection with convex-hull-based center prior, IEEE Signal Processing Letters 20 (7) (2013) 637-640.
[53] C. Liu, Beyond pixels: Exploring new representations and applications for motion analysis, Doctoral Thesis, Massachusetts Institute of Technology, May 2009.
[54] T. Brox, A. Bruhn, N. Papenberg, J.Weickert, High accuracy optical flow estimation based on a theory for warping, in: European Conference on Computer Vision, 2004, pp. 25-36.
[55] A. Bruhn, J.Weickert, C. Schnorr, Lucas/Kanade meets Horn/Schunk: combining local and global optical flow methods, International Journal of Computer Vision 61 (3) (2005) 211-231.
[56] B. Zhou, X. Hou, L. Zhang, A phase discrepancy analysis of object motion. In: Asian Conference on Computer Vision, 2010, pp. 225-238.
[57] H.R. Tavakoli, E. Rahtu, J. Heikkilä, Temporal saliency for fast motion detection, in: Asian Conference on Computer Vision Workshops, 2012, pp. 321-326.

[58] N.J. Butko, L. Zhang, G. W. Cottrell, J. R. Movellan, Visual saliency model for robot cameras. in: IEEE International Conference on Robotics and Automation, 2008, pp. 2398-2403.
[59] J. Harel, A Saliency Implementation in MATLAB, URL: http://www.klab.caltech.edu/~harel/share/gbvs.php
[60] E. Rahtu, J. Kannala, M. Salo, J. Heikkila, Segmenting salient objects from images and videos, in: European Conference on Computer Vision, 2010, pp. 366-379.
[61] J. Ferryman (Ed.), in: International Workshop PETS, 2001. URL: ftp://ftp.pets.rdg.ac.uk/pub/PETS2001, Accessed: September 2014.
[62] J. Davis, V. Sharma, Background-subtraction using contour based fusion of thermal and visible imagery, Computer Vision Image Understanding 106 (2-3) (2007) 162–182, URL: http://www.cse.ohio-state.edu/otcbvs-bench, Accessed: September 2014.
[63] K. Fukuchi, K. Miyazato, A. Kimura, S. Takagi, J. Yamato, Saliency-based video segmentation with graph cuts and sequentially updated priors, in: IEEE International Conference on Multimedia and Expo, 2009, pp. 638-641.

Rajkumar Kannan

Rajkumar Kannan received the B.Sc and M.Sc degrees in Computer Science from Bharathidasan University – Tiruchirappalli, India in 1991 and 1993 respectively and the PhD degree in Multimedia Datamining from National Institute of Technology – Tiruchirappalli, India in 2007. Rajkumar works for King Faisal University, Saudi Arabia in the College of Computer Science and Information Technology. His research activities primarily lie at the confluence of multimedia, information retrieval, semantic web, social informatics and collective intelligence. Rajkumar is a member of ACM and life member of CSI-India and ISTE-India.

Gheorghita Ghinea

Gheorghita Ghinea received the B.Sc. and B.Sc. (Hons.) degrees in computer science and mathematics, and the M.Sc. degree in computer science from the University of the Witwatersrand, Johannesburg, South Africa, in 1993, 1994, and 1996, respectively, and the Ph.D. degree in computer science from the University of Reading, Reading, U.K., in 2000. He is a Reader in the School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, U.K. His current research interests include multimedia computing, telemedicine, quality of service, as well as computer networking and security issues.

Sridhar Swaminathan

Sridhar Swaminathan received his bachelors and masters degrees in Computer Science from Bishop Heber College (Autonomous), Tiruchirappalli, India. He is currently pursuing the Ph.D. degree in Computer Science at the Department of Computer Science, Bishop Heber College (Autonomous), Tiruchirappalli, India. His main interests are in Computer Vision, Information Retrieval and Machine Learning.