

Technical Report 2004-01-CIRL-CS-JHU

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

14 May 2004

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 7, 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.

*A short version of this paper [5] is published in the Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004).

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 In these approaches, dominant eigenvectors of the affinity matrix are used to recursively partition the image yielding a global segmentation. 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 manner 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 function² $Q_s : \mathcal{I} \rightarrow [0, 1]$ 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

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

²0 being the best quality and 1 the worst quality.

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

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

where \mathcal{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.

2.1.1 Minimizing the Energy Functional

Following [3], we construct a set of graphs to minimize (1). The graphs are constructed without α -nodes and the α -expansion steps^{3,4} are employed to minimize the energy. We explain the minimization of (1) with an exemplary graph for two pixels during one α -expansion step is presented in Figure 1 where the $\hat{Q}_{L(p)}(p)$ terms are defined by

$$\hat{Q}_{L(p)}(p) = \begin{cases} \infty & L(p) = \alpha \\ Q_{L(p)}(p) & \text{otherwise} \end{cases} \quad (3)$$

In an α -expansion step, the max-flow is computed from the α node (source) to the $\bar{\alpha}$ node (sink). The cut is defined by those edges which are saturated during the max-flow algorithm. The pixels are then labeled depending on which of their terminal edges (either α or $\bar{\alpha}$) is cut. If the α edge is cut for a pixel p , the pixel takes on the new label α : i.e. $L_p = \alpha$. But if the $\bar{\alpha}$ edge is cut for a pixel, the labeling does not change. Figure 2 presents the possible cuts on the example graph:

³One α -expansion is a single step of the approximation algorithm used to minimize the energy functional [3].

⁴Hence the use of small values for high quality and large values for low quality (Section 4.1 for more information).

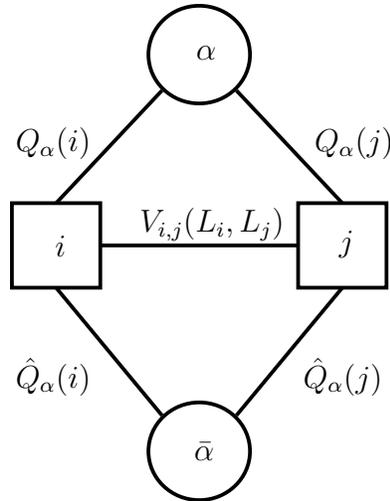


Figure 1: An exemplary graph for two pixels depicting one α -expansion step.

Cut 1 - Assigns i and j to label α .

Cut 2 - Preserves the previous labeling for i and j .

Cut 3 - Preserves i and assigns j to α .

Cut 4 - Assigns i to α and preserves j .

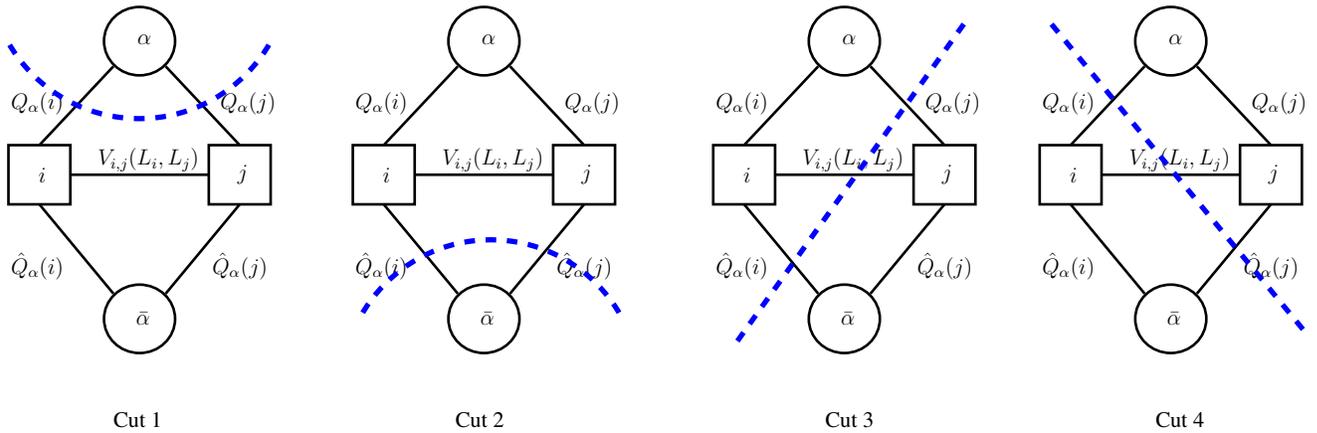


Figure 2: The four possible cuts (resulting labelings) from the max-flow procedure on the two-pixel example graph. The cuts are depicted by the blue-dashed lines.

2.1.2 The Interaction Term $V_{i,j}$

We explain the interaction term (2) by providing an intuitive motivation and then highlighting informative cases. 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 analyze the effect of the interaction term on two datasets in Section 2.1.3.

Referring to (2), for one α -expansion step there are five cases on the parameters of the interaction term. Here, the set of possible subspaces is $\{\alpha, \beta, \gamma\}$. The following table enumerates them (note that each case has either 2 or 4 subcases that depend on the subspace clustering $C_\alpha, C_\beta, C_\gamma$).

	L_i	L_j
1	α	α
2	β	β
3	β	α
4	α	β
5	β	γ

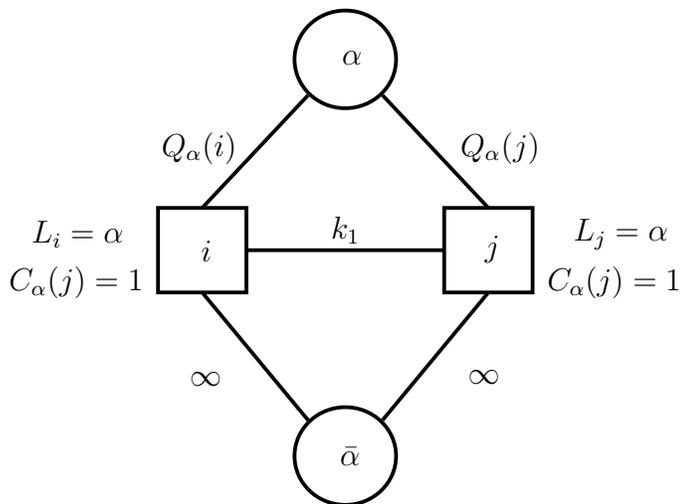


Figure 3: An example of case 1 where both i and j have the α label during an α -expansion step.

We examine a subcase of case 5 in detail through Figure 4. Both subspace clusterings have different labels, and the interaction term is high (k_2) and acts as a discontinuity preserving constraint between neighborhoods of pixels. The quality values drive the max-flow computation and govern the resulting cut; the only cut that has the potential to change the resulting segmentation here is the cut through (α, i) and (α, j) , and this should only happen if the quality for i and j in subspace α is very good. An important point to note is that the quality terms are the driving force in the α -expansion labeling with interaction terms present to enforce smoothness in some cases and to preserve discontinuities in others.

We note that case 1 is redundant. Refer to Figure 3 for a labeled graph highlighting case 1 with the subspace clustering at i and j being equal. It is easy to see that the $V_{i,j}$ term plays no role here because the two sink edges $(i, \bar{\alpha})$ and $(j, \bar{\alpha})$ have infinity weight and thus can never be cut.⁵

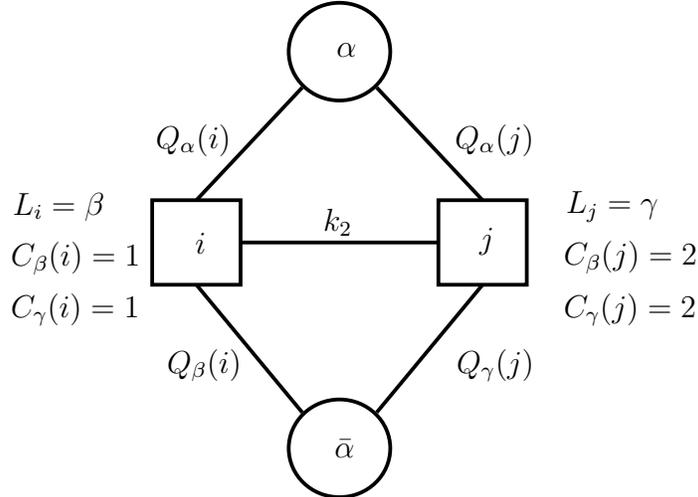


Figure 4: An example of case 5 where both i and j have labels different than α during an α -expansion step.

2.1.3 Influence of the Interaction Term $V_{i,j}$

In this section, we analyze the effect of the interaction term on the labeling of two example images. Figure 5 contains the two labelings for each image: on the left is the labeling with the interaction term and on the right is the labeling without the interaction term. For the checker board image, we see that the interaction term plays a very small role. This is due to the low frequency quality measures computed on this image (Figure 7).

However, for the zebra image, we find that the interaction term plays a strong role in preserving the smoothness of the resulting labeling. Referring to the zebra quality images in Figure 8, we find ambiguous quality in the mid-section of the smaller zebra. Thus, a greedy algorithm would perform similar to the right side (without any interaction term) and separate the smaller zebra into multiple clusters. However, our algorithm incorporates neighborhood information (through the interaction term) and the resulting labeling is smoother in the regions with ambiguous quality.

2.1.4 Proof that (1) is Graph-Minimizable

In order to prove that the energy function we use belongs to the class of functions in [10], we rewrite the energy function in the same notation as used there. Let $x_1, x_2, \dots, x_n, n = |\mathcal{I}|$ be the

⁵Remember, that an edge with high weight (capacity) can bear more flow than an edge with low weight. During the max-flow computation, in general, edges are saturated sooner if they have low capacity, and it is the saturated edges which become part of the cut. Thus, an ∞ -weighted edge can never be cut.

binary variables associated with each pixel $i \in \mathcal{I}$. The energy functional we minimize can then be rewritten in terms of these binary variables x_i as

$$E(x_1, \dots, x_n) = \sum_{i \in I} D_i(l_i(x_i)) + \sum_{i,j \in N} V_{i,j}(l_i(x_i), l_j(x_j)) \quad (4)$$

where

$$\forall i \in I \quad l_i(x_i) = \begin{cases} l_i^0, & x_i = 0 \\ \alpha, & x_i = 1 \end{cases}$$

Here l_i^0 is the labeling for pixel i before the α -expansion step. It has been shown in [10] that the energy function in (4) is graph-representable if and only if the $V_{i,j}$ satisfies the following inequality.

$$\underbrace{V_{i,j}(l_i(0), l_j(0)) + V_{i,j}(l_i(1), l_j(1))}_{L.H.S.} \leq \underbrace{V_{i,j}(l_i(0), l_j(1)) + V_{i,j}(l_i(1), l_j(0))}_{R.H.S.} \quad (5)$$

Claim 1 *The $V_{i,j}$ in (2) satisfies the inequality condition in (5).*

To show that $V_{i,j}$ in (2) is graph-representable, we consider all possible cases and show that (5) is satisfied by all of them.

Case 1. $l_i(0) = l_j(0) = \alpha$.

Substituting in the equation (5), all terms become equal to $V_{i,j}(\alpha, \alpha)$. The L.H.S. of the equation is equal to the R.H.S. thus satisfying the inequality condition.

Case 2. $l_i(0) = \alpha$ and $l_j(0) = \bar{\alpha}$ OR $l_i(0) = \bar{\alpha}$ and $l_j(0) = \alpha$.

In this case, after substitution equation (5) becomes

$$\text{L.H.S.} = V_{i,j}(\alpha, \bar{\alpha}) + V_{i,j}(\alpha, \alpha)$$

$$\text{R.H.S.} = V_{i,j}(\alpha, \alpha) + V_{i,j}(\bar{\alpha}, \alpha)$$

Since $V_{i,j}(\bar{\alpha}, \alpha) = V_{i,j}(\alpha, \bar{\alpha})$ as $V_{i,j}$ is symmetric by definition, the equality condition in (5) holds.

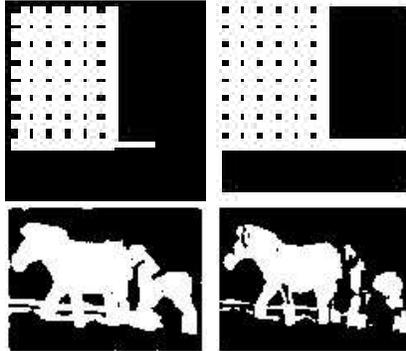


Figure 5: Labeling results from graph-cut methods with (left) and without (right) inclusion of the interaction term for the synthetic dataset and the zebra image.

Case 3. $l_i(0) = l_j(0) = \bar{\alpha}$.

After substitution, equation (5) for this case becomes

$$\text{L.H.S.} = V_{i,j}(\bar{\alpha}, \bar{\alpha}) + V_{i,j}(\alpha, \alpha)$$

$$\text{R.H.S.} = V_{i,j}(\bar{\alpha}, \alpha) + V_{i,j}(\alpha, \bar{\alpha})$$

Depending on the clustering of pixels i, j in both α and $\bar{\alpha}$ subspaces, we have four sub-cases outlined in the following table. One can see from the table for all the four cases that the $L.H.S. \leq R.H.S.$ and thus the inequality condition in (5) holds.

Clustering of i, j in α subspace	Clustering of i, j in $\bar{\alpha}$ subspace	L.H.S.	R.H.S.
Same	Same	$k_1 + k_1$	$k_1 + k_1$
Different	Same	$k_1 + k_2$	$k_2 + k_2$
Same	Different	$k_2 + k_1$	$k_2 + k_2$
Different	Different	$k_2 + k_2$	$k_2 + k_2$

Case 4. $l_i(0) = \beta, l_j(0) = \gamma$ s.t. $\beta, \gamma \in \bar{\alpha}$ & $\beta \neq \gamma$.

The equation (5) for this case becomes

$$\text{L.H.S.} = V_{i,j}(\beta, \gamma) + V_{i,j}(\alpha, \alpha)$$

$$\text{R.H.S.} = V_{i,j}(\beta, \alpha) + V_{i,j}(\alpha, \gamma)$$

There are eight sub-cases depending on the clustering of the pixels i, j in α, β and γ subspaces. In the following table, these sub-cases are enumerated. The $L.H.S. \leq R.H.S.$ in all of them and thus the inequality condition in (5) holds.

Clustering of i, j in α subspace	Clustering of i, j in β subspace	Clustering of i, j in γ subspace	L.H.S.	R.H.S.
Same	Same	Same	$k_1 + k_1$	$k_1 + k_1$
Same	Different	Same	$k_2 + k_1$	$k_2 + k_1$
Same	Same	Different	$k_2 + k_1$	$k_1 + k_2$
Same	Different	Different	$k_2 + k_1$	$k_2 + k_2$
Different	Same	Same	$k_1 + k_2$	$k_2 + k_2$
Different	Same	Different	$k_2 + k_2$	$k_2 + k_2$
Different	Different	Same	$k_2 + k_2$	$k_2 + k_2$
Different	Different	Different	$k_2 + k_2$	$k_2 + k_2$

As, the condition in (5) is satisfied by all the possible cases, the energy function (1) used for subspace fusion is graph representable.

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 $C_{L(i)}$:

$$C_f(i) = C_{L(i)}(i), \forall i \in \mathcal{I} \quad (6)$$

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 6. 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 \mathcal{I})

- 1 **foreach** (subspace $s \in \mathcal{S}$)
- 2 **foreach** (pixel $i \in \mathcal{I}$)
- 3 Compute quality $Q_s(i)$
- 4 Compute clustering $C_s(\mathcal{I})$
- 5 Compute labeling $L(\mathcal{I})$
- 6 **foreach** (pixel $i \in \mathcal{I}$)
- 7 $C_f(i) = C_{L(i)}(i)$
- 8 **return** C_f as final segmentation

Figure 6: Proposed segmentation algorithm.

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

$$A_{ij} = e^{\frac{-\|\mathbf{F}(i)-\mathbf{F}(j)\|_2}{\sigma_F}} * \begin{cases} e^{\frac{-X_{ij}}{\sigma_X}} & \text{if } X_{ij} < r \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

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 , $\mu^p(i) = \frac{1}{|\mathcal{N}|} \sum_{x \in \mathcal{N}} \mathcal{I}^p(x)$ where $p \in \mathbf{r}, \mathbf{g}, \mathbf{b}, \mathbf{i}$ is a plane of the color image or grayscale, and $\sigma^p(i) = \frac{1}{|\mathcal{N}|} \sum_{x \in \mathcal{N}} (\mathcal{I}^p(\mathbf{x}) - \mu^p(i))^2$.

$$\sigma_c(i) = \max[\sigma^{\mathbf{r}}(i), \sigma^{\mathbf{g}}(i), \sigma^{\mathbf{b}}(i)] \quad (8)$$

$$\sigma_t(i) = \sigma^{\mathbf{i}}(i) \quad (9)$$

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).

$$Q_c(i) = \begin{cases} 1 & \text{if } \sigma_c(i) > \eta \\ \frac{\sigma_c(i)}{\eta} & \text{otherwise} \end{cases} \quad (10)$$

$$Q_t(i) = 1 - \sigma_t(i) \quad (11)$$

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** - $\mathbf{F}(i) = [|\mathcal{I} * g_1|, \dots, |\mathcal{I} * g_n|](i)$, 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 7 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 8 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

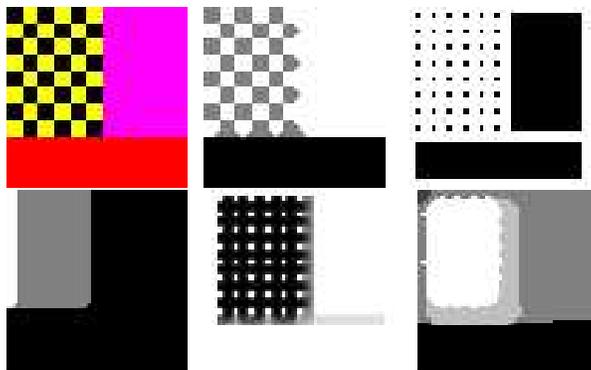


Figure 7: Result of subspace fusion on a synthetic image. First row shows the original image, and the color subspace segmentation and quality. Second row shows the texture subspace segmentation and quality and the final fusion segmentation. Note that quality ranges from 0 (black) to 1 (white) with 0 the best quality and 1 the worst quality.

texture information is subdued. We expect that on higher resolution images the results would be much better, but due to the high computation cost involved with segmenting high resolution images, we currently restrict ourselves to low resolution images.

In Figure 9, we show the quality information in both the color and texture subspace on a leopard image. The labeling assigned by the graph-cut energy minimization is also shown with black corresponding to color and white to texture. Finally the clusters and the segmentation overlaid on the original image are shown.

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.

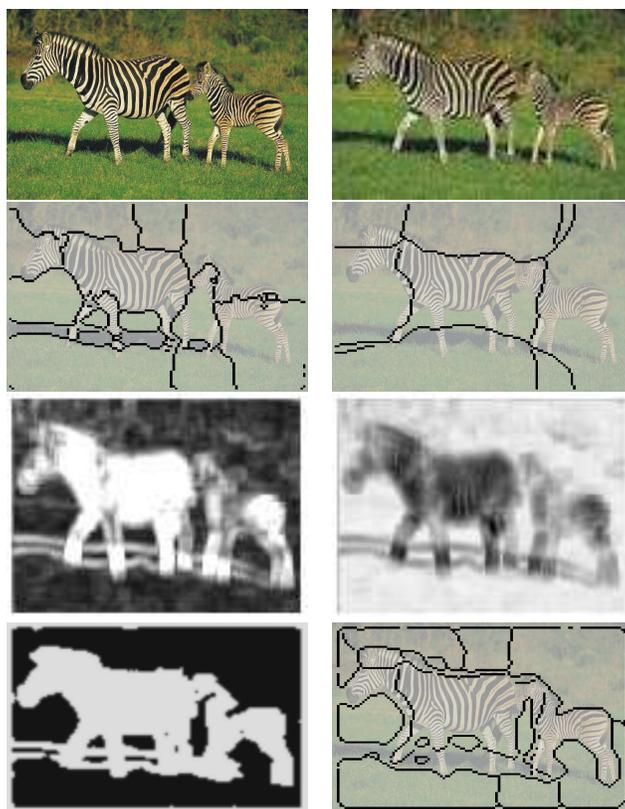


Figure 8: Complete segmentation for the zebra image. Top-left is the high resolution image and top-right is the low-resolution image used. Row 2-left is the color subspace clustering, Row 2-right is the texture subspace, Row 3-left is the color subspace quality, Row 3-right is the texture subspace quality, bottom-left is the labeling with black assigned to color, and bottom-right shows the final clustering.

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. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, 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.

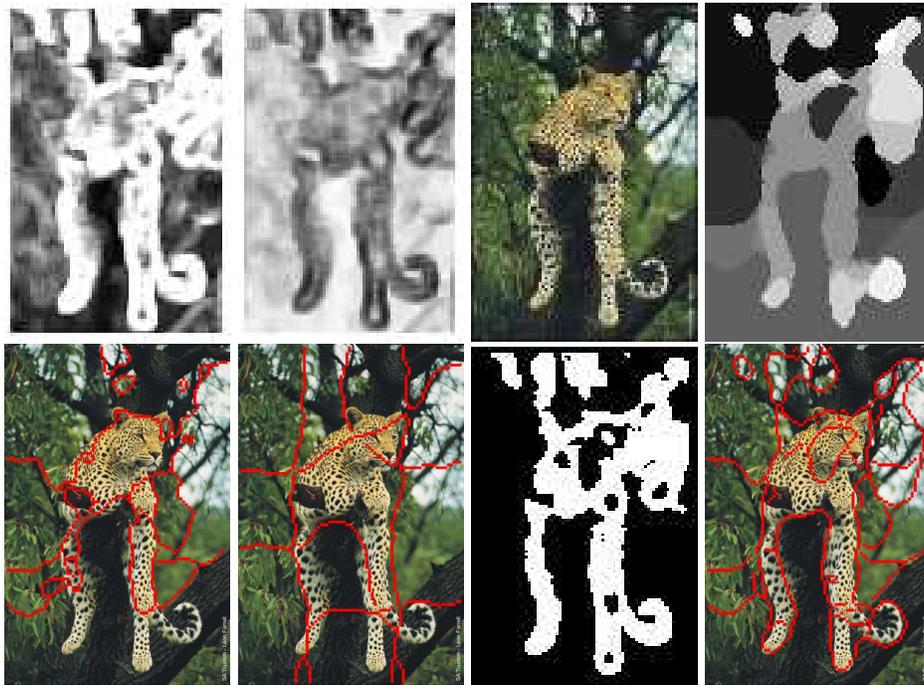


Figure 9: A leopard image dataset showing in column 1 the quality and clustering in the color subspace, column 2 the quality and clustering in the texture subspace, in column 3 the original image and the labeling, and in column 4 the final segmentation.