An Innovative Algorithm for Solving Jigsaw Puzzles Using Geometrical and Color Features

M. Makridis, N. Papamarkos[1], and C. Chamzas

[1] Image Processing and Multimedia Laboratory,
Department of Electrical & Computer Engineering,
Democritus University of Thrace,
67100 Xanthi, Greece
papamark@ee.duth.gr

Abstract. The proposed technique deals with jigsaw puzzles and takes advantage of both geometrical and color features. It is considered that an image is being divided into pieces. The shape of these pieces is not predefined, yet the background's color is. The whole method concerns a recurrent algorithm, which initially, finds the most important corner points around the contour of a piece, afterwards performs color segmentation with a Kohonen's SOFM based technique and finally uses a comparing routine. This routine is based on the corner points found before. It compares a set of angles, the color of the image around the region of the corner points, the color of the contour and finally compares sequences of points by calculating the Euclidean distance of luminance between them. At a final stage the method decides which pieces match. If the result is not satisfying, the algorithm is being repeated with new adaptive modified parameter values as far as the corner points and the color segmentation is concerned.

1 Introduction

The aim of this paper is to provide an automatic method for jigsaw puzzle solving. Automatic solution of jigsaw puzzles by shape alone goes back to 1967 [1]. Since then numerous papers have been written, yet few take advantage of color information. The majority of the proposed techniques works on curve matching. Some of them [11] divide the contour of each piece into partial curves through breakpoint. 2-D boundary curves are represented by shape feature strings which are obtained by a polygonal approximation. The matching stage finds the longest common sub-string and is solved by geometric hashing. In this paper we introduce a few new ideas about how color information and shape matching can go along in solving jigsaw puzzles.

There are many reasons for someone to work on this subject. Related problems include reconstructing archeological artifacts [2]-[6] and or even fitting a protein with known amino acid sequence to a 3D electron density map [7]. However, what is of most interest is that of simulating the human brain. It is very difficult to create an algorithm as effective as human apprehension yet it is very challenging.

In the proposed method, jigsaw puzzle solving algorithm is divided into three main stages. The inputs of the system are images that contain the pieces of the puzzle

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 966 – 976, 2005.

over a background color. In the first stage some basic features that can contribute to the final decision, whether two pieces are similar or not, are being extracted. We use an algorithm for detection of high curvature points, named 'IPAN' [12]. This algorithm finds the most significant points along the contour of an image and calculates the angle with a certain way, which is described below. This is the first feature and the most powerful in our method. The other one has to do with color segmentation. Color segmentation is being accomplished by a Kohonen's SOFM (Self Organized Feature Map) based technique proposed by Papamarkos et al. [8]-[10]. In the second phase we examine all the pieces in pairs. For every pair we examine all its corner points and we decide if two points, one for every piece, are similar according to the color of the neighborhood of each point, the angle and the similarity of their neighborhood's points. The algorithm ends if two pairs of corner points are found. Finally if no matching pairs are found the first two phases are being repeated with properly modified parameters, until all pieces are matched or a maximum number of recursions is reached. This technique has been implemented in Delphi 7 and handles with 24-bit depth color images.

2 Description of the Method

The purpose of the proposed technique is to solve the puzzle problem by combining color and geometrical information. Specifically, the technique takes in a number of color pieces (around 10-15) as inputs to reconstruct the original image. There are two principal assumptions. Firstly we define the color of the background (e.g. white), so we are able to distinguish the background from the foreground and the size of the images should be big enough so as IPAN algorithm can find at least 10 to 15 corner points. While we know the color of the background we can achieve binarization and extract the contour of our pieces. Furthermore, none of the pieces should fit wholly inside the contour of another, because the proposed technique handles only with the external contour of its piece. This means that every piece is concerned as a solid object.

The entire method consists of the following five main stages:

- Corner Detection
- Color Segmentation
- Comparing Routine
- Iteration of all stages above if it is needed
- Matching Routine

2.1 Background and Foreground

In order to extract the contour boundary of an image, which is necessary for corner detection, we should first convert our image in a binary one. Since we know the color of the background we can easily perform binarization. Yet, what happens if some pixels of the foreground have the same value of luminance? These pixels could contain useful color information for our algorithm. Having in mind this case we created a function that detects these pixels and works as follows: i.e. we consider background as white. We put the value 0 to all pixels that have not value 255. Then

we find the contour be separating black and white pixels. Afterwards we read each row of the image from left to right and after 2 pixels of contour are found, we set the following query: If the second pixel has on the left a 'black' pixel, then set all pixels between them black (foreground), else we consider these pixels as two peaks of the contour and leave them unattached.

2.1.1 Corner Detection (Corner Matching)

At this stage, every piece is being approximated with a polygon. The algorithm, which is used, is called IPAN. It is a two-pass algorithm which defines a corner in a simple and intuitively appealing way, as a location where a triangle of specified size and angle can be inscribed in curve. A curve is represented by a sequence of points p_i in the image plane. The ordered points are densely sampled along the curve and no regular spacing between them is assumed. A chain-coded curve can also be handled if converted to a sequence of grid points. The second pass is a post-processing procedure so as to remove superfluous candidates.

First Pass: In each curve point p the detector tries to inscribe in the curve a variable triangle (p^-, p, p^+) that satisfies the set of following rules:

- $d_{\min}^2 \le \left| p - p^+ \right|^2 \le d_{\max}^2$
- $d_{\min}^2 \le \left| p - p^- \right|^2 \le d_{\max}^2$
- $\alpha \le \alpha_{\max}$

(1)

where $| p - p^+ | = | a |$ is the distance between p and p^+, $| p - p^- | = | b |$ the distance between p and p^-, and $\alpha \in (-\pi, \pi)$ the angle of the triangle. The latter is computed as:

$$\alpha = ar\cos\left(\frac{a^2 + b^2 + c^2}{2ab} \right) \tag{2}$$

Variations of the triangle that satisfy the conditions are called admissible. Search for the admissible variations starts from p outwards and stops if any of the conditions (1) is violated. Among the admissible variations, the least opening angle $\alpha(p)$ is selected.

Second Pass: A corner detector can respond to the same corner in a few consecutive points. Similarly to edge detection, a post-processing step is needed to select the strongest response by discarding superfluous points. A candidate point p is discarded if it has a sharper neighbor p_v: $a(p) > a(p_v)$. A candidate point is a valid neighbor of p if $\left| p - p_v \right|^2 \le d_{\max}^2$

Parameters: $d_{\min}, d_{\max}$ and $\alpha_{\max}$ are the parameters of the algorithm and they are very important for the final stage, which will be discussed below.

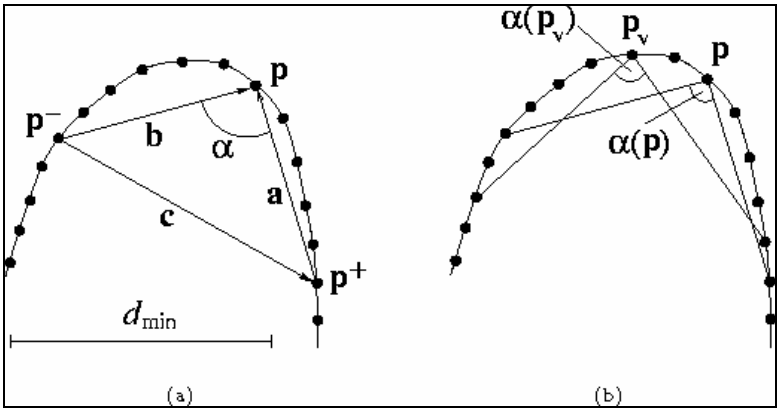


Fig. 1. Detecting High Curvature Points. (a) Determining if p is a candidate point. (b) Testing p in terms of his neighbors.

The parameters of the algorithm contribute to the flexibility of the algorithm by giving us each time a different number of corner points according to their value. Moreover, the angle of every candidate point p is being calculated toward its neighborhood's points and therefore, IPAN is rotationally and scale invariant. Finally, we compute the metacenter of each corner point's inscribed triangle. If metacenter is pixel of background, then the angle is concave, otherwise it is convex. The results of this method are shown in figure 2.

2.1.2 Color Segmentation (Color Similarity)

In the second stage of the technique, a Kohonen's SOFM based color reduction method proposed by Papamarkos et al. [8]-[10] is applied. The metric that uses Kohonen SOFM is the Euclidian color distance. It calculates, therefore, the distance for R, B and G layers of every pixel in order to conclude in which class is this pixel nearer. In order better results to be achieved the number of classes is not constant. It varies between 10 and 30 so as our technique to be effective but not time consuming. Pieces with many similarities, in terms of color information, cannot be distinguished, unless a large number of classes is used. On the other hand, it would be time consuming to apply 30 classes to well distinguished pieces, when the same result can be achieved with only 10.

A problem that we have to deal with here is that only pixels of each piece should be processed and not those of the background. So, we separate the background from the foreground. Samples for the algorithm are taken through each piece, in order to perform unified classes for all the pieces. Although the original color segmentation technique would consider a constant number of randomly taken samples, this would be a drawback for the proposed paper, because each application of the technique would result in different results. Thus, the number of the samples is a percentage of the total number of pixels and moreover we created a function that takes into account specified pixels of every piece, if they contain useful information, which means they are pixels of the foreground. The results are shown in figure 3.

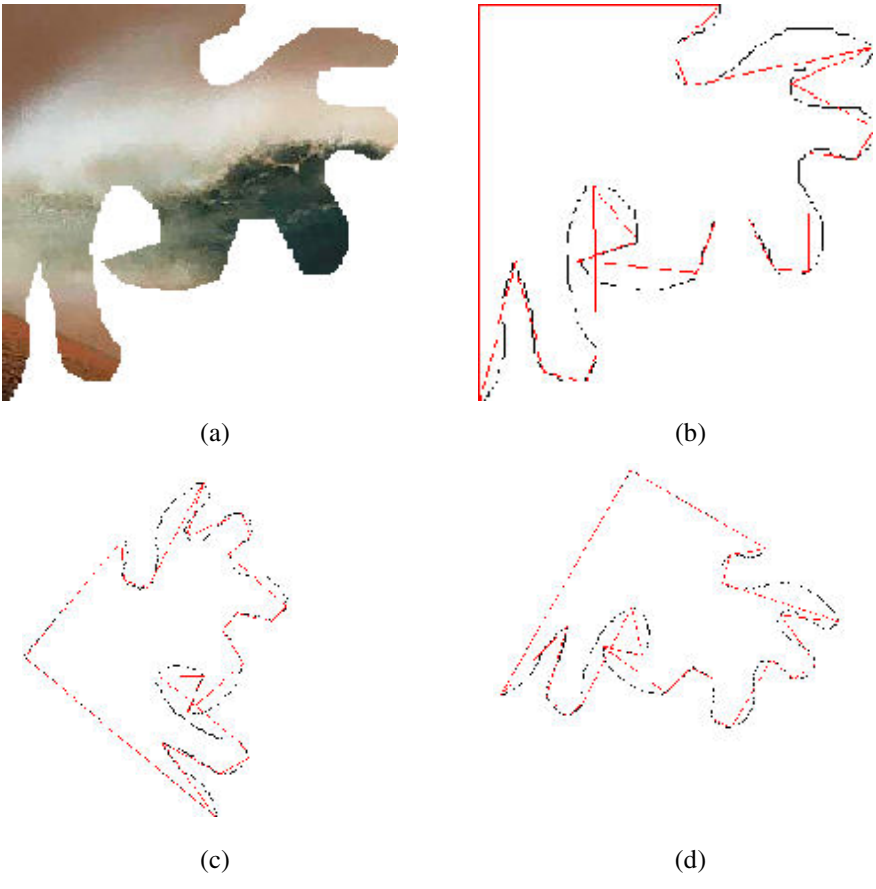


Fig. 2. Polygonal Approximation with a rotationally invariant technique called IPAN. (a) An image piece. (b) Corner points extracted with IPAN. (c) and (d) Ipan extracts certain corner points in the image through its rotation invariance.

2.1.3 Comparing Stage

At this stage we compare couples of pieces and determine whether they are matching or not. From the first stage we concluded to a number of important characteristic points along the contour boundary. We assume that each corner point of the first image corresponds to all corner points of the second. During the comparing stage we discard all corner points that fail to be one with comparing rules. The purpose of this stage is to minimize this number in order to choose 4 points, if it is possible. Having two points from each piece that correspond to precisely two points from the second one, it is easy for us to compute the rotation angle and the shift that is needed for the two pieces to match.

First of all we compare the angle (in degrees) of every point $\alpha(p_{1i})$ of the first image with the angle of every point $\alpha(p_{2i})$ of the second image. If

(a) (b)

Fig. 3. Color Segmentation with Kohonen SOFM (a) Original Image (b) Color Segmentation with vector's dimension equals to 3

$|\alpha(p_{1i}) - \alpha(p_{2i})| > 10$, then the corresponding points will be discarded and will not be examined any more.

Then we compare all points left as far as the luminance is concerned. All points with different luminance, after the application of Kohonen's algorithm, will be discarded as well.

The third criterion is a function F that compares the Euclidean distance of luminance between a connected sequence of points from the first image and another from the second one. If $\{c_{1,i-5},...,c_{1,i-1},p_{1,i},c_{1,i+1},...,c_{1,i+5}\}$ is a sequence around $p_{1,i}$ of the first image and $\{c_{2,j-10},...,c_{2,j-1},p_{2,j},c_{2,j+1},...,c_{2,j+10}\}$ another sequence around $p_{2,j}$ of the other image, where c, pixels of the contour. Then:

$$F = \min\left(\frac{(p_{1i} - p_{2j}) + \sum_{l=-5}^{5} c_{1,i+l} - c_{2,j+l+k}}{11}\right) \text{ for } k = -5,...,5 \tag{3}$$

After finding F, every point $p_{1,i}$ corresponds to 3 at most points $p_{2,j}$, those that minimize function F. All the other points are being discarded.

Furthermore we compare $(p_{1,i-1}, p_{1,i}, p_{1,i+1})$ with $(p_{2,j-1}, p_{2,j}, p_{2,j+1})$ as far as their angles ($|\alpha(p_{1,i}) - \alpha(p_{2,j})| < 10$) and the values of luminance is concerned.

At this point, we introduce two sets of points from both images, each one having five elements as shown in figure 4. Every point p_i of the first set corresponds to another point p_j of the second set. To minimize the chances for a mismatch we test them once more geometrically. As it is shown in figure 4, we compare angle a1 with a2 and part d1 with d3 as well as d2 with d4. All the possible combinations of angles for 5 points are 12. If 10 out of 12 are very similar we conclude that piece *I* and piece *J* match, and the algorithm ends. Otherwise we continue with all the possible combinations of 5-point sequences and in case we run out of combinations we go forward to the next and final stage.

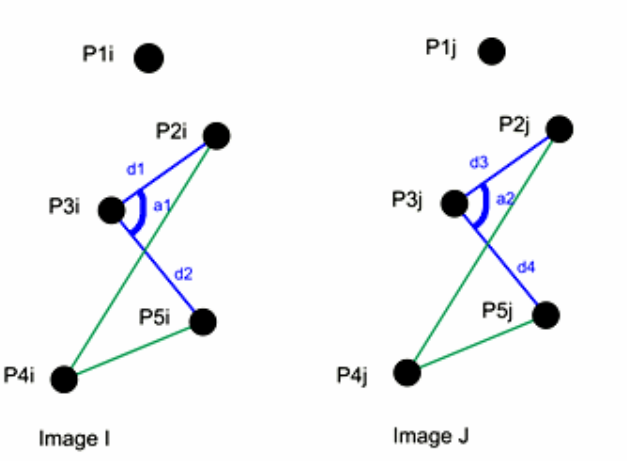


Fig. 4. Comparison of two 5-point sets with each other

2.1.4 Recursion Stage

At this stage what is of most importance are the parameters of stages 1 and 2. If we decrease d_{min} , d_{max} and α_{max} , IPAN algorithm takes out a large number of corner points and our algorithm falls short of speed, yet it can cope with low-analysis images where the search for important corner points should be meticulous in order to be effective. On the other hand, if we increase the values of the parameters we decrease the number of corner points and the algorithm is solving very fast puzzles with high-analysis images. So, the initial values of the parameters are high and if none solution has found the whole method is being repeated automatically with lower values after each recursion. The initial set of parameters is d_{min} =10, d_{max} =12 and α_{max} =160. Additionally Kohonen SOFM is very reliable with 30 classes but faster with 10. The initial value for the Kohonen SOFM is 10. When many recursions are needed is sometimes time-consuming but generally the algorithm is very flexible, as it can cope with various different shaped images. The values of the parameters have been chosen by trial and error. Once our goal is achieved, this stage stops.

2.1.5 Matching Routine

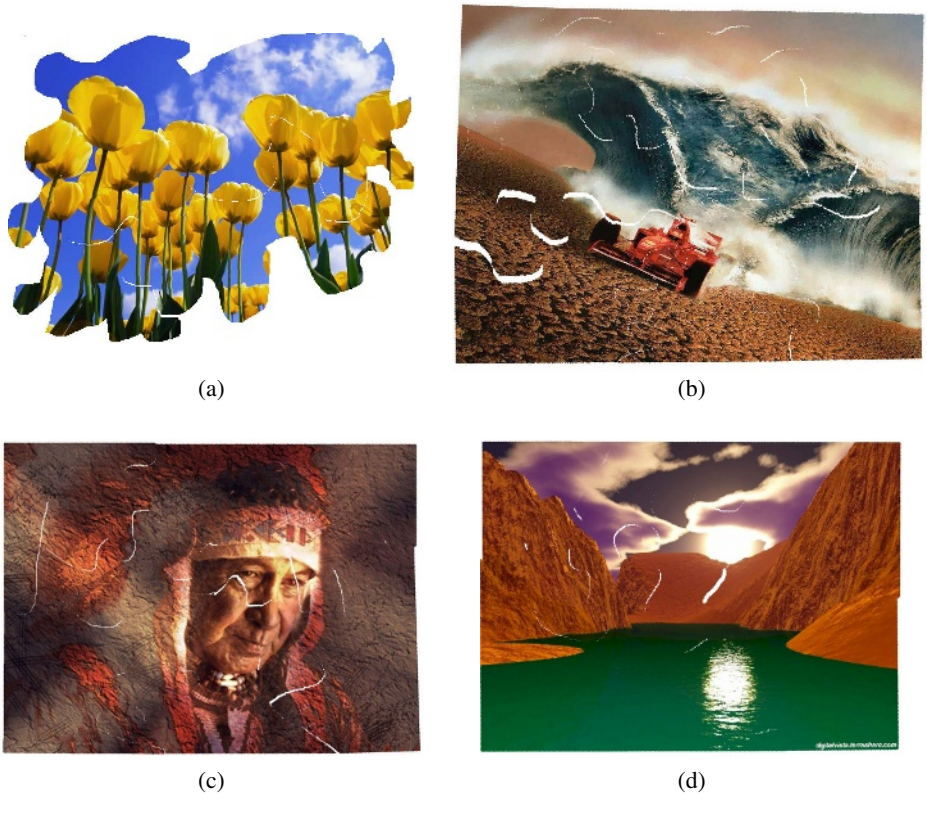
Since has been decided which images are matching together, what is left to do, is to merge the images and show up the results. The rotation and shift phase for all images is being done in terms of one image, which is called reference image. Firstly, if two images fit at points $p_{1,1}$ *and* $p_{1,2}$ and $p_{2,1}$ *and* $p_{2,2}$ respectively, shift and rotation angle are being calculated from the following equations:

$$\begin{aligned} &Shift: && CoordinateX: X_{Shift} = X_{p1,1} - X_{p2,1} \\ & && CoordinateY: Y_{Shift} = Y_{p1,1} - Y_{p2,1} \\ &Rotation: && R_{angle} = \arctan\left(\frac{Y_{p1,1} - Y_{p1,2}}{X_{p1,1} - X_{p1,2}}\right) - \arctan\left(\frac{Y_{p2,1} - Y_{p2,2}}{X_{p2,1} - X_{p2,2}}\right) \end{aligned} \tag{4}$$

However, it is possible that some images don't have common points with the reference image. That is why we created a routine that relates all the images together. So, if for example image 2 fits with the reference image since it has been rotated by angle 'a1' around center point 'c1' and it has been shifted for 'd1' pixels and image 3 fits with the image 2 since it has been rotated by angle 'a2' around center point 'c2' and it has been shifted for 'd2' pixels, we consider that image 3 should be rotated by 'a1' around 'c1' plus 'a2' around 'c2' and it should be shifted by 'd1+d2' so as to fit with the reference image. The whole procedure is being repeated for each image, until all images have been related with the reference one.

3 Experimental Results

The proposed technique for solving jigsaw puzzles, has been implemented in Delphi 7 and tested many images, after they have been cut from 3 to 10 pieces each. The success rate for those images with pieces that were not rotated reaches 90% and 80% for the others. Figure 6 shows the pieces of figure 5(g) image before the application of the technique.



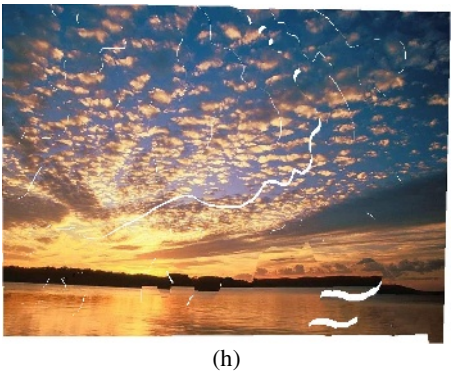
(a) (b)

(c) (d)

(e)

(f)

(g)



(h)

(i)

(j)

Fig. 5. Some experimental results. (a) Image reconstruction out of 6 pieces. (b) Image reconstruction out of 8 pieces (c) Image reconstruction out of 6 pieces. (d) Image reconstruction out of 5 pieces (e) Image reconstruction out of 5 pieces. (f) Image reconstruction out of 7 pieces (g) Image reconstruction out of 15 pieces. (h) Image reconstruction out of 9 pieces (i) Image reconstruction out of 8 pieces. (j) Image reconstruction out of 7 pieces.

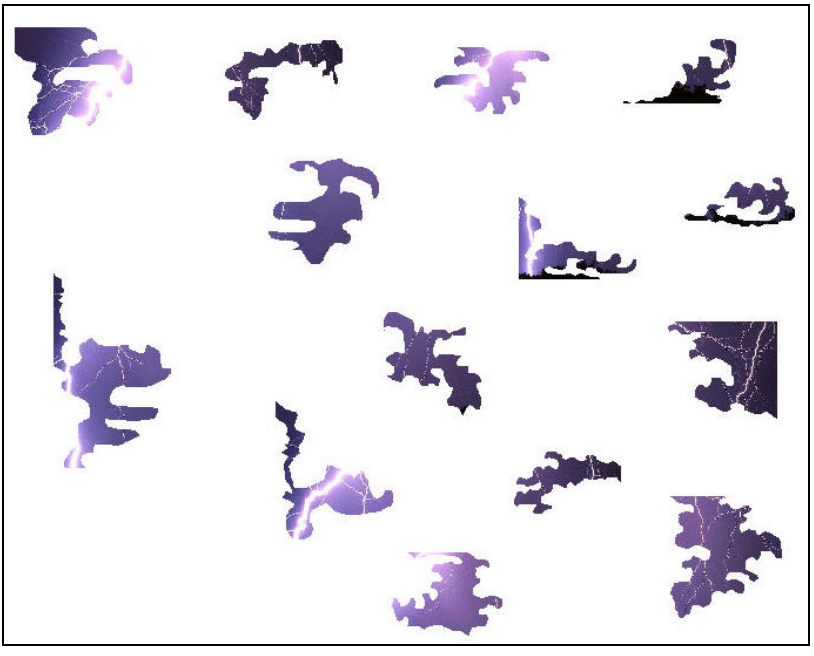


Fig. 6. The pieces of example g in figure 5 before the application of the algorithm

References

[1] H. Freeman and L. Gardner. Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. IEEE Trans. on Electronic Computers 13 (1964) 118–127.

[2] W. Kong and B.B. Kimia. On solving 2D and 3D puzzles using curve matching. Proc. IEEE Computer Vision and Pattern Recognition, 2001.

[3] H. C. da Gama Leitao and J. Stolfi. Automatic reassembly of irregular fragments. Tech. Report IC-98-06, Univ. of Campinas, 1998.

[4] H. C. da Gama Leitao and J. Stolfi. Information Contents of Fracture Lines. Tech. Report IC-99-24, Univ. of Campinas, 1999.

[5] K. Leutwyle. Solving a digital jigsaw puzzle. http://www.sciam.com/explorations/2001/062501fresco/

[6] M. Levoy. The digital Forma Urbis Romae project. http://www.graphics.stanford.edu/projects/forma-urbis/

[7] C. Wang. Determining the Molecular Conformation from Distance or Density Data. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, 2000.

[8] N. Papamarkos, A. Atsalakis and C. Strouthopoulos, "Adaptive Color Reduction", *IEEE Trans. on Systems, Man, and Cybernetics-Part B*, Vol. 32, No. 1, pp. 44-56, Feb. 2002.

[9] N. Papamarkos, "Color reduction using local features and a SOFM neural network", *Int. Journal of Imaging Systems and Technology*, Vol. 10, No 5, pp. 404-409, 1999.

[10] A. Atsalakis, N. Papamarkos , N. Kroupis , D. Soudris and A. Thanailakis, "A Color Quantization Technique Based On Image Decomposition and Its Hardware Implementation", IEE Proceedings Vision, Image and Signal Processing, Vol. 151, Issue 6, pp. 511-524, 2004.

[11] H. Wolfson. On curve matching. PAMI, 12:483–489, 1990.
[12] D. Chetverikov and Zs. Szabo. A Simple and E_cient Algorithm for Detection of High Curvature Points in Planar Curves. In M. Vincze, editor, Robust Vision for Industrial Applications 1999, volume 128 of Schriftenreihe der Osterreihicshen Computer Gesellschaft, pages 175,184, Steyr, Austria, 1999. Osterreihicshe Computer Gesellschaft.