ROBUST MATCHING OF AFFINELY TRANSFORMED OBJECTS

Herbert Suesse and Wolfgang Ortmann

Friedrich-Schiller-University Jena, Department of Computer Science
Ernst-Abbe-Platz 1-4
D-07743 Jena, Germany

ABSTRACT

This paper presents a general robust solution for the problem of affine object matching, whereby an object can be given as a discrete point set, a set of lines, or a closed region. Let be given two such objects which are related by a general affine transformation (up to noise and maybe some additional distortions of the object). Then we can determine the six parameters a_{ik} of the affine transformation using some new general moment invariants. These invariants are global, but assigned locally to any object point. With these invariants and using the Hungarian method or dynamic programming it can be computed a weighted point reference list. The affine parameters a_{ik} can be calculated from this list using the method of the least absolute differences (LAD) method. Our approach is very robust against noise and distortions. The algorithm can be used also for all subgroups of the affine group. Additionally, it is an unifying approach for all classes of objects: Discrete point sets, sets of lines, and closed regions. Many wellknown algorithms have problems with the case of symmetries of the objects, our approach is stable against symmetries. Experimental results both on simulated and real objects validate the robustness of the algorithm. In the case of closed regions our algorithm performes better than SQUID [6].

Keywords: Contour matching, affine matching, affine invariants, dynamic programming, Hungarian method

1. INTRODUCTION

Feature matching is an important task in multiview image analysis and in the registration of images. An overview about affine matching can be found in [2] and [3]. The main idea of this paper is an unifying approach for global matching of discrete point sets, sets of lines, and closed contours (closed regions) using moment invariants. Furthermore, we discuss the problem of finding corresponding points of two objects using techniques as the Hungarian method and dynamic programming. The object matching will be performed in the space of the so called Hu invariants, see [5]. In the original literature the Hu invariants are only invariant with respect to similarity transformations. In the present paper we develop the Hu invariants which are even affinely invariant using the ideas of normalization. We distinguish the invariants for the three types of objects, discrete point sets, lines and closed regions. With the help of these new invariants, a list of weighted point references can be calculated using assigment techniques as the Hungarian method and the method of dynamic programming. From the list of corresponding points the transformation can be calculated by the least absolute differences (so called LAD method) using linear programming. The algorithm is very robust against distortions especially in the case of closed regions. A global affine matching using invariants is not new: Such methods parameterize contours according to these invariants, normalize the contours by shifting their representation in parameter space, and correlate the resulting invariant representation. An disadvantage is that the calculation of these affine invariant quantities requires a high degree of differentiation along contours, and that contour smoothing be performed in order to avoid large errors in the differentiation due to noise. Our approach parameterize also contours, but according to novel global moment invariants under homogeneous affine transformations.

2. AFFINE INVARIANTS

2.1. Normalization of the moments (DM,LM,AM)

Let be given an affine transformation and two objects. We use the method of normalization well known in the theory of affine invariants for planar objects, see ([4, 8]). As features we use the discrete moments (DM) for discrete point sets, the line moments (LM) for line objects, or the area moments (AM) for closed regions. Therefore, at the beginning we define the three types (DM,LM,AR) of moments $M_{j,k}$ for the three types of objects. Central moments $m'_{j,k}$ are given by a translation $x' = x \quad \bar{x}$ and $y' = y \quad \bar{y}$ with $(\bar{x},\bar{y})$ as centroid, so that we have $m'_{1,0} = 0$ and $m'_{0,1} = 0$ in the first normalization step. This is correct for all three types of moments. In the second step we perform with $x'' = x' + \alpha y'$ and $y'' = y'$ a shearing (or “stretching”) in x direction. By this, the old central moments $m'_{j,k}$ are transformed in new moments $m''_{j,k}$.

With the simple requirement $m''_{1,1} = 0$, the parameter α is determined as $\alpha = \quad m'_{1,1}/m'_{0,2}$: This is correct for the moments (DM,AM), but not for the line moments. Now we can calculate all new moments $m''_{j,k}$ with given normalization values for $\quad m''_{1,0}, m''_{0,1}, m''_{1,1} \quad = (0, 0, 0)$.

Finally, a general anisotrope scaling $x''' = \beta x''$ and $y''' = \gamma y''$ yields the moments $m'''_{j,k}$. The moments $m'''_{2,0}$ and $m'''_{0,2}$ shall be normalized to 1 so that the parameters β and γ have to be $\beta = \sqrt[8]{m''_{0,2}/m''^{3}_{2,0}}$; $\gamma = \sqrt[8]{m''_{2,0}/m''^{3}_{0,2}}$ for area moments (AM), and $\beta = 1/\sqrt{m''_{2,0}}$; $\gamma = 1/\sqrt{m''_{0,2}}$ for discrete moments (DM). In the case of line moments (LM) we consider only isotrope scalings with $\beta = \gamma = \lambda$, and normalize $m'''_{0,0} = 1$ with $\lambda = 1/m''_{0,0}$. The line moments (LM) can only be normalized for translations and isotrope scalings.

2.2. Features which are invariant under full affine transformations

A robust normalization of the rotation is not possible, see [8]. For that reason, we try to find numerically stable expressions which are invariant against rotations and reflections. Such numerically stable expressions are introduced by Hu (see [5]). Hu has been derived 7 invariants H_1,H_2,..., H_7 including one to third order moments. For our purpose we need additional Hu invariants including up to fourth order moments. These invariants can be easily derived using the so called complex moments, see [4]. Some of the classical Hu invariants and our new Hu invariants are given in Table 1. It is very important that these features are invariant against rotations and reflections for all three types of moments (DM,LM,AM). If we compute these features in Table 1 with the above normalized moments than we get affine invariant features. But, this is not the case for line moments, with this normalization scheme we get only invariants with respect to similarity transformations. An additional advantage of our normalization scheme is that it is simple to derive invariants for any subgroup of the affine group.

2.3. Detection of corresponding points

The main idea is now to find corresponding points using global features of the objects. Let be given a fixed reference pair of points $(\mathbf{x_1}, \mathbf{x}'_1)$ and any reference pair $(\mathbf{x}, \mathbf{x}')$. With a given affine transformation we receive $\mathbf{x}' = \mathbf{Ax} + \mathbf{a}$ and $\mathbf{x_1}' = \mathbf{Ax_1} + \mathbf{a}$. It follows $\mathbf{x}' \quad \mathbf{x_1}' = \mathbf{A}(\mathbf{x} \quad \mathbf{x_1})$ and that means: If we put the origin of the coordinate system in the fixed points $\mathbf{x_1}'$ or $\mathbf{x_1}$,respectively, then we have only a homogeneous affine transformation (elimination of the translation, but not by the centroid). Now we put the origin of the coordinate system in any point and compute features in this system which are only invariant with respect to homogeneous affine transformations. With this procedure we have assigned to every point of the object a vector of global moment invariants. In such a fixed coordinate system we have to carry out the normalization of the moments described in section 2.1 without the normalization of the translation by the centroid. Using these normalized moments, the features H_k of Table 1 can be calculated which are now invariant to homogeneous affine transformations. Table 1 shows that it can be used low order invariants, e.g. H_1 that is constant under full affine transformations, but not under homogeneous affine transformations. In the case of closed regions we are using only the points of the border of the object, the so called contour points.

order	Affine “Hu invariants”
0 H_0	m_{00}
1 H_1	$m_{10}^2 + m_{01}^2$
2 H_2	$m_{20} + m_{02}$
3 H_3	$(m_{30} \quad 3m_{12})^2 + (3m_{21} \quad m_{03})^2$
3 H_4	$(m_{30} + m_{12})^2 + (m_{21} + m_{03})^2$
4 H_8	$(m_{40} \quad m_{04})^2 + 4\,(m_{31} + m_{13})^2$
4 H_9	$(m_{40} + m_{04})^2 + 16\,(m_{31} \quad m_{13})^2$
	$12m_{22}\,(m_{04} \quad 3m_{22} + m_{40})$
4 H_{11}	$m_{40} + 2m_{22} + m_{04}$

Table 1. Some affine Hu invariants up to 4-th order moments (not complete)

2.4. Problems of a simple next neighbor search

First of all we have to calculate in the space of all invariants a list of reference points $(x_i, y_i) \leftrightarrow (x'_j, y'_j), i = 1, \ldots, m; j = 1, \ldots, n$ where the numbers of the object points m and n of the objects are often different. This can be done by a simple nearest neighbor search using the distance of two points in the space of the invariants. This simple search can be improved by calculating the mimimal costs of all assignments (exactly one permutation in the case $m = n$) of the points using the so called *Hungarian Method* , see also [1]. In an experiment we have calculated the per cent errors of the correct assigments for a discrete set of points. We have chosen 100 randomely chosen points in an 400 400 image, additionally we have chosen randomely an affine transformation and all the transformed points are distorted by noise. This was done by different levels of noise. In the Table 2 the improvement of the candidate matches can be seen.

If we have “regular” objects the procedure works very well. But if there are objects with symmetries then the search arises problems. In these cases there are some points of the objects with the same invariants, e.g. considering the contours of triangles or rectangles (using discrete point sets, that is not often the case). In the left part of the Fig. 1 it is showed a “regular” object and the affine transformed

Pixel	*Nearest Neighbor*		*Hungarian*	
noise	*%errors in*		*%errors in*	
1	31%	2;5%	19;1%	2;6%
2	41%	2;5%	27;5%	3;0%
3	50%	2;5%	35%	3;3%
4	56%	2;3%	41%	3;3%
5	61;4%	2;1%	46;8%	3;3%
10	75%	2%	64%	2;8%

Table 2. Nearest neighbor search - Hungarian method

object. The correct correspondences of the contour points can be detected with a simple next neighbor search. In the right part we have a matrix or an image whereby the x-axis means the successive counting of the contour points of the first object, and the y-axis means the counting of the contour points of the second one. The black points in the right part of Fig.1 are the results of a nearest neighbor search, these black points form nearly a line. But if we have e.g. two triangles as objects then it can be seen a “chaos” in the correspondences. Therefore, we use another idea to use the information about the order of the contour points in the case of given closed regions.

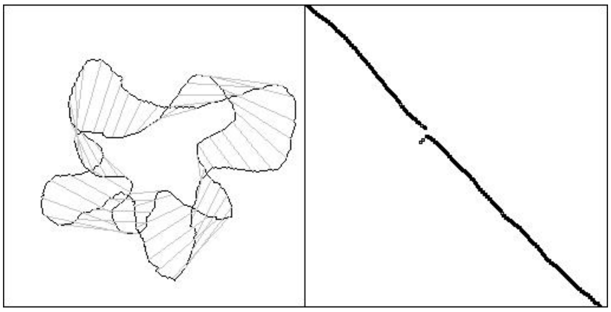

Fig. 1. Correct contour point correspondences

2.5. Dynamic programming

The idea of dynamic programming for matching of closed regions can be found also in [7],[9], but only for similarity transformations. The idea is the following: We compute the matrix $dist(i;j)$ of the distances in the space of all Hu-invariants using m points $i = 1;:::;m$ from the contour one and n points $j = 1;:::;n$ from the contour two. This matrix can be seen in Fig. 2. The x-axis is the order-numbering of the points from the first contour, the y-axis is with the points from the second contour, and the origin of the coordinate system is in the left upper corner. Both starting points are chosen arbitrarily. The white points mean that these reference pairs of points have a small distance. Now we use the information that there is an order in the contour points. We have to found a path beginning in the first column and ending in the last column with minimal distance costs. Because of the arbitrary starting points the calculation has to take into account a cyclic numbering of the points. Using dynamic programming, it must be calculated the costs in any point $c(i;j)$ by

$$c(i;j) = min(c(i \quad 1;j);c(i \quad 1;j \quad 1) + dist(i;j)) :$$

If the affine transformation contains a reflection then the orientation of the points in the transformed contour is vice versa to the first contour. Therefore, it must be calculated a second cost matrix by

$$c(i;j) = min(c(i \quad 1;j);c(i \quad 1;j + 1) + dist(i;j)) :$$

By $\min_{j=1;:::;n}(cost(m;j))$ and backpropagation it can be found the optimal path for the first cost matrix and the second one. Now it can be chosen the path with the lower costs. This optimal path gives us a list of corresponding points of both contours, and the $dist(i;j)$ ist a weight of the reference pair $(i;j)$. In the left part of Fig. 2 it is displayed the distance matrix of a triangle and its affinely transformed object. In the right part of Fig. 2 the detected optimal path can be seen. The complexity of the algorithm is $O(m \quad n)$. If we take into account that a contour is beginning at a point and is ending at the same point then the path finding problem is another one. If the starting point is $(0;j)$ in the first column of the distance matrix, and the end-point is $(m;(j + n)mod\ n)$ then it is to find an optimal path from the starting point to the end-point. That must me done for every point in the starting column. That implies a cubic complexity of the algorithm. But, it follows from our experiments that this effort is not necessary, the improvement of the results is not satisfactory. For that reason we are using only the quadratic algorithm.

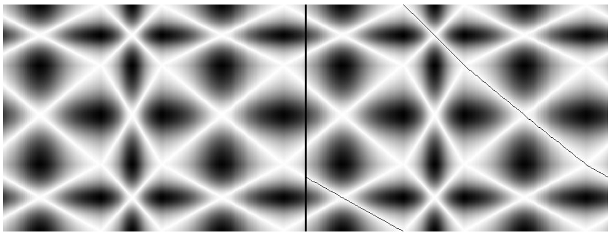

Fig. 2. Left part: The distance matrix of two triangles, Right part: The detected path

2.6. Estimators for the affine parameters

The next step is now the calculating of the affine transformation from the weighted correspondence list. This can be done in a common way by the least squares estimator using the L_2-metric or by the L_1-metric using linear programming, see [11].

3. EXPERIMENTAL RESULTS FOR SYNTETIC DATA SETS

In order to test the proposed approach, two types of experiments are shown in this section: (i) testing the matching with noise of the object points, (ii) testing the matching with disturbations of the transformed object. The idea of the algorithm implies that it is very robust against all forms of symmetries in the case of closed regions. The shape of the given object does not influence the quality of the matching result. In a lot of experiments we have chosen any contour and randomely an affine transformation. The tranformed contour or object is contaminated by different levels of noise. The matching result ist very robust against noise, even with very high levels of noise we get good matches, see e.g. Fig. 3. In another experiment we have done a lot of drastic distortions of the contour, the matching results are very robust, see e.g. Fig. 4. In experiments with real data concerning Euclidean transformations the rotation angle could be detected stable also in the cases of object symmetries. In the experiments using discrete point sets we have implemented the Hungarian method. Concerning the robustness of the method another application is successful, the fitting of any given object by a given primitive. Let be given e.g. an object, a triangle as the second object, and the class of affine transformations. Then we use the matching result as a fitting of the object by a triangle.

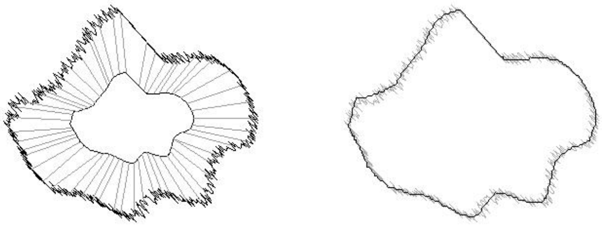

Fig. 3. References and matching result using a noisy contour

4. REFERENCES

[1] M.J. Atallah (Ed.), Algorithms and Theory of Computation. CRC Press, 1999

[2] Bachelder, I.A. Ivan A., and Ullman, S., Contour Matching Using Local Affine Transformations, *CVPR92* 798-801.

[3] Brown, L.G. Lisa G., A Survey of Image Registration Techniques, *ACM Computing Surveys (24), No. 4*,December 1992, pp. 325-376.

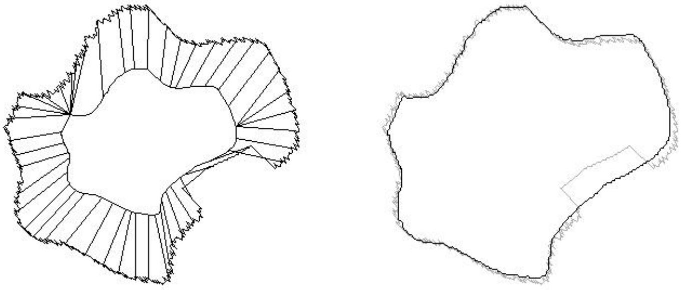

Fig. 4. References and matching result using a noisy contour and a stronger distortion

[4] Y.S. Abu-Mostafa and D.Psaltis, Image Normalization by Complex Moments, *PAMI 7* (1985) 46-55.

[5] M.K.Hu: Visual pattern recognition by moment invariants.*IT* 8 (1962) 179-187.

[6] F.Mokhtarian, S. Abbasi,, and J. Kittler: Efficient and Robust Retrieval by Shape Content through Curvature Scale Space. *Proc. Int. Workshop Image Databases and MultiMeida Search* , pp. 35-42, 1996

[7] E.G.M. Petrakis, A. Diplaros, and E. Milios: Matching and Retrieval of Distorted and Occluded Shapes Using Dynamic Programming. *IEEE Trans. PAMI* 24, No. 11, 1501-1516, 2002

[8] I.Rothe, H.Suesse, K. Voss: The method of normalization to determine invariants. *IEEE Trans. PAMI 18* , 366-377, 1996

[9] T.B. Sebastian, P.N. Klein, and B.B. Kimia: On Aligning Curves. *IEEE Trans. PAMI* 25, No. 1, 2003

[10] K.Voss, H.Suesse: A new one-parametric fitting method for planar objects. *IEEE Trans. PAMI* 21 , 646-651, 1999

[11] K.Voss, H.Suesse: Affine Point Pattern Matching. Pattern Recognition. Lecture Notes in Computer Science 2191, 2001, pp.155-162

[12] K.Voss, H.Suesse: Invariant fitting of planar objects by primitives. *IEEE Trans. PAMI* 19 (1997) 80-84