Illumination-Invariant Video Cut-Out Using Octagon Sensitive Optimization

Zhihua Chen, Jingye Wang, Bin Sheng, Ping Li, and David Dagan Feng, *Fellow, IEEE*

***Abstract*— This paper presents an effective video cut-out approach, which can be utilized to segment the moving object in video shots. We first introduce the Octagon-Sensitive-Filtering (OSF) and its illumination invariant feature (IIF), which is computed on each pixel of the image via adding contributions from neighboring pixels. We integrate our IIF into the variational model and obtain the seeds during preprocessing to help address large displacement and illumination changes. An effective seed update method based on tracking-then-refinement based on IIF is presented to compensate for location ambiguities, and the strategy is effective to deal with illumination variances and objects deformation. Furthermore, we apply the IIF-based graph-cut to deal with fuzzy boundaries. Multiple experiments on quantitative challenging datasets have shown the robustness, high-quality video cut-out and efficiency of our approach to acute variances of illumination and complex motion.**

***Index Terms*— Video cut-out, illumination-invariant, local features, seeds update, graph-cut.**

I. Introduction

VIDEO cut-out is one of the most active research topics in computer vision, also known as segmentating moving foreground objects from video frames. It is an essential task for further video editing applications such as video composition. However, there still remains many challenges in the area, including illumination change, complex motion, background variations and object occlusion. In general, the video cut-out tasks have been attempted by many methods including supervised and unsupervised approaches. For supervised approaches, training dataset is required such as video shots which are segmented manually in advance [1], [2]. The semi-supervised approaches require the user to annotate the presence of foreground objects in certain frames. The foreground segmentation accuracy depends on the labels indicated by the user, hence complicated post-processing such as random decision forest training [3] is required to enhance the correctness of the outputs. Therefore, both supervised and semi-supervised methods are labor-intensive and not applicable to automatic video cut-out applications.

Most existing popular methods in video cut-out incorporate computing optical flow between subsequent frames. Papazoglou and Ferrari [4] ropose a motion boundaries based method to make use of the appearances and location model to refine the labels. Zhang *et al.* [5] takes advantage of multiscale spatiotemporal cues to optimize the foreground extraction in video cut-out. Methods based on local motion trajectories and shapes prediction on the assumption that the objects are spatially cohesive [6], [7]. Supervoxel features based over-segmentation [8], [9], or computing incorporating saliency with spatial edges and temporal motion [10] are also used in some applications. These methods are typically based on objects motion and spatiotemporal cues, which utilize diverse models under the assumption that the object appearance changes smoothly over time. However, these existing methods we have introduced are not applicable for addressing video cut-out for sequences with significant illumination change.

In this paper, we present a new video cut-out approach, which is effective to deal with illumination change and significant object displacement. A novel Octagon Sensitive Filtering (OSF) is proposed, which is computed based on each pixel and contribution of the neighborhood pixels from eight directions. The Illumination Invariant Feature (IIF) based on it is also proposed to facilitate our video cut-out approach. The proposed feature can be applied to describe the similarity between image blocks. In preprocessing video sequences, we first integrate our IIF as local descriptor into a variational model, so that foreground and background seeds can be generated which will be further utilized for later segmentation. We formulate the problem of processing the rest of the video frames as a unified framework, which integrates tracking and segmentation techniques. We utilize the OSF to track the foreground object and then update the window

Manuscript received November 5, 2017; revised February 5, 2019; accepted February 28, 2019. Date of publication March 4, 2019; date of current version May 5, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61872241, Grant 61572316, Grant 61672228, and Grant 61370174, in part by the Project of Establishment of Shanghai Belt and Road Joint Laboratory under Grant 18410750700, in part by the National Key Research and Development Program of China under Grant 2017YFE0104000 and Grant 2016YFC1300302, in part by the Macau Science and Technology Development Fund under Grant 0027/2018/A1, in part by the Science and Technology Commission of Shanghai Municipality under Grant 18410750700, Grant 17411952600, and Grant 16DZ0501100, and in part by the Shanghai Automotive Industry Science and Technology Development Foundation under Grant 1837. This paper was recommended by Associate Editor J. Wang. *(Corresponding author: Bin Sheng.)*

Z. Chen and J. Wang are with the Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China (e-mail: czh@ecust.edu.cn).

B. Sheng is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China. (e-mail: shengbin@sjtu.edu.cn).

P. Li is with the Faculty of Information Technology, Macau University of Science and Technology, Macau 999078, China (e-mail: pli@must.edu.mo).

D. D. Feng is with the Biomedical and Multimedia Information Technology Research Group, School of Information Technologies, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: dagan.feng@sydney.edu.au).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2019.2902937

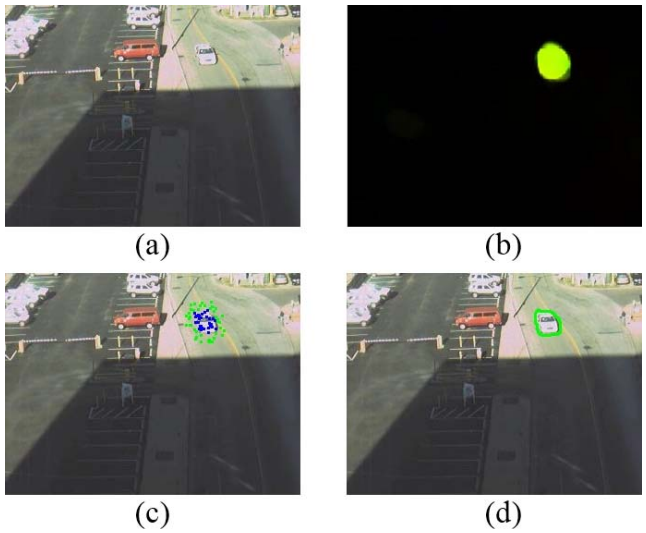


Fig. 1. Illustration for our video cut-out method. (a) The input frame of car sequence. (b) The visualizing result of optical flow field when estimating object region. (c) The background and foreground seeds are marked in green color and blue color, respectively. (d) The example of object cut-out result.

location. The location of seeds is firstly updated based on tracking displacement, and then correct it using our IIF to adapt the seeds to proper scales or pose changes. On the basis of the seeds, the objects are finally segmented through our IIF-based graph-cut approach. We integrate the proposed IIF into visual tracking and segmentation. Experiments show that the improved methods can effectively deal with illumination change and can be combined into the tracking-and-segmentation framework. Our approach has the following three main contributions:

- **A novel OSF and its corresponding illumination-invariant features.** We introduce a novel feature OSF and its corresponding illumination invariant feature. The OSF describes local spatial components and has low computational complexity, which can be utilized as a local feature to measure similarities.
- **Illumination-aware foreground region estimation.** We propose a variational-based optical flow approach using our IIF, which is robust to dramatic illumination change and large displacement of objects. We further estimate the foreground region and generate the foreground & background seeds.
- **IIF based Seeds-update and segmentation.** We propose a tracking-and-segmentation framework for video segmentation. We apply our IIF into multi-region tracking to update the object location, and an incremental seed update scheme, which can reduce the tracking errors when there is illumination or object deformation. Moreover, we integrate our IIF into a graph-cut-based image segmentation, and the segmentation accuracy is enhanced in fuzzy boundaries.

The remaining parts of the paper are organized as follows. In Section II, we review the related work on local features, video object segmentation and image segmentation. We also show the approach briefly in Fig. 1. We first present our OSF in Section III, and further introduce our approach through preprocessing the frames to estimate the initial foreground region for cutting out the following frames. Section IV presents the experiments and evaluations in comparison with existing methods. Finally, we conclude the paper with possible further work in Section V.

II. Related Work

Video cutting-out refers to segmenting foreground object in video shots, and motion segmentation is highly related to our work. In this paper, we also propose OSF (Octagon-Sensitive-Filtering) which can be used for similarity measuring. In this section, we introduce previous work including similarity measurement, video object segmentation and image segmentation.

A. Local Features and Similarity Measurement

Local features are spatially extended which refers to taking into account the contributions of neighborhood pixels typically. A good feature ought to possess properties distinctiveness, robustness and time efficiency at the same time. Liu *et al.* [11] propose a global and local structure preservation framework for feature selection integrating both pairwise sample similarity and local geometric data. SURF (Speeded Up Robust Features) [12] can be regarded as the incremental SIFT using Haar wavelet instead of magnitude and relying on integral images for convolutions. It reaches similar robustness and accuracy but 5 to 7 times faster compared with SIFT. Yang and Cheng [13] propose local difference binary by computing a binary string on the base of intensity and gradient difference, where the distinct pattern of patch is captured using the grid strategy. DAISY [14] is another HOG based dense descriptor which takes advantage of Gaussian convolution for HOG convergence in order to extract the image features densely. ASIFT (Affine SIFT) extracts the features by simulating all imaging perspectives, which is robust to large viewing-angle change [15]. Simonyan *et al.* [16] propose a local sparse feature for viewpoint invariant matching which is formulated as a convex optimization problem using sparsity. He *et al.* [17], [18] propose the locality sensitive histogram for visual tracking, which considers contribution of each pixel in an image. To conclude, existing local descriptors are not robust enough to dramatic illumination change, and time efficiency is still a challenge for most local dense descriptors.

B. Video Foreground Objects Segmentation

Supervised and semi-supervised foreground object segmentation methods require the user to annotate the object region at some key frames. Wang *et al.* [19] present a semi-supervised video cutout method by combining appearance and dynamic models and propagating user annotation to determine the labels in the uncertain region. Most unsupervised methods are based on optical flow and use temporal or spatial cues to optimize. Papazoglou and Ferrari [4] present an object segmentation method, in which they propose inside-outside map to estimate initial foreground then refine foreground labels with smoothness, appearance model and location model. Lim *et al.* [20] propose an on-line foreground segmentation algorithm in videos captured by a moving camera.

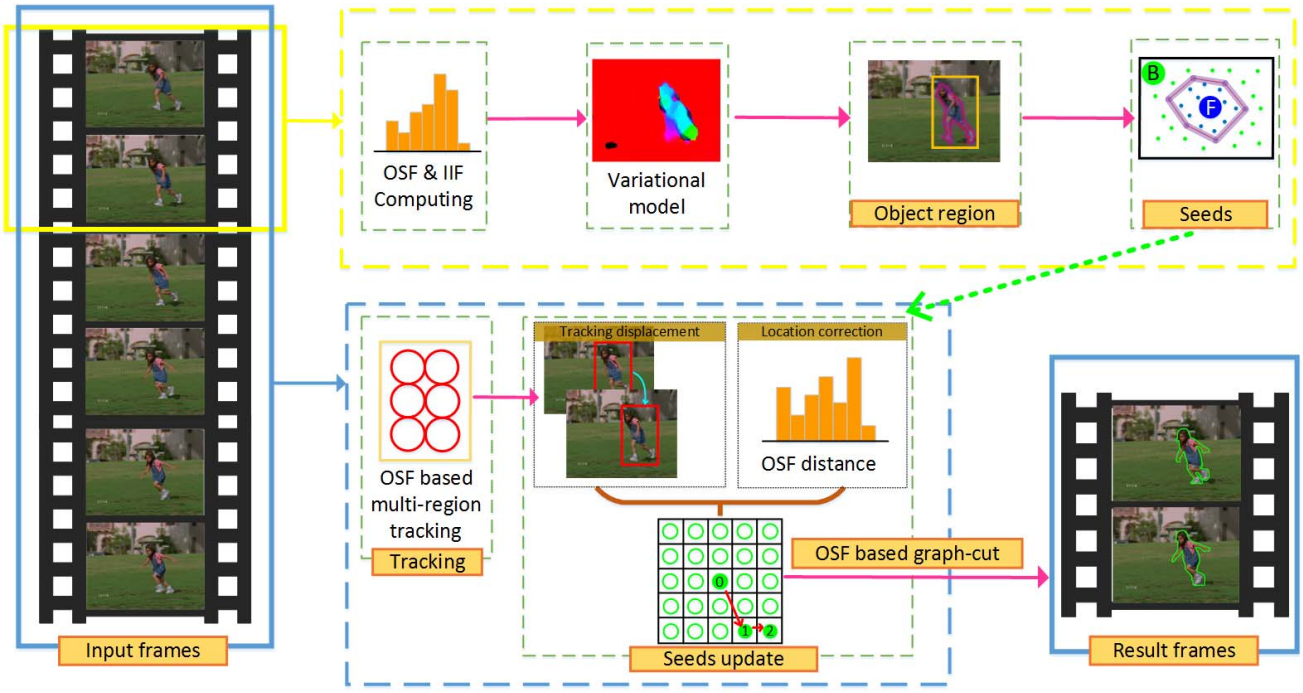


Fig. 2. The flowchart of our approach. OSF and IIF are Octagon-Sensitive-Filtering and Illumination Invariant Features. In the key-frames, we use the optimized variational model to generate the foreground and background seed points. When dealing with each frame, we first use our OSF based tracker to update the location of object and each seed point. We further use the refined graph-cut to obtain the result segmented frames.

Wang *et al.* [10] handle drastic appearance and pose variations by integrating intraframe saliency and interframe consistency. Wang *et al.* [10] introduce spatial edges and temporal motion boundaries to segment salient video object. Zhang *et al.* [5] present a layered directed acyclic graph based object model for optimized video segmentation to extract primary video objects. Ma and Latecki [21] addresses the video object segmentation problem as a problem of finding maximum weight clique in a weighted region graph, which is expected to have high objectness score and share similar appearance. Wang *et al.* [22] offers a strong object-level cue for unsupervised video segmentation in a sequence. Shen *et al.* [23] propose super-pixel segmentation using density-based clustering with noise algorithm. Both object models and temporal-spatial cues are based on the assumption of smooth object appearance changing, therefore the mentioned methods cannot always deal with tempestuous variance of illumination well. Apart from these, trajectories can also be utilized in object segmentation. Shen *et al.* [24] propose to use submodular optimization for better motion segmentation in videos based on trajectory clustering.

C. Image Segmentation

Semi-supervised image segmentation algorithms are highly related to our work, where annotated foreground and background labels are required as input. Shen *et al.* [25] address the problem by solving the cost function of Laplacian graph energy in interactive segmentation. The image segmentation algorithm of Arbelaez *et al.* [26] consists of generic machinery for transforming the output of contour detector into hierarchical region tree. Zhang *et al.* [27] propose a weakly-supervised image segmentation approach by learning distribution of structural superpixel sets. Gao *et al.* [28] propose a contour method combining Expectation-Maximization algorithm to segment complicated images. Grady [29] presents random walks for image segmentation, where users are involved to give seeds. Dong *et al.* [30] propose an incremental random walk with added auxiliary nodes using sub-Markov. Shen *et al.* [31] combines random walks and superpixel segmentation with the commute time and the texture measurement. Yang *et al.* [32] embed Markov random field into conventional energy function, and take use of algebraic multigrid and sparse field method to decrease computing domain. Li *et al.* [33] propose lazy snapping, which is based on graph-cut revised by preprocessing using clusters and contours. Gong *et al.* [34] formulate the fine-structured object segmentation as a label propagation problem on an affinity graph. In the recent years, deep learning methods are introduced into segmentation tasks, most of which focus on capturing the spatial and temporal saliency or consistency [35], [36]. In general, graph-cut based methods outperform other semi-supervised methods like random walks [29]–[31] in time efficiency.

III. Illumination-Invariant Video Cut-Out

A. Approach Overview

We present a new video cut-out approach, and we show it in Fig. 2. Preliminary theory of our algorithm is the OSF and the corresponding illumination invariant features (IIF) we propose, which takes into account contributions of all pixels from eight directions. We rely on the combination of two tasks that extracts foreground object: tracking and

semi-supervised image segmentation. Algorithm 1 gives the processing steps for a subsequence of frames. The foreground region is segmented using the proposed unified framework, which integrates IIF based multi-region tracking and segmentation techniques. The input video is divided into several subsequences. The input consists of input image sequence $\mathbb{I}$, the number of seeds T, iteration number of dilation and erode σ_1 and σ_2 as well as the border length of the search region ω. In the first step, the foreground region is estimated by computing optical flow between the first two frames and then the seeds. The next step contains seeds update and image segmentation, where we update the seed location based on tracking, and later we integrate OSF into graph-cut to segment foreground object. Finally, the foreground region $\mathcal{O}$ is obtained in each frame.

Algorithm 1 Illumination-Invariant Video Cut-Out Algorithm

Input: Image sequence $\mathbb{I}$, the number of seeds T, the iteration number of dilation and erode σ_1 and σ_2, the number of frames to process in a subsequence ϵ, border length of the seeds search region ω

Output: Segmented foreground region $\mathcal{O}$

1: Compute the OSF component using Eq. (4);
2: Compute the IIF of all frames using Eq. (5) in the pre-processing step;
3: Estimate foreground object region in the first frame of $\mathbb{I}$ using Eq. (11);
4: Generate foreground and background seeds $\mathbf{S}$;
5: **for** each $i = 2 \ldots \epsilon$ **do**
6: Use the multi-region based IIF to track the object;
7: Refine the seeds location using Eq. (14);
8: Segment foreground object and assign the result to $\mathcal{O}^i$;
9: **end for**
10: Return $\mathcal{O} = (\mathcal{O}^i)_{i=1}^{\epsilon}$;

B. Octagon Sensitive Optimization

The traditional histogram is a 1D array, and each of its value represents the frequency of occurrence of intensity values. Let $\mathbf{I}$ denote a grayscale image, and W is the number of pixels in image $\mathbf{I}$. The image histogram $\mathbf{H}$ of image $\mathbf{I}$ is a B dimensional vector which can be defined as:

$$\mathbf{H}(b) = \sum_{q=1}^{W} Q(\mathbf{I}_q, b) \tag{1}$$

where b is an integer between 1 and B, B is the number of bins, $Q(\mathbf{I}_q, b)$ is equal to 1 when the intensity value $\mathbf{I}_q$ of pixel q belongs to bin b, otherwise it is 0. A local histogram, on the other hand, is a 2D array which records statistics within a local region and is computed at each pixel location [17], [18]. The computational complexity of local histograms can be reduced using integral histograms [17], [37]. Let $\mathbf{H}_p^{\mathbf{I}}$ denote the integral histogram computed at the pixel p in image $\mathbf{I}$, then we have $\mathbf{H}_p^{\mathbf{I}}(b) = \sum_{q=1}^{p} Q(\mathbf{I}_q, b)$. For simplicity, let $\mathbf{I}$ denote a 1D image. It can be computed on the basis of the previous integral histogram computed at pixel $p-1$ to reduce computational complexity [37], and it is calculated as:

$$\mathbf{H}_p^{\mathbf{I}}(b) = Q(\mathbf{I}_q, b) + \mathbf{H}_{p-1}^{\mathbf{I}}(b) \tag{2}$$

In Eq. (2), the computational complexity of $\mathbf{H}_p^{\mathbf{I}}$ is $O(B)$ at each pixel location. $\mathbf{H}_p^{\mathbf{I}}$ contains contributions of itself and all previous pixels from pixel 1 to pixel $p-1$.

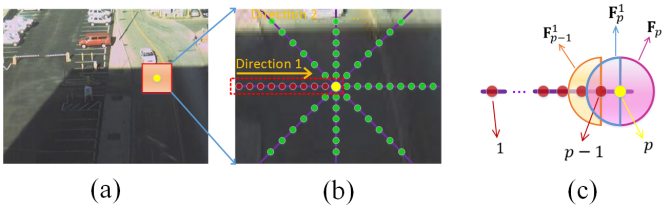


Fig. 3. Computing OSF component $\mathbf{F}_p^1$ in direction 1. (a) The original image, where the red rectangle represents the local region, and the yellow circular is the pixel at which OSF is computed. (b) The local region which shows how to compute OSF from 8 directions. (c) Select a single line in direction 1 from the input image as a 1D image. The orange semicircle represents the OSF component computed at pixel $p-1$, the blue semicircle is $\mathbf{F}_p^1$ at pixel p in direction 1 which can be computed with the previous part $\mathbf{F}_{p-1}^1$ times by α, and the magenta circular is the summing OSF vector at p in Eq. (5).

In this paper, we propose the Octagon Sensitive Filtering (OSF), which is a 2D array considering both spatial and color information of pixels. The OSF at each pixel location is a 1D vector which is the sum of 8 vectors computed from different directions. For the sake of simplicity, we show how to compute the OSF component in one direction. Let $\mathbf{I}$ be a 1D image, which is selected from a single line of the original 2D image in the direction θ, and θ is an integer between 1 and 8 which represent directions including left, top-left, up, top-right, right, bottom-right, down and bottom-left as Fig. 3(b) shows. Note that the selected line ends to the pixel where we compute the OSF component.

Denote the integral value of pixel contribution at pixel p in the direction θ on $\mathbf{I}$ as:

$$\mathbf{F}_p^{\theta}(b) = \sum_{q=1}^{p} \alpha^{|p-q|} \cdot Q(\mathbf{I}_q, b) \tag{3}$$

where $\alpha \in (0, 1)$ is a parameter controlling the decreasing factor from pixel p to q, $|p-q|$ is the distance of pixel location between pixel p and q. Similar to the integral histogram computed in Eq. (2), we can compute $\mathbf{F}_p^{\theta}$ based on the result computed at the previous pixel:

$$\mathbf{F}_p^{\theta}(b) = Q(\mathbf{I}_p, b) + \alpha \cdot \mathbf{F}_{p-1}^{\theta}(b) \tag{4}$$

$\mathbf{F}_p^{\theta}(b)$ is α attenuation of their previous value added by 1 if intensity value of p is in the bin b, and the initial value of them are set to 0. In this way, only B multiplication and addition operation steps are required at each pixel, and the computational complexity of $\mathbf{F}_p^{\theta}$ is $O(B)$ per pixel. Similar to the local histogram, $\mathbf{F}_p^{\theta}$ can be regarded as the integral value of contributions of all pixels in the 1D image $\mathbf{I}$ as described in Fig. 3(c) and the weight of contribution decreases exponentially with respect to the distance between pixels.

We select the line from 8 different directions to obtain the 1D image as shown in Fig. 3(b), so that contributions of pixels

are synthesized. We define the OSF $\mathbf{F}_p$ in bin b as:

$$\mathbf{F}_p(b) = \sum_{\theta=1}^{8} \mathbf{F}_p^{\theta}(b) - 7 \cdot Q(\mathbf{I}_p, b) \tag{5}$$

In Eq. (5), the summation of vectors combines contribution of pixels in 8 directions. Because each OSF component takes into account the contribution of pixel p itself, the integrated OSF in bin b should be the summation value minus $7 \cdot Q(\mathbf{I}_p, b)$. On the basis of Eq. (4), the computational complexity of $\mathbf{F}_p$ is $O(B)$. Hence, the total computational complexity of OSF is $O(PB)$ for all pixels in the input 2D image.

We introduce how to compute OSF based on intensity values of an image. However, most input images are color images and we need to preprocess the images before computing OSF. It is feasible to convert the input image to grayscale image and then computing OSF on it, but this strategy is not distinctive enough for some vision tasks [18]. To exploit color images, we first quantize the colors to 12 values in each channel. In a real world image, only a small portion of color space is used. On the basis of the discovery, we choose the most frequently occurring 64 colors from the quantized colors, and the remaining colors are replaced by one of the chosen colors with the least L2-Norm distance to it. This method is effective enough to cover over 80% the region of an image even in a complex scene. Though only 64 bins are utilized to represent an image, the strategy is still robust enough for most real world cases, and using quantized colors is more distinctive than using intensity to compute OSF.

Histograms are normalized in most cases in practice. It is necessary to do summation operation over all bins for obtaining the normalization factor in the normalization step. Denote n_p as the normalization factor at pixel p:

$$n_p = \sum_{b=1}^{B} \mathbf{F}_p(b) = \sum_{q=1}^{P} \alpha^{|p-q|} \cdot \sum_{b=1}^{B} Q(\mathbf{I}_q, b) = \sum_{q=1}^{P} \alpha^{|p-q|} \tag{6}$$

A brute-force implementation of n_p is $O(B)$. Nevertheless, we compute n_p recursively like Eq. (4) and Eq. (5). The normalization factor component in direction θ is $n_p^{\theta} = 1 + \alpha \cdot n_{p-1}^{\theta}$, and $n_p = \sum_{\theta=1}^{8} n_p^{\theta} - 7$. Thus, n_p can be computed in $O(1)$.

An image block can be matched against another block based on the Earth Mover Distance (Wasserstein metric), which is the L-1 distance between their cumulative normalized histograms [38]. The OSF we propose can be normalized with the normalized factor n_p, and we define the OSF difference between pixel p and q as the summation of cumulative normalized histogram difference in all bins. The two local regions which are centered at pixel p and q respectively can be computed as:

$$\mathcal{D}(p, q) = \sum_{b=1}^{B} |\mathcal{C}_p(b) - \mathcal{C}_q(b)| \tag{7}$$

In Eq. (7), $\mathcal{C}_p$ and $\mathcal{C}_q$ are cumulative histograms of normalized OSF $\mathbf{F}_p$ and $\mathbf{F}_q$, which is defined as: $\mathcal{C}(b) = \sum_1^B \mathbf{F}_p(b)$.

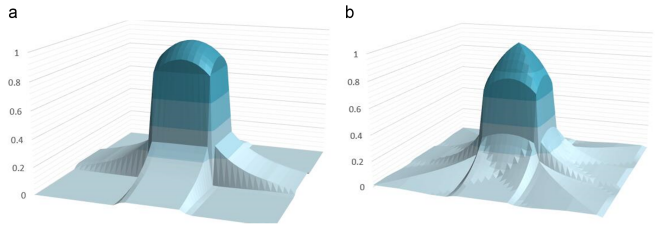


Fig. 4. Comparison of [17] and OSF on describing local components. (a) The contribution of pixels in the local region by LSH [17] distance. (b) The contribution of pixels by OSF distance of Eq. (5).

We adapt the fragment-based method [39] to represent the image block with multiple overlapping regions and use them to compute the difference between image blocks. The regions are used for region-to-region matching where the spatial relationship of the regions is fixed [17]. A region is weighted based on its location to the block center based on a weighting kernel. Thereafter, a vote map can be obtained after the matching scores of all regions within the block are computed, and we accumulate all the votes using least-median-squares estimator as the distance between two image blocks. As Fig. 4 shows, the proposed feature takes into account more factors of central pixels rather than locality sensitive histogram [17].

Decomposition of intrinsic images can be utilized to obtain illumination invariant component. However, it is an ill-posed problem which has two unknowns. The problem can be solved by specifying constrained cues including constant reflectance and illumination with the assumption of local reflectance consistency in [40]. We extract dense illumination-invariant feature based on OSF, which is a transform to convert an image into a new one where pixel value is invariable when illumination changes. Denote $\mathbf{I}_p$ and $\mathbf{I}'_p$ as the pixel intensity value before and after an affine illumination transformation, we have:

$$\mathbf{I}'_p = \mathcal{A}_p(\mathbf{I}_p) = a_1^p \mathbf{I}_p + a_2^p \tag{8}$$

where a_1^p and a_2^p are two parameters of affine transformation $\mathcal{A}_p$. Let b_p denote the bin corresponding to $\mathbf{I}_p$. The sum of values of $\mathbf{F}_p$ which reside in $[b_p - r_p, b_p + r_p]$ is the illumination invariant feature:

$$\mathcal{L}_p = \sum_{b=b_p-r_p}^{b_p+r_p} \mathbf{F}_p(b) \tag{9}$$

where r_p controls the interval of integration value $\mathcal{L}_p$. If r_p scales linearly with illumination, the new interval parameter $r'_p = a_1^p \cdot r_p$. Similar to Eq. (9), the integrated value $\mathcal{L}'_p$ in the new illumination condition is equal to the sum of values of the new OSF $\mathbf{F}'_p$ which reside in $[b'_p - r'_p, b'_p + r'_p] = [a_1^p b_p + a_2^p - a_1^p r_p, a_1^p b_p + a_2^p + a_1^p r_p] = [a_1^p (b_p - r_p) + a_2^p, a_1^p (b_p + r_p) + a_2^p] = [\mathcal{A}_p(b_p - r_p), \mathcal{A}_p(b_p + r_p)]$. With additional assumption that the illumination change is locally smooth so that the transform is the same for the pixels in the local region. In this case, $\mathcal{L}'_p$ is equal to $\mathcal{L}_p$ if ignoring quantization error because OSF records the weighted sum of occurrence of pixel colors in different bins. This means $\mathcal{L}_p$ is independent of affine illumination change and can be used

as illumination invariant features. The invariance still holds if integrating from bin 1 to B in Eq. (9). Nevertheless, to reduce quantization error, we use a soft interval, and refine Eq. (9) as:

$$\mathcal{L}_p = \sum_{b=1}^{B} \exp\left(-\frac{(b-b_p)^2}{2\max(0.1, r_p)^2}\right) \cdot \mathbf{F}_p(b) \tag{10}$$

where we let $r_p = 0.1 \cdot |\mathbf{F}_p(b) - \sum_{b=1}^{B} \mathbf{F}_p(b) \cdot b|$. The integrated value in Eq. (10) changes within a small range even under dramatic illumination change. The new feature $\mathcal{L}_p$ inherit the advantage of n_p, and further optimize n_p by adding illumination invariant features. Therefore, the illumination invariant features in Eq. (10) are used as inputs for seed points location update and segmentation task in Section III-C and Section III-D instead of n_p computed in Eq. (6).

C. Initial Object Regions Estimation

We first estimate the foreground object regions using the IIF-based optical flow computing, and cut out the frames with the unified framework of seed updating and segmentation. To deal with the large displacement of object and avoid incorrect estimation when object appearance changes dramatically due to illumination, we introduce IIF into variational motion estimation [44], [45] as matching terms to combine both advantages of the illumination invariant features of IIF with accurate dense motion estimation and large displacement correspondence of the variational model. The local spatial information is maintained well by using IIF, while many other descriptors will estimate incorrectly.

Let $\mathbf{w}$ denote the optical flow field, and Ω be the image domain. So that the optical flow field can be the function: $\mathbf{w} : \Omega \rightarrow \mathbb{R}^2$. We follow the approach [44] and define a variational model for optical flow field, which is the weighted sum of various terms:

$$E = E_{color}(\mathbf{w}) + \gamma E_{grad}(\mathbf{w}) + \lambda E_{smooth}(\mathbf{w}) + \beta E_{match}(\mathbf{w}, \mathbf{w}_1) + \beta E_{desc}(\mathbf{w}_1) \tag{11}$$

In Eq. (11), γ, λ and β are controlling parameters, E_{color} and E_{grad} penalize deviations from gray value constancy assumption and gradient constancy assumption, respectively. $\mathbf{w}_1$ is the correspondence vector obtained by descriptor matching. The sum of E_{color} and E_{grad} is the data term, E_{smooth} is the smoothness term which penalizes the total variation of the flow field to enforce regularity. In this paper we use the method in [45] to compute the data term and smoothness term. E_{color} is the square sum of gray value difference of the original pixel gray value and the mapping pixel gray value. Similar to E_{color}, E_{grad} is that of magnitude gradient value instead of grayscale value. E_{smooth} is the sum of square sum value of two vector components on each pixel. The optical field can be computed by minimizing Eq. (11).

Pixel correspondences from descriptor matching are integrated into the variational model by adding two matching terms E_{match} and E_{desc}, and we use OSF as the descriptor for matching. The proposed illumination-invariant features based on OSF can also be utilized to exploit the images where illumination changes dramatically. The problem of computing the minimal E_{desc} can be converted into block matching problem in the image domain Ω:

$$E_{desc}(\mathbf{w}_1) = \int_{\Omega} \delta(p)\phi(p)\psi(\mathcal{D}(p, p + \mathbf{w}_1(p)))\mathrm{d}p \tag{12}$$

where $\delta(p)$ is 1 if the OSF descriptor exists at p otherwise 0, $\psi(s^2) = \sqrt{s^2 + 0.001^2}$ is the robustness function, $\mathbf{w}_1$ is the candidate block. $\phi(p)$ is the weight of E_{match}, which is reflected by OSF value of the best and worst match block. E_{match} shows similarity between two blocks $\mathbf{w}(p)$ and $\mathbf{w}_1(p)$. In our experiments, the three tradeoff parameters γ, λ and β of the variational model are set to 10, 3 and 200, respectively. However, the target boundary estimated by optical flow is not accurate to cover the moving foreground object. Besides, the foreground region may be falsely estimated when the target partially remains static. In order to refine the optical flow and foreground-background labels, we use a novel algorithm Fast Object Segmentation [4] to identify the foreground region. The algorithm overcomes the shortcomings of inaccurate boundary estimation due to partially static region of foreground, thus an accurate object region can be obtained.

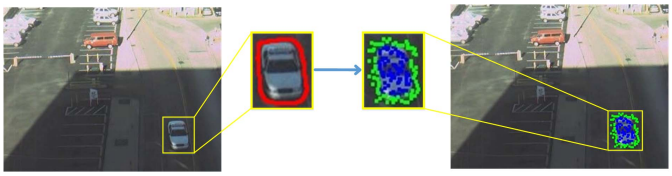


Fig. 5. The formulation for generating seeds. We first estimate foreground region which is surrounded with red color, and obtain the search window. The foreground and background seeds are shown in blue and green points, respectively.

The last work in this section is to generate foreground and background seeds and a search window for tracking as is illustrated in Fig. 5. Seeds are set of pixels which are labeled foreground or background, and the search window which is marked with the yellow rectangle indicates the size and location of target. The window is set big enough to cover the target, which will be used for visual tracking in the next section. Denote T as the number of total number of seeds. We keep the number of object region seeds 1.2 times that of background seeds, and both 30 percent of them are selected uniformly from the region which are the difference between erosion or dilation object image and the original image in experiments, respectively. The iteration times of erosion and dilation are denoted as σ_1 and σ_2 which are determined by the object size.

D. Seeds Update and Objects Segmentation

In each loop, it is required to update the seeds location and then extract the object based on the seed points. As described in Section III-B, the difference between blocks is computed by the vote map which is based on pixel OSF distance. We search within the neighbor area of the window, and the matching result is the candidate block with the lowest joint score. The marginal regions are given less weight, which facilitates robustness to occlusion. We illustrate how to update

Fig. 6. Tracking with OSF on birdfall sequence in SegTrack. (a) The initial frame #18, where object is covered by search window shown in red. The orange overlapped circles show the multi-region with which to measure block dissimilarity. (b) Tracking results in frame #31. (c) Tracking results in frame #40. In (b) and (c), results of TLD [41], KCF [42], BOOSTING [43], LSH [17] and OSF tracker are indicated in cyan, yellow, blue, green and purple rectangles respectively.

the window with OSF using multi-region in Fig. 6, and give the contrast experiment between OSF tracker and other effective trackers. Note that only a few regions are displayed for the sake of clarity. It can be observed that TLD [41] and KCF [42] trackers cannot deal with fuzzy object boundaries and appearance change due to illumination variance, while our tracker obtains the more accurate tracking results than other trackers. Our tracker can obtain more precise results compared with LSH [17].

If ignoring the pose and scale change of foreground object, the seeds and the window location can be updated after tracking only with the object displacement. Let the location of the window center in frame i be $\mathbf{L}_i$, and let $\mathbf{S}_i^p$ denote the location of seed p in frame i, then:

$$\mathbf{S}_i^p = \mathbf{S}_{i-1}^p + (\mathbf{L}_i - \mathbf{L}_{i-1}) \tag{13}$$

where $\mathbf{L}_i - \mathbf{L}_{i-1}$ is the displacement vector which represents moving from $\mathbf{L}_{i-1}$ to $\mathbf{L}_i$, and update the seed location based on window displacement. Owing to object deformation, the seeds would be assigned with false foreground or background labels in this strategy. If the pixel color remains unchanged, a convenient way to reduce seeds updating error is to find a pixel in the local region with minimal color difference. Nevertheless, updating results based on this assumption is not always robust due to occlusion and illumination change. Updating the seeds with tracking displacement and minimal color distance can fit in well with scale change of object, but would also lead to incorrect update in frames where the object appearance changes partially. In this paper we use the minimal IIF distance instead.

The proposed algorithm for seeds updating is illustrated in Fig. 7(c). Let ω be the border length of the seeds search region as in Fig. 7(c), the new seed location should be:

$$\mathbf{S}_i^p = \mathbf{S}_{i-1}^p + (\mathbf{L}_i - \mathbf{L}_{i-1}) + \arg\min_{\mathbf{v}} \mathcal{D}(p, q) \tag{14}$$

where q denote pixels which are moved from pixel p by the updating vector $\mathbf{v}$. Let $\mathbf{v} = (u, v)$, where u and v are integers in $[-\frac{\omega-1}{2}, \frac{\omega-1}{2}]$. The seeds updating is robust to dramatic object pose or scale changes. The seeds can be correctly updated by setting ω greater if there is large object displacement. The seeds update strategy can handle the appearance and scale change of object well, however, mistakes of seeds update still occur accidentally, which is a disaster for subsequent object

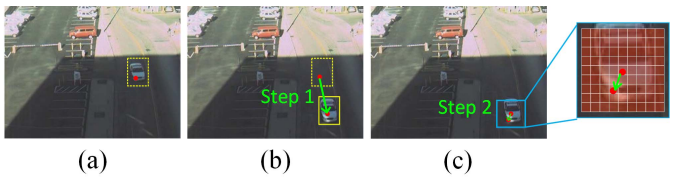


Fig. 7. The deployed seeds update strategy. (a) The location of seed p in frame $i-1$ is marked red, and the dashed yellow rectangle represents the search window in frame $i-1$. (b) Firstly we move the seed based on the tracking displacement $\mathbf{L}_i - \mathbf{L}_{i-1}$. The rectangle with solid yellow line is the search window in frame i. (c) Secondly we obtain the new pixel with least OSF difference to the intermediate result in (b) within the region marked by dark orange squares.

segmentation. In view of this problem, the new seeds and the search window are generated every frame to adapt them to shape and illumination change. After the window and the seeds have correctly generated, seeds location is updated based on tracking and OSF distance in pixel level. Fig. 8 shows the seeds update results of our strategy. In sequence (a) and (b), foreground objects both undergo illumination change. The light is suddenly turned on in (a) and the car enters into shadow area in (b). It can be observed that the seeds are kept in correct location to present foreground location. It can also be discovered in (c) that our seeds update strategy can also fit in object deformation.

With the updated foreground and background seeds, the foreground region can be segmented based on graph-cut methods. Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ denote an image, where $\mathcal{V}$ is the set of all pixels and $\mathcal{E}$ is the set of adjacency relationships between neighborhood pixels, respectively. The foreground region is segmented by minimizing Gibbs energy:

$$E_G(X) = \sum_{p \in \mathcal{V}} E_1(x_p) + \eta \sum_{(p,q) \in \mathcal{E}} E_2(x_p, x_q) \tag{15}$$

where $E_1(x_p)$ and $E_2(x_p, x_q)$ are the likelihood and prior energy, respectively [33]. Segmenting the foreground region is a binary labeling problem, and for each node $p \in \mathcal{V}$, x_p is equal to 1 if node p is assigned with the foreground label, and 0 otherwise. We use the watershed algorithm to pre-segment the image into small regions, then denote $\mathcal{V}$ as the set of the small regions instead of pixels and denote $\mathcal{E}$ as the set of arcs connecting adjacent regions. We compute the likelihood energy E_1 using the method in [33], which encodes the cost of color similarity. We integrate the color gradient between nodes into color difference to represent E_2 and use the proposed OSF as the additional term in order to deal with ambiguous boundaries. We compute the prior energy E_2 as:

$$E_2(x_{\mathbb{P}}, x_{\mathbb{Q}}) = \frac{|x_{\mathbb{P}} - x_{\mathbb{Q}}|}{1 + \mathcal{T}(\mathbb{P}, \mathbb{Q}) + \mu \cdot \mathcal{D}(p, q)} \tag{16}$$

where p and q denote the center pixel of small regions $\mathbb{P}$ and $\mathbb{Q}$ which are generated by pre-segmentation, respectively. $\mathcal{T}(\mathbb{P}, \mathbb{Q})$ is the L2-Norm distance of RGB difference between mean colors of $\mathbb{P}$ and $\mathbb{Q}$. We combine the color and proposed IIF matching with the weight parameter μ, and set it to 50 in the experiments. E_2 penalizes adjacent nodes which are assigned with different labels. Different from RGB distances, the proposed OSF takes spatial information

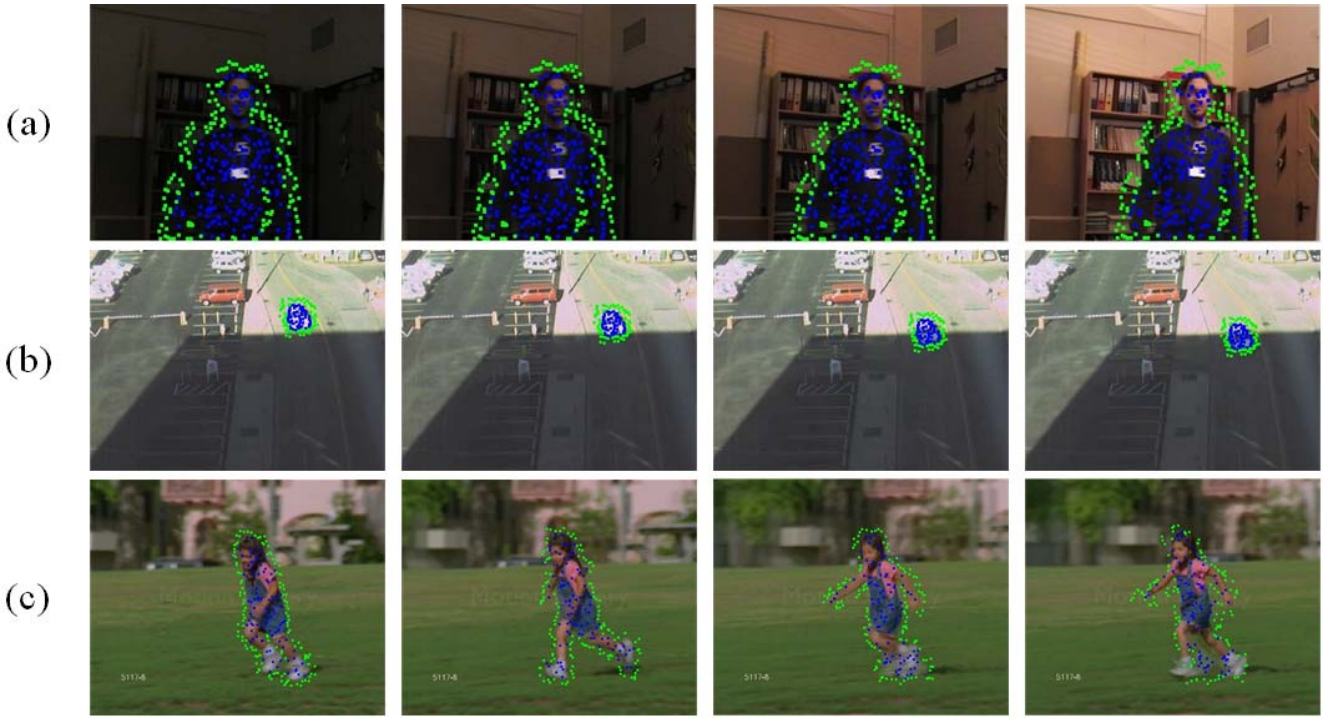


Fig. 8. Examples of seeds update results. Images from (a) to (c) are frames selected in man, car and girl sequence respectively. The points marked in green and blue represent the background and foreground seeds, respectively.

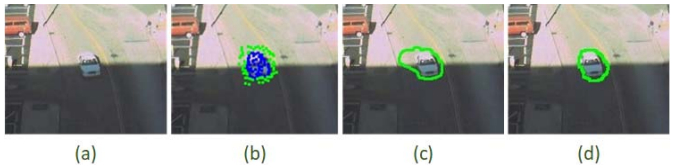


Fig. 9. Comparison on the traditional graph-cut and the improved method by integrating the IIF. (a) is the zoomed-in orignal frame. (b) is the seed points image. (c) and (d) are the results of graph-cut and ours, respectively.

into consideration, thus adjacent regions will have higher probability to be assigned with the same label to the seed region. In Fig. 9, we show the results of traditional graph-cut [33] and ours. Because graph-cut only takes into account the color information while we also try to achieve the space consistency on the basis of seed points, our method can obtain better segmentation results when the boundary is fuzzy. This scheme has been shown robust to ambiguous and low-contrast boundaries since some foreground and background seeds are obtained near the object boundary when seeds are generated. We can cut out the video shots quickly because all pixels outside the search window are undoubtedly assigned with background labels thus decreasing computation domain, and the acceleration effect will be more significant if the object size is relatively small.

IV. Experimental Results

We used 5 sequences of SegTrack [46] and 13 sequences of Visual Tracking via Locality-sensitive-histogram Data (VTLD) [17] to perform evaluations, which offer various challenges for moving object extraction. In the two datasets, a video is excluded if it contains several moving objects while only one of them is labeled in the ground-truth. All tested video shots are handled on an Intel Core 5, 2.5 GHz PC with 8 GB RAM, and our method is implemented using C++ on visual studio.

Fig. 10. Results for the effect of the seeds number T on board, car and lemming sequence in VTLD [17] evaluated by mIU (mean Intersection-over-Union). We set T to various numbers from 2 to 117 for the sequences.

A. Experiments on VTLD

We use video shots in VTLD [17] to evaluate the video cut-out effect of our method. At first, we use car and skiing sequences in VTLD [17] to assess the performance of our method. In different situations where image size or object size varies, and whether rotation, deformation and illumination

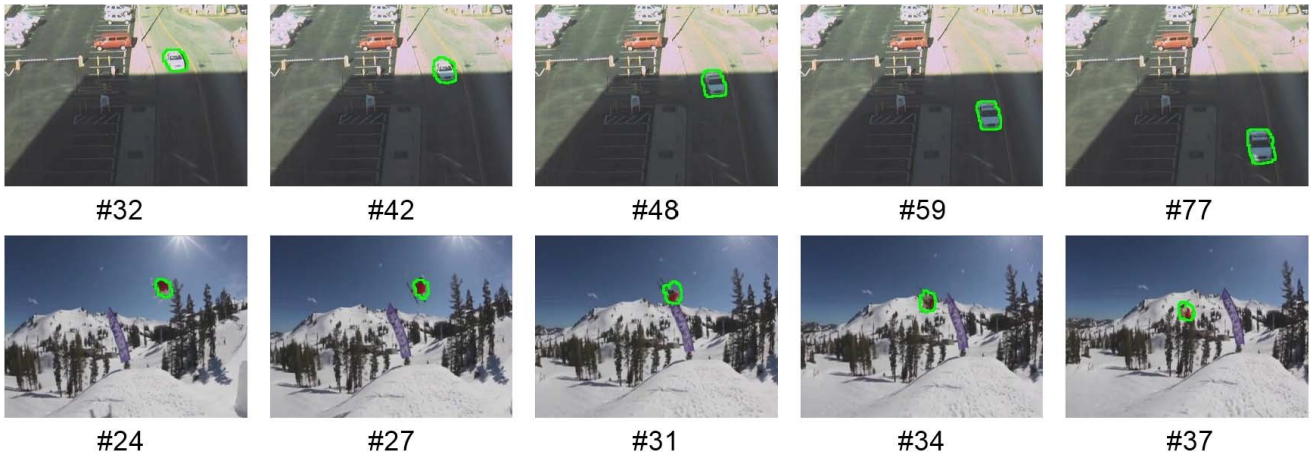


Fig. 11. Qualitative video cut-out results of our algorithm on car and skiing sequence. There are challenges of illumination change caused by cast shadow and large object displacement in the two sequences, respectively.

change exist or not, we have to set parameters to different values. We give an example in Fig. 10 of car sequence in VTLD [17]. We try the attenuation value α by setting it to many different values, and find that setting it from 0.015 to 0.04 is appropriate. Therefore, α is fixed at 0.03 in all experiments. In the step of generating local search window and seeds, iteration times of erosion and dilation σ_1 and σ_2 are set as 5 for most cases. If the two parameters are set rather little, the seed points would be too dense and lack of discrimination. On the other hand, the false points would be generated if σ_1 and σ_2 are set too large. Experiments show that they can be adjusted from 3 to 12 by the user, which is determined by the image resolution. Here we take car sequence as an example. We set $\epsilon = 10$ and $\omega = 7$ to fit in appearance change of object. η is set to 50 when segmenting. The performance of our algorithm with different seeds number T is shown in Fig. 9. Theoretically, better results would be achieved when the number of seeds grows larger, because it means more guidance is given for semi-supervised segmentation. However, it may not be necessarily appropriate because of the amplification effects of false seeds estimating in initial steps. From Fig. 9, we can find that mIU tends to be stable when T is greater than 20. For instance, the mIU peaks when T is set to 85 on the car sequence. Similar to σ_1 or σ_2, T also depends on the image resolution and object scale. We conduct many experiments where we use various T value. We conclude that good segmentation results can be obtained for most cases if T is set to default value 80.

Representative illumination challenging frames from the car and skiing sequence are shown in Fig. 10. The selected video shot has 80 frames at 320×240 resolution. The car in the video moves forwards to camera gradually whose appearance has dramatic change due to shadows, and that is where difficulty lies. In Fig. 10, the reasonable objects are cut out when cast shadow and large displacement exist. Our method outperforms the state-of-the-art methods in accuracy as is shown in Fig. 11. In Fig. 12, we further give comparison experiments results in VTLD [17] evaluated by PER (average Per-frame Error Rate) with VOS [5], SAG [10], FOS [4], SLD [8], CVS [47] and ours by annotating ground-truth images in a subset of

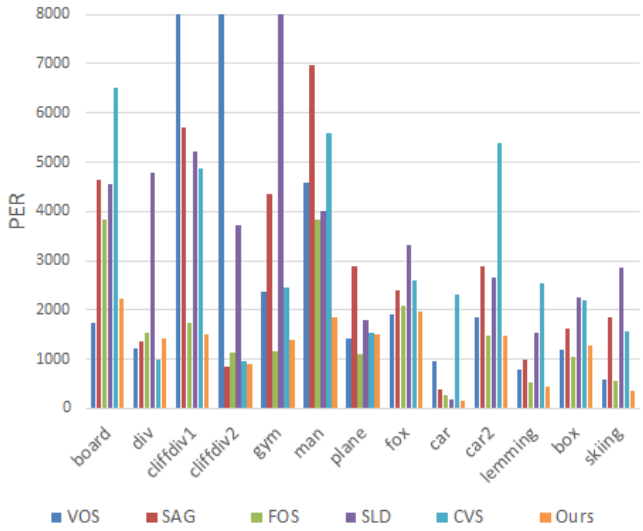


Fig. 12. Quantitative comparison of VOS [5], SAG [10], FOS [4], SLD [8], CVS [47] and ours on VTLD [17] dataset. Note that PER is set to 8000 when it is over 8000 or there are no available output images.

frames. As shown in Fig. 13, our method presents better performance compared with other implementations when handling sequences where the cast shadow exists. The proposed algorithm also performs well in scenarios where illumination varies, such as abrupt dramatic illumination change in man sequence. Furthermore, our method performs comparatively well with large displacement as shown in cliffdiv1 and skiing sequence. In other scenarios where object deforms, scale changes or rotates, our method still performs no worse than state-of-the-art algorithms like cliffdiv2 and lemming.

B. Experiments on SegTrack

We test our algorithm on 5 sequences from SegTrack [46] which offers various challenges including non-rigid deformations, fast camera motion and low contrast between foreground and background. The penguin video in this dataset is discarded because there are several moving foreground objects in these frames. We show cut-out results in Fig. 14 where ground-truth in pixel level is provided in every frame. We can observe

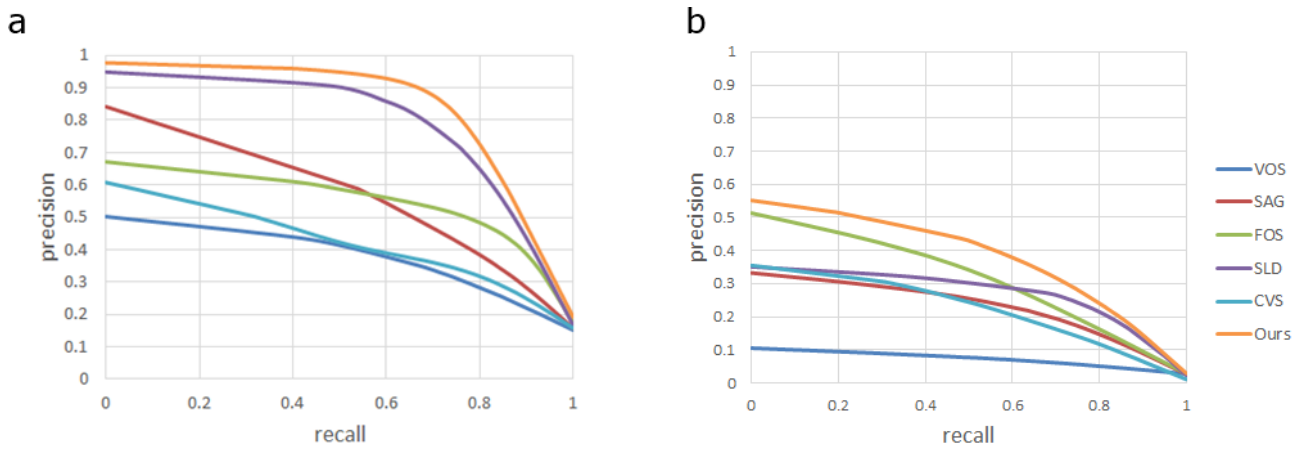


Fig. 13. The precision-recall curve showing the cut-out accuracy when evaluating the performance of VOS [5], SAG [10], FOS [4], SLD [8], CVS [47], and ours. (a) and (b) are the precision-recall curves for the car and skiing sequence, respectively.

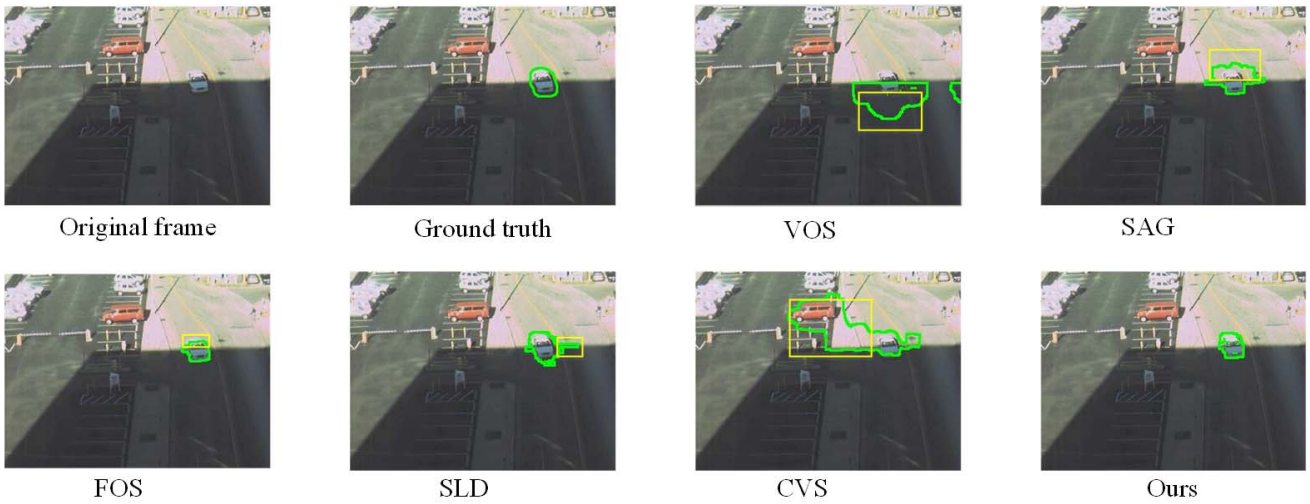


Fig. 14. Qualitative segmentation result comparison for #42 frame in car sequence of our methods with the results of VOS [5], SAG [10], FOS [4], SLD [8], and CVS [47]. The manifestly false segmented region is marked with a yellow rectangle.

TABLE I

COMPARISON OF AVERAGE RUN TIME PER-FRAME (SECONDS) FOR THE 5 VIDEOS IN SEGTRACK [46]

Video	Resolution	[17]	[22]	[16]	[20]	[59]	Ours
birdfall	259 × 327	> 30	11.5	9.8	> 30	9.5	4.8
cheetah	320 × 240	> 30	12.5	10.4	> 30	10.3	6.8
girl	400 × 320	> 30	11.6	11.1	> 30	9.5	6.4
monkeydog	320 × 240	> 30	10.5	10.3	> 30	9.5	6.4
parachute	320 × 240	> 30	8.5	8.5	> 30	10.0	5.1

that our method outperforms others when there is illumination change in birdfall and parachute.

Fig. 15 shows the qualitative comparison between our method and other state-of-the-art methods on SegTrack [46] benchmark. Fig. 16 is the bar graph of quantitative comparison in terms of PER showing the effectiveness of the proposed video cut-out method. Overall, objectness algorithms like VOS [5], FOS [4] and ours. Our method outperform SAG [10] and SLD [8], especially in dynamic background scenario. The proposed video cut-out method is robust to illumination change and rotation as is shown in birdfall and parachute sequence, which can be attributed to the fact that we take local information of seeds into account and the seeds update strategy. Still, our method performs relatively well when the object undergoes scale or pose change. In the girl and cheetah sequence, there is severe occlusion challenge. Our method is based on color and local feature matching. However, it does not focus on handling occlusion. Similar to [48], [49], our approach has relatively satisfying outputs. We also replace IIF in the video cutting out framework with DAISY [14] and HOG [50] in Fig. 16. HOG [50] is effective to deal with optical deformation and slight deformation. DAISY [14] is a dense descriptor and has rotation invariance. We use them to update the seed point location and perform segmentation, and the parameters are reset in the experiments. In Fig. 16, we can find that IIF outperforms these two features on most sequences, especially on birdfall sequence where the brightness changes.

Table I compares the average run time between the proposed approach and other methods on the SegTrack [46] dataset. In Table I, the VOS [5] takes over one hours time

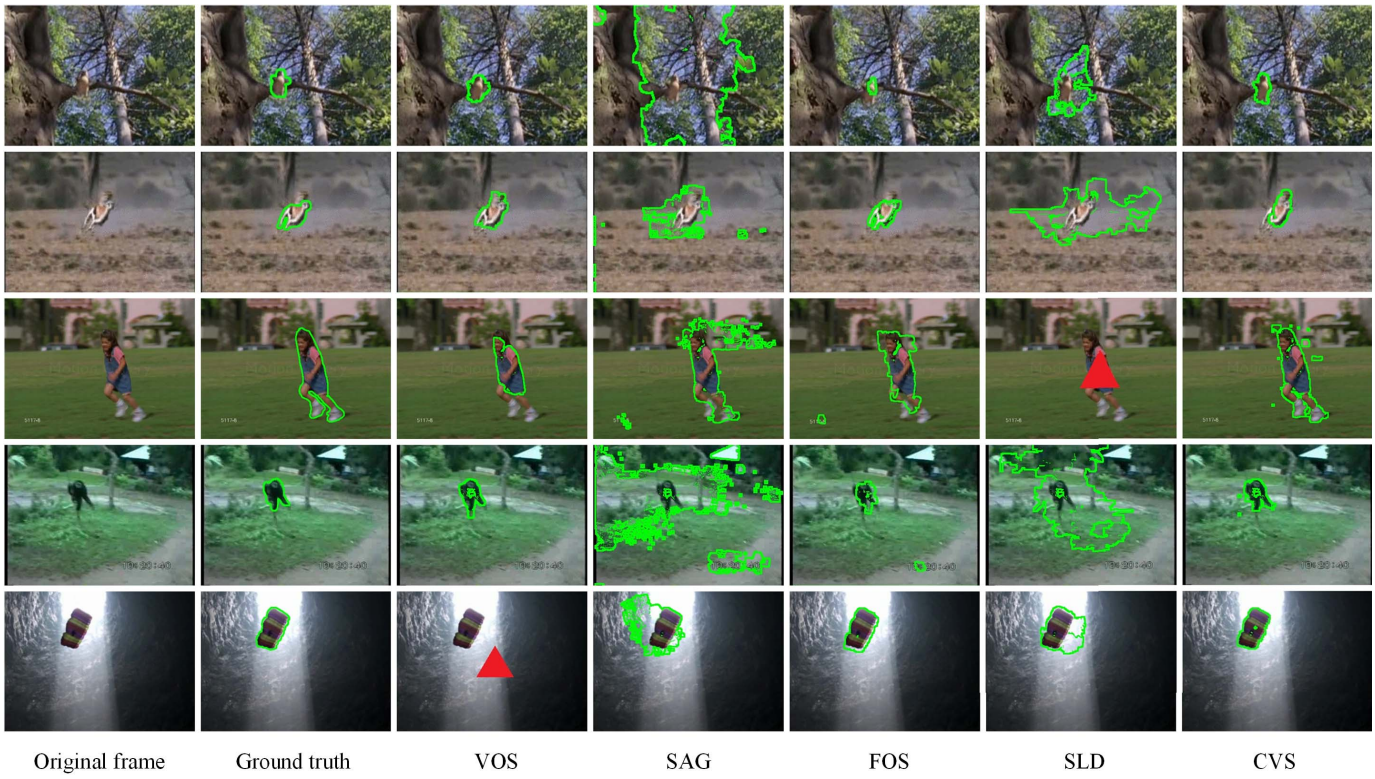


Fig. 15. Qualitative comparison between our method and other state-of-the-art methods on SegTrack [46] benchmark. The first column shows the first frame of birdfall, cheetah, girl, monkeydog and parachute. The second column is the ground truth image used for evaluation. The last six columns are outputs of VOS [5], SAG [10], FOS [4], SLD [8], CVS [47], and ours, respectively. The cutting-out object boundaries are marked green and the dark red triangle means no available outputs.

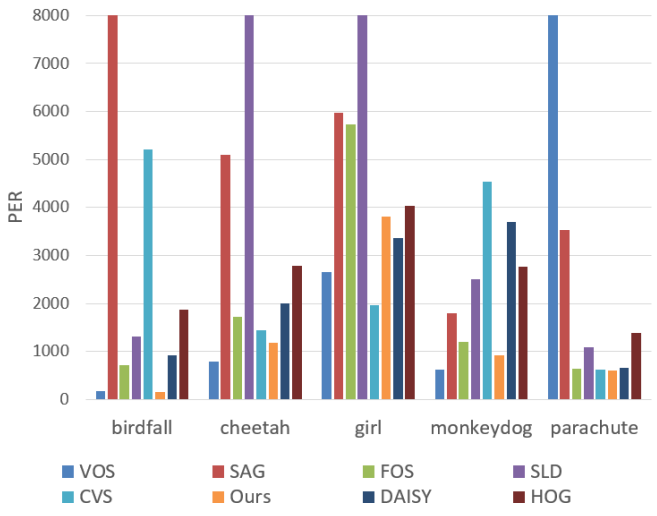


Fig. 16. Quantitative comparison between our method and other state-of-the-art methods of VOS [5], SAG [10], FOS [4], SLD [8], CVS [47], DAISY [14], and HOG [50] on SegTrack [46] benchmark. From left to right are birdfall, cheetah, girl, monkeydog and parachute sequence. We replace OSF and IIF with DAISY [14] to evaluate the effectiveness of the proposed IIF in our framework.

until convergence and get the final segmentation results, and SLD [8] requires hours to finish over-segmentation for handling one image sequence. VOS [5] optimizes object proposals iteratively, which requires massive time cost. Our method outperforms FOS [4], SAG [10] and CVS [47] in time efficiency because the other three algorithms compute the dense optical flow between every two subsequent frames. The reduction in run time is caused by our tracking and segmentation framework. Our method takes around 15 seconds to estimate the foreground moving object when processing the first two frames, and taking 0.4 to 1.5 sec/frame for the rest frames in the subsequence which is determined by target size (0.3 seconds for tracking and seeds updating and the rest for segmentation which depends on the object size). The run time of our method can be further reduced if the foreground object has no significant pose or scale change.

V. Conclusion

In this paper, we have presented a new unsupervised approach for video cut-out, which is robust to illumination change. A local feature OSF and the corresponding illumination invariant features are introduced in the paper. We also propose a new tracking-and-segmentation framework to conduct video cutting-out. We first integrate the OSF into our variational model to estimate the objects regions in the first frame. We then apply an IIF tracker to update the seed locations and further refine the locations in pixel level to accommodate to poses or scales changes of objects. We address the foreground extraction with our IIF-based graph-cut to extract the foreground objects in case of vague boundaries. Various experiments on challenging datasets have shown the robustness of our video cut-out to illumination changes.

Our approach outperforms the existing state-of-the-art methods in the perspective of time efficiency in experiments. We will work on related video editing methods as our future work, including enhancing the latest accurate optical flow techniques to handle incorrect estimation at pixels around target boundaries, and investigating seeds updating skills to adapt to significant pose changes of objects. We will conduct researches on image/video segmentation with deep learning techniques [35], [36].

REFERENCES

[1] B. L. Price, B. S. Morse, and S. Cohen, "LIVEcut: Learning-based interactive video segmentation by evaluation of multiple propagated cues," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 779–786.

[2] E. Borenstein, E. Sharon, and S. Ullman, "Combining top-down and bottom-up segmentation," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, Jun./Jul. 2005, p. 46.

[3] I. Budvytis, V. Badrinarayanan, and R. Cipolla, "Semi-supervised video segmentation using tree structured graphical models," in *proc. IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2751–2764, Jun. 2013.

[4] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1777–1784.

[5] D. Zhang, O. Javed, and M. Shah, "Video object segmentation through spatially accurate and temporally dense extraction of primary object regions," in *Proc. IEEE CVPR*, Jun. 2013, pp. 628–635.

[6] W. Wang, J. Shen, X. Li, and F. Porikl, "Robust video object cosegmentation," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3137–3148, Oct. 2015.

[7] L. Chen, J. Shen, W. Wang, and B. Ni, "Video object segmentation via dense trajectories," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2225–2234, 2015.

[8] C. Li, L. Lin, W. Zuo, W. Wang, and J. Tang, "An approach to streaming video segmentation with sub-optimal low-rank decomposition," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 1947–1960, May 2016.

[9] Y. Liang, J. Shen, X. Dong, H. Sun, and X. Li, "Video supervoxels using partially absorbing random walks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 5, pp. 928–938, May 2016.

[10] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. IEEE CVPR*, Jun. 2015, pp. 3395–3402.

[11] X. Liu, L. Wang, J. Zhang, J. Yin, and H. Liu, "Global and local structure preservation for feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1083–1095, Jun. 2014.

[12] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[13] X. Yang and K.-T. T. Cheng, "Local difference binary for ultrafast and distinctive feature description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 188–194, Jan. 2014.

[14] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.

[15] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM J. Imag. Sci.*, vol. 2, no. 2,/break pp. 438–469, 2010.

[16] K. Simonyan, A. Vedaldi, and A. Zissenman, "Learning local feature descriptors using convex optimisation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1573–1585, Aug. 2014.

[17] S. He, Q. Yang, R. W. H. Lau, J. Wang, and M.-H. Yang, "Visual tracking via locality sensitive histograms," in *Proc. IEEE CVPR*, Jun. 2013, pp. 2427–2434.

[18] S. He, R. W. H. Lau, Q. Yang, J. Wang, and M.-H. Yang, "Robust object tracking via locality sensitive histograms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 5, pp. 1006–1017, May 2017.

[19] W. Wang, J. Shen, and F. Porikli, "Selective video object cutout," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5645–5655, Dec. 2017.

[20] T. Lim, B. Han, and J. H. Han, "Modeling and segmentation of floating foreground and background in videos," *Pattern Recognit.*, vol. 45, no. 4, pp. 1696–1706, 2012.

[21] T. Ma and L. J. Latecki, "Maximum weight cliques with mutex constraints for video object segmentation," in *Proc. IEEE CVPR*, Jun. 2012, pp. 670–677.

[22] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-aware video object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 20–33, Jan. 2018.

[23] J. Shen, X. Hao, Z. Liang, Y. Liu, W. Wang, and L. Shao, "Real-time superpixel segmentation by DBSCAN clustering algorithm," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5933–5942, Dec. 2016.

[24] J. Shen, J. Peng, and L. Shao, "Submodular trajectories for better motion segmentation in videos," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2688–2700, Jun. 2018.

[25] J. Shen, Y. Du, and X. Li, "Interactive segmentation using constrained laplacian optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 7, pp. 1088–1100, Jul. 2014.

[26] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[27] L. Zhang, Y. Gao, Y. Xia, K. Lu, J. Shen, and R. Ji, "Representative discovery of structure cues for weakly-supervised image segmentation," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 470–479, Feb. 2014.

[28] G. Gao, C. Wen, and H. Wang, "Fast and robust image segmentation with active contours and student's-t mixture model," *Pattern Recognit.*, vol. 63, pp. 71–86, Mar. 2017.

[29] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[30] X. Dong, J. Shen, L. Shao, and L. V. Gool, "Sub-Markov random walk for image segmentation," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 516–527, Feb. 2016.

[31] J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for superpixel segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1451–1462, Apr. 2014.

[32] X. Yang, X. Gao, D. Tao, X. Li, and J. Li, "An efficient mrf embedded level set method for image segmentation," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 9–21, Jan. 2014.

[33] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 303–308, 2004.

[34] Y. Gong, S. Xiang, and C. Pan, "Fine-structured object segmentation via neighborhood propagation," *Pattern Recognit.*, vol. 60, pp. 130–144, 2016.

[35] W. Wang and J. Shen, "Deep visual attention prediction," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2368–2378, May 2018.

[36] W. Wang, J. Shen, and L. Shao, "Video salient object detection via fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 38–49, Jan. 2018.

[37] F. Porikli, "Integral histogram: A fast way to extract histograms in Cartesian spaces," in *Proc. IEEE CVPR*, Jun. 2005, pp. 829–836.

[38] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.

[39] A. Adam, E. Rivlin, and I. Shimshoni, "Visual tracking via locality sensitive histograms," in *Proc. IEEE CVPR*, Jun. 2006, pp. 798–805.

[40] J. Shen, X. Yang, Y. Jia, and X. Li, "Intrinsic images using optimization," in *Proc. IEEE CVPR*, Jun. 2011, pp. 3481–3487.

[41] Z. Kalal, K. Mikolajczyk, and J. Matis, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[42] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. ECCV*, 2012, pp. 702–715.

[43] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE CVPR*, Jun. 2006, pp. 260–267.

[44] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.

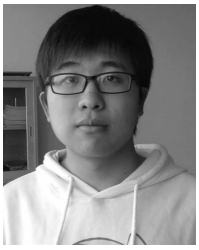
[45] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. ECCV*, pp. 25–36, 2004.

[46] D. Tsai, M. Flagg, and J. Rehg, "Motion coherent tracking using multi-label MRF optimization," *Int. J. Comput. Vis.*, vol. 100, no. 2, pp. 190–202, 2012.

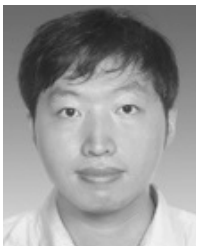
[47] W. Wang, J. Shen, and L. Shao, "Consistent video saliency using local gradient flow optimization and global refinemen," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4185–4196, Nov. 2015.

[48] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, "Occlusion-aware real-time object tracking," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 763–771, Apr. 2017.
[49] J. Shen, D. Yu, L. Deng, and X. Dong, "Fast online tracking with detection refinement," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 162–173, Jan. 2018.
[50] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE CVPR*, vol. 1, Jun. 2005, pp. 886–893.

Zhihua Chen received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China. He is currently a Full Professor with the Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai. His current research interests include image/video processing and computer vision.



Jingye Wang received the B.Eng. degree in computer science from the East China University of Science and Technology, Shanghai, China, where he is currently pursuing the postgraduate degree with the Department of Computer Science and Engineering. His current research interests include illumination-invariant video cut-out, video processing, deep learning, and computer vision.



Bin Sheng received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, China. He is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include image-based rendering, machine learning, virtual reality, and computer graphics.

Ping Li received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, China. He is currently an Assistant Professor with the Macau University of Science and Technology, Macau, China. His current research interests include image/video stylization, big data visualization, GPU acceleration, and creative media. He has one image/video processing national invention patent and an excellent research project reported worldwide by *ACM TechNews*.

David Dagan Feng (F'03) received the M.Eng. degree in electrical engineering and computer science (EECS) from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.Sc. degree in biocybernetics and the Ph.D. degree in computer science from the University of California, Los Angeles, Los Angeles, CA, USA, in 1985 and 1988, respectively. He is currently the Head of the School of Information Technologies, the Director of the Biomedical and Multimedia Information Technology Research Group, and the Research Director with the Institute of Biomedical Engineering and Technology, University of Sydney, Sydney, Australia. He has published over 700 scholarly research papers, pioneered several new research directions, and made a number of landmark contributions in his field. More importantly, however, is that many of his research results have been translated into solutions to real-life problems and have made tremendous improvements to the quality of life for those concerned. He received the Crump Prize for Excellence in Medical Engineering from UCLA. He has served as the Chair in the International Federation of Automatic Control (IFAC) Technical Committee on Biological and Medical Systems, has organized/chaired over 100 major international conferences/symposia/workshops, and has been invited to give over 100 keynote presentations in 23 countries and regions. He is a Fellow the Australian Academy of Technological Sciences and Engineering.