SHAPE AND DATA DRIVEN TEXTURE SEGMENTATION USING LOCAL BINARY PATTERNS

Erkin Tekeli, Müjdat Çetin, and Aytül Erçil
Faculty of Engineering Sciences, Sabanci University
Sabanci University Orhanlı/Tuzla, 34956, Istanbul, Turkey
phone: + (90) 532 4209519, email: erkintekeli@su.sabanciuniv.edu, mcetin@sabanciuniv.edu, aytulericil@sabanciuniv.edu
web: http://vpa.sabanciuniv.edu/

ABSTRACT
We propose a shape and data driven texture segmentation method using local binary patterns (LBP) and active contours. In particular, we pass textured images through a new LBP-based filter, which produces non-textured images. In this “filtered” domain each textured region of the original image exhibits a characteristic intensity distribution. In this domain we pose the segmentation problem as an optimization problem in a Bayesian framework. The cost functional contains a data-driven term, as well as a term that brings in information about the shapes of the objects to be segmented. We solve the optimization problem using level set-based active contours. Our experimental results on synthetic and real textures demonstrate the effectiveness of our approach in segmenting challenging textures as well as its robustness to missing data and occlusions.

1. INTRODUCTION
Image segmentation has an important role in image processing. It is a fundamental step which can be defined as isolating homogeneous regions within an image or finding the boundaries between such regions. Image segmentation has been approached from a wide variety of perspectives but still it is a challenging problem. One of those perspectives involves active contours [1, 2, 5, 6, 7, 8], which we use in our work as well. In several cases of image segmentation problems; intensity values, color, mean or variance of image intensity distributions and edge information cannot play a discriminative role. In such scenarios, texture might be a good feature to handle the segmentation problem. There are many application areas of image processing in which texture plays an important role. Some important application areas are biomedical image analysis, industrial inspection, analysis of satellite imagery, content-based retrieval from image databases, document analysis, biometric person authentication, scene analysis for robot navigation, texture synthesis for computer graphics and animation, and image coding. The need to deal with textures makes the segmentation problem more challenging.
In recent years there has been increasing interest in incorporating shape information into segmentation problems. While that could in principle improve the segmentation process, effective and practical incorporation of such information still involves many challenges.
In this work, our aim is to combine existing successful active contour and shape based image segmentation methods in texture segmentation problems. Our first contribution is the development of an LBP based texture filter to reduce the texture segmentation problem to an intensity based image segmentation problem to enable the use of existing image segmentation techniques. Our second contribution is the use nonparametric density estimation based shape prior information in texture segmentation problems.
Local Binary Pattern (LBP) is a high discriminative powered texture analysis technique which is an ordered set of binary comparisons of pixel intensities between the center pixel and its neighboring pixels [4]. By using LBP we have developed a texture filter to remove the textural structures of the texture images. We pass the textured image through our filter to produce a “filtered image” where the regions can be discriminated according to structurally independent pixel values. Then we produce a data term for our energy equation which involves maximizing mutual information between the region labels and the image pixel intensities. Also we produce a nonparametric shape prior by estimating underlying shape distribution by using Parzen density estimator in space of shapes. Then we combine the data term and shape prior term within a Bayesian framework to form the energy functional which we minimize using active contours for segmentation. We show the effectiveness and robustness of our approach on synthetic and natural textured images in experimental results section.

2. SHAPE AND DATA DRIVEN TEXTURE SEGMENTATION USING LOCAL BINARY PATTERNS

2.1 LBP Based Texture Filtering
The local binary pattern (LBP) is a non-parametric operator which is used for describing the local spatial structure of an image. Local binary pattern has been first introduced in [4] as a high discriminative powered texture analysis technique. At a given pixel position, LBP is defined as an ordered set of
Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.

Thresholding according to the center pixel value.
we assign \( L_1(H_{\text{train}}, H_{\text{test}}) \) distance value as pixel intensity value of our image in image domain.

\[
G(x) = \| H_{\text{train}} - H_{\text{test}} \|_1 \tag{4}
\]

We apply this procedure to every pixel in the test image in the LBP domain. If the texture in the window is similar to the texture in training image, then this pixel will have a high value. If the texture in the window is not similar to the texture in training image, then this pixel will have a low value. For the textured image in Figure 2(a) the resulting image with \( L_1 \) distance intensity values is in Figure 2(b).

By using window based comparison in LBP domain we take into account the structural information about all the pixels in window. For this reason the \( L_1 \) distance value which we assign as a pixel value is the result of strong information about the structure of the texture. This is the property that makes our filter robust. However, while the windowing process provides robustness; it also results in smoothing in forming the filtered images. Smoothness is a good effect for removing off the image from the textural structure but on the other hand it causes boundary effects. During the windowing process when the windows are shared by two different textured regions smoothing occurs at the boundaries.

Yet, contribution of the windowing process to robustness is more important compared to the boundary effect. Also, the boundary effect can be minimized by several approaches. For instance, at the boundaries using a smaller window size would be a good approach for boundary effect. The size of the windows can be decided according to the homogeneities inside the windows at the boundaries. The homogeneity criterion can be formed by calculating the entropies inside the windows.

### 2.2 Energy Functional

We pose segmentation as an optimization problem in filtered domain. We use the energy functional in equation (4) which combines a data term and a shape prior term under a Bayesian framework.

\[
E(C) = -\log p(\text{data}|C) - \log p_{\nu}(C) \tag{4}
\]

We describe the selection of each of the two terms in this energy functional in the following two subsections.

#### 2.2.1 Data Term

After filtering process, we reduced the texture segmentation problem to a pixel intensity based image segmentation problem, as we have eluded the textured image from the structural properties of the textures. In filtered image we produce our data term by using mutual information based data term in [2]. However, this data term is not the only choice we can use. Because, especially in uniform textured images like in Figure 2(a), the output images of our filter mostly consist of two easily separable regions by intensity distributions. For this reason using other data terms can be suitable for our approach. For instance separation of means approach or Chen and Vese’s approach [1] can be used. However in difficult textures (like nonuniform textures) it is difficult to reach clear filtering results as in uniform textures. When texture filter produces noisy results at difficult textures using separation of means or other approaches does not give satisfactory results. We have used several data terms in our experiments and we have reached good segmentation results, but for generalizing we choose a nonparametric mutual information-based data term which can deal with variety of intensity distributions [2]. We cast the segmentation problem as maximizing of mutual information between region labels and image pixel intensities. The mutual information based data term is as follows:

\[
-\mathbb{I}(G(X);L_C(X)) \tag{5}
\]

Where \( \mathbb{I} \) is the estimated mutual information between intensity values \( G(x) \) and binary label \( L_C(X)) \). \( \mathcal{O} \) is the area of the image domain and \( X \) is the pixel index.

#### 2.2.2 Shape Priors

Shape prior information can aid the segmentation problems involving missing data, occlusions, or low quality images. For the shape term we use the nonparametric shape prior approach in [3]. We produce a shape prior distribution such that given an arbitrary shape; likelihood of this shape can be evaluated. We estimate the underlying shape distribution by using Parzen density estimator in space of shapes. Density estimates are expressed in terms of distances between shapes. The Parzen density estimate in shape space is as follows:

\[
\hat{p}_C(\vec{C}) = \frac{1}{n} \sum_{i=1}^{n} \delta(d_C(\vec{C}, \vec{C}_i), \sigma) \tag{6}
\]

Equation (6) is the Parzen density estimate in space of curves. \( k \) is the number of training shapes. \( \vec{C}_i \) is the aligned candidate curve and \( \vec{C}_i \) are the aligned training curves in the curve space. \( \sigma \) is the Gaussian kernel size. \( d_C \) is the respectively distance metric in space of curves. We have used two distance metrics which are template metric and \( L_2 \) distance between signed distance functions. Template metric can be expressed as \( L_1 \) distance between two binary maps \( I_1 \) and \( I_2 \) whose values are 1 inside the shape and 0 outside the shape, namely:

\[
d_C(\vec{C}, \vec{C}_i) = \int_0^1 \|I_1(x) - I_2(x)\| dx \tag{7}
\]

and \( L_2 \) distance is a norm of difference between two signed distance functions. For shape based texture segmentation we combine the nonparametric shape prior term in equation(6) and the mutual information based data term in equation (5) under a Bayesian framework in Equation (4). We solve this optimization problem through gradient descent using level sets. During each iteration of this process the evolving curve is aligned with the training shapes as well.
3. EXPERIMENTAL RESULTS

In our experiments we have used natural and synthetic textured images for different shapes. We have chosen the synthetic textured images from Brodatz database and we have made our experiments with hand and fighter shapes. We have tried to choose the test images to point the cases when other features are not able to discriminate regions. We have used eleven jet fighter and six hand shapes for training in our experiments. For our shape space we align the shapes to eliminate the variations in shapes due to pose differences. We align our shapes by using the technique in [6]. The training jet fighter shapes before alignment process are in figure 3(a). Our aligned training shapes for jet fighter and hand are shown in figure 3(b)(c).

3.1 Segmentation Results for Uniform Textures:

In figure 4(a) we have two textured regions which are taken from a jumper and from a doormat. These textures have low contrast between each other. Their first order pdfs are close and it is difficult to distinguish these regions by using other features than texture. By using texture as a feature with shape prior information about the hand, we have reached a satisfactory segmentation result in figure 4(c). When we apply the method in [2] without our filtering process the segmentation results are unsatisfactory in figure 4(d). Also for the synthetic textured test image in figure 5(a) we can see the robustness of our approach in figure 5(c). The regions in figure 5(a) are not separable by using intensity distributions. Our texture filter provides output images with high contrast between regions which can be distinguished clearly. Then by the information provided from our mutual information based data term and with shape prior term we can reach good segmentation results as shown in figures 5(c). Since the filtering results of the images in figures 4(a) and 5(a) consist of two clearly separable regions, we have reached very close segmentation results by using separation of means based data term. When the outputs of the filter become noisy mutual information based data term provide more successful results.

3.2 Segmentation Results for Nonuniform Textures:

For nonuniform textures we have used challenging test images which are structurally difficult to analyze. The difficulty with nonuniform textures is, it is probable that nonuniform textures include similar structures with other textures. For this reason texture analysis tools have some problems to clearly identify and discriminate nonuniform textures from other textures. Our texture filter produces some noisy results

---

Kindly view the experimental results on the screen or please take colored print out since the contours may not be visible in black and white print out.
with nonuniform textures but we compensate this fact by our mutual information based data term and shape prior information in our energy equation.

In figure 6(a) we have used two challenging nonuniform textures in the foreground and in the background. Although our texture filter produces noisy filtering results we can reach satisfactory segmentation result as shown in figure 6(c).

3.3 Segmentation Results in Occluded Shapes:

We have occluded the head of the jet fighter in the textured image in figure 7(a). In our segmentation result in figure 7(b) our curve estimates the occluded parts of the fighters according to the information from data and the training shapes. Our algorithm does not have any prior information on the missing data of the shape in the image.

4. CONCLUSION

We have proposed a shape and data driven texture segmentation method using local binary patterns (LBP) and active contours method. We have passed textured images through an LBP-based texture filter to produce non-textured images. In this “filtered” domain we have posed the segmentation problem as an optimization problem in a Bayesian framework. For the cost functional we have used a mutual information based data-driven term, as well as a nonparametric shape prior term that brings in information about the shapes of the objects to be segmented. We have solved the optimization problem through gradient descent using level set-based active contours. Finally we have demonstrated the effectiveness and robustness of our approach to missing data and occlusions in our experiments on challenging textured images.

ACKNOWLEDGEMENTS

This work was partially supported by the European Commission under Grants FP6-2004-ACC-SSA-2 (SPICE) and MIRG-CT-2006-041919. Also we would like to share our gratitude with Mr. Junmo Kim for sharing his implementation codes with us.

REFERENCES


©2007 EURASIP

1446

EURASIP, Poznań 2007