

Patch-Based Texture Edges and Segmentation

Lior Wolf¹, Xiaolei Huang², Ian Martin¹, and Dimitris Metaxas²

¹ Center for Biological and Computational Learning,
The McGovern Institute for Brain Research and dept. of Brain & Cognitive Sciences,
Massachusetts Institute of Technology, Cambridge, MA

² Division of Computer and Information Sciences,
Rutgers University, New Brunswick, NJ

Abstract. A novel technique for extracting texture edges is introduced. It is based on the combination of two ideas: the patch-based approach, and non-parametric tests of distributions.

Our method can reliably detect texture edges using only local information. Therefore, it can be computed as a preprocessing step prior to segmentation, and can be very easily combined with parametric deformable models. These models furnish our system with smooth boundaries and globally salient structures.

1 Introduction

The detection of image edges has been one of the most explored domains in computer vision. While most of the effort has been aimed at the detection of intensity edges, the study of color edges and the study of texture edges are also well developed fields.

The dominant approach in texture edge analysis is to construct a description of the local neighborhood around each pixel, and then to compare this descriptor to the descriptors of nearby points. This approach is often referred to as “patch-based” since a fragment around each pixel is used in order to compute the outputs of the filters. In this work, however, the term “patch-based” is quite distinguishable from the above: it means that the gray values of the patch are used as-is, and that the basic operation on patches is the comparison of two patches using image correlation measures, such as normalized cross correlation between the gray values, or their Euclidean distance.

What makes this approach novel for texture edge detection is that since texture is a stochastic property, this kind of descriptor would traditionally be considered unfit. In other words, since the gray values of two neighboring patches from the same texture could be very different, most methods rely on more elaborate descriptors. This is in contrast to the dominant trend in current texture synthesis research, where patches of the original texture are stitched together in order to generate a new texture image – a trend that seems to be much more successful than the best descriptor based methods.

The main idea of this work is simple to grasp: if a point lies on the left-hand side of a texture edge, the distribution of similarities of the patch centered at this point to the patches on its left is different from the distribution of similarities to

the patches on its right. Detection of the texture edges can therefore be achieved by examining these differences in the similarity distributions.

As this paper will show, sampling from the distributions of similarities can be done very efficiently. In order to estimate whether the distributions are the same, we use a non-parametric test called the Wilcoxon Mann-Whitney Test [38]. It is similar to the t-test but performs well even for small sample sizes with unknown distributions.

In contrast to intensity edges, which have many uses in computer vision, texture edges have been used primarily for image segmentation. In order to make this work complete, we couple it with a segmentation scheme. Since texture edges are often gapped, we use a hybrid deformable model to capture the image contour. This hybrid deformable model is an adaptation of the general class of Metamorphs Deformable models [19]. This type of deformable model borrows the best features from traditional parametric deformable models [20, 35] and geometric level-set based deformable models [7, 22], and enjoys the advantage of bridging over gaps in contours, topology freedom during evolution, and fast convergence. In particular, the model shape is implicitly represented in a higher dimensional space of distance transforms as a distance map “image”, and model deformations are efficiently parameterized using a space warping technique: the Free Form Deformations (FFD) [1, 3] based on cubic B-splines.

2 Previous Work

Below we discuss traditional texture segmentation approaches, the emerging patch-based techniques, and explain the background for our statistical test.

Feature-Based Texture Edge Detection and Segmentation. Traditional methods for texture analysis are often grouped into three major categories: statistical, structural and spectral. In the statistical approach, texture statistics (e.g., moments of the gray-value histogram, or co-occurrence matrices) serve as texture descriptors. In structural approaches, the structure is analyzed by constructing a set of rules that generates the texture. In spectral approaches, the texture is analyzed in the frequency domain. In contrast to the wealth of approaches suggested in the past, the last decade has been dominated by the filter bank approach, to which we will suggest an alternative.

“There is an emerging consensus that for texture analysis, an image should first be convolved with a bank of filters tuned to various orientations and spatial frequencies.” [11]

Of the many studies that employ banks of filters, the most common set of filters used seems to be the Gabor filters [10, 16, 17, 11, 23, 32]. We would like to specifically mention the work of [32] which, like our work, emphasizes the detection of texture edges, not texture segmentation. In relation to our work, we would also like to point out that non-parametric tests have been used in the past for texture segmentation, [16, 17], where nearby blocks of the image were grouped

together if the distributions of filter outputs in those blocks were not statistically distinguishable. Similar to our work, the statistical distinguishability has been measured using non parametric tests: [16] used the Kolmogorov-Smirnov distance and [17] used the χ^2 statistic.

On a more abstract level, we find relation to the work of [14] in which characteristics of small segments in the image are used as part of the texture description in addition to filter banks. We conjecture that, similar to the move in object recognition from semantic-object-parts to patches at random locations [36], patches from textured areas may be similar in strength to identified sub-segments for texture segmentation.

Patch Based Methods. The filter bank approach was popular in the field of texture synthesis as well (e.g., [15,28]), up until the advent of the patch based methods. In the few years since its publication [9,21], the patch-based method has dominated the field of texture synthesis.

The basic use of the patch for texture synthesis consists of stitching together small overlapping patches of the input texture, such that their boundaries overlap (i.e., the gray value differences at the boundaries are minimal). This results in a new texture image, which seems to match the original texture in appearance, and has similar statistical properties. A similar approach was used for super-resolution [13] and for class-based edge detection [5]. The success of the patch-based methods has been extended to image completion [8] and to image denoising [2]. Patch-based methods were also shown to be extremely successful in object recognition [36].

Non-parametric Statistical Tests. Non-parametric statistical tests are preferred over their parametric counterparts, when certain assumptions about the data cannot be made. For example, the t-test assumes that the difference between the two independent samples it is applied to is normally distributed, while its non-parametric analog, the Wilcoxon Mann-Whitney Test [38], does not.

The WMW Test is one of the most powerful of the non-parametric tests for comparing two samples. It is used to test the null hypothesis that two samples have identical distribution functions against the alternative hypothesis that the two distribution functions differ only with respect to location (median), if at all.

This test has several advantages that make it especially suitable for our application. First, it is valid for data from any distribution and is robust to outliers. Second, it reacts to differences both in the location of the distributions (i.e., to the difference of their median), and to the shape of the distributions. The test is well known, however, since it is uncommon in Computer Vision circles, and in order to keep this paper self-contained, we describe it in Fig. 1.

Deformable Models for Segmentation. Deformable models or Active Contours are curves and surfaces that move toward edges under the influence of internal smoothness forces and external image forces. In traditional deformable models, the external image forces come from image gradient or intensity edge information, which are not reliable guides for texture segmentation. Region Competition [41] performs texture segmentation by combining region growing and active

Given two vectors of samples v_a and v_b , of lengths n_a and n_b we wish to find a measure for the similarity of the underlying distributions.

1. Combine the samples into one vector of length $n_a + n_b$ and sort this vector.
2. Each observation in the combined vector has a rank. The first observation has a rank of 1, the second has a rank of 2, etc.
3. Let w_a be the sum of all of the ranks of elements originating from the vector v_a , and let w_b be a similar sum for v_b .
4. Use the statistic $w = \min(w_a, w_b)$ to determine if the two distributions are different. Very low values of w suggest they are.

Fig. 1. The Wilcoxon Mann-Whitney Test

contours using multi-band input after applying a set of gabor filters. The method assumes multivariate Gaussian distributions on the filter-response vector inputs. Geodesic Active Regions [27] deals with supervised texture segmentation in a frame partition framework using level-set deformable model implementation; the assumptions of the method are that the number of regions in an image are known beforehand and statistics of each region are learned offline. The active unsupervised texture segmentation approach proposed in [31] uses feature channels extracted based on structure tensor and nonlinear diffusion to discriminate different texture regions, the statistics of these features are then incorporated in a level set based deformable model segmentation process to partition the image into a foreground and a background region. Another level-set based algorithm proposed in [33] detects texture edges by applying multiple gabor transforms and an vector valued active contour model; the method supports both supervised and unsupervised forms of the model, although it is limited by the selection of proper gabor filter parameters and the Gaussian assumption on filter responses within each region. Our unsupervised segmentation method overcomes the difficulties faced by these methods by decomposing the problem into the two stages of an initial local texture edge detection and a follow-up segmentation using a hybrid deformable model that smoothly bridges over the missing gaps.

Work Related on an Abstract Level. In this work we detect discontinuities (edges) by comparing distributions to the left and to the right of each point. This idea can be tracked back to [40]. Comparing gray values of adjusted curves was used in [39] in order to classify hand marked edges into the occluding contour or the cast shadow types, in a manner that has faint similarities to our method.

3 Patch Based Texture Edge Detection

Our method is straightforward and is illustrated in Fig. 2(a). In essence, it tests whether a point (x, y) in the image is near a texture edge. Assume a situation where (x, y) is not near a texture edge. Then the similarities between the patch surrounding (x, y) and the nearby patches to its left and right are drawn from the same distribution. In our experiments we measure similarities by simply computing the Euclidean distance between the patch at (x, y) and the nearby

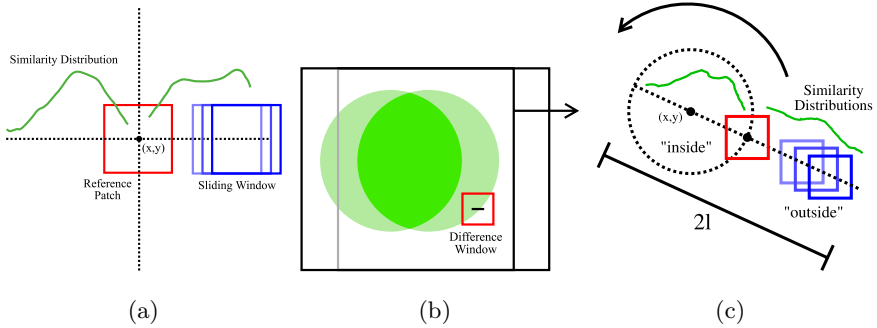


Fig. 2. (a) An illustration of the grid method, in which the patches near the center patch are used in order to compute the similarity distributions. Four distributions are sampled: D_{up} , D_{down} , D_{left} and D_{right} . The pixel at the center would be considered to lie on a texture edge if, according to the Wilcoxon Mann Whitney test, the distribution D_{up} is determined to be different from the distribution D_{down} , or if D_{left} is determined to be different from D_{right} . (b) An illustration of the efficient method to sample the four distributions, using vector operations. To simultaneously sample all of the differences between all of the patches in the image, and all of the patches which are Δx pixels to the right or to the left, a copy of the image is translated Δx pixels left, and then subtracted from the original image. The difference is squared, and then summed at each 5×5 patch in the image, which is a separable operation. (c) An alternative architecture using the flux idea. A pixel would not be on a texture edge if the similarity of points along a circle around it are as likely to be similar to points inside the circle, as they are to points outside the circle. For each point on the circle of radius r , the similarity of the patch around it is compared to patches along the line of length $2l$, which passes through the center point, the point on the circle, and outside of the circle.

patches. Our use of the actual image patch as a template, instead of a predefined filter bank, has the potential to be very sensitive to changes in the local texture.

Let D_{right}, D_{left} be the distributions of similarities between the patch surrounding (x, y) and the nearby patches. If there is a texture edge on the left side of (x, y) , it is natural to expect the distributions D_{right} and D_{left} to be different. For example, it might be reasonable to assume larger similarities within D_{right} .

In order to determine whether the two distributions are the same, we sample patches slightly to the left and to the right of the point (x, y) . In the experiments we used a maximum distance of 15 pixels, and sampled at each pixel, therefore sampling 15 similarities from each distribution.

As mentioned above, we use the Wilcoxon Mann-Whitney Test, which excels for samples small in size, and assumes very little about the nature of the distributions. The horizontal texture edge points are those points for which the test determines that the two distributions D_{right} and D_{left} are different. The same process is then applied vertically, and two similar distributions D_{up} and D_{down} are compared. For our application we combine the two edge directions by taking the minimum value returned from the two tests.

Note, that since measurements from patches as far as 15 pixels away affect the distribution, we can expect the test score to change gradually. Moreover,

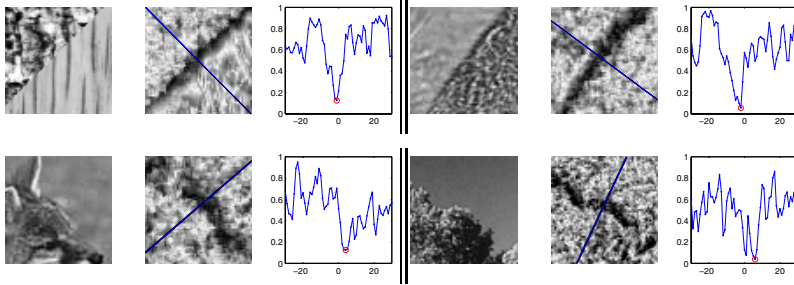


Fig. 3. Profile of the edges we obtain by using our method. Left of each triplet: the original part of the image. Middle: the texture edge we get. Right: the profile as a 2D plot. Note that the profile has a double edge effect, but it is rather minimal.

when (x, y) lies exactly on a texture edge, the patch around it is a hybrid patch, composed of two textures, and we expect the difference between the distributions to be lower exactly at the edge, this could create a double edge. It turns out that for the small patch size we used in the experiments (5×5 pixels), these concerns did not affect the texture edges dramatically. This is demonstrated in Fig. 3 with plots of several edge profiles (See also Fig. 2(c), for an brief illustration of a method developed to solve the double edge problem).

Another important implementation detail is the way ties are handled inside the non-parametric test [29]. While, in general, this question has a critical effect on the results, and should be addressed with caution, exact ties in patch similarity scores obtained from images are rare. An exception is when applying our method to areas where the gray value is exactly fixed. Adding a negligible amount of random noise to the Euclidean distances solves this problem by producing a random order in such cases.

3.1 Efficient Computation

Every pixel in the image contributes to many patches, which are in turn compared with many overlapping patches. A naïve implementation might compute the difference of the same two pixels multiple times. Another important facet of efficiency is that in some programming environments or hardware configurations (e.g., Matlab, designated graphics hardware) vector computations are done more efficiently than the repeated index-by-index computation.

The implementation we suggest is illustrated in Fig. 2(b), and is based on computing all of the patch comparisons to patches at a distance of Δx in either the vertical or horizontal direction at once. In order to do so, one only needs to translate the image Δx pixels in either the horizontal or vertical direction, and subtract the resulting image from the original image. Since we are interested in the Euclidean distance, we square each value in the difference image, and we then sum across all patches in the difference image. Since the summing operation is separable (can be done first horizontally, and then vertically), the procedure can be made extremely efficient.

4 Segmentation Using Hybrid Deformable Models

The detected texture edges can be coupled with a hybrid deformable model that moves in the manner of free form deformations to achieve segmentation over the entire image domain. The Euclidean distance transform is used to implicitly embed an evolving model as the zero level set of a higher dimensional distance function [25]. If we denote the model as \mathcal{M} , and the implicit model representation as a distance map $\Phi_{\mathcal{M}}$, then the shape defines a partition of the image domain: the region enclosed by \mathcal{M} , $[\mathcal{R}_{\mathcal{M}}]$; the background region $[\Omega - \mathcal{R}_{\mathcal{M}}]$; and the model itself, $[\partial\mathcal{R}_{\mathcal{M}}]$, which corresponds to the zero level set. Such a model shape representation provides a feature space in which objective functions that are optimized using a gradient descent method are stable enough to use.

The deformations that a model can undergo are defined using a space warping technique: the Free Form Deformations (FFD) [34]. In essence, FFD deforms an object by manipulating a regular control lattice $F = \{(F_{m,n}^x, F_{m,n}^y)\}$ of $M \times N$ control points overlaid on top of a region Γ in the volumetric embedding space that contains the object (below we use $\Gamma = \{(x, y) | 1 \leq x \leq X, 1 \leq y \leq Y\}$ in the object-centered coordinate system). In the Incremental Free Form Deformations (IFFD) formulation used in [19], the deformation parameters, \mathbf{q} , are the deformations of the control points in both x and y directions:

$$\mathbf{q} = \{(\delta F_{m,n}^x, \delta F_{m,n}^y)\}; (m, n) \in [1, M] \times [1, N].$$

When the control lattice deforms from F^0 to $F = F^0 + \delta F$, the deformed position of a pixel $\mathbf{x} = (x, y)$ is given by $D(\mathbf{q}; \mathbf{x})$, which is defined in terms of a tensor product of Cubic B-spline polynomials:

$$D(\mathbf{q}; \mathbf{x}) = \sum_{k=0}^3 \sum_{l=0}^3 B_k(u) B_l(v) (F_{i+k, j+l}^0 + \delta F_{i+k, j+l}) \quad (1)$$

where $i = \lfloor \frac{x}{X} \cdot (M - 1) \rfloor + 1$, $j = \lfloor \frac{y}{Y} \cdot (N - 1) \rfloor + 1$.

To find texture region boundaries given a simple-shape model initialized around a seed point, the dynamics of the free-form deformable model are derived from edge energy terms. Instead of intensity edges, which fail to separate textured regions, we use the texture edges computed using our patch-based method above. The result of our patch-based filtering is a texture edge map (e.g., Fig. 3, middle columns), on which true edges between different texture regions correspond to low values. Denote the texture edge map as I_t , the boundary data term E_b below encourages model deformations that map the model boundary to the pixel locations with the smallest values on I_t .

$$E_b = \frac{1}{V(\partial\mathcal{R}_{\mathcal{M}})} \iint_{\partial\mathcal{R}_{\mathcal{M}}} (I_t(D(\mathbf{q}; \mathbf{x})))^2 d\mathbf{x} ,$$

where $V(\mathcal{R})$ represents the volume of a region \mathcal{R} . The above boundary term E_b can help the model to converge to the exact edge location where the difference between two neighboring texture patches is maximized. However, it may cause

the model to get stuck in local minima when the model is initialized far-away from the true boundary. To address this problem, we compute a binary edge map by thresholding on the texture edge map I_t . The threshold is computed automatically using Otsu’s method [26]; the method selects the threshold that minimizes the weighted sum of within-group variances on the histogram of the texture edge map. Given the binary edge map, we encode information from this binary edge map by computing its un-signed distance transform. The resulting distance map image is denoted by Φ_e . The data term E_e below aims to minimize the sum-of-squared-differences between the implicit shape representation values both on the model and inside the model and the underlying distance values on Φ_e at corresponding deformed positions.

$$E_e = \frac{1}{V(\mathcal{R})} \iint_{\mathcal{R}} (\Phi_{\mathcal{M}}(\mathbf{x}) - \Phi_e(D(\mathbf{q}; \mathbf{x})))^2 d\mathbf{x},$$

where $\mathcal{R} = \mathcal{R}_{\mathcal{M}} \cup \partial\mathcal{R}_{\mathcal{M}}$. During optimization, when the model is still far-away from the true edges, this term serves as a two-way ballooning force that expands or shrinks the model along the gradient direction of Φ_e . At an edge with small gaps, this term also constrains the model to follow the “geodesic” path (i.e., the shortest smooth path connecting the two open ends of a gap).

Combining the two data terms – the boundary term E_b and the thresholded edge term E_e , the overall energy functional is: $E = E_b + kE_e$, where k is a constant balancing the contributions from the two terms ¹. Both terms are differentiable with respect to the free-form-deformation parameters \mathbf{q} , and a gradient-descent based method is used to derive the model evolution equation for each element \mathbf{q}_i in \mathbf{q} :

$$\frac{\partial E}{\partial \mathbf{q}_i} = \frac{\partial E_b}{\partial \mathbf{q}_i} + k \frac{\partial E_e}{\partial \mathbf{q}_i}, \tag{2}$$

where

$$\frac{\partial E_b}{\partial \mathbf{q}_i} = \frac{1}{V(\partial\mathcal{R}_{\mathcal{M}})} \iint_{\partial\mathcal{R}_{\mathcal{M}}} 2I_t(D(\mathbf{q}; \mathbf{x})) \cdot (\nabla I_t(D(\mathbf{q}; \mathbf{x}))) \cdot \frac{\partial}{\partial \mathbf{q}_i} D(\mathbf{q}; \mathbf{x}) d\mathbf{x}$$

$$\frac{\partial E_e}{\partial \mathbf{q}_i} = \frac{\iint_{\mathcal{R}_{\mathcal{M}} \cup \partial\mathcal{R}_{\mathcal{M}}} 2(\Phi_{\mathcal{M}}(\mathbf{x}) - \Phi_e(D(\mathbf{q}; \mathbf{x}))) \cdot (-\nabla \Phi_e(D(\mathbf{q}; \mathbf{x}))) \cdot \frac{\partial}{\partial \mathbf{q}_i} D(\mathbf{q}; \mathbf{x}) d\mathbf{x}}{V(\mathcal{R}_{\mathcal{M}} \cup \partial\mathcal{R}_{\mathcal{M}})}.$$

In the above formulas, the partial derivatives $\frac{\partial}{\partial \mathbf{q}_i} D(\mathbf{q}; \mathbf{x})$ can be easily derived from the model deformation formula in Eq. 1.

The whole image is processed in the following manner: the first region is segmented by starting a deformable model at the center of the image. Another point well outside the first region is then used to initialize a second model, and a second region is segmented. The process continues until almost all of the points in the image are segmented. In the case where a new region grows into an old region, the two regions are joined together.

¹ We are able to omit an explicit model smoothness term here because of the strong implicit smoothness constraints imposed by FFD.

5 Experiments

Comparing Methods for Texture Edge Detection. The main purpose of these experiments is to demonstrate that Gabor based filter bank methods cannot be used directly in *local* methods of deriving texture edges. Indeed, in [16, 17] a global clustering method was used to combine regions based on the filter bank descriptors; in [32] a method based on anisotropic diffusion in the direction of the global principle direction was suggested; in [14] the filter bank output was integrated along a region and was modified with statistics on the shape of small segments. One can also refer to the text of [32, 14], where the limitations of the local filter bank measurements are discussed.

In Fig. 4, we compare our method, Canny edge detection, and a method based on [17], in which, for each pixel we plot the maximum difference (using the original parameters and distance measure) of the block around it to the nearest four blocks (the results are similar if using Wilcoxon Mann Whitney instead of χ^2). As can be seen, this “alternative” is not doing well. A verification of this can be found in Fig. 4(a) of [32].

Experiments on Texture Mosaics. In Fig. 5, we show examples results on the texture mosaic images of [17], available online at <http://www-dbv.cs.uni-bonn.de/image/mixture.tar.gz>. This data set contains mosaics generated from a set of 86 micro-patterns from the Brodatz album [6].

Real Image Experiments. In Fig. 6 we present experiments on images taken from the first 25 gray level testing images of the *Berkeley Segmentation Dataset*

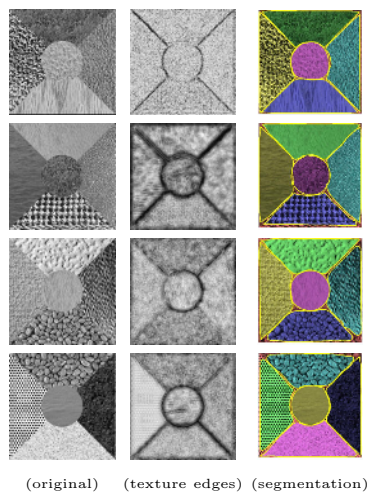
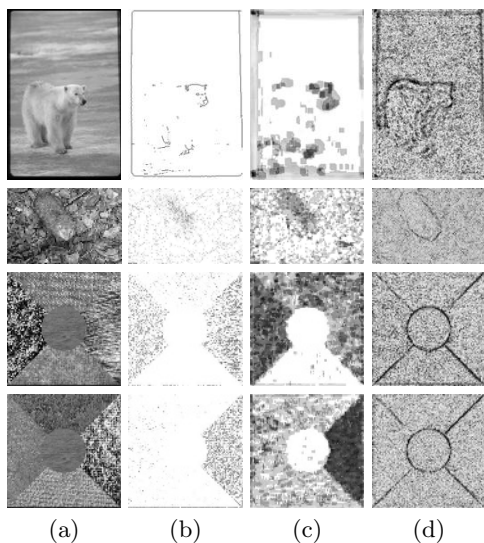


Fig. 4. Comparison of edge detection performed on the original gray images (a), using the Canny edge detector (b), using the filter bank dissimilarity based on [17](c), and using our method (d)

Fig. 5. Results of our edge detection and texture segmentation methods on several mosaics constructed by the authors of [17]

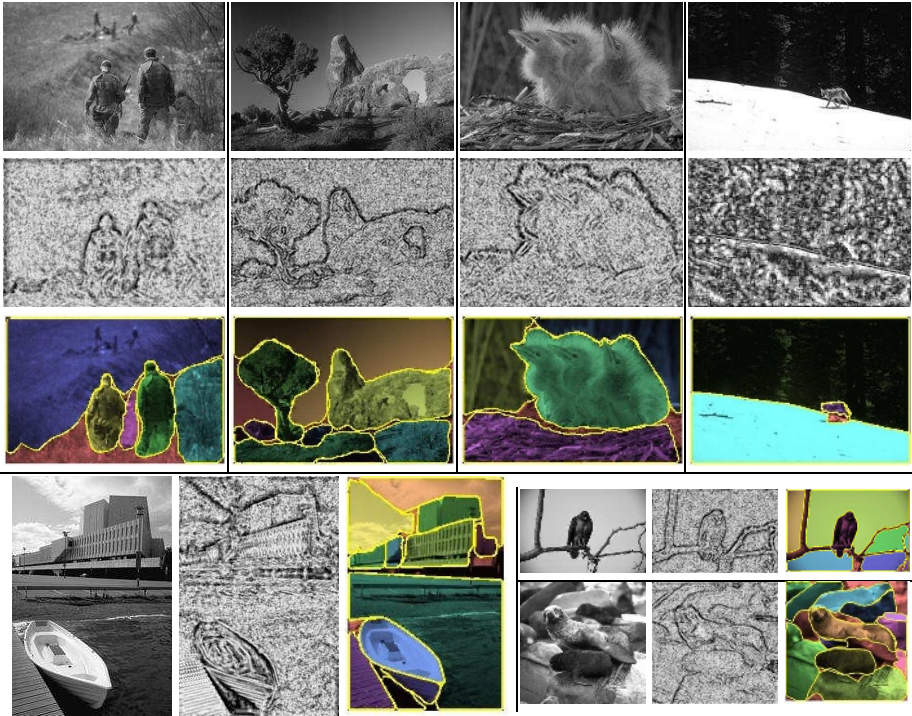


Fig. 6. Results of our edge detection and texture segmentation methods on several real images taken from the *Berkeley Segmentation Dataset*, including the original images, the recovered texture edges, and the resulting segmentation. The images in the lower right demonstrate the detection of texture edges that also constitute intensity edges.

(<http://www.cs.berkeley.edu/projects/vision/grouping/segbench/>). We did not use any dedicated intensity edges method, but as can be seen in the image of the bird, edges between regions of uniform but different intensities, are detected.

6 Summary and Conclusions

The patch based technologies, which are based on local gray value representations and correlations between gray values, have proven to be successful in many computer vision domains, and suggest an appealing alternative to filter bank approaches. While there is no doubt that their recent proliferation is partly due to the increasing computational power available, the representation itself seems inherently powerful. In this work, we use patches in order to compute texture edges. The edge representation (as opposed to a representation of regions using some form of descriptor) is powerful in that it can be readily combined with global optimization based-segmentation (e.g., “snakes”). Most energy-based methods do not deal with texture edges. Attempts that have been made in the

past to incorporate texture into these methods used simple texture descriptors such as mean intensity of a region or the variance of the intensity in that region [27, 30], and were computationally expensive.

By using our patch based texture edge detection technique, combined with Free-Form Deformations, we are able to suggest a tractable solution, which enjoys both rich texture information, and the advantages of a global solution. These advantages include the detection of a smooth boundary, which is globally salient. In this work we focused solely on texture edges, but it had not escaped us that in our framework one can easily add the traditional energy terms for intensity edges and color edges, thus making our framework complete for image segmentation. This completeness was available in the affinity based approaches, but not in the energy based methods.

Acknowledgments

The research was done collaboratively at the Center for Biological & Computational Learning (CBCL), which is in the McGovern Institute for Brain Research at MIT, as well as in the Dept. of Brain & Cognitive Sciences, and which is affiliated with the Computer Sciences & AI Laboratory (CSAIL), and at the Center for Computational Biomedicine, Imaging and Modeling (CBIM) at Rutgers University, which is part of the CS and BME departments. This research was sponsored by CBCL grants from: Office of Naval Research (DARPA) Contracts No. MDA972-04-1-0037 and No. N00014-02-1-0915, National Science Foundation-NIH (CRCNS) Contract No. EIA-0218506, and National Institutes of Health (Conte) Contract No. 1 P20 MH66239-01A1. Additional CBCL support was provided by: Central Research Institute of Electric Power Industry (CRIEPI), Daimler-Chrysler AG, Eastman Kodak Company, Honda Research Institute USA, Inc., Komatsu Ltd., Merrill-Lynch, NEC Fund, Oxygen, Siemens Corporate Research, Inc., Sony, Sumitomo Metal Industries, Toyota Motor Corporation, and the Eugene McDermott Foundation. The research was also supported by CBIM grants NSF 0205671, NSF 0313184, NSF 0428231, AFOSR and ONR.

References

1. A.A. Amini, Y. Chen, M. Elayyadi, and P. Radeva. Tag surface reconstruction and tracking of myocardial beads from SPAMM-MRI with parametric b-spline surfaces. *IEEE Transactions on Medical Imaging*, 20(2), 2001.
2. S. Awate and R. Whitaker, Image Denoising with Unsupervised, Information-Theoretic, Adaptive Filtering. University of Utah TR, UUCS-04-013, 2004.
3. E. Bardinet, L.D. Cohen, and N. Ayache. A parametric deformable model to fit unstructured 3D data. *Computer Vision and Image Understanding*, 71(1), 1998.
4. S. Bouix, K. Siddiqi and A. Tannenbaum. Flux Driven Automatic Centerline Extraction. *Medical Image Analysis*, 2004.
5. E. Borenstein, and S. Ullman. Class-specific, top-down segmentation. *ECCV2002*.
6. P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, '66.
7. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *ICCV'95*.

8. I. Drori, D. Cohen-Or and H. Yeshurun. Fragment-based image completion. SIGGRAPH, 2003.
9. A. Efros, and W.T. Freeman, Image quilting for texture synthesis and transfer. In *SIGGRAPH*, 2001.
10. I. Fogel and D. Sagi. Gabor Filters as Texture Discriminator, Bio.Cybernetics '89.
11. C. Fowlkes, D. Martin, J. Malik. Learning Affinity Functions for Image Segmentation: Combining Patch-based and Gradient-based Approaches. *CVPR*, 2003.
12. W.T. Freeman, E.C. Pasztor, and O.T. Carmichael. Learning low-level vision. *IJCV* 40(1), 2000.
13. W.T. Freeman, T.R. Jones, and E. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 2002.
14. M. Galun, E. Sharon, R. Basri, A. Brandt. Texture Segmentation by Multiscale Aggregation of Filter Responses and Shape Elements. In *ICCV*, 716-723, 2003.
15. D.J. Heeger and J.R. Bergen. Pyramid-based texture analysis/synthesis. SIGGRAPH, 1995.
16. T. Hofmann, J. Puzicha and J.M. Buhmann. Unsupervised Segmentation of Textured Images by Pairwise Data Clustering. *ICIP*, 1996.
17. T. Hofmann, J. Puzicha and J.M. Buhmann. An Optimization Approach to Unsupervised Hierarchical Texture Segmentation. *ICIP*, 1997.
18. T. Hofmann, J. Puzicha and J.M. Buhmann. Unsupervised Texture Segmentation in a Deterministic Annealing Framework PAMI, 1998.
19. X. Huang, D. Metaxas, and T. Chen. Metamorphs: Deformable Shape and Texture Models. *CVPR*, 2004.
20. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1, 1987.
21. L. Liang, C. Liu, Y. Xu, B. Guo, and H.Y. Shum. Real-time texture synthesis by patch-based sampling. MSR-TR-2001-40, Microsoft Research, 2001.
22. R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modelling with front propagation: a level set approach. In *TPAMI*, 17(2):158-175, 1995.
23. J. Malik and P. Perona. Preattentive Texture Discrimination with Early Vision Mechanisms. *J. Optical Soc. Am.*, vol. 7, no. 2, pp. 923- 932, 1990.
24. D. Martin, C. Fowlkes, J. Malik, Learning to Detect Natural Image Boundaries Using Local Brightness, Color and Texture Cues, *TPAMI* 26 (5), 2004.
25. S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed : Algorithms based on the Hamilton-Jacobi formulation. *J. of Comp. Physics*, 1988.
26. N. Otsu. A threshold selection method from gray level histograms. *IEEE. Trans. Systems, Man and Cybernetics*, 9:62-66, 1979.
27. N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *IJCV*, 2002.
28. J. Portilla and E.P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV*, 40(1), 2000.
29. J. C. W. Rayner and D. J. Best, A Contingency Table Approach to Nonparametric Testing. Chapman & Hall/CRC, 2001.
30. M. Rousson and R. Deriche. A variational framework for active and adaptive segmentation of vector valued images. *Motion and Video Computing*, 2002.
31. M. Rousson, T. Brox and R. Deriche. Active Unsupervised Texture Segmentation on a Diffusion Based Feature Space. *Proc. of CVPR*, Vol. 2, pages 699-704, 2003.
32. Y. Rubner and C. Tomasi. Coalescing Texture Descriptors. *Proc. of the ARPA Image Understanding Workshop*, 1996.
33. B. Sandberg, T. Chan and L. Vese. A level-set and Gabor based active contour algorithm for segmenting textured images. *UCLA Math. Report*, 02-39, July 2002.

34. T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Proc. Annual Conference on Computer Graphics*, 1986.
35. L.H. Staib and J.S. Duncan. Boundary finding with parametrically deformable models. *TPAMI*, 14(11), 1992.
36. S. Ullman, E. Sali. Object Classification Using a Fragment-Based Representatio. *BMVC 2000, Proc. Lecture Notes in CS 1811 Springer*, 73-87, 2000.
37. H. Voorhees and T. Poggio. Computing Texture Boundaries from Images. *Nature*, 333, 364-367, 1988.
38. F. Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics* 1, 1945.
39. A. W. Witkin. Intensity-based Edge Classification. *Proc. AAAI-82*, 1982.
40. Y. Leclerc. Computing the Local Structure in Two Dimensions. *Readings in Computer Vision: issues, problems, principles and paradigms*, 1987.
41. S.C. Zhu, A. Yuille. Region Competition: Unifying snakes, region growing and Bayes/MDL for multiband image segmentation. *PAMI*, 18, pages 884-900, 1996.