

Spatially Coherent Clustering Using Graph Cuts

Ramin Zabih
Cornell University
Ithaca, NY 14853

Vladimir Kolmogorov
Microsoft Research
Cambridge, UK

Abstract

Feature space clustering is a popular approach to image segmentation, in which a feature vector of local properties (such as intensity, texture or motion) is computed at each pixel. The feature space is then clustered, and each pixel is labeled with the cluster that contains its feature vector. A major limitation of this approach is that feature space clusters generally lack spatial coherence (i.e., they do not correspond to a compact grouping of pixels). In this paper, we propose a segmentation algorithm that operates simultaneously in feature space and in image space. We define an energy function over both a set of clusters and a labeling of pixels with clusters. In our framework, a pixel is labeled with a single cluster (rather than, for example, a distribution over clusters). Our energy function penalizes clusters that are a poor fit to the data in feature space, and also penalizes clusters whose pixels lack spatial coherence. The energy function can be efficiently minimized using graph cuts. Our algorithm can incorporate both parametric and non-parametric clustering methods. It can be applied to many optimization-based clustering methods, including k -means and k -medians, and can handle models which are very close in feature space. Preliminary results are presented on segmenting real and synthetic images, using both parametric and non-parametric clustering.

1. Feature-Space Analysis

Many problems in computer vision require estimating local properties at each pixel, such as texture or motion, and then using these properties for segmentation. To accurately compute these local properties, of course, would require knowing the segmentation ahead of time. A popular way to overcome this difficulty is to use feature-space analysis [4, 19, 5, 6, 20, 17, 27]. In this technique, a vector of local properties (“features”) is computed at each pixel without a segmentation, and then mapped into the feature space. Significant features will be shared by numerous pixels, and thus form a dense region in feature space. These dense regions

can be detected by one of the numerous available clustering techniques (see section 8 of [11] for a recent review of clustering methods in vision). Clusters in feature space can then be used for image segmentation, typically by fitting a parametric model to each cluster and then labeling the pixels whose feature vectors lie in the cluster with the parameters.

We can formalize the feature-space analysis approach as follows. The input image will have a set of pixels \mathcal{P} . The feature space \mathcal{X} has d dimensions. For each pixel $p \in \mathcal{P}$ there is a corresponding feature vector $x_p \in \mathcal{X}$. The feature space will be summarized by a set of K models. Each model k describes a cluster in \mathcal{X} , and has some parameters θ_k . The entire set of clusters can be described by the parameter set $\theta = \{\theta_1, \dots, \theta_K\}$. The desired output will consist of the parameter set θ , plus a labeling f that assigns each pixel p to a model $f_p \in \{1, \dots, K\}$. A cluster can also be viewed as a subset of the pixels, where $P_k(f) = \{p | f_p = k\}$, or as a set of points in feature space $X_k(f) = \{x_p | p \in P_k(f)\}$. Note that in this formulation we have made a few simplifying assumptions.¹ These assumptions are purely to simplify the notation, and are not required for our method.

As an example, consider a very simple color image segmentation algorithm. The feature space would have 3 dimensions (one per color channel). An even simpler image segmentation algorithm would use grayscale intensity as its single feature, and compute thresholds between the different clusters. Each model might be a gaussian, parameterized by its mean and variance $\theta_k = (\mu_k, \sigma_k)$. The pixel p will be labeled with the model f_p that has the highest density at the point x_p .

1.1. Spatial coherence

The natural approach to these vision problems is to do feature space clustering and pixel labeling as two separate phases. However, a good cluster in feature

¹We assume a single input image, even though there can be more than one (for example, in motion segmentation). We also assume that each pixel has a different feature vector, since otherwise $X_k(f)$ is a bag rather than a set. Our notation also implicitly assumes that the models are parameterized.

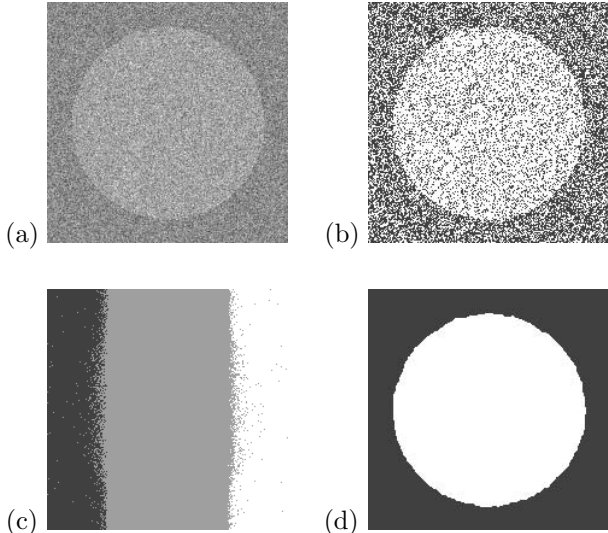


Figure 1: Example of segmentation in feature space (see text for details). (a) Input image. (b) Segmentation from a mixture of gaussians, without position (c) Same as (b) with position (d) Segmentation from spatially coherent clustering.

space will often not be coherent in image space. Similarly, the correct segmentation in image space may not correspond to a highly distinctive group of feature vectors. The feature space clustering algorithm could miss potential clusters which are less distinctive in feature space, but which give rise to spatially coherent segmentations.

To see the importance of spatial coherence, consider the synthetic image shown in figure 1(a). Clearly, it should be possible to segment this image into two regions using intensity as a feature. However, the gaussians that were used to generate the intensity distributions in the different regions are quite close to each other. As a consequence, clustering the intensity space gives poor results. Clustering the feature space into two gaussians results in extremely poor spatial coherence, as shown in the binary segmentation in figure 1(b).

A great deal of work has been done on feature-based clustering (see [11] for a recent review, and [6] for one recent clustering algorithm). However, surprisingly little work has gone into how to make the resulting clusters spatially coherent. (An exception is the work of [18, 19] which we will discuss in section 4.) The most common approach is to use the spatial coordinates of a pixel as additional features. However, this does not overcome the basic difficulties with performing clustering and labeling as separate and independent phases.

In fact, using position as a feature tends to result in

poor segmentations, especially when (as in figure 1) the desired regions do not form simple regions with similar coordinates. Large regions tend to be divided in somewhat arbitrary appearing ways, and there are often multiple tiny regions. This in turn has led researchers to use post-processing to correct for errors in the original segmentation. This post-processing cleanup is usually done by morphological operators [4, 5]. Obviously, it would be preferable to have the segmentation algorithm not produce these kinds of errors in the first place.

As a recent paper [4] notes:

“[L]arge, uniform background areas in the image are sometimes arbitrarily split into two pieces due to the use of position as a feature. On the whole, however, including position yields better segmentation results than excluding it.”

For example, on the synthetic image in figure 1, clustering in feature space gives much worse results if position is included as a feature. The input image in figure 1(a) is a mixture of two gaussians, and if feature space is clustered into two gaussians the resulting segmentation is shown in figure 1(b). We then added position as a feature, without enforcing any relationship between the different components of the feature vectors and allowing the covariance matrix to determine the relative importance of these features automatically. Adding position and clustering into two gaussians gives an essentially identical output to figure 1(b) (this is not shown due to space constraints). Adding position and clustering feature space into three gaussians gives the spatially coherent yet completely wrong segmentation shown in figure 1(c). Our algorithm produces the more accurate segmentation shown in figure 1(d). The experimental setup used to generate these results is described in more detail in section 5.

1.2. Summary of our results

This paper is organized as follows. We begin by presenting an energy minimization formulation, where the energy function enforces both clustering quality in feature space and spatial coherence in image space. The energy function can be minimized in an Expectation-Maximization [7] style using graph cuts, as long as the clustering quality measure obeys a linearity criterion defined in section 2.1. Fortunately, we can easily show that many common clustering methods, including k-means, obey this criterion. In section 3 we describe two clustering methods that we have used, one parametric and one non-parametric. The parametric clustering method obeys the linearity criterion, while the

non-parametric method obeys it under certain assumptions, which we have empirically validated. In section 4 we give a brief description of some related work, and in section 5 we demonstrate how our method performs segmentation on grayscale and color images.

2. Energy Minimization

The energy function that we will minimize has terms that embody the following constraints on f and θ .

- **Cluster quality in feature space:** The clusters should be good according to some quality measure Q . For example, they should have low intra-cluster variation and/or high inter-cluster variation.
- **Spatial coherence in image space:** The labeling f should be spatially coherent.

The energy function that we will minimize is

$$E(f, \theta) = Q(f, \theta) + \sum_{p, q \in \mathcal{N}} V(f_p, f_q). \quad (1)$$

Here, Q measures the (lack of) quality of a particular clustering of the feature space. To incorporate spatial coherence, we have a set \mathcal{N} of pairs of neighboring pixels, and impose a penalty specified by V for neighboring pixels to have different labels. V will be some kind of robust metric in the space of clusters. Note that if we ignore spatial coherence, we are simply performing clustering in feature space according to some objective function specified by Q .

We minimize the energy function $E(f, \theta)$ iteratively, in a manner that is similar to EM [7]. The first step (which resembles the ‘‘E’’ step in EM) is to fix the cluster parameters θ and find the best labeling f . The second step (the ‘‘M’’ step) is to fix the labeling f and find the best clusters θ . The energy minimization in the E step is done using graph cuts, which is a technique that has proven to be quite powerful, especially in stereo. For example, according to the Middlebury stereo database [22] the top 2 stereo algorithms, as well as 7 of the top 10 algorithms, are based on graph cuts. As we will see, we can use our method as long as the clustering objective function Q meets a natural linearity criterion. Fortunately, many popular objective functions meet this criterion.

2.1. Fixing the clusters (E step)

In this step we seek the best labeling given a fixed set of clusters. To solve this efficiently, we will take advantage of the fact that there are fast algorithms based on graph cuts that solve a closely related problem

[3]. In particular, it is possible to efficiently minimize an energy function over pixel labelings f of the form

$$\hat{E}(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{p, q \in \mathcal{N}} V(f_p, f_q) \quad (2)$$

under fairly broad constraints on the form of D and V (see [14] for details). Here, $D_p(f_p)$ is the penalty for assigning the label f_p to the pixel p .

The most effective graph cut algorithm for minimizing \hat{E} is the expansion move algorithm of [3]. For a label k , we will define $f_p^{\mathcal{S} \leftarrow k}$, for an arbitrary set $\mathcal{S} \subset \mathcal{P}$, by $f_p^{\mathcal{S} \leftarrow k} = k$ if $p \in \mathcal{S}$ and otherwise $f_p^{\mathcal{S} \leftarrow k} = f_p$. Thus, when we go from f to $f^{\mathcal{S} \leftarrow k}$ the pixels in the set \mathcal{S} acquire the label k and all other pixels remain the same. We will say that $f^{\mathcal{S} \leftarrow k}$ is a k -expansion move from f .

Obviously the number of possible expansion moves is exponential in $|\mathcal{S}|$, and thus in the number of pixels. However, [3] shows that graph cuts can be used to efficiently find the best k -expansion move, i.e. for a given k the subset $\mathcal{S} \subset \mathcal{P}$ that minimizes $\hat{E}(f^{\mathcal{S} \leftarrow k})$. The expansion move algorithm of [3] works iteratively: given an input labeling it picks a label k and uses graph cuts to find the best k -expansion move, and moves to that labeling if it decreases the energy. The expansion move algorithm is actually an approximation algorithm, in that it guarantees that the solution lies within a constant factor of the global minimum. The constant factor is at least 2, and depends on V (see [3] for details).

Our task is to minimize $E(f, \theta)$ for a fixed set of clusters θ . Obviously, this is nearly in the same form as equation 2. All that is required is that for any fixed θ there must exist some function D_p such that

$$Q(f) = \sum_{p \in \mathcal{P}} D_p(f_p). \quad (3)$$

This is our **linearity criterion**, and will refer to a clustering objective function Q as linear if it meets this criterion. If Q is linear then we can minimize the energy function in the E step by using the expansion move algorithm.

2.1.1 What clustering methods are linear?

Fortunately, it is easy to prove that a large class of optimization based clustering methods meet our linearity criterion, and hence that our method can be applied. In particular, many clustering methods represent clusters by a single point (the cluster centers c_k). The objective function is

$$Q(f, \theta) = \sum_k \sum_{x \in X_k(f)} \rho(c_k, x). \quad (4)$$

Here ρ is some metric.

Obviously this class of clustering methods meets our linearity criterion, with $D_\rho(f_p)$ being the distance under ρ between the cluster center for cluster f_p and the feature vector x_p . When ρ is the L_2 distance, this method is the well-known k-means clustering method; when ρ is the L_1 distance, this method is k-medians. As a result, our algorithm can be applied with both these clustering methods.

2.2. Fixing the labeling (M step)

In the second step in our method, we fix the labeling f and optimize for the cluster parameters θ . The details of how this is done depend upon the way that clusters are represented. With parametric models, this is fairly simple. For example, if the models are gaussians then this step simply consists of computing the model parameters for each cluster. Since this step is so fast, it makes sense to do this after every expansion move in the E step. If non-parametric clustering is used, this step is not necessary (since by definition there are no parameters to be estimated).

3. Clustering Methods

In this section we discuss the two different clustering functions that we have used with our method. The clustering functions that we consider are based upon some objective function $Q(f, \theta)$ that determines the (lack of) quality of the clustering f given the set of model parameters θ . We will also consider a non-parametric clustering method, where (by definition) there are no parameters θ . As usual, clustering methods are easiest to describe in probabilistic terminology.

3.1. Parametric clustering method

We now describe a particular parametric clustering method that we have explored. The clustering method that we use is based on an objective function Q . We use the same method as in [18, 29].

We will derive our choice of Q on probabilistic grounds within the framework of Markov Random Fields [15]. We wish to maximize the posterior probability

$$Pr(f, \theta|x) \sim Pr(x|f, \theta)Pr(f)$$

(we assumed that f and θ are independent and that θ has a uniform prior). The likelihood term and prior on f are given by $Pr(x|f, \theta) = \prod_{p \in \mathcal{P}} Pr(x_p|\theta_{f_p})$ and $Pr(f) \sim \exp(-\sum_{p, q \in \mathcal{N}} V(f_p, f_q))$.

By taking the negative logarithm we obtain an energy function of the form in equation (1) with the clus-

tering objective function given by

$$Q(f, \theta) = \sum_{p \in \mathcal{P}} -\log Pr(x_p|\theta_{f_p}). \quad (5)$$

Obviously, this choice of Q meets our linearity criterion. In our experiments we used gaussian distributions parametrized by mean μ_k and covariance Σ_k .

It is well-known that this model can be viewed as a generalization of the finite mixture model (FMM) (see [29], for example). Let us briefly review a relationship between them. In FMM the labels f_p are assumed to be independent for different pixels p . In this case the prior on f can be written as $Pr(f) = \prod_{p \in \mathcal{P}} \pi_{f_p}$, where π_k is the probability of label k . The posterior becomes $Pr(f, \theta|x) \sim \prod_{p \in \mathcal{P}} \pi_{f_p} Pr(x_p|\theta_{f_p})$. By marginalizing over labelings we get

$$\begin{aligned} Pr(\theta|x) &= \sum_f Pr(f, \theta|x) \sim \sum_f \prod_{p \in \mathcal{P}} \pi_{f_p} Pr(x_p|\theta_{f_p}) \\ &= \prod_{p \in \mathcal{P}} \sum_{k=1}^K \pi_k Pr(x_p|\theta_k) \end{aligned} \quad (6)$$

This distribution is called a mixture of models $Pr(x|\theta_k)$ with mixing parameters π_k .

An advantage of FMM is that it takes care of uncertainty in f by marginalizing over it. However, it lacks spatial coherence, which is an important clue for deciding what label a pixel has.

3.2. Non-parametric clustering method

Our non-parametric clustering method works by representing a cluster k as a collection of points in the feature space X_k . Let $Pr(x|X_k)$ be an estimate of the distribution in this cluster. (For example, we compute the histogram of X_k and convolve it with a kernel). As in equation (5) we can write

$$Q(f, \theta) = \sum_{p \in \mathcal{P}} -\log Pr(x_p|X_{f_p}(f)) \quad (7)$$

(recall that $X_k(f)$ is the set of features of pixels with the label k according to labeling f).

This clustering method does not actually meet our linearity criterion in the general case. However, when the proposed α -expansion does not change the cluster properties (i.e. the histogram in the feature space) it is linear. This allows us to use our proposed technique, but after each α -expansion we check whether the proposed step decreases the energy function, and reject the change if it does not. As we found experimentally, most α -expansions do decrease the energy, which validates the approximation. This is similar to the work of [13].

4. Related Work

Feature-space analysis has been widely used in computer vision for segmentation, especially when the segmentation is based on motion or texture. For example, the original paper on layered motion estimation [26] used local affine motion parameters as features and then clustered them using k-means. Much of the large amount of work on layers that followed also made use of feature-space clustering, either explicitly or implicitly.

As another example, a number of graph-based segmentation schemes such as [9, 10, 24, 28] rely on an affinity matrix which represents the similarity between a pair of pixels. However, unless the segmentation is purely based on intensity, they need to base affinity on some kind of aggregate property, such as texture. This is why, for instance, the Berkeley group’s work on image segmentation [17, 4] does not use normalized cuts directly on the image, but first performs clustering in feature-space to generate the affinity matrix.

4.1. Incorporating spatial coherence into feature-space analysis

Since the use of positional information as a feature is known to be problematic, several groups have explored alternatives. One possibility is to perform a fairly fine-grained segmentation at the very beginning, and then compute feature vectors from these regions rather than from pixels. This approach is sometimes called “superpixels” [21], and was used in some work about a decade ago in motion segmentation [8, 12], as well as in more recent work [27]. Superpixels will result in a segmentation that does not cross superpixel boundaries. This, obviously, is problematic if the superpixels themselves cross the boundaries of objects. Moreover, the same difficulties with doing clustering and segmentation in two different passes still apply, since a good cluster in feature space will not necessarily correspond to a connected set of superpixels. Another approach, which has similar motivations to our work, is Matas and Kittler’s [19] FSD algorithm. This method cannot handle adjacent regions whose clusters are too close in feature space. As a result, FSD cannot handle most of the example images that we use in this paper to demonstrate our results. Finally, several researchers in the machine learning community have explored EM-style methods for clustering with additional constraints [23, 25]. However, these methods are not powerful enough to impose spatial coherence.

4.2. Soft membership methods

The most closely related work to ours is a class of methods that uses “soft membership” clustering [11], where

a feature vector is labeled with a probability distribution over clusters instead of a single cluster. A number of techniques for doing spatially coherent soft clustering have been developed in a Bayesian framework. Marroquin *et al.* [18] refer to such methods as Segmentation/Model Estimation Methods.

The first such technique appears to have been developed by Weiss and Adelson [27] as a way of incorporating spatial coherence into motion segmentation in feature space. More recently, Hidden Markov Random Fields (HMRF’s) [29] and Hidden Markov Measure Fields (HMMF’s) [18] have also been proposed.

The critical difference between these methods and our approach lies in two (related) areas: the choice between hard and soft membership; and the optimization method used to solve the pixel labeling problem in the E step. Previous techniques used soft membership, which makes the M step more effective. Unfortunately, computing soft membership in the E step is usually intractable in the MRF framework. Researchers were thus forced to use approximation schemes for MRF’s, such as the mean field approximation [27], Iterated Conditional Modes [29] and Hidden Markov Measure Fields [18]. By using hard membership, our formulation permits the use of the graph cut optimization methods that have proven highly effective for solving pixel labeling problems in stereo.

4.3. Graph cut techniques

Several recent researchers have used graph cuts to label sloped surfaces with affine parameters [1] or with splines [16]. These approaches do not rely on clustering in feature space; instead, they create a point in feature space for each connected component of the labeling, then iterate. They are aided significantly by the use of graph cuts stereo with integer disparities to create a good initial segmentation.

5. Experimental Results

Spatially coherent clustering technique is a general method for finding a spatially coherent clustering in feature space. The output is a set of spatially coherent clusters, which could then be used (for example) to compute the affinity matrix for a graph-based segmentation algorithm such as [9, 24]. It can also be used directly for segmentation. In our preliminary experiments, we did image segmentation with four methods: spatially coherent parametric clustering with gaussians (“Parametric SCC”), spatially coherent non-parametric clustering (“Non-parametric SCC”), finite gaussian mixture (“FGM”) and FGM with spatial coordinates added as features (“FGM with position”).

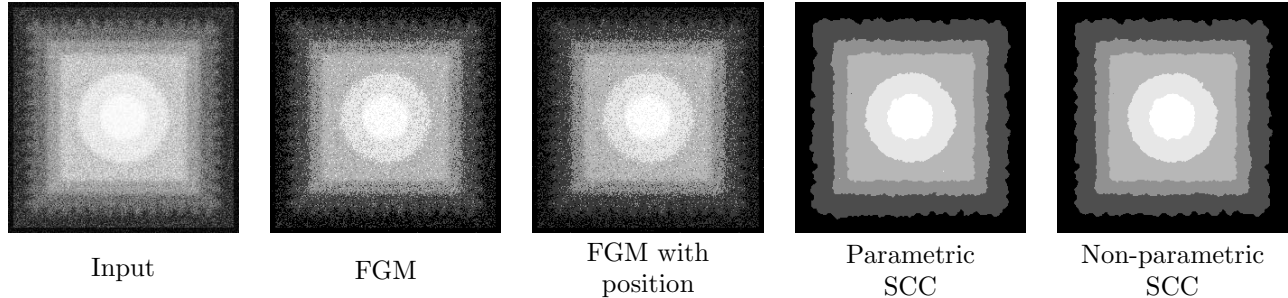


Figure 2: Results on stamp image of [19]

Smoothness term. We used an 8-neighborhood system \mathcal{N} and the Potts smoothness term:

$$V_{p,q}(p, q) = \frac{\lambda}{|e_{p,q}|} \cdot T(f_p \neq f_q),$$

where $|e_{p,q}|$ is the length of edge $e_{p,q}$ and $T(\cdot)$ is 1 if its argument is true and 0 otherwise. The choice of coefficient $\frac{\lambda}{|e_{p,q}|}$ was motivated by [2] which showed that the separation cost becomes approximately the euclidian length of the boundary between regions. In our experiments we always used $\lambda = 1$. For this particular choice of V , the expansion move algorithm is guaranteed to converge to a solution whose energy is no more than twice the global minimum’s.

Initialization. We found that initialization has a significant effect on all four methods. The coarseness of the resulting segmentation strongly depends on the number of clusters that we start with. In one version of our algorithm we implemented cluster merging, assuming that each cluster has a fixed penalty. It is not clear, however, how to set this penalty automatically. Moreover, sometimes a small visually distinctive region would be merged with another cluster while two more similar but large regions would remain unmerged.

The results that we present do not use any merging. For initialization of FGM we divided the image into a small number of rectangles and used them as an initial labeling. This initialization is obviously fairly coarse, and can easily be improved upon. Other algorithms were initialized with the results of FGM.

Convergence. Our algorithms are guaranteed to decrease the energy and thus, theoretically, must converge. For certain sequences they converged for a few iterations of the expansion move algorithm. There were cases, however, when several tens of iterations were required. Yet the changes made after the first ten iterations were always negligible, and the most rapid decrease in energy occurred in the first few iterations.

Synthetic test images. The synthetic image shown in figure 1 was generated as follows. We generated an image with two regions of equal area (a bright

disk in a dark background). Then we added independent gaussian noise to the image with variance equal to the gap between region intensities. We initialized FGM with two gaussians at far ends of the intensity range. Results for FGM with positional features were close to FGM results since the centers found were in the middle of the image. We show results obtained with FGM with positional features and three labels in figure 1(c).

We also experimented with a version of the stamp image used in [19]. Results are shown in figure 2.

Color test image. In figure 3 we show the result of parametric SCC on a color image using 16 labels. The color image has been used for a previous segmentation algorithm [9]. With such a large number of labels, there are a number of small segments in the output. However, the spatial coherence we impose also results in many large regions (such as the truck, the street, and much of the baseball field).

Grayscale test image. We performed grayscale image segmentation using 6 labels on a baseball image that has been used in several algorithms [9, 24].

It is, of course, very difficult to evaluate or compare segmentation algorithms on real images. However, the preliminary results suggest that spatially coherent segmentation obtains smoother outputs than FGM. This is particularly noticeable in the baseball image, where FGM without positional features yields a great deal of noise, while FGM with positional features yields some very puzzling segmentations.

6. Conclusions and Future Work

We have described a new technique to find spatially coherent clusters, and shown that it can be used with a wide variety of clustering techniques. In order to determine the effect of spatially coherent clustering, we will look at several segmentation algorithms that use FGM internally and see how their performance changes when we replace FGM with our technique. This will be particularly interesting for segmentation algorithms that are based on complex local properties such as tex-



Figure 3: Color image segmentation example. The input image is at top, the result of spatially coherent clustering is at bottom.

ture or motion. We also plan to investigate ways to automatically determine the number of clusters in an energy-minimization framework.

Acknowledgements

We wish to thank Junhwan Kim for initial discussions that led to this work, and the anonymous reviewers for pointing out related work. This work was supported by NSF grant CCR-0113371.

References

- [1] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *International Conference on Computer Vision*, pages 489–495, 1999.
- [2] Yuri Boykov and Vladimir Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *International Conference on Computer Vision*, pages 26–33, 2003.
- [3] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.
- [4] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using Expectation-Maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, August 2002.
- [5] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 750–755, 1997.
- [6] 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.
- [7] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data using the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.
- [8] M. Etoh and Y. Shirai. Segmentation and 2D motion estimation by region fragments. In *International Conference on Computer Vision*, pages 192–199, 1993.
- [9] Pedro Felzenszwalb and Daniel Huttenlocher. Image segmentation using local variation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–104, 1998.
- [10] Y. Gdalyahu, D. Weinshall, and M. Werman. Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1053–1074, October 2001.
- [11] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.
- [12] A.D. Jepson and M.J. Black. Mixture models for optical flow computation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 760–761, 1993.
- [13] Junhwan Kim, Vladimir Kolmogorov, and Ramin Zabih. Visual correspondence using energy minimization and mutual information. In *International Conference on Computer Vision*, pages 1033–1040, 2003.
- [14] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, February 2004. An earlier version of this paper appeared in *ECCV*.
- [15] S. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, 1995.

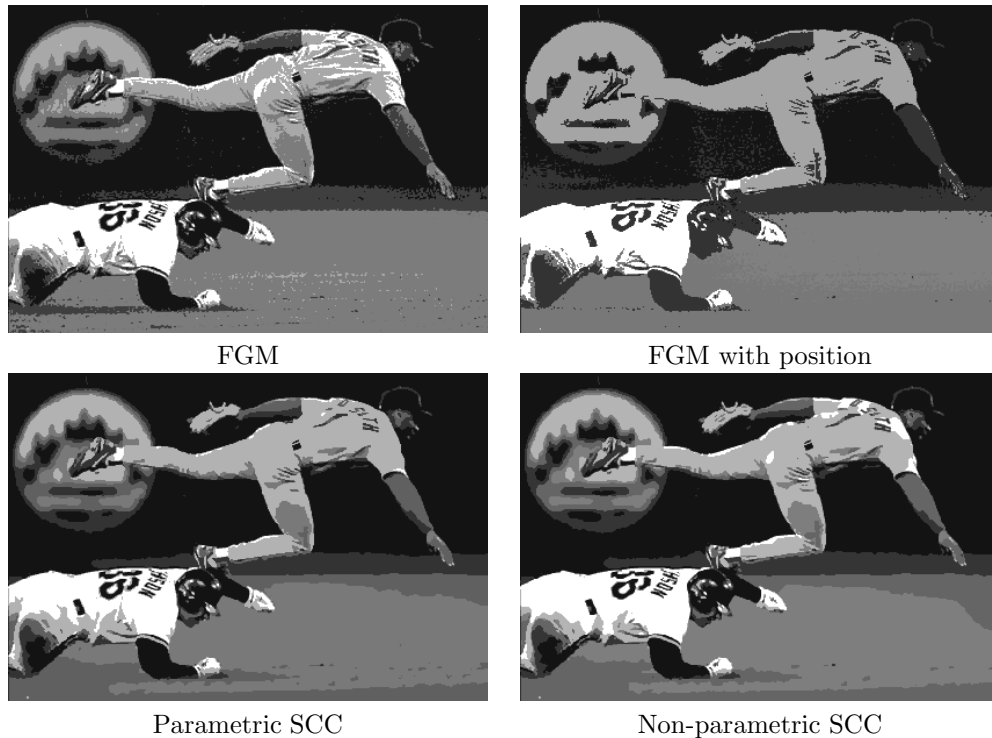


Figure 4: Segmentation results on the baseball image

- [16] M.H. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages I: 710–717, 2003.
- [17] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, June 2001.
- [18] J. Marroquin, E. Santana, and S. Botello. Hidden Markov measure field models for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1380–1387, November 2003.
- [19] Jiri Matas and Josef Kittler. Spatial and feature space clustering: Applications in image analysis. In *Computer Analysis of Images and Patterns*, pages 162–173, 1995.
- [20] David Murray and Bernard Buxton. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2), 1987.
- [21] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *International Conference on Computer Vision*, 2003.
- [22] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, April 2002.
- [23] Noam Shental, Aharon Bar-Hillel, Tomer Hertz, and Daphna Weinshall. Computing gaussian mixture models with EM using side-information. Technical Report 43, Hebrew University of Jerusalem, 2003.
- [24] J. Shi and J. Malik. Normalized cuts and image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–737, 1997.
- [25] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning*, pages 577–584, 2001.
- [26] John Wang and Edward Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, September 1994.
- [27] Y. Weiss and E.H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 321–326, 1996.
- [28] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, November 1993.
- [29] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain MR images through a hidden Markov random field model and the Expectation-Maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1):45–57, January 2001.