MOMI-Cosegmentation: Simultaneous Segmentation of Multiple Objects among Multiple Images

Wen-Sheng Chu[1], Chia-Ping Chen[2,3], and Chu-Song Chen[1,2]

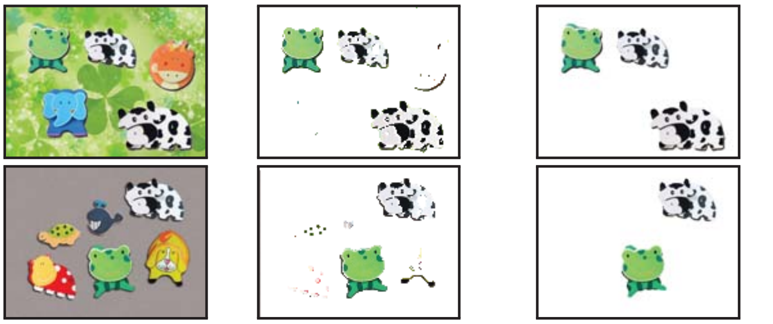
[1]Research Center for Information Technology Innovation, Academia Sinica,
Taipei 115, Taiwan
[2]Institute of Information Science, Academia Sinica, Taipei 115, Taiwan
[3]Dept. of CSIE, National Taiwan University, Taipei 106, Taiwan

Abstract. In this study, we introduce a new cosegmentation approach, *MOMI-cosegmentation*, to segment *multiple objects* that repeatedly appear among *multiple images*. The proposed approach tackles a more general problem than conventional cosegmentation methods. Each of the shared objects may even appear more than one time in one image. The key idea of MOMI-cosegmentation is to incorporate a common pattern discovery algorithm with the proposed Gibbs energy model in a Markov random field framework. Our approach builds upon an observation that the detected common patterns provide useful information for estimating foreground statistics, while background statistics can be estimated from the remaining pixels. The initialization and segmentation processes of MOMI-cosegmentation are completely automatic, while the segmentation errors can be substantially reduced at the same time. Experimental results demonstrate the effectiveness of the proposed approach over state-of-the-art cosegmentation method.

1 Introduction

Cosegmentation refers to simultaneous segmentation of similar objects from two or more images. While many studies [1–3] have shown that better segmentation from a single image could be achieved by interactive user inputs, completely automatic segmentation is possible for cosegmentation by using multiple images. The commonality across the images provides the information needed for facilitating the cosegmentation task. This idea was first introduced by Rother *et al.* [4] to segment an object of interest from an image pair, and has been applied to concurrent foreground extraction tasks, such as segmentation of image sequences [5] and several other problems [6–8]. Besides apparent applications in image or video editing, cosegmentation also implies several potential applications in other important areas, including biomedical imaging, video tracking, and content-based image retrieval.

The original goal of cosegmentation is to facilitate segmentation of common objects or regions by providing minimal additional information (such as just one additional image) so that better results could be obtained without user inputs. It is typically designed in a class-constrained fashion, i.e., a given set of images



(a) Input image pair (b) Cosegmentation [9] (c) The proposed method

Fig. 1: Given two (or more) images (a), the objective of cosegmentation is to segment the common objects in these images. Note that the problem here is more general since multiple objects could occur multiple times in an image. (b) and (c) show the results of state-of-the-art cosegmentaion algorithm [9] and the proposed method.

is assumed to be of the same object class. Because each image contains only one instance of the same object, one could consider the problem as approximating the position of the common object. In practice, many images, such as our daily photos, often share more than one object in common. An object may even appear more than one time in an image. Take Fig. 1 (a) for example. Two objects, *frog* and *cow*, simultaneously appear in the image pair and *cow* appears twice in the top image. As shown in Fig. 1 (b), the cosegmentation algorithm [9] produces segmentation errors when similar colors appear both in the foreground and background regions.

In this paper, we tackle a more general problem without the assumption that only one object appears in each image. The problem becomes more difficult than conventional cosegmentation and object detection (or recognition) in several aspects. First, no prior knowledge is provided for the common objects or regions: we have no idea about what and how many the common objects are, and how many times each object appears in an image. So how can we detect common objects in an unannotated image set? An intuitive way is to exhaustively compare all sub-images at all possible positions and scales among these images. The search domain, however, is extremely huge and the computational cost increases exponentially with the number of input images. Therefore, we present a new approach, *MOMI-cosegmentation*, to address the above issues in an unsupervised framework. The novelty of MOMI-cosegmentation lies in incorporating a common pattern discovery algorithm with the proposed MRF model, which is extended from a Gibbs energy [10]. We propose to use the common pattern discovery algorithm [11] to detect coherent objects among an unannotated image set. Besides, the initialization and segmentation of the proposed approach is completely automatic, which is vital for real world applications. Fig. 1 (c) shows the results of the proposed method, where segmentation errors are significantly reduced in comparison to Fig. 1 (b) obtained by [9].

2 Related Work

This paper lies in the intersection between the fields of cosegmentation and common pattern discovery. In this section, we briefly review previous works in each field.

Cosegmentation belongs to the category of unsupervised techniques. Existing approaches [4, 9, 12] cast this problem as a minimization problem of a Markov random field (MRF), which discourages histogram dissimilarities of foreground regions between two input images. The idea proposed in [4] penalized the MRF energy by the L_1 histogram dissimilarities of foreground regions. However, the optimization problem regularized by the L_1-norm becomes more difficult to solve. Mukherjee *et al.* [12] considered the problem using the squared L_2 distance and showed the modified objective function leads to an optimal linear programming solution of only "half-integrality" values. Hochbaum and Singh [9] claimed that the regularization terms of histogram difference lead to difficult optimization, and proposed to replace these terms by the "carrot or stick" strategy. The optimization problem was solved more efficiently in polynomial time using only one maximum flow maximization. However, these works implicitly assumed that only one object appears in each image.

Recent approaches for common pattern discovery are [11, 13–15]. Quack *et al.* [13] use a data mining technique to find spatial configurations of local features that frequently occur in an image set. A random partitioning approach is adopted in [14] to match all pairs of sub-images. A common pattern is then detected as the sub-image with the highest matching score. Yuan *et al.* [15] find a common pattern by gradually pruning possible candidates. Common patterns can be found by aggregating the voting maps in above methods. However, many methods implicitly assume that only one object appears in one image. In [11], common patterns are found as dense clusters in a correspondence graph represented by an incompatibility matrix. Because this work finds common patterns in a density-based clustering framework, it by nature relaxes the assumption that each image contains only one object. Nevertheless, these methods do not consider segmentation and produces undesirable segmentation artifacts.

The rest of this paper is organized as follows. In Section 3, we describe the concepts of [11] that is used to detect common objects that appears repeatedly in a set of images. The proposed segmentation model is presented in Section 4. We show the experimental results in Section 5 and give conclusions in Section 6.

3 Common Pattern Discovery

In this section, we review the concepts of the common pattern discovery algorithm [11]. Given a set of N unannotated images, the goal is to unsupervisedly detect common objects (or regions) shared by the image set. Note that the assumption that only one common object in each image is relaxed in this approach.

Candidate matches. Given the n-th image I_n, we extract a set of local appearance features $\mathcal{F}_n = \{(\mathbf{p}_n^i, s_n^i, \mathbf{d}_n^i) | i = 1, \ldots, |\mathcal{F}_n|\}$, where $\mathbf{p}_n^i$ and s_n^i are

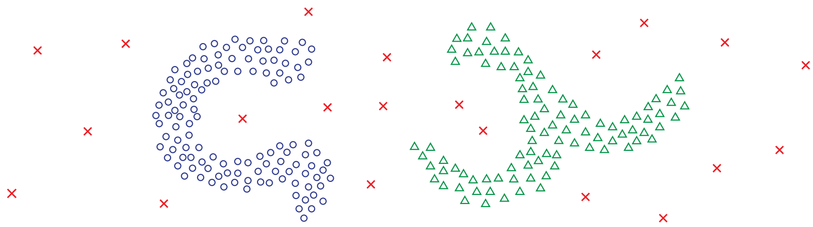


Fig. 2: Illustration of the correspondence graph used for density-based clustering [11]. Each node represents a candidate match ii'; each dense cluster can be considered as a common object (or region).

the position and the scale of the i-th feature in I_n, $\mathbf{d}_n^i$ is the corresponding feature descriptor and $|\mathcal{F}_n|$ is the number of features in I_n. Here, we use the Harris-Laplace corner detector and the OpponentSIFT descriptor [16] for feature extraction. Note that other options [17,18] are also applicable for computing the feature descriptor $\mathbf{d}_n^i$.

Given two images I_m and I_n as two sets of local features, the number of all possible correspondences across each pair of local features is enormous. It is computationally prohibitive to establish such a correspondence between each image pair. Therefore, we filter out the candidate matches $\mathcal{M}$ by

$$\mathcal{M} = \{ii' \mid \|\mathbf{d}_m^i - \mathbf{d}_n^{i'}\| < \lambda\}, \tag{1}$$

where ii' stands for the match between the i-th feature in I_m and the i'-th feature in I_n. λ controls the maximum dissimilarity between two appearance features. Typically, $\mathcal{M}$ will contain only a small subset of all possible matches.

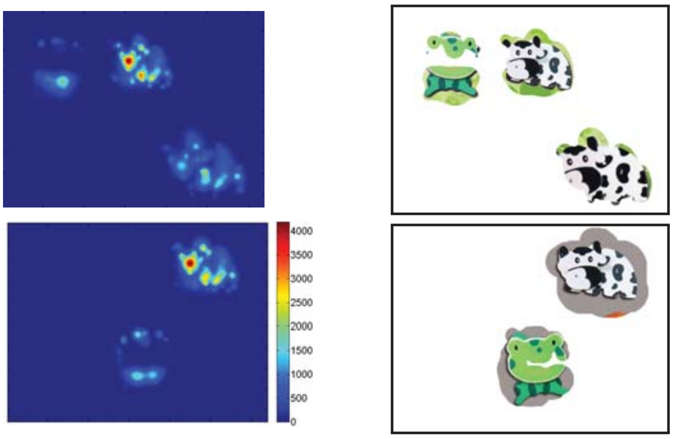
Incompatibility matrix. After the relatively small candidate matches $\mathcal{M}$ is filtered out for each image pair, the next goal is to construct an incompatibility matrix $\mathbf{D}$ for these matches between two images I_m and I_n. The incompatibility matrix $\mathbf{D}$ measures the incoherence between a pair of "matches" in the two images. Let i_1 and i_2 denote two local features in I_n, $sd^n(i_1, i_2) = \|\mathbf{p}_n^{i_1} - \mathbf{p}_n^{i_2}\|$ indicates the spatial distance between i_1 and i_2. Considering each candidate match $i_2 i'_2$ within the spatial ε-neighborhood of $i_1 i'_1$, i.e., $sd^m(i_1, i_2) < \varepsilon$ and $sd^n(i'_1, i'_2) < \varepsilon$, we can compute their incompatibility as:

$$\mathbf{D}(i_1 i'_1, i_2 i'_2) = \alpha_1 \times unary(i_1 i'_1, i_2 i'_2) + \alpha_2 \times binary(i_1 i'_1, i_2 i'_2), \tag{2}$$

where *unary* and *binary* are the constraints used to capture the appearance dissimilarity and geometric inconsistency for each pair of candidate matches, respectively. A possible choice of the *unary* and *binary* constraints can be given as in [11]:

$$unary(i_1 i'_1, i_2 i'_2) = \frac{\|\mathbf{d}_m^{i_1} - \mathbf{d}_n^{i'_1}\| + \|\mathbf{d}_m^{i_2} - \mathbf{d}_n^{i'_2}\|}{2}, \tag{3}$$

$$binary(i_1 i'_1, i_2 i'_2) = \frac{|sd^m(i_1, i_2) - sd^n(i'_1, i'_2)|}{\sqrt{sd^m(i_1, i_2) sd^n(i'_1, i'_2)}}. \tag{4}$$



(a) Confidence maps I_n^C (b) Preliminary segmentation results I_n^P

Fig. 3: The confidence maps of Fig. 1 (a). The larger value given in the confidence map implies higher possibility that the pixel is a part of a shared object.

Correspondence graph. Small values in $\mathbf{D}$ reflects potential correct matches of a shared object in the image pair, because appearance differences and geometric inconsistency between correct matches shall be small. Incorrect matches are likely to be inconsistent with each other with large incompatibilities. From this point of view, we can see the candidate matches $\mathcal{M}$ as nodes that forms the *correspondence graph* with corresponding linkage weights specified by $\mathbf{D}$. As illustrated in Fig. 2, correct matches tend to form dense clusters (blue circles and green triangles) with small linkage weights. The isolated nodes (red crosses) in the correspondence graph indicate the incorrect matches with large linkage weights.

Density-based clustering. Given the correspondence graph, the problem of finding common objects in an image set is reduced to a dense cluster discovery problem. A dense cluster, i.e., a set of nodes linked by small weights, represents a possible shared object appearing in an image pair. As we do not know in advance the shape of each cluster, clustering methods that assume each cluster has a globular shape, such as K-means and affinity propagation, are not adequate for this case. Furthermore, the number of dense clusters in the correspondence graph is not known either.

Therefore, the density-based algorithm [19] is utilized to discover clusters with arbitrary shapes in the presence of a large number of outlier matches. One of the benefits of the algorithm is that we do not have to specify the number of clusters in advance. The only parameters used are the radius ϵ of neighborhood and the density d in the ϵ-neighborhood. In our implementation, we fixed $\epsilon = 2000$ and $d = 20$ across all experiments.

The confidence map. After performing the density-based algorithm for each pair of images, we derive $(N-1)$ feature masks for each image in the unannotated image set. Each feature mask records the confidence of each local feature and indicates that how likely a local feature is a part of a common object. The confidence of the i-th local feature $\mathcal{F}_n^i$ in the image I_n is accumulated across all $(N-1)$ feature masks. By fusing these feature masks for each image, we then obtain a confidence map of positive real values. The confidence map of Fig. 1 (a) is shown in Fig. 3. Preliminary segmentation results can be obtained by performing a simple thresholding. The preliminary segmentation results obtained from the image pair of 1 (a) are shown in Fig. 3 (b). See Fig. 4 (c) for more examples. Although this algorithm can successfully detect common objects across input images, the objects are only partly included in the preliminary segmentation results.

4 MOMI-Cosegmentation Incorporating Common Pattern Discovery

Conventional cosegmentaion methods [4,9,12] are restrictive in two assumptions: the input is an image pair and each image contains the same object in different backgrounds. In order to detect *multiple objects* that may appear *multiple times* in one image, we incorporate the preliminary segmentation results I_n^P and the confidence maps I_n^C images of N images generated from the common pattern discovery algorithm [11]. We then consider the cosegmentation problem as an individual foreground/background segmentation on each image $I_n, n = 1, \ldots, N$.

The segmentation problem can be interpreted as a binary labelling problem: each pixel p has to be assigned a unique label x_p, where x_p is a binary label of 0 (background) or 1 (foreground). Let $\mathcal{V}$ be the set of all pixels in I_n and $\mathcal{E}$ be the set of all adjacent pixel pairs in I_n. We formulate the problem of computing the optimal labels $X = \{x_p | p \in \mathcal{V}\}$ as an energy minimization of the following cost function:

$$
\begin{aligned}
E(X) = \lambda_{\text{color}} \sum_{p \in \mathcal{V}} E_{\text{color}}(x_p) + \lambda_{\text{smoothness}} \sum_{(p,q) \in \mathcal{E}} E_{\text{smoothness}}(x_p, x_q) + \\
\lambda_{\text{confidence}} \sum_{p \in \mathcal{V}} E_{\text{confidence}}(x_p) + \lambda_{\text{locality}} \sum_{p \in \mathcal{V}} E_{\text{locality}}(x_p). \qquad (5)
\end{aligned}
$$

We introduce the different energy terms corresponding to various cues from the prior knowledge of color models, smoothness, confidence maps and locality relationship. The parameters λ_{color}, $\lambda_{\text{smoothness}}$, $\lambda_{\text{confidence}}$ and $\lambda_{\text{locality}}$ balance the contribution of each energy term. Each energy term is then described in the following subsections.

4.1 Color term & smoothness term

The color and smoothness terms are frequently used in segmentation problems [1, 20, 21]. We first explain the two terms as the *fundamental model.*

Color term. The idea of color term is to exploit the fact that different groups of foreground or background segments tend to follow different color distributions. For an image I_n, we train two Gaussian mixture models (GMMs), one for the foreground and one for the background, from the given preliminary segmentation result I_n^P. The purpose of each GMM is to estimate the likelihood of each pixel p that belongs to foreground or background based on the color cue. Each GMM is taken as full-covariance Gaussian with K components (typically $K = 5$). The color term is defined as

$$E_{\text{color}}(x_p) = -\log G(p|x_p), \tag{6}$$

where each color model G is given by the mixture of Gaussians:

$$G(p|x_p) = \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{\det \Sigma_k}} \exp\left(-\frac{1}{2}(p-\mu_k)^T \Sigma_k^{-1}(p-\mu_k)\right). \tag{7}$$

$G(p|x_p)$ indicates the probability that pixel p belongs to the label x_p. Note that if the pixel p is assigned to be foreground ($x_p = 1$), the summation in Eq. (7) is over the foreground GMMs for estimating the foreground likelihood of p; otherwise, the summation is over the background GMMs. The color term encourages the pixels to follow the labels of the most similar color model.

Smoothness term. The smoothness term is designed to preserve the coherence between two neighboring pixels of similar pixel values and imply a tendency to solidity of objects. This is useful in situations where matching constraints are weak, such as too sparse candidate matches or too many ambiguous colors that both occur in the foreground and the background GMMs. The smoothness term between two adjacent pixels p and q is defined as

$$E_{\text{smoothness}}(x_p, x_q) = [x_p \neq x_q] \exp\left(-\beta \|p-q\|^2\right), \tag{8}$$

where $[expr]$ denotes the indicator function taking value $0, 1$ for the predicate $expr$ and the constant β can be chosen to be $\left(1/2\langle\|p-q\|^2\rangle\right)$ as suggested in [10]. This term is a smoothness penalty when the neighboring pixels are labelled differently, i.e., $x_p \neq x_p$. In other words, the less similar colors of p and q are, the smaller cost $E_{\text{smoothness}}$ would produce, and therefore the more likely the edge between p and q is on the object boundary.

The minimization problem using E_{color} and $E_{\text{smoothness}}$ alone is similar to that proposed in GrabCut [1]. The main distinction is that we extend the segmentation domain from the initial user-defined rectangle trimap to the entire image. The results of GrabCut used in this manner are shown in third column of Fig. 7. Although color coherence and smoothness are preserved by E_{color} and $E_{\text{smoothness}}$, noticeable segmentation errors occur because of the imperfect preliminary segmentation in I_n^P and the non-discriminative GMMs of the foreground and background. We then introduce two more energy terms, $E_{\text{confidence}}$ and E_{locality}, to recover correct foreground pixels as well as remove false "background artifacts".

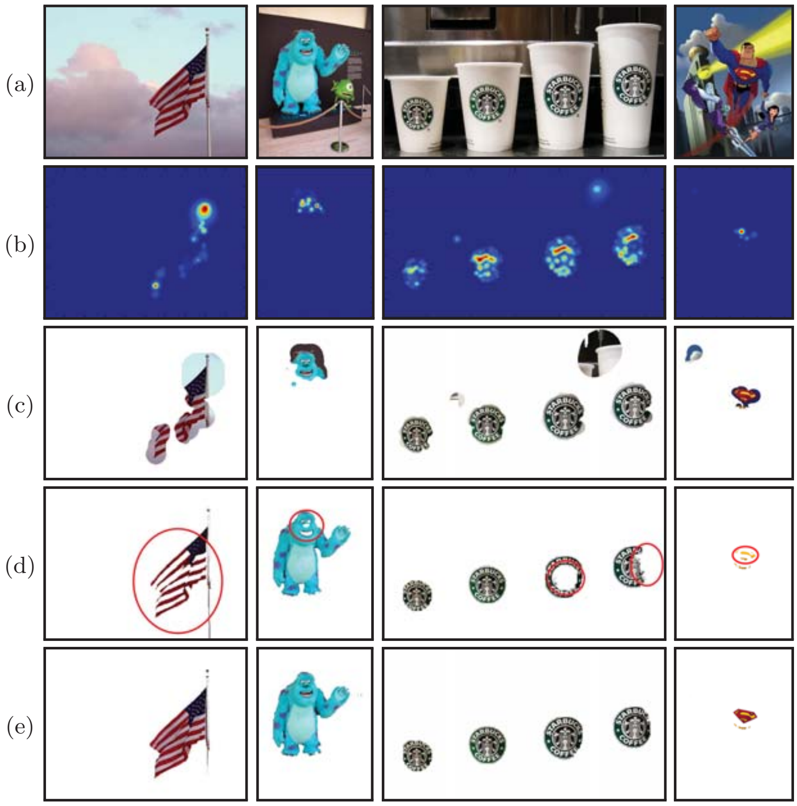
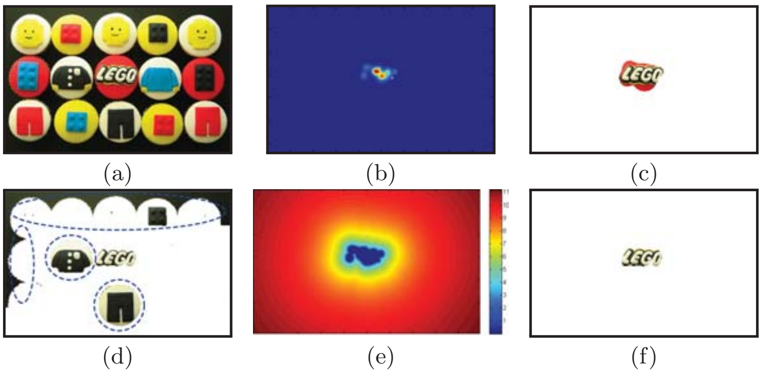


Fig. 4: Effects of confidence terms. (a) input images, (b) given confidence maps, (c) preliminary segmentation results, (d) GrabCut with only color terms and smoothness terms based on the preliminary results (red circles indicate the segmentation errors) and (e) segmentation with additional confidence terms.

4.2 Confidence term

The energy functions discussed in the above section may cause the segmentation errors where correct foreground pixels are assigned to background labels. This is because the similar colors between the foreground and the background models distract the labelling of foreground pixels. Fig. 4 shows examples when this type of segmentation errors occur. Take the *American flag* in Fig. 4 (d) for example, the white stripes are wrongly labelled because of its uncertain likelihood of white color between foreground and background. The goal in each column (from left to right) of this figure is to segment the *American flag*, the animation character *Sulley*, the trademark of *Starbucks* and *Superman*'s S shield, respectively.

Therefore, we resort to the cues of confidence map I_n^C, produced by the common pattern discovery algorithm discussed in Section 3, to resolve the color ambiguity. Specifically, we exploit the prior knowledge of confidence values to



(a) (b) (c)

(d) (e) (f)

Fig. 5: Effects of locality terms. (a) input images, (b) given confidence maps, (c) preliminary segmentation results, (d) GrabCut based on the preliminary results (blue dashed circles indicate the background artifacts), (e) logarithmic distance map and (f) segmentation with additional confidence terms.

encourage good and coherent segmentations; pixels with high confidence values should be retained as foreground. Given $c(p)$ as the original confidence value of p in I_n^C, we define the confidence term as

$$E_{\text{confidence}}(x_p) = \begin{cases} (2x_p - 1)\tilde{c}(p), & \tilde{c}(p) > 0 \\ (1 - 2x_p)\tilde{c}(p), & \textit{otherwise}. \end{cases} \tag{9}$$

where $\tilde{c}(p)$ is the normalized confidence energy of pixel p in $[-1, 1]$ by the sigmoid function

$$\tilde{c}(p) = 4\left(\frac{1}{1 + \exp\left(-c(p)\right)} - \frac{3}{4}\right). \tag{10}$$

Larger $\tilde{c}(p)$ refers to larger value of $c(p)$, which indicates more confidence that the pixel p belongs to common objects appearing repeatedly among an image set. When $\tilde{c}(p) > 0$, p has high possibility of belonging to the foreground, and thus the confidence term encourages the foreground ($x_p = 1$) likelihood by adding $\tilde{c}(p)$ and penalizes the background ($x_p = 0$) by subtracting $\tilde{c}(p)$. On the other hand, when $\tilde{c}(p) \leq 0$, we subtract $\tilde{c}(p)$ from $x_p = 1$ and add $\tilde{c}(p)$ to $x_p = 0$. As shown in Fig. 4 (d) and (e), most neglected foreground pixels could be recovered by incorporating the confidence term.

4.3 Locality term

The color, smoothness and confidence cues could usually produce good results in most image sets. However, when there are color ambiguities between background and foreground GMMs, or when the number of background GMMs are not large enough to model the colors in cluttered backgrounds, incorrect segmentations in the background often occur. We call the undesirable background segments the "background artifacts".

An example is illustrated in Fig. 5. The first row displays the input image, given correspondences map and the preliminary segmentation. Segmentation based on color and smoothness cues is shown in Fig. 5 (d), where the background artifacts are marked as red circles. In order to remedy these background artifacts, we introduce the locality term

$$E_{\text{locality}}(x_p) = \log\left(\min_{q\in\mathcal{V},c(q)>\delta} dist(p,q)\right), \tag{11}$$

where $dist(p,q) = \|\mathbf{p}_p - \mathbf{p}_q\|^2$ is the spatial distance between any pixel pairs (p,q), δ controls the threshold for candidates of the reference pixel q and σ is a parameter (typically $\sigma = 20$). We use the locality term to impose the distance penalty on pixels that are away from those with confidence values higher than δ. The further a pixel p is away from the reference pixel q, the less possible p belongs to the foreground. The locality term, from this perspective, is helpful to remove the background artifacts that have similar colors as foreground pixels. Fig. 5 (e) and (f) display the logarithmic distance maps and segmentation results incorporating the locality term, respectively.

5 Experimental Results

In this section, we discuss the experiments for evaluating the performance of the proposed method. Qualitative and quantitative analysis of the proposed approach are presented. We used the min-cut algorithm [10] to minimize the energy function $E(X)$. Throughout the following experiments, ϵ and d in [11] were set to be 2000 and 20, and K for the color models was fixed at 5. Parameters $\lambda_{\text{color}} = 1$ and $\lambda_{\text{smoothness}} = 40$ were set for the proposed Gibbs model, while the choices of $\lambda_{\text{confidence}}$ and $\lambda_{\text{locality}}$ were user-specified.

Comparison with cosegmentation. We firstly compare the proposed method with state-of-the-art cosegmentation [9]. Because the cosegmentation algorithm [9] considers only two input images, the proposed method was evaluated using only two images for fairness. In addition, [9] takes a large memory storage of additional nodes, hence the segmentation errors for [9] were reported on lower-resolution images while those were reported on full-resolution images for the proposed method. Although [9] was introduced for automatically extracting common foreground from two images, it requires manually labelling of RGB intensities for foreground and background. Our method, on the other hand, performs an automatic preliminary labelling from the results of the common pattern discovery.

The segmentation errors, i.e., the percentage of wrongly labelled pixels with respect to the whole image, were presented for five image sets as shown in Fig. 7. We also performed GrabCut [1] on each image pair as a baseline algorithm. As shown in the 3^{rd} and 4^{th} columns, GrabCut and [9] could extract the objects of interest, but suffer from the problem of color ambiguity: similar colors between foreground and background pixels. Note that in the last example, *Leaning Tower of Pisa*, the tower exhibits different colors because of different illumination. [9] fails to extract all correct foreground pixels, while our method, shown in the

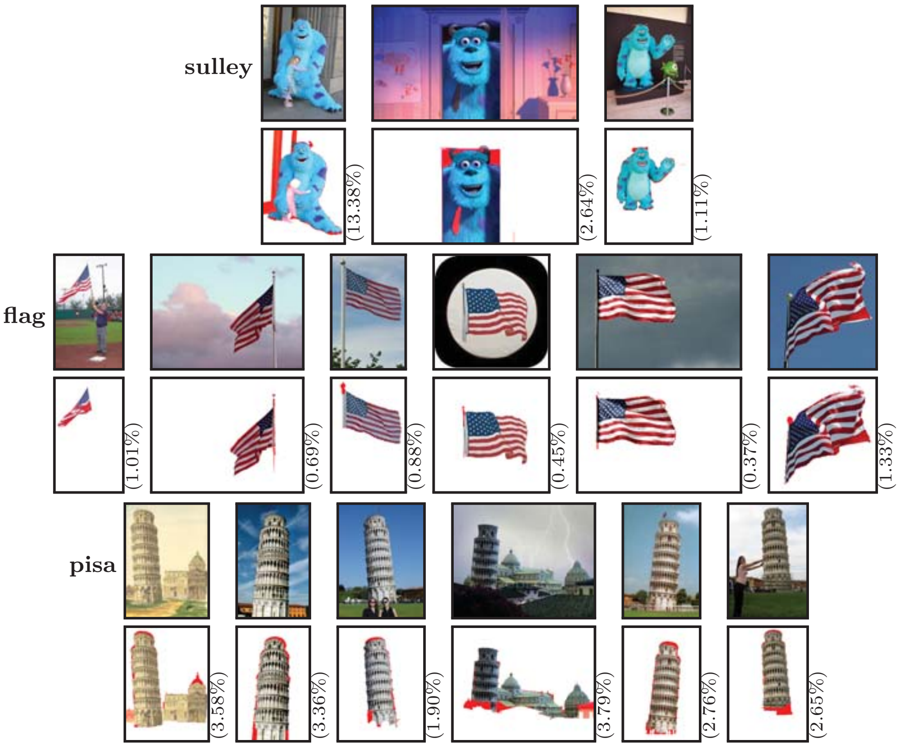


Fig. 6: The first row shows an input image set and the second row shows corresponding segmentation obtained by our method. Segmentation errors are shown as percentages and marked as red colors in the segmentation results.

5^{th} column, utilized the confidence term to retain correct foreground pixels and the locality term to remove mislabelled pixels in the background. Foreground misses and background artifacts can be thus considerably reduced. Our method in these image sets produces nearly perfect results as groundtruths with very low segmentation errors. Results on more image sets will be presented shortly.

Comparison with the *fundamental model*. For image sets containing more than two images, we compare the performance between the proposed approach and the fundamental model used in [1]. The datasets[1] were collected from Flickr with moderate variations in illumination and scale. Groundtruths were manually labelled. Averaged segmentation errors of each dataset were presented in Table 1. The results show that good segmentation for concurrent objects can still be obtained using our method, although each dataset contains more than two images. The fundamental model used in [1] considers only color and smooth-

[1] The dataset is available at `http://imp.iis.sinica.edu.tw/ivclab/research/coseg/`.

Table 1: Comparison between the *fundamental model* (FM) used in [1] and the proposed *MOMI-cosegmentation* (MOMI-CS). Each method is evaluated by averaging the segmentation errors across the 12 datasets.

set(#img)	sulley(3)	starbucks(3)	magnet(4)	flag(6)	pisa(6)	superman(7)
FM	20.50	2.68	22.56	7.71	17.88	18.62
MOMI-CS	5.71	0.41	1.20	0.79	3.01	1.38
set(#img)	domino(6)	heineken(8)	warcraft(6)	kfc(6)	lego(4)	pringles(8)
FM	26.65	18.47	26.65	35.21	43.52	15.24
MOMI-CS	2.46	1.25	2.63	6.78	1.08	4.17

ness cues, therefore produces worse results when similar colors appear in both foreground and background, as shown in the third column of Fig. 7. The proposed method achieved an average of 2.57% segmentation errors across the 12 image sets.

Besides rigid objects, we also evaluated the proposed method on some deformable objects. Both qualitative and quantitative results are shown in Fig. 6. Note that some objects of the same class may appear in heterogeneous circumstances, e.g., the second image in *sulley* is from animation while the others are real models. Similar circumstances could be found in the 4^{th} image (from left to right) in *flag* and the 1^{st} image in *pisa*. Moreover, some images are very challenging because of their cluttered backgrounds. The proposed method is capable of successfully segment the shared objects, and produced satisfactory results with less than 6% averaged segmentation errors in these mega-pixel images.

6 Conclusion

In this paper, we proposed a new cosegmentation approach called MOMI-cosegmentation, which is more general and scalable in many aspects. Compared to conventional cosegmentation methods, the proposed approach can deal with more than two input images, and allow multiple objects to appear more than one time in an image. Although the domain of searching the common objects in multiple images is computationally prohibitive, we combined color, smoothness, confidence and locality cues and incorporated a common pattern discovery algorithm to achieve satisfactory segmentation. Foreground misses and background artifacts can be efficiently reduced using our method. In addition, label initialization and segmentation process are automatic in MOMI-cosegmentation. The experiments have demonstrated that the performance of the proposed method outperforms state-of-the-art cosegmentation method [9].

Acknowledgement. This work was supported in part by the National Science Council, Taiwan, under the grants NSC99-2631-H-001-020 and NSC98-2221-E-001-012-MY3. We would also like to thank Professor Vikas Singh for providing the source code of [9].

References

1. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In: ACM SIGGRAPH, ACM (2004) 314
2. Li, Y., Sun, J., Tang, C.K., Shum, H.Y.: Lazy snapping. ACM Trans. on Graphics **23** (2004) 303–308
3. Riklin-Raviv, T., Sochen, N., Kiryati, N.: Shape-based mutual segmentation. IJCV **79** (2008) 231–245
4. Rother, C., Minka, T., Blake, A., Kolmogorov, V.: Cosegmentation of image pairs by histogram matching-incorporating a global constraint into MRFs. In: CVPR. Volume 1. (2006)
5. Cheng, D.S., Figueiredo, M.: Cosegmentation for image sequences. In: International Conference on Image Analysis and Processing. (2007) 635–640
6. Sun, J., Kang, S.B., Xu, Z.B., Tang, X., Shum, H.Y.: Flash Cut: Foreground Extraction With Flash And No-flash Image Pairs. In: CVPR. (2007) 1–8
7. Cao, L., Fei-Fei, L.: Spatially coherent latent topic model for concurrent object segmentation and classification. In: ICCV. (2007)
8. Gallagher, A.C., Chen, T.H.: Clothing cosegmentation for recognizing people. In: CVPR. (2008) 1–8
9. Hochbaum, D.S., Singh, V.: An efficient algorithm for co-segmentation. In: ICCV. (2009)
10. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in ND images. In: ICCV. Volume 1. (2001) 105–112
11. Chen, C.P., Chu, W.S., Chen, C.S.: Common pattern discovery with high-order constraints by density-based cluster discovery. Submitted for publication (2010)
12. Mukherjee, L., Singh, V., Dyer, C.R.: Half-integrality based algorithms for cosegmentation of images. In: CVPR. (2009)
13. Quack, T., Ferrari, V., Leibe, B., Gool, V.L.: Efficient mining of frequent and distinctive feature configurations. In: ICCV. (2007) 1–8
14. Yuan, J., Wu, Y.: Spatial random partition for common visual pattern discovery. In: ICCV. (2007) 1–8
15. Yuan, J., Li, Z., Fu, Y., Wu, Y., Huang, T.S.: Common spatial pattern discovery by efficient candidate pruning. In: International Conference on Image Processing. (2007)
16. van de Sande, K., Gevers, T., Snoek, C.: Evaluating color descriptors for object and scene recognition. IEEE Trans. on PAMI **(in press)** (2010)
17. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
18. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. IJCV **60** (2004) 63–86
19. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Knowledge Discovery and Datamining. (1996) 226–231
20. Sun, J., Zhang, W., Tang, X., Shum, H.Y.: Background Cut. ECCV **3952** (2006) 628–641
21. Guillemaut, J.Y., Kilner, J., Hilton, A.: Robust graphcut scene segmentation and reconstruction for free-viewpoint video of complex dynamic scenes. In: ICCV. (2009)

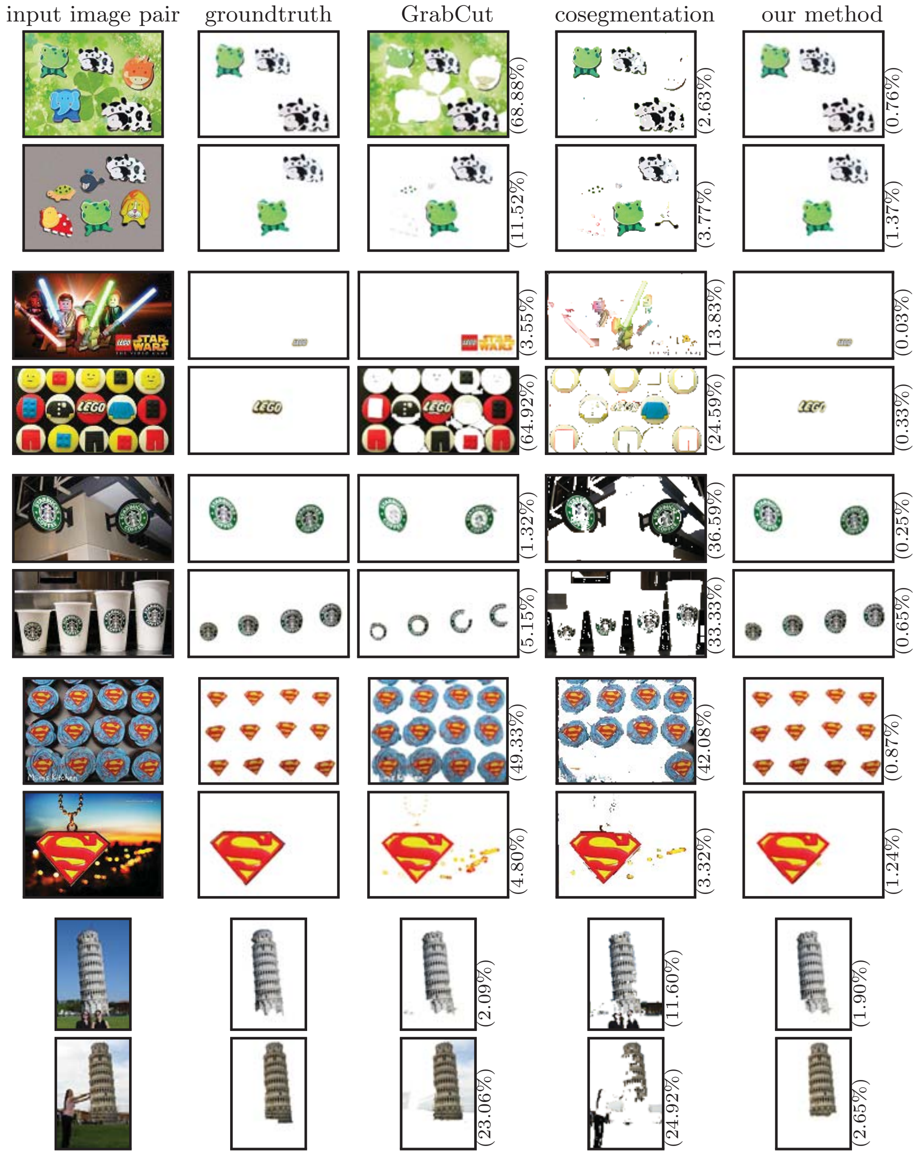


Fig. 7: Five examples of image pairs. Each column (from left to right) shows the input image pairs, groundtruth, GrabCut [1] results, cosegmentation [9] results and results of our method, respectively. (Errors are denoted as the percentages.)