International Journal of Computer Applications (0975 – 8887)
Volume 19– No.5, April 2011

Reassembly of 2D Fragments in Image Reconstruction

Ravi Saharan
Department of CSE
Central University of Rajasthan, Kishangarh, Ajmer

Choudhary Vijaypal Singh
Department of CSE
BMIT, Jaipur

ABSTRACT

Digital images may be considered as collection of pixels. If a single image is divided into more than one part, then these subparts are treated as fragments for an image.Joining of 2D fragments of an in image means we have to reassemble these image's fragments. The joining of fragments to reconstruct images and objects is a problem associated in several applications, like archeology, medicine, art restoration, and forensics.We mainly focused on 2D Image Reconstruction by joining two 2D fragments. This approach is based on the information generated from the boundary and from the color contents of the two fragments. Local curvature is calculated to obtain transformation independent coordinates. Based on this information comparison is done to obtain maximum matching parts among fragments. Finally longest matching parts can be joined to obtain single image.

Keywords—Archeology, Boundarypixels, Color contents, Curvature, Digital images, Fragments.

1. INTRODUCTION

The joining of fragments to reconstruct images and objects is a problem occurring in a number of fields like in archeology[1], art restoration, forensics, computer-aided design, chemistry, and medicine.Manually solving such puzzles may take very long time.Fragments can be analyzed in terms of several properties like color, texture, material, etc. The classification is helpful by reducing the number of fragment pairs that need to be compared manually. If we have very large collection of plain fragments, a procedure is required that can automatically check similarity between fragments based on their shapes so that fragments can be joined accordingly which is the problem that we concentrate in this paper.Also similar problem can be solved by applying techniques for automatic assembly of jigsaw puzzles [2] but we can't directly apply because it requires particular shape of jigsaw puzzle fragments.

2D Image Fragments joiningis our prime attention.

In this paper we present anapproach for joining of fragments to reconstruct images.

We mainly focused on 2D Image Reconstruction by joining several 2D fragments.

Main outline of this approach for fragments [3] are as follows.

(1) Analysis of the two fragments and calculates relative information. Fragment is represented by the sequence of its border pixels with associated information about coordinates, color. The outlines of the fragments are represented by curvature value, namely by a circular sequence of samples, where each sample is the local curvature of the outline at the corresponding point.

(2) Comparison of the two fragments to identify the common portions. The common portions represent the candidate matching portions of the fragments.

(3) Apply transformation in order to join first fragment with second one along the longest matching portion of the fragments.

This paper is constructed as follows.

In section 2 Overview for fragments description of a 2D image which includes flood fill algorithm and generation of local curvature.

Section 3 shows approach for joining of image fragments.

Section 4 describes experimental scenario & results.

Section 5highlights the conclusion derived from the work.

Section 6 pointers for future work.

2. OVERVIEW OF FRAGMENT DESCRIPTION

In this section we describe about fragment analysis. The input to an image which has been partitioned into several part or fragments as 2D objects which can be defined by their closed boundary values. Object recognition [4] is one of curve matching algorithm's applications. The analysis is based on information which is obtained from all sample points on the curve rather than on information obtained from some special points which may or may not exist. Given the two image's fragments, first calculate boundary pixel in order to separate fragments from the background pixel. For this flood fill algorithm[5] is used that separates every pixel either foreground pixel(fragment) or background pixel. The pixels that compose the boundary of a fragment are then located as those belonging to fragment that are adjacent to a background pixel.

We calculate the boundary of the fragment by moving from one boundary pixel to another. The next boundary pixel is located by searching among the eight neighbors of the current boundary pixel. The result is stored in a boundary array containing the coordinated and the color of every pixel belonging to the fragment's boundary. This algorithm is based on conversion into numerical strings i.e. arrays. Curve matching complexity is in time O(n),n being the number of sample points on the curves.

2.1 Flood Fill Algorithm

Floodfill algorithm is a basic algorithm in computer graphics, which is used to connectneighboring and related elements of an array. The floodfill algorithm is commonly usedin paint programs such as Adobe Photoshop and CorelPaintshop, in computer puzzle games, in solving mazes etc. They establish the area connected to a given node in a multi-dimensional array. Two elements are defined as connected if a path exists between them along which the value of all elements exceeds some threshold for a given node and threshold. The flood fill is performed recursively on all elements of connected node. In this method flood fill algorithm is used for detection background pixels and boundary pixels.

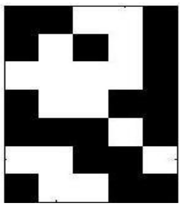


Fig 1: Original Image

2.1.1 Floodfill Algo Implementation

Floodfill algorithm can be implemented in various ways. Here a very simple recursive implementation is used. Along with the surface being explored, two parameters are input into the flood fill algorithm: a threshold and a start node (whose value exceeds the threshold). The algorithm finds all points that are connected to the start node by a path along which all values exceed the threshold. This group of connected points is referred to as a stack. The start node seeds the stack (it is the first point added to the stack). All points neighboring the start node are then checked. The neighborhood includes 8 points i.e. above, below, to the left and right of the node and the 4 corner points (this is referred to as an 8 point flood fill in graphics). Each neighbor whose value exceeds the threshold is added to the stack and a flood fill algorithm is applied with it as the start node. The procedure continues until all connected points have been added to the stack.

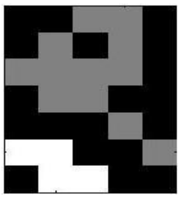


Fig2: When All Points Have Been Added to Stack

2. 2 Generation of Curvature

Generating of boundary value is rotation and translation dependent, to overcome this problem we use the local curvature [6] which can be treated as one to one mapping between a regular curve and its curvature function. The curvature function is the derivative of the tangent angle to the curve, parameterized as a function of its arc length. The local curvature of the boundary in a pixel pis calculated as the magnitude of the derivative of the local tangent in p. The local curvature of the border pixels of the first and the second fragment are stored in real-values arrays, respectively. Since the number of pixels in two segments with the same physical length is different if the segments are not parallel, we use an adjusting algorithm that stretches the border and curvature arrays adding interpolated pixels depending on the slope of the local tangent.

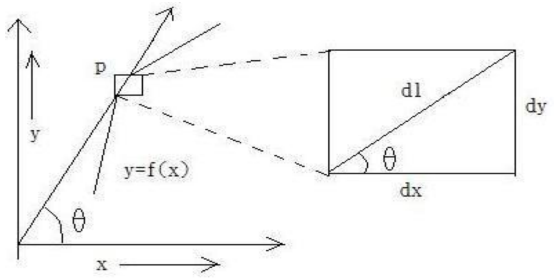


Fig3: Illustration of Curvature

Curvature for an image pixel is one of the important property by which one can detect features characterized by an edge.

The detected features which are used by a higher level system for further processing and understanding of the scene are of two types (1) perceptual features such as edges, corners, contours, boundaries etc., and (2) topographic features such as ridges, valleys, watersheds etc. We are dealing with perceptual features in which edges, corners, boundaries are covered via curvature. It has been used to detect value of derivative of tangent at every pixel. Curvature can be used to detect features where the image surface bends sharply. There exist two curvature measures, namely, the maximum and minimum principal curvatures along two orthogonal principal directions which measure the bend in the image surface and are typically used in order to get translation and rotation independent values of a digital images. A pixel is defined as a ridge pixel if the magnitude of the maximum principal curvature (MPC) at that pixel is a local maximum in some direction. Depending on the reference coordinate system, a high negative curvature indicates strong ridge strength while a high positive curvature indicates a strong valley strength, or vice versa.

The direction along which the MPC is a maximum is the direction perpendicular to the orientation of the ridge (or valley) at that pixel. Curvature-based feature detection approach can be described as follows. The algorithm involves following steps:

Fit a image surface I(m,n) to the neighborhood of the point of interest (a local graph representation); compute the first and second partial derivatives of the image function; determine the principle curvatures of the surface; and finally evaluate curvature measures to find desired features. We define a curvature measure, called the Surface Tangent Derivative (STD), as an estimate of the curvature of image surfaces. STD tends itself for an efficient implementation. This ability makes it a superior measure than the standard curvature for use in real time feature detection applications.

3. AN APPROACH FOR JOINING IMAGE FRAGMENTS

In this section we describe about methodology by which fragments are matched and joined accordingly. In fragment analysis, the input is represented by two bitmap images of the fragments which have closed boundary curves over a background. The output is in terms of boundary pixel values. In fragments matching algorithm fragments are matched for longest common portion and joined in order to reconstruct an image.

3.1 Fragments Analysis

Given the two images, first separate the fragments from the background. For this flood fill based algorithm is used that labels every pixel as fragment pixel or background pixel. The pixels that compose the boundary of a fragment are then located as those belonging to the fragment that are adjacent to a background pixel. We calculate the boundary of the fragment by moving from one boundary pixel to another. The next boundary pixel is located by searching among the eight neighbors of the current boundary pixel. The result is a border array containing the coordinates and the color of every pixel belonging to the fragments boundary.

To obtain a rotation- and translation independent model of the border, we use the local curvature. It is well known that there is a one to one correspondence between a regular curve and its curvature function. The curvature function is the derivative of the tangent angle to the curve, parameterized as a function of its arc length. The local curvature of the boundary in a pixel p is calculated as the magnitude of the derivative of the local tangent in p. The local curvature of the border pixels of the first and the second fragment are stored in real-values arrays, called Shape1 and Shape2, respectively. Since the number of pixels in two segments with the same physical length is different if the segments are not parallel, we use an adjusting algorithm that stretches the border and curvature arrays adding interpolated pixels depending on the slope of the local tangent.

3.2 Generation of Common Subsequences using Local Curvature

Since two-dimensional objects are completely described by their closed boundary curves, the reconstruction of fragments participating in a composite scene can be done by matching the boundary curve. Using this approach the reconstruction of an image object can be obtained by matching directly its boundary curve of one fragment to the boundary curve of another fragment and finding their matching portion. Object recognition is one of curve matching algorithms applications. The matching is based on information which is obtained from all sample points on the curve rather than on information obtained from some special points which may or may not exist. This algorithm is based on conversion of curves into numerical strings i.e. arrays, which stores candidate long matching subsequences. Suppose we have two planar curves arbitrarily positioned in the plane, having at least one enough long common subsequences. We want to find the longest common subsequences of these two curves.

3.3 Fragment Matching Algorithm

This section we describe fragment matching algorithm.

Input for this algorithm consists of two border pixel arrays called shape1 and shape2. Their outputs are start points and endpoints of the longest matching subsequences.

1. Obtain border arrays (shape1 shape2).

2. Find long subsequences appearing in both shapes.

3. The outer loop cycles the different placements of arrays. Initially position is fig. 4,then Shape2 is shifted by one position fig.5 and so on until the end of the loop fig. 6.

4. The number of evaluated placements is equal to the length of Shape1.

5. For each single placement, these arrays are compared element by element (pixel by pixel) from left to right. The total number of element comparisons for each placement is equal to the double of length of Shape 2.

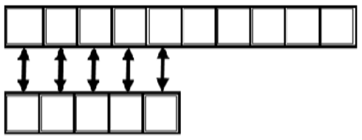


Fig.4: Initial position of two shapes

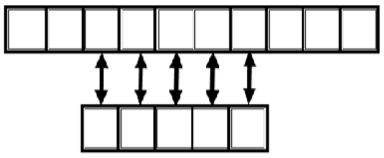


Fig.5: WhenShape2 is shifted by certain position

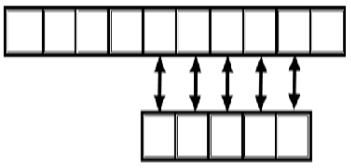


Fig. 6: When Almost Comparison has been done

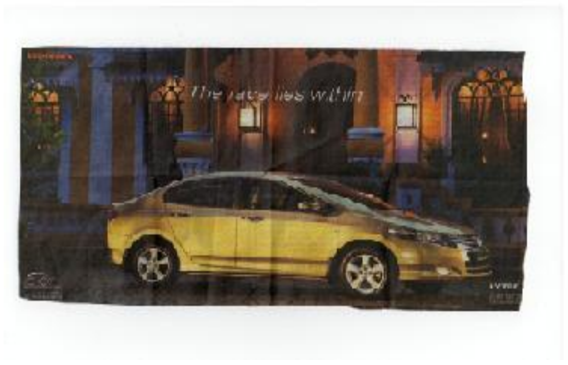
4. EXPERIMENTAL SCENARIO AND RESULT

In this section some of the experimental results are presented to obtain from the implementation of the method described above. For the implementation of the algorithm following are the steps for algorithm.
1.Firstly,Take image (object) that has been broken into n no of fragments(here n=2).
2.Fragments are digitized to get a binary image for each fragment.
3.Calculate boundary value of each fragment which is extracted from the binary image.
4.Find several long common subsequences of the two border pixel arrays, and return their starting points and endpoints.
5.Choose the pair of candidates which has the longest common part.
6.Join these two longest part.
Results are shown in following figures:

For implementation of algorithm, first calculate boundary pixel array using flood fill algorithm. While using flood fill algorithm either 4-connected or 8-connected approach can be used. Here 8-connected approach is used in order to get more accurate results. After implementing border pixel algorithm fragment matching algorithm is applied. Here some images and its fragments are shown which shows the desired outcomes.

Case 1.When fragments are aligned.
Case2.When fragments are situated at a particular angle.

As for both cases in order to obtain rotation and translation independent calculation local curvature value is calculated. The local curvature of each boundary pixel pis calculated as the magnitude of the derivative of the local tangent in p. These values are used in fragment matching algorithm.



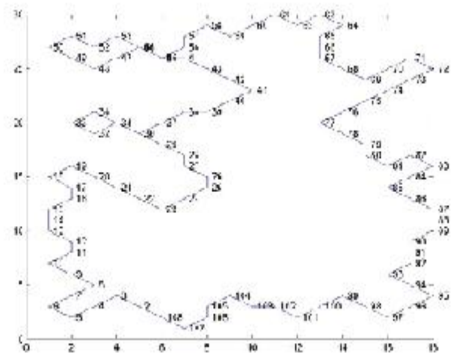


Fig. 7: Original Image and Image Analysis

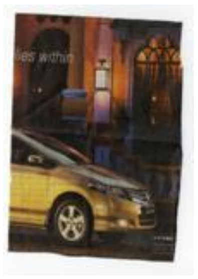


Fig8: Fragment of an Image

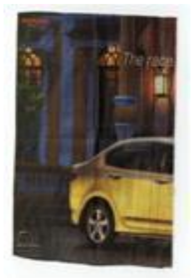


Fig9: Fragment of an Image

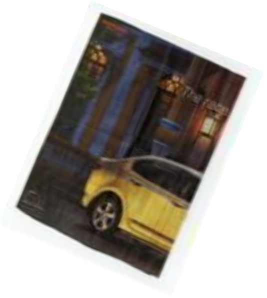


Fig10: Fragment at an angle of an Image

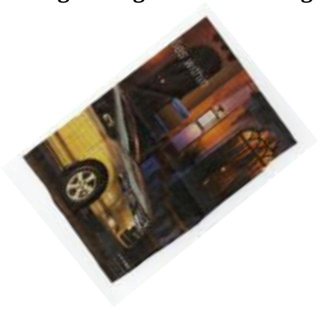


Fig 11: Fragment at an angle of an Image

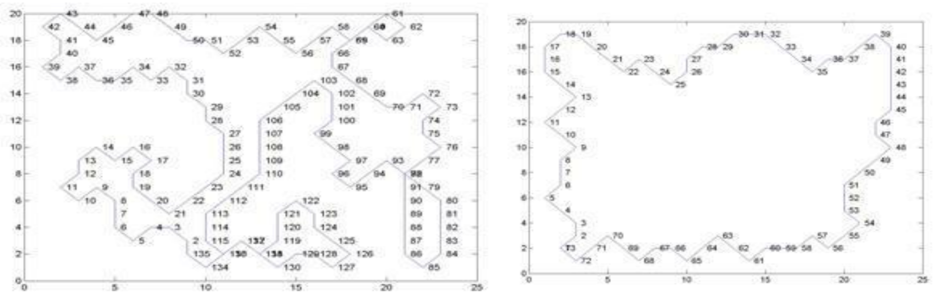


Fig. 12: Fragment Analysis

Graphs show the results of border pixel array. Now output of border pixel array is input to fragment matching algorithm. Fragment matching algorithm results longest matched subsequences. Now matched subsequences can be joined accordingly.

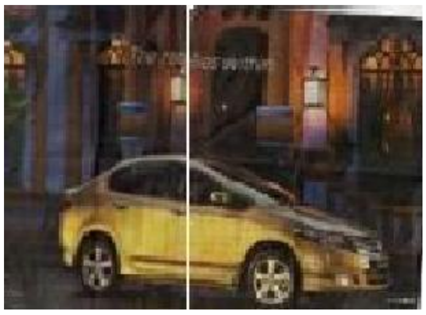


Fig 13: Reconstructed Image

5. CONCLUSION

In this paper joining of 2D fragmentsof an image is present. The method presented uses information about boundaries and color content of fragments. The outlines of the fragments are represented by local curvature value, which is the magnitude of the derivative of the local tangent in p. This paper mainly focused to improve method for reassembling several fragments and to experimentally test the method in real-case applications.

In this method we do not employ a multiscale procedure for matching the fragments and we match only two fragments, so we cannot directly compare the two methods. This method deals more robustly with noise since it allows scattered pairs of significantly different curvature points within a sequence of pairs of significantly equal curvature points. Also is different in terms of fragment joining problem because in object recognition techniques object is known and goal is to find occurrence of the object in a scene by using matching but in fragment joining problem the fragments are to be matched and then joined.

5.1 Future Scope

Another direction of future work is to capable the method in order to joining of n fragments. The reassembled images have been obtained by designing an algorithm and from the collection of fragments, take two fragments with the best match (the longest common portion of borders), add this new fragment obtained by joining the two fragments and remove fragments that recently have been matched from the collection, and continue until we get a single image.

6. REFERENCES

[1] Guide to the Recovery, Recomposition, and Restoration of Shattered Wall Paintings:Experience Gained at the Basilica di San Francesco in Assisi, ICR, Rome, Italy, 2001.

[2] G. Budea and H. Wolfson, “Solving jigsaw puzzles by a robot,” *IEEE Transactions on Robotics and Automation*, vol.5, no. 6, pp. 752–764, December 1989.

[3] A Method For Reassembling Fragments In Image Reconstruction,Francesco Amigoni, Stefano Gazzani, Simone Podico,Artificial Intelligence and Robotics Laboratory; Dipartimento di Elettronica e Informazione Politecnico di Milano; Piazza Leonardo da Vinci 32, 20133 Milano, Italy.

[4] A. Pope, “Model-based object recognition: A survey of recent research,” Tech. Rep. TR-94-04, University of California at Berkeley, Berkeley, CA, USA, January 1994.

[5] MIT Computer Graphics Group, “Intro to computer graphics,” 2003.

[6] H. Wolfson, “On curve matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp.483–489, May 1990.

[7] H. C. de Gama Leitao and J. Stol, "A multiscale method for the joining of two-dimensional fragmented objects", IEEE Transactions on Pattern Analysis and Machine Intelligence,vol. 24, no. 9, pp. 1239-1251, September 2002.

[8] K. Hori, M. Imai, and T. Ogasawara, "Joint detection for potsherds of broken earthenware", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 1999, vol. 2, pp. 440-445.