# Saliency Cut:

# an Automatic Approach for Video Object Segmentation

# Based on Saliency Energy Minimization

by

Yilin Wang

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2013 by the
Graduate Supervisory Committee:

Baoxin Li, Chair
Yalin Wang
David Claveau

ARIZONA STATE UNIVERSITY

August 2013

# ABSTRACT

Video object segmentation (VOS) is an important task in computer vision with a lot of applications, e.g., video editing, object tracking, and object based encoding. Different from image object segmentation, video object segmentation must consider both spatial and temporal coherence for the object. Despite extensive previous work, the problem is still challenging.

Usually, foreground object in the video draws more attention from humans, i.e. it is salient. In this thesis we tackle the problem from the aspect of saliency, where saliency means a certain subset of visual information selected by a visual system (human or machine)[1]. We present a novel unsupervised method for video object segmentation that considers both low level vision cues and high level motion cues. In our model, video object segmentation can be formulated as a unified energy minimization problem and solved in polynomial time by employing the min-cut algorithm. Specifically, our energy function comprises the unary term and pair-wise interaction energy term respectively, where unary term measures region saliency and interaction term smooths the mutual effects between object saliency and motion saliency. Object saliency is computed in spatial domain from each discrete frame using multi-scale context features, e.g., color histogram, gradient, and graph based manifold ranking. Meanwhile, motion saliency is calculated in temporal domain by extracting phase information of the video. In the experimental section of this thesis, our proposed method has been evaluated on several benchmark datasets. In MSRA 1000 dataset [2] the result demonstrates that our spatial object saliency detection is superior to the state-of-art methods. Moreover, our temporal motion saliency detector can achieve better performance than existing motion detection approaches in UCF sports action analysis dataset [3] and Weizmann dataset [4] respectively. Finally, we show the attractive empirical result and quantitative evaluation of our approach on two benchmark video object segmentation datasets.

To my family who support me over the years.

# ACKNOWLEDGEMENTS

My sincerest gratitude first goes to my research advisor Baoxin Li. Working him over two years has been an invaluable experience to me. He discovered my research potential after we first met in person. He brought me to this great research group, mentored me, and gave courage to me. Not only did he give me the insightful opinion and constructive comments but also spent lots of time to dig the essential nature behinds the errors I made.

I also need to give thanks to my dear lab mates: Qiang Zhang, Lin Chen, Qiongjie Tian ,Xu Zhou, Collin Walker, Bindu, Parag, Ragave, Achtchna and Peng Zhang. They gave invaluable suggestions and comments to me. Moreover, all of them are very kind, answered the life and school related questions and shared their own experience to me.

At last I need gave me gratitude to my families. Thanks to my girl friend Yi Qin for a wonderful time we spent together.

TABLE OF CONTENTS

# LIST OF TABLES

Table Page

## LIST OF FIGURES

# Chapter 1

# Introduction

In past two decades, video object segmentation (VOS) has been widely used in high level computer vision applications, e.g., video editing, object tracking, object based encoding, and activity recognition. Compared to image object segmentation which aims to group the similar pixels into labeled regions, video object segmentation takes both spatial coherence and temporal coherence into consideration. The concept of video object was proposed in MPEG-4 standard, which represented most object-like regions in an image or video frame. It usually was a meaningful unit, e.g. human, car, man-made object. Since, lacking of knowledge of video content, lots of object detectors are trained from a large number of labeled images and designed for one single object class detection [11, 12]. However, their performances will be inevitably decreased when applied to videos, which have diverse conditions that distinguish from training data. Several works [13, 14] explored motion tracking or region clustering over time for a general object detection in the video. Besides the difficulties in tracking, e.g., occlusion, drifting, these techniques do not have a clear definition of what is a foreground object. Therefore, grouping the interest pixels usually leads to over segmentation and lack of semantic meanings.

In this thesis we propose an unsupervised video object segmentation approach (Figure 1.1) which can automatically detect the object like regions by modeling the object appearance. Our main idea is using spatial saliency features to identify the object regions and employing temporal motion saliency to estimate the dynamic cues through the motion trajectory. We begin with a general object detector by adapting saliency features. Inspired by [2, 15, 16, 17], we assume that video foreground object should be the most saliency object among the frame based on human visual attention mechanism [18, 19]. To capture the object like regions, we

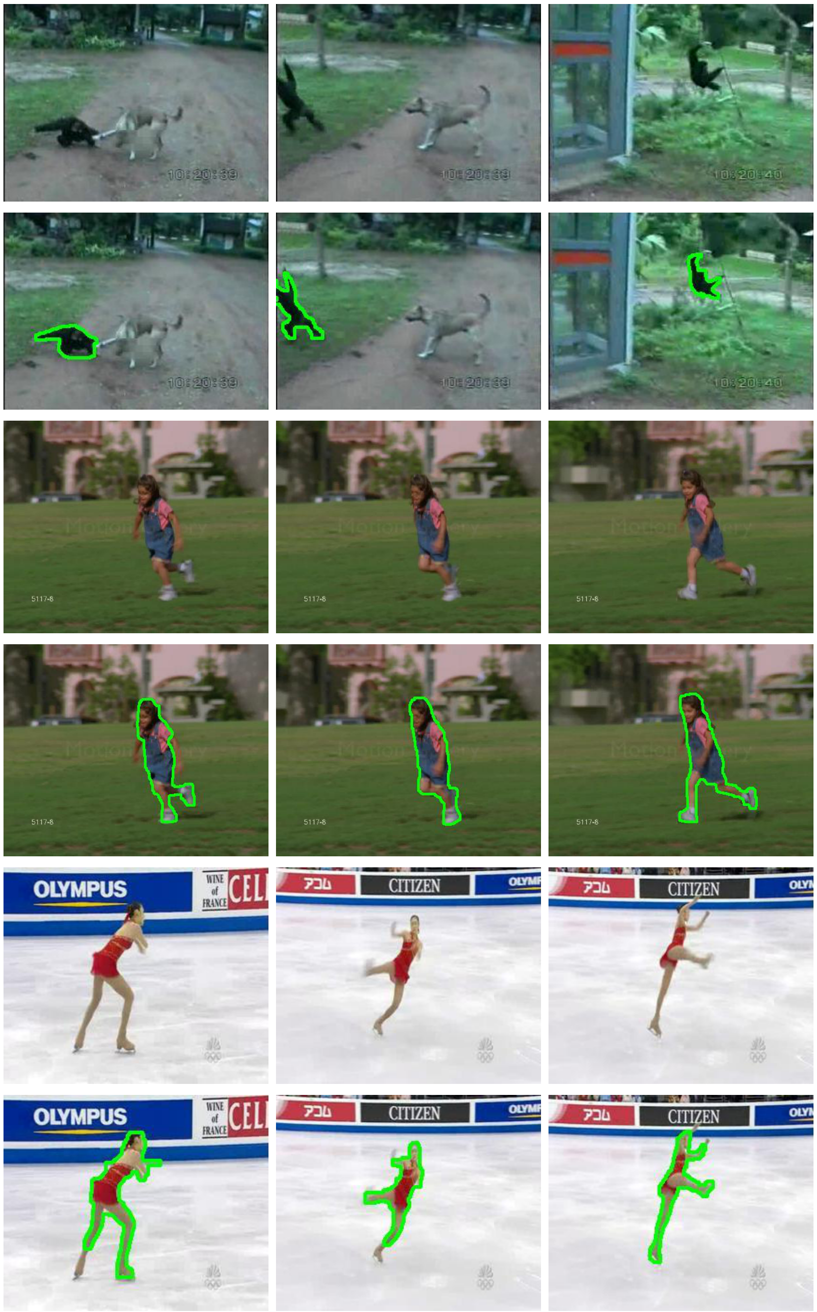


Figure 1.1: Our result on well known benchmark datasets [5][6].

define three characters of saliency object based on psychological evidence[20]:

- the salient object is different from its surrounding context.
- the salient object is probably in the center of an image.
- the salient object should have the property of object (strong gradient, clear boundary).

Following these characters and our assumption, our proposed method starts from object measurement by measuring the probability of a region containing an object. Given a single frame, we use superpixels to segment it into small regions and each region is assigned a saliency value by multi-scale context features, e.g. spatial location, gradient. To link the saliency and object measurement, we propose a ranking function learned from spatial observations. Then the saliency region will be used as foreground object queries for object measurement. To capture motion cues, we define motion saliency by extracting the phase information of the whole video. Finally, estimating the foreground object in the video can be achieved by minimizing an energy function construed from a graphical model.

Why our saliency energy is related to video object? First, our saliency energy using appearance information, which is related to the human visual system and follows the classic bottom up saliency model [18], to measure the most object like regions. Second, our motion saliency suppresses the background motion recurrence, which means the saliency region moves differently from its background, i.e. it likely to be a human interest object [21]. Why our approach is better? Our saliency energy combination not only considers the object appearance but also integrates the object motion cues especially for the fast moving object. More importantly, our energy function can utilize the efficient min cut algorithm which can solve this challenging problem in polynomial time.

In this thesis we have three contributions: first, we propose a new generic saliency object detection and show its good performance in benchmark dataset. Second, we show our temporal saliency detector can be used in two key vision tasks, e.g., action analysis and abnormal event detection. Third, we view the video object detection problem in a new angle, which can solve the task in polynomial time.

## 1.1 Video Object Segmentation

Both video object detection and video object segmentation (VOS) are two important tasks in video analysis area. Compared to video object detection, video segmentation focuses on finding the exact object location and boundary. Despite extensive researches in the literature, VOS is still a challenging problem in computer vision area. Generally, the VOS can be achieved with the following two procedures : foreground object segmentation and motion filed segmentation.

### *1.1.1 foreground object detection and segmentation*

Object detection has been well researched in these years and addressed in some different scenarios: co-object segmentation, image parsing and figure-ground segmentation. Overall, foreground object in the video often means a moving object without repeated patterns, e.g., a boat which is sailing on the sea, a car which is on the road, therefore, comparing to foreground, background represents static objects. Recently, some graph based approaches detected foreground object by extracting the similar patterns through pairs or groups of unlabeled images, [22, 23]. As the object size is uncertain, such approaches can not guarantee the accuracy of detected region. To this end, [24] proposed an iterative boundary refinement approach for the foreground detection, which ranked the interest region in each image and optimized its location by other reference image. A similar approach [25] called ”key segment” has drawn lots of attentions, the author first used motion and appearance cues to get a set of key frames which were segmented into two regions: foreground and background. Then the foreground regions of other frames can be detected by ranking according to the foreground region of the key frames. These group based foreground approaches are heavily based on reference images, if the key frame object is not annotated correctly, e.g. a video contains a fast moving object, the result will degrade significantly. For semi automatic object detection approaches, foreground segmentation often needs human input, e.g., snapcut [26], user will annotate the foreground region or mark the foreground boundaries in some key frames or initial frames. These semi automatic approaches achieve better results than automatic ones, however, when the video dataset is huge, the time cost will be significant in the sense that user will need to mask the regions; in addition, the results have to rely on the user input.

### *1.1.2 motion field object segmentation*

Motion based object segmentation can be divided into two categories: over segmented region matching and tracking based object segmentation. The methods [14, 27] in the first group utilized bottom up features, e.g., shape, color, gradient, to segment each frame into different regions and match them with nearby frames. For tracking based methods, [5] used supervised approach for object initialization and SIFT points for object tracking. [13] employed dense flow to cluster the pixels through long term motion trajectories. There are two common problems of motion based video object segmentation, first, these methods tend to over segment the object, second, the motion regions are lack of semantic meaning.

## 1.2 Visual Saliency

Visual saliency means the region which draws most human attention. It is the capability of vision system to select the interest information which is a subset of received visual information. In this thesis, we propose two saliency based approaches for foreground object detection in static images and action, motion detection in video respectively.

In the recent years, visual saliency has attracted a lot of interests and efforts in the vision society. One of the most widely known works of visual saliency can be found in [18]. Since then, a lot of different models have been proposed for computing the visual saliency, which can be roughly divided into two groups: bottom-up models are mainly based on features of the visual scenes; and top-down model analyzes the human data and learns from these knowledge, tasks etc. Visual saliency has also been applied in conventional vision task, e.g., object detection [15, 16, 2, 17]. The saliency based object detection detects most salient regions and extracts the whole extent of the object from the background. Like the visual saliency model, the output of saliency object detection(SOD) is a salient map which shows the probability of each pixel belonging to a salient object. Moreover, SOD is also a special case of image segmentation problem It only gets salient object from background, while image segmentation problem aims to find the region where the pixels have the same property and coherence. Another question, which cannot be ignored, is the difference between saliency model and SOD. The answer is saliency model trying to predict the location of human eye fixation data and SOD segments the salient object. They are interact with each other. There are lots of recent approaches on SOD, [28] proposed a Bayesian approach for SOD and [29] improved by integrating superpixel and Harris interest points. Context based approach can be found in [16, 17]. [2] presented a supervised approach to detect saliency object by learning high level vision cues e.g. human face, street signs.

Recently, spectral-based approach has gained more interest due to its simplicity and good performance. In [30], the saliency map was computed based on spectrum residual together with the phase information. In [31], it was found that it is the phase information rather than the spectrum related to the saliency regions. The saliency model was based on quaternion Fourier transform which combined color feature and adjacent frame motion vectors. However, However, there was a lack of theoretic justification for such methods until [21], where it was shown that, if the background is sparsely supported in the DCT domain and the foreground is sparsely supported in the spatial domain the foreground will receive high value on the computed saliency map.

In the real world, we are interacting with visual information over the time, thus visual saliency should not only focus on the information of spatial domain, but should also consider the information along the temporal domain. To this end, motion saliency has been proposed, which tries to capture the region that is visually attractive in the video. Currently, there are some works: [32] used saliency for motion detection. [33] achieved good result in action analysis with spatiotemporal saliency features. However, current approaches are time consuming and not well explicating what is saliency represents in temporal domain.

## 1.3 Thesis Outline

In this thesis, we try to solve three computer vision tasks: salient object detection, action and motion analysis, video object segmentation. For each task, we develop an unsupervised algorithm and evaluate it on well known benchmark dataset.

The structure of this thesis is organized as following : the chapter 2 presents the technique details about saliency object detection. In chapter 3, we propose a novel spatiotemporal visual saliency detector for video analysis, based on the phase information of the video. In chapter 4, we further research our saliency object detection and spatiotemporal saliency detection. We demonstrate that our saliency approach can be applied to video objection segmentation tasks. In chapter 5, we conclude our three proposed algorithms and talk about future work.

# Chapter 2

## Foreground Object Detection

Image object detection has been a challenging problem and extensive researched in past decades, some successful approaches can be found in [12, 11, 23]. Most state of art approaches are designed for one object class and need large scale training data, which yield to the various environment. A generic object detector would cost expensively in the sense of extensive training set and the performance may degrade in some particular environment. Recently, salient object detection, which based on human vision system, has attracted lots of attention. [16] claimed a saliency object should be different from its context appearance, [34] presented a global color histogram contrast approach. [15] proposed a supervised saliency approach for generic object class detection by measuring the region ”objectness”, which is a notation of how likely a window contains an object. Inspired by these previous work, we propose a novel saliency object approach, we argue that saliency object should not only depend on the context appearance, but also related to the boundary and divergence values. Different with [15] which measures object boundary by a sliding window. We measure boundary by regions, and compare the saliency region with its four connected neighborhood context. Because lack of knowledge of saliency object, recent approaches [16, 17] employed multi-scale context, intuitively, we follow this idea in a different way by employing multiple superpixel scales.

Graph cut algorithm [8] has been successful applied in image segmentation task, however, because the foreground object location is uncertain, most state of art graph cut based algorithms need user input. In this chapter, we demonstrate that our saliency approach can be effectively used in graph cut algorithm for data term initialization. In our case, source node S and terminal node T represent image foreground and background respectively, e.g. most saliency region is assigned to S node. At last, the object segmentation problem can be solved by min cut algorithm efficiently. In evaluation part, we use well known MSRA saliency object detection dataset and manually binary mask ground truth provided by [35].

The structure of this chapter is as following, section 2.1 introduces the superpxiel. In section 2.2 we introduce our proposed method and give the analysis. In section 2.3 we discuss about the application of our proposed method. At last, both quantitative and qualitative experimental results for MSRA dataset are demonstrated in section 2.4.

## 2.1 SLIC superpixel

Recently, superpixel has become a popular technique for image segmentation, which not only represents richer feature than pixel, but also greatly reduces computational complexity. [36] proposed a graph based superpixel method based on local nearest neighborhood, and other state-of-art approach can be found in [37, 38]. However, these approaches are to slow. In [39], the SLIC superpixel is computed by clustering the pixels based color similarity and spatial distance. Give an image $I$ with size $M \times N$ and the number of superpixels $K$, where Figure 2.1 demonstrates the result given different superpixel numbers for one image. Instead of RGB color space, for more perceptual accuracy all the pixels are clustered in $LAB$ color distance and spatial Euclidean distance. The cluster center $C$ is computed by grid interval:

$$C = \sqrt{\frac{M \times N}{K}} \tag{2.1}$$

The $Lab$ color distance and spatial Euclidean distance is computed as :

$$D_l ab = \sqrt{(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2} \tag{2.2}$$

$$D_s = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2.3}$$

Finally the cluster measurement is:

$$D = D_l ab + \frac{10}{C} D_s \tag{2.4}$$

Where $i$ and $j$ are different pixels belong to $I$. Usually the clustering will be converged within 20 iterations and the time complexity of this algorithm is $O(N)$.

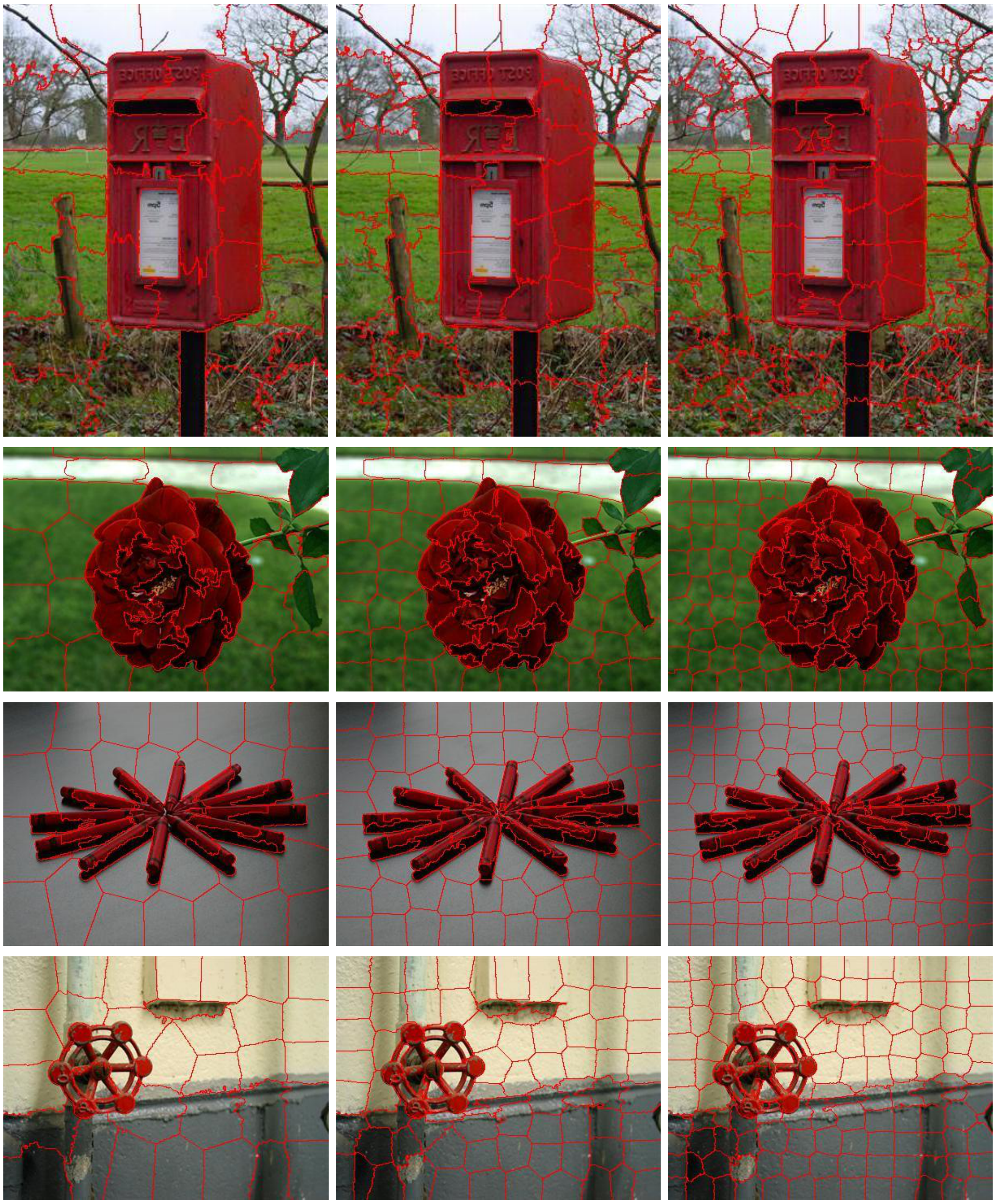


Figure 2.1: The result of different superpixel scale.

## 2.2 Region Based Saliency Feature Extraction

Based on the saliency object characters defined in Chapter 1, a superpixel (region) is salient depends on how is it different with its context which is a four-neighbor connected superpixels system, in appearance and boundary information. To achieve a better result and meaningful context information, we employ multi scale of superpixel regions. Given a image $I$ and scales $n$, for each scale $I$ will be segmented into $K^n$ regions $\{SP_i^n\}_{i=1}^K$. For one superpiexl $SP_i^n$, its context would be its four neighborhood superpixels and its saliency value is computed as:

$$Sal_{SP_i} = \sum_{i \neq j}(G(i,j) \times d(i,j)) + \rho B(i,j)) + \sum_{j \in R}(H(SP_i, SP_j)) \tag{2.5}$$

Where $i, j \in SP$. The first term of equation calculates the global contrast, $G$ and $B$ represent the gradient magnitude and boundary difference respectively and $\rho$ is weight ratio, here boundary information is calculated by Canny boundary detector. $d$ is Gaussian distance between two pixels. The second term in this equation computes the local context contrast. $H$ is $\chi$ distance between CIE LAB histograms of two pixels. The $\rho$, $d$ and $\chi$ distance are defined as below:

$$\rho = \sum_{j \in R_i} \frac{B(i,j)}{\sum B} \tag{2.6}$$

$$d = e^{\frac{-(c(i)-c(j))^2}{2\sigma^2}} \tag{2.7}$$

$$H(x,y) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)} \tag{2.8}$$

Finally, the saliency map is calculated as:

$$SalMap = \frac{1}{n}\sum_{i=1}^{n} Sal \cdot SpatialPrior \tag{2.9}$$

The saliency map is computed as the mean of different scales. The *spatialprior* is our second saliency cue, which satisfies center object should be more salient. It defined as:

$$SpatialPrior = e^{-0.5(\frac{dx_i^{(n)}}{w^2} + \frac{dy_i^{(n)}}{h^2})} \tag{2.10}$$

where $i$ is the $ith$ pixel(not superpixel) in image $I$, $dx, dy$ is Euclidean distance from pixel $i$ to image center and $h, w$ is image height and width respectively. Figure

2.2 and Figure 2.3 demonstrate the saliency map of different approaches: first row is input image,from second to sixth saliency map of Itti *et al*.[18], Goferman *et al*.[16], Achanta *et al*. [40], Zhang *et al*.[28], Fang *et al*. [41], Jiang *et al*.[17], and our proposed model. From visual comparison, foreground object in our proposed model are almost uniformly detected

## 2.3 Interactive with Graph Cut

In Figure 2.4 we show that our saliency map can help saliency object segmentation. First column (left) in the figure is input image, second column is our saliency map with graph cut, third column is our saliency map without graph cut

In recent years, graph cut has been proved to be to a useful tool in low level computer vision problem, such as image smoothing, image editing, stereo correspondence. Such problems can be formulated as an energy minimization problem via constructing a graphical model and solved by max flow/min cut algorithm in polynomial time.

For a directed graph $G =< V, E >$ with two nodes which called source node $s$ and sink node $t$, and each edge with a capacity $c(u, v)$ , a flow $f$ starts from $s$ to $t$ and can not exceed than the capacity in each edge. The maxflow problem is defined as finding a maximization flow in the network, Figure 2.5 shows an example of maxflow problem, the current flow is 28 and it is the maxflow. Moreover, there many algorithms can solve this problem in polynomial time, e.g. Ford-Fulkerson, Edmond-Carp.

Figure 2.6 shows the binary label graph cut. Let denote A as a binary edge map of an image, and L be a set of labels, P represents the set of pixels. The energy function can be written as

$$E(A) = \rho R(A) + B(A) \tag{2.11}$$

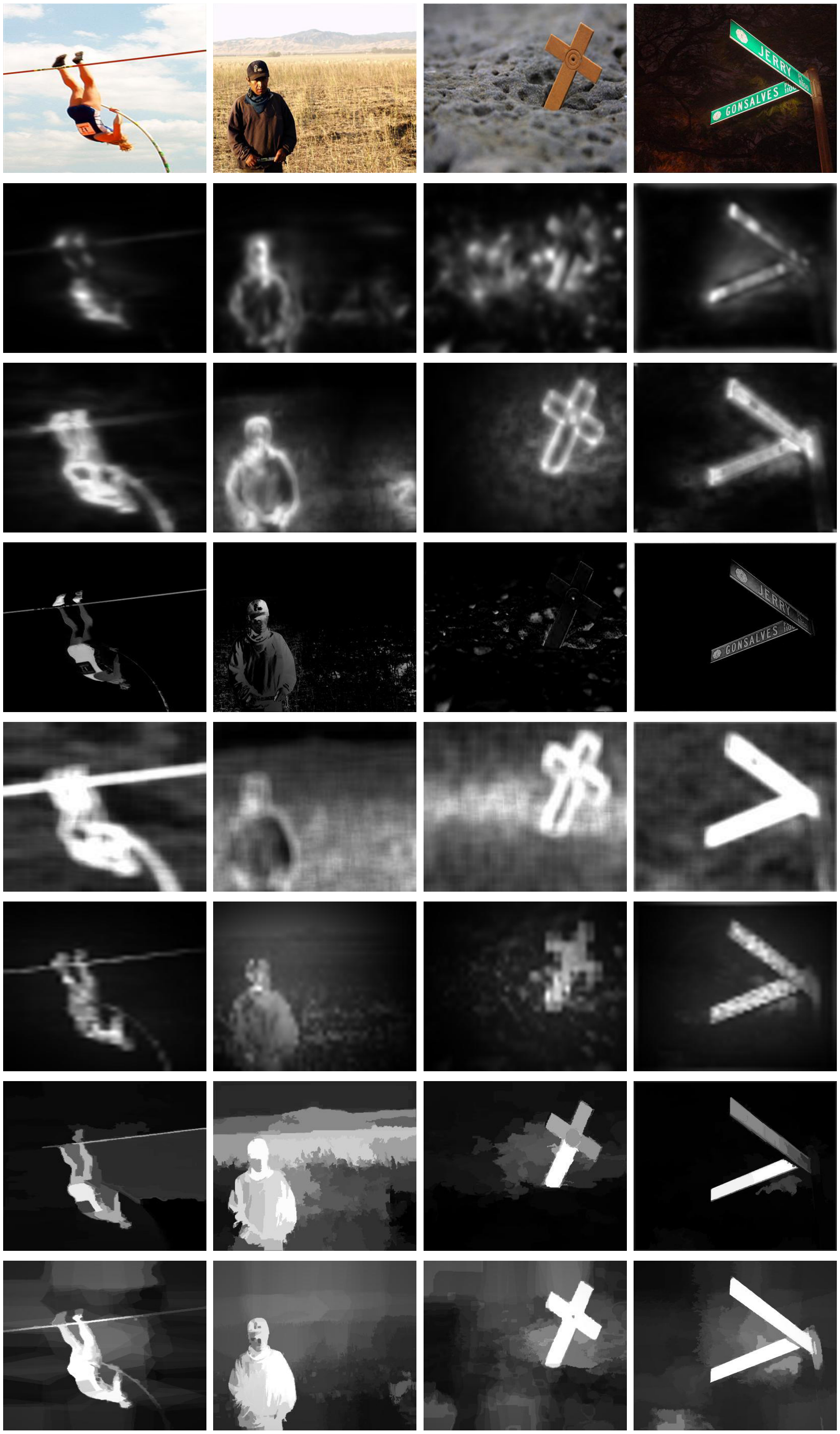


Figure 2.2: Visual comparison the saliency map of different approaches(nature).

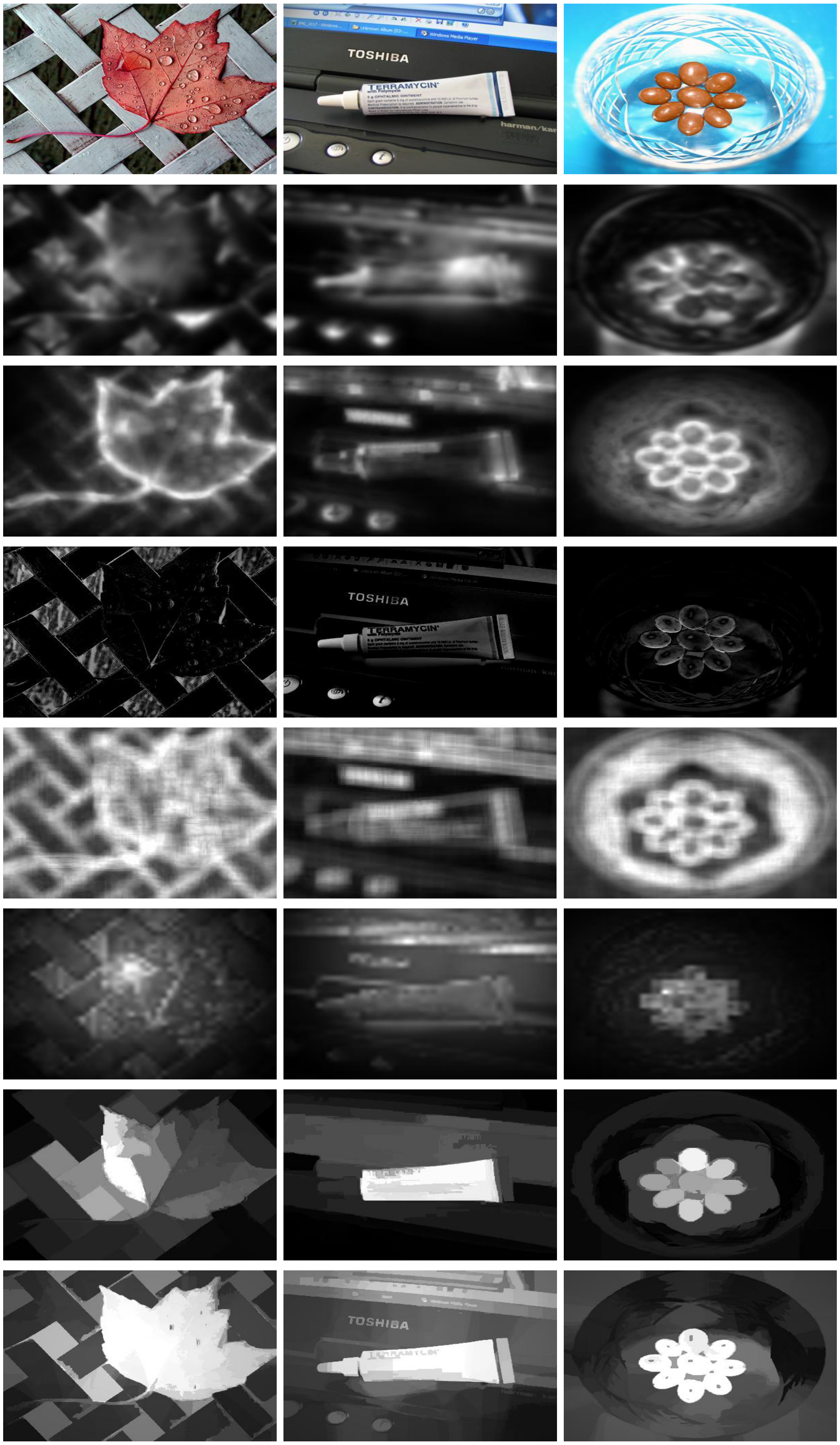


Figure 2.3: Visual comparison the saliency map of different approaches(manmade).

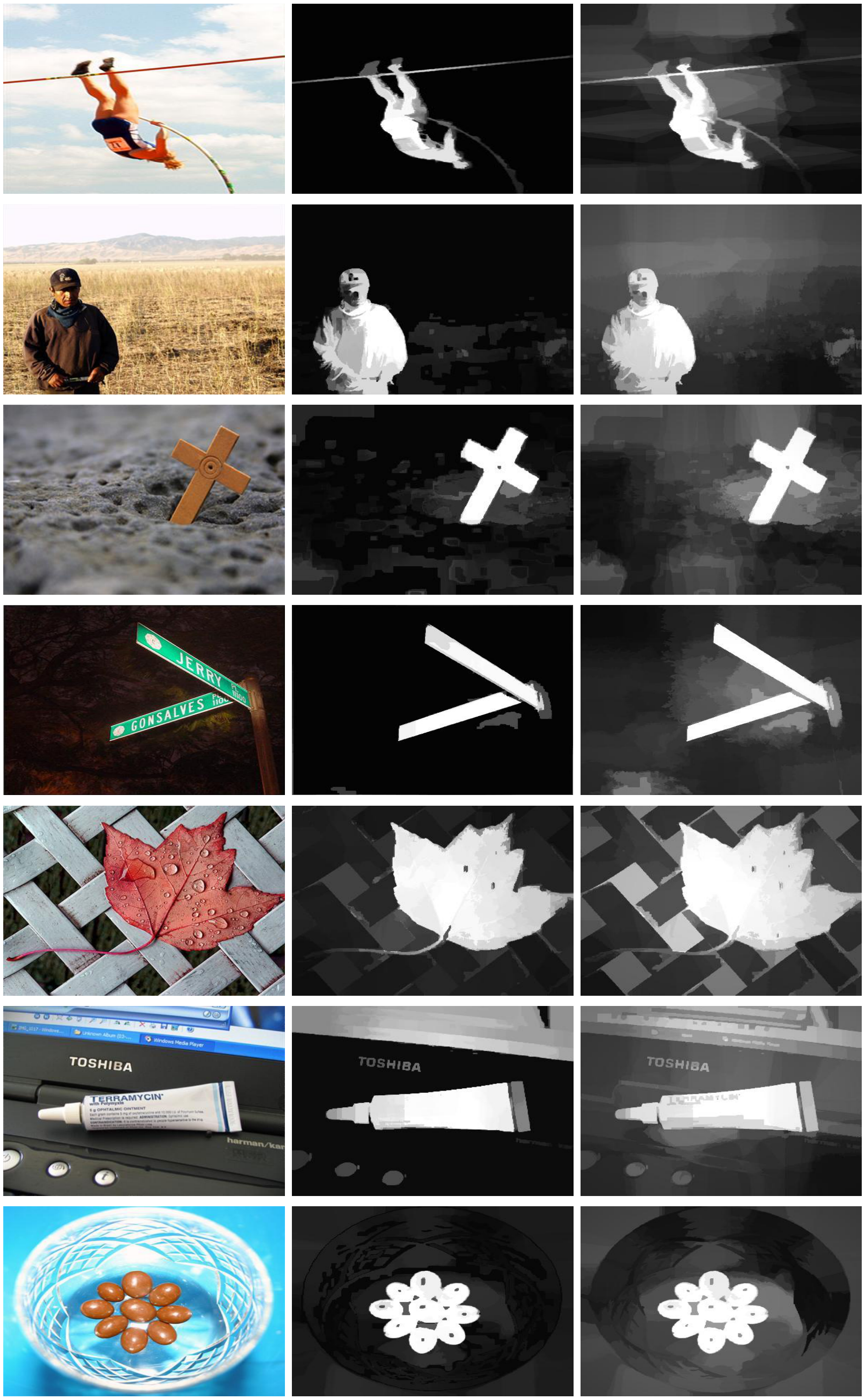


Figure 2.4: Visual comparison of saliency map and saliency map with graph cut.

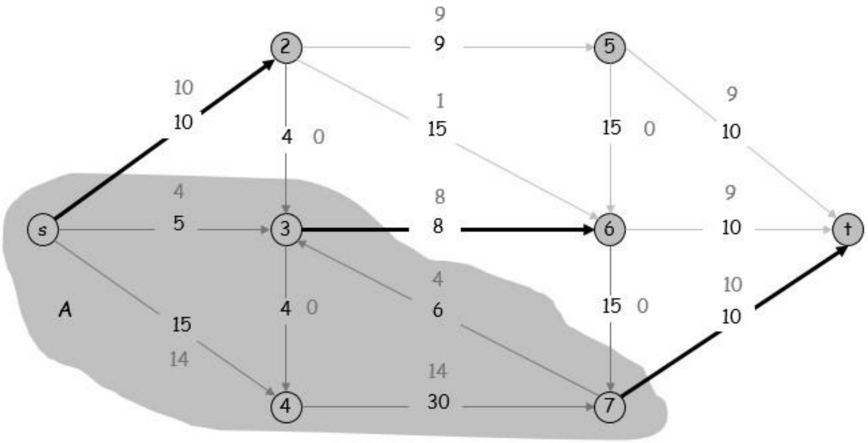
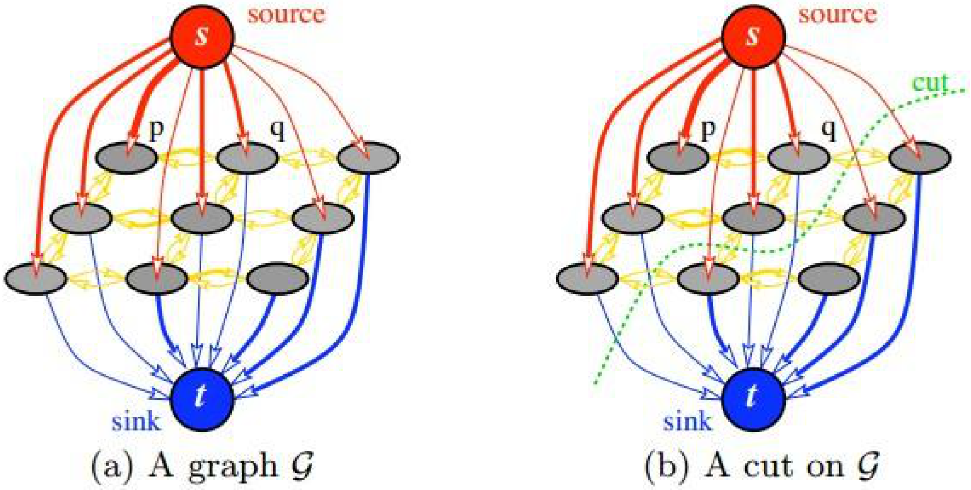


Figure 2.5: Example of maxflow, taken from [7].



(a) A graph $\mathcal{G}$

(b) A cut on $\mathcal{G}$

Figure 2.6: Graph cut demonstration ,taken from[8].

$$B(A) = \sum_{v1,v2 \in N, A_{v1} \neq A_{v2}} e^{-\frac{(I_v 1 - I_v 2)^2}{2\sigma^2} \cdot \frac{1}{d(v1,v2)}} \tag{2.12}$$

where $B$ is smoothness term for image region, $R$ is unary term and it represents the penalty to assign label to pixel $v$, $N$ represents the four neighborhood region, and $d$ is Euclidean distance between two pixels. This energy model will group all the pixels into two group $foreground, background$ . Usually, to identify these two label, people usually applied K-means clustering for initialize the two labels (shown in Figure 2.7). Another way needs user the initialization shown in Figure 2.8.

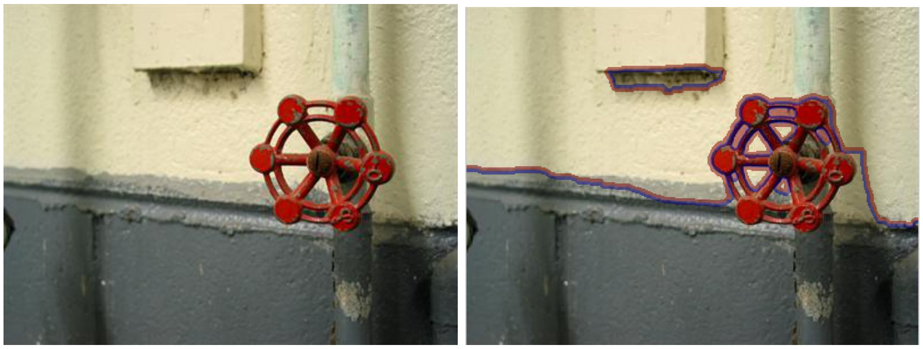


Figure 2.7: Example of automatic segmentation by kmeans initialization.

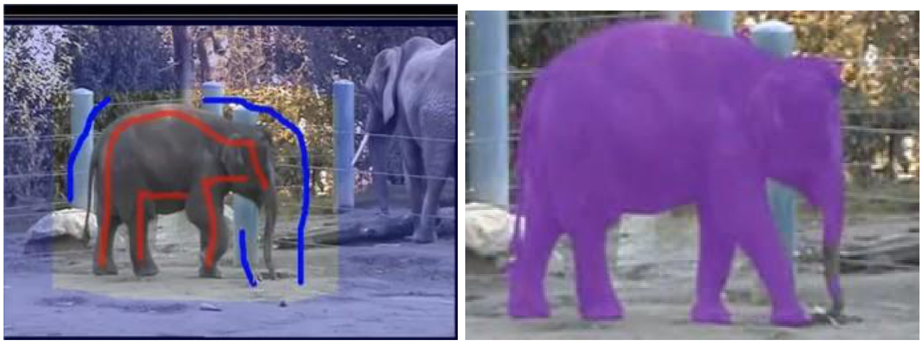


Figure 2.8: Example of semi-automatic segmentation by user input.

Thus , in our interactive model we define saliency values as the initialize value for the term $R$.

$$R_p(object) = Probability(saliency|v) \tag{2.13}$$

$$R_p(background) = 1 - Probability(saliency|v) \tag{2.14}$$

To make our model more robust we propose multi-scale in this part. Give an image with different prior knowledge of regions, we use different saliency value threshold. The final energy model would be:

$$E(A) = \frac{1}{K}\sum(\rho R(A) + B(A) + \omega) \tag{2.15}$$

where $\omega$ is a small value. In our experiment we use $\rho = [1.3, 1.4, 1.5]$.

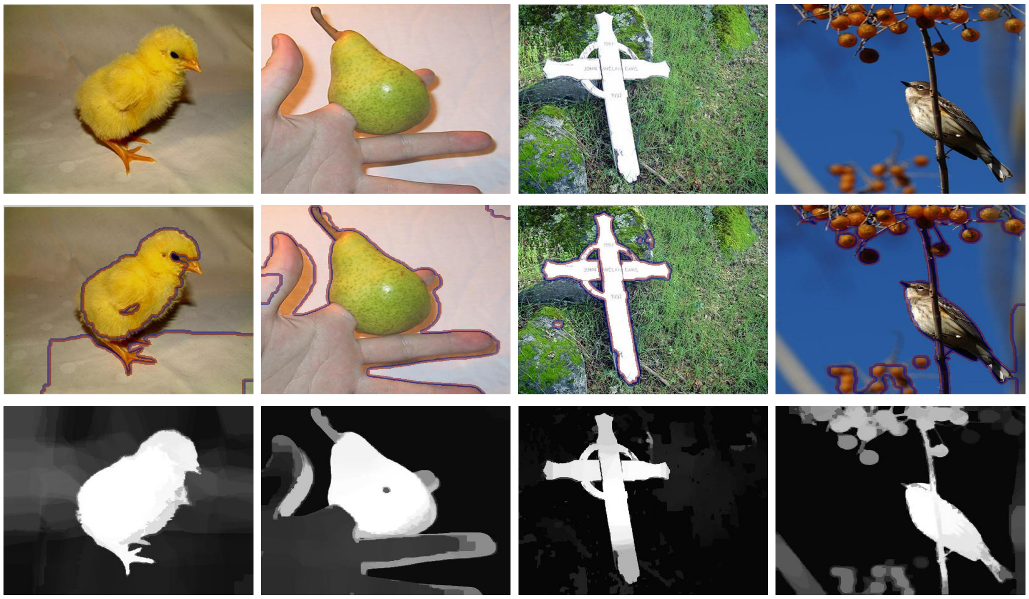


Figure 2.9: Image segmentation based on graph cut kmeans initialization and our proposed saliency map.

## 2.4 Experiment

This section presents both quantitative and qualitative evaluation on benchmark saliency object detection dataset: MSRA dataset which consists 20 categories up to 25000 images with manually marked bounding boxes. For better evaluation , Achata [40] creates 1000 binary object contour masks from this dataset. Our quantitative result is measured by ROC curve and time consuming, for qualitative result evaluation, we presents sample images and compared saliency map with nine state-of-art approaches.

**Segmentation by fixed thresholding.** For accurate evaluation, the easiest way is segment saliency map into binary images, we follow Achata's method with fixed threshold ranges from $[0,255]$ for a saliency map. For each threshold we compute *precision* and *recall* and draw the ROC curve for each method Fig 2.8. This method compares how well the saliency detection varies with different threshold in the image. The precision and recall is defined as:

$$precision = \frac{saliencymap \cap groundtruth}{saliencymap} \tag{2.16}$$

$$recall = \frac{saliencymap \cup groundtruth}{groundtruth} \tag{2.17}$$

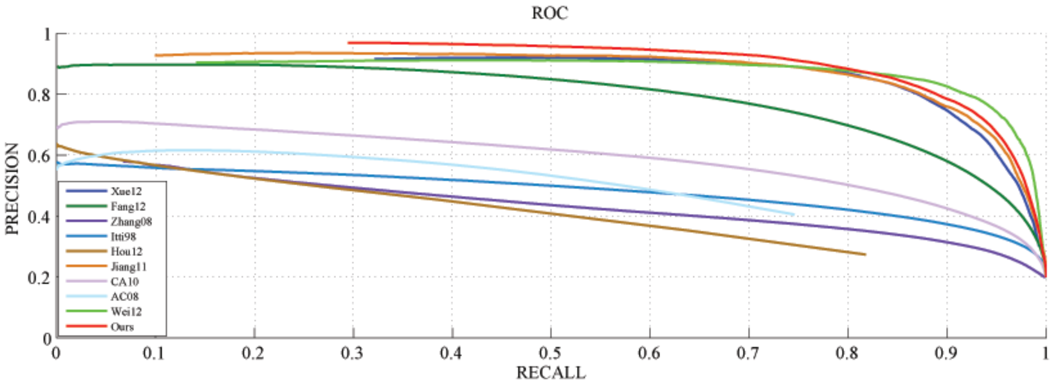


Figure 2.10: Precision recall curve for fixed thresholding, our proposed method is compare with nine existing method.

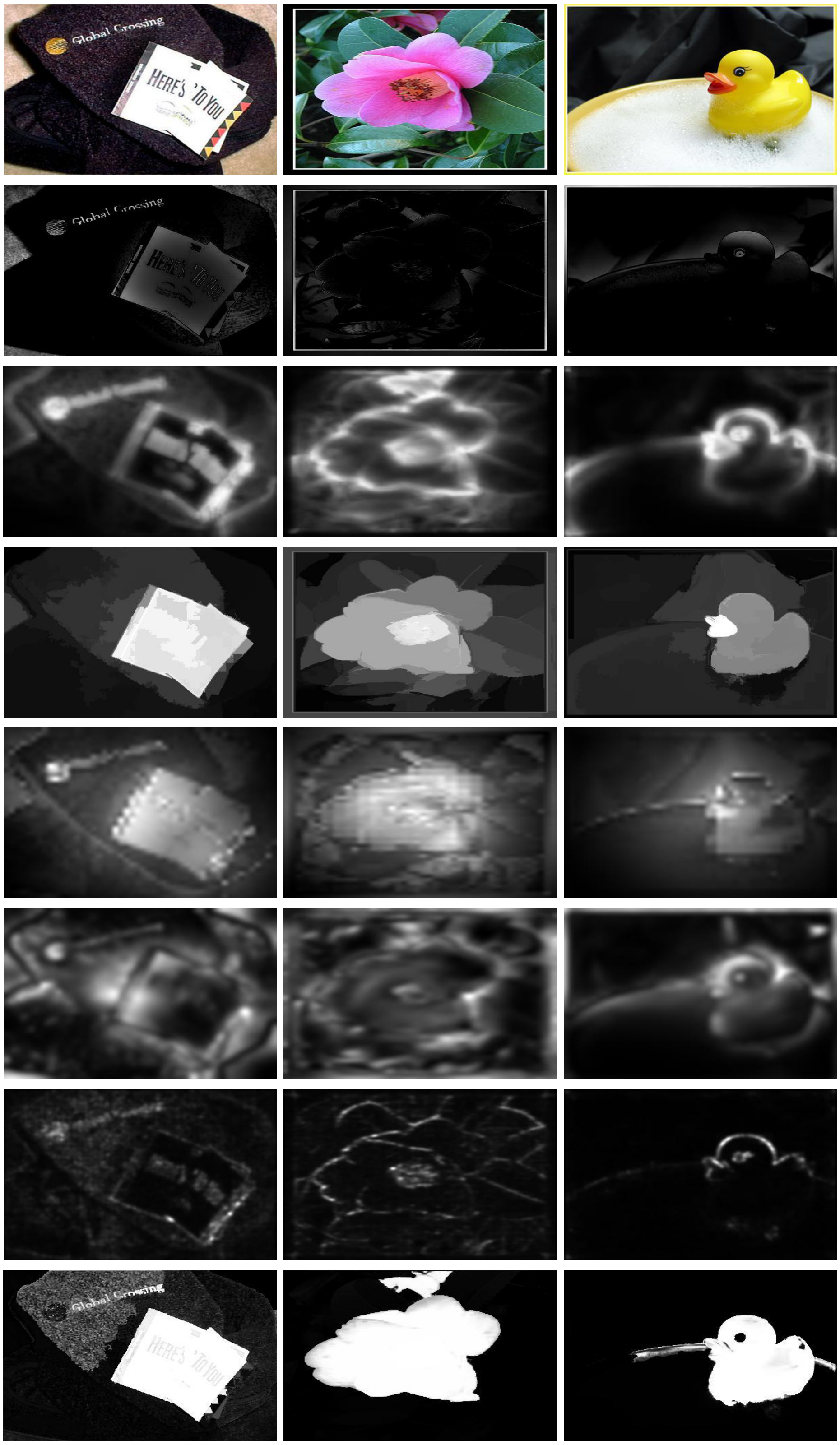
We select these nine method: xue12[29], fang12[41], itti98[18], hou12[21], jiang11[17], ac09[40], wei12[42] zhang08[28], ca12[16] according to: number of citation (itt98, hou12), recent reported best result (xue12, jiang12, wei12), various methods (itti98's motivation is from biologic mechanism, while ac10's is from fuzzy angle, hou12 compute saliency map in frequency domain, fang12 proposes a compress domain computational model for saliency detection, zhang08 presents an approach based on Bayesian inference ).

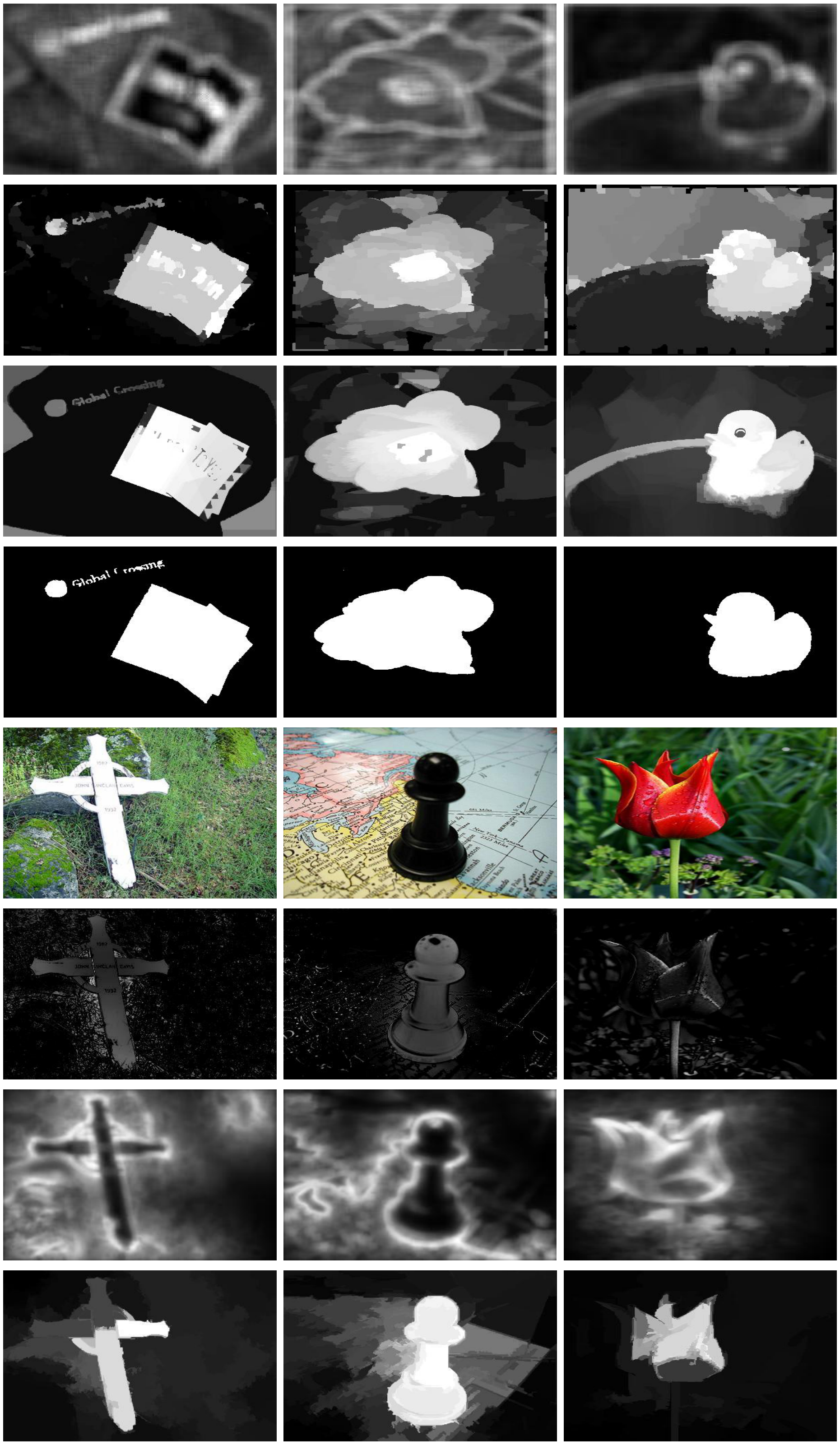
The ROC curve shows ours is comparable to wei's and superior to others' approaches. When Threshold $T = 255$, our method achieves about 30% of ground truth area while the result of rest of approaches are much lower. More importantly, our method can reach the converged precision value 95% at the recall 70%. For the time consuming , we show in Table 1. We implement our algorithm in Matlab and all others' code are download from their original authors' homepage and followed by their instructions. In order for a fair comparison we didn't rescale the image for itti98 and hou12 , all the programs are run on a same computer with Dual Core i7 2.4 GHz machine which with RAM 12GB. From the result we can see that our algorithm is comparable in terms of accuracy and complexity .

Table 2.1: Average time taken for each method to compute a single image in MSRA 1000 database.

| Method | [18] | [21] | [29] | [16] | [41] | [40] | [28] | ours |
|---|---|---|---|---|---|---|---|---|
| Code | matlab | matlab | matlab | matlab | matlab | c++ | matlab | matlab |
| Time(s) | 2.5 | 1.2 | 363.2 | 53.4 | 7.2 | 0.9 | 8.3 | 3.5 |

**Visual comparison** is taken from MSRA 1000 dataset. Fig 2.10 shows the results of different approaches, first row is input image, from second row to eleventh row is ac09[40], ca12[16], jiang11[17], fang12[41], itti98[18], hou12[21], xue12[29], zhang08[28], wei12[42], our proposed method with graph cut, and ground-truth. For different types of input image, proposed method is more robust than others. Ac09[40] fails in flower images, while ca12[16] is very sensitive to boundaries, which will cause much more noise. Jiang11[17], Xue12[29], Itti[18] can not segment object from its background well. For zhang08[28] and hou12[21] and fang12[41] only detect the saliency region and their result is not accurate. Wei12's[42] approach tests well on unified background, but for the images with more textures, e.g. grass image and chess image in Fig 2.9. While our proposed approach is much more robust in different type of images.


Global Crossing
HERE'S 'TO YOU



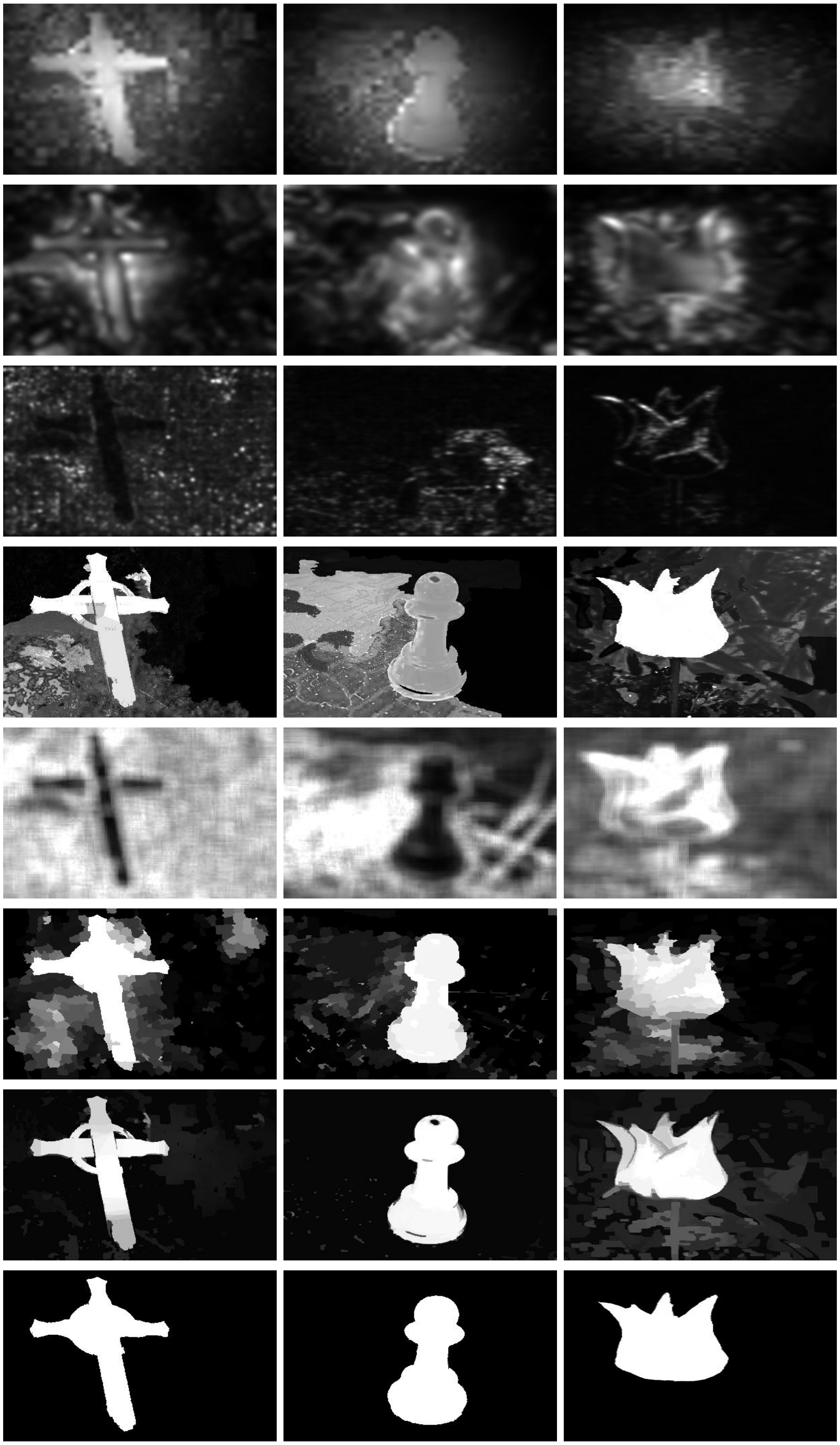


Figure 2.11: Visual comparison of nine existing approaches.

## 2.5 Conclusion

In this chapter, we proposed a context based model for saliency object detection, which computes from multi-scale contexts and global boundary information. The time complexity of our proposed method is $O(MNCK)$ where $M$ and $N$ is image height and width respectively, $C,K$ are the number of superpixels and scales. From Table 2.1, it is necessary to notice that our approach compute 5 different scales, but still comparable to others' computation time. Experimental result tested from the benchmark dataset demonstrates our approach is superior to existing state-of-art methods in terms of quantitative and qualitative evaluation.

# Chapter 3

## Motion Saliency Detection

In this chapter we propose a novel method for video analysis, based on the phase information of the video. In addition, multi visual cues and scales are fused into proposed method. Based on the saliency map computed using the proposed method, we demonstrate that two fundamental vision tasks can be benefited from proposed method, e.g. abnormal event detection and action recognition. In the experiment part, we demonstrates the quantitative and qualitative evaluation between our proposed method and state-of-art approaches.

The proposed method, compared with existing video saliency approaches, has several advantages. Firstly, it computes the saliency information from the whole video instead of adjacent frames, which is different from most of existing approaches in the literature. In the experiment we have shown that motion vector captured from two adjacent frames can not guarantee the accuracy of global motion trajectories, especially for the complex scenes. Secondly, the proposed approach is easy to implement and efficient. The time complexity of proposed method is only $O(NlnN)$, where $N$ is the size of input. Last but no least, compared to most of existing state-of-art approaches, our method is unsupervised which is more feasible for practical tasks, e.g. action recognition.

This chapter is organized as following: in section 1-2 we describe the proposed method including the analysis and the relationships between the existing methods; section 3 is the experiment part, which presents the comparison of proposed method with existing state of art approaches in quantitative and qualitative evaluation on two vision tasks; and the chapter is concluded in section 4.

## 3.1 Proposed method

Most of existing approaches are predicting saliency region in spatial domain, however, the visual information of human being s are processed over the time by vision system. Such that, it is necessary to consider temporal information for salient objects detection. A research in [43] found that, the objects will get more attraction if it is moving differently from its surroundings. To this end, we propose a method to compute the saliency map of dynamic scenes by utilizing the phase information of temporal domain. In the proposed method, we compute the saliency map for video data $\mathbf{X} \in \mathbb{R}^{M \times N \times T}$ as:

$$\mathbf{Z} = \left| \mathscr{F}^{-1} \left( \frac{\mathbf{Y}}{|\mathbf{Y}|} \right) \right|^2 \tag{3.1}$$

where $\mathbf{Y} = \mathscr{F}(\mathbf{X})$, $\mathscr{F}$ is 3D discrete Fourier transform and $\mathscr{F}^{-1}$ is the corresponding inverse transform. After we get the saliency map, we smooth it with a 3D Gaussian smooth filter. The 3D Fourier transform can be computed as:

$$\begin{aligned} \mathbf{Y}(u,v,w) &= \sum_t \sum_i \sum_j \mathbf{X}(i,j,t) e^{-i2\pi\left(\frac{ui}{M}+\frac{vj}{N}+\frac{wt}{T}\right)} \\ &= \sum_t e^{-i2\pi\frac{wt}{T}} \sum_i \sum_j \mathbf{X}(i,j,t) e^{-i2\pi\left(\frac{ui}{M}+\frac{vj}{N}\right)} \end{aligned} \tag{3.2}$$

i.e., the 3D Fourier transform can be computed as 1D Fourier transforms respectively.

We refer the proposed method as video motion saliency. This saliency was already used in existing works, e.g., [31], which computed the video saliency based on color information in spatial domain and motion vectors from adjacent frames. As a result, this temporal information based on two adjacent frames yield to complex scenes. Instead, the video motion saliency proposed in this chapter which considers whole the temporal span is more robust.

The method in Eqn. 3.1 evaluates the saliency region by exploring the information of the whole video. However in practical work, we may also be interested in detecting the saliency region it terms of a small period. For example, if a video contains multiple sessions, where each session is captured over activities of different salient regions, then we may be more interested in analysis the saliency within each session instead of the whole video. We can divided the videos into multiple sessions, however the session size is usually unknown. To this end, we propose multi-scale analysis for motion saliency. We first apply the window function to the input signal $\mathbf{X} \cdot \mathbf{w}(i,j,t)$, where $\cdot$ is the element-wise multiplication and $\mathbf{w}(i,j,t)$ the window function centered at position $(i,j,t)$, which is nonzero for only a small support (i.e., the size of window function). The saliency map is computed for the windowed signal:

$$\mathbf{Y} = \mathscr{F}[\mathbf{X} \cdot \mathbf{w}(i,j,t)] \tag{3.3}$$

$$\mathbf{Z}(i,j,t) = \mathscr{F}^{-1}\left[\frac{\mathbf{Y}}{|\mathbf{Y}|}\right] \tag{3.4}$$

The video motion saliency is computed by sliding the window function through the whole video. For different window size it has different meaning: for a larger window size, the saliency value reflects more global motion cues and more background scenes suppressed, while for a smaller window size, saliency value is affected by the local information and contains more background information. Either temporal and spatial domain can be applied with window function by our proposed method, where saliency varies with different scales.

In [44], the author demonstrated it is important for video saliency to combine different visual cues. In their model, both of color features and motion features are combined together. Similarly, if there are two visual cues, we could utilize the complex Fourier transform, where the two visual channels are encoded into the real component and image component of input:

$$\mathbf{X} = \mathbf{I} + i\mathbf{V} \tag{3.5}$$

where **I** and **V** are the two visual channels, e.g., intensity and motion magnitude.

Even though, Hypercomplex Fourier transform (HFT) has be used to combine multiple feature channels. In practice, the HFT is very hard to implement and time consuming. In our experiments we run 1000 simulations and in each simulation we generate a $r \times c \times 4$ array, where r and c is a random number between $[1,1000]$ and 4 is the number of feature channels. We compute the saliency map with different methods then measures their similarities via cross-correlation, where 0.91 is reported for QFT and FFT. After smoothing the saliency map with a Gauss kernel, the correlation is over 0.998. For natural image, we could expect an even higher correlation. The result shows: if a data with multiple feature channels, the correlation of saliency computed by Hypercomplex Fourier Transform and Fast Fourier Transform is very high (0.998), Thus, for saliency computation the results of combination each channel feature which calculated independently approximates the result of using Hypercomplex Fourier Transform. In our experiment, we use summation to combine all the feature channel, note that, other fusion scheme can also be used other than adding.

Finally, we summarize the proposed algorithm below:

---

```
Algorithm
Input: data $\mathbf{X}$, Gauss filter $g$, window function $\mathbf{w}$
Output: saliency map $\mathbf{Z}$
For each window location
    For each feature channel
        Apply $\mathbf{w}$ to the input $\mathbf{X}$;
        Compute Fourier coefficient $\mathbf{Y} = \mathscr{F}[\mathbf{X}]$;
        Extract the phase information $\hat{\mathbf{Y}} = \frac{\mathbf{Y}}{|\mathbf{Y}|}$;
        Do the inverse transform $\mathbf{Z} = \left|\mathscr{F}^{-1}[\hat{\mathbf{Y}}]\right|^2$;
        Smooth saliency map $\mathbf{Z} = \mathbf{Z} * g$;
    End
    Combine the $\mathbf{Z}$ of all channels together;
End
```

where **w** is optional and $*$ is the convolution.

## 3.2 Analysis

In the literature, abnormal means being deviating from what is normal or usual. [45] found that the abnormal events easily draw people's attention, i.e., they are salient. Such that the proposed motion saliency map can be used to detect abnormality in the videos.

To demonstrate the reason why our proposed method can detect the abnormal event, we employ two abnormal datasets: UMN abnormal dataset, UCSD dataset, and compute magnitude spectrum along the temporal domain. It computed by summing the magnitude spectrum in spatial domain. Figure 3,13.1 shows the result from one example where spectrum roughly follows the $\frac{1}{f^a}$ distribution, where $a$ is related to the slope of the curve. Meanwhile, our proposed method can be viewed as a high pass filter for it sets all magnitude to ones.

According to [46], a low pass filter and a band pass filter can model the mechanism of visual system to detect event. Especially, the band pass filter will dominate this mechanism when the interest signal frequency goes higher. For abnormal event detection, the magnitude of hihg frequency of abnormal event is higher than the background, thus the abnormal event can be detected by our pro-

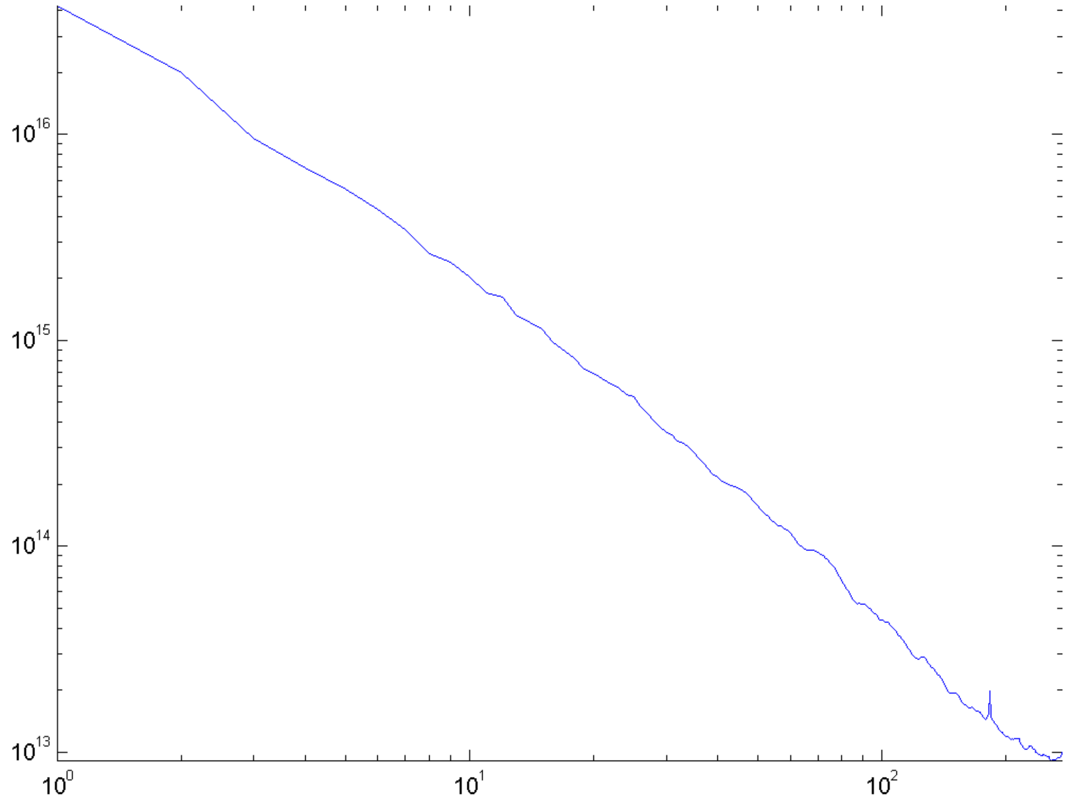


Figure 3.1: The magnitude spectrum along the temporal direction of one video from UMN abnormal dataset.

posed method which will suppress the lower frequency component.

Aside from abnormal event detection, our proposed method is also related to existing works in terms of temporal information computation. Currently, most of the existing approaches compute the temporal cues in two directions, one is based on the difference of adjacent frames, another is spatiotemporal cuboid which often needs large labeled training set. Compared with these approaches, our proposed method does not need any training stage and prior information. In addition, the sliding window can control the saliency in terms of local salient and global salient. At last, the time complexity of our proposed method for a data $X \in \mathbb{R}^{M \times N \times T}$ is $O(KMNT\log(MNT))$, where $K$ is the number of feature channels. Compared to current approaches, our method is polynomial time.

## 3.3 Experimental Result

In this section, we demonstrate that three fundamental vision tasks can be benefited from our proposed motion saliency detection. i.e., motion detection, abnormality detection and action recognition. The proposed algorithms are evaluated on several benchmark datasets, e.g. UMN abnormal dataset, UCSD dataset, Wezimann dataset, KTH dataset and UCF sports action dataset, with the comparisons to several state-of-arts methods.

### *3.3.1 Motion and Simulation Experiment*

Motion detection is essential vision task for many applications, e.g., object tracking, object recognition. According to the analysis in last section, the proposed method will highlight the moving objects and suppress the background, thus we can use the saliency map for foreground object motion detection. Such that, we carry out the following simulation experiment. With the dynamic background (Fig. 3.2) which is generated by two images with complex texture, the foreground objects have have uniformed appearance and their motion trajectories are defined as:

$$\Gamma_1(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} 128 + 64cos(\frac{\pi t}{32}) \\ 128 + 64sin(\frac{\pi t}{32}) \end{bmatrix} \tag{3.6}$$

$$\Gamma_2(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} 64 + 32cos(\frac{\pi t}{32}) + \varepsilon \\ 64 + 32sin(\frac{\pi t}{32}) + \varepsilon \end{bmatrix} \tag{3.7}$$

where $\varepsilon$ is a random variable between $[0,128]$. The object with trajectory $\Gamma_1$ can be regarded as moving regularly, While another is not. In addition, we set the Gaussian filter to smooth the result with standard deviation is $0.006\sqrt{N^2+M^2}$ and filter size with $1+6\sigma$, where $N \times M$ is the size of each frame.

In Fig. 3.3, we demonstrate the result, where the red circle indicates the trajectory $\Gamma_1$; top: some sample frames from the input video, middle: the saliency map computed with the proposed method, the comparison to the results of the method proposed in [31] (bottom). From the result, we can find the proposed method highlight the moving objects with random motion trajectory (i.e., with trajectory $\Gamma_2$) and suppress the background texture and another moving object which moving in a circle (the one with trajectory $\Gamma_1$). However, the method of [31] fails to distinguish the foreground object from background, besides, it can not detect the motion saliency in terms of motion trajectory (moving irregularly). A probably answer for the result of [31] is the simple calculated adjacent frames can not keep the motion coherence in terms of fast moving object.

For real data, we apply the proposed method to the input video; we then binarize the output saliency map to segment out the moving objects. From the experiment, we found that $4\mu \leq \rho \leq 8\mu$ is a good choice for the binarization threshold, where $\mu$ is the mean of the saliency map

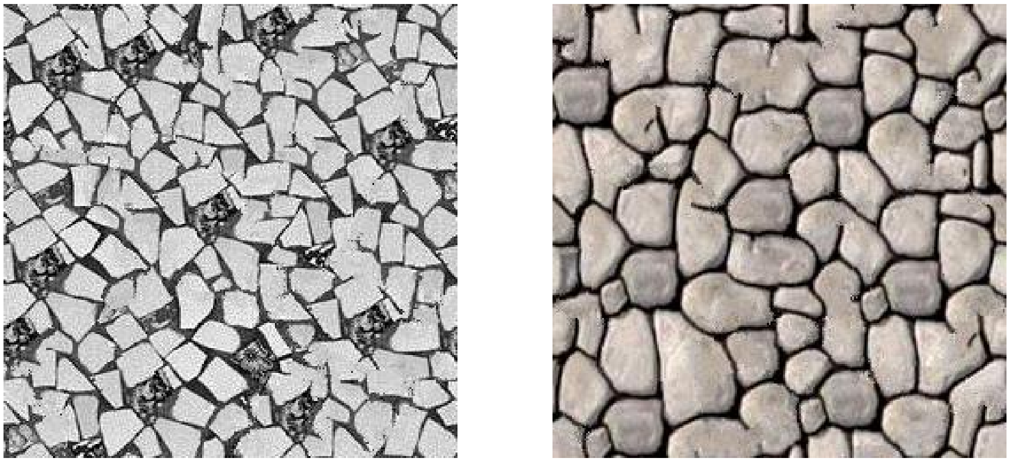

(a) (b)

Figure 3.2: The dynamic background is a mixture the two images.

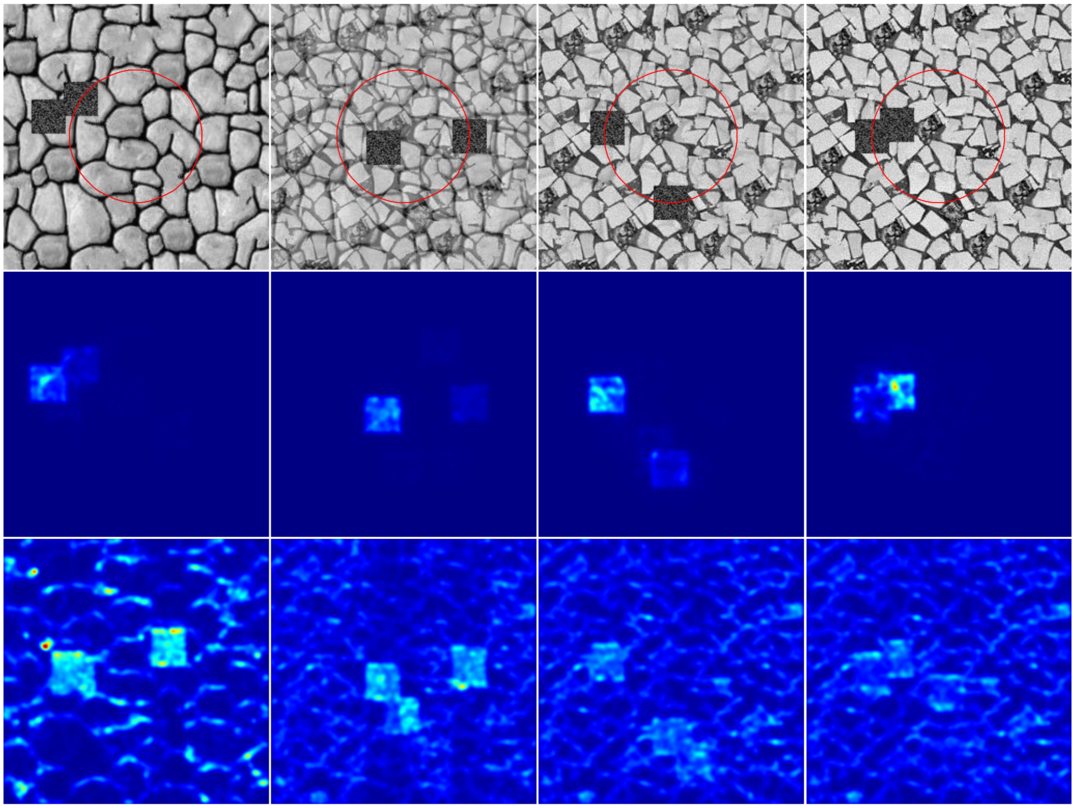


Figure 3.3: Example of motion detection.

### 3.3.2 Abnormal Event Detection

For abnormal event detection, the abnormality can be described as the region which has higher value in the saliency map from our proposed method. In the experiment, we apply simply thresholding for the frames and the value higher than the threshold may be the frame occurs abnormal event, the threshold in our experiment used as mean score of all the frames and score for each frame is defined as"

$$\mathbf{s}(t) = \sum_i \sum_j \mathbf{X}(i, j, t) \tag{3.8}$$

where $\mathbf{s}(t)$ is the saliency score of $t_{th}$ frame, $i$, $j$, $t$ are row, column and frame index of the 3D saliency map accordingly. The frame with high saliency score would contain abnormality.

Table 3.1: The result on UMN dataset.

| Method | AUC |
|---|---|
| Optical flow [48] | 0.84 |
| Social force [48] | 0.96 |
| Chaotic invariants | 0.99 |
| NN [49] | 0.93 |
| Sparse reconstruction [49] | 0.978 |
| Interaction force | 0.9961 |
| Proposed | 0.9378 |

We evaluate the proposed method for abnormal event detection on two datasets UMN abnormal dataset( as shown in Figure 3.5, the top is the saliency value (Y-axis) for each frame (X-axis) and bottom are sample frames picked from different frames ) and UCSD dataset [47] and compared with six existing approaches, especially some of the approaches are supervised, e.g., social force [48], sparse reconstruction [49], MPPCA [50], MDT [47], while ours dose not need any training data.

Table 3.1 shows the result of different method. In addition, we shows the ROC curve with frame-level true positive rate and false positive rat. Figure 3.5 shows our result of three scenes, where we point out the saliency value for each sample frame . The result on UCSD dataset is shown in Tab. 3.2, where we report frame-level equal-error rate (EER) [47]( the lower the better) . Figure 3.6 shows the ROC for UCSD dataset with the proposed method; Figure 3.7 shows our result of eight sample frames. From the result we have shown, although our result is unsupervised, we still outperform some state-of-art method.

### *3.3.3 Action Analysis*

Human beings select the region where draws their most attestation for further processing, in this section, we demonstrate that our proposed method can be used as a

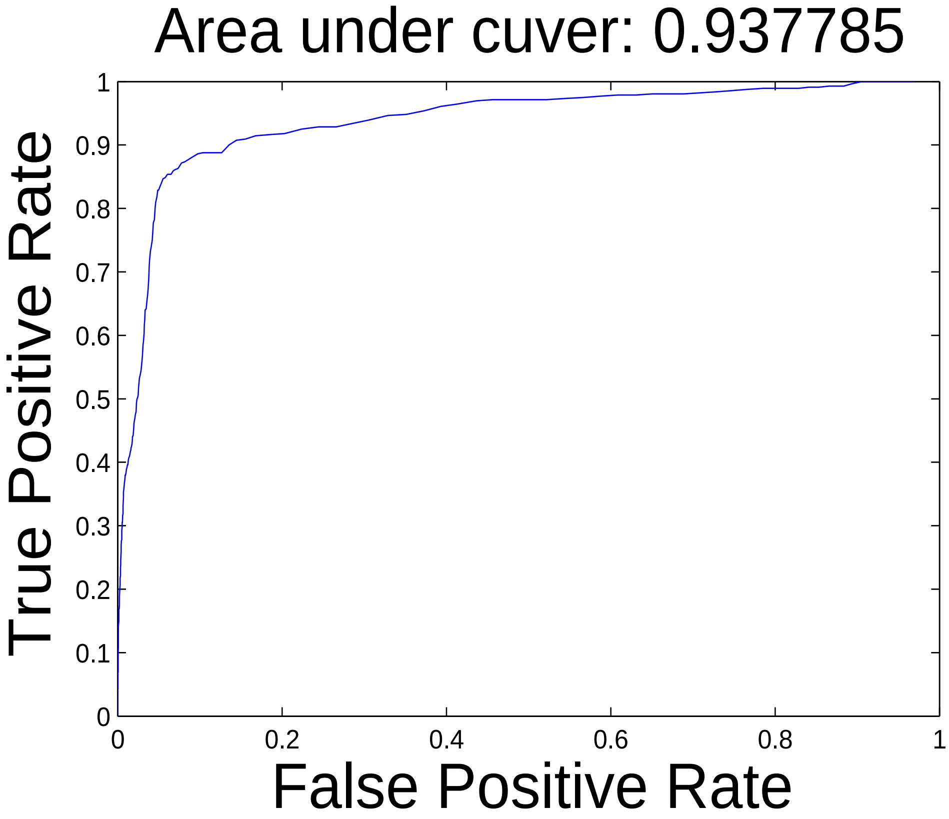


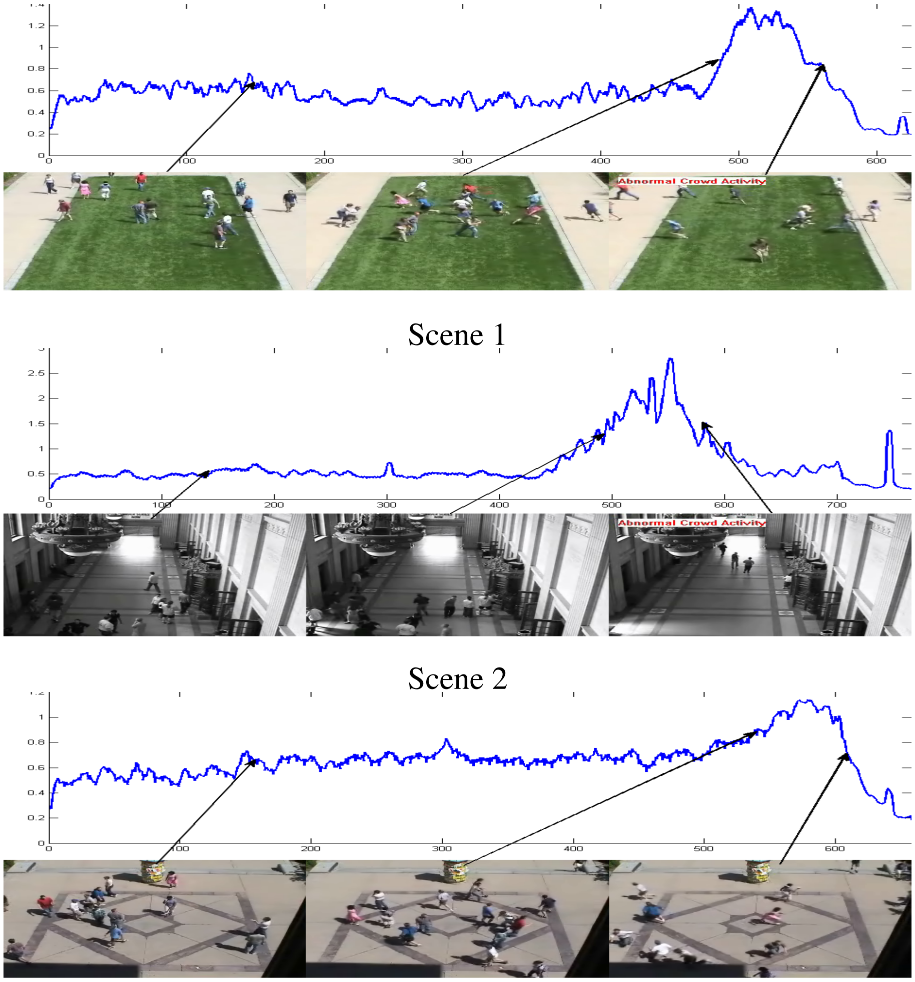
Figure 3.4: The ROC for the UMN dataset computed with the propose method.

Table 3.2: The frame level EER for UCSD dataset.

| Method | Ped1 | Ped2 | Overall |
|---|---|---|---|
| Social force [48] | 31% | 42% | 37% |
| MPPCA [50] | 40% | 30% | 35% |
| MDT [47] | 25% | 25% | 25% |
| Adam | 38% | 42% | 40% |
| Reddy | 22.5% | 20% | 21.25% |
| Sparse [49] | 19% | *N.A.* | *N.A.* |
| Proposed | 27% | 19% | 23% |

saliency interest point detector by sampling interest points on our proposed saliency map.

At beginning, we compute the saliency map $\mathbf{Z}$ for the input data $\mathbf{X}$. Then the interest points are sampled based on non-maximum suppression: an interest point



Scene 1

Scene 2

Scene 3

Figure 3.5: Some sample results for the UMN datasets.

$(x,y,t)$ is defined as:

$$\begin{aligned} \mathbf{Z}(x,y,t) &\geq \rho \\ \mathbf{Z}(x,y,t) &\geq \mathbf{Z}(i,j,k)\ \forall (i,j,k) \in N(x,y,t) \end{aligned} \tag{3.9}$$

where $\rho$ is a predefined threshold (e.g., $2\mu$) and $N(x,y,t)$ is the set of positions near $(x,y,t)$.

Each interest point $(x,y,t)$ is assigned a descriptor $(x,y,t,\sigma,\tau)$ extracted from its four connected neighborhood area, where $\sigma$, $\tau$ are the spatial and temporal scales respectively. The descriptor is computed as following: each neighborhood is

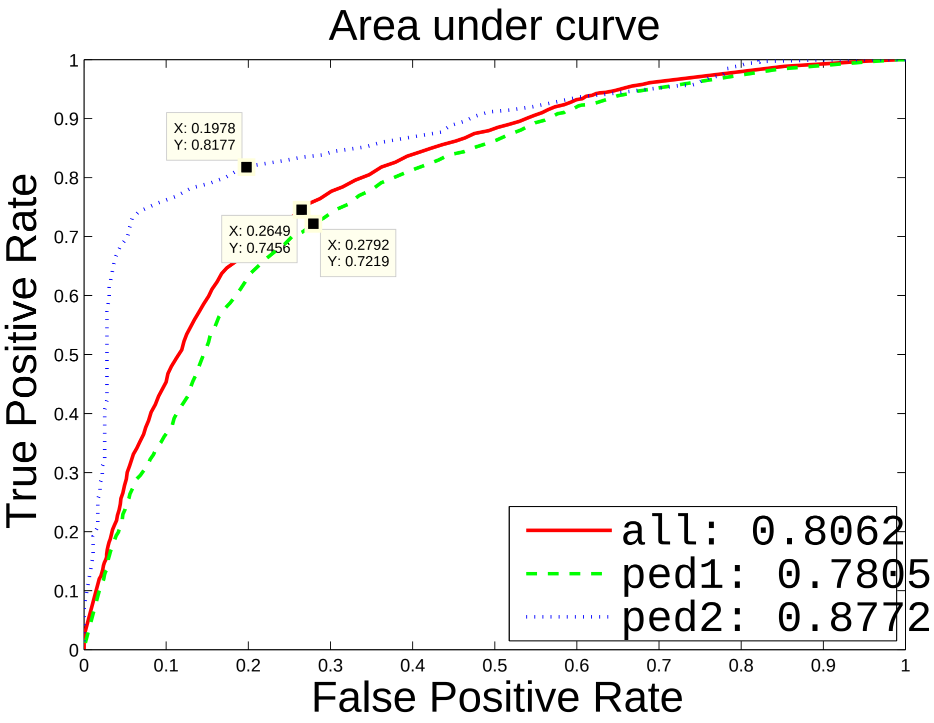
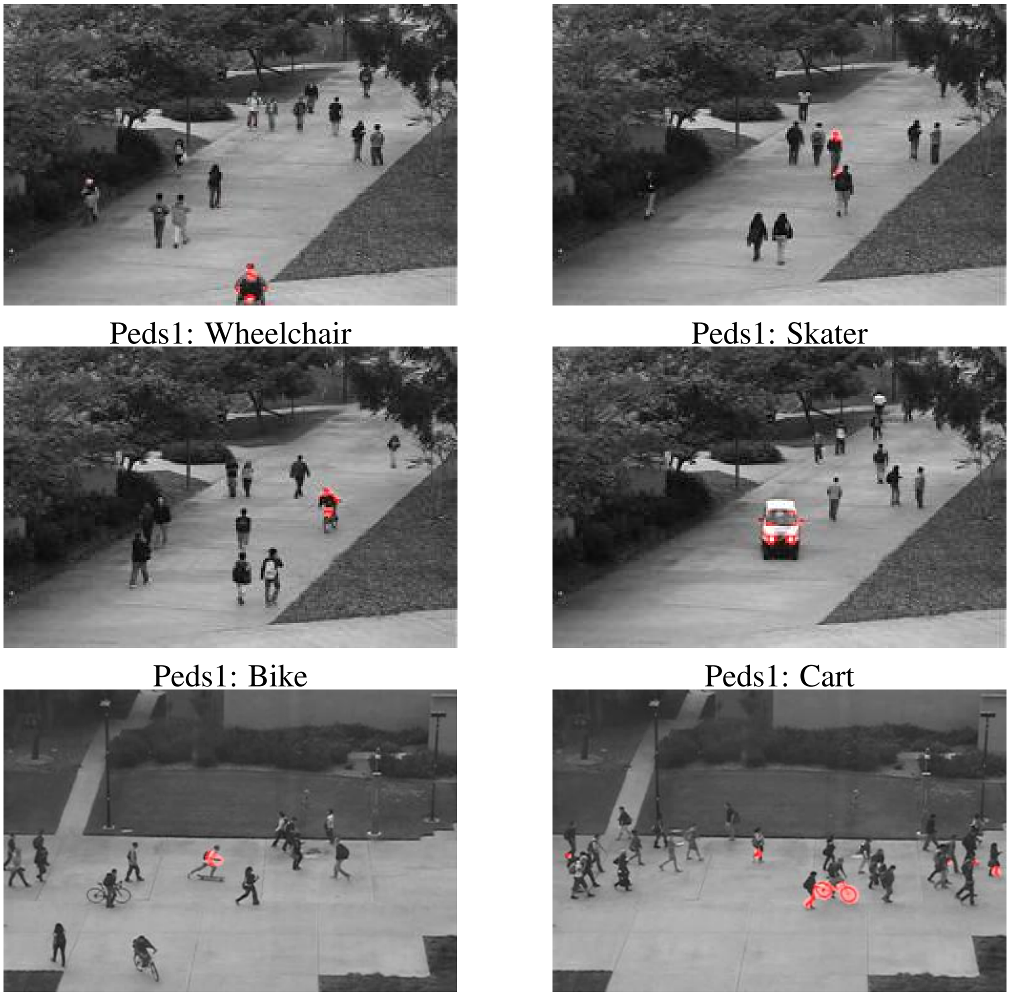


Figure 3.6: The ROC for the UCSD dataset computed with the propose method.

divided into subblocks(e.g., $3 \times 3 \times 2$ along spatial and temporal direction accordingly); for each subblock, we computed its gradient histogram by quantizing the orientation of the subblock 3D gradient $g = [g_x, g_y, g_t]$. At last, we normalize each subblock and combine them into one histogram as a descriptor.

The interest points (e.g., [51]) have been received a lot of interest, since the popularity of "bag of words" for action recognition. We evaluate the proposed method on detecting motion interest points. For quantitative result comparison, we employ three datasets: Weizmann dataset [52], KTH dataset [53] and UCF sports dataset [3]. Since the method is proposed for detecting interest points, we only compare it with several state-of-art interest point detectors: Harris3D [51], Gabor [54], Hessian3D [55]. All the codes are download from author's official homepage and followed the instructions.

Figure 3.8 shows sample frames of videos from UCF sports action dataset



Peds1: Wheelchair

Peds1: Skater

Peds1: Bike

Peds1: Cart

Peds2: Skater

Peds2: Bike

Figure 3.7: Some sample results for the UCSD datasets.

and KTH dataset and corresponding saliency map respectively. From the figure, we can see that most of background are suppressed and moving regions are highlighted. More importantly, the proposed method is also robust in terms of background moving, cluster background and scale variation. In addition, we can find that our saliency interest points detector will be mostly sampled from those moving parts , e.g., hands, gets higher saliency value (red color) then other static parts.

For the proposed method, we use the following parameters. Same as [51], the neighborhood of interest point is divided into $3 \times 3 \times 2$ equal-sized blocks, where for each block we compute the histogram of gradient/optical flow. We also

Table 3.3: The performances of different detectors on three datasets.

| Method | Weizmann | KTH | UCF sports |
|---|---|---|---|
| Harris3D | 85.6% | 91.8% | 78.1% |
| Gabor | N.A. | 88.7% | 77.7% |
| Hessian3D | N.A. | 88.7% | 79.3% |
| Dense | N.A. | 86.1% | 81.6% |
| Proposed | 84.5% | 88.0% | 86.7% |
| Proposed* | 89.3% | 92.5% | 82.4% |

apply multiscale scheme, where size of neighbor of each interest point is $18 \times 18 \times 10$, $25 \times 25 \times 14$ and $36 \times 36 \times 20$.

For the interest point descriptor, we employ both histogram of gradient (HoG) and histogram of optical flow (HoF) and the video is described by bag of words. The codebook is represented by the histogram and SVM is used as classifier. The size of of codebook is $k = 2000$, for SVM we use $\chi^2$ kernel, where $C = 100$. For Weizmann dataset and UCF sports dataset, we use leave-one-out scheme for training and testing; for KTH dataset. Tab. 3.3 reports the performances of different detectors on these three dataset, where we test extracting feature on the original video and also extracting feature on the saliency map of the original video (refer as "proposed*"). From the table we find that, the proposed method (and "proposed*") achieves the best result over all three datasets. Especially "proposed*" achieved the best results for KTH dataset and Weizmann data; "proposed" achieved the best results for UCF sports action dataset.

## 3.4 Conclusion and Discussion

In this chapter, we proposed a novel approach for detecting video motion saliency, which is easy to implement and computationally efficient. Inspired by recent development of visual saliency approaches based on spectrum analysis, we extracting the phase information for video saliency computation. In addition saliency in image has

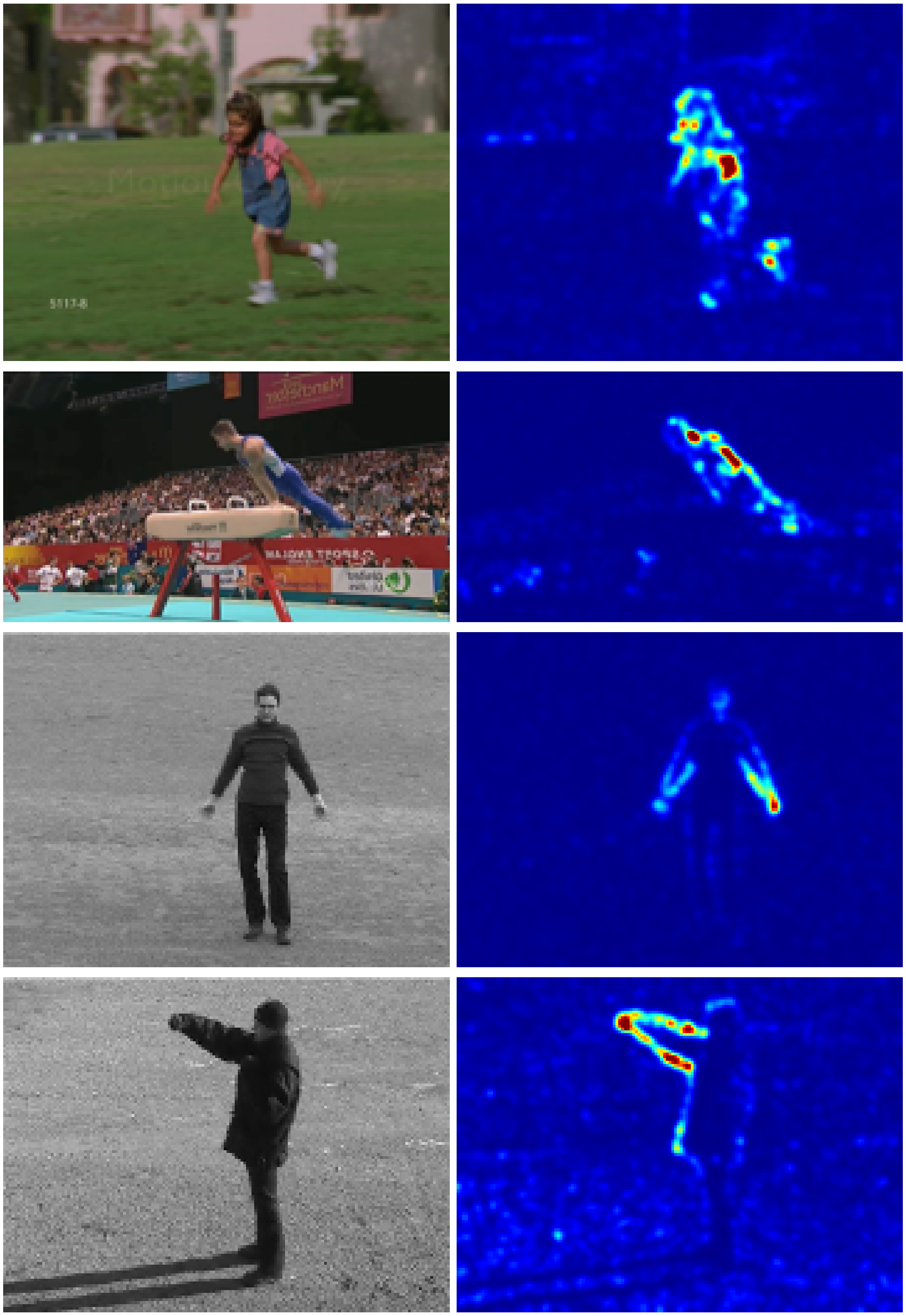


Figure 3.8: Some samples frames (left) from UCF sports action dataset (Row 1, 2) and KTH dataset (Row 3, 4) with their saliency maps (right).

been applied in more and more vision tasks recently, e.g., object detection, image classification. A natural question arises: whether saliency in video is also helpful to key vision tasks. Considering this, we designed algorithms and performed experiments for applying saliency in video in abnormality detection (Sec. 3.2) and action recognition (Sec. 3.3). The experiment results indicate that video saliency can be used to facilitate these visions tasks.

# Chapter 4

# Saliency Cut

Video object segmentation has been extensively researched in recent years. However, it is still a challenging problem in vision area. In Chapter 1, we briefly introduced the problem background, in this chapter we focus on the related automatic approaches in the literature. Currently, there are two predominant approaches to VOS problem, one is tracking interest point through motion trajectory [13, 55], another is clustering pixels from all the frames[25, 56]. However, both of them have disadvantages in some aspects. In first approach, interest points are computed from some ”key points” whose motion trajectory has higher correlation than other points. However, it is difficult to guarantee the spatial coherence of the object shapes. Therefore, the quality of object segmentation yields to the location of the points. While for the methods based on pixel clustering from three dimensions of the video, e.g. Gaussian Mixture Model [5], Key frame ranking [25], Graph cut [6] , they are infeasible, first, it is difficult to identify the foreground objects [6], second, these approaches need user input, e.g., annotate the object region in the first frame [5]. Recently, [25] has attracted lots of interest, the author proposes a contour matching algorithm which leads to segment the foreground object with highest score. However, this approach does not predict the location of foreground object in adjacent frames, which fails to detect the fast moving object.

In this chapter, we propose a novel video object segmentation algorithm based on saliency features, as we described in the previous two chapters, the proposed saliency object detection algorithm can be used as a generic object class detector while spatial temporal detector can help analyze the motion and actions. Motivated by graph cut [8] which is one of most well known algorithms for image segmentation, we combine it with our saliency methods and introduce a new automatic video segmentation method which we call saliency cut. Compared to existing approaches, our saliency cut has two advantages: first, we model the foreground appearance in a saliency angle, second, our algorithm is efficient and easy implement which is more feasible to practical problem, e.g. object based encoding.

More specifically, our model is defined as following: for a input video, we load the whole video and extract the motion saliency feature as proposed method described in Chapter 3. For each frame, all the pixels $P = \{p_1, p_2, p_3, ...\}$ are assigned distance score $S(P)$ by measuring the spatial distance from its location to motion boundary. Besides, we also compute the object saliency $Sal(P)$ based on graph based manifold ranking and proposed method described in Chapter 2 for saliency object detection in spatial domain. Intuitively, the object segmentation in each frame is formulated as a saliency energy minimization problem based on. For $S(P)$ we consider it as a pair-wise interaction term in the graph, which influences the segmentation result based on the motion boundary . For unary term, we measure the foreground object energy by ranking the saliency values.

The chapter is organized as following: in section 1, we describe our proposed method and include analysis, section 2 presents the experimental result of ours and comparison with three state-of-art approaches. And we conclude this chapter in section 3.

## 4.1 Proposed Method

In Chapter 2, we observe that the an object can be measured by saliency values. In this section, to further extract the object's intrinsic appearance structure, we proposed a graph based manifold ranking algorithm to extract the foreground object based on our precomputed saliency map. The reason why we need rank our saliency map is, compared to images, in a video with dynamic scene, the foreground's saliency value will be suppressed by background changing or noise. Moreover,

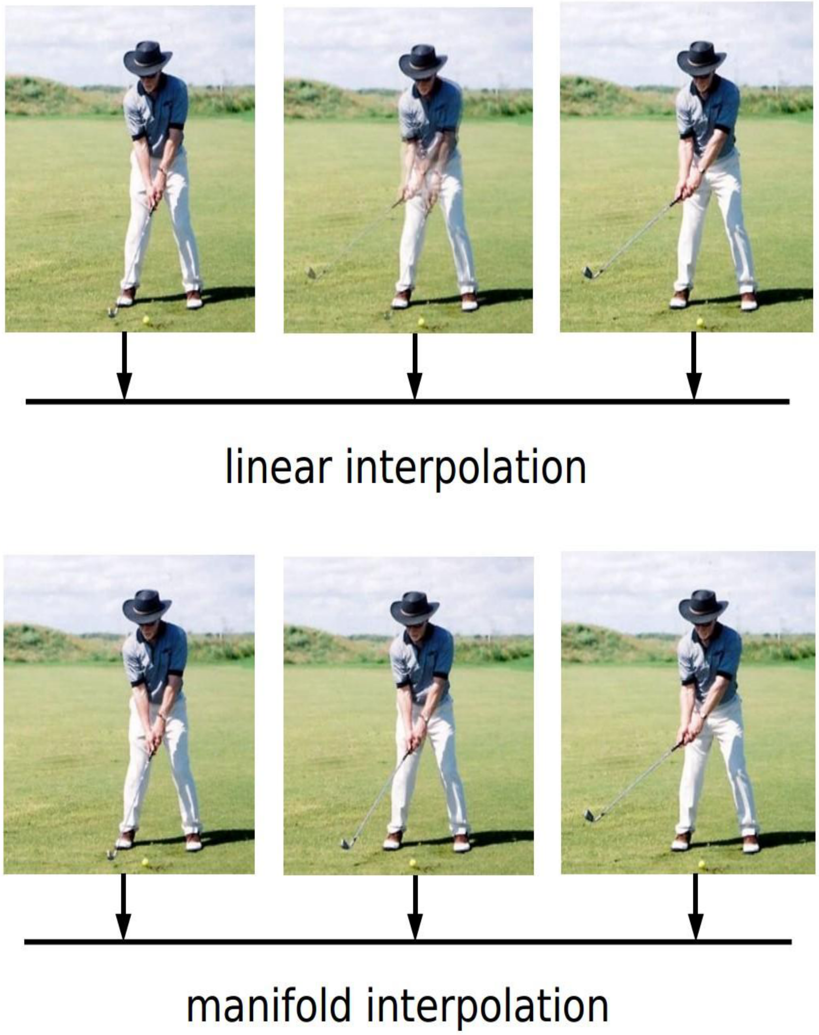


Figure 4.1: Difference between linear interpolation and manifold interpolation.

the manifold learning is a nonlinear dimension reduction technique and it's a better way for image data distance metric (Figure 4.2) [57]. Saliency value computed from one frame or spatial domain can not satisfy the motion coherence. If we simply estimate the motion value by linear interpolation, the result will not be accurate (Figure 4.1). Such that, it is necessary to learn from adjacent frames and rank the saliency region based on their relevance, such that, ranking the object regions will help us to achieve a better result.

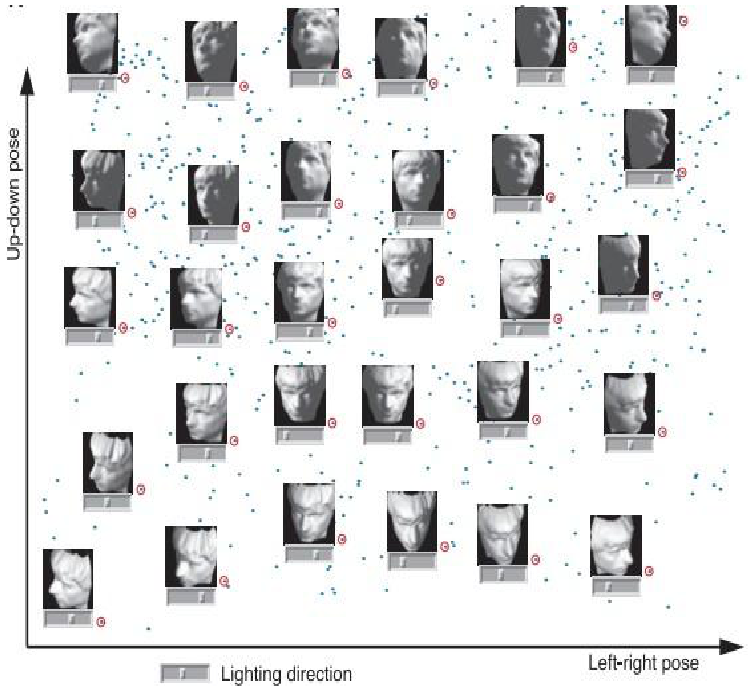


Figure 4.2: Nonlinear dimensional reduction via manifold [9].

### *4.1.1 Ranking the Saliency Data*

The Google is well known for its web page ranking algorithm called **PageRank**, which employs the global hyper-links of the web. Recently, the idea of ranking the data has been successful applied into image retrieval [58], video classification [59] and other multimedia communications. In [57], the author proposed a ranking algorithm which based on the intrinsic manifold data structures. Given a set of data $X \in \left\{x_1, x_2, ...x_q, x_q+1..x_n\right\} \in R^m$, we can label some points as queries and leave rest points for ranking based on the relevance to the query labels. The ranking function can be written as $X \rightarrow R$ , each data point $x_i$ will be assigned with a ranking value $f_i$, also the function $f$ can be represents as a vector $f = [f_1, f_2, f_3...]^T$. Let $y$ be a label vector, which means if $x_i$ is query then $y_i = 1$ otherwise $y_i = 0$. Then affinity matrix $W$ will represent the edge weight between any two data points. Thus the ranking function can be written as :

$$f^* = (I - \alpha S)^{-1} y. \tag{4.1}$$

Where $I$ is Identical matrix, $\alpha$ is a parameter from zero to one, it smooths the neighborhoods data points and initialize the ranking score. $S$ is a symmetric matrix defined as $S = D^{-1/2} W D{-}1/2$ in which $D$ is the diagonal matrix with diagonal element $i$ equals to the summation of $i_{th}$ row of $W$.

In order to find the a more accurate location of foreground, we proposed a ranking measurement for object detection. Given a graph $G =< V, E >$, $V$ is a nodes set which represents data points and $E$ is a set of edges which denotes the weight between two nodes. Like chapter 1, we employ a superpixel to represent a graph node. As neighborhood nodes, we change the definition, given a node, its neighbors are the nodes connected it and nodes which share the boundaries with its neighborhood. By this connection expansion, the result will exploit the spatial appearance more. Then weight function is defined as :

$$w_{ij} = e^{-|I_i - I_j|} \tag{4.2}$$

where $i, j \in V$, and $I_i$ denotes the mean CIELab value of a superpixel region. In conventional ranking problem, the queries are usually labeled by the ground truth. In proposed method, we view the query as our proposed saliency regions. At beginning, we binarize our saliency map by its mean value, those regions whose value larger than the threshold will be the foreground and rest are background. As foreground query is given, the label set $y$ will make our ranking function to find the regions which have higher correlation to our foreground. Finally, the final saliency map would be:

$$Sal_i(k) = f_i^*(k) \tag{4.3}$$

where $i$ is video frames, $k$ is superpixel node in previous frame foreground region and its neighborhood nodes. Which means , the foreground region should near the foreground region in previous frame, this location constraint significantly improved the performance of our method for it naturally captures the motion coherence.

### *4.1.2 Extract Motion Constrain*

In chapter 2, we have introduced the background of graph cut algorithm, however, for video object segmentation we need consider both spatial coherence and temporal coherence. To this end, we first compute motion boundaries from saliency map which computed by our proposed spatiotemporal saliency detector. The boundary detector which we use is classic Canny operator. After that, motion score of each pixel is calculated by distance transform.

Distance transform is an important image processing technique in computer vision area, it has many applications, e.g. line detection, corner detection. A general method was proposed in [10]. For an image $I$ with height$H$ and width$W$. The two dimensional grid $G$ is defined as $G = 0,1,2...H-1 \times 0,1,2...W-1$ and $f$ is an arbitrary function, the distance transform of $f$ under the Euclidean distance is:

$$D_f(x,y) = min((x-x^{'})^2 + (y-y^{'})^2 + f(x^{'},y^{'})) \tag{4.4}$$

Specifically, in binary image, distance transform represents the distance of any pixel to its nearest non-zero pixel. To this end, we normalize the pixel score and get the final motion score $S(P)$ for each pixel :

$$S(p) = 1 - exp(D_f(p)) \tag{4.5}$$

where $D_f(P)$ represents the motion boundary map.

### *4.1.3 Energy Function for Video Object Segmentation*

Given a video sequence $S = \{s_1, s_2, ...\}$, object saliency map for each frame $Sal = \{Sal_1, Sal_2, ....\}$ and motion constrains $S_p = \left\{S_p(s_1), S_p(s_2), ...\right\}$. The video object

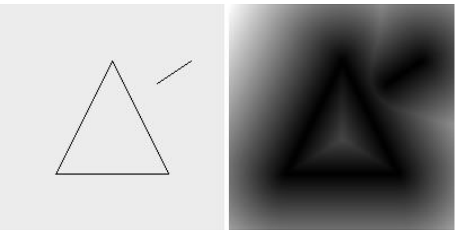


Figure 4.3: Example of Distance transform, the picture is taken from [10].

segmentation for $i-th$ frame is formulated as an energy minimization problem which is aimed to find the binary label set $L_i$, where the element of $L$ $l_i \in \{f,b\}$ represents the pixels in $i_{th}$ frame, label $f$ is foreground object, $b$ is background scenne. The formula is defined as:

$$E_I(L) = \sum_{i\in I}^{L=f} U_i + \sum_{j\in I}^{L=b} V_j + \sum_{p,q\in N,p\neq q} W(p,q) \tag{4.6}$$

where $I$ is $i-th$ frame in the video sequence, $i,j$ is any pixel in $ith$ frame. $N$ is four connect neighborhood in $I$. The two terminal term and smoothness term is defined as :

$$U_i = Sal_i \tag{4.7}$$

$$V_i = 1 - Sal_i \tag{4.8}$$

$$W = \alpha e^{(}\frac{|I_p - I_q|^2}{\beta}) + \lambda S_p(\frac{p+q}{2}) \tag{4.9}$$

where $\beta = E|I_p - I_q|^2$ and $S_p(\frac{p+q}{2}) \approx \frac{S_p(P)+S_p(q)}{2}$ The whole algorithm can be described as:

---

**Algorithm**

**Input**: $ith$ frame $I$, its object saliency map $Sal_i$, motion constrains $Sp(I)$

**Output**: binary image

**Begin**

```
        For each frame
        Create node for each the pixel in $I$;
        Create source node to represent label $f$ and sink node for label $b$;
        Add edge from source node to node for the pixels with weight $u_i$ accordingly;
        Add edge from node for the pixels to the sink node with weight $v_i$ accordingly;
        Add edge among the nodes for the pixels with weight $w$;
        Apply max flow to proposed function;
End
```

## 4.2 Experiment

In this section, we demonstrate our quantitative and qualitative result in two benchmark datasets[5] [6]. The first dataset contains six different type of videos (bird, girl, birdfall, parachute, penguin) with manually marked binary masks ground-truth. To our best knowledge, this is the well known and largest video dataset with pixel level ground-truth. This dataset includes common challenge tasks in video segmentation, e.g., scale moving object, fast camera motion. We carefully follow the work[5] and [25] to compute the pixel-error for each video (penguin is discarded) and which is :

$$error = \frac{XOR(GT \cap f)}{F} \tag{4.10}$$

where $f$ is segmentation result in each frame, $GT$ is ground-truth mask for each frame. $F$ is the total frame number.

We compare our proposed method with three state-of-art methods [25], [5], [60] and the result are shown in Table 4.1. Note that, our method is an unsupervised and less time consuming. It achieves best result out of the compared method in parachute video and the result is comparable in terms of girl and monkey dog video.

Table 4.1: Segmentation errors as measured of the average number of incorrect pixels.

| Video | ours | [25] | [5] | [60] |
|---|---|---|---|---|
| $birdfall$ | 751 | 288 | 252 | 454 |
| $cheetah$ | 1553 | 905 | 1142 | 1217 |
| $girl$ | 1956 | 1785 | **1304** | 1698 |
| $monkeydog$ | 573 | 521 | 563 | 683 |
| $parachute$ | **192** | 205 | 235 | 502 |
| $supervised?$ | N | N | Y | Y |
| $timeconsuming(s)$ | 5.4 | 312 | 65.4 | N/A |

Specifically, [5] and [60] are supervised methods which need annotated ground truth at first frame. Besides, for qualitative evaluation we also followed [25], we use two videos from dataset[6]. Figure 4.3 shows the results from our method (first row) and [6](second row). From the result we can see that our proposed can capture the foreground in the sense of shape deformable over the time, while [6] does not include foreground detector which causes over segmented.

## 4.3 Conclusion

In this chapter, we development an automatic video object segmentation method based on graphical model, we use object saliency for foreground initialization and object motion saliency for keeping temporal coherence. Our proposed method clearly explicates the question *what is the object in the video*, besides, our method is easy to implement and compute efficiently. Experimental result demonstrates that our method is comparable to existing methods.

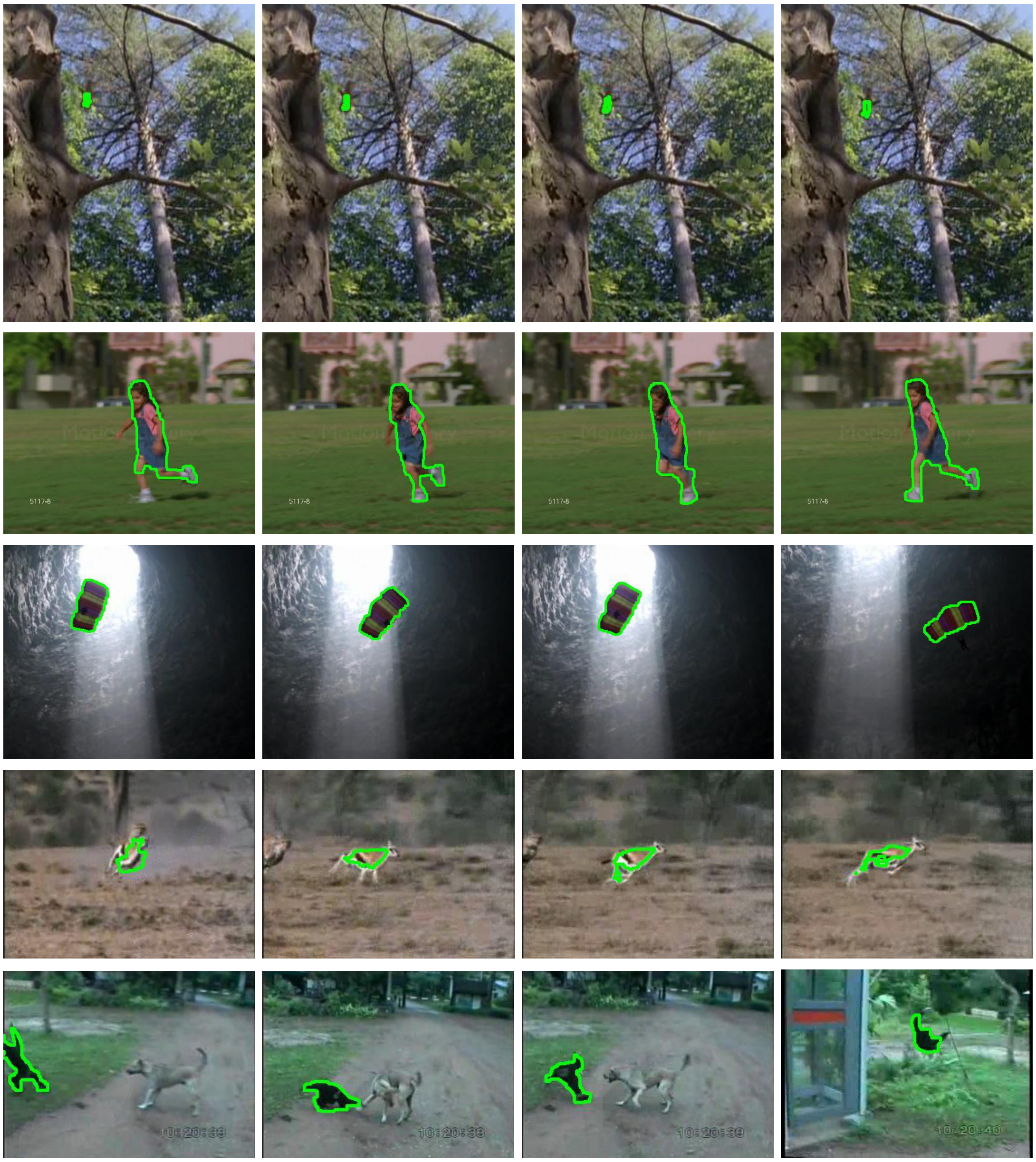


Figure 4.4: Our result on dataset [5],from top to bottom: birdfall, girl, parachute, cheetah, monkeydog.

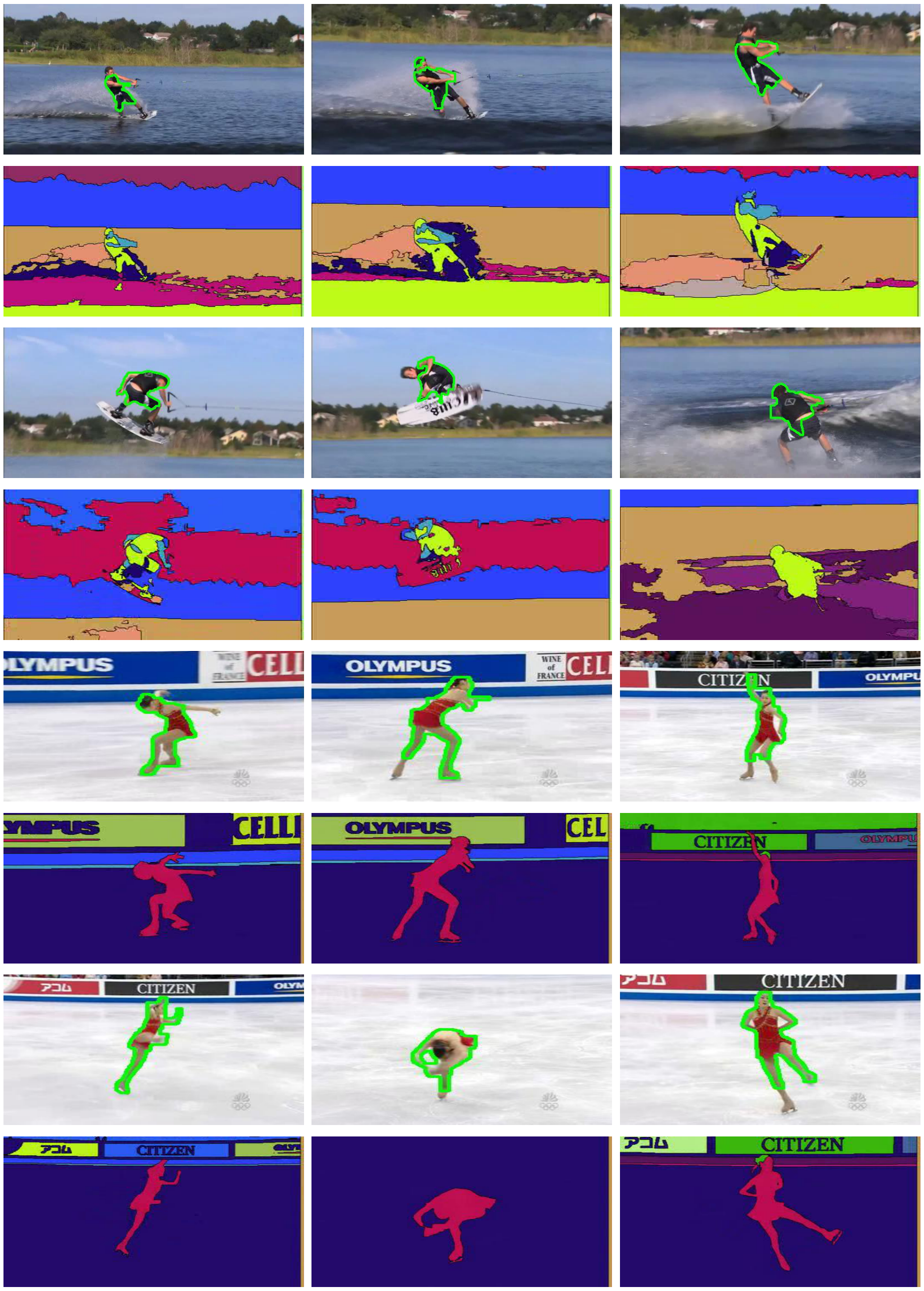


Figure 4.5: Result comparison on dataset[6].

# Chapter 5

## Conclusion and Future Work

In this thesis, we have proposed a novel efficient video object segmentation algorithm, which can automatically extract the foreground object and predict its locations through the video. Inspired from previous work [25, 6, 14], we address two insufficiencies in VOS task: first, to our best knowledge, current works are lack of definition or notation on *what is foreground object in the video*. Second, because of lacking efficient computation model, there is no VOS work considers keeping global motion coherence. Starting from these motivations, we introduce two unsupervised saliency detectors for object detection and motion detection.

For saliency object detection, we consider spatial correlation between different regions in static image. Following the basic principles of human vision attention model[18, 16, 2]. We introduce three characters of saliency object : 1, local surrounding character. 2, location character. 3, object character. To this end, we build an object saliency computational model based on its context appearance, boundary and gradients. Since deciding the object size in input image is difficult, we employ multi-scale surrounding contexts based on the Gestlaw. In the experimental section, we demonstrate our approach outperforms nine existing state of art approaches on well known benchmark dataeset.

For temporal motion saliency detection, we propose a novel efficient model which extracts the phase information from video. To overcome the computational complexity of long term video, we provide a sliding window function, the size of sliding window determines the motion cues, for large window size, more global information will be considered, for smaller window size, more local information will be included which will improve the resolution. More importantly, we demonstrate the multiple feature channels in our proposed model can be solved by complex FFT.

Finally, in experimental part, proposed motion detector has shown its advantages in three vision tasks: motion detection, abnormal event detection, and action recognition.

Motivated by [8], we combine our two saliency features into a graphical model. We argue that the foreground should have characters of a saliency object, first, the foreground object is more salient than background, which draws more human attention. Second, foreground object should have coherence closed boundary, third, compare to static object, foreground object in the video should have motion trajectories. To this end, the data term in graph model is initialized by our object saliency map, for the smoothness term we considers both local interactive and motion constrains. For our video object segmentation algorithm, we evaluate it on two common datasets and compare it with three state of art methods in quantatitively and qualitatively results.

In order to achieve better result in the future, we consider to improve three aspects from our proposed method, first, in saliency object detection section, we only apply simple canny edge detector for boundary detection, however, many better approaches have been proposed in recent year, we can explore boundary influence for saliency detection. Second, our video segmentation algorithm only segments the foreground once, which may not guarantee a high quality result (Figure 5.1), we will continue our research and find efficient model for result refinement. Last but not least, since video object based video encoding is widely used in practice, we will apply our proposed object segmentation algorithm to object extraction part in automatic multimedia community, e.g., video conference,video phones, and compare with existing approaches.

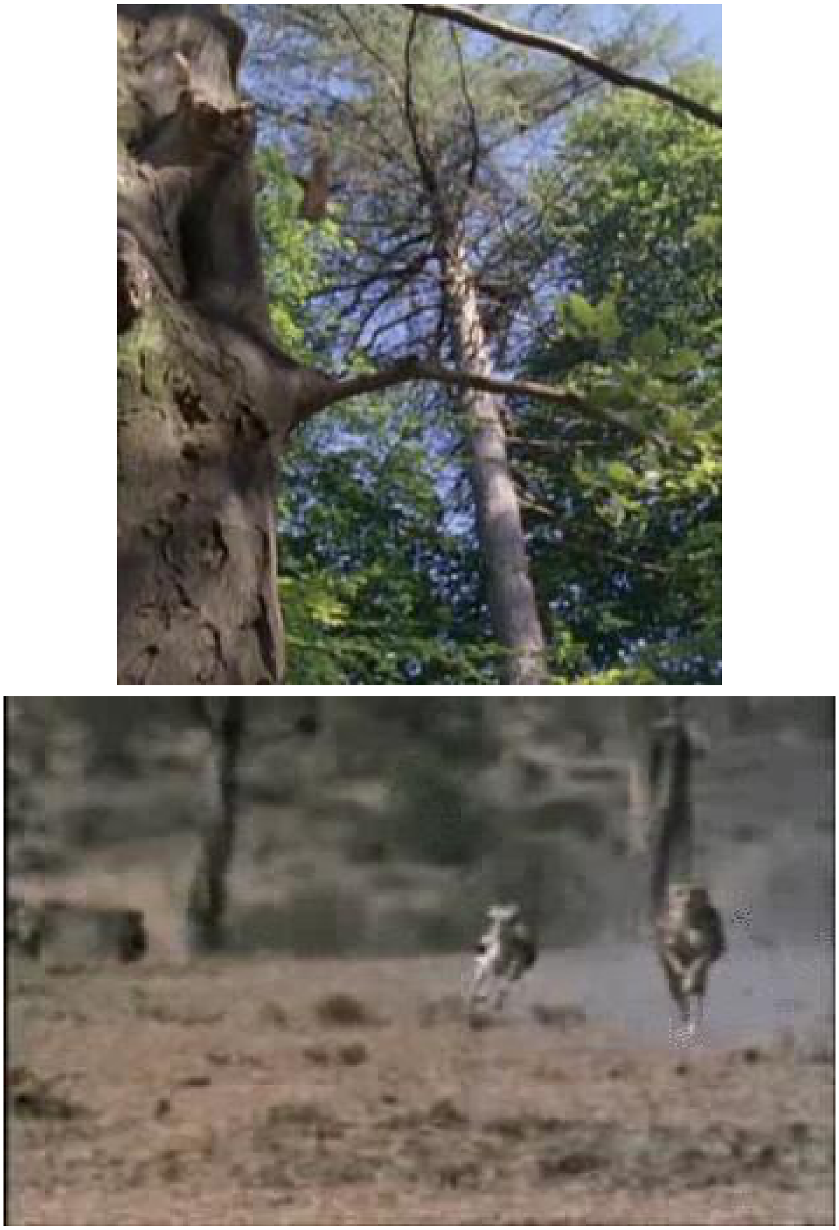


Figure 5.1: Failure cases of proposed method.

## REFERENCES

[1] A. Borji and L. Itti, “State-of-the-art in visual attention modeling,” 2013.

[2] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, “Learning to detect a salient object,” in *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[3] M. D. Rodriguez, J. Ahmed, and M. Shah, “Action mach a spatio-temporal maximum average correlation height filter for action recognition,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, pp. 1395–1402, IEEE, 2005.

[5] D. Tsai, M. Flagg, and J. Rehg, “Motion coherent tracking with multi-label mrf optimization,” *algorithms*, 2010.

[6] M. Grundmann, V. Kwatra, M. Han, and I. Essa, “Efficient hierarchical graph-based video segmentation,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2141–2148, IEEE, 2010.

[7] J. Kleinberg and E. Tardos, *Algorithm design.* Pearson Education India, 2006.

[8] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1222–1239, 2001.

[9] J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[10] P. Felzenszwalb and D. Huttenlocher, “Distance transforms of sampled functions,” tech. rep., Cornell University, 2004.

[11] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.

[12] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 304–311, IEEE, 2009.

[13] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *Computer Vision–ECCV 2010*, pp. 282–295, Springer, 2010.

[14] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller, "Multiple hypothesis video segmentation from superpixel flows," in *Computer Vision–ECCV 2010*, pp. 268–281, Springer, 2010.

[15] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 73–80, IEEE, 2010.

[16] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 1915–1926, 2012.

[17] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li, "Automatic salient object segmentation based on context and shape prior," in *BMVC*, vol. 3, p. 7, 2011.

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

[19] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision research*, vol. 40, no. 10-12, pp. 1489–1506, 2000.

[20] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[21] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 1, pp. 194–201, 2012.

[22] Y. J. Lee and K. Grauman, "Collect-cut: Segmentation with top-down cues discovered in multi-object images," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3185–3192, IEEE, 2010.

[23] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, “Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 993–1000, IEEE, 2006.

[24] G. Kim and A. Torralba, “Unsupervised detection of regions of interest using iterative link analysis,” 2009.

[25] Y. J. Lee, J. Kim, and K. Grauman, “Key-segments for video object segmentation,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1995–2002, IEEE, 2011.

[26] X. Bai, J. Wang, D. Simons, and G. Sapiro, “Video snapcut: robust video object cutout using localized classifiers,” in *ACM Transactions on Graphics (TOG)*, vol. 28, p. 70, ACM, 2009.

[27] Y. Huang, Q. Liu, and D. Metaxas, “Video object segmentation by hypergraph cut,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1738–1745, IEEE, 2009.

[28] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, “Sun: A bayesian framework for saliency using natural statistics,” *Journal of Vision*, vol. 8, no. 7, 2008.

[29] Y. Xie, H. Lu, and M. Yang, “Bayesian saliency via low and mid level cues,” 2012.

[30] X. Hou and L. Zhang, “Saliency detection: A spectral residual approach,” in *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[31] C. Guo, Q. Ma, and L. Zhang, “Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[32] C. Liu, P. C. Yuen, and G. Qiu, “Object motion detection using information theoretic spatio-temporal saliency,” *Pattern Recognition*, vol. 42, no. 11, pp. 2897–2906, 2009.

[33] K. Rapantzikos, Y. Avrithis, and S. Kollias, “Dense saliency-based spatiotemporal feature points for action recognition,” in *Computer Vision and Pattern*

*Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1454–1461, IEEE, 2009.

[34] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 409–416, IEEE, 2011.

[35] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," 2012.

[36] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[37] A. Levinshtein, S. Dickinson, and C. Sminchisescu, "Multiscale symmetric part detection and grouping," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2162–2169, IEEE, 2009.

[38] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Computer Vision–ECCV 2008*, pp. 705–718, Springer, 2008.

[39] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels," *École Polytechnique Fédéral de Lausssanne (EPFL), Tech. Rep*, vol. 149300, 2010.

[40] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1597–1604, IEEE, 2009.

[41] Y. Fang, Z. Chen, W. Lin, and C.-W. Lin, "Saliency detection in the compressed domain for adaptive image retargeting," *Image Processing, IEEE Transactions on*, vol. 21, no. 9, pp. 3888–3901, 2012.

[42] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Computer Vision–ECCV 2012*, pp. 29–42, Springer, 2012.

[43] B. P. Ölveczky, S. A. Baccus, and M. Meister, "Segregation of object and background motion in the retina," *Nature*, vol. 423, no. 6938, pp. 401–408, 2003.

[44] C. Guo and L. Zhang, “A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression,” *Image Processing, IEEE Transactions on*, vol. 19, no. 1, pp. 185–198, 2010.

[45] O. P. Popoola and K. Wang, “Video-based abnormal human behavior recognition—a review,” 2012.

[46] R. Fredericksen and R. Hess, “Temporal detection in human vision: Dependence on stimulus energy,” *JOSA A*, vol. 14, no. 10, pp. 2557–2569, 1997.

[47] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, “Anomaly detection in crowded scenes,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 1975–1981, IEEE, 2010.

[48] R. Mehran, A. Oyama, and M. Shah, “Abnormal crowd behavior detection using social force model,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 935–942, IEEE, 2009.

[49] Y. Cong, J. Yuan, and J. Liu, “Sparse reconstruction cost for abnormal event detection,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 3449–3456, IEEE, 2011.

[50] J. Kim and K. Grauman, “Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2921–2928, IEEE, 2009.

[51] I. Laptev, “On space-time interest points,” *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[52] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 12, pp. 2247–2253, 2007.

[53] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local svm approach,” in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3, pp. 32–36, IEEE, 2004.

[54] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in *Visual Surveillance and Performance*

*Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pp. 65–72, IEEE, 2005.

[55] G. Willems, T. Tuytelaars, and L. Van Gool, “An efficient dense and scale-invariant spatio-temporal interest point detector,” in *Computer Vision–ECCV 2008*, pp. 650–663, Springer, 2008.

[56] W. Brendel and S. Todorovic, “Video object segmentation by tracking regions,” in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 833–840, IEEE, 2009.

[57] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, “Ranking on data manifolds,” *Advances in neural information processing systems*, vol. 16, pp. 169–176, 2003.

[58] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang, “Manifold-ranking based image retrieval,” in *Proceedings of the 12th annual ACM international conference on Multimedia*, pp. 9–16, ACM, 2004.

[59] X. Yuan, X.-S. Hua, M. Wang, and X.-Q. Wu, “Manifold-ranking based video concept detection on large database and feature pool,” in *Proceedings of the 14th annual ACM international conference on Multimedia*, pp. 623–626, ACM, 2006.

[60] P. Chockalingam, N. Pradeep, and S. Birchfield, “Adaptive fragments-based tracking of non-rigid objects using level sets,” in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1530–1537, IEEE, 2009.