Video Object Cosegmentation

Ding-Jie Chen, Hwann-Tzong Chen, and Long-Wen Chang
Department Of Computer Science
National Tsing Hua University, Taiwan

ABSTRACT

We introduce and address the problem of video object cosegmentation, which concerns the task of segmenting the common object in a pair of video sequences. We present a new algorithm that works on super-voxels in videos to solve this task. The algorithm computes *i)* the intra-video relative motion derived from dense optical flow and *ii)* the inter-video co-features based on Gaussian mixture models. The experimental results show that, by integrating the intra-video and inter-video information, our algorithm is able to obtain better results of segmenting video objects.

Categories and Subject Descriptors

I.4.6 [**Image Processing and Computer Vision**]: Segmentation—*Pixel classification*

General Terms

Algorithms

Keywords

Co-segmentation, Video, Motion, Graph cut, GMM

1. INTRODUCTION

An important issue of image segmentation is that the regions found by a typical image segmentation algorithm usually tend to be fragmented or lack semantical meanings. The concept of image cosegmentation, which is introduced by Rother et al. [16], provides a way to implicitly define the region of interest via multiple observations of common objects. The idea of making use of multiple observations can be applied to other tasks too. For example, an image can be associated with a large image database to solve the recognition problem [12]. Furthermore, in iCoseg [1], the system recommends the user where to draw scribbles for cosegmenting a foreground object from a group of related images.

The cosegmentation algorithm of Rother et al. [16] is based on the common appearance histogram, which is used as a global constraint in Markov random field (MRF) optimization. They define an objective function that incorporates the standard MRF terms with the global constraint. The MRF terms encode the intra-image spatial coherence and the global constraint encodes the inter-image common-region similarity. Rother et al. use the trust-region graph cut to minimize the objective function and segment the common regions from the input image pair. Recent work on image cosegmentation shows that the constraint of common appearance histogram can also be used as the regularization term [13] or the reward term [8]. In [18], Vicente et al. compare the aforementioned methods ([8, 13, 16]), and propose a *dual decomposition* technique to optimize the cosegmentation objective function. In addition, the rank-one global term is proposed in [14] to simultaneously segment multiple images that may contain common objects of different scales. Another approach to cosegmentation is to model it as a clustering problem, as presented in [4, 9]. Additionally, the co-saliency map [10] can be also used with graph cut to perform cosegmentation. The notion of object cosegmentation presented in [19] is to learn a similarity scoring function for choosing the common object from a pool of candidate object-like segmentations derived by the algorithm of [3].

The goal of this paper is to address the problem of video object cosegmentation, which is aimed at segmenting the common object in a pair of input video sequences. Our algorithm works on a super-voxel representation of videos. We introduce a motion-based video grouping method that can identify candidate common object regions according to relative motion. The common object appearance is then characterized by a Gaussian mixture model. The experimental results show that, by integrating the intra-video motion cues and the inter-video co-features, our algorithm is able to obtain better results of segmenting video objects than applying image cosegmentation or conventional video segmentation.

2. ALGORITHM

We propose a video cosegmentation algorithm that can extract the common object regions from a given pair of video sequences. The algorithm includes three main stages: The first stage is to partition each video sequence into a foreground set and a background set, according to the motion similarities. The second stage aims to construct a Gaussian mixture model (GMM) for the joint foreground set across the two input video sequences. For each video sequence, we also construct a GMM for the intra-video non-common regions. In the third stage, the common object in each video sequence is segmented according to the GMMs by solving graph cuts. Fig. 1 illustrates the proposed algorithm.

MM'12, October 29–November 2, 2012, Nara, Japan.

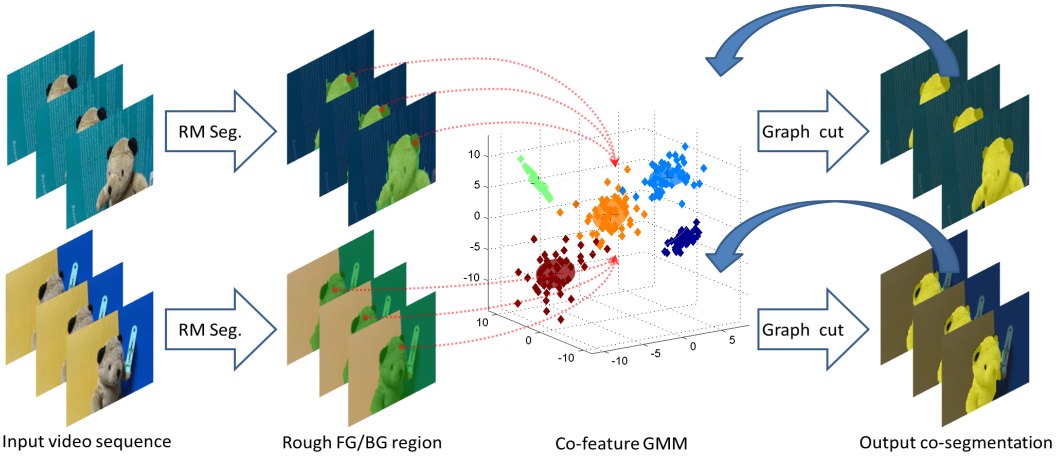

Figure 1: Flowchart of the proposed algorithm. Each input video sequence is roughly grouped into foreground regions and background regions according to the motion similarity. The matched candidate regions are used to initialize the co-feature Gaussian mixture model. Iteratively, we use graph cut to segment the common object regions, and relearn the parameters of GMMs from current segmentation results.

2.1 Space-Time Graph Construction

We adopt a similar space-time graph structure used in [7] as the video representation. Given a video sequence, we construct a 3D graph from the video volume using the graph-based over-segmentation technique [5]. To model the variations caused by object motion, we apply dense optical flow [11] to video frames. The nodes in the space-time graph are the pixels in each frame. Each node x_p^t in frame t connects to its 8-connected neighbors $\{x_q^t | x_q^t \in N(x_p^t)\}$ in the same frame, where $N(x_p^t)$ denotes the 8-connected neighborhood of x_p^t. Based on the dense optical flow, we may follow the backward flow vector of node x_p^t to find its corresponding node $x_{p'}^{t-1}$ in the previous frame, and then we connect node x_p^t to node $x_{p'}^{t-1}$ and all its neighbors $N(x_{p'}^{t-1})$. Similarly, we connect node x_p^t to node $x_{p'}^{t+1}$ and all its neighbors $N(x_{p'}^{t+1})$ in the next frame. In comparison with applying image over-segmentation to each frame independently, the proposed approach to space-time over-segmentation provides better temporal coherence.

As proposed in [5], over-segmentation can be computed by building a minimum spanning tree on a graph. Two regions are merged if the appearance difference between them is less than their individual internal variations. We use two different values for the threshold of internal variation to obtain coarse-level super-voxels (space-time over-segmentation) and fine-level super-voxels. Owing to the merging process of minimum spanning tree, the coarse-level super-voxels must contain at least one fine-level super-voxels.

2.2 Features for Super-voxels

We use three kinds of features to describe a super-voxel: color, texture, and relative motion. The Cb and Cr chroma channels are used to represent the color appearance for fine-level super-voxels. For each color channel, we compute the average color values over all pixels in a fine-level super-voxel. The Y luma channel is discarded to prevent problems caused by illumination variations across video sequences.

The texture feature is also derived from the chroma channels. We apply the *maximum response* (MR) filter bank [6] to each chroma channel. The MR filter bank consists of anisotropic Gaussian-like filters, which provide features such as edges and ridges. We select a subset of the MR filter bank (MRS4), which are invariant to scale and orientation changes. As is done for the color feature, the filter responses are averaged over a super-voxel to derive the texture feature.

Since the fine-level super-voxels have lower internal variations, they do not exhibit much textureness. Therefore we only compute the MRS4 features for coarse-level super-voxels. All fine-level super-voxel inherit the MRS4 features from their corresponding coarse-level super-voxels. Notice that the fine-level super-voxels are the actual building blocks of common object regions. The coarse-level super-voxels are simply used for the computation of texture features.

2.3 Relative Motion Segmentation

We present a new motion-based feature called the *relative motion* (RM) as the third type of feature for super-voxels. Super-voxels with similar relative motion are grouped by spectral clustering [15, 17]. We use relative motion to characterize the motion coherence of video objects. An example of grouped relative motion is shown in the second row of Fig. 2.

Given the dense flow F_x computed at each pixel x, a relative motion matrix $\mathcal{R}$ can be defined over T frames by

$$\mathcal{R}_{u,v} = \sum_{t=1}^{T-1} ||\bar{F}_u^t - \bar{F}_v^t||\,, \tag{1}$$

where

$$\bar{F}_u^t = mean(\{F_x^t\}),\ \text{for all pixels } x \in \text{ super-voxel } u\,;$$
$$\bar{F}_v^t = mean(\{F_x^t\}),\ \text{for all pixels } x \in \text{ super-voxel } v\,.$$

After calculating the relative motion matrix $\mathcal{R}$, we use the spectral clustering technique to group the super-voxels into clusters.

2.4 Co-feature Gaussian Mixture Models

We aim to identify the common object candidate from the super-voxel clusters for a given pair of input video sequences. We represent a super-voxel as a feature vector f consisting of the chroma features Cb, Cr, and the texture features

$MRS4_{Cb}$, $MRS4_{Cr}$. Each super-voxel cluster is modeled as a distribution of feature vectors. We use the χ^2 distance to compare the distributions across the two video sequences and identify the best match as the common object candidate.

To build the Gaussian mixture models (GMMs) in an unsupervised manner, we select one frame from each video sequence, and then apply a bounding box to the common object candidate. The super-voxels covered by the common object candidate are used to compute the co-feature GMM, while the super-voxels outside the bounding box are used to compute background-feature GMMs.

2.5 Markov Random Fields and Graph Cuts

Given a video frame, we define a Markov random field over the fine-level super-voxels in that frame. The goal of segmentation is to label a super-voxel as either the common object or the background. Such an MRF labeling problem can be solved by graph cuts [2]. As in the standard MRF formulation, our MRF energy function consists of two terms: the data term E_d and the smoothness term E_s, which are detailed as follows.

First, we use the Gaussian mixture models to represent the feature distributions of the common object and the background. Given the feature vector f_i for super-voxel i and a possible labeling α_i, the data term is computed based on the distance to the co-feature GMM $\mathcal{C}$ or the background GMM $\mathcal{B}$:

$$\begin{aligned} E_d(f_i, \alpha_i \in \mathcal{C}) &= -\log \pi_k^{\mathcal{C}} + \frac{1}{2}\log\det \Sigma_k^{\mathcal{C}} \\ &\quad + \frac{1}{2}[f_i - \mu_k^{\mathcal{C}}]^\top \Sigma_k^{-1}[f_i - \mu_k^{\mathcal{C}}] \end{aligned} \tag{2}$$

and

$$\begin{aligned} E_d(f_i, \alpha_i \in \mathcal{B}) &= -\log \pi_k^{\mathcal{B}} + \frac{1}{2}\log\det \Sigma_k^{\mathcal{B}} \\ &\quad + \frac{1}{2}[f_i - \mu_k^{\mathcal{B}}]^\top \Sigma_k^{-1}[f_i - \mu_k^{\mathcal{B}}] \end{aligned} \tag{3}$$

where k denotes the number of GMM components, and $\pi_k^{(\cdot)}$ denotes the weight for the kth GMM component. $\mu_k^{(\cdot)}$ denotes the sample mean of features, and $\Sigma_k^{(\cdot)}$ denotes the covariance matrix.

The smoothness term E_s is defined in terms of the feature distance weighted by the relative motion distance between neighboring super-voxels:

$$\begin{aligned} E_s(f_i, f_j, \alpha_i \neq \alpha_j) &= \exp(-\beta||f_i - f_j||^2) \\ &\quad \cdot \exp(-\gamma||\mathcal{R}_{i,j}||)\,, \end{aligned} \tag{4}$$

where β and γ are parameters. If the assigned labels of neighboring super-voxels are the same (i.e. $\alpha_i = \alpha_j$), the smoothness term is set to zero.

3. EXPERIMENTS

The main advantage of our algorithm is that it can model the intra-video and inter-video cues and use them to obtain a better segmentation of video objects. Such a mechanism is not available in conventional video segmentation methods and image cosegmentation approaches. To illustrate this point, we compare our algorithm with the video segmentation method presented in [7] and the image cosegmentation algorithm of [9].

Comparison with Video Segmentation.

We first compare our algorithm with the video segmentation method proposed by Grundmann et al. [7]. Their method uses a hierarchical graph structure to derive multilevel segmentation results, and the selection of a specific level (granularity) for generating the final segmentation has to be done by the user. Fig. 2 shows an example that highlights the difference in segmenting video objects using our algorithm and the algorithm of [7]. The segmentation results of [7] are obtained by selecting a level that can best preserve the object boundaries.

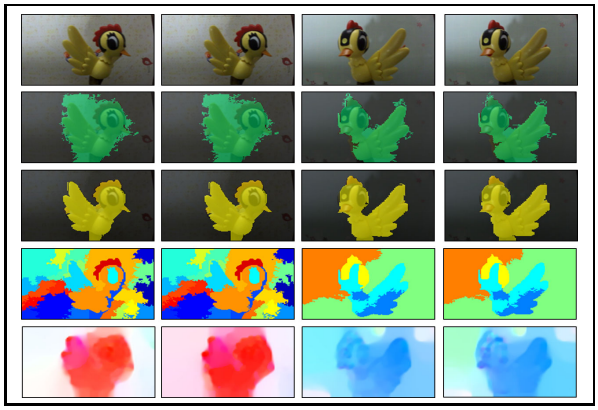

Figure 2: Comparison with video segmentation [7]. Row 1: The two images at the left are two successive frames of an input video sequence. The two images at the right are two successive frames of another video sequence. Row 2: The common-object candidates determined by relative motion. Row 3: The cosegmentation results obtained by our algorithm. Row 4: The results obtained by the video segmentation method of [7]. Row 5: The corresponding optical flows.

The example in Fig. 2 illustrates that there might exist ambiguities in optical flow, and thus to perform the relative motion segmentation on a single video sequence would be unsatisfactory. This problem can be addressed by our video cosegmentation algorithm, in which the information from another video sequence can help to build a better appearance model for the common object.

Comparison with Image Cosegmentation.

It is possible to apply image cosegmentation to videos by ignoring the temporal correlations and considering the video frames as a collection of images. We compare our algorithm with the image cosegmentation method proposed by Joulin et al. [9]. The best segmentation results generated by their algorithm are selected for comparison, as shown in Fig. 3. The experimental results indicate that the intra-video motion information is useful to resolve ambiguities in specifying the region of an object.

Finally, additional results of video cosegmentation generated by our algorithm are shown in Fig. 4.

4. CONCLUSION

We have presented a new algorithm to solve the problem of video object cosegmentation. We demonstrate that, by taking account of the intra-video motion cues and the inter-video appearance model together, we may devise a more powerful algorithm for segmenting video objects. We believe that the new problem of video object cosegmentation addressed in this paper is of sufficient interest to the multimedia community and is worth further investigating.

5. REFERENCES

[1] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, pages 3169–3176, 2010.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.

[3] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, pages 3241–3248, 2010.

[4] Y. Chai, V. S. Lempitsky, and A. Zisserman. Bicos: A bi-level co-segmentation method for image classification. In *ICCV*, pages 2579–2586, 2011.

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[6] J.-M. Geusebroek, A. W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. *IEEE Transactions on Image Processing*, 12(8):938–943, 2003.

[7] M. Grundmann, V. Kwatra, M. Han, and I. A. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148, 2010.

[8] D. S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *ICCV*, pages 269–276, 2009.

[9] A. Joulin, F. R. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, pages 1943–1950, 2010.

[10] H. Li and K. N. Ngan. A co-saliency model of image pairs. *IEEE Transactions on Image Processing*, 20(12):3365–3375, 2011.

[11] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Cambridge, MA, USA, 2009. AAI0822221.

[12] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV (3)*, pages 28–42, 2008.

[13] L. Mukherjee, V. Singh, and C. R. Dyer. Half-integrality based algorithms for cosegmentation of images. In *CVPR*, pages 2028–2035, 2009.

[14] L. Mukherjee, V. Singh, and J. Peng. Scale invariant cosegmentation for image groups. In *CVPR*, pages 1881–1888, 2011.

[15] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856. MIT Press, 2001.

[16] C. Rother, T. P. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *CVPR (1)*, pages 993–1000, 2006.

[17] J. Shi and J. Malik. Normalized cuts and image segmentation. In *CVPR*, pages 731–737, 1997.

[18] S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation revisited: Models and optimization. In *ECCV (2)*, pages 465–479, 2010.

[19] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, pages 2217–2224, 2011.

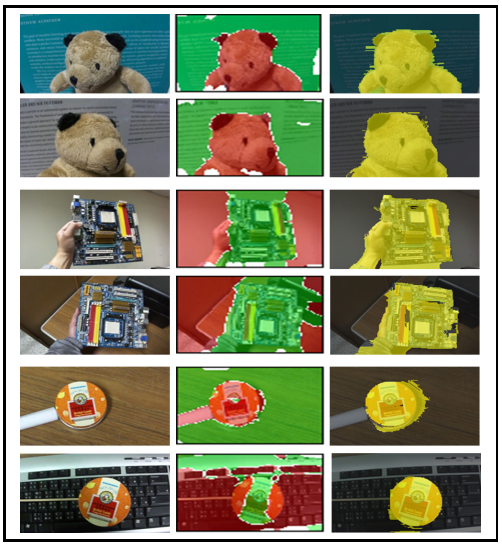

Figure 3: Comparison with image cosegmentation [9]. Column 1: The frames from three pairs of video sequences. Column 2: The results generated by the algorithm of Joulin et al. Column 3: The video cosegmentation results obtained by our algorithm.

Figure 4: More results of video cosegmentation generated by our algorithm.