

Multiscale Modeling and Constraints for Max-flow/Min-cut Problems in Computer Vision

Matthew W. Turek and Daniel Freedman
Rensselaer Polytechnic Institute
Troy, NY 12180
{turekm, freedman}@cs.rpi.edu

Abstract

Multiscale techniques have been used for many years in computer vision. Recently multiscale edges have received attention in spectral graph methods as an important perceptual cue. In this paper multiscale cues are used in the context of max-flow/min-cut energy minimization. We formulate multiscale min-cut versions of three typical computer vision applications, namely interactive segmentation, image restoration, and optical flow. We then solve across all scales simultaneously. This use of multiscale models and constraints leads to quantitatively and qualitatively improved experimental results.

1. Introduction

Recently Yu [30, 29] has shown that multiscale edges are an important segmentation cue. Similarly, Benezit et al [2] have shown that multiscale constraints in spectral methods can be efficiently implemented by decomposing the affinity matrix into multiple scales. Much earlier, Kato et al [17] used a multiscale MRF formulation with a novel multitemperature annealing algorithm to solve a multiclass segmentation problem. The first two techniques, however, use approximation algorithms to solve their energy function and the latter technique uses an optimization technique that is only optimal in the asymptotic sense.

The goal of this paper is understanding the role of multiscale cues in the context of an algorithm which is globally optimal or generates a provably strong minimum [5] for vision problems. Globally optimal techniques are important because the results reflect directly the energy formulation. We formulate min-cut versions of three typical vision problems, interactive segmentation, optical flow, and image restoration and then evaluate the effect of multiscale cues. A global minimum for the segmentation formulation can be realized via min-cut as demonstrated by Boykov et al [6]. Similarly, the optical flow and image restoration prob-

lems can be minimized to a strong local minimum using the α -expansion technique from [5].

We will not describe the background involved in max-flow/min-cut energy minimization in this paper due to space constraints. Instead we refer the reader to [5], [18], [12].

Consider a set of graphs representing a binary energy minimization problem of the form described by Boykov et al [5, 6]. Let each of these graphs represent the problem at a different scale. We apply constraints between these graphs and their solutions by adding weighted edges connecting pixels in one scale to pixels in another scale. The weight of the edges determines how strong the constraint is between the problems at various scales and the topology of the connections encodes how the solution varies across scales. The enforcing of smoothness across scales is equivalent to a multiscale smoothness prior. Multiscale information, reflecting *a priori* knowledge, can also be incorporated into the modeling for the problem at each scale. In particular, we will show a multiscale appearance model to aid interactive segmentation. A min-cut is computed on the resulting connected graph providing a simultaneous solution to the multiscale problems.

We begin by adding multiscale terms to the energy function:

$$E(L) = \sum_{s \in S} \left(\lambda \sum_{p \in \mathcal{P}_s} D(L_p) + (1 - \lambda) \sum_{\substack{p \in \mathcal{P}_s \\ q \in \mathcal{N}_s(p)}} V_{in}(L_p, L_q) \right) + \mu \sum_{s \in S} \sum_{\substack{p \in \mathcal{P}_s \\ r \in \mathcal{N}_\gamma(p)}} V_{btw}(L_p, L_r). \quad (1)$$

where S is a set of scales, \mathcal{P}_s is the set of pixels at scale S , $\mathcal{N}_s(p)$ is the neighborhood of pixel p at scale s , $\mathcal{N}_\gamma(p)$ is the neighborhood of pixel p across across scales, $\lambda \in [0, 1]$ controls the tradeoff of the within scale fit and smoothness terms and μ controls the strength of the smoothness across scales. Note that the terms inside the first summation over

scales are the set of graphs representing the multiscale problems and the last additive term contains the constraints between the multiscale problems.

For our applications we choose a Gaussian pyramid representation. Specifically, we smooth with a Gaussian with standard deviation 1.0 and downsample by 2. The neighborhood for a particular pixel includes the usual neighbors on the same scale and the parents or children in the multiscale structure. The number of levels used for a particular problem will be detailed in the section containing the experiments.

We have found it beneficial to introduce area based weighting of the fit terms similar to the modified cost terms used in Bouman and Shapiro [4]. As we move higher in the scale representation, pixels cover a larger area. Incorrectly fitting a pixel at the lowest level should have a lesser impact on the energy than incorrectly fitting a pixel at the highest level. Therefore, we weight our $D(\cdot)$ terms by the area that a pixel projects to in the finest scale.

We use the max-flow/min-cut minimization library described in [18].

2. Multiscale Modeling and Constraints

2.1. Multiscale Interactive Segmentation

Previous authors have used min-cut techniques to find segmenting contours [28, 6, 25, 19, 20, 27, 21], however none of these techniques incorporates simultaneous multiscale models or constraints. Wang et al [27] uses mean-shift presegmentation to group pixels into regions to reduce node count, but does not operate in scale-space. Similarly, Li et al [19] groups regions within and between frames, but again does not operate in scale-space. Lombaert et al [21] used a coarse-to-fine minimization scheme in conjunction with max-flow energy minimization to improve running time. This technique, however, does not contain multiscale models or simultaneous multiscale constraints. Xu et al [28] developed an active contour technique based on graph cuts for object segmentation and tracking. However, their technique iterates only in the neighborhood of the initialization or prior solution. As such, the final solution is no longer a global minimum. Kato et al [17] constructs a multiscale MRF for segmentation, but does not use multiscale segmentation models; instead they use Gaussian models that are the same at all scales.

Our segmentation technique begins with the user specifying an initialization region inside the object of interest. From this region we derive both a multiscale appearance model and hard constraints for the location of the object. A set of optimal contours (in terms of the energy in (1)) is computed over multiple scales.

2.1.1 Multiscale Appearance Modeling

Our appearance model is a set of multiscale densities. Hadjemetriou et al [15] have shown that this model encodes spatial intensity information and is effective for texture discrimination.

To generate the multiscale model, we first compute a Gaussian pyramid for the input image. Next, we compute the projection of the initialization region onto the Gaussian pyramid generating a pyramid of binary masks. The probability for the intensity of pixels inside the initialization mask at each level is computed using a univariate kernel density estimate¹:

$$\hat{p}(z) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{z - z_i}{h}\right) \quad (2)$$

where K is the smoothing kernel, h is the bandwidth, and z_i are the points in the initialization region at the same scale as z . For K we use the Epanechnikov kernel, which is optimal in the asymptotic mean integrated squared error (AMISE) sense [26]. Several techniques exist for estimating the bandwidth based on the sample data [10], [9], [26]; however, in the experiments discussed later we leave the bandwidth as a user specified parameter.

2.1.2 Hard constraints for foreground and background

Hard constraints for the foreground object are modeled using a large negative constant for the pixels inside the initialization region. This forces certain pixels to be included in the final object segmentation. In previous work [6], hard constraints for the foreground and background were specified by the user. Rother et al [25] attempted to reduce the need for user interaction by removing foreground seeds. Instead the user specified a region containing the object of interest and some of the background. A model for the background was derived from the perimeter of the user specified region.

We do something similar, but instead ask the user to specify a region inside the object. This allows us to set foreground hard constraints and model the appearance of the object. For the background, we automatically set hard constraints around the perimeter of the image which forces a cut to be made between the edge of the image and the user specified foreground region. Only using background constraints as in [25] may make modeling difficult because the background may be spatially varying and provide conflicting cues. Additionally, we hope to eventually apply a foreground only model to object tracking where the object appearance is relatively constant, but the background may vary substantially across frames. Having a foreground model is important for this future application.

¹This can easily be extended to RGB or n-component images through the use of multivariate density estimation. In this paper, our work is on grayscale images only.

2.1.3 Fit Terms

The foreground fit terms are estimates of how well a particular pixel in the image matches the appearance model. The foreground cost is :

$$D(L_p) \propto -\ln Pr(I_p|L_p) \quad (3)$$

where $Pr(I_p|L_p)$ is the probability of the pixel intensities given the region label (object, "obj" or background "bkg"). We estimate $Pr(I_p|L_p = obj)$ using the kernel density estimate described above. In modeling the background, we assume no information about the background except that it is different than the foreground. Without this assumption, the object is camouflaged and will be difficult to identify even for a human observer. Regions with low probability of being foreground should be strong candidates for being in the background. Consequently the background appearance model is taken to be the complement of the foreground model.

We estimate $Pr(I_p|L_p = bkg)$ as:

$$Pr(I_p|L_p = bkg) = \frac{H(I_p)}{C} (\max_p Pr(I_p|L_p) - Pr(I_q|L_q = obj)). \quad (4)$$

where $H(I_p)$ is 1 if I_p is in the dynamic range of the image representation and 0 if it is not and C is chosen to make the distribution integrate to 1.

2.1.4 Smoothness terms

Typically, neighborhood weights are dependent on the image intensity. For example Boykov and Jolly [6] use:

$$V_{in}(L_p, L_q) \propto \frac{\delta(L_p, L_q)}{dist(p, q)} \cdot e^{-\frac{(I_p - I_q)^2}{2 * \sigma^2}} \quad (5)$$

where $\delta(L_p, L_q)$ is 1 only if L_p and L_q are different; otherwise the term is zero.

As Wang et al [27] notes, this neighborhood weight assumes that there are higher gradients between the foreground and background than there is within the foreground object. Consequently, this weighting may be problematic at times because it over emphasizes image edges. Wang et al [27] proposed a modification to the neighborhood costs that takes into account whether the edge is unusually strong. However because of the continued reliance on edges, images with strong contrast can still be a problem. For example, in Figure 5, the image in the third row contains a man with a white shirt and black jacket. Initializing the foreground distribution to cover both the shirt and jacket results in a Boykov-Jolly segmentation that omits a large portion of the shirt and instead follows the strong edges around the

jacket (column 1). These edge between the jacket and shirt are close to the maximum possible contrast. With the model based boundary terms proposed below, the segmentation includes the shirt and jacket (single scale, column 2 and multiscale, column 3).

To deal with edges in the foreground, we suggest the following neighborhood edge weights based on our foreground intensity distribution:

$$V_{in}(L_p, L_q) \propto \frac{\delta(L_p, L_q)}{dist(p, q)} \cdot e^{-\frac{(Pr(I_p|L_p=obj) - Pr(I_q|L_q=obj))^2}{2 * \sigma_N^2}} \quad (6)$$

where σ_N is a scale parameter. We use a fraction of the foreground weights' dynamic range to set σ_N . Instead of emphasizing image edges, our smoothness weight emphasizes changes in the strength of the foreground model. Boundaries will be placed at rapid changes in the appearance model strength that outweigh the influence of the appearance model region terms.

The between layer smoothness weights are defined similarly to the within layer smoothness weights.

$$V_{btw}(L_p, L_r) \propto \delta(L_p, L_r) e^{-\frac{(Pr(I_p|L_p=obj) - Pr(I_r|L_r=obj))^2}{2 * \sigma_\sigma^2}} \quad (7)$$

where σ_σ is also a scale parameter. Again, we use a fraction of dynamic ranges of the two layers' foreground models to set σ_σ .

2.2. Multiscale Restoration

We frame the restoration problem as follows: given an image corrupted with zero mean Gaussian additive noise, estimate the original image. In the presence of strong noise, nearby pixel information may be unreliable. Increasing the role of smoothing in the energy function may help, but this can lead to oversmoothed solutions. As we will demonstrate in Section 3.2, using multiscale information can provide a more accurate and reliable solution. Using a Gaussian pyramid for our multiscale representation gives us a way of combining directly the original, noisy data and the smoothed, lower resolution data. For this particular problem we rely on multiscale constraints.

For our fit energy we chose the truncated L_2 norm:

$$D(L_p) = \min(\|L_p - I_p\|_2, C_D) \quad (8)$$

where $\|\cdot\|_2$ represents the L_2 norm and I_p is the observed image intensity at pixel p . The truncation of the norm de-emphasizes large outliers, thus preventing the solution from being too strongly biased by poor local fits.

We use a truncated L_1 norm for the within scale neighborhood:

$$V_{in}(L_p, L_q) = \min(\|L_p - L_q\|_1, C_V) \quad (9)$$

where $\|\cdot\|_1$ represents the L_1 norm, L_p and L_q are the restoration labels for pixels p and q , respectively, and C_V is a constant. This is the same smoothness term used in [5] for image restoration. Again, truncating the smoothness penalty increases our robustness to large outliers.

Similarly, we use a truncated L_1 norm for the between scale neighbors:

$$V_{btw}(L_p, L_r) = \min(\|L_p - L_r\|_1, C_V) \quad (10)$$

where $\|\cdot\|_1$ represents the L_1 norm and L_p and L_r are the restoration labels for between scale pixels p and r , respectively, and C_V is the same constant in (9).

2.3. Multiscale Optical Flow

Multiscale techniques have been used before for optical flow [7, 8, 16, 22]. These techniques often involve a multi-scale warping strategy where a lower resolution problem is solved first to remove large motions and then that solution is propagated to a higher resolution version of the problem as initialization. This improves the results for large scale displacements.

Prior to the techniques in Boykov et al [5], Bonneville [3] computed optical flow using max-flow for the limited case of 1D disparities at a single scale. More recently Freedman et al [13] introduced a single scale illumination invariant optical flow technique using graph cuts.

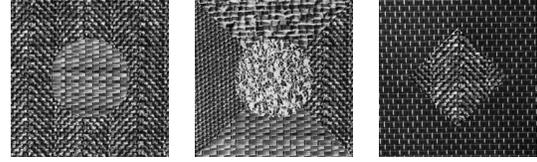
In this paper, we compute the optical flow field across all scales simultaneously. This is in the spirit of coarse-to-fine followed by fine-to-coarse iterations, however, we've removed the iteration across scales and instead have simultaneous estimation at multiple scales.

The fit term for a label is the cost of matching the pixel in the current frame to the location indicated by the displacement label in the second frame. To compute the pixel matching cost, we use a robust loss function to avoid overpenalizing bad matches. Our fit energy is:

$$D(L_p; s) = \rho \left(\frac{I_t(p) - I_{t+1}(p + T(L_p; s))}{\sigma} \right) \quad (11)$$

where $I_t(p)$ is the intensity of the first image at pixel p , $I_{t+1}(p + T(L_p; s))$ is the intensity of corresponding pixel in the second image, σ is a scale estimate, and $\rho(\cdot)$ is defined to be the Beaton-Tukey robust estimator with $B=4.5$ for all experiments. Attempting to deal with the different scales directly in the between scale smoothness terms leads to violations of the regularity condition. Instead, we use $T(L_p; s)$ here to denote that the quantized displacement vector, L_p , must be scaled appropriately for the current level of the Gaussian pyramid. In our case, we use a Gaussian pyramid with scales at each octave and isotropic scaling. Then $T(L_p; s)$ becomes:

$$T(L_p; s) = \frac{1}{2^s} L_p. \quad (12)$$



(a) D17D55 (b) nat-5b (c) nat3

Figure 1. Test textures for the contour segmentation technique

Note that our scales are zero indexed.

We choose to use a weighted, truncated L_2 norm for our within scale smoothness penalty. The weights incorporate static cues information and are based on the weight function from the Beaton-Tukey loss function. Specifically, our smoothness cost is:

$$V_{in}(L_p, L_q) = (1.0 + w[I(p) - I(q)]) * \min(\|L_p - L_q\|_2, C_{V_o}) \quad (13)$$

where $w(u) = (1 - (u/B)^2)^2$ if $u \leq B$ and 0 otherwise. We set $B=4.5$. Similarly, we use the truncated L_2 norm for our between scale smoothness penalty, but without the static cues:

$$V_{btw}(L_p, L_r) = \min(\|L_p - L_r\|_2, C_{V_o}). \quad (14)$$

Since we've accounted for the scaling of the flow vectors between scales in the fit term, nothing special needs to be done in the between scale smoothness weights.

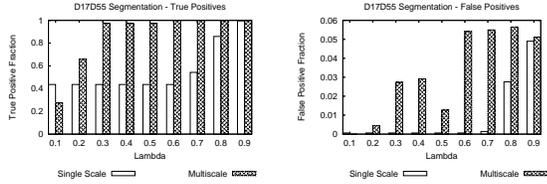
3. Experimental Results

In this section we experimentally compare the multiscale and single scale energy formulations for interactive segmentation, restoration, and optical flow. Quantitative error measures are used to critique the results. The main purpose of our experiments is to assess whether multiscale energy function lead to quantitatively better solutions for our chosen problems.

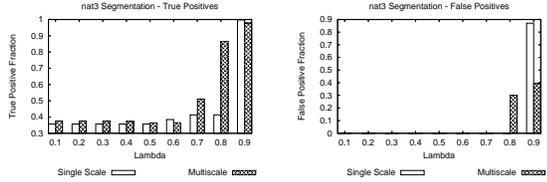
3.1. Multiscale Interactive Segmentation

Here we compare the results of the multiscale and single scale contour segmentation techniques on a set of texture images². We choose to test initially on texture images for two reasons: they highlight the power of the multiscale density modeling and we have ground truth for the regions. As part of quantitatively comparing the multiscale and single scale techniques we also wish to study how robust the techniques are to the initialization region and the λ smoothing parameter. Sensitivity to initialization is an important issue for user guided segmentation techniques. We do not want our results largely dependent on where (within reason) the user chooses the initialization region. It is also useful to assess whether the multiscale formulation is less sensitive to λ which trades off the fit and smoothness terms.

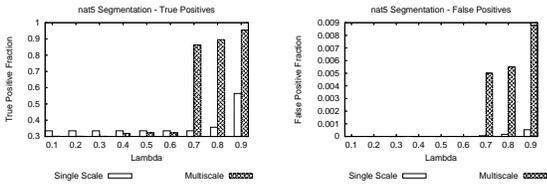
²Images are available at <http://www.ux.his.no/tranden/data.html> courtesy of Trygve Randen.



(a) Texture D17D55, plot of true positives fraction vs. λ . (b) Texture D17D55, plot of false positives fraction vs. λ .

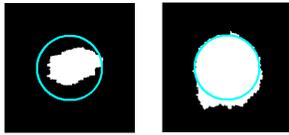


(c) Texture nat3, plot of true positives fraction vs. λ . (d) Texture nat3, plot of false positives fraction vs. λ .



(e) Texture nat5, plot of true positives fraction vs. λ . (f) Texture nat5, plot of false positives fraction vs. λ .

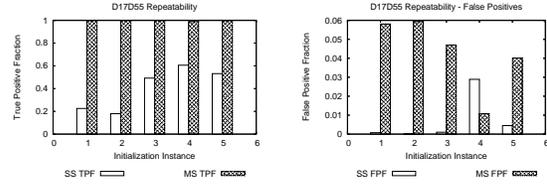
Figure 2. True positive and false positive segmentation results.



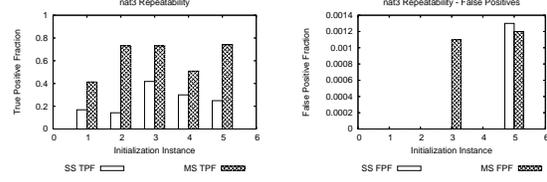
(a) Texture D17D55 Single scale (b) Texture D17D55 Multiscale

Figure 3. Example results for Texture D17D55. This result corresponds to Figure 2 (a),(b) at $\lambda = 0.6$. The multiscale technique overshoots the desired boundary (shown as the contour) slightly around the bottom of the region, producing false positives. The single scale technique fails to segment most of the desired region and only approaches the boundary of the desired region on the right side. The single scale technique also overshoots the boundary slightly on the right side, but produces lower false positives because it significantly undersegments the region, thus avoiding most of the difficult boundary decisions. This is typical of the results for textures 1-3 and indicates the cause for the increased false positives of the multiscale technique.

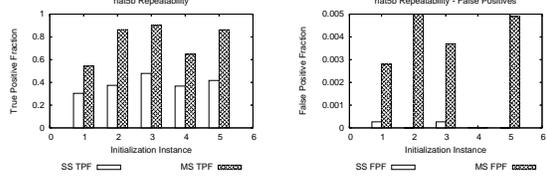
Figure 1 contains three test textures used to evaluate the performance of the contour segmentation algorithm. All experiments attempted to segment the center region (circles or diamond) and ground truth is known. The bandwidth was fixed at 30 for all experiments. The between scale smoothness coefficient μ (multiscale algorithm) was fixed at 0.2. The multiscale algorithm used a four layer Gaussian pyramid.



(a) D17D55 Repeatability True Positives (b) D17D55 Repeatability False Positives



(c) nat3 Repeatability True Positives (d) nat3 Repeatability False Positives



(e) nat5b Repeatability True Positives (f) nat5b Repeatability False Positives

Figure 4. Segmentation Repeatability Experiment. Five initialization regions with different sizes and locations were used to compare the multiscale and single scale segmentation techniques. The true positive plots all have the same scale, while the false positive plots have varying scales.

Figure 2 contains the results of the texture experiments. The multiscale segmentation algorithm does substantially better than the single scale technique across all λ values. The multiscale algorithm does have higher false positives, however, this is because the single scale technique often significantly undersegments the region, thus avoiding difficult decisions on the boundary between two types of texture. See Figure 3 for an example. The multiscale result in that figure is quite good except at the bottom boundary. Examining a small version of the test texture in Figure 1 demonstrates the problem, however. The boundary between the two textures is even difficult for a human observer.

Several experiments were run to assess the sensitivity to user initialization. For each image, five different initialization regions of different sizes were chosen in different locations. λ was fixed at 0.7 with the remaining parameters the same as in the earlier experiments. The results for the different initializations are shown in Figure 4. The multiscale technique shows consistently better results across all images. Again false positives in the multiscale technique are the result of facing difficult boundary localization decisions that the single scale technique avoids by not segmenting near the boundary.

Finally, experiments were run on images from the Berkeley segmentation database [23]. The parameters were the same as in the initialization experiment. The experiments also include an implementation of the single scale edge-based method for comparison. The results of these experiments can be seen in Figures 5 and 8. No user interaction or editing, besides choosing the initialization region, was used.

The multiscale technique often does better in these real images than either the single scale, edge-based smoothness technique or the single scale, density based smoothness technique. On simple scenes, like the church in Figure 5, row 2, the edge-based single scale technique may do slightly better because of the strong edges and relatively simple appearance of the object. However, in more complicated scenes, such as Figure 5, row 10, the multiscale technique does a better job isolating the object of interest. Even though the edge-based technique was configured with low boundary weights, resulting in a tendency to segment a lot of the image, areas with edges in the foreground are often avoided. For example, Figure 5 row 8, there are holes in the segmentation of the elephants. Similarly in rows 4,5,6 and 9 in Figure 5, and rows 2,3 in Figure 8 there are holes in the foreground objects. The appearance based smoothness techniques, both at single and multiple scales, have less tendency to generate holes around foreground edges. In row 4 Figure 5 the multiscale technique does a particularly good job ignoring the background texture erroneously identified by both single scale techniques.

3.2. Multiscale Restoration

Experiments were conducted on the three images shown in Figure 6. Zero-mean Gaussian noise with $\sigma = 25$ was added to the wedding cake and penguin images while zero-mean Gaussian noise with $\sigma = 15$ was added to the Lena image. Both the single scale and multiscale restoration algorithms were applied to each image over a variety of smoothness parameters (λ). The λ values used correspond to visually meaningful results. Two layers were used in the multiscale formulation. The constants C_V and C_D were set to 10 for all images. The coefficient on the between layer weights, μ , was set to 0.05 for the wedding cake and penguin images and 0.2 for the Lena image. The restoration used 255 labels.

The results were quantitatively compared to the original, uncorrupted images using the sum of squared differences. Figure 7 contains graphs of the percent reduction in error resulting from the use of the multiscale restoration for the wedding cake, penguin, and Lena images, respectively. A negative value indicates that the multiscale restoration was worse than the single scale restoration.

In general, the multiscale technique produced significantly better results than the single scale energy function. Across all restoration experiments, the mean improvement

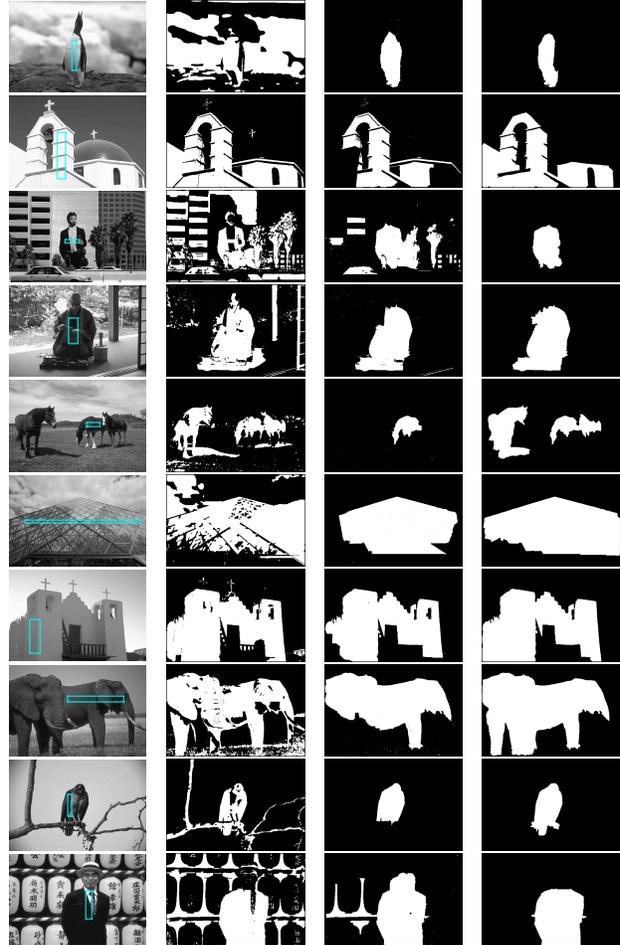


Figure 5. Segmentation results on images from the Berkeley database. The first column shows the input image and the initialization. The second column contains the result of the single scale Boykov-Jolly. The third column contains the result of the single scale technique with appearance model based smoothness. The fourth column contains the result of the multiscale appearance model in conjunction with the appearance model based smoothness. See Figure 8 for more results.

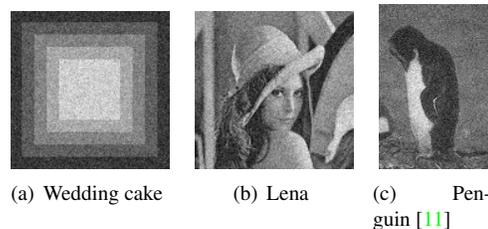


Figure 6. Noisy restoration images. The wedding cake and penguin images were corrupted with additive Gaussian noise (mean = 0, std = 25) and the Lena image was corrupted with additive Gaussian noise (mean = 0, std = 15).

in SSD from using the multiscale energy function was 20%. The multiscale energy function was only worse than the single scale version once. In that case, the SSD energy was only 1% different.

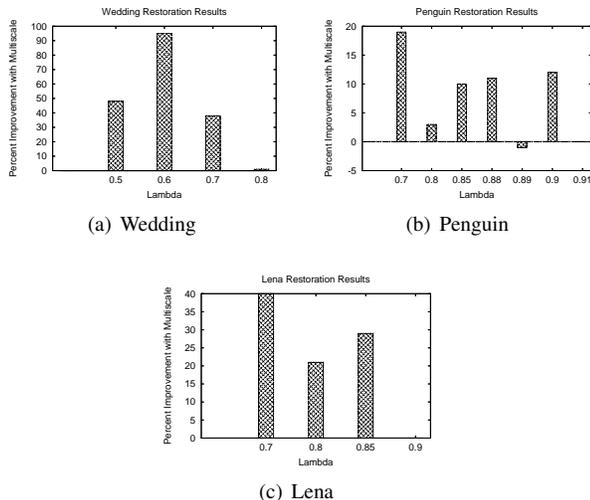


Figure 7. Improvement in image restoration from the use of the multiscale technique.

Sphere sequence		
Error Measure	Single scale	Multiscale
Mean error vector length (pixels)	0.4157	0.2283
Mean angular error (degrees)	8.5486	5.8943
Yosemite sequence (without clouds)		
Error Measure	Single scale	Multiscale
Mean error vector length (pixels)	0.2017	0.1657
Mean angular error (degrees)	12.9060	12.1242
Street sequence		
Error Measure	Single scale	Multiscale
Mean error vector length (pixels)	0.1639	0.1750
Mean angular error (degrees)	5.0378	5.1463

Table 1. Error measurements on the optical flow sequences. Both algorithms were run at $\lambda = 0.4$. The multiscale algorithm had μ fixed at 0.2 and four scales.

3.3. Multiscale Optical Flow

Several experiments were conducted on three previously published optical flow sequences with ground truth: the sphere and street sequences from McCane et al [24] and the Yosemite sequence without clouds.

We use two error measurements to compare our results: the mean error vector length and the mean angular error [14],[8],[1].

To estimate the optical flow field as a labeling problem, we need to discretize the flow vectors. For all the experiments, we allow flow vectors from $(-6, -6)$ to $(6, 6)$ in steps of a half pixel yielding 625 possible flow labels.

As with the previous applications, the multiscale energy leads to improved results. The multiscale technique is often significantly better than the single scale energy in terms of the mean error vector length and mean angular error. When the multiscale technique is not better, it is either as good or only slightly worse. In particular, the multiscale is slightly worse (about 6% in terms of mean error vector length and about 2% in terms of angular error) on the

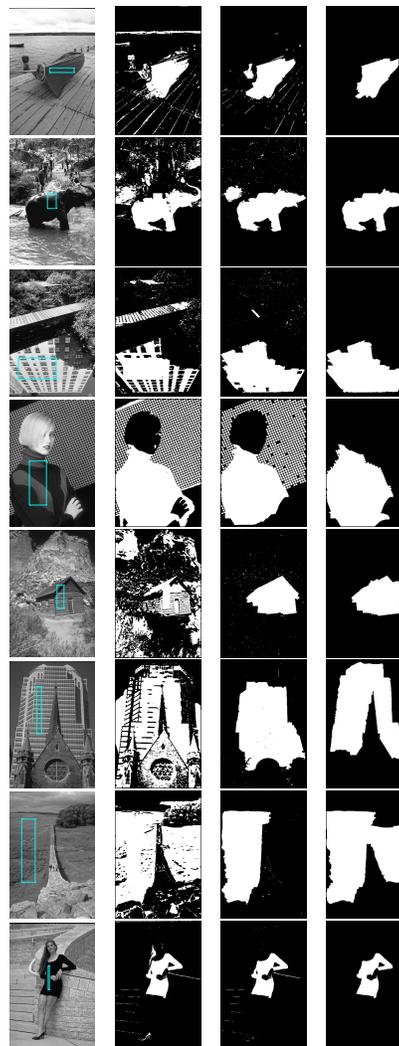


Figure 8. More segmentation results on the Berkeley database. The first column is the input image and initialization. The second column is the single scale Boykov-Jolly, the third contains single scale with appearance model based smoothness and the fourth is a multiscale appearance model with appearance model based smoothness.

street sequence. In this sequence, the flow field contains a large homogeneous background with a relatively small foreground with small disparity in flow vectors. In contrast, the sphere sequence and yosemite sequences have more complicated flow patterns where the multiscale smoothness constraints become more important.

4. Conclusion

We've formulated three important computer vision problems in the max-flow/min-cut energy minimization framework and solved them simultaneously at multiple scales. The experimental results show that this leads to quantitatively and qualitatively improved results, lending evidence to the hypothesis that multiscale cues are important in low-level computer vision problems.

Acknowledgements

This research was supported by U.S. Army Intelligence and Security Command contract W911W4-F-04-0131.

References

- [1] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994. 7
- [2] F. Benezit, T. Cour, and J. Shi. Spectral segmentation with multi-scale graph decomposition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1124–1131, June 2005. 1
- [3] M. Bonneville and J. Meunier. Finding the exact optical flow: a maximum flow formulation. In *Vision Interfaces*, pages 418–423, 1999. 4
- [4] C. A. Bouman and M. Shapiro. A multiscale random field model for bayesian image segmentation. *IEEE Transactions on Image Processing*, 3(2):162–177, March 1994. 2
- [5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1–18, November 2001. 1, 4
- [6] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *IEEE International Conference on Computer Vision*, pages 105–112, 2001. 1, 2, 3
- [7] T. Brox, A. Bruhn, N. Papenbergh, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proceedings of the 8th European Conference on Computer Vision*, volume 4, pages 25–36, May 2004. 4
- [8] A. Bruhn and J. Weickert. Lucas/Kanade Meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005. 4, 7
- [9] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):281–288, February 2003. 2
- [10] 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. 2
- [11] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004. 6
- [12] D. Freedman and P. Drineas. Energy minimization via graph cuts - settling what is possible. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 1
- [13] D. Freedman and M. Turek. Illumination invariant tracking via graph cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 4
- [14] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: An evaluation of eight optical flow algorithms. In *Proceedings of the British Machine Vision Conference*, September 1998. 7
- [15] E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar. Multiresolution histograms and their use for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):831–847, July 2004. 2
- [16] B. Jahne, H. Haussecker, and P. Geissler, editors. *Handbook of Computer Vision and Applications*. Academic Press, 1999. 4
- [17] Z. Kato, M. Berthod, and J. Zerubia. A hierarchical Markov random field model and multitemperature annealing for parallel image classification. *Graphical Models and Image Processing*, 58(1):18–37, January 1996. 1, 2
- [18] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147 – 159, February 2004. 1, 2
- [19] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. In *ACM Transactions on Graphics (SIGGRAPH'05)*, volume 24, pages 595–600, 2005. 2
- [20] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. In *ACM Transactions on Graphics (SIGGRAPH'04)*, volume 23, pages 303–308, 2004. 2
- [21] H. Lombaert, Y. Sun, L. Grady, and C. Xu. A multilevel banded graph cuts method for fast image segmentation. In *Tenth IEEE International Conference on Computer Vision*, volume 1, pages 259–265, 2005. 2
- [22] M. R. Luetzgen, W. C. Karl, and A. S. Willsky. Efficient multiscale regularization with applications to the computation of optical flow. *IEEE Transactions on Image Processing*, 3(1):41–64, January 1994. 4
- [23] 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. 6
- [24] B. McCane, K. Novins, D. Crannitch, and B. Galvin. On benchmarking optical flow, computer vision and image understanding. *Computer Vision and Image Understanding*, 84(1):126–143, 2001. 7
- [25] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (SIGGRAPH'04)*, volume 23, pages 309–314, 2004. 2
- [26] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1st edition, 1986. 2
- [27] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. In *ACM Transactions on Graphics (SIGGRAPH'05)*, pages 585–594, 2005. 2, 3
- [28] N. Xu, R. Bansal, and N. Ahuja. Object boundary segmentation using graph cuts based active contours. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001. 2
- [29] S. X. Yu. Segmentation using multiscale cues. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 247–254, June 2004. 1
- [30] S. X. Yu. Segmentation induced by scale invariance. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 444–451, June 2005. 1