AUTOMATIC OBJECT EXTRACTION IN SINGLE-CONCEPT VIDEOS

Kuo-Chin Lien and Yu-Chiang Frank Wang

Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
{iker, ycwang}@citi.sinica.edu.tw

ABSTRACT

We propose a motion-driven video object extraction (VOE) method, which is able to model and segment foreground objects in single-concept videos, i.e., videos which have only one object category of interest but may have multiple object instances with pose, scale, etc. variations. Given such a video, we construct a compact shape model induced by motion cues, and extract the foreground and background color information accordingly. We integrate these feature models into a unified framework via a conditional random field (CRF), and this CRF can be applied to video object segmentation and further video editing and retrieval applications. One of the advantages of our method is that we do not require the prior knowledge of the object of interest, and thus no training data or predetermined object detectors are needed; this makes our approach robust and practical to real-world problems. Very attractive empirical results on a variety of videos with highly articulated objects support the feasibility of our proposed method.

Index Terms— Video object extraction, conditional random field, sparse representation

1. INTRODUCTION

Video object extraction (VOE) aims to segment foreground objects of interest from video data. The success of VOE helps machines understand the content of videos, and thus VOE is typically considered as a preprocessing technique for high-level computer vision tasks such as human pose estimation, event recognition, and video annotation. For videos captured by a static camera, extraction of foreground objects can be treated as a background subtraction problem. In other words, foreground objects can be detected simply by subtracting the current frame from a background model, which is determined beforehand. Several sophisticated approaches have been proposed to model the background from a video sequence [1, 2]. However, if the background is consistently occluded by foreground objects, background modeling becomes a very challenging task.

In contrast with background subtraction methods for VOE, one can construct a foreground model if the target can be well described by training data. Once the concept of a video (i.e., the object of interest) is known, a category-specific detector can be trained to extract the object across video frames. To segment articulated objects such as human or animals, several methods were proposed to detect object parts rather than the entire object. For example, Nevatia et al. [3] and Davis et al. [4] both decomposed an object shape model in a hierarchical way to train object part detectors. These detectors are used to describe all possible configurations of the object of interest (e.g., pedestrians). Gorelick and Basri [5] collected a set of object silhouette exemplars. To extract the object of interest, the authors over-segmented the input image and determined the segments which best matched the associated templates. To deal with multiple human instances with large pose deformations, Niebles et al. [6] applied a human body detector on each frame, and their detection results were refined by pose density estimation function and probability diffusion between adjacent frames. Recently in [7], the authors further utilized template matching between the result produced by pedestrian detectors and a set of upright human pose templates, which is to simultaneously regularize and reduce the search space of possible object model configurations. However, these part-based methods typically assume that the object categories are known in advance, and they need to collect the object part templates to design each part detector.

Besides the above approaches, graph-based methods have been shown to be effective for foreground object segmentation. Using such methods, an image is typically represented by a graph, in which each observed node indicates an image pixel and the associated hidden node corresponds to its label. By determining the cost between adjacent hidden nodes using color, motion, etc. information, one can segment the foreground object by dividing the graph into disjoint parts while minimizing the total cost. Previous work such as [8] and [9] focused on an interactive scheme and required users to manually provide the ground truth label information. Although excellent results were produced, methods which do not require user interaction are more practical for real-world applications. Recently, several automatic segmentation techniques have been proposed. For example, Wu et al. [10] used a stereo camera setting which provides depth information as a cue for ground truth label. For videos captured by a monocular camera, literatures such as [11, 12] used a CRF framework which maximizes a joint probability of color, mo-
tion, etc. models to predict the label of each image pixel. Although the color features can be automatically determined from the input video, these methods required the trained object detectors to extract shape or motion features. A recently proposed method in [13] addressed the VOE problem without the use of any training data. It assumes that the motion of the background is dominant throughout the video, so the authors apply RANSAC [19] to extract candidate foreground regions, followed by a CRF which combines the associated color and motion features to determine the final foreground region.

In this paper, we focus on VOE in single concept videos captured by a monocular camera in static or arbitrary types of background. Instead of assuming that the background motion is consistently dominant and different from that of the foreground (as [13] did), we relax this assumption and allow foreground objects to be present in scenes which have marginal but complex background motion (e.g., motion induced by sea waves, swaying trees, etc.). We also ignore the video frames with significant motion variations due to shot changes or abrupt camera movements. To make our method robust and not require any user interaction, we start from multiple local motion cues, and integrate the induced shape and color models into a CRF. As we will discuss in Section 2, shape features better preserve local information of the foreground object than motion cues do, and our proposed framework allows the use of both foreground and background color models to provide better generalization in formulating the associated CRF model. It is worth noting that, our method does not require the prior knowledge of the object category, and thus no training data or object part detectors are needed. All the feature models we utilize in our CRF are automatically extracted from the test input video in an unsupervised setting, and this cannot be easily achieved by most prior work.

2. AUTOMATIC OBJECT MODELING AND EXTRACTION

Since not all the parts of a moving object will produce motion cues, or some of these cues might be negligible due to low contrast, etc. effects, it is not surprising that motion cues are not sufficient for VOE problems. To overcome this limitation, we propose to first extract the motion cues from the moving object across video frames, and we combine the motion-induced shape, foreground and background color models into a CRF. Without prior knowledge of the object of interest, this CRF model is designed to address VOE problems in an unsupervised setting. In Section 2.1, we first briefly review the use of CRF for object segmentation/extraction. We will detail the construction of our motion, shape, foreground and background color models, and discuss how we integrate them into a unified CRF framework in the remaining of this section.

2.1. Conditional random field

By utilizing an undirected graph, CRF [14] is a powerful technique to estimate the structural information (e.g. class label) of a set of variables. For object segmentation, CRF is used to predict the label of each observed pixel in an image \( I \). As shown in Figure 1, pixel \( i \) is associated with observation \( z_i \), while the hidden node \( F_i \) indicates its corresponding label (i.e. foreground or background). In this CRF framework, the label \( F_i \) is calculated by the observation \( z_i \), while the spatial coherence between this output and neighboring observations \( z_j \) and labels \( F_j \) are simultaneously taken into consideration. Therefore, predicting the label of an observation node is equivalent to maximizing the following posterior probability function:

\[
p(F|I, \psi) \propto \exp\left\{- \left( \sum_{i \in I} (\psi_i) + \sum_{i \in I, j \in \text{Neighbor}} (\psi_{i,j}) \right) \right\}, \tag{1}
\]

where \( \psi_i \) is the unary term which infers the likelihood of \( F_i \) by the associated observation \( z_i \), and \( \psi_{i,j} \) is the pairwise term modeling the relationship between neighboring pixels \( z_j \) and predicted outputs \( F_j \). Note that the observation \( z \) can be a particular feature model (e.g. color), or a combination of multiple types of features (as our approach does).

Alternatively, one can covert the above optimization problem into an energy minimization task, and the object energy function \( E \) of (1) can be derived as:

\[
E = -\log(p) = \sum_{i \in I} (\psi_i) + \sum_{j \in \text{Neighbor}} (\psi_{i,j}) \tag{2}
\]

From the above formulations, it is obvious that there are two issues to address when using such a CRF for VOE problems. The first task is to define the types of features (observations) to use in the CRF. These observed features should be robust and effective in producing satisfying VOE results on a variety of video data. Secondly, the determination of unary and pairwise potential functions \( \psi_i \) and \( \psi_{i,j} \) in (1) would affect the VOE performance. The following sub-sections will discuss our feature models, and detail how we integrate them into our CRF to produce promising VOE results.
2.2. Extraction of motion cues

In our work, each moving part of a foreground object is assumed to form a complete sampling of the entire object of interest (as [11, 12, 13] did). We aim to extract different feature information from these moving parts for the later CRF construction. To detect the moving parts and their corresponding pixels, we perform dense optical-flow forward and backward propagation [15] at every frame. A moving pixel \( q_t \) at frame \( t \) is determined by:

\[
q_t = \hat{q}_{t, t-1} \cap \hat{q}_{t, t+1},
\]

where \( \hat{q} \) denotes the pixel pair detected by forward or backward optical flow propagation. Only if a pixel is identified by the optical-flow trajectories in both directions, we will denote it as a pixel of a moving object. To alleviate the influence of camera shake, we ignore the frames which result in a large number of moving pixels after this step. After determining the regions induced by the moving object (or its parts), we will extract the associated shape and color information from these regions, as we discuss next.

2.3. Learning shape cues

Since we assume each moving part of an object forms a complete sampling of the entire object, part-based shape information induced by the above motion cues can be advanced to characterize the foreground object. To describe each moving part, we apply the histogram of oriented gradients (HOG) feature. To describe each moving part, we apply the histogram of oriented gradients (HOG) feature. To describe each moving part, we apply the histogram of oriented gradients (HOG) feature. To describe each moving part, we apply the histogram of oriented gradients (HOG) feature. To describe each moving part, we apply the histogram of oriented gradients (HOG) feature.

Since the use of sparse representation has been shown to be very effective in many computer vision tasks [16], once the HOG descriptors of the moving foreground regions are extracted, we learn an over-complete codebook and determine the associated sparse representation of each HOG. Now, for a total of \( N \) HOG descriptors \( \{ h_n, n = 1, 2, \ldots, N \} \) in a \( p \)-dimensional space, we construct an over-complete dictionary \( \mathbf{D}^{p \times K} \) which includes \( K \) basis vectors, and we determine the corresponding sparse coefficient \( \alpha_n \) of each HOG descriptor. Simply speaking, the sparse coding problem can be formulated as:

\[
\min_{\beta} \frac{1}{N} \sum_{n=1}^{N} \frac{1}{2} \| \mathbf{h}_n - \mathbf{D} \alpha_n \|_2^2 + \lambda \| \alpha_n \|_1,
\]

where \( \lambda \) balances the sparsity of \( \alpha_n \) and the \( l_2 \)-norm reconstruction error. We use the software developed by [17] to solve the above problem. Figure 2(a) shows example basis vectors (codewords) in terms of images. Each codeword is illustrated by averaging image patches with the top 15 \( \alpha_n \) coefficients (see Figure 2(b) for examples (only top 5 matches shown)). Since the background might be present in each extracted image patch, we further calculate the mask \( M \) for each codeword by averaging the moving regions with these top 15 \( \alpha_n \) coefficients. Recall that the moving region of each patch is induced by motion cues and has non-zero pixel values; the remaining parts of that patch are considered as static background and thus are zeroes. Figure 2(c) shows example masks for each codeword shown in Figure 2(a).

![Visualization of sparse shape representation](image)

**Fig. 2.** Visualization of sparse shape representation

After obtaining the dictionary and the masks to represent the shape of foreground object, we use them to encode all image patches at each frame. This is to recover non-moving regions of the foreground object which does not have significant motion and thus cannot be detected by motion cues. For each image patch, we derive its sparse coefficient vector \( \alpha \), and each entry of this vector indicates the contribution of each shape codeword. Correspondingly, we use the associated masks and their weight coefficients to calculate the final mask for each image patch. The reconstruction image using foreground shape information is then formulated as:

\[
\tilde{X}_t^S = \sum_{n \in h_t} \sum_{k} \frac{(\alpha_{n,k} \cdot M_k)}{\sum_n \alpha_{n,k}}.
\]

Figure 3 shows an example of the reconstruction of a video frame using shape information of the foreground object (induced by motion cues only). We note that \( \tilde{X}_t^S \) serves as the likelihood of foreground object at frame \( t \) in terms of shape information. This shape likelihood function contributes to the shape energy function in CRF, i.e.

\[
E^S = -w^s \log(\tilde{X}_t^S),
\]

where \( w^s \) controls the contribution of this shape energy term in the final CRF formulation. Comparing to the motion likelihood in Section 2.2 and [13], it is expected that better candidate foreground object can be discovered using the above
motion-induced shape information. This makes the use of foreground and background color models more feasible, as we discuss next.

![The original frame](image1) ![Shape likelihood](image2)

**Fig. 3.** An example shape likelihood image derived by our sparse shape representation.

### 2.4. Learning color cues

Besides the motion-induced shape information, we also combine the color cues into our CRF framework to better model the object of interest. Later in Section 3, our empirical results will confirm the integration of both shape and color models for improved VOE performance.

Constructing the background model is difficult because the sparse motion cues throughout the video might not be sufficient to indicate the foreground/background regions. This difficulty can be depicted in the Figure 4(b) which is an example of foreground region extraction using only motion cues in a CRF. To apply the color information of both foreground objects and the remaining background regions into our CRF, we utilize the shape likelihood image obtained from the previous step, and threshold the resulting shape posterior probability $X^S$. For the pixels of $X^S$ whose probability values are above a predetermined threshold, the associated regions will be potentially considered as foreground; those below the threshold will be thus grouped as candidate background regions. For these candidate foreground and background regions, we use Gaussian Mixture Models (GMM) $G^{CF}$ and $G^{CB}$ to model the RGB distribution for each, with the number of Gaussian components set to 10 for both cases. The parameters of GMM such as mean vectors and covariance matrices are determined by performing an expectation-maximization (EM) algorithm.

We now detail the learning of color cues for foreground object extraction. A single energy term which is associated with both foreground and background color models in our CRF is defined as follows:

$$ E^C = E^{CF} - E^{CB}, $$

where

$$ E^{CF} = -w^{CF} \log(\sum_{i \in I} G^{CF}(i)), $$

$$ E^{CB} = -w^{CB} \log(\sum_{i \in I} G^{CB}(i)). $$

Similar to (6), $w^{CF}$ and $w^{CB}$ in (7) weight the corresponding color models in the CRF. It is worth noting that only foreground color information is modeled in [13]. As we will show later in our experiments, disregard of the background color model would limit the performance of object segmentation.

![Object extraction example](image3)

**Fig. 4.** Object extraction example of (a) the input frame, using (b) motion, (c) foreground color, and (d) our proposed CRF integrating multiple types of motion-induced features.

### 2.5. Integration of multiple feature models via CRF

Induced by motion cues, we combine both shape and color (foreground and background) models into our CRF framework. Since we do not require prior knowledge of the object category, use of multiple types of motion-induced features allows us to model the foreground object of interest without the need of user interaction or any training data. To provide the property of spatial coherence into our CRF model, we introduce a pairwise term to preserve local foreground/background structures. As suggested by [18], this pairwise term $E_{i,j}$ is defined as:

$$ E_{i,j} = \sum_{j \in Neighbor} |F_i - F_j| \left( \lambda_1 + \lambda_2 \exp\left( -\frac{\|z_i - z_j\|}{\beta} \right) \right). $$

Note that $\beta$ is set as the averaged pixel color difference of all pairs of neighboring pixels, $\lambda_1$ is a data-independent Ising prior to smoothen the segmentation labels, and $\lambda_2$ is to relax the tendency of smoothness if observations $z_i$ and $z_j$ form an edge (i.e. when $\|z_i - z_j\|$ is large). We note that the above pairwise term is able to produce coherent labeling results even under low contrast or blurring effects. Finally, by integrating (6), (7) and (8), the objective energy function (2) of our CRF can be re-written as:

$$ E = E_{\text{unary}} + E_{\text{pairwise}} = (E^S + E^{CF} - E^{CB}) + E_{i,j} = E^S + E^C + E_{i,j}. $$

To solve the above optimization problem, one can apply graph-based energy minimization techniques such as max-flow/min-cut algorithms. When the above energy function is minimized, the labeling function output $F$ indicates the class label (foreground or background) of each observed pixel.

### 3. EXPERIMENTS

We first evaluate the performance of object extraction using different single types of features, and we compare their results with those produced by our method. We consider a sampled frame from the dancing sequence as shown in Figure 4(a). Figure 4(b) shows the object segmentation result.
using only motion features, i.e. the unary term in (9) only considers motion information, while the pairwise term aims to preserve the smoothness of output label set based on the observations of neighboring pixels. Since the dancer is simply moving his legs in the adjacent video frames, it is clear that only parts of his legs were segmented from the video frame. If only color information is utilized in the CRF for object extraction, only foreground regions with similar colors will be segmented as shown in Figure 4(c), and the right hand of the dancer was not successfully detected. Note that we do not consider the case using shape features only, since we use this as an intermediate step to produce the foreground object shape likelihood as shown in Figure 3(b), and we do expect some background clutter might be extracted if simply using this type of feature for VOE. Finally, we consider the case using our CRF which combines motion-induced shape, foreground/background color models, as shown in Figure 4(d). From Figure 4, we see that our CRF produced a very promising VOE result compared with those obtained by the use of single type of feature models.

Next, we consider a state-of-the-art approach proposed in [13] for more comparisons, which also combined multiple types of features in a CRF framework. Recall that the authors in [13] proposed to extract motion, color, and locality information from the observed video data to segment the object of interest. They used color information to model the foreground object only, since such information was induced by motion cues from moving foreground object parts. We note that the locality constraint in their CRF propagates the label of a pixel across consecutive frames. While this locality term provided additional temporal consistency in their VOE results, it might not work well for cases of significant camera motion or shot changes. Figure 5(a) shows a VOE example of a video sequence in four consecutive frames. We reproduce the CRF framework of [13] (with our motion cues), and the object segmentation results are shown in Figure 5(b).

As discussed in Section 2, our CRF utilizes motion-induced shape information, which results in a shape likelihood model and allows us to construct both foreground and background color models via GMMs. Therefore, better object segmentation results than those produced by use of only foreground color information can be expected. However, as shown in Figure 5(c) which shows segmentation results using only foreground and background color models for object extraction, we do not expect that the use of color information is sufficient for VOE since foreground and background regions might share the similar color information. Figure 5(d) illustrates the object segmentation results using our proposed CRF framework. With the integration of motion-induced shape, foreground and background color models, it can be observed that our approach exhibits excellent VOE ability than other approaches mentioned above. It is worth noting that, for a fair comparison, all parameters for each method in Figure 5 are tuned to produce their best segmentation results.

Finally, we conduct VOE experiments on several types of single-concept videos shown in Figure 6. Some videos in Figure 6 have shot changes (e.g. between the second and the third frames in the first video example), while one video has more than one foreground objects (of the same type). It is clear that all foreground objects are highly articulated, and most of the videos are with a non-static background. Using our CRF framework, very attractive object segmentation results were achieved in Figure 6. This again confirms the use of our approach for practical VOE problems, which require an unsupervised setting without the prior knowledge of the object category of interest, or applications with limited/no training data available.

4. CONCLUSION

In this paper, we proposed a method which utilizes multiple motion-induced features such as shape and foreground/background color models to extract foreground objects in single-concept videos. We advanced a unified CRF framework to integrate the above feature models. Using sparse representation techniques, our motion-induced shape model describes the shape information of the foreground object in a probabilistic way, which allows us to extract and construct both foreground and background color models for the object of interest. Compared with prior work, our approach better models the foreground object due to the use of multiple types of motion-induced feature models, while no prior knowledge of the object of interest, collection of training video data, or the design of object part detectors are required. Experiments on a variety of single-concept videos with highly
articulated objects verified the effectiveness of our proposed method.

Future research will be directed at extensions of our approach for videos with multiple concepts (i.e. multiple foreground objects of interest), and the applications of VOE for higher-level tasks such as action/activity recognition and video retrieval. For these applications, we expect to integrate features from heterogeneous domains (e.g. visual, audio, temporal, text, etc.), and we will provide a systematic way to select proper feature models for extracting particular types of the object of interest.

Fig. 6. Video object extraction results using our method. Note that the background is not static in the first three video examples, while the first one has a shot change between the second and the third frames; the third video is a single-concept video with multiple object instances.

Acknowledgements This work is supported in part by National Science Council of Taiwan via NSC 99-2221-E-001-020 and NSC 100-2631-H-001-013.

5. REFERENCES