Image Curvelet Feature Extraction and Matching

Zhibin Lei
Bell Labs, Lucent Technologies
101 Crawfolds Corner Road
Holmdel, NJ 07733

Yin Chan
Dept of Electrical Engineering
Princeton University
Princeton, NJ 08544

Daniel Lopresti
MITL
Panasonic Technologies Inc.
Princeton, NJ 08540

Abstract

In this paper, we introduce a method of extracting and comparing geometric structures within images for content based query-by-sketch image retrievals. These structures are modeled by a set of curvelets, i.e., line segments, circular arcs, and higher degree implicit polynomials (IPs). A similarity computation based on this geometric representation is also introduced. It is more robust and tolerant towards distortions in query sketches. We show experiment results for the image query applications where inaccurate or only partial sketch information is provided for the search.

1 Introduction

There is a rapid growth in interest in pictorial databases and digital libraries recently. To be able to access the vast amount of pictorial information now available, many content-based query and indexing methods have been introduced [1, 2, 3, 4]. Various features such as colors, textures, shape etc. have been used to characterize the content of image data so that efficient searching can be implemented. In this paper, we describe an approach to automatically extract prominent geometric shape structures, i.e., curvelet features (line segments, circular arcs, or high degree curve segments) from images, and use them to compute the similarity values between images so that efficient geometric shape structure-based image retrieval is possible. A prototype, built upon curvelet feature extraction and the Java applet technology, has been introduced in [3] that allows users to draw sketches and pose queries over the WWW.

Our process contains several steps (Figure 1). The edge detection step first identifies perceptually significant edge pixels. Hough transform is then applied to the edge map to relate edge pixels that lie on the same straight line. The line segment extraction step subsequently discard line segments which are too short and thus may be noise or ignored by the user. Finally an higher degree curve extraction phase groups and joins small line segments together and generates a set of prominent curvelet features.

The goal of our salient edge extraction step is quite different from that required by an object recognition application. We make no attempt to group edges and extract contours according to different objects. The reason is that our target database images can contain objects embedded in slightly cluttered background. An object recognition approach may not be realistic in this domain.

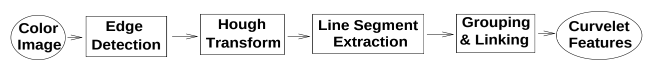

Figure 1: Processing stages for prominent curvelet feature extraction.

2 Curvelet Feature Extraction

Psychophysics studies suggest that in the early stage of human visual system, an edge abstraction process is employed to obtain rich feature structures for later stages of visual processing. Edge pieces alone cannot resolve the complex 3D object organization or scene structure. Grouping, linking or other high level information processings may be necessary in later stages.

We propose here a prominent edge feature extraction approach (Figure 1) that is similar to the human visual system. The edge detection step first identifies perceptually significant edge pixels. Unorganized pixels in the edge map convey little information regarding the structures of objects in the original image. Applying the Hough transform to the binary edge map, we can identify subset of pixels that lie on the same line. The prominent line segment extraction step subsequently discard line segments which are too short and thus may be noise or ignored by the user. Finally an higher degree curve extraction phase groups and joins small line segments together and generates a set of geometric curvelet features.

Curvelet structures are modeled by implicit polynomials, i.e., polynomial function

$$f(x,y) = \sum_{i,j\geq 0, i+j\leq n} a_{ij}x^i y^j = 0$$

whose zero set represents shape [5, 6]. They are simple and concise curve structure representations. The simplest implicit polynomial models are line segments with degree one. The circles (or conics in general) are IPs of degree two.

The model representation error can be approximately written as the function of the IP coefficients directly. A commonly used first order distance approximation is:

$$d(z_i, z(f)) = \frac{\mid f(z_i)\mid}{\parallel \nabla f(z_i)\parallel} \quad (1)$$

Given the IP model, it can be calculated directly for any point in a data set. The average squared distance is:

$$\overline{d}^2 = \frac{1}{N}\sum_{i=1}^{N} d^2(z_i, Z(f)) = \frac{1}{N}\sum_{i=1}^{N}\frac{\mid f(z_i)\mid^2}{\parallel \nabla f(z_i)\parallel^2} \quad (2)$$

Equation 2 is the representation error metric we use for the IP curvelet models.

We use a line segment's tendency or preference of joining with its neighbors as the main factor in the grouping stage. Under the assumption that points on the line segments are actually generated according to a feature curvelet model M_j plus white noise, the probability that edge map $S = \{S_k\}$ is generated by a set of curvelets $M = \{M_j\}$ is:

$$\begin{aligned} P(S|M) &= P(\textstyle\bigcup_k S_k | \bigcup_j M_j) = \Pi_k P(S_k|M_{j_k}) \\ &= \frac{1}{\theta(S,\sigma)} e^{-\Sigma_k \Sigma_i (dist^2(p_i{}^k, M_{j_k}))} \end{aligned}$$

Here σ is the standard deviation of noise model, $dist^2(p_i{}^k, M_j))$ is the distance from a point p_i to the curvelet model M_j, and $\theta(S,\sigma)$ is a scale factor. The net result of the distance errors from all the data points indicates the fitness of the model for the given data set, or how well a set of data points is represented by a given curvelet model M. It is the property of the data set itself, and is not dependent on the relative location or ordering of the points along the line segments. The goal here is to generate a set of features or curvelets that is the *best* or most probable representation of the original edge map.

This is a global optimization problem. To reduce the complexity of searching, we use the local geometric properties of line segments, i.e., a neighborhood criterion. We only search a line segment's neighboring segments for possible merging. A line segment is another's neighbor if they are close both in location and in tangent direction. A greedy searching algorithm is used and the search starts from the most distinctive line segment. After two line segments are merged, their neighborhood structures are modified correspondingly. If no more line segments can be joined to it, a resulting curvelet structure is generated and a new search starts among those line segments left. When one iteration is finished, we have a set of curvelets. We can run another iteration on this set and improve the quality of joining. Figure 2 is an example of painting shape of a plant. It shows first the edge map after the Hough transformation, and then the result for each of the two iterations. Here each curvelet is separated and shaped differently. After each iteration, the total number of curvelets is reduced by about half. We see that after two iterations, the result is very satisfactory.

Figure 3 shows the result of feature extraction algorithm applied to an airplane image. Although there are many small details in the original image, the number of final curvelet set is small and capture the essential geometric structure. From the computational point of view, grouping actually reduces the total number of line segments and hence reduces the number of comparisons in the searching stage. This is very important for large pictorial database applications.

3 Curvelet Feature Matching

The similarity between two images is computed by calculating the similarities between pairs of curvelets from the two images. It is based on their lengths, angles of inclination, IP model coefficients, invariants and other global/local geometric features.

The purpose of salient curvelet feature extraction in previous section is different from that required by a traditional object recognition application. Images come from photos, clip arts, computer graphics and drawings etc. Low level features do not necessarily correspond to physically meaningful objects. To simplify the problem, we make no attempt to group edges and extract curvelet features according to their corresponding sources. This could create a problem if the similarity computation is only applied to exactly matched curvelets between two images. There may not exist such perfectly matched curvelets at all. We propose a method called PIMs (polynomial interpretated measures) or *inexact* partial matching [7] for calculating the similarity between two curvelet sets. We use the algebraic invariants of curvelet IP models. IPs are global geometric representations and they can

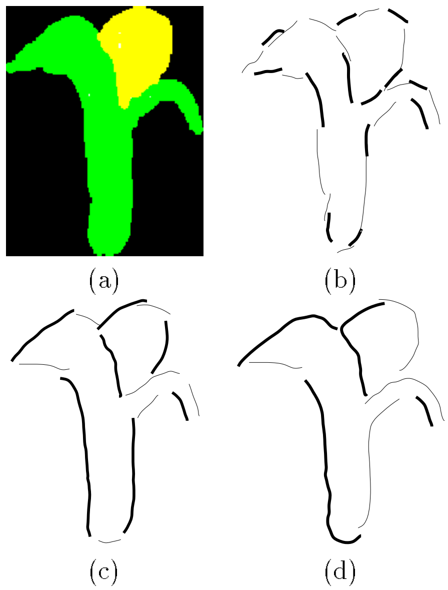

Figure 2: (a) original plant image. (b) edge map after Hough transformation. (c) curvelets after first join iteration. (d) curvelets after second join iteration.

interpolate the missing data. Their invariants capture the useful geometric structure information and are independent of the location and orientation of the curvelets with respect to each other. To compensate for the erroneous or inconsistent grouping or linking exhibited in the curvelets, a windowed Mahalanobis distance is used to account for the partial matching of sub-structures.

Figure 4 shows three situations of partial matching: line-to-line, conic-to-line and conic-to-conic. The overlap of the windows, degrees of curvelets, and the matching sub-structures are combined together to obtain the final similarity value. One important aspect of curvelet feature based matching is that images similar in structures are ranked very close as well, which is a desired property for content-based image retrievals.

Figure 5 is an example of the query result for a bowl shape image using our query-by-sketch program [3] based on this technique. The query sketch is shown with the target image and a few other images having similar structures. They are all within the 10 best results returned from a database of 137 candidate images.

Figures 6 and 7 show results of query-by-sketch application when user cannot sketch the desired shape very well to start the query. It might be because the user can not recall the exact original image structure or the user is not good at sketching shapes. We demonstrate two approaches in dealing with this problem. One approach is to interactively search for a sequence of shapes in the database: start with a shape one can draw and arrive at one the user was originally unable to draw. For example (Figure 6), if the user want an image of a car from a somewhat front and above view, he or she can first draw a car shape of a side view which is easier to sketch, and look for a query return that is rotated more to the front, but may not be rotated as much as desired. Based on one such image found (4th return image in the first row), he or she can sketch a modification which has additional rotation. Another approach is to start with a minimal set of sketch and gradually add more to it by observing the images returned. It is not a good idea to provide too much sketch at the beginning when the user is not sure what exactly he or she is looking for. Inaccurate and erroneous information is misleading to the query system. Our curvelet structure and IP models can support both approaches very easily.

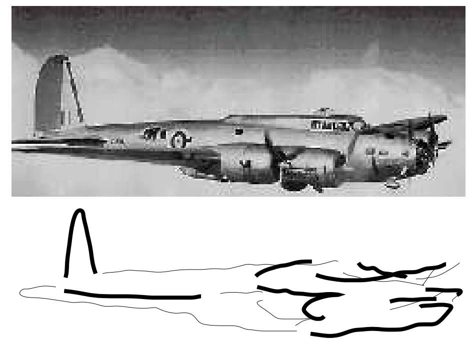

Figure 3: Top: original B-17 aircraft image; Bottom: curvelets after two join iterations.

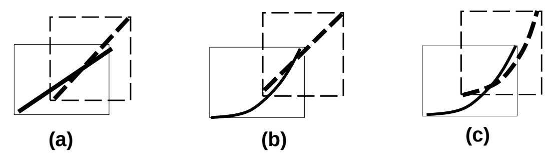

Figure 4: Partial curvelet structure matching: (a)line-to-line; (b) conic-to-line; (c) conic-to-conic.

4 Conclusions and Future Work

We have introduced a method for extracting curvelet like geometric structure from an image and for comparing the similarity of two images based on

Figure 5: An example query sketch and some best matches returned using curvelet features.

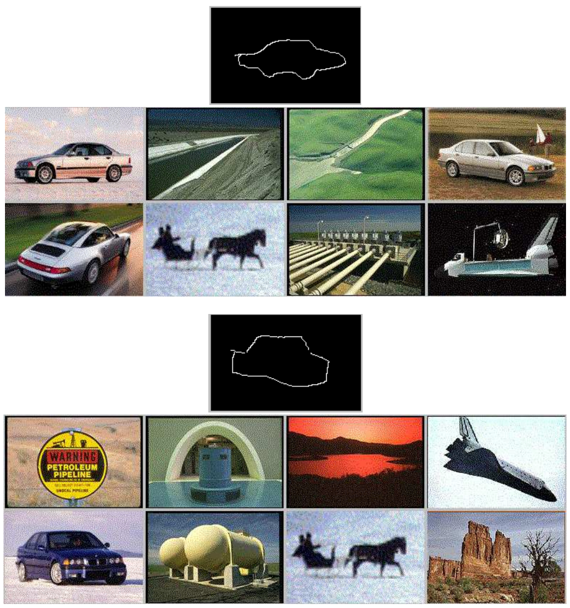

Figure 6: Query by sketch: gradually modify the sketch to arrive at the desired shape.

these curvelet structures. The method is more robust and tolerant towards distortion in the sketches. It is suitable for query-by-sketch kind of application. We attribute this quality to the fact that it uses more structure information to compute the similarity between two images. There are many other issues involved that we have not addressed for large image databases, e.g., data structure and storage, indexing property, and speed of computation, etc.. We are currently extending our method to larger databases and investigating new methods for more robust and efficient curvelet grouping using higher degree models.

References

[1] K. Hirata and T. Kato, "Query by Visual Example", in *Advances in Database Technology EDBT '92, Third International Conference on Extending Database Technology*, March 1992.

[2] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System," in *Computer*, pp. 23–32, September, 1995.

[3] Y. Chan, Z. Lei, D. Lopresti and S.Y. Kung. "A Feature-based Approach for Image Retrieval by Sketch." In submission.

[4] D. Lopresti and A. Tomkins, "Temporal-Domain Matching of Hand-Drawn Pictorial Queries," in K.L. Simner, C.G. Leedham, A.J.W.M. Thomassen, editors, *Handwriting and Drawing Research: Basic and Applied Issues*, IOS Press 1996, pp. 387-401.

[5] Z. Lei, D. Keren and D.B. Cooper. "Computationally Fast Bayesian Recognition of Complex Objects Based on Mutual Algebraic Invariants," in *Proceedings, 1995 IEEE ICIP,* pp. 635-638, October 1995.

[6] Z. Lei, M.M. Blane, and D.B. Cooper. "3L fitting of higher degree implicit polynomials," *Proceedings of Third IEEE Workshop on Applications of Computer Vision,* Sarasota, FL, December 1996.

[7] Z. Lei, T. Tasdizen, and D.B. Cooper. "PIMs and Invariant Parts for Shape Recognition," *Proceedings of International Conference on Computer Vision,* Bombay, India, January 1998.

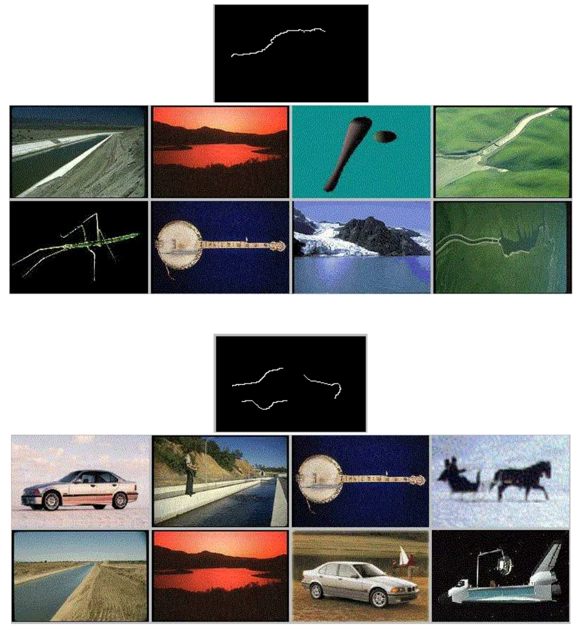

Figure 7: Query by sketch: start with less sketch and add more later.