Discovering Primary Objects in Videos by Saliency Fusion and Iterative Appearance Estimation

Jiong Yang, Gangqiang Zhao, *Member, IEEE,* Junsong Yuan, *Senior Member, IEEE,* Xiaohui Shen, *Member, IEEE,* Zhe Lin, *Member, IEEE,* Brian Price, *Member, IEEE,* Jonathan Brandt, *Member, IEEE*

***Abstract*—In this paper, we propose a new method for detecting primary objects in unconstrained videos in a completely automatic setting. Here, we define the primary object in a video as the object that presents saliently in most of the frames. Unlike previous works only considering local saliency detection or common pattern discovery, the proposed method integrates the local visual/motion saliency extracted from each frame, global appearance consistency throughout the video and spatio-temporal smoothness constraint on object trajectories. We first identify a temporal coherent salient region throughout the whole video and then explicitly learn a discriminative model to represent the global appearance of the primary object against the background to distinguish the primary object from salient background. In order to obtain high quality saliency estimations from both appearance and motion cues, we propose a novel self-adaptive saliency map fusion method by learning the reliability of saliency maps from labelled data. As a whole, our method can robustly localize and track primary objects in diverse video content, and handle the challenges such as fast object and camera motion, large scale and appearance variation, background clutter and pose deformation. Moreover, compared to some existing approaches which assume the object is present in all the frames, our approach can naturally handle the case where the object is only present in part of the frames, *e.g.* , the object enters the scene in the middle of the video or leaves the scene before the video ends. We also propose a new video dataset containing 51 videos for primary object detection with per-frame ground truth labeling. Quantitative experiments on several challenging video datasets demonstrate the superiority of our method compared to the recent state of the arts.**

***Index Terms*—primary object, automatic object detection, saliency fusion, appearance model.**

I. Introduction

With the prevalence of on-line social video sharing, considerable amounts of videos are being created and processed every day. In many of those videos, there exists a primary object that we want to focus our attention on, *e.g.* , a child or a pet in a "homemade" personal video. We define the primary object in a video sequence as the object that presents saliently in most of the frames and some examples are shown in Figure 1. In this paper we address the problem of automatically discovering the primary objects in videos, which is an essential step for many applications such as advertisement design [36] and video summarization [20], [44], [28]. Traditional video object detection and localization methods, however, are either too category specific (*e.g.* , face[47] and pedestrian detection[13]) or heavily rely on manual initialization (*e.g.* , object tracking [19] and interactive object segmentation [18]). They are suitable for targeted object detection that is tailored to users' interests, but are too limited for many multimedia applications that require automatically processing large volumes of video data with diverse content. Throughout the paper, we will also use the term "foreground object" or simply "foreground" interchangeably with the term "primary object".

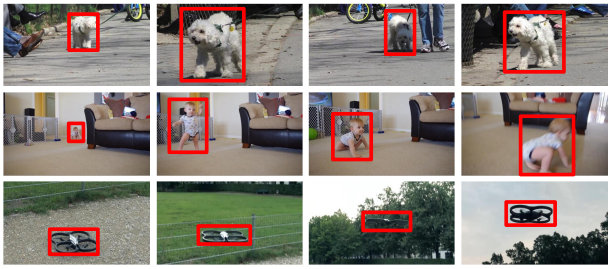


Fig. 1. Examples of primary object discovery. Each row corresponds to one video and the red rectangle highlights the primary object.

One way to automatically discover the primary objects is by resorting to saliency detection [35], [23], as the primary objects are usually distinctive either in appearance or in motion compared to the background. Regarding to the three different saliency levels introduced in [16], we are referring to high level salient object detection instead of visual attention modeling or pixel wise salient object segmentation. Although, different image and motion saliency cues are explored and combined [31], [54], [34], these methods simply combine the appearance and motion saliency maps by weighted average where the weights are empirically determined. As a result, the final saliency map can be inevitably affected by the noise in each single map. Moreover, primary object detection is not equivalent to salient object detection. Besides being locally salient, the primary object also needs to be common throughout the video sequence, *i.e.* , present in most of the frames. Saliency is only one of the cues to determine where the primary object is but there are more factors to consider, such

J. Yang is with Rapid Rich Object Search Lab, Nanyang Technological University, Singapore, 637553 (e-mail: yang0374@e.ntu.edu.sg).

G. Zhao and J. Yuan are with the Department of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, 639798 (e-mail: gangqiangzhao@gmail.com, jsyuan@ntu.edu.sg).

X. Shen, Z. Lin, B. Price and J. Brandt are with Adobe Research, 345 Park Ave, San Jose, CA 95110 (e-mail: {xshen, zlin, bprice, jbrandt}@adobe.com).

This work is supported in part by Adobe gift grant and Singapore Ministry of Education Academic Research Fund (AcRF) Tier 1 grant M4011272.040.

This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Prime Ministers Office, Singapore, under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office.

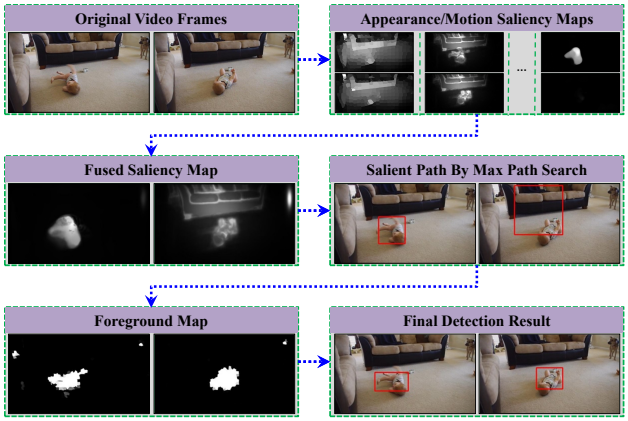


Fig. 2. Work flow of the proposed detection framework by example.

as temporal smoothness and appearance consistency across frames. In some pure saliency-based detection framework [31], little appearance information of the video object is captured, which may cause the detection to drift from the primary object to other salient objects or background regions. On the other hand, in order to model the object appearance for automatic primary object discovery, common visual pattern mining methods have been investigated [43], [29], [58], [53], [11], [50], and significant progress has been made. However, these unsupervised pattern mining methods require the object to appear frequently and have consistent appearances across the whole video sequence such that its visual pattern can be discovered. Their performance would degrade if the primary object appears with large visual variation due to the illumination, scale and viewpoint variation, partial occlusion and deformation. Moreover, the static background with rich features may be more common than the primary objects and treated as common object incorrectly.

In summary, pure saliency-based detection can easily drift among different salient objects or include salient background regions due to the lack of explicit appearance modeling. Hence, in order to tackle this problem we propose to first use local saliency cues to automatically produce some weakly supervised information about the foreground object by considering the temporal consistency of the salient regions. This weak information is then used to explicitly learn a foreground appearance model against the background regions in an iterative and discriminative manner. The evaluation results using a newly proposed dataset, *NTU-Adobe* primary object discovery dataset, and two other challenging video datasets, *i.e.* , the 10-video-clip dataset [15] and some selected categories in the UCF sports action dataset [37], demonstrate the efficacy of our method. We briefly introduce our method in the following.

Firstly, in order to obtain more accurate saliency estimations, we fuse different types of saliency cues including appearance based image saliency, motion saliency and semantic saliency. Instead of simply combining these different saliency cues empirically, we propose a novel learning-based saliency fusion technique, *SVM-Fusion* that can judge the quality of each saliency map and determine its combination weight in an adaptive manner. Then in order to find spatio-temporally coherent salient regions, we formulate the problem into the framework of max path search [46].

Secondly, in order to ensure appearance consistency in the detection process and better distinguish the foreground object against the background, we use the pure saliency-based detections as weakly supervised information to explicitly learn the foreground object appearance in an iterative manner. Then the learned model is used to produce a much cleaner and appearance-consistent foreground detection map, based on which much better detection results can be obtained.

In summary, the target of our work is to automatically discover and localize the primary object in a given video sequence without any human interaction. The overall work flow of the proposed method is shown in Figure 2. The major contributions of this work are:

1) We propose a unified framework for automatic primary object detection and localization by exploring the local saliency cues and explicitly modeling the foreground/background appearance in a discriminative manner.
2) We propose a novel learning based saliency map fusion technique which can adaptively fuse appearance and motion saliency maps to make use of their strong complementation.
3) We propose a new multi-category video object dataset for automatic primary object detection with per-frame ground truth bounding box labeling, which will be shared with the research community.

II. Related Works

In this section, we will review two important cues for visual object discovery in the literature: visual saliency detection and appearance models of visual objects.

Visual saliency has attracted wide attention of researchers in different fields. As mentioned in the Introduction, we are referring to the object level saliency estimation instead of human eye fixations [16]. Many preliminary studies [5], [38], [35], [23], [17] showed that several bottom-up factors (*e.g.* , contrast in intensity, color or texture) are important for image saliency detection. Later on, it was discovered that top-down factors (*e.g.* , faces, cars, animals, and text) [25][41] can be incorporated to complement bottom-up features and obtain better image saliency results [59][33]. To detect the salient regions in videos, motion information is further incorporated into the saliency model [27][54]. Recently, other cues related to visual saliency have also been explored, *e.g.* , prior knowledge [48][30][24], image segmentation results [51][14], and sparse analysis [21]. More details about visual saliency detection can be found in a recent evaluation study [39]. Combining appearance and motion saliency cues has also been explored in the literature. The primary motivation is to leverage their strong complementation to produce high quality spatio-temporal saliency map. These methods can be divided into three categories based on their adaptability. The first category of methods uses predefined fusion functions such as mean, multiplication and maximization[34]. The second category of methods uses some empirical measures, *e.g.* ,

spatial variance[12] or motion variance [54], to assess the quality of each saliency map and combine them based on the quality measure. The last type of methods uses learning based approach to learn the fusion process such as [39] and [32]. However, [39] works on human fixation estimations and [32] only adapts to each saliency estimation method, *i.e.* , which method is good or bad in general instead of which method is good or bad for a particular image. The more adaptive version of [32] heavily relies on the training data even during testing. In contrast, our proposed method is dedicated to fuse object level appearance and motion saliency. It is very fast, does not require training data during testing and can be seamlessly applied to any new object level saliency estimation techniques. Besides visual saliency cues, visual pattern mining research has shown that the object's appearance also provides essential information for object discovery [29][53]. Probabilistic topic models were used to model the appearance of visual objects. Russell *et al.* [42] discovered the visual object categories from image collections using Latent Dirichlet Allocation (LDA) [4]. Liu *et al.* [28] and Zhao *et al.* [57] employed a LDA model to discover the visual objects from videos.

The primary object discovery problem is also related to the automatic foreground object segmentation in image collections and videos [55][7]. Batra *et al.* [3] proposed a method to interactively co-segment the foreground objects from a group of related images. Li *et al.* [26] proposed to discover co-salient objects from a group of images by fusing the intra-image saliency map and the inter-image saliency map. Papazoglou and Ferrari [40] estimated the foreground by motion boundary detection and used two Gaussian mixtures to model the appearance of foreground and background for video object segmentation. In [56], object proposals are first generated, and a layered Directed Acyclic Graph is constructed to automatically discover and segment the primary object.

III. Video Saliency Detection

As discussed in the related work section, visual saliency has been extensively studied in the literature. For the primary object discovery problem, however, any single saliency cue cannot provide robust detections due to the diversity of video content. For example, motion saliency cues are more suitable for the case where the foreground object moves differently from the background, while image saliency cues are more suitable when the foreground object is visually very different from the background. Moreover, current techniques in visual saliency estimation are not always perfect and may produce noisy and incorrect saliency maps. As each type of saliency map only works for specific cases in identifying the primary object, the incorporation of different saliency cues is essential for robust object discovery. In this section, we first introduce the individual saliency cues employed in our work, *i.e.* , static image saliency, motion saliency and semantic saliency. The fusion of saliency maps is subsequently described.

A. Saliency Cue Estimation

1) Image Saliency: Static image saliency is computed individually for each video frame. Image saliency generally emphasizes local regions with distinct textures and colors compared to the rest of the image. In this work, we use two state-of-the-art image saliency measures: the *PCA Image Saliency* proposed in [35] and the *Absorbed Markov Chain Image Saliency* proposed in [23]. The *PCA Image Saliency* detects the saliency of each image patch in a sliding window manner by considering the global contrast of the local color and pattern, while the *Absorbed Markov Chain Image Saliency* detects the saliency of each superpixel as its absorbing time in an Absorbed Markov Chain. These two methods can complement each other as they use different underlying techniques. Other image saliency measures such as [9] and [17] can certainly be applied as well. Examples of these two types of saliency maps are shown in the second and third row of Figure 7, respectively.

2) Motion Saliency: Similar to the static image saliency, we measure the motion saliency as distinct motion patterns based on dense optical flow. Two types of motion saliency are used in this work. The first one is computed as the magnitude of the *"ω-flow"* [22]. *"ω-flow"* emphasizes local motion by compensating the global motion from the original dense flow field. Similar to [22], the global motion here is estimated as a 6-parameter affine model using the Motion2D software[1] which is robust to complicated global motions like camera zooming or rotation. However, it is sensitive to the global motion model estimation. If the global motion is wrongly estimated, the obtained saliency map will be totally corrupted (an example is shown in the fifth row and the fifth column of Figure 7). Hence, we use a second type of motion saliency based on global motion contrast. We use a simple but effective voting based approach to estimate this global motion contrast: we use each pixel's flow vector to vote in the quantized x-y parameter space and then take the logarithm of the reciprocal of each cell's voting score as the global motion contrast score of those pixels voting in that cell. Mathematically, the global motion contrast saliency map of an image can be expressed as:

$$\begin{aligned} GCS(x,y) &= V(f(x,y)) \\ V(u,v) &= \log(\frac{1}{|P(u,v)|}) \\ P(u,v) &= \{(x,y) \mid f(x,y) = (u,v)\} \end{aligned} \tag{1}$$

where $GCS(x,y)$ is the global motion contrast saliency score at pixel location (x,y), $f(x,y)$ gives the quantized bin of pixel (x,y)'s optical flow, $V(u,v)$ is the saliency score of bin (u,v) on the quantized optical flow space, $P(u,v)$ is the collection of pixels whose optical flow values are quantized to bin (u,v) and $|P|$ is the cardinality of set P. Median and Gaussian filtering are applied afterwards to both types of motion saliency maps for abrupt noise rejection and smoothing. Examples of these two types of saliency maps are shown at the fourth and fifth row of Figure 7, respectively.

3) Semantic Saliency: In order to model object-level saliency, we use two types of higher level semantically meaningful priors, face and human body. Other semantical priors can also be added. Each prior will produce a separate saliency map for each frame. Viola-Jones face detector [47] is used to detect faces and the Latent SVM object detector [13] with

[1] http://www.irisa.fr/vista/Motion2D

the human body model trained on the VOC 2007 dataset[2] is used to detect human bodies. Since both detectors give bounding boxes as detection results, we use a Butterworth filter like smoothing function to re-weight the boxes to obtain a smoother saliency map. We also perform a median filter like approach to filter the bounding box detections along the temporal axis to suppress the false detections without neighbor support and recover missed detections with strong neighbor support. Examples of these two types of saliency maps are shown in the sixth and seventh row of Figure 7, respectively.

Finally, all saliency maps are normalized linearly to range between 0 and 1.

B. Saliency Fusion

Although we have obtained 6 types of saliency maps per video frame, using more than a single map does not necessarily produce better results unless we have a proper fusion technique. For example, if averaging is used, the overall map quality can be significantly affected by even a single corrupted map. On the other hand, if we can selectively reject or assign a lower weight to those corrupted maps and only use or mainly focus on the good ones, we will then have a higher chance to obtain more robust saliency estimation. This is most useful when the input maps can complement each other such as the appearance and motion saliency maps. Hence, we propose a novel $SVM-Fusion$ technique which can adaptively judge the quality of each saliency map and determine its combination weight. In the following, we first discuss an experiment in Section III-B1 to demonstrate the potential performance gain achievable through adaptive maps fusion, which is also our initial motivation to propose the *SVM-Fusion* technique. We then elaborate the fusion technique and two post-processing steps in detail in Section (III-B2) and (III-B3), respectively.

1) Best Fusion Weight: We have explicitly conducted an experiment to explore the potential performance improvement achievable with a proper map fusion technique by fusing the maps linearly using the "best" possible weights. The "best" weights are computed as follows.

Let's first denote our 6 different types of saliency maps as $\{S_i \colon 1 \leq i \leq 6\}$. The ground truth saliency map, G, is computed by filling the labeled bounding box with 1 and the rest with 0. We then formulate the computation of the best weights of each saliency map as the following least square optimization problem with linear constraints:

$$\begin{aligned} \mathbf{w}^* &= \arg\min_{\mathbf{w}} \quad \|A\mathbf{w} - \mathbf{b}\|_2 \\ s.t.& \sum_i w_i = 1, w_i > 0, \end{aligned} \tag{2}$$

where A is a matrix of 6 columns and the i^{th} column of A is the vectorized version of S_i, $\mathbf{w}$ is a 6×1 column vector where w_i is the combination weight of map S_i and $\mathbf{b}$ is the vectorized version of G. The objective function requires the combined map to be as close as possible to the ground truth map and the two linear constraints require the weights to be non-negative and sum up to 1. Certainly such "best" weights can be hardly achieved without knowing the ground truth, but its superior performance compared to the individual saliency maps before fusion, as shown in Table I and III, motivates us to seek a good fusion approach which does not require the ground truth to estimate the fusion weights.

2) SVM-Fusion: To automatically measure the saliency map quality, a Support Vector Machine is trained and used to predict the quality of each saliency map. We design 13 features to represent each saliency map and collect training samples from an independent video dataset based on the bounding box annotations on the primary object.

Based on our observation, a good saliency map will have the majority of its saliency scores concentrated on the foreground object region and thus exhibits a compact distribution, while a bad saliency map will have most of its saliency scores spread all over the frame. Hence, in order to reflect the quality of a saliency map, we extract the following features from each saliency map: (1) *Distribution Measure of Saliency Value (4 features)*: this includes the mean, variance, skewness and kurtosis of the saliency scores on each map. (2) *Spatial Pyramid Entropy (4 features)*: we first partition the saliency map into N regular grids, *e.g.* , $N = 256$ for a 16×16 partition. In the following, we use the term "saliency energy" to denote the summation of the saliency scores inside a grid/region. The set of saliency energy $\{s_i\}$ of all the grids in each partition is normalized to be a discrete probability distribution and the entropy is computed as $E = -\sum_{k=1}^{N} \frac{s_k}{\sum_{p=1}^{N} s_p} \log \frac{s_k}{\sum_{p=1}^{N} s_p}$. Essentially, this value will be low when most of the saliency scores are concentrated at a few grids and vice versa. In our experiment, we use four different partition levels to form the spatial pyramid, *i.e.* , 8×8, 16×16, 24×24 and 32×32, and each level contributes one feature. (3) *Spatial Variance (2 features)*: we use the concept of spatial variance from [10], [12]. It measures the variance of a distribution in which the random variable is the spatial location of each pixel and the probability is in proportional to its saliency score. We measure the spatial variance along the vertical and horizontal directions separately as two features. A small spatial variance implies that most of the saliency scores are concentrated at a compact region on the map. (4) *Inter-Map Coherence (3 features)*: this set of features aims to measure the coherency among different saliency maps. For each map, we first threshold the saliency scores to obtain a binary map in which '1' represents salient region and '0' represents background region. We then compute the percentage of the salient region on a map that are also salient on each of the other maps and take the maximum value as its inter-map support. We use three different values from high to low to threshold the saliency map and obtain three inter-map support features for each map. In total, a feature vector of 13 dimensions is used to characterize each saliency map.

In order to have enough training samples and avoid overfitting, we have collected a separate training dataset composed of 24 video clips with manually labeled ground truth bounding boxes on the foreground objects. This dataset is completely independent with those used in experiments. We simply use it to train the fusion model and the trained model will be fixed

[2]http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/

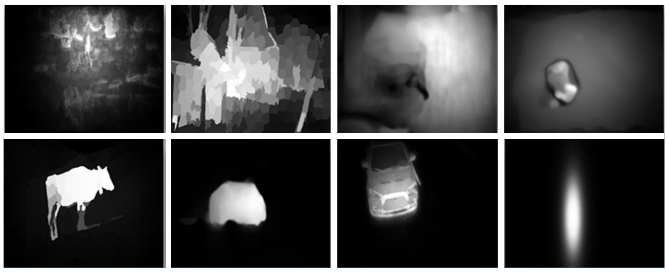


Fig. 3. Some selected saliency map training samples: the first row shows some negative training samples (saliency maps with low quality) and the second row shows some positive training samples (saliency maps with high quality).

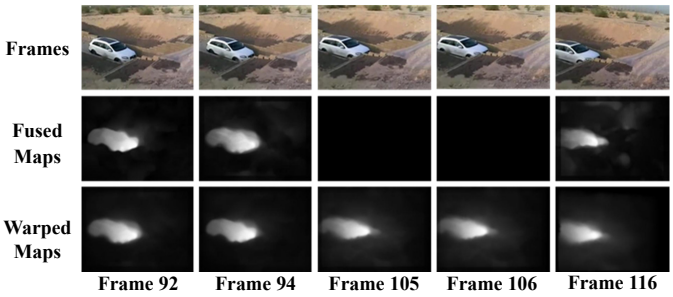


Fig. 4. An example showing how map warping recovers missed saliency detections. The top row is the original frames, the second row is the fused saliency maps and the third row is the warped saliency maps.

throughout all the experiments. Each training sample $\{\mathbf{x}_i, y_i\}$ corresponds to an actual saliency map S_i where $\mathbf{x}_i$ is the 13 dimensional feature vector extracted from S_i and y_i is the binary label indicating the quality of S_i. The value of y_i is determined by the percentage of S_i's saliency energy inside the ground truth bounding box with respect to the saliency energy of the whole map, *i.e.* , $z_i = \frac{Trace({S_i}^T G_i)}{\mathbf{1}^T S_i \mathbf{1}}$ where G_i is the ground truth saliency map corresponding to S_i which is obtained by filling its ground truth bounding box with 1 and the rest with 0. We then set $y_i = +1$ if $z_i \geq 0.8$, $y_i = -1$ if $z_i \leq 0.2$ and discard the rest. In total, we have collected 2078 positive samples and 1982 negative samples. See Figure 3 for some examples.

To predict the quality of a given saliency map, a support vector machine with RBF kernel is trained on the training samples and the parameters are selected by cross validation. We observe that our cross validation accuracy is about 97% which implies that the 13 dimensional features can well reflect the quality of the saliency map. We then use the learned SVM model to predict the quality of a given saliency map. Since the raw decision value d is unbounded, we use the probability estimates [6] as the combination weight, *e.g.* a saliency map with 90% probability of being a good map will have weight 0.9. Note that we only use the SVM model to predict the combination weights of the four images and motion saliency maps. The weights of the two semantic saliency maps are always set to 1 because they are indeed smoothed bounding boxes and always exhibit very compact distributions. Some examples of the combined maps using this learned weight are shown at the last row of Figure 7, which is significantly better than averaging. Another advantage of the proposed method is that it is very fast and convenient to use as only the trained SVM model is required during testing and the trained model is applicable to any new saliency estimation techniques without retraining.

3) Post processing: In general, high quality motion saliency cues are more reliable than image saliency cues in videos as it is more robust to cluttered background [54]. This is because, in an automatic setting without initialization, regions moving together is more likely to correspond to an object than regions with uniform appearance. Hence, we empirically emphasize motion cue by suppressing the weights of the two image saliency cues when the former is of good quality. More specifically, if we use w_p, w_a, w_g and w_ω to denote the weights of *PCA Image Saliency*, *AMC Image saliency*, *GC Motion Saliency* and ω *Motion Saliency*, respectively, this nonlinear adjustment can be expressed as:

$$(w_p, w_a, w_g, w_\omega) := \begin{cases} (0, 0, w_g, w_\omega), & w_g \text{ or } w_\omega > \eta \\ (w_p, w_a, w_g, w_\omega), & \text{otherwise} \end{cases} \tag{3}$$

where η is a threshold (set to 0.8 in the experiment) to determine whether the quality of motion saliency is good enough. The saliency maps are then fused linearly using the adjusted weights and normalized to range from 0 to 1.

Temporal warping is also applied to the fused saliency maps based on the optical flow directions to further enforce the temporal consistency of the fused saliency maps. Similar to [40], we apply both forward and backward warping. The forward warping is formulated as:

$$S_f := w_f S_f + \tau w_{f-1} S_{f,f-1} \tag{4}$$

where S_f is the saliency map of frame f and $S_{f,f-1}$ is the warped saliency map from frame $f-1$ to frame f, w_f is the *SVM-Fusion* quality measure on S_f and τ is a positive decay weight smaller than one. Note that the warping is done sequentially from the first frame to the last frame and the warped version of S_{f-1} is used to update S_f. This means that we have implicitly used all the previous frames before S_f while updating it. Similarly, the backward warping is formulated as

$$S_f := w_f S_f + \tau w_{f+1} S_{f+1,f} \tag{5}$$

and is performed from the last frame to the first frame. These two warping processes are performed separately and the resultant maps are averaged to give the final warped saliency map.

This temporal warping process is very useful in our experiment. It can effectively reject noise and recover missed detections by considering its neighbor frames. Figure 4 shows an example in which the warping process successfully recovers the missed saliency detections. However, in rare cases where many adjacent frames are corrupted by consistent noise, this warping process will also propagate this noise to nearby frames.

IV. Primary Object Discovery by Max Path Search

After we have obtained the fused saliency maps for all the frames, each video is now represented as a collection of

saliency maps, $V = \{S_i\}$ where S_i denotes the saliency map of the i^{th} frame, and in the following sections we will use $S(t, x, y)$ to denote the saliency score at pixel location (x, y) on frame t. The fused saliency maps, even when correctly highlighting the primary object, may still contain other salient objects or salient background regions. Therefore, we employ the max path search algorithm [46] to detect temporally consistent salient regions for our primary object detection. The detection result is in the form of a spatial temporal path where each node corresponds to a bounding box on a frame. In the following sections, we will first give an overview of the max path search algorithm and then discuss how we formulate our detection problem in the max path search framework.

A. Overview of Max Path Search

Max path search algorithm [46] is an optimal path discovery technique that searches for a global optimal spatio-temporal path in the 3D volume or trellis. Suppose we have a 3D spatio-temporal volume $\mathbb{G}$ composed of a set of nodes, $n_i \in \mathbb{G}$, indexed by its location (x_i, y_i) and time, t_i. A path p in $\mathbb{G}$ is defined as a temporal sequence of nodes, $p = \{n_1, n_2, ..., n_m\}$, which satisfies the path connectivity constraints between adjacent nodes. For example, a temporal adjacent connectivity constraint can be expressed as $t_{i+1} = t_i + 1$ and a spatial 8-neighbor connectivity constraints can be expressed as $x_i - 1 \leq x_{i+1} \leq x_i + 1$, $y_i - 1 \leq y_{i+1} \leq y_i + 1$. Each n_i has an associated score s_i and we define the overall score of a path p, $M(p)$, as the accumulated scores of all its nodes:

$$M(p) = \sum_{i=1}^{N(p)} s_i, \tag{6}$$

where $N(p)$ is the length of path p. The max path search algorithm can then be used to find the global optimal path p^* (with highest path score) in linear time complexity, *i.e.* , $O(whn)$ where w, h and n denote the width, height and length of the spatio-temporal volume, respectively:

$$p^* = \underset{p \in path(\mathbb{G})}{\arg\max}\, M(p), \tag{7}$$

where path($\mathbb{G}$) denotes the set of all possible paths in $\mathbb{G}$. The detailed description of the algorithm and proof of the global optimality and time complexity can be found in [46].

B. Salient Path Discovery via Max Path Search

Since our saliency map assigns per pixel saliency score, it is natural to treat each pixel as a node. However, our detection requires finding a path representing a spatio-temporal volume instead of a spatio-temporal trajectory. Hence we want each node on the path to correspond to a bounding box instead of a pixel. We adapt a similar approach to [46] where each node still corresponds to a pixel location but the score of the node is the saliency energy of a window centered at that pixel. We now use $\Omega_{p,n}$ to denote the n^{th} node of path p and its score can be expressed as

$$s_{p,n} = \sum_{i=t_{p,n}}^{i=b_{p,n}} \sum_{j=l_{p,n}}^{j=r_{p,n}} S(k_{p,n}, i, j), \tag{8}$$

where $b_{p,n}$, $t_{p,n}$, $l_{p,n}$ and $r_{p,n}$ denotes the bottom, top, left and right coordinates of the bounding box corresponding to node $\Omega_{p,n}$ and $k_{p,n}$ denotes the frame in which node $\Omega_{p,n}$ resides. In order to support scale variations in our detection, each pixel location can correspond to more than one node differed by the window size. The window size can be represented as two extra parameters, scale and aspect ratio, which can be embedded into the original 3D spatio-temporal trellis. For example, if we allow s different scales and a different aspect ratios in the detection, the original $w \times h \times t$ 3D trellis will become a $w \times h \times t \times s \times a$ 5D trellis. Similarly, we can add connectivity constraints to these new dimensions, *e.g.* , a connectivity constraint requiring the two immediately connected nodes to have the same or adjacent scale and aspect ratio levels can be expressed as $s_i - 1 \leq s_{i+1} \leq s_i + 1$ and $a_i - 1 \leq a_{i+1} \leq a_i + 1$, where s_i and a_i denote the scale and aspect ratio levels of node n_i, respectively. Mathematically, our object discovery problem can be formulated as the following optimization problem which can be solved by the max path search:

$$p^* = \underset{p \in path(\mathbb{G})}{\arg\max} \sum_{n=1}^{N(p)} s_{p,n}. \tag{9}$$

In addition, due to the score summation operation in the node and path score computation, the saliency score must be discriminative, *e.g.* , positive score means salient region and negative score means non-salient region. Otherwise the optimal path will always span from the first frame to the last frame and each node will correspond to the maximum possible bounding box. Hence, we subtract a small positive offset, γ, from the original saliency score such that both the path and bounding box can exclude non-salient regions. We will evaluate the selection of this small positive number in the experimental section.

V. Iterative Foreground Modelling

Although our max path search over fused saliency maps can apparently improve several baseline approaches as shown in the experiment, it still lacks an explicit appearance model among the nodes on a path. As a result, this can cause the detected path to drift from the primary object to other salient objects or salient background regions. In other words, the pure saliency-based detection framework can only identify if region A and region B are salient but is not able to identify whether salient region A and salient region B correspond to the same salient object. An appearance model to enforce the inter-node consistency would largely alleviate this problem. In this section, we will discuss how we explicitly model the appearance of the foreground object using the initially discovered salient path. In short, we use the initial foreground and background detections based on the discovered salient path as weakly supervised information to iteratively learn the foreground and background appearance model.

We first represent each video sequence as a collection of superpixels, $\mathcal{T} = \{p_i\}$ and the saliency score of the i^{th} superpixel, s_i, is defined as the average saliency value of its enclosing pixels. Three types of features are used to describe each superpixel, *i.e.* , dilated dense SIFT histogram, dilated

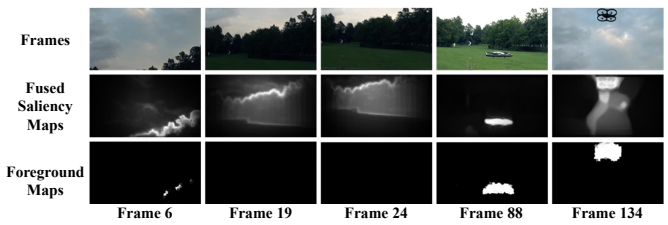


Fig. 5. Comparison between the saliency maps and the foreground maps for several frames of an example video clip. In this video clip, the primary object is the aerial vehicle which enters the scene at frame 31.

texton histogram and mean color in the RGB space. These features have also been shown to be very useful in image parsing [45]. The total feature vector dimension is 203 as 100 visual words are used for both the dilated dense SIFT histogram and dilated texton histogram. Note that we don't use the color histogram and color thumbnail features as in [45] because the size of our superpixel is quite small and its color is very uniform. In the following, we use f_i to denote the feature vector of the i^{th} superpixel.

To explicitly model the appearance of the foreground object, we iteratively train a linear SVM classifier which treats the background superpixels as negative samples and foreground superpixels as positive samples. At the first iteration, we select the foreground and background superpixels based on the fused saliency map and the discovered salient path. A superpixel will be selected as a positive training sample if it is completely inside the salient path and its saliency score is high enough while a superpixel will be selected as negative training sample if it is completely outside the salient path. Then a linear SVM is trained and used to assign each superpixel, p_i, a probability of being foreground, q_i. Subsequently, the positive and negative training samples are reselected for the next iteration based on this probability. The iteration will stop when the training samples in adjacent iterations does not change much, *i.e.* , more than 99.5% of the training samples are the same. This whole process is summarized in Algorithm 1 and we will discuss the selection of the involved parameters in the experimental section. The final foreground probability estimate, q_i of each superpixel, s_i is then used to vote a pixel-wise foreground map.

Although some related appearance modeling techniques have been explored in videos and images (collections) such as [8], [40], [18], our method has some unique properties: (1) it is fully automatic and does not require any human interventions; (2) it is a global model for the entire video; (3) it integrates the foreground and background appearance in a single unified discriminative model. A global model allows information sharing between distant frames and a discriminative model can better differentiate the foreground against background. This is especially helpful for videos because many frames share common background and the relative position of the foreground object usually changes in the background across frames, *i.e.* cover different portions of the background. An example is shown in Figure 5 where the primary object only enters the scene after frame 31. The pure saliency-based detection will also include the salient regions around the trees in the beginning frames as shown in the second row. However, these tree regions can be successfully suppressed in our foreground modeling process because of the negative (background) training samples selected around the tree regions in the later frames where the detections are correct. The foreground maps in the later frames after the object enters the scene also look cleaner compared to the saliency map. Last but not least, thanks to the efficient implementation of linear SVM with bag of words like sparse features, the modeling process is very efficient as shown in Table VI even with large training sizes, *i.e.* , thousands of superpixels per frame.

After obtaining the foreground maps F, the max path search is run to produce the final detection result. In addition, we also convolve the detected path by a median and mean filter along the temporal axis to get a smoother detection.

Algorithm 1 Iterative Foreground Modelling

Input: salient path P, the collection of superpixles $\mathcal{T} = \{p_i\}$ and their corresponding saliency score $\{s_i\}$ and feature vector $\{f_i\}$,
Parameters: θ_s is the threshold to select salient superpixels before the first iteration, θ_u and θ_l are the thresholds to select foreground and background superpixels, respectively in the following iterations
Output: foreground probability estimate, $\{q_i\}$, of each superpixel.

$\mathcal{F} = \mathcal{B} = \emptyset$
for each $p_i \in \mathcal{T}$ **do**
  if $p_i \notin P$ **then**
    $\mathcal{B} = \mathcal{B} \cup f_i$
  else if $s_i > \theta_s$ **then**
    $\mathcal{F} = \mathcal{F} \cup f_i$
  end if
end for

while true **do**
  $\mathbf{M} = TrainLinearSvm(\mathcal{F}, \mathcal{B})$
  for each $p_i \in \mathcal{T}$ **do**
    $q_i = PredictLinearSvm(\mathbf{M}, f_i)$
  end for

  $\mathcal{F} = \mathcal{B} = \emptyset$
  for each $p_i \in \mathcal{T}$ **do**
    if $q_i > \theta_u$ **then**
      $\mathcal{F} = \mathcal{F} \cup f_i$
    else if $q_i < \theta_l$ **then**
      $\mathcal{B} = \mathcal{B} \cup f_i$
    end if
  end for
  if $\mathcal{B}$ and $\mathcal{F}$ do not change **then**
    break
  end if
end while

VI. EXPERIMENTS

We have evaluated the performance of the proposed detection framework on the *NTU-Adobe* dataset and some existing benchmark datasets, *i.e.* , the 10-video-clip dataset and some selected categories from the UCF Sports Action dataset. We first introduce the employed evaluation metrics in Section (VI-A) and the new *NTU-Adobe* dataset in Section (VI-B). Then we evaluate the proposed *SVM-Fusion* and iterative foreground modeling in Section (VI-C) and (VI-D), respectively. Finally we compare the proposed technique with two state-of-the-art object discovery methods, [57] and [40], and one state-of-the-art object tracking method, [19], using the *NTU-Adobe* dataset in Section VI-E. We also compare with another video salient object detection method [31] using the *Ten-Video-Clip* dataset [15] and some selected categories of the *UCF Sports Action* dataset [37]. The computational cost of the proposed method is discussed in Section (VI-F).

A. Evaluation Metrics and Experimental Setup

Three metrics are used in the evaluation process: (1) *CDR (correct detection ratio)*: This metric measures the quality of the detected path for each video. A frame is considered to be correctly detected if the *overlap over union* ratio between the detected bounding box and the ground truth bounding box is greater than 0.5. The frames that are neither on the ground truth path nor the detected path will not be considered; (2)*FMS (f-measure of saliency map)*: This metric directly measures the quality of the saliency map compared with the ground truth saliency map. We follow the standard definition of *f-measure* in terms of *precision* and *recall*: $f\text{-}measure = \frac{2 \times precision \times recall}{precision + recall}$. Let S_g and S_d denote the ground truth saliency map and the estimated saliency map, respectively, the precision and recall are computed as $precision = \frac{Trace({S_g}^T S_d)}{\mathbf{1}^T S_d \mathbf{1}}$ and $recall = \frac{Trace({S_g}^T S_d)}{\mathbf{1}^T S_g \mathbf{1}}$, respectively; (3) *FMP (f-measure of path)*: This is the metric used in [31] and it measures the quality of a detected path. It is defined as:

$$FMP = \frac{(1+\alpha) \times \text{precision} \times \text{recall}}{\alpha \times \text{precision} + \text{recall}} \tag{10}$$

where $\text{precision} = \frac{|M_g \cap M_d|}{|M_d|}$ and $\text{recall} = \frac{|M_g \cap M_d|}{|M_g|}$. M_g and M_d is the mask on the ground truth and detected primary object region, respectively. It is computed for each frame and averaged for each video. We use it to compare with the results reported in [31].

After obtaining the combined saliency maps, we subtract a small positive number, 0.2, from the original saliency scores to get discriminative values. We apply the immediate neighbor connectivity constraint for all the dimensions, *e.g.* , $t_{i+1} = t_i + 1$, $x_i - 1 \leq x_{i+1} \leq x_i + 1$, $y_i - 1 \leq y_{i+1} \leq y_i + 1$, $s_i - 1 \leq s_{i+1} \leq s_i + 1$ and $a_i - 1 \leq a_{i+1} \leq a_i + 1$ and, hence, each node will have $3^4 - 1 = 80$ neighbors. To support a wide range of scale variations, we set the allowed bounding box scale (width) as $40, 60, ..., 240$ and the aspect ratio (*width/height*) as 0.4, 0.8, 1.0, 1.4, 1.8, 2.0. In addition, we choose a step size of 10 pixels vertically and horizontally in the two spatial dimensions while scanning the 5-D trellis in the max path search algorithm for efficiency. All the following experiments will use the same parameter configurations. In addition, in order to evaluate the effect of the empirically chosen small offset γ subtracted from the saliency value, we run the saliency based detection on the 51-video dataset without foreground modeling using different γ values and the results in terms of the path precision, recall and f-measure (computed based on the definition of *FMP* with $\alpha = 1$) are shown in Figure 6. As expected, the larger the γ, the lower the recall and the higher the precision. The highest detection accuracy in terms of f-measure is achieved when γ is around 0.2 to 0.3 and the detection result is quite stable around this range.

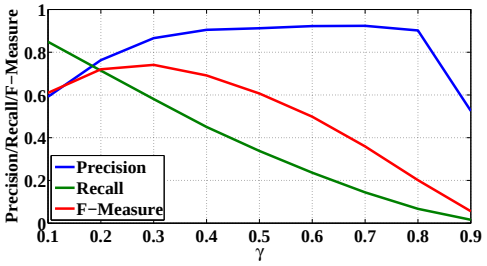


Fig. 6. The precision, recall and f-measure of the detected salient path while different offset values, γ, are subtracted from the saliency maps.

B. NTU-Adobe dataset

Most of the existing video object detection benchmark datasets have focused on specific categories of objects such as the UCF Sports Action dataset [37] for human action detection and the UIUC-NTU Youtube Walking dataset [46] for walking pedestrian detection. The huge Youtube Object dataset [3] is a multi-category dataset but only weakly annotated. A densely-annotated dataset for automatic primary object discovery with diverse object categories is desirable. Hence we collect a new multi-category dataset containing 51 video clips including animals (11 videos), babies (19 videos), walking or standing pedestrians (7 videos), cars (5 videos), motorcycles (3 videos), helicopters (2 videos), toy cars (2 videos), boat (1 video) and parachute (1 video). Example frames can be seen in Figure 1, 2, 4, 5, 7 and 9. This new dataset contains 18834 frames in total and the resolution ranges from 320×240 to 640×360. In this new dataset, 9 video clips are borrowed from the Youtube Object Dataset, 3 video clips are borrowed from the SegTrack dataset[4], 3 videos are borrowed from the UIUC-NTU Youtube walking dataset and the other 36 videos are downloaded from YouTube. Most of the videos are "home-made" videos without advanced video editing because we are mainly targeting personal videos in this work instead of professional ones like films or commercial advertisement. As a result, there are few shot changes and the objects always moves smoothly during its presence. The ground truths are manually labeled in the form of bounding boxes on each frame containing the primary object. Note that each video only has one primary object. Since the primary object detection itself

[3]http://people.ee.ethz.ch/ presta/youtube-objects/website/
[4]http://cpl.cc.gatech.edu/projects/SegTrack/

TABLE I
EVALUATION RESULTS OF DIFFERENT SALIENCY MAP FUSION TECHNIQUES ON NTU-ADOBE DATASET.

	CDR
PCA Saliency	35.93%
AMC Saliency	33.92%
GC Saliency	47.72%
W Saliency	36.26%
Max[34]	32.34%
Mean[34]	59.82%
Multiplication[34]	17.98%
Spatial Variance[12]	63.23%
Motion Variance[54]	63.71%
PW[32]	65.00%
SVM-Fusion(ours)	68.81%
Best (upper bound)	79.00%

is a subjective concept, we only include videos in which the primary object is obvious to humans to avoid ambiguity and bias. In addition, this dataset can also be used for primary object discovery in a group of videos because some videos share the same primary object. The dataset can be downloaded from our project website[5].

C. Saliency Fusion

In this section, we compare the performance of each individual saliency map and the different saliency fusion techniques using *CDR* which measures the quality of the detected salient path. We first compare our proposed *SVM-Fusion* technique (without nonlinear weight adjustment and map warping) with the individual saliency maps, the "best" fusion weight based on Equation 2 and some other existing map fusion techniques in the literature. These methods include *Max* [34], *Multiplication* [34], *Mean* [34], *Spatial Variance* [12] *Motion Variance* [54] and *Pixel-wise Aggregation (PW)* [32]. Please refer to the respective papers for the technical details. The results are summarized in Table I. As expected, the "best weight" has the highest detection accuracy and can be regarded as an upper bound of the best results achievable using weighted combination. Note that *PW* is not using weighted combination and, hence, its performance is not bounded by this "best weight" theoretically. It can be seen that our proposed *SVM-Fusion* technique outperforms the other techniques. We also show some qualitative results in Figure 7 comparing the *Mean* and *SVM-Fusion* technique to demonstrate the effectiveness of the proposed learning based fusion approach. From the result we can see that the proposed *SVM-Fusion* technique can adaptively assign lower weights to the corrupted saliency maps and emphasize the good ones.

We have also evaluated the effectiveness of the two post processing steps, *i.e.* , *nonlinear weight adjustment* and *map warping*, and the results are shown in Table II. The results show that the map warping process can apparently improve the performance while the *nonlinear weight adjustment* seems to degrade the performance when used alone. However, when the *nonlinear weight adjustment* is used together with *map warping*, it improves the performance.

[5]http://jiongsresearch.weebly.com/primary-video-object-discovery.html

TABLE II
EVALUATION RESULTS OF THE NONLINEAR FUSION WEIGHT ADJUSTMENT AND MAP WARPING.

SVM-Fusion	*Nonlinear Adjustment*	*Warping*	*CDR*
✓			68.81%
✓		✓	70.58%
✓	✓		67.72%
✓	✓	✓	72.92%

TABLE III
EVALUATION RESULTS OF DIFFERENT SALIENCY MAP FUSION TECHNIQUES ON THE FT DATASET.

	FMS
PCA Saliency	0.63
AMC Saliency	0.73
Global Contrast Saliency	0.71
GBMK Saliency	0.77
Max[34]	0.71
Mean[34]	0.75
Multiplication[34]	0.63
PW[32]	0.72
Spatial Variance[12]	0.76
SVM-Fusion(ours)	0.79
Best (upper bound)	0.80

Although our fusion method is proposed to fuse appearance and motion cues, we conduct one more experiment to explore its potential to fuse only image saliency cues. The FT dataset [1] is used for evaluation. Four image saliency estimation algorithms are used, *i.e.* , PCA Saliency [35], AMC Saliency [23], GBMK(Graph Based Manifold Ranking) Saliency [52] and Global Contrast Saliency [9]. We use the previously trained SVM-Fusion model in this experiment since it is independent of the saliency estimation methods. We sampled 1K images from the MSRA10K [9] dataset, *i.e.* , rank all the 10K images in descending order and use the first 1K images, to train the PW model since it requires the training data and testing data to use the same set of saliency estimation techniques. The saliency estimation accuracy is measured by *FMS*. The experiment result is shown in Table III.

It can be seen that our fusion method can improve the saliency accuracy from 0.77(best accuracy of individual map) to 0.79 and all the other fusion method drop the performance. Note that the best possible fusion accuracy according to ground truth is only 0.80. This implies that the potential performance gain of fusing different image saliency cues is not very significant. This is easy to understand because if one image saliency estimation algorithm performs badly on a particular image, other image saliency algorithms are also likely to perform badly on that image since all of them rely on appearance cue at the first place. This further confirms that it is more meaningful to fuse appearance and motion cues.

D. Foreground Modelling

In this experiment, we present the evaluation results on the foreground modeling. We use the SLIC [2] algorithm to segment each video frame into roughly 1500 superpixels. The parameters for training samples selection are set as: $\theta_s = 0.3$ and $(\theta_l, \theta_u) = (0.5, 0.5)$, respectively. The *CDR* on the 51-video dataset without and with the appearance

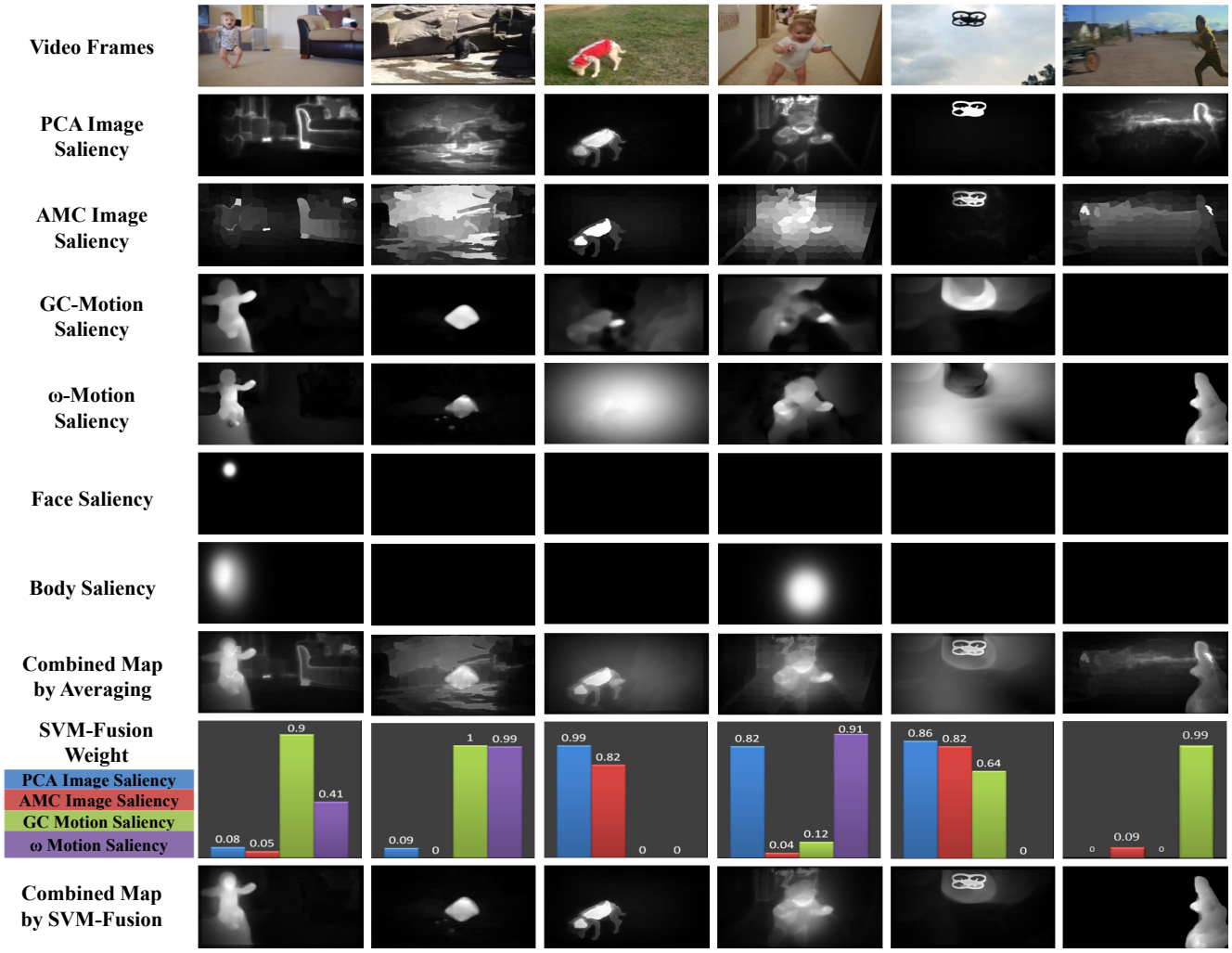


Fig. 7. Examples of the various types of saliency maps and the map fusion results by averaging and our proposed *SVM-Fusion* method without *nonlinear weight adjustment* and *warping*. The first five examples are from the *NTU-Adobe* dataset and the last example is from the Camo and Hollywood2 dataset [39].

model is 72.92% and 81.19%, respectively. Note that our detection framework without appearance model has already done a good job in many videos. But there are still cases where the saliency detection is distracted by the background even after *SVM-fusion* or the primary object only appears in part of the video. The adaption of the foreground modeling can alleviate the distraction of the background and significantly boost the performance in these cases. Some quantitative and qualitative results are shown in Figure 9. The first row shows an example in which the foreground object only enters the video after frame 31 and the second row shows an example in which the foreground object leaves the scene before the video ends. In both cases, the explicit foreground modeling can successfully exclude those irrelevant frames. The third and fourth rows show examples where saliency maps are noisy and the pure saliency-based detections include many background regions or cannot cover the entire object. The foreground modeling significantly improves the detections in these cases. In addition, we have also evaluated the sensitivity of our foreground modeling technique with respect to the choice of θ_s, θ_l and θ_u. We first evaluate θ_s by fixing θ_l and θ_u to be 0.3 and 0.7, respectively. The evaluation result is shown in the left column of Figure 8. From the result we can see that 0.3 is a reasonable choice as the detection performance is very stable around 0.0 to 0.3 and starts to drop apparently from 0.4 onwards. This is expected as the purpose of θ_s is to reject the background regions around the boundary of the bounding box and a relatively small value should be appropriate. For θ_l and θ_u, we fix θ_s to be 0.3 and evaluate them in pair, *e.g.* , (0.1, 0.9), (0.2, 0.8). The evaluation result is shown in the right column of Figure 8. It can be seen that the detection accuracy is the highest at (0.5, 0.5) and remains very stable from $(0.3, 0.7)$ to $(0.7, 0.3)$. At first glance, it may look unreasonable that the performance is still high at $\theta_l = 0.7$ and $\theta_u = 0.3$ since this seems to include many background superpixels into the positive training set and vice versa. However, we have observed that during the iterations, most of the superpixels are assigned near extreme values, *i.e.* , approaching 0 or 1. Hence, adjusting this parameters within the range from (0.3, 0.7) to (0.7, 0.3) will have a relatively small impact on the sample selection during each iteration. This also implies that our method is not sensitive to these two

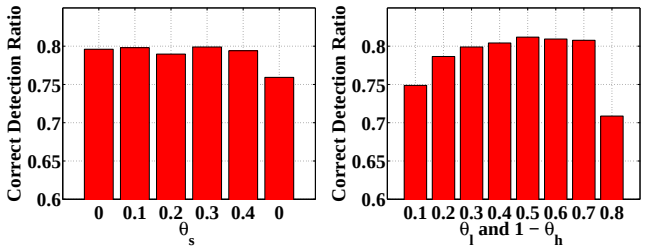


Fig. 8. Correct detection ratio with different θ_s and (θ_l, θ_u)values; the left curve shows the effect of θ_s while fixing $(\theta_l, \theta_u) = (0.3, 0.7)$ and the right curve shows the effect of (θ_l, θ_u) while fixing $\theta_s = 0.3$.

parameters and (0.5, 0.5) is a reasonable good choice. Note that the detection accuracy is relatively low around $(0.1, 0.9)$ and $(0.2, 0.8)$ because it may be too strict in selecting training samples with this setting.

E. Comparison with state of the arts

We first compare the performance of the proposed framework with [57], [19] and [40] on the NTU-Adobe dataset using *CDR*. [57] employs the LDA model to discover the primary video objects. In the experiment, we use the same setting as [57] but do not incorporate the word co-occurrence prior as it is not reliable in the employed videos. The output of this method is a set of bounding boxes which localize the objects in frames. [19] is one of the best video object tracking methods evaluated in [49]. In the experiment, we use the ground truth bounding box on the first frame (or the nearest frame if the first frame does not contain the primary object) of each video to manually initialize the tracking. The output of this technique is a set of tracked bounding boxes on the subsequent frames. [40] is the most recent state-of-the-art automatic video foreground object segmentation method. The output of this technique is per-frame segmentation masks. For comparison, we fit a minimum bounding box on the largest connected regions on the segmentation mask of each frame as the detection result. We have summarized the *CDR* of these methods as well as our proposed detection framework in Table IV. Besides the overall comparison results, we also list the results for each category in the dataset. Our method performs better than the others on the "*animals*", "*babies*", "*cars*", "*motorcycles*" and "*people walking*" categories, and worse than [40] on the "*others*" category. In the "*others*" category, there is one video sequence in which the primary object occupies the whole width of the frame throughout the video and the max path search space does not cover that large bounding box size for efficiency. In another video sequence, the primary object becomes very small for a long duration and both the two image saliency cues incorrectly focus on a very compact background region which corrupts the fused saliency map. [40] correctly identifies the primary object because it does not rely on the image saliency cues. However, we cannot simply abandon the image saliency cues because they are very useful in many other videos as shown in the comparison in Table I. On average, our method outperforms all the others, *e.g.* , around 14% improvement compared to [40]. Note that the tracking method [19] needs manual initialization and our method is completely automatic in terms of user interaction.

TABLE IV
COMPARISON WITH STATE OF THE ARTS ON NTU-ADOBE DATASET USING THE CORRECT DETECTION RATIO.

	[57]	[19]	[40]	Ours
animals (11 videos)	26.54%	35.11%	71.11%	**76.46%**
babies (19 videos)	16.50%	47.05%	61.01%	**84.31%**
cars (5 videos)	52.00%	44.38%	86.79%	**90.20%**
motorcycles (3 videos)	34.00%	39.35%	63.50%	**83.59%**
people walking (7 videos)	32.39%	49.55%	57.29%	**85.73%**
others (6 videos)	20.33%	59.55%	**83.08%**	64.93%
all (51 videos)	25.79%	44.99%	67.95%	**81.19%**

TABLE V
COMPARISON WITH [31] USING THE FMP ON THE 10-VIDEO-CLIP DATASET AND THREE CATEGORIES OF THE UCF SPORTS ACTION DATASET.

	10-video-clip	*skate*	*swing*	*run*
[31]	0.72	0.43	0.50	0.55
Ours	**0.74**	**0.59**	**0.60**	**0.57**

We also compare with [31] using our detection framework on the dataset used in their paper. [31] is a pure saliency-based detection approach. It fuses two saliency maps, *i.e.* , image saliency and motion saliency, by average and uses the max path search to find the primary object. In addition, it uses the optical flow connectivity to model the edge score between two temporally adjacent nodes in the trellis. In our approach, we don't model the edge score because in the max path search framework, each edge score can only depend on its two immediately connected nodes which limits its effectiveness, while the incremental computational cost is significant. We run our detection framework on the *10-video-clip* dataset [15] and the *skate boarding* (12 videos), *swing side angle* (13 videos) and *run side* (13 videos) category of the *UCF Sports Action* dataset [37]. In order to compare with their reported result, we use the same metric as in [31] to evaluate the detection accuracy. Note that the *horse riding* category is not chosen because the ground truth labeling of many sequences are not suitable for primary object detection, *i.e.* , both the horse and person should be the primary object but only the person is labeled. The results are summarized in Table V. From the results we can see that our detection framework outperforms [31] especially for the skate and swing dataset. The relatively small improvement, 0.02, in the *10-video-clip* and *run* dataset is because there are several video clips where all our four saliency maps miss the correct primary object and we are unable to recover the detections by fusion and foreground modeling. However, the performance on most of the other videos are superior. For example, in the *10-video-clip* dataset, our method outperforms [31] for 8 out of the 10 clips. We don't have the per-video statistic for the *run* dataset as they only provide the final score in their paper.

F. Computational Cost

We summarize the averaged per frame computational time on the newly proposed dataset in Table VI. We exclude the computational time of the various saliency maps, optical flow,

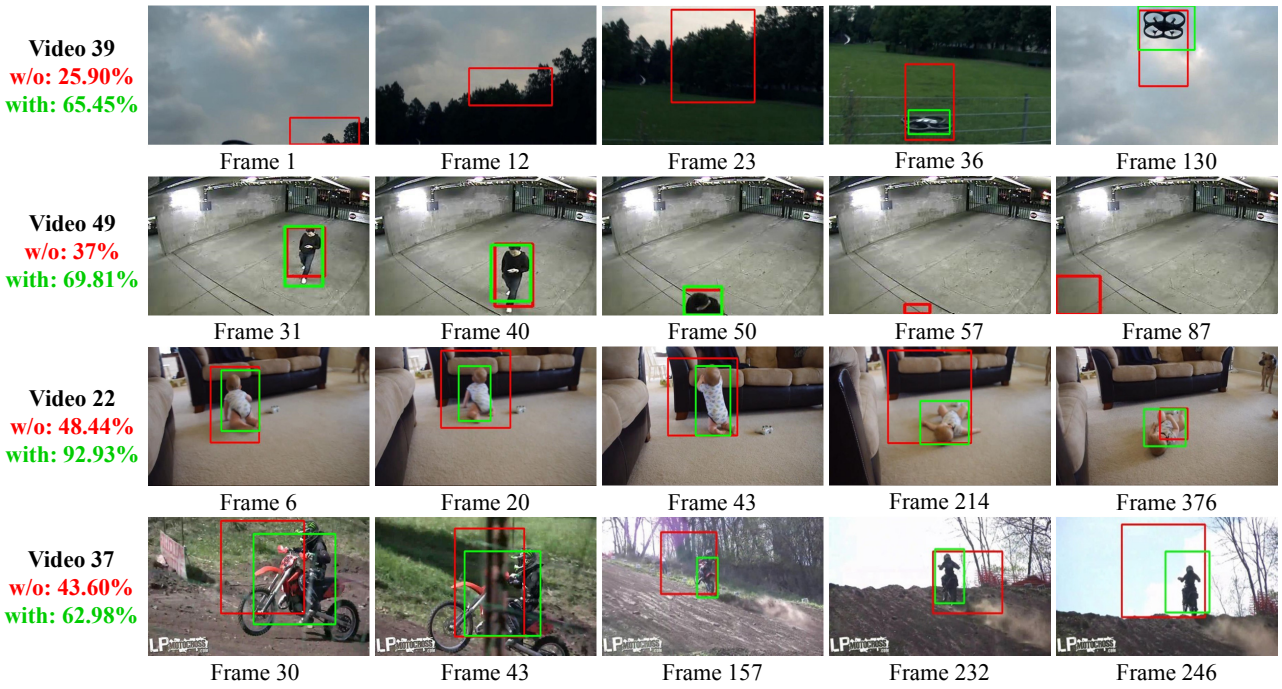


Fig. 9. Comparisons of the detection results with and without foreground modeling for some cases where the pure saliency-based detection fails. Each row refers to one video and the first column shows the overall detection accuracy in *CDR*. In the subsequent columns, the red and green box indicates the detection results before and after foreground modeling, respectively.

TABLE VI
THE AVERAGED (MEAN ± STANDARD DEVIATION) PER FRAME COMPUTATIONAL TIME FOR THE VARIOUS MODULES.

	Time (ms)
SVM-Fusion Feature Extraction	47 ± 11
Fusion Weight Computation	0.019 ± 0.002
Fused Saliency Map Warping	97 ± 28
Iterative Foreground Modelling	219 ± 127
Max Path Search	58 ± 16

SLIC superpixel, SIFT/Texton feature extraction as these are not our main contributions and different implementations can have different efficiency. Note that the proposed framework needs to extract the *SVM-Fusion* features from 5 maps (the 4 saliency maps plus the fused saliency map) and run the max path search twice (once on the warped saliency map and once on the foreground map). The max path search algorithm is implemented in C++ and the rest is implemented in Matlab. The experiments were conducted on a normal desktop computer with a quad-core i5 processor and 8GB of RAM.

VII. CONCLUSION

In this work, we propose a novel approach for fully automatic primary object discovery in videos. We first discovery a smooth spatio-temporal salient path in the video and then explicitly model the foreground and background appearance in a global and discriminative manner. To make use of the strong complementation between appearance and motion saliency cues, we propose an effective fusion technique to adaptively fuse these two types of cues. The proposed fusion method is not only effective, but also very fast and easy to use compared with similar methods in the literature. In addition, a new dataset containing 51 videos with per-frame bounding box labeling is proposed to better suit the performance evaluation purpose of automatic primary object detection in personal videos. Experimental evaluations validate the superior performance of the proposed method compared to state-of-the-art approaches on both the new dataset and some existing benchmark datasets.

REFERENCES

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1597–1604, Jun 2009.
[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012.
[3] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. Interactively co-segmentating topically related images with intelligent scribble guidance. *Int'l J. Computer Vision*, 93(3):273–292, Jul 2011.
[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, Jan 2003.
[5] R. Carmi and L. Itti. Visual causes versus correlates of attentional selection in dynamic scenes. *Vision Research*, 46(26):4333–4345, Dec 2006.
[6] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Trans. on Intelligent Systems and Technology*, 2(3):27:1–27:27, Apr 2011.
[7] D.-J. Chen, H.-T. Chen, and L.-W. Chang. Video object cosegmentation. In *Proc. ACM Multimedia Conf.*, pages 805–808, Nov 2012.
[8] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu. Salientshape: Group saliency in image collections. *The Visual Computer*, 30(4):443–453, Aug 2013.
[9] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 37(3):569–582, Aug 2014.

[10] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook. Efficient salient region detection with soft image abstraction. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1529–1536, Dec 2013.

[11] W.-T. Chu and M.-H. Tsai. Visual pattern discovery for architecture image classification and product image search. In *ACM Int'l Conf. on Multimedia Retrieval*, pages 27:1–27:8, Apr 2012.

[12] Y. Fang, W. Lin, Z. Chen, C. Tsai, and C. Lin. A video saliency detection model in compressed domain. *IEEE Trans. on Circuits and Systems for Video Technology*, 24(1):27–38, Jul 2013.

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, Sep 2010.

[14] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun. Salient object detection by composition. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1028–1035, Nov 2011.

[15] K. Fukuchi, K. Miyazato, A. Kimura, S. Takagi, and J. Yamato. Saliency-based video segmentation with graph cuts and sequentially updated priors. In *Proc. IEEE Int'l Conf. Multimedia Expo*, pages 638–641, Jul 2009.

[16] A. Furnari, G. M. Farinella, and S. Battiato. An experimental analysis of saliency detection with respect to three saliency levels. In *Proc. European Conf. Computer Vision Workshops*, pages 806–821, Sep 2014.

[17] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, Oct 2012.

[18] M. Gong and L. Cheng. Foreground segmentation of live videos using locally competing 1svms. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2105–2112, Jun 2011.

[19] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 263–270, Nov 2011.

[20] R. Hong, J. Tang, H.-K. Tan, S. Yan, C. Ngo, and T.-S. Chua. Event driven summarization for web videos. In *Proc. of the first SIGMM workshop on Social media*, pages 43–48, 2009.

[21] X. Hou, J. Harel, and C. Koch. Image signature: Highlighting sparse salient regions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(1):194–201, Jan 2012.

[22] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2555–2562, Jun 2013.

[23] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang. Saliency detection via absorbing markov chain. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1665–1672, Dec 2013.

[24] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2083–2090, Jun 2013.

[25] T. Judd, K. A. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 2106–2113, Oct 2009.

[26] H. Li, F. Meng, and K. N. Ngan. Co-salient object detection from multiple images. *IEEE Trans. Multimedia*, 15(8):1896–1909, Jun 2013.

[27] J. Li, Y. Tian, T. Huang, and W. Gao. Probabilistic multi-task learning for visual saliency estimation in video. *Int'l J. Computer Vision*, 90(2):150–165, Nov 2010.

[28] D. Liu, G. Hua, and T. Chen. A hierarchical visual model for video object summarization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(12):2178–2190, 2010.

[29] H. Liu and S. Yan. Common visual pattern discovery via spatially coherent correspondences. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1609–1616, Jun 2010.

[30] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(2):353–367, Feb 2011.

[31] Y. Luo and J. Yuan. Salient object detection in videos by optimal spatio-temporal path discovery. In *Proc. ACM Multimedia Conf.*, pages 509–512, 2013.

[32] L. Mai, Y. Niu, and F. Liu. Saliency aggregation: A data-driven approach. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1131–1138, Jun 2013.

[33] Z. Mao, Y. Zhang, K. Gao, and D. Zhang. A method for detecting salient regions using integrated features. In *Proc. ACM Multimedia Conf.*, pages 745–748, 2012.

[34] S. Marat, T. Ho Phuoc, L. Granjon, N. Guyader, D. Pellerin, and A. Guérin-Dugué. Modelling spatio-temporal saliency to predict gaze direction for short videos. *Int'l J. Computer Vision*, 82(3):231–243, Feb 2009.

[35] R. Margolin, A. Tal, and L. Zelnik-Manor. What makes a patch distinct? In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1139–1146, Jun 2013.

[36] T. Mei, L. Li, X.-S. Hua, and S. Li. Imagesense: Towards contextual image advertising. *ACM Trans. on Multimedia Computing, Communications, and Applications*, 8(1):6:1–6:18, Feb 2012.

[37] M. S. Mikel D. Rodriguez, Javed Ahmed. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3361–3368, Jun 2008.

[38] N. Murray, M. Vanrell, X. Otazu, and C. A. Parraga. Saliency estimation using a non-parametric low-level vision model. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 433–440, Jun 2011.

[39] T. V. Nguyen, M. Xu, G. Gao, M. Kankanhalli, Q. Tian, and S. Yan. Static saliency vs. dynamic saliency: A comparative study. In *Proc. ACM Multimedia Conf.*, pages 987–996, 2013.

[40] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1777–1784, Dec 2013.

[41] S. Ramanathan, H. Katti, N. Sebe, M. Kankanhalli, and T.-S. Chua. An eye fixation database for saliency detection in images. In *Proc. European Conf. Computer Vision*, pages 30–43, 2010.

[42] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1605–1614, Jun 2006.

[43] H.-K. Tan and C.-W. Ngo. Localized matching using earth mover's distance towards discovery of common patterns from small image samples. *Image and Vision Computing*, 27(10):1470–1483, Feb 2009.

[44] S. Tarashima, G. Irie, K. Tsutsuguchi, H. Arai, and Y. Taniguchi. Fast image/video collection summarization with local clustering. In *Proc. ACM Multimedia Conf.*, pages 725–728, 2013.

[45] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *Proc. European Conf. Computer Vision*, pages 352–365, 2010.

[46] D. Tran, J. Yuan, and D. Forsyth. Video event detection: From subvolume localization to spatio-temporal path search. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 36(2):404–416, Jul 2013.

[47] P. Viola and M. Jones. Robust real-time object detection. *Int'l J. Computer Vision*, 57(2):137–154, 2001.

[48] Y. Wei, F. Wen, W. Zhu, and J. Sun. Geodesic saliency using background priors. In *Proc. European Conf. Computer Vision*, pages 29–42, 2012.

[49] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2411–2418, Jun 2013.

[50] H. Xie, K. Gao, Y. Zhang, J. Li, and H. Ren. Common visual pattern discovery via graph matching. In *Proc. ACM Multimedia Conf.*, pages 1385–1388, 2011.

[51] L. Xu, H. Li, L. Zeng, and K. N. Ngan. Saliency detection using joint spatial-color constraint and multi-scale segmentation. *Journal of Visual Communication and Image Representation*, 24(4):465–476, May 2013.

[52] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3166–3173, Jun 2013.

[53] J. Yuan, G. Zhao, Y. Fu, Z. Li, A. K. Katsaggelos, and Y. Wu. Discovering thematic objects in image collections and videos. *IEEE Trans. Image Processing*, 21(4):2207–2219, Apr 2012.

[54] Y. Zhai and M. Shah. Visual attention detection in video sequences using spatiotemporal cues. In *Proc. ACM Multimedia Conf.*, pages 815–824, 2006.

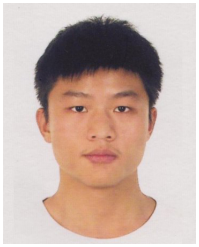
[55] B. Zhang, H. Zhao, and X. Cao. Video object segmentation with shortest path. In *Proc. ACM Multimedia Conf.*, pages 801–804, 2012.

[56] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 628–635, Jun 2013.

[57] G. Zhao, J. Yuan, and G. Hua. Topical video object discovery from key frames by modeling word co-occurrence prior. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1602–1609, Jun 2013.

[58] G. Zhao, J. Yuan, J. Xu, and Y. Wu. Discovery of the thematic object in commercial videos. *IEEE Multimedia Magazine*, 18(3):56–65, Jun 2011.

[59] Q. Zhao and C. Koch. Learning visual saliency by combining feature maps in a nonlinear manner using adaboost. *Journal of Vision*, 12(6):22, 2012.



Jiong Yang is currently a PhD candidate in Rapid Rich Object Search Lab, College of Engineering, Nanyang Technological University, Singapore. He received the B.Eng degree with first class honor in School of Electrical Electronic Engineering, Nanyang Technological University, Singapore in 2013. His research interests include computer vision and machine learning.

Gangqiang hao received the B.Eng. degree in computer science from Qingdao University, Qingdao, China, in 2003, and the Ph.D. degree in computer science from Zhejiang University (ZJU), Hangzhou, China, in 2009.

From September 2003 to December 2009, he was a Research Assistant in the Pervasive Computing Laboratory, Zhejiang University. From March 2010 to September 2014, he was a Research Fellow at Nanyang Technological University, Singapore. Since October 2014, he has been a Senior Research Scientist at Morpx Inc., Hangzhou, China. His current research interests include computer vision, multimedia data mining, and image processing.

Junsong Yuan is currently a Nanyang Assistant Professor and program director of video analytics at School of EEE, Nanyang Technological University, Singapore. He received Ph.D. from Northwestern University, USA, and M.Eng. from National University of Singapore. Before that, he graduated from Special Class for the Gifted Young of Huazhong University of Science and Technology, China. His research interests include computer vision, video analytics, action and gesture analysis, large-scale visual search and mining, etc. He has authored and co-authored 3 books, and 130 conference and journal papers.

He serves as Program Co-Chair of IEEE Visual Communications and Image Processing (VCIP15), Organizing Co-Chair of Asian Conf. on Computer Vision (ACCV14), Area chair of IEEE Winter Conf. on Computer Vision (WACV'14), IEEE Conf. on Multimedia Expo (ICME'1415), and Asian Conf. on Computer Vision (ACCV'14). He also serves as guest editor for International Journal of Computer Vision (IJCV), associate editor for The Visual Computer journal (TVC), IPSJ Transactions on Computer Vision and Applications (CVA), and Journal of Multimedia (JMM). He co-chairs workshops at SIGGRAPH Asia14, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'12'1315), IEEE Conf. on Computer Vision (ICCV'13), and gives tutorials at ACCV14, ICIP'13, FG'13, ICME'12, SIGGRAPH VRCAI'12, and PCM12. He received Nanyang Assistant Professorship from Nanyang Technological University, Outstanding EECS Ph.D. Thesis award from Northwestern University, Best Doctoral Spotlight Award from CVPR'09, and National Outstanding Student from Ministry of Education, P.R.China.

Xiaohui Shen received the BS and MS degrees from the Automation Department of Tsinghua University, China, in 2005 and 2008, respectively, and the PhD degree from the EECS Department of Northwestern University in 2013. He is currently a research scientist at Adobe Research, San Jose, California. His research interests include image/video processing and computer vision. He is a member of the IEEE.

Zhe Lin received the BEng degree in automatic control from the University of Science and Technology of China in 2002, the MS degree in electrical engineering from the Korea Advanced Institute of Science and Technology in 2004, and the PhD degree in electrical and computer engineering from the University of Maryland, College Park, in 2009. He has been a research intern at Microsoft Live Labs Research. He is currently a senior research scientist at Adobe Research, San Jose, California. His research interests include deep learning, object detection and recognition, image classification, content-based image and video retrieval, human motion tracking, and activity analysis. He is a member of the IEEE.

Brian Price is a Senior Research Scientist in Adobe Research specializing in computer vision. His research interests include semantic segmentation, interactive object selection and matting in images and videos, stereo and rgbd, and image processing, as well as broad interest in computer vision and its intersections with machine learning and computer graphics.

Before joining Adobe, he received his PhD degree in Computer Science from Brigham Young University under the advisement of Dr. Bryan Morse. As a researcher at Adobe, he has contributed new features to many Adobe products such as Photoshop, Photoshop Elements, and AfterEffects, mostly involving interactive image segmentation and matting.

Jonathan Brandt is the Director of the Media Intelligence Lab at Adobe Research. His areas of interest span a broad range of topics in computer vision and machine learning, in particular, image similarity, image classification and tagging, image and video retrieval, object detection and recognition, face detection and recognition.

The Media Intelligence Lab contributes vision and imaging technologies across Adobe's product line, including image selection, upsampling, and sharpening for Photoshop, similarity search for Lightroom and Scene7, matting for Photoshop and After Effects, camera stabilization and 3D tracking for After Effects, and face detection for numerous products. In the last several years, the Media Intelligence Lab has been pioneering the application of Deep Learning to a range of vision problems relevant to Adobe's products.

Prior to joining Adobe in 2003, Jonathan was a member of the technical staff at Silicon Graphics and a visiting professor at the Japan Advanced Institute of Science and Technology in Ishikawa, Japan.

Jonathan received his Master's and Ph. D. in Computer Science at the University of California, Davis, and a B.S. in Electrical Engineering from the University of Illinois, Urbana Champaign.