A General Framework for Reconciling Multiple Weak Segmentations of an Image

Soumya Ghosh
Department of Computer Science
Brown University
Providence, RI 02904
sghosh@cs.brown.edu

Joseph J. Pfeiffer, III
Department of Computer Science
Purdue University
West Lafayette, IN 47907
jpfeiffer@purdue.edu

Jane Mulligan
Department of Computer Science
University Of Colorado
Boulder, CO 80309
jane.mulligan@colorado.edu

Abstract

Segmentation, or partitioning images into internally homogeneous regions, is an important first step in many Computer Vision tasks. In this paper, we attack the segmentation problem using an ensemble of low cost image segmentations. These segmentations are reconciled by applying recent techniques from the consensus clustering literature which exploit a Non-negative Matrix Factorization (NMF) framework. We describe extensions to these methods that scale them for large images and also incorporate smoothness constraints. This framework allows us to uniformly and easily combine segmentations from different algorithms or feature modalities. We then demonstrate that popular bottom up image segmentation algorithms, Mean Shift and Efficient Graph Based segmentation, perform no better than our simple combination of multiple image segmentations derived from k-means clustering (of various feature spaces) or from "naive" RGB quantizations. The algorithms are evaluated on the Berkeley image segmentation dataset.

1. Introduction

Image segmentation, or partitioning an image into regions with internal segment coherence, has a long history in the Computer Vision literature and yet still has no generally accepted solution [24]. The goal is to represent the image with fewer, more meaningful parts, which makes processing more tractable and robust for subsequent vision tasks such as correspondence [27] or terrain classification [9].

In general, we want to assign pixels to segments based on their similarity to other members of that segment and dissimilarity to those of other segments. This implies a distance or homogeneity metric based on some set of cues which may include image distance, various region properties such as color and texture, or boundary/gradient information. Often these different feature modalities give us different results for the pixel and region similarity required to define segments, and considerable work has been devoted to combining them [1, 3, 15, 18].

Clustering techniques are a natural approach to computing image segmentations and a great variety of methods have been applied to the problem. Generic methods such as K-means often have difficulty integrating the spatial continuity or smoothness implied by image segmentation. These limitations are overcome by more expensive graph-based techniques such as spectral clustering [6] or normalized cuts [20, 21], which explicitly represent neighborhood linkages. Amongst the many segmentation algorithms proposed in the literature, few work well on natural images, and all are finely tuned to work on certain sets of images. For example, algorithms which work well segmenting biological images rarely work well on outdoor scenes.

Our work is motivated by the observation that even though a single segmentation algorithm or similarity metric by itself might produce some poor segments, there often exist sub-parts of the image which it explains well. So if diverse segmentations explain different parts of the image well, an ensemble of these could produce a superior consensus segmentation than any of the original segmentations. Our contribution in this paper is to provide a general framework for seamlessly combining segmentations from heterogeneous sources of information.

The problem of combining multiple segmentations can be posed as a cluster ensemble problem. While classifier ensembles have been widely used in the Machine Learning and Data Mining communities, researchers have only recently started exploring cluster ensemble problems [22]. This is primarily because the cluster ensemble problem is inherently more difficult, since we no longer have well defined classes. From a linear algebra point of view clustering has been studied as a matrix factorization problem. Traditionally, SVD based methods have been used for this purpose. However, for image data (which is non-negative), the bases produced by these methods are not easily interpretable since they do not enforce non-negativity constraints. *Non-Negative Matrix Factorizations(NMF)* [10] produces a matrix factorization which respects non negative constraints, thereby producing directly interpretable and more representative bases. On a parallel front, recently Li *et al*. [11] have shown that consensus clustering (an algorithm for solving cluster ensembles) may be posed as a NMF problem. In this paper we propose the use of NMFs for finding the consensus segmentation. We also explore incorporating domain constraints in the consensus process to produce higher quality segmentation maps.

Cho and Meer [2] also proposed a consensus segmentation approach. The system is based on a bottom up Region Adjacency Graph (RAG) pyramid method which merges regions until a threshold on similarity is reached. The base segmentations are generated using a grayscale consistency metric, these are then used to compute a co-occurrence probability field for pixels grouped together in the segmentations. The probability field is in turn used as the metric to compute the final consensus segmentation again via the RAG pyramid. Our approach is related in that we search for an assignment of pixels to segments which best matches the mean co-occurrence $\tilde{M}_{ij}$ (see Section 2) for each pixel or object pair. We determine the final segmentation using NMF rather than a multiscale approach, which allows us to avoid setting thresholds on region similarity.

Zhang *et al*. [26] propose combining an ensemble of Spectral Clustering results computed using randomly generated scale parameters to construct a consensus segmentation of SAR images. The authors propose several approaches for combining segmentation maps including a majority voting scheme and a hypergraph-based metaclustering algorithm. They conclude that, of the voting and hypergraph techniques, the segmentation which maximizes sum of the normalized mutual information between the base segmentations and the consensus is the best solution. However, it has been shown in [11] that the NMF approach outperforms both the naive voting scheme and the more advanced hypergraph approach.

In fields such as object labeling and image retrieval, researchers prefer segmentation approaches which exploit prior knowledge or models related to the ultimate task goals. Obviously the more context and semantic information that can be included the better the segmentation will be from the task viewpoint. Our goal is to generate useful segmentations for tasks where there is no prior knowledge which can be applied, for example systems using superpixels to simplify images [9, 17], or using coherent patches for Stereo correspondence [27]. There are however several authors who exploit multiple segment maps which are worth mentioning in this context. Russel et al. [19] use multiple base segmentations for object labeling, but rather than combine the maps, they select the best segment among all maps for an object, based on learned object class appearance. Malisiewicz and Efros [13] explore whether arbitrarily shaped segments provide better support for object recognition and demonstrate that sampling many segmentation maps allows them to find tighter segments for objects. Hoiem et al. [8] attempt to label a scene with geometric classes again by sampling and evaluating multiple segmentations to find the one with the best spatial support for the labeling task. Rabinovich et al. [16] use a set of segmentations determined to stable under slight image changes, a signature for each segment is computed and used to classify it based on training image signatures. All of these systems use multiple segmentations as a sampling of segmentation space, to obtain the best support (tightest fit) for labeling objects in the scene.

Our NMF framework provides a flexible general method for combining a set of maps as well as additional constraints such as smoothness or potentially boundary information. The base maps themselves can arise from any combination of segmentation algorithm and feature modality. A significant advantage is the ability to combine information from many sources in a uniform way. Allowing each modality to contribute a map has advantages over combining attributes in a single high dimensional feature vector. We see in Figure 1, that the curse of dimensionality causes poor segmentation performance for a simple k-means based segmentation using the stacked features.

Techniques such as ours, which use connectivity (cooccurrence) matrices between pixels, present a problem due to their size. We describe a method to scale the problem by essentially computing regions which have a preconsensus: spatially linked pixels which belong to one segment in all segmentation maps. These superpixels or *objects* allow us to compute consensus segmentations even for large images. We also present a no bells and whistles alternative which is able to perform fast consensus segmentation.

Finally, Sections 2 and 3 present details of the proposed approach and Section 4 presents the evaluation of our system on the Berkeley image segmentation database and comparison of our results to those for Mean Shift [4] and Effi-

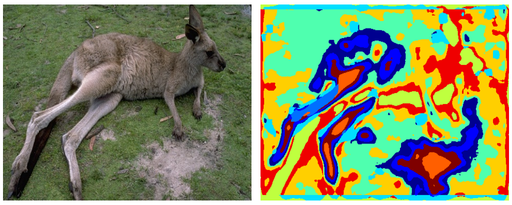

Figure 1. Image and its stacked K-means segmentation

cient Graph-based Segmentation [5].

2. Consensus Segmentation Framework

The consensus segmentation problem seeks to reconcile T different segmentations (base segmentations) of a $p \times q$ image. Equivalently, the consensus segmentation is a segmentation closest to all T segmentations. Let $B = \{S^1, S^2, ..., S^T\}$ be the set of base segmentations. For each segmentation S^t, we have K segments $\{S^t_1, S^t_2, ..., S^t_K\}$, where K is not necessarily the same for each segmentation S^t, and every pixel must belong to some segment S^t_k for each segmentation S^t. By representing each segmentation S^t as a $pq \times pq$ connectivity matrix, M:

$$M^t_{ij} = \begin{cases} 1 & (i,j) \in S^t_k \\ 0 & \text{Otherwise} \end{cases} \qquad (1)$$

we can compute the distance (Δ) between any two segmentations S^1 and S^2 as:

$$\Delta(S^1, S^2) = \sum_{i=1}^{pq} \sum_{j=1}^{pq} \delta_{ij}(S^1, S^2) \qquad (2)$$

where δ_{ij} is the pairwise pixel distance:

$$\delta_{ij}(S^1, S^2) = \begin{cases} 1 & (i,j) \in S^1_k and (i,j) \notin S^2_k \\ 1 & (i,j) \in S^2_k and (i,j) \notin S^1_k \\ 0 & \text{Otherwise} \end{cases} \qquad (3)$$

or equivalently

$$\delta_{ij}(S^1, S^2) = (M^1_{ij} - M^2_{ij})^2 \qquad (4)$$

Now, the problem of finding the consensus segmentation can be formulated as the following optimization [7]

$$\underset{S^*}{min} \frac{1}{T} \sum_{t=1}^{T} \Delta(S^t, S^*) = \underset{S^*}{min} \frac{1}{T} \sum_{t=1}^{T} \sum_{i,j=1}^{pq} [M^t_{ij} - M^{S^*}_{ij}]^2 \qquad (5)$$

equivalently,

$$\underset{U}{min} \sum_{i,j=1}^{pq} (\tilde{M}_{ij} - U_{ij})^2 \qquad (6)$$

where, $\tilde{M} = \frac{1}{T} \sum_{t=1}^{T} M_{ij}(S^t)$ and we adopt $U_{ij} = M^{S^*}_{ij}$ as the solution of the optimization problem for notational simplicity.

Unfortunately, we also have constraints on U that need to be dealt with. Consider any three pixels i, j, and l. Suppose $U_{ij} = 1$; that i and j belong to the same segment. If j and l belong to the same segment, then i and l must also belong to the same segment. However, if j and l do not belong to the same segment, then i and l cannot belong to the same segment. Now, consider the case where i and j belong to separate segments. We can now have i in the same segment as l, j in the same segment as k, or none of them in the same segment. These constraints can be expressed as [11]:

$$U_{ij} + U_{jl} - U_{il} \leq 1 \qquad (7)$$

$$U_{ij} - U_{jl} + U_{il} \leq 1 \qquad (8)$$

$$-U_{ij} + U_{jl} + U_{il} \leq 1 \qquad (9)$$

Note that the above constraints are indexed by individual pixels. Thus there are 3 constraints per pixel of the image. The constrained optimization problem turns out to be np-hard [12].

Following [11], we use an alternate specification of the above optimization problem using row stochastic(rows sum to 1) indicator matrices $H = \{0, 1\}^{n \times k}$. It is easy to see that $U = HH^T$. Our consensus segmentation problem now becomes:

$$\underset{H}{min} \parallel \tilde{M} - HH^T \parallel^2 \qquad (10)$$

with H restricted to the space of indicator matrices. However, since restricting H to be a indicator matrix is hard, we could reformulate the above problem as follows:

$$\underset{H^T H = D, H \geq 0}{min} \parallel \tilde{M} - HH^T \parallel^2 \qquad (11)$$

where $D = diag(H^T H)$. By restricting $H^T H$ to be diagonal we indirectly enforce the constraint that each row of H can have only one non zero element. However, this formulation involves a priori knowledge of D (cluster sizes) which is usually unavailable. As a result Eqn 11 is further reformulated as:

$$\underset{\tilde{H}^T \tilde{H} = I, \tilde{H}, D \geq 0}{min} \parallel \tilde{M} - \tilde{H} D \tilde{H}^T \parallel^2 \text{ s.t. } D \text{ Diagonal} \qquad (12)$$

where $HH^T = \tilde{H} D \tilde{H}^T$. D is now obtained as a solution to the optimization problem. In practice, the constraint on D being diagonal is relaxed to D being any symmetric nonnegative matrix, recasting the above problem as the familiar orthogonal nonnegative matrix tri-factorization problem, which is solved using iterative solution techniques.

Scaling. The algorithm as described above, however does not lend itself to be used practically for image segmentation. The major problem in adopting the above algorithm is the $\tilde{M}$ matrix. For a $p \times q$ image, the corresponding $\tilde{M}$ matrix has $pq \times pq$ entries. This quadratic growth in storage severely limits the size of images which can be processed by the algorithm. For instance, we found that, at best, we could work on 70×70 images on a computer with 2GB of RAM. To alleviate the scaling problem we observe that the NMF problem may be interpreted as finding the closest connectivity matrix to the given arbitrary (not necessarily a connectivity matrix) $\tilde{M}$ matrix. Now, if a certain set of pixels always occur in the same segment, establishing connectivity to any one of them is equivalent to explicitly establishing connectivity to all of them. Thus such sets can be collapsed into singleton entries in the $\tilde{M}$ matrix. We refer to the entries in the $\tilde{M}$ matrix as "objects". Note that these objects cover a wide gamut of sizes from large sets to singleton pixels. The dimension of the matrix to be processed now is $n \times n$, where $n < pq$ is the number of objects[1]. We find that employing such a scheme results in considerable savings both in terms of memory and computational cost, allowing us to comfortably processes 321×481 images. Figure 2 (Pre-consensed Image) displays the various objects of a 321×481 image.

Smoothness Constraints. Smoothness is incorporated in the consensus framework through additional constraints using the *Penalized Matrix Factorization* [25] formulation. We now minimize the following augmented objective function:

$$J = \min_{\tilde{H}^T\tilde{H}=I,\tilde{H},D\geq 0} \parallel \tilde{M} - \tilde{H}D\tilde{H}^T \parallel_F^2 + tr(\tilde{H}^T\Theta\tilde{H}) \tag{13}$$

where Θ is a $n \times n$ matrix which encodes the smoothness constraints.

We compute the pairwise overlap d_{ij} between the entries of $\tilde{M}$ (objects). We define overlap as the length(in pixels) of the shared boundary between two objects. The values Θ_{ij} are computed as follows:

$$\Theta_{ij} = \frac{1}{1 + e^{-(\frac{1}{d_{ij}})}} \tag{14}$$

Thus, objects having a smaller overlap would have a larger Θ_{ij} values[2]. The logistic function has the effect of normalizing the values in Θ to lie in the {0,1} range. Minimizing the augmented objective function has the effect of encouraging neighboring objects to have the same cluster label, thereby preferring smoother solutions to noisier ones.

The optimization problem of Equation13 does not result in closed form multiplicative updates needed for solving NMF problems. Instead, following [25] we solve a relaxed version of Equation 13 which only enforces the non-negativity of $\tilde{H}$. The necessary update equations are:

$$D = (\tilde{H}^T\tilde{H})^{-1}\tilde{H}^T\tilde{M}\tilde{H}(\tilde{H}^T\tilde{H})^{-1} \tag{15}$$

$$\tilde{H}_{ij} = \tilde{H}_{ij}\sqrt{\frac{(\tilde{M}\tilde{H}D)^+_{ij} + (\tilde{H}(D\tilde{H}^T\tilde{H}D)^-)_{ij}}{(\tilde{M}\tilde{H}D)^-_{ij} + (\tilde{H}(D\tilde{H}^T\tilde{H}D)^+)_{ij} + (\Theta\tilde{H})_{ij}}} \tag{16}$$

where

$$(\tilde{M}\tilde{H}D) = (\tilde{M}\tilde{H}D)^+ - (\tilde{M}\tilde{H}D)^- \tag{17}$$

$$(D\tilde{H}^T\tilde{H}D) = (D\tilde{H}^T\tilde{H}D)^+ - (D\tilde{H}^T\tilde{H}D)^- \tag{18}$$

The effect of the smoothness constraints on the resulting consensus segmentation is shown in Figure 2.

3. Algorithmic Details

Base segmentation diversity. We employ *k-means* to generate the base segmentations of the ensemble. Diversity in the base segmentations is an important prerequisite for avoiding degenerate solutions. We incorporate diversity in our base segmentations in three ways:

Firstly, we inject diversity in the feature space by using four different feature spaces

1. Hue-Saturation (HS): We use Hue and Saturation from the Hue-Saturation-Intensity color space. We leave out intensity in the hope of buying robustness to variation in lighting condition.

2. Laws Texture space (Tex): We compute the laws texture energy measure in a neighborhood of size varying between $\{3, 32\}$.

3. Color Histogram space (CH): We use a 15 bin histogram over the RGB color space, computed over a 5×5 window. Each color channel is allocated 5 bins.

The above set of features is meant to capture both color and texture variations. However, which features are optimal, often varies from image to image. Our flexible framework allows alternate feature spaces to be used just as easily.

Secondly, the base segmentations are computed at two different scales. This is achieved by using different k values in the k-means clustering. In this work we used values of {4,6}. We, thus form an ensemble of 6 segmentations.

Finally, we randomly initialize k-means and run it only for a modest number(200) of iterations. Terminating k-means when the algorithm might not have converged not only adds instability but is also computationally more efficient. Figure 2 shows the 6 base segmentations and the resulting final segmentation for an image from the Berkeley image segmentation database.

Iterative Optimization. Akin to other NMF optimization algorithms an iterative scheme is used in this paper. The $\tilde{H}$ matrix is initialized by clustering the matrix using

[1] In our experiments we found that for images with 321×481 pixels, n never exceeded 1200 and was frequently ≤ 800

[2] A small value is added to d_{ij} when i and j do not overlap. In our experiments we found a value of 0.1 to be effective.

$k - means$. Next the initial estimate of D is computed using 15 while fixing $\tilde{H}$ to it's initial value. This is followed by $\tilde{H}$ computation using 16 while keeping D fixed at it's previously computed value. This process of alternating minimization is repeated till either the change in J falls below a certain threshold ϵ or for a preset number of iterations. We set $\epsilon = 10^{-3}$ and the threshold to 600 in our experiments.

Speedup. Our overall method will be expensive, if the base segmentations are expensive to compute. To remove this bottleneck we use fast minimum variance quantization to produce the base segmentations. The RGB color space is split into a user specified number of levels such that each level minimizes the variance of its constituent pixel values. Thus the obtained segmentation accounts for both color and textural variation in a naive fashion. We further find that not imposing the smoothness constraints further improves performance. This is primarily because computing the constraint matrix Θ proves to be expensive. In our experiments we use an ensemble of 4 base segmentations obtained by using two different color levels $\{6, 10\}$ at different degrees of smoothing.

4. Experiments and Results

We compare consensus segmentation against two other widely used segmentation algorithms, the efficient graph based segmentation (GBIS) algorithm [5] and the mean shift algorithm [4]. GBIS treats the image as a graph. Segmentation is achieved by splitting the graph into a collection of connected components. Two connected components are merged when the weight of the edge connecting the two components is less than the maximum weight in either components' minimum spanning tree, plus some constant user controlled parameter M. The publicly available implementation [5] of this algorithm has two other tunable parameters, σ a smoothing parameter and min the minimum number of pixels in segment. In this paper we fix $\sigma = 0.8$ and vary M through {100,200,300,400,500} and min through {20,50,100,300,500}.

The Mean Shift (MS) algorithm involves a mean shift filtering of the image data followed by a clustering of the filtered data. The mean shift filtering is a search for modes of the underlying pdf of the image data. In this paper we have used the open source EDISON [4] implementation of the mean shift segmentation. The EDISON system converts the original RGB image into the LUV space. The mean shift filtering is carried out in a 5 dimensional feature space, containing the (x, y) image coordinates and the LUV values. The algorithm has three tunable parameters spatial bandwidth (h_s), color bandwidth (h_r) and min. We, following popular trend [24] set $h_s = 7$ and vary h_r through $\{3, 5, 7, 9, 11, 13, 15\}$ and the range of min is chosen to be the same as GBIS.

Our algorithms *consensus* and sped-up consensus *fastCon* have two tunable parameters. We apply a Gaussian filter on our images, and the standard deviation of the filter σ is the first parameter, while the number of color clusters C present in an image is the second parameter. σ is varied through {0.75,1.0,1.75} for *consensus* and through {2,4} for *fastCon*, while C varies from image to image. In our experiments C took an integer value between 3 and 12 depending on the image.

For quantifying the performance of the segmentation algorithms, we use the Probabilistic Rand Index (PRI) proposed in [23]. We compare an image segmentation S^{test} with a set of "ground truth" human segmentations $\{\mathcal{H}^1, \mathcal{H}^2, ..., \mathcal{H}^{\mathbb{H}}\}$. The human segmentations are obtained from the Berkeley Image Segmentation Dataset [14], which contains a test set of 100 images.

For completeness we briefly describe the computation of PRI. A segmentation is considered "good" if it agrees with the human segmentations provided. The PRI score increases if the labels l_i and l_j of two pixels i and j are the same, i.e. if they are classified in the same segment of S^{test}, and they are also classified in the same segment for a human segmentation $\mathcal{H}^h$. The score is hurt if this is not the case. Formally, PRI is computed as:

$$\begin{aligned}\text{PRI}(S^{test}, \{\mathcal{H}^1, ..., \mathcal{H}^{\mathbb{H}}\}) \quad = \quad & \frac{1}{\binom{N}{2}} \sum_{i<j} [\mathcal{I}(l_i^{S^{test}} = l_j^{S^{test}}) p_{ij} \\ & + \mathcal{I}(l_i^{S^{test}} \neq l_j^{S^{test}})(1 - p_{ij})]\end{aligned} \tag{19}$$

Where, $\mathcal{I}$ is the identity function and

$$p_{ij} = \frac{1}{\mathbb{H}} \sum_{i<j} [\mathcal{I}(l_i^{S_k} = l_j^{S_k})] \tag{20}$$

PRI takes values in the range $[0, 1]$, with a value of 0 resulting when S^{test} and $\{\mathcal{H}^1, ..., \mathcal{H}^{\mathbb{H}}\}$ have no similarities, and a value of 1 when S^{test} matches $\{\mathcal{H}^1, ..., \mathcal{H}^{\mathbb{H}}\}$ exactly. The PRI values obtained for the three segmentation algorithms as well as the best $k - means$ base segmentation are presented in Table 1. Figure 3 displays segmentations of a representative subset of the Berkeley test images, for visual comparison. The consensus algorithm significantly outperforms the best base segmentation,thus providing a quantitative measure of utility of the consensus process. Consensus also performs comparably with both MS and GBIS. Predictably (Figure 3), Consensus performs well when regions of the image can be distinguished on the basis of color, texture or some combination of the two. The sped up version *fastCon* produces cheap and competitive but somewhat worse segmentations. On a Intel core 2 duo machine with 4GB of RAM *fastCon* required 0.5 ± 0.015 seconds to execute. The consensus took 0.2365 ± 0.02 seconds while

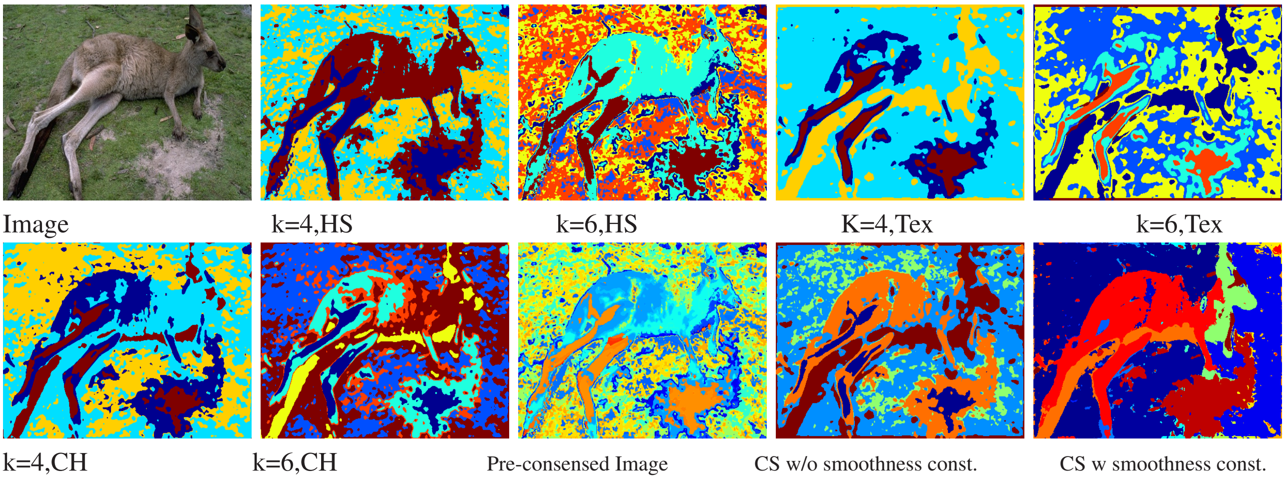

Image k=4,HS k=6,HS K=4,Tex k=6,Tex

k=4,CH k=6,CH Pre-consensed Image CS w/o smoothness const. CS w smoothness const.

Figure 2. An image from the Berkeley database, base segmentations and the resulting Consensus Segmentations(CS)

Segmentation	Mean PRI	Median PRI	St. Deviation
Kmeans	0.6924	0.7094	0.11
MS	0.7627	0.8044	0.14
GBIS	0.7759	0.8024	0.13
Consensus	0.7806	0.7986	0.11
fastCon	0.7688	0.7964	0.11

Table 1. Comparison on the Berkeley test images.

creating the base segmentations took 0.2671 ± 0.002 seconds. The algorithm was implemented using non-optimized Matlab code.

Finally, Figure 4 displays examples where our algorithms do not perform well. These usually correspond with images with homogeneous color and texture patterns. In such cases Consensus finds a large number of spurious segments while merging other semantically distinct segments together, producing noisy segmentations.

5. Conclusion and Future Work

This paper presents a penalized non negative matrix factorization framework for combining multiple image segmentations. Our framework is agnostic of the underlying algorithms and features used to create the base segmentations. It is thus able to seamlessly combine segmentations derived using different modalities. A method to scale the framework for large images is also presented. We find that the performance of an ensemble of *k-means* segmentations compares favorably against popular state of the art "bottom-up" segmentation algorithms. Furthermore, we find that the combination of rudimentary quantization based segmentation performs just as well as more sophisticated segmentation schemes. These observations suggest that producing better semantic segmentations would require "top down" information. This, in turn, necessitates the modeling of uncertainty in both the objects present in an image and their sizes, before a truly general semantic segmentation(of the image) can be produced. These considerations would guide our future work. We also plan to incorporate prior information to bias the ensemble of base segmentations, such that more favorable base segmentations receive higher weights. This would lead to task specific image segmentation as opposed to the general purpose segmentation framework proposed in this paper.

References

[1] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using em and its application to content-based image retrieval. In *Proc. Intl. Conf. on Computer Vision (ICCV'98)*, 1998.

[2] K. Cho and P. Meer. Image segmentation from consensus information. *COMPUTER VISION AND IMAGE UNDERSTANDING*, 68(1):72–89, 1997.

[3] C.-C. Chu and J. K. Aggarwal. The integration of image segmentation maps using region and edge information. *IEEE TRANS. ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 15(12), 1993.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

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

[6] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nystrom method. *IEEE TRANS. ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 26(2), 2004.

[7] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *ACM Trans. Knowl. Discov. Data*, 1(1):4, 2007.

[8] D. HOIEM, A. A. EFROS, and M. HEBERT. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.

[9] D. Kim, S. M. Oh, and J. M. Rehg. Traversability classification for ugv navigation: A comparison of patch and superpixel representations. In *Intl. Conf. on Intelligent Robots and Systems (IROS'07)*, 2007.

[10] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

[11] T. Li, C. Ding, and M. I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 577–582, Washington, DC, USA, 2007. IEEE Computer Society.

[12] T. Li, M. Ogihara, and S. Ma. On combining multiple clusterings. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 294–303, New York, NY, USA, 2004. ACM.

[13] T. Malisiewicz and A. A. Efros. Improving spatial support for objects via multiple segmentations. In *Proc. British Machine Vision Conference (BMVC'07)*, 2007.

[14] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[15] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE TRANS. ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 26(1), 2004.

[16] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Proc. Intl. Conf. on Computer Vision*, 2007.

[17] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. Intl. Conf. on Computer Vision (ICCV03)*, 2003.

[18] O. Rotem, H. Greenspan, and J. Goldberger. Combining region and edge cues for image segmentation in a probabilistic gaussian mixture framework. In *CVPR'07: IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[19] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006.

[20] E. Sharon, A. Brandt, and R. Basriy. Fast multiscale image segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'00)*, 2000.

[21] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TRANS. ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 22(8), 2000.

[22] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research (JMLR)*, 3:583–617, December 2002.

[23] R. Unnikrishnan and M. Herbert. Measures of similarity. In *Seventh IEEE Workshop on Applications of Computer Vision (WACV)*, pages 394–400, 2005.

[24] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):929–944, June 2007.

[25] F. Wang, T. Li, and C. Zhang. Semi-supervised clustering via matrix factorization. In *SDM*, pages 1–12, 2008.

[26] X. Zhang, L. Jiao, F. Liu, L. Bo, and M. Gong. Spectral clustering ensemble applied to sar image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 46(7):2126–2136, 2008.

[27] C. L. ZITNICK and S. B. KANG. Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision*, 75(1):49–65, 2007.

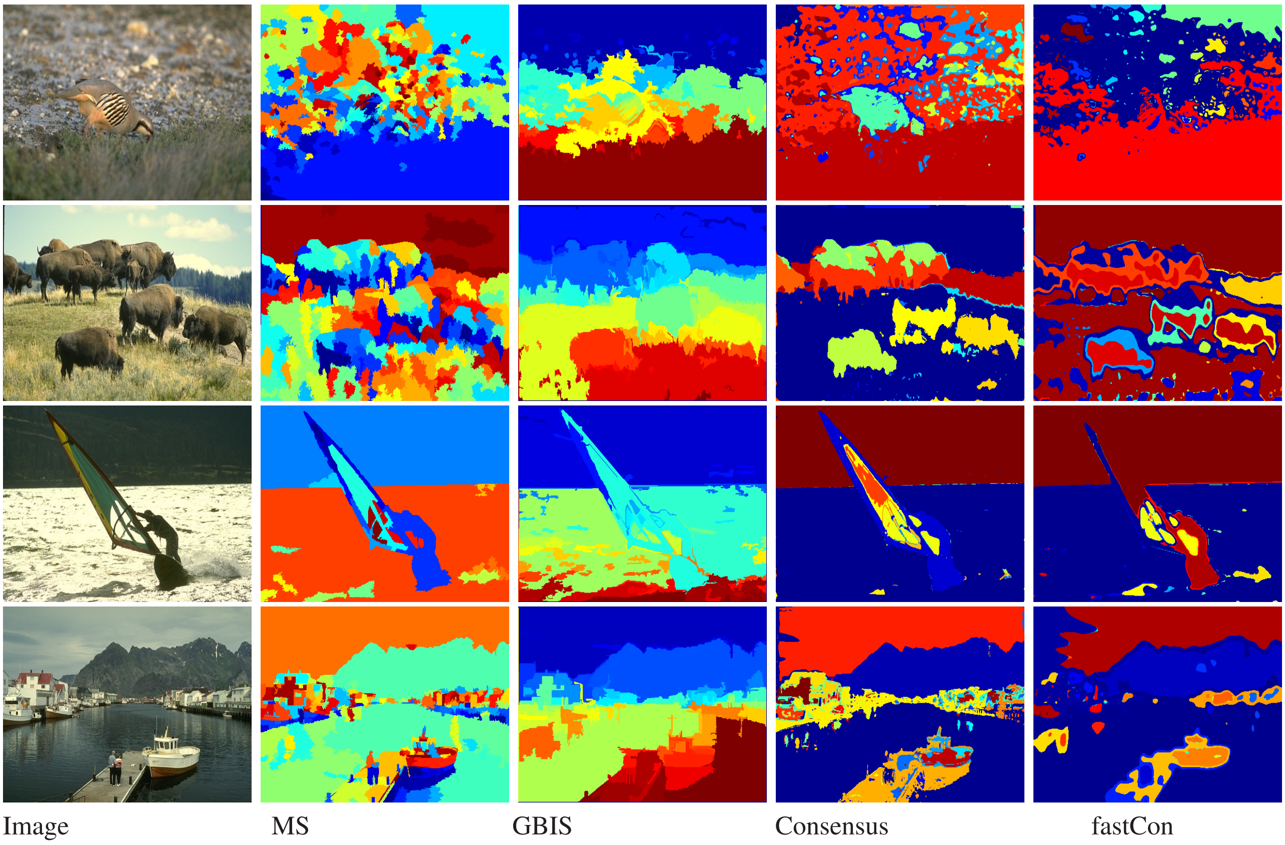

Figure 3. Images from the Berkeley test set and their segmentations

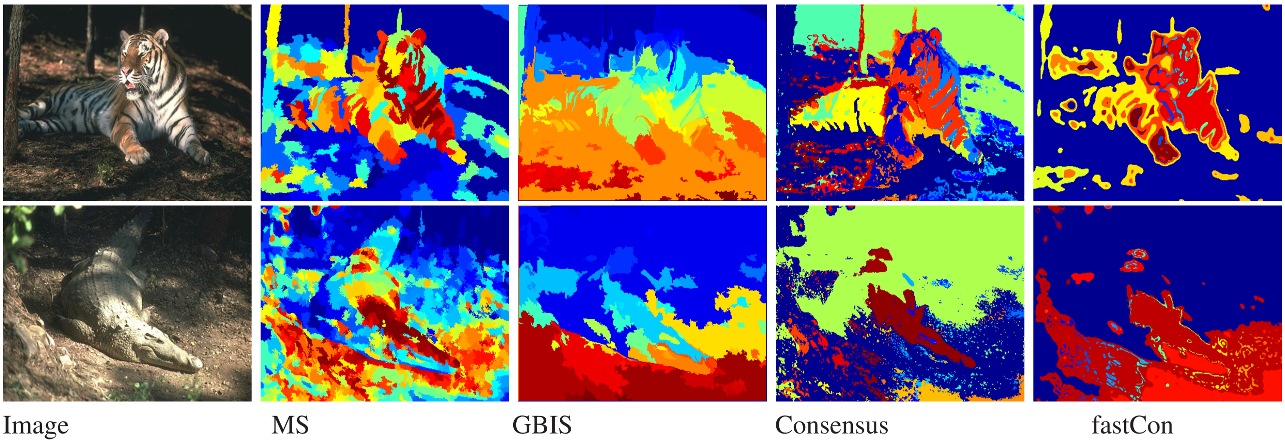

Figure 4. Poorer Quality Segmentations.