Contour-Based Joint Clustering of Multiple Segmentations

Daniel Glasner[1,*], Shiv N. Vitaladevuni[2,*], and Ronen Basri[1,†]

[1]Dept. of Computer Science and Applied Mathematics, The Weizmann Institute of Science
[2]Raytheon BBN Technologies. {daniel.glasner,ronen.basri}@weizmann.ac.il, svitalad@bbn.com

Abstract

We present an unsupervised, shape-based method for joint clustering of multiple image segmentations. Given two or more closely-related images, such as nearby frames in a video sequence or images of the same scene taken under different lighting conditions, our method generates a joint segmentation of the images. We introduce a novel contour-based representation that allows us to cast the shape-based joint clustering problem as a quadratic semi-assignment problem. Our score function is additive. We use complex-valued affinities to assess the quality of matching the edge elements at the exterior bounding contour of clusters, while ignoring the contributions of elements that fall in the interior of the clusters. We further combine this contour-based score with region information and use a linear programming relaxation to solve for the joint clusters. We evaluate our approach on the occlusion boundary data-set of Stein et al.

1. Introduction

We present a method that combines contour- and region-based information to produce a joint clustering of two or more closely-related images. By "closely-related", we mean that the same objects are present in the images and that the objects roughly maintain their shapes, e.g., nearby frames in a video sequence, images of the same scene taken under varying illumination, or adjacent tissue slices in 3-dimensional biomedical images.

Since accurate segmentations are often difficult to obtain, a large number of recent methods utilize "super-pixels," i.e., regions obtained by oversegmentation of the input image. In this work we consider the problem of co-clustering oversegmentations to improve segmentation quality. Consider the problem of combining information from automatic superpixel maps generated independently for consecutive frames of a video sequence. We assume that true object boundaries persist in most of the frames, while the false boundaries will be random and unlikely to be consistent in all the frames.

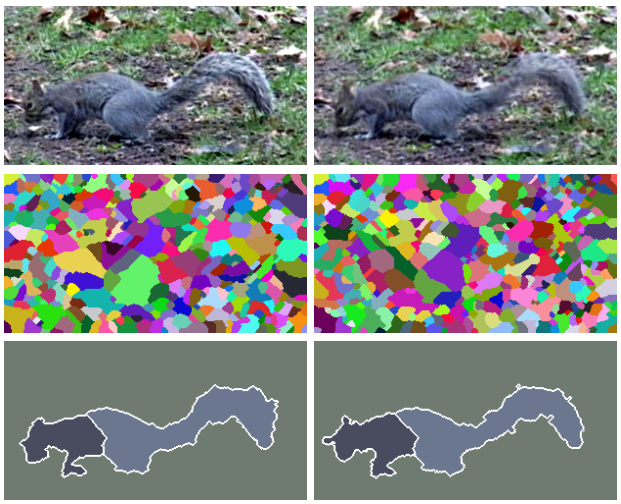


Figure 1: **Co-clustering super-pixels**. Oversegmentations of two consecutive frames from the 'squirrel3' sequence [14] are generated using a watershed transform (first and second rows). Our shape-based co-clustering method seeks to maximize the agreement between clusters of super-pixels across images. The result obtained by our algorithm is shown in the bottom row. Each segment is shown with its average color and surrounded by a white outline.

Specifically, we approach the problem of co-clustering of segments (super-pixels) by defining a quadratic optimization function with complex-valued affinities in order to optimally match the bounding contour elements of the clustered segments. We achieve this by defining a measure that is *additive* with respect to segments. For a union of seg-

*Equal contribution authors, listed alphabetically.

†Research was supported in part by the Israel Science Foundation grant number 764/10. The vision group at the Weizmann Institute is supported in part by the Moross Laboratory for Vision Research and Robotics.

ments, this measure sums the score of matching the contour elements at the exterior bounding contour of the union while ignoring the contributions of elements that fall in the interior of the union. This measure can then be combined with analogous measures that utilize region information. To optimize our measure we follow the convex relaxations of [19] and cast our problem as a linear program with a number of variables that depends only on the number of segments. Furthermore our optimization determines the number of clusters automatically. We evaluate our approach on the "Video Dataset for Occlusion/Object Boundary Detection" benchmark [13]. Figure 1 illustrates our results, showing two video frames from the benchmark, their segmentation to super-pixels, and our co-clustering results.

2. Related work

The problem of co-clustering of image segments was recently addressed in [19], where it was posed as a convex optimization problem and applied to electron microscopy images. There the co-clustering aimed to maximize agreement between clusters of segments across images. The agreement was measured using mere *region information*. The authors suggested two measures: pixel area overlap and merge-confidences computed by a boosted classifier. The latter compares the color histograms of the segments, while the former maximizes the pixel overlap between corresponding segments (or, equivalently, minimizes the symmetric difference). These measures may not be ideal for *shape comparison*, however. Shapes often differ by the composition of their (possibly narrow) protrusions. Such protrusions may be semantically important, but they contribute very little to the pixel-wise difference. Furthermore, even small translations of the same shape can result in a large drop in pixel overlap.

Closely-related to segment co-clustering is the problem of co-segmentation, first introduced by [10] and studied by e.g., [6, 8]. See [18] for a good recent overview of the problem and existing methods. These methods take as input two or more images containing a single common foreground object with varying backgrounds, and attempt to segment the foreground object from the background.

Co-segmentation algorithms are best-suited for handling images whose content differs significantly, but whose common object maintains its color/texture. Our formulation, in contrast, allows for multiple common objects. It aims to generate a full image segmentation identifying all of the objects in the image. Our method is designed to handle closely related images, such as two consecutive video-frames, in which the objects undergo only moderate deformation. Co-segmentation algorithms would generally fail on such input, since in consecutive frames the background objects also maintain their appearance.

The problem we address is also related to motion segmentation. However, unlike many motion segmentation studies ([3, 7]), we do not assume parametric motion for the individual segments.

In other related work, [1] segments multiple images while simultaneously learning class models. [16] seeks to match segments across images by optimizing their "co-saliency" using inter and intra image interactions. Finally, [17] provides an elegant shape representation based on chord histograms and employs it within an SDP optimization framework to detect model shapes in superpixel maps.

3. Joint Clustering Formulation

Let $\{I^{(i)}\}_{i=1}^{N}$ be a sequence of closely-related images in which the same objects are present and roughly maintain their shapes. For simplicity, let us assume that all images are defined on the same domain $\Omega \in \mathbb{R}^2$. Let $\{P^{(i)}\}_{i=1}^{N}$ be partitions of Ω generated by segmenting the respective images. Each of these segmentations is of the form $P^{(i)} = \{S_1^{(i)}, S_2^{(i)}, ..., S_{n_i}^{(i)}\}$, so that $\Omega = \cup_{j=1}^{n_i} S_j^{(i)}$, and n_i is the number of segments in $P^{(i)}$.

A joint clustering of all segments in all images $\cup_{i=1}^{N} P^{(i)}$, is defined by a binary-valued matrix X of size $n \times c$ where $n = \sum_{i=1}^{N} n_i$ is the total number of segments in all images and c is the number of clusters in the joint clustering. A column x_k of the matrix $X = (x_1, x_2, \cdots, x_c)$ corresponds to a single cluster, which consists of subsets of segments from the different images. Each x_k is a concatenation $x_k = \left(x_k^{(1)T}, x_k^{(2)T}, \cdots, x_k^{(N)T}\right)^T$, where each $x_k^{(i)}$ of size n_i indicates the segments from image i that participate in cluster k. By requiring X to have unit norm rows, we enforce the constraints that each segment is assigned to exactly one cluster.

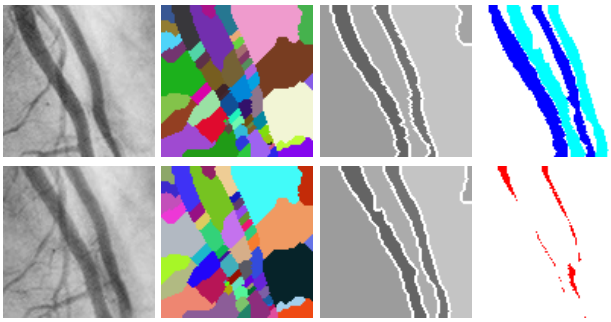
We measure the quality of a cluster by considering the intra and inter image interactions between the subsets of segments chosen from each image. We construct a complex-valued Hermitian affinity matrix

$$Q = \begin{pmatrix} Q^{(1,1)} & \cdots & Q^{(1,N)} \\ \vdots & \ddots & \vdots \\ Q^{(N,1)} & \cdots & Q^{(N,N)} \end{pmatrix} \tag{1}$$

of size $n \times n$ with N^2 blocks. The score associated with a clustering matrix X is given by

$$\mathrm{tr}(X^T Q X) = \sum_{k=1}^{c} x_k^T Q x_k = \sum_{k=1}^{c} \sum_{i=1}^{n} \sum_{j=1}^{n} x_k^{(i)T} Q^{(i,j)} x_k^{(j)}. \tag{2}$$

The diagonal-block terms $x_k^{(i)T} Q^{(i,i)} x_k^{(i)}$ measure the interactions amongst a subset of segments from the same image. The off-diagonal-block terms $x_k^{(i)T} Q^{(i,j)} x_k^{(j)}$ for $i \neq j$



(a) (b) (c) (d)

Figure 2: **Shape vs. symmetric difference**. Two frames taken from an x-ray cardiac image sequence (2a), their over-segmentations (2b) and the result of our shape-based joint clustering algorithm (2c). 2d shows the symmetric difference (blue and light blue) and the intersection (red) of the clusters corresponding to the main arteries in the two images, indicating that a method based on region overlap is likely to fail to cluster them correctly. Our shape-based method is better-suited to this example than the symmetric difference approach of [19].

measure interactions between two distinct subsets of the segments from images i and j which participate in cluster k.

3.1. Inter Image Interactions

Recall that we assume that each pair of over-segmentations $(P^{(i)}, P^{(j)})$ roughly agree on the correct object boundaries, but disagree otherwise. We wish to construct an inter-image interaction matrix that (1) encourages agreement on the boundary edges of the correct segmentation and (2) allows for moderate translations and deformations even when such distortions affect the area of overlap of the correct segments across images. Figure 2 illustrates possible shortcomings of a measure that relies on the area of overlap of the segments rather than on the shape-based approach.

We therefore propose a novel method that seeks to match the bounding edge elements of unions of segments in a cluster. Key to our method is the use of an *additive score function*. This function sums the scores of matching the individual contour elements of segments, such that only the elements in the exterior bounding contours of the union participate in determining the score, while the elements that fall in the interior of the union cancel out.

Definition Let $P = \{S_1, ..., S_n\}$ be a segmentation and $\{b_J\}_{J\subseteq\{1,...,n\}}$ be a representation of the subsets of segments. We call $\{b_J\}$ an *additive representation* if

$$b_{J_1\cup J_2} = b_{J_1} + b_{J_2} \tag{3}$$

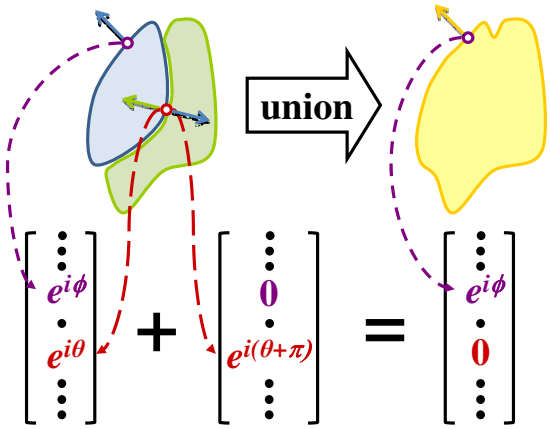


Figure 3: **Additive representation**. A segment, or a union of segments, is represented by a complex-valued vector. Non-zero entries represent the angle of the normal to the edge elements along a segment's contour. The blue and green segments share a common boundary. Therefore, their representations share non-zero entries for edge elements at the common portion of the boundary. However, the normal angles at these common boundaries differ by π. As a result when the two representation vectors are added, the entries of the shared edge elements vanish. These are exactly the edge elements in the interior of the union. Consequently the resulting vector represents only those edge elements which lie along the exterior boundary of the entire union.

for all disjoint subsets $J_1, J_2 \subseteq \{1, \ldots, n\}, J_1 \cap J_2 = \emptyset$.

We construct an additive representation as follows. Consider in each image a set of q_i *edge elements*, densely sampled along the boundaries of the segments in $P^{(i)}$. We describe the union of a subset of segments $S_J^{(i)} = \cup_{j\in J} S_j^{(i)}$ using a vector $b_J^{(i)} \in \mathbb{C}^{q_i}$ with complex-valued elements. For each contour element $k \in \{1, \ldots, q_i\}$ that bounds the union $S_J^{(i)}$ we set $b_J^{(i)} = e^{\imath\theta_k}$, where $\imath = \sqrt{-1}$ and θ_k is the angle between the outward-pointing normal of $S_J^{(i)}$ at the kth contour element and the X-axis. Let $B^{(i)} = \left(b_1^{(i)}, b_2^{(i)}, \cdots, b_{n_i}^{(i)}\right)$ of size $q_i \times n_i$ be a matrix describing the segments in image i. Each column $b_j^{(i)}$ of the matrix $B^{(i)}$ describes the orientations of all the edges along the contour of segment $S_j^{(i)}$. In general, we allow each contour element to participate in exactly two segments, one with $e^{\imath\theta_k}$ and the other pointing in the opposite direction, i.e. $e^{\imath(\pi+\theta_k)} = -e^{\imath\theta_k}$. The rest of the elements in $B^{(i)}$ are set to zero.

Note that our suggested representation is indeed additive. Consider the union of two segments, $S' = S_1 \cup S_2$. In terms of the boundary elements, the corresponding operation is $b' = b_1^{(k)} + b_2^{(k)}$. Boundary elements exclusive to either of the two segments are retained with no change,

while boundary elements common to them have opposing normals and cancel to zero. This idea is illustrated in Figure 3. More generally, consider the application $B^{(i)}x^{(i)}$ where $x^{(i)} \in \{0,1\}^{n_i}$ indicates a subset of segments in $P^{(i)}$. Let $C = \cup_{\{j|x_j^{(i)}=1\}} S_j^{(i)}$. Then $B^{(i)}x^{(i)}$ is a vector in $\mathbb{C}^{q_i}$ whose elements encode the outward normal angles of the bounding contour of C. The internal contours are eliminated, since for such contour elements, $e^{\imath\theta_k} + e^{\imath(\pi+\theta_k)} = 0$.

To encode potential matches between contour elements in a pair of images $I^{(i)}$ and $I^{(j)}$ with q_i and q_j contour elements respectively we define a real $q_i \times q_j$ matrix $W^{(i,j)}$. In an idealized setting in which $q_i = q_j$ and we have a one-to-one correspondence between the contour elements, $W^{(i,j)}$ would be a permutation matrix. In general, however, we do not commit to a specific one-to-one correspondence; instead we assign soft values to $W^{(i,j)}$. Details on the construction of $W^{(i,j)}$ are provided in Section 3.1.1.

A compatible segmentation of $I^{(i)}$ and $I^{(j)}$ should be one in which many contour elements in $I^{(i)}$ find good matches in $I^{(j)}$. A good match between contour elements is one for which the score in $W^{(i,j)}$ is high and the orientations of the elements are similar. We capture both requirements in the bilinear form

$$(B^{(i)}x^{(i)})^H W^{(i,j)} (B^{(j)}x^{(j)}), \tag{4}$$

where X^H denotes the Hermitian transpose of X. In this form, $B^{(i)}x^{(i)}$ represents the bounding contour elements of the union of the segments in $I^{(i)}$ indicated by $x^{(i)}$. For each two contour elements k in $I^{(i)}$ and l in $I^{(j)}$ this form produces a term of the form

$$e^{-\imath\theta_k} W_{k,l}^{(i,j)} e^{\imath\theta_l} = W_{k,l}^{(i,j)} e^{\imath(\theta_l - \theta_k)}, \tag{5}$$

where the minus sign in the leftmost exponent is due to the Hermitian conjugate. Equation (4) sums these terms for all pairs of elements. If we now add (4) to its Hermitian transpose, i.e., $(B^{(i)}x^{(i)})^H W^{(i,j)} (B^{(j)}x^{(j)}) + (B^{(j)}x^{(j)})^H W^{(j,i)} (B^{(i)}x^{(i)})$ where $W^{(j,i)} = W^{(i,j)\,T}$, we obtain for every pair of contour elements k and l a real-valued term proportional to the cosine of the angle between them, $W_{k,l}^{(i,j)} \cos(\theta_l - \theta_k)$.

We can now ready define the inter image interaction matrix:

$$Q^{(i,j)} = B^{(i)H} W^{(i,j)} B^{(j)}. \tag{6}$$

The $n_i \times n_j$ matrix $Q^{(i,j)}$ encodes the interaction between the input segments of $I^{(i)}$ and $I^{(j)}$. Consequently, once the $Q^{(i,j)}$'s are computed, the rest of the optimization depends only on the number of segments and is independent of the number of boundary elements.

We should emphasize, that $Q^{(i,j)}$ is a complex-valued matrix whose entries summarize the interactions between *all* the contour elements in each pair of segments in a way that is ready for additive manipulations, as per our definition. Consequently, once we apply $Q^{(i,j)}$ to a pair of indicator vectors, i.e., $x^{(i)\,H} Q^{(i,j)} x^{(j)}$, it will sum the complex-valued scores for all pairs of segments indicated by $x^{(i)}$ and $x^{(j)}$. Due to the additivity property, all the interactions that involve contour elements in the interior of the unions of these segments will vanish, and only the interactions between pairs of contour elements on the exterior boundaries of the unions of these segments will contribute to the score.

Finally, we define $Q^{(j,i)} = Q^{(i,j)H}$, ensuring that the score function in Equation (2) is real.

3.1.1 Inter Image Correspondence

The matrix $W^{(i,j)}$, which encodes the correspondence between contour elements in images $I^{(i)}$ and $I^{(j)}$, plays a critical role in our framework; the accuracy of our clustering will increase if we assign high scores to the correct correspondences. However, identifying the correct correspondences prior to optimization can be difficult, and so we will set $W^{(i,j)}$ with soft values to allow multiple matching options for each contour element. Our experimental results demonstrate that our formulation is able to cope with such matching ambiguities.

Ideally we would like the q_i rows and q_j columns of $W^{(i,j)}$ to encode a matching probability for each edge-element in one image with all the edge elements in the other image. To achieve this we initialize $W^{(i,j)}(k,l) = \exp\left((f_k - f_l)^T \Sigma^{-1} (f_k - f_l)\right)$, where f_k and f_l are feature vectors computed for contour elements k and l in images i and j respectively. The feature vectors are concatenations of multiple cues. Σ is a diagonal matrix with weights proportional to the estimated variances of the cues. We use a HOG-type descriptor [5] along with the edge direction and its location. Affinities between descriptors at a distance of more than 10 pixels are truncated to 0. Finally, for smoothing we add a small positive constant to all entries of W.

Since we want the rows and columns oof $W^{(i,j)}$ to encode matching probabilities we apply an iterative procedure of normalizing the row sums to $1/q_i$ and the column sums to $1/q_j$ alternately. In [12] it is shown that when all the elements of W are strictly positive such a procedure is guaranteed to converge to a matrix with any prescribed row and column sums. When W is square ($q_i = q_j$) this procedure will produce a (scaled) doubly stochastic matrix. According to the Birkhoff–von Neumann theorem, doubly stochastic matrices are convex combinations of permutation matrices, and so they can be interpreted as mixtures of possible one-to-one matchings.

3.2. Intra Image Interactions

The intra image interaction matrix $Q^{(j,j)}$ can be used to encode the affinity between different segments within image

$I^{(j)}$. This matrix plays an analogous role to affinity matrices in standard segmentation algorithms.

Below we use an affinity matrix whose entries are products of two terms. The first term stipulates that two segments are more likely to be clustered when their shared bounding contour is longer. The second term determines the likelihood of a clustering based on the similarity between the segments. For two segments $S_k^{(j)}$ and $S_l^{(j)}$ in $P^{(j)}$ $(1 \leq k, l \leq n_j)$ let $v_{k,l}^{(j)}$ denote the length of the common boundary of $S_k^{(j)}$ and $S_l^{(j)}$. Let $u_{k,l}^{(j)}$ further express our belief that segments k and l belong to the same cluster. (For example, in Section 5.1 we construct $u_{k,l}^{(j)}$ by assessing the similarity of $S_k^{(j)}$ and $S_l^{(j)}$ in both color and motion.) We then set

$$Q_{k,l}^{(j,j)} = \lambda v_{k,l}^{(j)} u_{k,l}^{(j)}. \tag{7}$$

$Q^{(j,j)}$ is $n_j \times n_j$ real symmetric, and $\lambda \geq 0$ is a preset parameter that controls the fragmentation of the final clustering result. Higher values of λ lead to a smaller number of clusters by encouraging mergers between similar segments with long common boundaries.

3.3. Optimization of the Clustering Objective

Using (2) our optimization objective is

$$\begin{aligned} \max_X \quad & \mathrm{tr}(X^T Q X) \\ \text{s.t.} \quad & X_{i,j} \in \{0,1\}\ \forall i,j \quad \sum_j X_{ij} = 1\ \ \forall i. \end{aligned} \tag{8}$$

This is a Quadratic Semi-Assignment Problem (QSAP) [19], and can be written as

$$\begin{aligned} \max_Y \quad & \mathrm{tr}(QY) \\ \text{s.t.:} \quad & Y \succeq 0\ ,\ Y_{ij} \in \{0,1\}\ \text{ and }\ Y_{ii} = 1, \end{aligned} \tag{9}$$

where $Y = XX^T$ is of (unknown) rank c. The requirement that every segment participate in exactly one cluster is expressed in the constraint $Y_{ii} = 1$.

In [2] Charikar et al. present a Linear Programming relaxation for QSAPs. They compute the optimization function from distances between segments in the cluster space. Metric properties are imposed on the distances between the segments using linear constraints that enforce non-negativity, symmetry, and triangular inequality.

The LP relaxation has a crucial limitation for practical applications – the number of triangular inequalities grows as $O(n^3)$ where n is the number of input segments. [19] presents a further relaxation by enforcing the metric property only within cliques in an adjacency graph on the segments. They showed that for the problem of co-clustering image segments, this bounds the number of constraints to $O(n^2)$, and in practice is almost linear in n. Both their and our experiments indicate that the method is very efficient with good empirical performance. Denoting by D the matrix of segment distance, $d_{i,j} = 0$ implies that segment i and j should belong to the same cluster. Our final optimization problem becomes

$$\begin{aligned} \min_D \quad & \sum_{i,j} q_{i,j} d_{i,j} \\ \text{s.t.} \quad & 0 \leq d_{i,j} \leq 1 \\ & d_{i,i} = 0\ \ \forall i, \quad d_{i,j} = d_{j,i}\ \ \forall i,j \\ & d_{i,j} \leq d_{i,k} + d_{k,j}\ \ \forall e_{i,j}, e_{i,k}, e_{k,j} \in A, \end{aligned} \tag{10}$$

where A encodes the reduced sparse connectivity. In our case, A is the region adjacency graph computed from the input segmentation maps.

We note that the approach of [19] was subject to trivial solutions which had to be avoided by adding a regularizing parameter. For example, the minimal symmetric difference between clusters is trivially obtained when all segments are put into one cluster. In contrast, our approach of co-clustering maximizes the sum of support for segment boundary elements remaining after the mergers, and naturally avoids trivial mergers. The trivial solution of putting all segments into one cluster eliminates all boundary elements from the optimization so their contribution vanishes. The trivial solution of making no mergers at all leaves the objects in the image fragmented - decreasing the total support. The optimal operating point therefore lies somewhere in between these two ends of the spectrum.

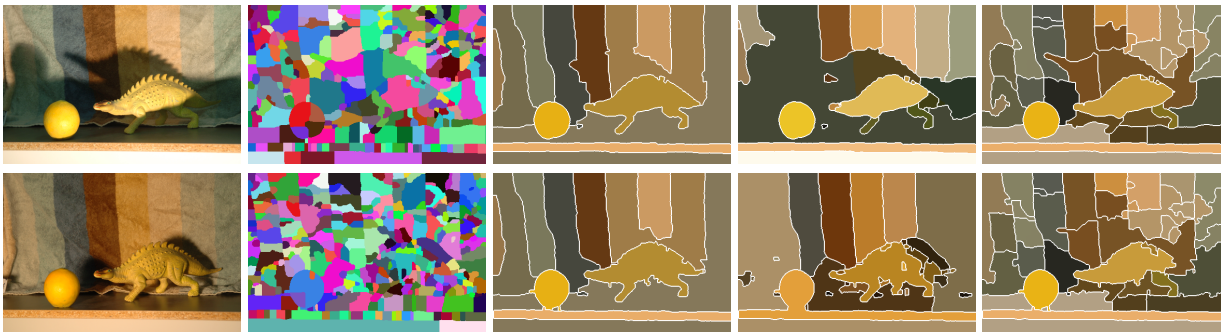
3.4. Co-clustering Example

We conclude this section by illustrating the contributions of the components of the co-clustering framework. Figure 4 shows an application of our framework to simultaneously segment multiple objects taken under different lighting conditions. The scene has strong variation in color, reflectance, shadows and shading. Our approach correctly matches the segments corresponding to the orange and dinosaur across the images, and obtains the correct segmentation. Notice that the intra-image regional affinities on their own are insufficient to handle the deep shadows cast on the dinosaur, and shape information on its own is insufficient to handle cases when the false boundaries in the oversegmentation happen to coincide. However, taken together, these two cues obtain the correct segmentation.

4. Segmentation of Video Frames

To obtain a segmentation result on a designated reference frame from a sequence, we suggest a two-step process with clustering as a sub-routine. The process is illustrated in Fig. 5.

First, for each frame $I^{(i)}$, we compute a boundary map using gPb [9]. For each frame, a superpixel map, $P_w^{(i)}$, is



(a) Input images (b) Initial super-pixels (c) Joint clustering (d) Only intra (e) Only inter

Figure 4: **Lighting example**. The dinosaur and the orange were photographed under different lighting conditions 4a. The input oversegmentations can be seen in 4b. There are 247 super-pixels for image 1 and 320 for image 2. The result of our joint-clustering algorithm is shown in 4c. The dinosaur and the orange are common to both images and are put in distinct segments; in contrast, their shadows are merged with the background as they shift under illumination variation.. Results obtained by restricting the method to use the intra-image affinities or inter-image *alone* are shown in 4d, and 4e respectively.

computed by running watershed on the boundary map. In addition, m *randomized oversegmentations* are computed by randomly seeding watershed, denoted by $P_{rw}^{(i)}$'s.

Next, the randomized oversegmentations, $P_{rw}^{(i)}$'s, are co-clustered with $P_w^{(i)}$. The co-clustering procedures result in m co-occurrence matrices $\{Y_j^{(i)}\}_{j=1}^m$. The consensus segment map frame i is obtained by $\tilde{Y}^{(i)} = \frac{1}{m}\sum_{j=1}^m \tilde{Y}_j^{(i)} \geq \tau$ where $\tilde{Y}_j^{(i)}$ is the sub-matrix of $Y_j^{(i)}$ corresponding to the super-pixels of $P_w^{(i)}$. The parameter $0 \leq \tau \leq 1$ is a threshold on the fraction of joint clustering results which should agree on a merge. Note that the idea of using multiple segmentations of a single input image for reliable image analysis has been gaining ground in the community, e.g., [11, 4]. Unlike our method, however, these works do not produce full image segmentations.

In the second stage, subsets of frames are jointly-clustered along with the reference frame. The co-clustering result with the highest optimization objective value (2) is chosen as the final segment map.

Combining multiple randomized segmentations for each frame provides two advantages: (1) robustness to possible false mergers in any few random oversegmentations, and (2) helps reduce repetition of false segment boundaries in different frames. Joint clustering of segmentations across frames emphasizes object boundaries that persist under small movements and camera motion.

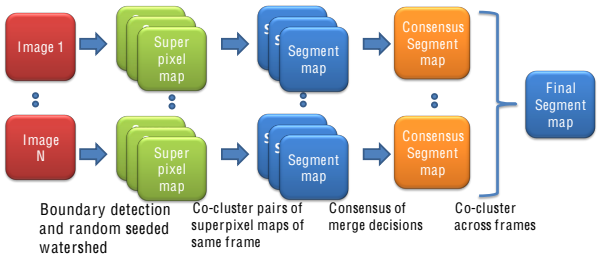


Figure 5: Schematic of joint clustering schedule.

5. Experimental Results

5.1. Video Dataset for Occlusion/Object Boundary Detection

We evaluate our framework on the "Video Dataset for Occlusion/Object Boundary Detection" of Stein and Hebert [13]. This dataset is appropriate because unlike other segmentation benchmarks it includes more than one image per object.

The dataset includes 30 short image sequences, and is quite challenging, with a variety of indoor and outdoor scene types, significant noise and compression artifacts, unconstrained handheld camera motions, and some moving objects. The occlusion boundaries are labeled as ground truth in the reference (middle) frame of each sequence. The task is to detect the occlusion boundaries delineating the foreground objects.

We use the procedure described in Section 4 to generate a segmentation of the reference frame in each sequence, comparing the result to the provided ground truth. For the inter-frame stage we use the global alignment provided with the dataset. We generate results on all 38 objects for which [14] report segmentation results.

Motion cues are necessary in order to detect the occlusion boundaries. Accurate motion estimation is not a trivial task. The segmentation results reported in [14] rely on occlusion boundaries detected using two sophisticated motion

cues [13]. For our experiments we use a very simple motion cue, namely the median optical flow within a segment. We then set our intra-image affinities to

$$u_{k,l}^{(j)} = \frac{1}{2}\left[\exp\left(\frac{-\|\mathbf{d}_k - \mathbf{d}_l\|^2}{\sigma_c^2}\right) + \exp\left(\frac{-\|\mathbf{f}_k - \mathbf{f}_l\|^2}{\sigma_f^2}\right)\right], \tag{11}$$

where $\mathbf{d}_k$ is the normalized $L^*a^*b^*$ color histogram of segment k, and $\mathbf{f}_k$ is the median optical flow inside it computed using [15]. We use $\sigma_c = 0.3$ and $\sigma_f = 0.1$.

5.2. Segmentation Evaluation

We compare our results to those reported in [14] using their evaluation methodology. For any two unions of subsets of segments R and G define their *consistency* as $c(R,G) = \frac{|R \cap G|}{|R \cup G|}$. The *efficiency* is defined as the size of the minimal subset of segments, R, required to achieve a specified desired consistency, c_d, according to the ground truth object segmentation, G. Formally,

$$e_{c_d}(S,G) = \min |R|, \;\; \text{s.t.} \;\; c(R,G) \geq c_d. \tag{12}$$

In [14] the authors generate a sequence of 19 increasingly finer segmentations for each image, with the number of segments varying between 2 and 20. For each object and for each of ten desired consistency levels, from 0.5 to 0.95, they report an efficiency value. This efficiency is the minimizer of Equation 12 over all subsets of segments and over all 19 different segmentations in the collection.

Our algorithm determines the number of clusters automatically. Therefore, we use settings of parameters τ and λ to generate a sequence of increasingly finer segmentations. The higher the value of τ, the stricter the merging of randomized segmentations. Higher values of λ encourage mergers within a frame. We use a fixed set $\Theta = \{(\tau_i, \lambda_i)\}_{i=1}^{19}$ [1] to generate the segmentation sequences for all objects. Θ was selected from a range of subsets of size 19, as the one that produced the best results.

We present a quantitative summary of our results compared to those reported by [14] in Figure 6. Note that for the majority of the objects our method achieves the same consistency levels with fewer segments. Some of our segmentation results can be seen in Figure 7 alongside those of Stein *et al.* More detailed graphs and the full quantitative and qualitative results and comparison can be found in the attached material.

We also compare to a method based on area of overlap following [19]. We reproduce the experimental schedule substituting our shape-based inter-image complex-valued affinities with $[0,1]$-valued affinities measuring the ratio between area of intersection and area of union. A summary of this comparison is shown in Figure 8. Overall consistency bins and all objects our method achieves an average efficiency of 2.81 compared to 3.21 achieved by [14] and 3.37 achieved by the overlap method.

[1] $\Theta = \{(.5,.25)(.5,1.25)(.5,2)(.5,4.5)(.6,.25)(.6,1.25)(.6,1.75)(.7,1.75)(.7,2)(.7,4.5)(.8,1.25)(.8,2.5)(.8,4.5)(.9,.25)(.9,1.25)(.9,1.75)(.9,2)(.9,2.5)(.9,4.5)\}$

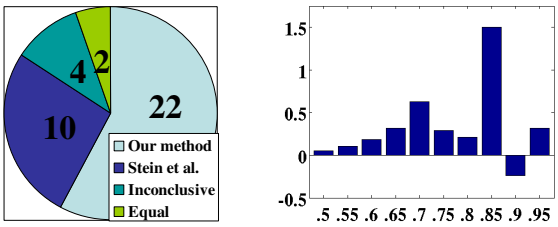


Figure 6: **Quantitative segmentation results**. Here we compare segmentation results to those of [14]. The pie chart shows a partition of the 38 objects. On 22 objects we outperform [14], in that we do at least as good at all consistency levels and do *better* on at least one consistency level. In 10 of the cases [14] outperforms our results in the same sense. For 4 objects the result is inconclusive (i.e. we do better at some values while [14] is better at others.) Finally, on two objects we have identical performance. The bar chart on the right shows the *average efficiency gain*. We measure for each object, and each consistency level, how many segments *less* than the result reported in [14] we needed, in order to achieve at least the designated consistency level. The average is over all objects in the data-set.

We note that the authors of [14] compare their method to three others, "Color Distribution Affinity", Multiscale N-Cuts, and the Berkeley Segmentation Engine (BSE). For clarity of presentation we do not show those results in the current comparison, and concentrate on the comparison to the method of [14] which significantly outperforms all three.

Running un-optimized Matlab code on a 1GB laptop we compute a co-clustering result on a pair of images with hundreds of super-pixels in less than a minute. For the linear program we use the open-source solver lp_solve.

6. Conclusion

We presented an approach to jointly segment multiple closely-related images by combining contour and region information, and applied our approach to video data. The

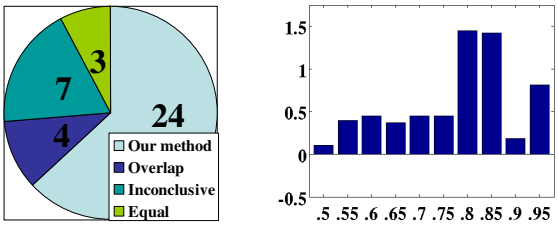


Figure 8: **Comparison to a method based on area of overlap**. Format identical to that of Figure 6.

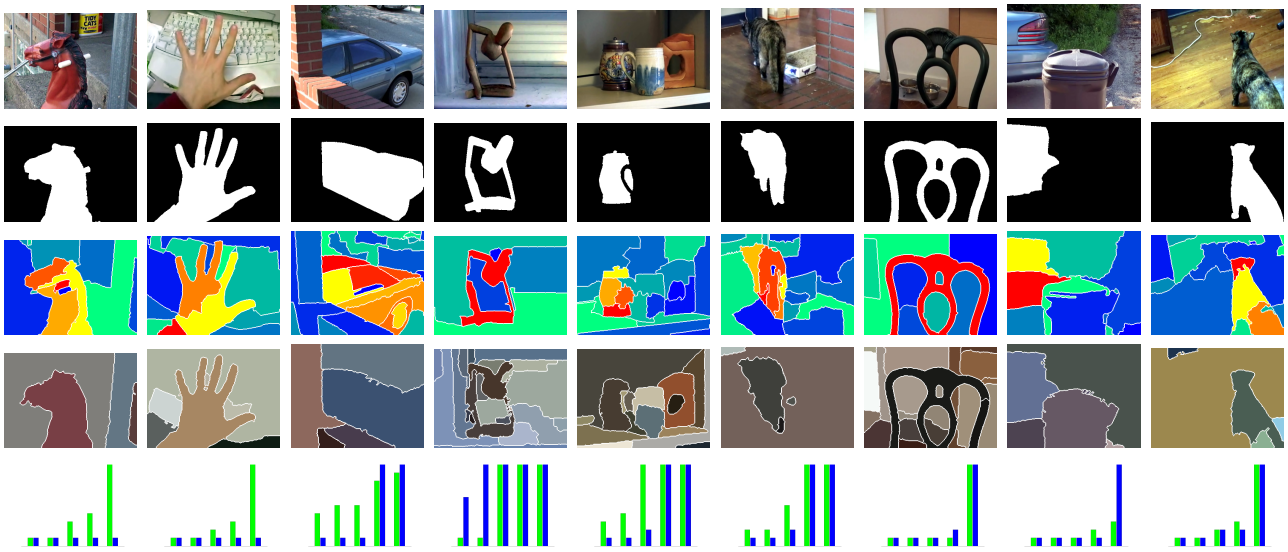


Figure 7: **Segmentation results**. This figure shows segmentation results on a number of sequences from [13]. The top row shows the reference frame from each sequence. The second row shows the ground truth segmentation of a designated object in each image. In the third row we show the segmentation result of [14] corresponding to the highest consistency achieved by their method for each one of the objects. We extracted these images from the attached material of that paper. Similarly we show our highest consistency results in the fourth row. Finally in the bottom row we report the minimal number of segments needed to achieve a desired consistency level. Our results (in blue) are compared to [14] (in green) at the top 5 consistency levels (between 0.75 and 0.95). Lower values indicate that a given consistency was achieved with fewer segments. The highest bars have a value of 10 segments, this value was assigned to consistency levels that could not be attained.

problem was posed as quadratic optimization and solved using LP relaxation. We proposed an additive representation for segment boundary contours such that when taking the unions of segments, only the exterior bounding contour portions contribute to the score, while the contributions of boundaries common to the merged segments are eliminated from the optimization. While we have presented this representation in the context of co-clustering of image segments, we believe it can also be applied in other domains of computer vision. In future work we plan to explore how to incorporate into our approach top-down cues such as those available from object detections or user input.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. ClassCut for Unsupervised Class Segmentation. *ECCV*, 2010. 2386

[2] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. *Journal of Computer and System Sciences*, 71(3), 2005. 2389

[3] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *IJCV*, 29(3), 1998. 2386

[4] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010. 2390

[5] D. Glasner and G. Shakhnarovich. Nonparametric voting architecture for object detection. *TTI-C Technical Report*, (1), 2011. 2388

[6] D. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009. 2386

[7] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *IJCV*, 12(1), 1994. 2386

[8] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010. 2386

[9] M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008. 2389

[10] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into MRFs. 2006. 2386

[11] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 2390

[12] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *Am. Math. Monthly*, 1967. 2388

[13] A. Stein and M. Hebert. Occlusion boundaries from motion: low-level detection and mid-level reasoning. *IJCV*, 82(3), 2009. 2386, 2390, 2391, 2392

[14] A. Stein, T. Stepleton, and M. Hebert. Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection. In *CVPR*, 2008. 2385, 2390, 2391, 2392

[15] D. Sun, S. Roth, and M. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 2391

[16] A. Toshev, J. Shi, and K. Daniilidis. Image matching via saliency region correspondences. In *CVPR*, 2007. 2386

[17] A. Toshev, B. Taskar, and K. Daniilidis. Object detection via boundary structure segmentation. In *CVPR*, 2010. 2386

[18] S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation Revisited: Models and Optimization. *ECCV*, 2010. 2386

[19] S. Vitaladevuni and R. Basri. Co-clustering of image segments using convex optimization applied to EM neuronal reconstruction. In *CVPR*, 2010. 2386, 2387, 2389, 2391