Content Based Retrieval for Remotely Sensed Imagery

Badrinarayan Raghunathan and Scott T. Acton
The Oklahoma Imaging Laboratory
School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, OK 74078
Email: raghuna@fajita.ecen.okstate.edu,sacton@ceat.okstate.edu

Abstract

We present a framework for content based retrieval (CBR) of remotely sensed imagery. The main focus of our research is the segmentation step in CBR. A bank of gabor filters is used to extract regions of homogeneous texture. These filter responses are utilized in a multiscale clustering technique to yield the final segmentation. Novel area morphological filters are utilized for the purpose of scaling. The resultant segmentation yields regions that are homogeneous in terms of texture and are significant in terms of scale. These regions are used for the purpose of extracting shape and textural features (on a global and local basis) that provide important similarity cues in CBR of remotely sensed imagery. In comparison to solutions which use region merging, the segmentation from the texture / scale space does not require heuristic post-processing, nor knowledge of the number of significant regions.

1. INTRODUCTION

To manage and utilize large multimedia databases, content based retrieval (CBR) tools for diverse applications like automated inspection, database management and web based searches [1], [2] have emerged recently. Due to increases in the number of satellites, available bandwidth, and in commercial applications of remote sensing, image databases containing remotely sensed images have been expanding rapidly. Remote sensing data are finding applications in many diverse areas including agriculture, meteorology, geology and urban planning. Traditionally these databases have been accessed by means of metadata which contain keyword descriptors of the database entities. Obvious issues such as access automation and metadata relevancy limit the effectiveness of such keyword based search methods.

In this paper we present a CBR method for remotely sensed imagery based on the efficient extraction of texture and shape features. Our work hinges on an image segmentation technique that exploits both texture and scale. This approach explores the clustering of image positions in a 4-D texture / scale space. We also study an important aspect of matching large-scale texturally homogeneous regions. Finally we present our feature similarity computation strategies and give experimental results involving a prototype database consisting of Landsat TM imagery.

2. TEXTURE ANALYSIS FOR SEGMENTATION

Our CBR approach models each remotely sensed image as a union of segments with various texture and shape properties. A vast literature exists on texture segmentation and classification including gray level co-occurrence matrices, Gabor filters, moments, entropy, Moran autocorrelation functions, fractal methods and principal component analysis. The Gabor filterbank approach provides a versatile model for texture description and optimizes the balance between localization in the spatial and frequency domains [3]. Moreover, Gabor filters are known to mimic the biological perception of texture. The texture analysis presented in this paper utilizes a bank of Gabor filters to create a *texture space*. Within the texture space, we can perform both texture-based feature extraction and segmentation.

A 2-D Gabor function is a complex sinusoidal grating modulated by a 2-D Gaussian function:

$$h(x, y) = g(x', y') \exp[2\pi j(Ux + Vy)],$$

This work was supported by NASA EPSCoR.

$$g(x,y) = \left(\frac{1}{2\pi\lambda\sigma^2}\right)\exp\left[-\frac{\left(x/\lambda^2\right)^2 + y^2}{2\sigma^2}\right]$$

where $(x', y') = (x\cos\phi + y\sin\phi, -x\sin\phi + y\cos\phi)$ are rotated coordinates, and $g(x,y)$ is a 2-D Gaussian function with aspect ratio λ, scale parameter σ, and major axis oriented at an angle ϕ from the x-axis. The pair (U, V) gives the center frequencies of the Gabor filter. Textures at different orientations and frequencies can be extracted by means of applying a number of Gabor filters tuned at different frequencies and orientations. These frequencies and orientations can be specified in terms of the Gabor half-peak frequency and orientation bandwidths [3]. Ideally, several Gabor filters (e.g., 41 filters) would be used to provide coverage and to discriminate between the possible textures. In our experiments we have used a reduced set of g Gabor filters to balance the tradeoffs between computation time and efficient feature extraction, where g ranges from 5 to 17. The texture space is realized by convolving each of the Gabor filters with the input image.

Recently a fuzzy c-means (FCM) based clustering technique was introduced to segment images through a *scale-space* [4]. (A scale-space contains a set of scaled images varying from coarse to fine.) A similar vector FCM based technique could be used here to cluster vectors within the texture space. However, segmentation results based solely on the texture space may contain insignificant or spurious regions in terms of area, which lead to errors and increased computational burden in CBR. Our emphasis within the context of CBR has been on matching large scale regions that are similar in terms of texture. The solution presented here overcomes the difficulties with small scale regions by utilizing a combined texture / scale space. For each texture space layer (each Gabor-convolved result), a set of scaled representations is created via area morphological operations.

Specifically, the area open-close operation [6] is used to remove connected components within the image level sets (thresholded image representations) that do not meet the specified minimum area (where the minimum area depends on the sensor resolution and the particular remote sensing application). For a level set **B**, we can define the area open operation by

$$(x, y) \in \overset{s}{\circ}(\mathbf{B}) \text{ if } |\mathbf{C_B}(x, y)| \geq s,$$

where $|\mathbf{C_B}(x, y)|$ is the cardinality (area) of the connected component at location (x, y), and s is the minimum area. On the other hand, the area open implies that $(x, y) \notin \overset{s}{\circ}(\mathbf{B})$ if $|\mathbf{C_B}((x, y))| < s$. The area close operation is similarly constructed, using instead the Boolean complement of **B**.

For grayscale imagery, each level set is area open-closed independently, and the grayscale result is computed by a stacking operation. Traditionally these operations have been cumbersome and time-consuming due to connected component labeling at each level set. The drawback of computational cost has been overcome by means of a fast algorithm for this area morphological process. Given a fast algorithm for the area open algorithm, we can produce an area close result by employing the Boolean complement of the input level sets used in the area open operation.

In the fast algorithm used here, we have utilized the standard open filter as the starting point. Any connected components within the level sets that (partially) survive the open operation are fully reconstructed. In the morphology literature, the original image has been termed the mask image **I**, the opened image a marker image **M**, and the final product the grayscale reconstruction **R**. Although the fast algorithm improves vastly upon the computational cost required, it is not equivalent to the area open operation. It should be regarded instead as an approximate algorithm. This is because the opening by reconstruction is not equivalent to an area opening, since some connected components that exceed the area criterion may not survive the opening. We now elaborate on the fast algorithm for area open-close.

2.1. Marker image creation using the open filter

In order to ensure that connected components of insufficient area do no survive the filtering process we apply a square or circular structuring element **K** of total size of a. In this case, a is the minimum area parameter. This marker image is used as a precursor to the reconstruction process discussed below.

2.2. Reconstruction by geodesic dilation

The process of reconstruction involves taking a partial connected component (within any level set, obtained from marker image **M**) and reconstructing the entire connected component, based on intensities in the input image **I**. The reconstruction of the connected components is achieved by selectively dilating these components (one pixel at a time). Essentially, if a connected component in the marker image has a neighboring pixel (in 4-connectivity) of lower intensity than that of the input image, then that pixel is increased to the minimum intensity between the dilated image and the input image:

$$R_t(x,y) = \min\left\{\left(\mathbf{R}_{t-1} \oplus K^+\right)(x,y), I(x,y)\right\}$$

where $\mathbf{R}_0 = \mathbf{M}$, and K^+ is a 3x3 cross-shaped structuring element with the origin at the center. Upon complete reconstruction of each of the marked connected components the update in the above equation stabilizes. This requires *T* total updates. This value *T* is bound by the maximum geodesic distance between a boundary pixel of a connected component in **M** and the boundary in the reconstructed connected component in **R**. Further details regarding a formal computational complexity analysis of such fast algorithms for area morphology can be found in [6].

For each position (*x*, *y*) in the input image, each scale *s* and each texture layer *t*, we have intensities *I*(*x*, *y*, *s*, *t*) in the 4-D texture / scale space. These intensities are used to cluster vectors for each position (*x*, *y*) using the FCM algorithm. We can consider a vector **I**(*x*, *y*) as the evolution of the pixel intensity at (*x*, *y*) through scale *s* and texture *t*. The fuzzy clustering technique is based on minimizing an objective functional that quantifies the distance between cluster centers and the data within the various clusters. This objective functional is given by

$$J_m(U,\mu) = \sum_{\Omega} \sum_{i=1}^{C} (u_i(x,y))^m \| d_i(x,y) \|^2$$

Here, *U* is the fuzzy *C* – class partition of the texture/scale-space, where *C* is the number of classes. μ is the set of cluster centers, and Ω is the domain over which the clustering is done, i.e. $(x,y) \in \Omega$. Given a texture / scale space vector **I**(*x*,*y*) at location (*x*, *y*), the measure

$$\| d_i(x,y) \| = \| \mathbf{I}(x,y) - \mu_i \|$$

is the distance between the texture / scale space vector and the *i*th cluster center μ_i. The distance is weighed by the fuzzy membership value (of each texture / scale space vector) $u_i(x,y)$ corresponding to *i*th class. The fuzzy exponent *m* has the range $m \in [1, \infty]$. The objective functional is iteratively minimized. This iteration is subject to the conditions

$$\sum_{\Omega} \sum_{i=1}^{l} u_i(x,y) = 1,$$

$$0 < \sum_{\Omega} u_i(x,y) < |\Omega|$$

and $u_i(x,y) \geq 0$.

At each iteration the fuzzy membership value for each texture / scale-space vector **I**(*x*,y) is computed by

$$u_i(x,y) = 1/[\sum_{\Omega} \sum_{j=1}^{C} \left(\frac{d_i(x,y)}{d_j(x,y)} \right)^{2/(m-1)}]$$

Initially this membership value is computed using a uniformly-distributed random number generator. At each iteration, the cluster center μ_i is updated according to

$$\mu_i = \frac{\sum_{\Omega} (u_i(x,y))^m \mathbf{I}(x,y)}{\sum_{\Omega} (u_i(x,y))^m}$$

The clustering proceeds in this iterative fashion until convergence, where convergence is defined by insignificant changes in the observed objective functional between two consecutive iterations.

The resultant segmentation provides region that are homogeneous in terms of texture and are significant in terms of minimum scale. In contrast to the solutions which use heuristic region merging, the segmentation from texture / scale space does not require post-processing, nor knowledge of the number of significant regions. The motivation for multi-scale, multi-texture clustering has been that it improves classification by clustering members of similar objects more effectively than a fixed scale classifier [4]. An example of the superiority of the texture / scale space approach over the segmentation approach that only utilizes texture is given in the results.

3. GLOBAL AND LOCAL FEATURE EXTRACTION

In addition to segmentation, the texture / scale space is used for texture feature extraction. The types of texture features extracted are the mean, standard deviation and entropy of the Gabor-filtered images. Global and local features are computed. The global features include the global texture features and the number of segments, while the local features are the segment specific texture and shape features. In terms of matching regions of homogeneous texture, it is not sufficient to simply match by texture (granularity and directionality). Our CBR engine also takes segment shape into account. Kauppinen *et al.* [5] have described an experimental comparison of various shape classification techniques and have shown results that characterize Fourier based shape features as having a high degree of accuracy. We have used Fourier based features as well as simple features such as the compactness parameter (*Area/Perimeter*^{2}) for matching segments based on shape.

The Fourier based features are derived from the segment boundary representations. The boundary representations used in our experiments include the curvature, radius, complex coordinate and affine invariant representations [5]. The actual Fourier based features are appropriately weighted versions of the Fourier transform coefficients of the boundary positions [5]. An affine invariant shape feature has also been incorporated with a view of matching remote sensing images of overlapping land areas (w.r.t. a query image) taken at different satellite positions (inferring a change in viewpoints, *viz.* affine transformations). Notice that the circular objects in the original image **I** in Figure 1 are a result of center pivot irrigation systems (CPIS). For example, these circular segments can provide strong shape cues in a CBR query where the user is interested in retrieving images with similarly irrigated fields. Figures 1 and 2 show the sample image **I** and the corresponding scaled Gabor filter responses.

Figure 3 provides an example of the importance of the texture / scale space in segmentation. In Figure 3, a Landsat TM image (band 3) is shown, along with the corresponding segmentations achieved by clustering in the texture space alone and by clustering in the texture / scale space. The resultant segments from the texture / scale space can be used for texture and shape based CBR.

4. FEATURE SIMILARITY COMPUTATION AND QUERY PROCESSING

An important aspect of our research includes the actual computation of feature similarity. Simple matching based on combination of local feature errors can be undesirable in view of one feature error dwarfing another feature error and thus skewing the total error magnitude. Moreover, appropriate weighting of the texture and shape features involved in CBR is still an evolving area of research. We have used a novel *sieve* methodology to combine different features intelligently. The *sieve* method essentially trims the initial search space by using a combination of the global features and a local feature – the compactness parameter. A cutoff is imposed on the matches from the first stage, and then those matches are provided to the second stage search using local texture and Fourier based shape features.

The second stage matches are presented as the final matches. Segmentation and feature extraction are performed off-line in view of the considerable size of the database. The actual query processing is executed on-line. The user specifies a query image while browsing the database library. The query may be specified in the form of a specific image or in the form of a specific segment / region in a specific image. Figures 4 and 5 show some sample CBR queries and results using the texture / scale space approach.

5. CONCLUSIONS

A general texture / shape based framework has been presented for content based retrieval of remotely sensed imagery. A texture based segmentation scheme has been utilized effectively to extract and match large texturally homogeneous regions. Potential applications of the remote sensing CBR engine include precision agriculture, hydrology, meteorology and retrieval of cloud-free imagery for image analysis.

REFERENCES

[1]. Badrinarayan Raghunathan and S.T. Acton, “A Content Based Retrieval Engine for Circuit Board Inspection ,” *Proc. of the IEEE Int. Conf. on Image Processing,* Kobe, Japan, Oct. 25-29, 1999.

[2]. W. Y. Ma and B. S. Manjunath, “NeTra: A Toolbox for Navigating Large Image Databases,” *Proc. of the IEEE Int. Conf. on Image Processing,* Santa Barbara, CA, pp. 568-571, Oct. 26-29, 1997.

[3]. A.C. Bovik, M. Clark, and W.S. Geisler, “Multichannel texture Analysis Using Localized Spatial Filters,” IEEE Trans. PAMI, vol. 12, no. 1, pp. 55-73, Jan. 1990.

[4]. D.P. Mukherjee and S.T. Acton, “Document page segmentation using multiscale clustering,” *Proc. IEEE Int. Conf. On Image Processing,* Kobe, Japan, Oct. 25-29, 1999.

[5]. H. Kauppinen, T. Seppnaaen and M. Pietikainen, “An experimental comparison of autoregressive and Fourier based descriptors in 2D shape classification,” *IEEE Trans. PAMI*, vol. 17, no. 2, pp. 201- 207, 1995.

[6]. S.T. Acton, “Fast Algorithms for Area Morphology,” submitted to *Digital Signal Processing: A Review Journal.*

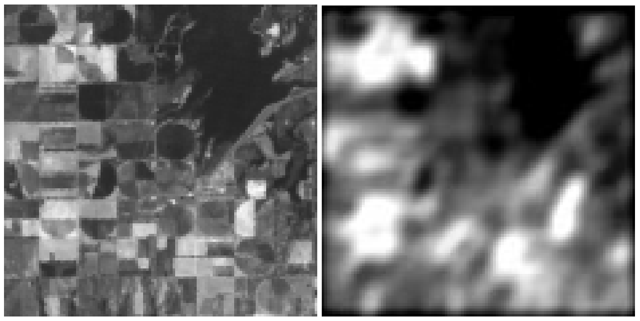
Figure 1. (from left to right) **Original image I and its scaled gabor filter response (area=25)**

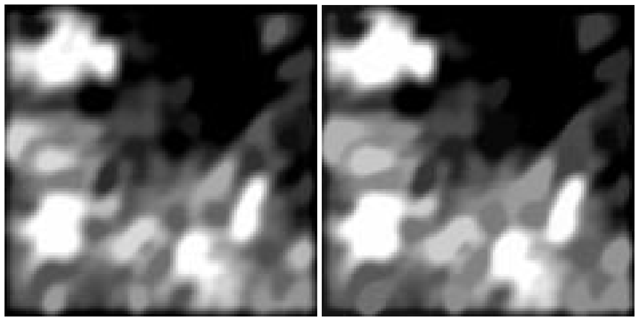

Figure 2. (from left to right) **Scaled gabor filter responses of image I from Fig. 1, Areas=100 and 225**

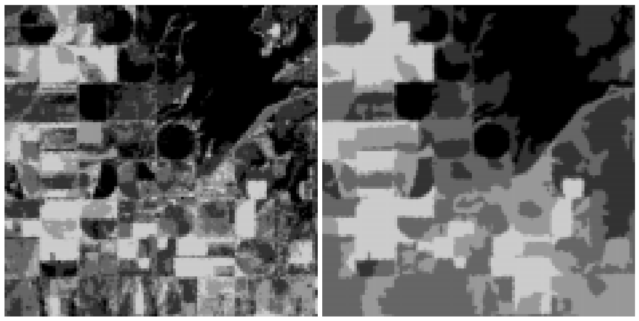

Figure 3. (from left to right) **Texture Space segmentation, Texture / Scale Space segmentation of image from Fig. 1**

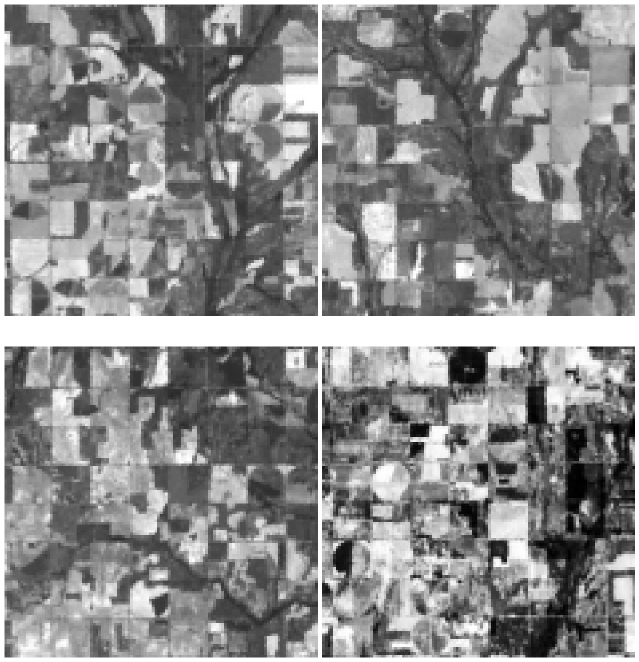

Figure 4. CPIS Query Image, Best matches – in decreasing order of similarity (left to right, top to bottom)

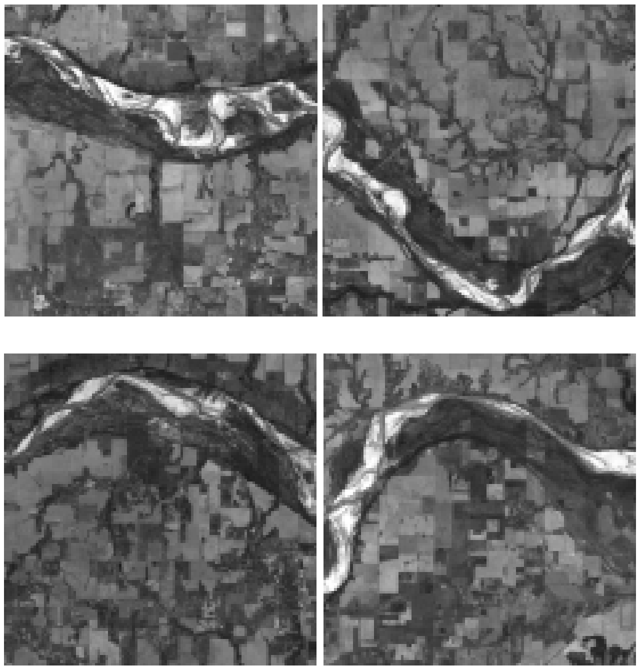

Figure 5. Riverbank query image, Best matches in decreasing order of similarity (left to right, top to bottom)