To Appear: International Conference on Pattern Recogntion, August 2004.

Image Segmentation Through Energy Minimization Based Subspace Fusion

Jason J. Corso, Maneesh Dewan, and Gregory D. Hager
Computational Interaction and Robotics Lab
The Johns Hopkins University
Baltimore, MD 21218
{jcorso,maneesh,hager}@cs.jhu.edu

Abstract

In this paper we present an image segmentation technique that fuses contributions from multiple feature subspaces using an energy minimization approach. For each subspace, we compute a per-pixel quality measure and perform a partitioning through the standard normalized cut algorithm [12]. To fuse the subspaces into a final segmentation, we compute a subspace label for every pixel. The labeling is computed through the graph-cut energy minimization framework proposed by [3]. Finally, we combine the initial subspace segmentation with the subspace labels obtained from the energy minimization to yield the final segmentation. We have implemented the algorithm and provide results for both synthetic and real images.

1. Introduction

Image segmentation in a particular feature subspace is a fairly well understood problem [13]. However, it is well known that operating in only a single feature subspace, e.g. color, texture, etc, seldom yields a *good* segmentation for real images. In the simple case given in Figure 2, segmenting in either the color or the texture subspace does not yield all of the regions. However, combining information from multiple subspaces in an optimal manner is a difficult problem to solve algorithmically. Although there is recent interest in combining two subspaces of information, primarily color and texture [1, 4], we are unaware of any general technique for combining multiple subspaces.

The obvious approach one might take to combine information from multiple subspaces is to construct a large feature vector with components from each of the subspaces, and then employ a standard segmentation approach. However, it is not clear that such an approach makes good use of each subspace, and the *curse of dimensionality* may render the problem intractable.

In this paper we propose a method that fuses separate segmentations from multiple subspaces into one final segmentation. The fusion is performed in the graph-cuts based energy minimization framework [3, 10]; such an energy minimization approach attempts to capture the global structure of an image where local techniques would fail.

1.1. Related Work

The literature on image segmentation is dense. Here, we sample relevant papers. Jain et al. [8] provide a complete survey of data clustering techniques. Our approach uses the normalized cuts [12] image segmentation in each subspace independently. This approach is one of a class of techniques characterized by Weiss [13] as an eigendecomposition of the affinity matrix.[1] The normalized cuts technique is also used in [11] where the authors provide a technique for combining texture information and brightness information for image segmentation.

The strength of energy minimization techniques lie in their ability to model global image structure on which local methods would normally fail. However, before the recent work by Boykov et al. [3, 2], energy minimization algorithms were typically solved through simulated annealing methods which rendered them computationally intractable. Their framework provides an approximate solution to the minimization in polynomial time. While not all energies can be minimized using graph cuts, Kolmogorov and Zabih [10] define the class of energy functions which are graph representable and thus can be minimized using graph cuts. The graph cut minimization techniques have been successfully applied to image restoration [3], stereo [9], and joint stereo-background separation [7].

2. Approach

We propose a technique that computes segmentations in multiple separate subspaces and then fuses them in a man-

[1] The affinity matrix is a large matrix containing information about inter-pixel affinity (similarity or dissimilarity).

ner that attempts to capture the global image structure. The thrust of our work is to select an appropriate subspace for regions in the image based on the underlying image structure in that region. We propose a general energy minimization algorithm that computes this selection, or *labeling*.

We first concentrate on our approach for combining the results of segmentation algorithms. To this end, let $\mathcal{S}$ be the set of labels for the considered subspaces and $\mathcal{I}$ be the set of image pixel locations. For every subspace $s \in \mathcal{S}$ we define two functions: the quality function[2] $Q_s : \mathcal{I} \rightarrow [0, 1]$ and the segmentation $C_s : \mathcal{I} \rightarrow \mathbb{Z}_s$ with the restriction that $\mathbb{Z}_{s_1} \cap \mathbb{Z}_{s_2} = \emptyset$. Intuitively, the quality function should measure the expectation that a pixel neighborhood will yield a promising segmentation. $\mathcal{Q}$ is the set of quality functions $\{Q_s, s \in \mathcal{S}\}$ over all subspaces and $\mathcal{C}$ is similarly the set of segmentations $\{C_s, s \in \mathcal{S}\}$ over all subspaces. One can employ a variety of techniques to compute $\mathcal{Q}$ and $\mathcal{C}$; in Section 3 we provide the complete algorithm and examples of techniques to compute $\mathcal{Q}$ and $\mathcal{C}$ in Section 4.1.

2.1. Labeling via Energy Minimization

As discussed in [10], there is a certain class of energy functionals that can be minimized using graph cuts. The energy function we choose belongs to that class of functionals. The input to the energy minimization algorithm is the set of quality measures $\mathcal{Q}$ and segmentations $\mathcal{C}$. The output is to associate a labeling $L : \mathcal{I} \rightarrow \mathcal{S}$ for each pixel $i \in \mathcal{I}$ that is the *best* subspace from the set of subspaces $\mathcal{S}$. We assume that the clustering obtained in each subspace is the optimal one for that subspace. Thus, wherever the quality measure is good for a particular subspace, the clustering for that subspace will also be good. Based on [3], the energy functional that we minimize is

$$E(\mathcal{L}) = \sum_{i \in I} Q_{L(i)}(i) + \sum_{i,j \in N} V_{i,j}(L(i), L(j)) \quad (1)$$

where N is the neighborhood over a set of pixels, $V_{i,j}$ is the discontinuity preserving smoothness or the interaction term between neighboring pixels i and j. It is defined based on the clustering in each of the subspaces. We simplify the notation $L(i)$ to be L_i here.

$$V_{i,j}(L_i, L_j) = k_2 + (k_1 - k_2) \cdot \delta(C_{L_i}(i), C_{L_i}(j)) \cdot \delta(C_{L_j}(i), C_{L_j}(j)) \quad (2)$$

where C_{L_i} denotes the clustering in the subspace with label L at pixel i, $k_2 > k_1$ and δ is the Kronecker delta function.

Intuitively, the interaction term (2) attempts to preserve the boundary at L_i and L_j if either subspace segmentation C_{L_i} or C_{L_j} indicates a boundary. Additionally, it enforces smoothness in the labeling by linking adjacent pixels. We include a detailed discussion of the interaction term in [5].

The energy function we use satisfies the constraint defined in [10]. Due to space limitations, the proof is included in [5]. In order to minimize this energy functional, we create a set of graphs as outlined in [3] without the a-nodes and perform α-expansion[3] at every step to compute the labeling that minimizes the energy.

2.2. Fusion of Subspaces

The final step of our algorithm performs the fusion of the subspace segmentations $\mathcal{C}$ based on the labeling L. Each pixel $i \in \mathcal{I}$ in the final segmentation is assigned to the region based on its labeling $L(i)$ and the subspace segmentation $C_{L(i)}$:

$$C_f(i) = C_{L(i)}(i), \ \forall i \in \mathcal{I} \quad (3)$$

Note that a boundary in L creates a boundary in the final segmentation. Likewise, note that boundaries in the highest quality regions (over which L is constant) are preserved. To reduce noise, we first median filter L.

3. The Complete Algorithm

The complete segmentation algorithm is presented in Figure 1. We initially compute a standard single-subspace segmentation using the normalized cuts algorithm [12] and a pixel-coefficient for each subspace that measures the subspace quality. These two quantities drive the energy minimization which assigns a per-pixel subspace label that is then used in a fusion procedure.

```
SEGMENT ( Input Image I )
foreach (subspace s ∈ S )
    foreach (pixel i ∈ I)
        Compute quality Q_s(i)
    Compute clustering C_s(I)
Compute labeling L(I)
foreach (pixel i ∈ I)
    C_f(i) = C_L(i)(i)
return C_f as final segmentation
```

Figure 1. Proposed segmentation algorithm.

We employ the standard normalized cuts algorithm [12] to compute subspace segmentation ($C_k, \ k \in \mathcal{S}$) and choose

[2]0 being the best quality and 1 the worst quality.

[3]Hence the use of small values for high quality and large values for low quality (Section 4.1 for more information).

to use the dominant 6 eigenvectors of the affinity matrix instead of performing explicit recursion. We follow the same formula to compute the affinity matrix A as [12]; namely, we populate the entries of A as the product of the feature subspace similarity term and the clamped spatial proximity term. Specifically, for pixels i and j let $X_{ij} = \| \mathbf{X}(i) - \mathbf{X}(j) \|_2$ where $X(i)$ is the spatial location of pixel i,

$$A_{ij} = e^{\frac{-\|\mathbf{F}(i)-\mathbf{F}(j)\|_2}{\sigma_F}} * \begin{cases} e^{\frac{-X_{ij}}{\sigma_X}} & \text{if } X_{ij} < r \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $\mathbf{F}(i)$ is the feature vector based on feature subspace (see Section 4.1 for examples of $\mathbf{F}$) and r is the neighborhood truncation threshold. We follow [12] in our choices of parameters, r, σ_F, σ_X: $5, 0.01, 4.0$ respectively.

4. Experiments

In this paper, we consider techniques and results for the color and texture feature subspaces only; however, there is no inherent limitation in our approach limiting it to these two subspaces. We present results for both synthetic and real data.

4.1. Subspace Definitions

In both the color and texture subspaces, we use an intuitive approach to measuring quality (Q_c and Q_t): the variance of the image $\mathcal{I}$. For the color subspace, regions with low color variance are expected to yield good segmentation. We take a conservative measurement and use the color-channel with the worst variance. Let $i \in \mathcal{I}$ be a pixel in the image, $\mathcal{N}_i$ be a neighborhood of pixels about i, $\mu^p(i) = \frac{1}{|\mathcal{N}|}\sum_{x\in N} \mathcal{I}^p(x)$ where $p \in$ r, g, b, i is a plane of the color image or grayscale, and $\sigma^p(i) = \frac{1}{|\mathcal{N}|}\sum_{x\in\mathcal{N}} (\mathcal{I}^p(\mathbf{x}) - \mu^p(i))^2$.

$$\sigma_c(i) = \max[\sigma^{\mathrm{r}}(i),\ \sigma^{\mathrm{g}}(i),\ \sigma^{\mathrm{b}}(i)] \quad (5)$$

$$\sigma_t(i) = \sigma^{\mathrm{i}}(i) \quad (6)$$

By convention (Section 2.1), we define quality 0 as best quality and 1 as worst quality. Let η be a threshold on the max color variance allowed (we use 0.2 in our experiments).

$$Q_c(i) = \begin{cases} 1 & \text{if } \ \sigma_c(i) > \eta \\ \frac{\sigma_c(i)}{\eta} & \text{otherwise} \end{cases} \quad (7)$$

$$Q_t(i) = 1 - \sigma_t(i) \quad (8)$$

For the color and texture subspaces, we define $\mathbf{F}(i)$ as follows:

- **Color** - $\mathbf{F}(i) = [v, v\cdot s\cdot \sin h, v\cdot s\cdot \cos h](i)$, where h, s, v are the HSV values.

- **Texture** - $\mathbf{F}(i) = [|\mathcal{I} * g_1|, \cdots, |\mathcal{I} * g_n|](i)$, where the g_k are the Gabor convolution kernels [6] at various scales and orientations. In our implementation we use 4 scales and 6 orientations.

4.2. Results

Figure 2 shows the result of our algorithm on a synthetic image that contains three regions: one from the texture domain and two from the color domain. It is clearly visible that fusing the subspaces improves the segmentation significantly over either the color or texture subspace alone.

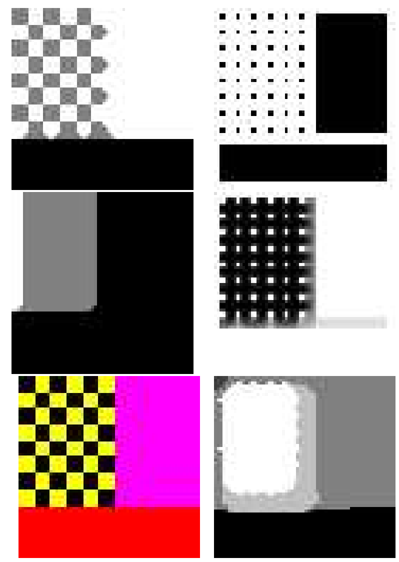

Figure 2. Result of subspace fusion on a synthetic image. First row is the color subspace segmentation and quality. Second row is the texture subspace, and the final row is the original image and the fusion segmentation. Note that quality ranges from 0 (black) to 1 (white) with 0 the best quality and 1 the worst quality.

Figure 3 shows the result of subspace fusion on a real zebra image. The segmentation in both the color and texture subspaces is shown with the resultant segmentation obtained by subspace fusion.

It is important to note that the segmentation in the texture subspace in the zebra image is not very good as the segmentation was run on a very low resolution image (i.e. 100x80) where the texture information is subdued. We expect that on higher resolution images the results would be much better,

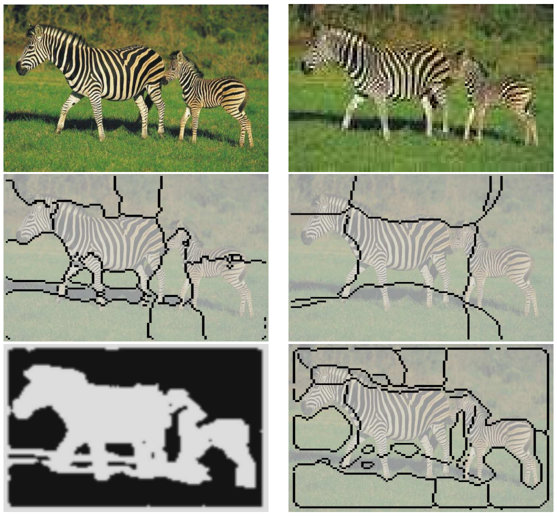

Figure 3. Complete segmentation for the zebra image. Top-left is the high resolution image and top-right is the low-resolution image used. Row 2-left is the color subspace clustering, Row 2-right is the texture subspace, bottom-left is the labeling with black assigned to color, and bottom-right shows the final clustering.

but due to the high computation cost involved with segmenting high resolution images, we currently restrict ourselves to low resolution images.

5. Conclusion and Future Work

In this paper we have presented a technique for image segmentation which employs information from multiple subspaces. We make two contributions: First, we propose a technique to measure subspace quality based on the expectation that a local image neighborhood will yield a good segmentation for a given subspace. Second, we propose a novel use of the graph-cut based energy minimization framework [3] to fuse the subspace segmentations into a final segmentation.

Presently, we show techniques and results for the color and texture subspaces. However, there is no inherent limitation in our approach limiting it to these two subspaces.

For future work, we plan to experiment with additional subspaces: reconstruction geometry for multiple cameras, contours, neighborhood statistics, etc. Also, the texture subspace we are currently using is based on the Gabor filter response which assumes a repetitive texture. However, for cases with scale-varying texture, this measure will perform poorly. Alternatively, since the Gabor convolution provides a measure for repetitiveness, a better quality measure would be one that judges a neighborhood's repetitiveness. We are currently investigating both scale invariant repetitive texture measures and the use of Fourier analysis for computing a measure of repetitiveness. For the graph cut results, we plan to study the effect of using different size neighborhoods.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant Nos. 0112882 and IIS-0099770. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

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 *International Conference on Computer Vision*, 1998.

[2] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. In *Energy Minimization Methods In Computer Vision and Pattern Recognition*, pages 359–374, 2001.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[4] J. Chen, T. Pappas, A. Mojsilovic, and B. Rogowitz. Adaptive Image Segmentation Based on Color and Texture. In *Proceedings on International Conference on Image Processing (ICIP)*, 2002.

[5] J. J. Corso, M. Dewan, and G. D. Hager. Image Segmentation Through Energy Minimization Based Subspace Fusion. Technical Report CIRL-TR-04-01, The Johns Hopkins University, 2004.

[6] D. Field. Relations Between the Statistics of Natural Images and the Response Properties of Cortical Cells. *Journal of The Optical Society of America*, 4(12):2379–2394, 1987.

[7] B. Goldlucke and M. A. Magnor. Joint 3D-Reconstruction and Background Separation in Multiple Views using Graph Cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 683–688, 2003.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[9] V. Kolmogorov and R. Zabih. Multicamera Scene Reconstruction via Graph-Cuts. In *European Conference on Computer Vision*, pages 82–96, 2002.

[10] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized via Graph Cuts? In *European Conference on Computer Vision*, volume 3, pages 65–81, 2002.

[11] J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, Contours, and Regions: Cue Combination in Image Segmentation. In *International Conference on Computer Vision*, 1999.

[12] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[13] Y. Weiss. Segmentation Using Eigenvectors: A Unifying View. *International Conference on Computer Vision*, 2:975–982, 1999.