ROBUST IMAGE MATCHING ALGORITHM

Abdul Ghafoor[1], Rao Naveed Iqbal[2], and Shoab Khan[2]

[1]Department of Electrical Engineering, College of Electrical and Mechanical Engineering,
National University of Sciences and Technology,
Rawalpindi, Pakistan
a_ghafoor30@yahoo.com
[2]Department of Computer Engineering, College of Electrical and Mechanical Engineering,
National University of Sciences and Technology,
Rawalpindi, Pakistan
rao_naveed@yahoo.com, kshoab@yahoo.com

Abstract: *Image matching is an important task. There are many available methods for occluded image matching, e.g., Hausdorff distance and wavelet transform based matching. In this paper we propose new simple image-matching algorithm, robust image matching algorithm (RIMA), an extension to distance transform (chamfer) matching. Distance transform and conventional chamfer matching algorithm are explained. Different matching measures for RIMA are presented. Examples to demonstrate the algorithm and necessary results are also included. Proposed RIMA is time efficient, robust, rotation, scale and perspective invariant method.*

Key words: *Distance transformations, edge matching, chamfer matching, robot vision, image matching.*

1. INTRODUCTION

Image matching remains a central problem in the pattern recognition, image analysis, robotics and computer vision. The exact position of objects, units and features related to image must be known. The noise and errors in preprocessing of images aggravate the matching problems. In most of the image matching algorithms, parameters like translation, rotation, scaling and perspective invariance must often be taken into account. Moreover, the input to different sensors at the same time or input to same sensor at different times will always result in occluded output, and matching an occluded image is always been a challenging task. For quick and reliable object recognition we need to know kind of feature to be used for matching and criterion for best matching. Based on the level of image feature extraction, the matching methods developed in the past can be divided into three classes: algorithms that use image pixel values directly, e.g., correlation methods; algorithms that uses low level features such as edges and corners; and algorithms that uses high level features such as identified objects or relation between the features, e.g., graph-theoretic-methods. The methods making use of image pixel value directly are sensitive to changes between images. High level matching methods are very insensitive to these disturbances. But, the disadvantage is, the high level features must first be extracted and identified and that is, in most cases, a difficult matching problem in itself.

Chamfer matching [1] is a technique for finding the best fit of edge points from two different images, by minimizing a generalized distance between them. The edge points of one image are transformed by a set of parametric transformation equations that describes how the image can be geometrically distorted in relation to one another. Various variants on improvement of chamfer matching have been proposed [2]-[4].

Comparing images using Hausdorff distances was proposed [5] claiming three advantages: relative insensitivity to small perturbations; simplicity and speed of computations; natural allowance for portions of one shape to be compared with another. Various variants of Hausdorff matching are proposed, e.g., [7]-[9].

Rotation invariant pattern matching using wavelet decomposition [10], where input image is decomposed into multi resolution levels in the wave transformed domain and ring projection transform is used which is rotation invariant.

RIMA is low level feature based method, in which edge points or low level feature points are extracted from digital images (using any suitable edge extraction scheme), converted to binary images, which are distance transformed, and then distance transform is used for matching. The distance transform of template is superimposed on the distance transform of the model and values are subtracted pixel wise and matching is found as per the metric. We propose few matching measures: ranked highest numbers of zeros, range, minimum average and RMS value. RIMA is fast, robust, capable of matching the occluded images, independent of rotation, scale and perspective invariance.

In the following section we describe the basic concept regarding distance transform and chamfer matching. Proposed RIMA is explained in section-3, simulation results are given in section-4 and finally conclusion in section-5.

2. BASIC CONCEPTS

In this section we discuss the distance transform and conventional chamfer matching algorithm. Two binary images based on feature and non-feature points, are to be matched. The feature can be any well defined object visible in both images. The two images are not treated symmetrically [4]. One is called predistance (distance transformed) and other is called prepolygon with arbitrary assignation in most of the cases. The predistance image is formed by assigning each non-edge pixel a value that is a measure of distance to the nearest edge pixel.

2.1 Distance Transform

The true Euclidian distance [11] is resource demanding (time, memory) to compute, therefore an approximation is used. Good integer approximations of Euclidean distance can be computed by a process known as chamfer 3-4 distance [1], [2] and [4]. The process of converting a binary image to an approximate distance image is called distance transformation (DT). Chamfer distances are the distances between horizontal/vertical neighbors and two local distances in a 3x3 neighborhood.

In the binary edge image each edge pixel is first set to zero and each non edge pixel is set to infinity. If the DT is computed by parallel propagation of local distances then at each iteration each pixel obtains a new value using the expression [4]:

$$v^{k}_{i,j} = \text{minimum}(v^{k-1}_{i-1,j-1} + 4,\ v^{k-1}_{i-1,j} + 3\ ,\ v^{k-1}_{i-1,j+1} + 4,\ v^{k-1}_{i,j-1} + 3\ ,\ v^{k-1}_{i,j}\ ,\ v^{k-1}_{i,j-1} + 3\ ,\ v^{k-1}_{i+1,j-1} + 4\ ,\ v^{k-1}_{i+1,j-1} + 3\ ,\ v^{k-1}_{i+1,j+1} + 4) \qquad (1)$$

where $v^{k}_{i,j}$ is the value of the pixel in position (i,j) at iteration k. The iteration continues until no value changes. The number of iterations is proportional to the longest distance occurring in the image. Figure 2 shows the distance transform of binary image (Figure 1).

To speed up the distance computation process, sequential DT algorithm [4] is used where two passes are made over the image: "forward" left to right; "backward" right to left.

2.2 Prepolygon

In the prepolygon image (see Figure 3), the edge pixels are extracted and converted to a list of coordinate pairs, each pair being the row and column numbers of an edge pixel. From this list the edge points that are actually used are later chosen according to some criterion, which is application dependent. These chosen points are called the polygon, even though the points may be scattered or representing several polygon segments.

0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0

Figure 1 Binary Image

13	10	7	4	3	4	7	8	11
12	9	6	3	0	3	4	7	10
11	8	7	4	3	0	3	6	9
8	7	4	3	0	3	4	7	10
7	4	3	0	3	4	7	8	11
6	3	0	3	4	6	7	8	11
6	3	0	3	4	3	4	7	10
7	4	3	0	3	0	3	6	9
8	7	4	3	0	3	4	7	10
11	8	7	4	3	4	7	8	11

Figure 2 DT of Reference/Target

				1			
					1		
				1			
			1				
		1					
		1					
			1		1		
				1			

Figure 3 Prepolygon

2.3 Matching Measure

The polygon is superimposed on the distance image (see figure 4). An average of the pixel values that the polygon hits is the measure of correspondence between the edges, called the edge distance. A perfect fit between the two edges will result in edge distance zero, as each polygon point will then hit an edge pixel. The actual matching consists of minimizing the edge distance. To make this minimization as simple as possible, the edge distance function should be as smooth and convex as possible. There are many variants of matching measure averages, e.g. arithmetic, root mean square and median.

$$\text{Arithmetic average distance} = (v_1 + v_2 + \ldots + v_N)/N \qquad (2)$$

$$\text{Root mean square average distance} = \{(v_1^2 + v_2^2 + \ldots + v_N^2)/N\}^{1/2} / 3 \qquad (3)$$

where N is total number of points in polygon and $v_1, v_2, \ldots, v_N$ are pixel values of corresponding coordinate pairs.

$a_{rithmetic} = (3+3+3+3+0+3+3+3+3)/9 = .8889$

$d_{rms} = \{(3^2+3^2+3^2+3^2+0+3^2+3^2+3^2+3^2)/9\}^{1/2}/3$

$= .9428$

Each position of the polygon corresponds to an edge distance. The position with minimum edge distance is matching position. The translation and rotational parameters defines the polygon matching position.

13	10	7	4	3	4	7	8	11
12	9	6	3	0	3	4	7	10
11	8	7	4	3	0	3	6	9
8	7	4	3	0	3	4	7	10
7	4	3	0	3	4	7	8	11
6	3	0	3	4	6	7	8	11
6	3	0	3	4	3	4	7	10
7	4	3	0	3	0	3	6	9
8	7	4	3	0	3	4	7	10
11	8	7	4	3	4	7	8	11

Figure 4 Polygon Superimposition on DT

3. RIMA

Like conventional chamfer matching, RIMA is low level feature based method using DT. Unlike conventional chamfer matching, where model image is translated to distance transform, and template image which is polygon is superimposed onto the model and matching is found as per matching measure, we propose RIMA, in which edge points or low level feature points are extracted from digital images (using any suitable edge extraction scheme), converted to binary images, which are distance transformed, and then distance transform is used for matching. The distance transform of template is superimposed on the distance transform of the model and values are subtracted pixel wise and matching is found as per the metric. We propose few matching measures: ranked highest numbers of zeros, range, minimum average and RMS value.

3.1 Algorithm and Matching Measure

Consider a reference image A of size MxN and a template image B of size OxP with elements $a_{m,n}$ and $b_{o,p}$ respectively. Assume $A \geq B$; $m=1,2,\ldots,M$; $n=1,2,\ldots,N$; $o=1,2,\ldots,O$; $p=1,2,\ldots,P$; range1>range2. We can define C, such that $C = o\in O, p\in P\,[c_{o,p}]$ for all o and p.

```
for q=0: m-o
   for r=0: n-p
        $C= o\in O, p\in P[\,[a_{o+q,p+r}]-B]$
        Measure1=find[$o\in O, p\in P[c_{o,p}=0]$]
         Measure2=find[$o\in O, p\in P$ [range1<$c_{o,p}$<range2]]
OR      Measure2=find[ $|c_{o,p}|$<range]
        Measure3= arithmetic average[$o\in O, p\in P[c_{o,p}]$]
        Measure4=rms average [$o\in O, p\in P\,[c_{o,p}]$
   end
end
```

We define four different measures in algorithm for matching.

In first, the maximum number of zeros elements are searched after subtracting the template DT from reference DT. If the number of zeros elements are equal to the maximum size of the template, then there is a perfect match. However if the number of zero elements after subtraction is not equal to the maximum array size, the match is not the perfect and that will happen in case of occluded, rotated, scaled and perturbed images.

In second, after subtraction of template from reference, the resultant image points are checked for the predefined range and we maximize the difference in that range. To avoid false matching, range is reduced. Goal is to find a translation at which the numbers in the range maximizes, which will be a match and we can also give the confidence level of matching.

In third, we propose taking the arithmetic average (2) of all the numbers and find the absolute minimum value. The minimum value is the match, and we can have a confidence matching level. If absolute minimum value is zero at a specific translation we have perfect match.

And in fourth, propose taking the RMS average (3) of all the numbers and find the minimum value. The minimum value is the match, and here too we have a confidence level. If minimum RMS value is zero at a specific translation we have perfect match.

4. EXPERIMENTAL RESULTS

In this section we present the experimental results for evaluating the efficiency of the proposed matching algorithm. All experiments are implemented on a personal computer with Pentium IV, 1.7GHz processor. We have conducted various experiments, but only one result is shown here. The images of different sizes having 8 bit gray level were used. All the images (reference/template) were applied soble operator for edge detection. After finding the DT image, templates were matched to the reference as per the matching measures. Results were simulated in Matlab 6.1.

Figure 5 Gray Level reference Image

Figure 6 Edge Reference Image

Figure 5 is a 487X204, 8 bit gray level reference image. Figure 7 is a 108x38, 8 bit gray level template image. Template is rotated and perturbed with noise. Figure 6 and Figure 8 are edge images of reference and template images , respectively. These edge images are then converted to 3-4 DT images, applied as input to RIMA, and translation parameters are found as per matching measure. Matching measure three was used in this example. It took about 168 seconds for producing the match including about 2-3 minutes DT computation time. For real time matching, DT of reference can be computed off shelf, and more that 50% computation time is saved.

Figure 9 shows that template is matched to reference. Only translational parameter is used for matching.

A rotated image was also perfectly matched with 95.5% confidence level.

Figure 7 Gray Level Template

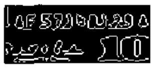

Figure 8 Edge Template

4. CONCLUSIONS

In this paper we proposed a robust image matching algorithm, RIMA. We discussed the matching measures of conventional chamfer matching algorithm and RIMA. We discussed four matching measures. One experimental example is presented. Experimental results show RIMA, a time efficient, robust, rotation and scale invariant. Computational time of real time system can be reduced manifold by having a dedicated hardware and off shelf computation of DT.

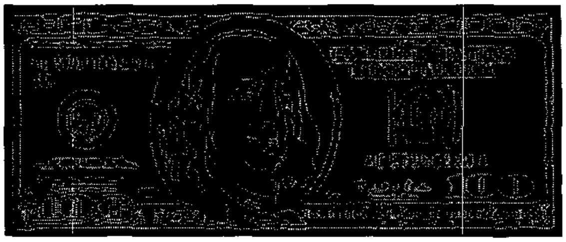

Figure 9 Match

REFERENCES

[1] H.G. Barrow, J.M. Tanenbaum, R.C. Botles and H. C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," In *Proceedings of 5th International Joint Conference on Artificial Intelligence*, Cambridge, MA, pp. 659-653,1977.

[2] G. Borgefores, "An improved version of the chamfer matching algorithm," *in 7th Int. Conf, Pattern Recognition* , Montreal, P.Q.,Canada, pp. 1175-1177, 1984.

[3] A. Rosenfeld, "Multi resolution image representation", in *Digital Image Analysis*, S. Levialdi, Ed., London: Pitman, pp. 18--28, 1984.

[4] Gunnila Borgefores, "Hierarchical chamfer matching: A parametric edge matching algorithm", *IEEE transaction on Pattern Analysis and Machine Intelligence.* Vol. 10, no. 6, pp. 849--856, 1988.

[5] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Ruklidge, "Comparing images using the Hausdorff distance", *IEEE Trans. Patt. Anal. Machine Intell.*, vol 15, no 9, pp 850--863,1993.

[6] L.Paul Chew, Kalara Kedem, "Getting around a lower bound for the minimum Hausdorff distance", *Computational Geometry 10*, pp197--202, 1998.

[7] David M. Mount, Nathan S. Netanyahu, Jacqueline Le Moigne, "Efficient algorithms for robust feature matching", *Pattern Recognition* 32 pp 17--38, 1999.

[8] Oh-Kyu Kwon, Dong-Gyu Sim, Rae-Hong Park, "Robust Hausdorff distance matching algorithms using pyramidal structures", *Pattern Recognition* 34, pp 2005-2013, 2001.

[9] Yongsheng Gao, Maylor K.H. Leung , "Line segment Hausdorff distance on face matching", *Pattern Recognition* 35, pp 361-371, 2002.

[10]D. M. Tsai ,C. H. Chiang, " Rotation invariant pattern matching using wavelet decomposition", *Pattern Recognition Letters* 23, pp 191-201, 2002.

[11]S. Hezel, A. Kugel, R. Manner, D.M. Gavrila, "FPGA-based Template Matching using Distance Transforms", *Proceedings 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, 2002.