A Novel Approach for Affine Point Pattern Matching

H. Suesse, W. Ortmann, and K. Voss

Friedrich-Schiller-University Jena, Department of Computer Science
Ernst-Abbe-Platz 2
D-07737 Jena, Germany
{nbs,noo}@uni-jena.de
http:/www.inf-cv.uni-jena.de

Abstract. Affine point pattern matching (APPM) is an integral part of many pattern recognition problems. Given two sets P and Q of points with unknown assignments $p_i \rightarrow q_j$ between the points, no additional information is available. The following task must be solved:

– Find an affine transformation $\mathbf{T}$ such that the distance between P and the transformed set $Q' = \mathbf{T}Q$ is minimal.

In this paper, we present a new approach to the APPM problem based on matching in bipartite graphs. We have proved that the minimum of a cost function is an invariant under special affine transformations. We have developed a new algorithm based on this property. Finally, we have tested the performance of the algorithm on both synthetically generated point sets and point sets extracted from real images.

1 Introduction

Point pattern matching (PPM) is an important problem in image processing. Cox and Jager have already given in 1993 a survey with respect to different types of transformations and methods [4], also Wamelen et.al. have given a survey in [22]. An overview and a new statistical approach is published by Luo et.al. in [15]. In recent years there have been several significant advances in this area, ranging from $O(n^6)$ complexity to $O(n^2)$ algorithms. These advances have focused on developing faster algorithms and better statistical interpretation [15]. Our proposed method is theoretically well founded and tackles the problem of affine point pattern matching in $O(n^3)$ complexity only using geometric distances between the points. The algorithm is based on the Hungarian method and is completely different to the widely used "Nearest Neighbors Search", see e.g. [3]. The advantage of our algorithm is the robustness against noise and outliers in practical applications. The algorithm even works very well in the case of projective transformations that are typical for real cameras. Our approach can be separated in two steps:

– Problem A (**A**ssignment): Find all point references between two given point-sets with respect two an unknown affine transformation. This means that it is to find the correct permutation resp. it is to solve the assignment problem. For this step we have developed a novel theory.

– Problem B (**B**acktransformation): Compute from the references the affine transformation.

2 Graph Theory Domain - The Assignment Problem

Now we will solve the first problem A. Let P be a given set of n points $\mathbf{p}_i$, $i = 1, ..., n$. The ordering of the points is unknown and no additional knowledge about the image is given. There is a geometric transformation $\mathbf{T}$ that holds $\mathbf{q}_i = \mathbf{T}\mathbf{p}_i$, $i = 1, ..., n$. This means that $\mathbf{q}_i$ from the point set Q is the reference point of $\mathbf{p}_i$ with respect to the given transformation $\mathbf{T}$. However, in practical applications the transformation is unknown and also the references between the points of P and the points of Q are unknown. Only the transformation group is known as a priori knowledge. The important goal in matching of point sets is to find all corresponding pairs of points.
First of all we assume in the following that the number of points of both point sets are equal to n. We show later that the problem is not strongly limited by this constraint in practice. We are searching for the correct point correspondences, this means we have to find a correct permutation of the indices.
The main idea is now to reduce our problem to matching in a bipartite graph. A bipartite graph is a graph whose vertex set can be split into two nonempty, disjoint sets P and Q in such a way that every edge joins a vertex of P to a vertex of Q. A graph is weighted if we give a cost function c that associates each edge with a real value c. We denote by $(k_1, k_2, ..., k_i, ..., k_n)$ any permutation of the indices of the points $\mathbf{p}_1, ..., \mathbf{p}_n$. For our problem we have to solve for a permutation

$$\sum_{i=1}^{n} c(\mathbf{p}_i, \mathbf{q}_{k_i}) \quad \rightarrow \quad minimum.$$

Now we have the assignment problem of finding a perfect minimum cost weighted matching in bipartite graphs. The matching on bipartite graphs is a well studied problem in combinatorial optimization, see e.g. [14].
For the solution we use an algorithm, known as *Hungarian Method* [12],[16], [14]. It takes as input a matrix of the weights of the edges that relate the two disjoint sets of the bipartite graph and outputs the optimal permutation and minimum cost. The original Hungarian method is of $O(n^4)$ runtime complexity [12]. Using special data structures [16], the runtime complexity is reduced to $O(n^3)$. It is possible to improve the runtime complexity using parallel algorithms, see [7]. We have made the assumption that the two point sets P and Q are of equal cardinality n. For practical applications the approach could be extended to include the case of different size as well, by simply adding empty transactions (e.g. zero) to the smallest point set. But, our proof and the theory are only exact in the case of equal cardinality.

3 The Novel Theory

3.1 Translations

The main problem is now to find a correct measure for the costs so that we can determine the correct permutation. For the cost function we will investigate the properties of geometric distances $d(\mathbf{p}_i, \mathbf{q}_{k_i})$.

At first we choose the simplest transformation, a translation. This means

$$\mathbf{q}_i \;=\; \mathbf{p}_i \;+\; \mathbf{a} \quad , \quad i = 1, ..., n \quad ,$$

whereby $\mathbf{a}^T = (a_{10}, a_{20})$ is a translation vector. If we now choose for the costs the Euclidean distances of the points $d(\mathbf{p}_i, \mathbf{q}_{k_i}) = |\mathbf{p}_i \;-\; \mathbf{q}_{k_i}|$, then it is simple to show by an counter-example that the method does not work. Now we try to use the squares of the Euclidean distances

$$S \;=\; \sum_{i=1}^{n} |\mathbf{p}_i \;-\; \mathbf{q}_{k_i}|^2 \;\;\rightarrow\;\; minimum. \tag{1}$$

We are substituting the translation and get

$$S \;=\; \sum_{i=1}^{n} (\mathbf{p}_i \;-\; (\mathbf{p}_{k_i} + \mathbf{a}))^2 \;=$$

$$= \;\sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{p}_i \;-2 \sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{p}_{k_i} \;-\; 2 \sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{a} \;+\; \sum_{i=1}^{n} (\mathbf{p}_{k_i} + \mathbf{a})^2 \;\rightarrow\; minimum.$$

The solution is the searched permutation. The translation $\mathbf{a}$ is also unknown but given for the problem and therefore a constant. We can see that the terms $\sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{p}_i$, $\sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{a}$ do not depend on the searched permutation. Since

$$\sum_{i=1}^{n} (\mathbf{p}_{k_i} + \mathbf{a})^2 \;=\; \sum_{i=1}^{n} \mathbf{p}_{k_i}^2 + 2 \sum_{i=1}^{n} \mathbf{p}_{k_i}^T \mathbf{a} + \sum_{i=1}^{n} \mathbf{a}^2 \;=\; \sum_{i=1}^{n} \mathbf{p}_i^2 + 2 \sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{a} + \sum_{i=1}^{n} \mathbf{a}^2$$

the term $\sum_{i=1}^{n} (\mathbf{p}_{k_i} + \mathbf{a})^2$ also does not depend on the searched permutation. For this reason we can reduce the problem to

$$\sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{p}_{k_i} \;\rightarrow\; maximum \tag{2}$$

for the determination of the unknown permutation.

Now we make an estimate of the sum $\sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{p}_{k_i}$. Using $\mathbf{p}_i^T = (x_i, y_i)$ and $\mathbf{x}^T = (x_1, x_2, ..., x_n)$, $\mathbf{x}_{perm}^T \;=\; (x_{k_1}, x_{k_2}, ..., x_{k_n})$ (analog for $\mathbf{y}, \mathbf{y}_{perm}$) we get

$$\sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{p}_{k_i} \;=\; \sum_{i=1}^{n} x_i x_{k_i} \;+ \sum_{i=1}^{n}\; y_i y_{k_i} \;=\; \mathbf{x}^T \mathbf{x}_{perm} \;+\; \mathbf{y}^T \mathbf{y}_{perm} \quad .$$

Using the Cauchy-Schwarz's inequality we get

$$(\mathbf{x}^T\mathbf{x}_{perm})^2 \ \leq \| \mathbf{x} \|^2 \| \mathbf{x}_{perm} \|^2 = \ \| \mathbf{x} \|^2 \| \mathbf{x} \|^2 \ = \| \mathbf{x} \|^4 \quad .$$

This means that $\mathbf{x}^T\mathbf{x}_{perm} \ \leq \| \mathbf{x} \|^2$. Finally, it follows

$$\mathbf{x}^T\mathbf{x}_{perm} \ + \ \mathbf{y}^T\mathbf{y}_{perm} \ \leq \| \mathbf{x} \|^2 \ + \ \| \mathbf{y} \|^2 = \ \sum_{i=1}^{n} \| \mathbf{p}_i \|^2 \quad .$$

From this follows that

$$\sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{p}_{k_i} \ \leq \ \sum_{i=1}^{n} \| \mathbf{p}_i \|^2 \quad .$$

The identity (this means the maximum of the left side) in the Cauchy-Schwarz's inequality is be accepted if and only if both vectors are linear combinations of each other. This means that $\mathbf{x}_{k_i} \ = \ \mathbf{x}_i$, $\mathbf{y}_{k_i} \ = \ \mathbf{y}_i$ and therefore it follows $\mathbf{p}_{k_i} \ = \ \mathbf{p}_i$, $i = 1, 2, ..., n$. This statement is very important for the uniqueness of the solution. The result is that we get the correct permutation for any translation of the points by solving the extremal problem (1).

3.2 Special Affine Transformations

In the following we will try to generalize the results to general affine transformations

$$\mathbf{q}_i \ = \ \mathbf{A}\mathbf{p}_i + \mathbf{a} \ = \ \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \ + \ \begin{pmatrix} a_{10} \\ a_{20} \end{pmatrix} \quad .$$

First of all, we are substituting the affine transformation in the same way using the extremal problem (1):

$$S \ = \ \sum_{i=1}^{n} (\mathbf{p}_i \ - \ (\mathbf{A}\mathbf{p}_{k_i} + \mathbf{a}))^2 \ = \tag{3}$$

$$= \ \sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{p}_i \ - 2 \sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{A}\mathbf{p}_{k_i} \ - \ 2 \sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{a} \ + \ \sum_{i=1}^{n} (\mathbf{A}\mathbf{p}_{k_i} + \mathbf{a})^2 \ \rightarrow \ minimum.$$

Again, the first term $\sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{p}_i$ and the third term $\sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{a}$ do not depend on the permutation. It is very simple to see that also the term $\sum_{i=1}^{n} (\mathbf{A}\mathbf{p}_{k_i} + \mathbf{a})^2$ doesn't depend on the permutation. Finally, we have to solve

$$\sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{A}\mathbf{p}_{k_i} \ \rightarrow \ maximum. \tag{4}$$

The idea is now to reduce the problem (4) to the problem (2). For that we introduce a matrix $\mathbf{C}$ with $\mathbf{p}'_i \ = \ \mathbf{C}\mathbf{p}_i$ and compose

$$\sum_{i=1}^{n} \mathbf{p}_i{'}^T \mathbf{p}'_{k_i} \ = \ \sum_{i=1}^{n} (\mathbf{C}\mathbf{p}_i)^T (\mathbf{C}\mathbf{p}_{k_i}) \ = \ \sum_{i=1}^{n} \mathbf{p}_i^T \mathbf{C}^T \mathbf{C}\mathbf{p}_{k_i} \ \rightarrow \ maximum. \tag{5}$$

If we now can find a matrix $\mathbf{C}$ with the relation $\mathbf{A} = \mathbf{C}^T\mathbf{C}$, then the correct permutation is the solution of the problem (4) resp. problem (3). Such a matrix $\mathbf{C}$ exists, if the affine matrix $\mathbf{A}$ is symmetric and positive definite. If this factorization is possible then we can refer it to as "taking the square root" of $\mathbf{A}$. In this case it follows $p'_{k_i} = p'_i$, i.e. $\mathbf{C}p_{k_i} = \mathbf{C}p_i$, and finally $p_{k_i} = p_i$. Our novel theory for the determination of the correct alignment works for the special affine transformation

$$q_i = \mathbf{S}p_i + \mathbf{a} \quad , \quad \text{S is symmetric and positiv definite}$$

by a minimization of the squares of the Euclidean distances of the points. This means that the optimal permutation is an invariant against this special kind of affine transformations. A presumption may be that the theory also works in the case of higher exponents than two of the Euclidean distances. However,

$$S = \sum_{i=1}^{n} |\mathbf{p}_i - \mathbf{q}_{k_i}|^n \quad \rightarrow \quad minimum$$

does not work in the case of $n \neq 2$. This is a very surprising fact. We can not prove this fact, however we can find a lot of counter-examples by doing computer experiments.

3.3 Affine Transformations

Now we will investigate the case of general affine transformations. For the general case it is not possible to find the correct alignment by minimization of the costs. For that reason we try to decompose the general case in a special affine transformation (with a symmetric, positiv definite matrix and a translation) and in a rotation. For the rotation we generate a matching process to find the best permutation.

> **Lemma**: Any affine matrix $\mathbf{A}$ with $det(\mathbf{A}) > 0$ can be decomposed in a symmetric, positiv definite matrix $\mathbf{S}$ and a rotation matrix $\mathbf{R}$ ($\mathbf{R}^{-1} = \mathbf{R}^T, det(\mathbf{R}) = +1$), i.e. $\mathbf{A} = \mathbf{S} \cdot \mathbf{R}$. The constraint $det(\mathbf{A}) > 0$ means only that there is no reflection in the affine transformation.

For the proof we are using the inverse relation $\mathbf{A}\mathbf{R}^T = \mathbf{S}$ with

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \cdot \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} .$$

Using the proposed symmetry, it follows for the non-diagonal elements

$$a_{11}\sin\varphi + a_{12}\cos\varphi = a_{21}\cos\varphi - a_{22}\sin\varphi .$$

The solution is

$$\tan\varphi = \frac{a_{21} - a_{12}}{a_{11} + a_{22}} \text{ with the periodicity } \pi .$$

Within the interval $[0, 2\pi]$ we have two solutions with the period π. We choose those solution with

$$a_{11} \cos\varphi \ - \ a_{12} \sin\varphi \ = \ a \ > \ 0 \ .$$

This is always possible because $\cos\varphi$ and $\sin\varphi$ change their sign with π. From the supposition $det(\mathbf{A}) > 0$ follows that also $a_{21} \sin\varphi + a_{22} \cos\varphi \ = \ c \ > \ 0$ applies. A symmetric matrix S with the conditions $det(\mathbf{S}) > 0$, $a > 0$ and $c > 0$ is positiv definite.
It follows from this fact that we can combine our theory with a direct optimization for the rotation space. Thus, the main conclusion is that we have reduced the 6-dimensional matching problem for affine transformations to an one-dimensional problem. It needs only an optimization process regarding the rotation. It can be chosen any point for the rotation center, e.g. the centroid of the point set.

4 Estimating the Affine Transformation

Using the Hungarian method we get a list of point references. With the help of this list it is simple to solve the second problem B. The simplest method is to compute the transformation $\mathbf{T}$ by an optimization process using e.g. the Least Squares Method (LSE). In our practical implementation we use instead of the LSE method the so called Least Absolute Differences method (LAD), see [19],[20],[21]. The LAD method is more robust against noise and outliers than the LSE method and is easy to implement using linear programming methods, e.g. the simplex method.

5 Algorithm

For practical applications now we have to do:

- We rotate the first point set P (or alternatively the set Q) from 0 degree to 360 degrees with an increment of about 5 until 10 degrees. In most applications an increment of only ten degrees is sufficient. Any point can be chosen as the rotation center, e.g. the centroid.
- For every rotation we solve the permutation problem (1) using the *Hungarian Method* to determine an optimal permutation of the assignment problem A.
- By the references $p_i \ \rightarrow \ q_{k_i}$ we calculate the affine transformation using the LAD-Method (problem B, see e.g. [19],[20]). We transform the first point set P by this affine transformation to a point set P' and consider the references $p'_i \ \rightarrow \ q_{k_i}$.

– We calculate the distance between P' and Q. An advantage is that it is not necesarry to use a modified Hausdorff distance (see [6]). We use the calculated permutation $p'_i \rightarrow q_{k_i}$ for the following point set distance:

$$d(P', Q) = \sum_{i=1}^{n} |p'_i - q_{k_i}| \quad .$$

– We choose those angle for the rotation with the lowest distance $d(P', Q)$. The permutation concerning this angle is the searched permutation.

It follows that the algorithm does not contain a strong separation in problem A and problem B. The algorithm is a combination of solving the assignment problem and the problem of determination of the affine transformation.

6 Implementation

The algorithm described above was implemented in the C++ programming language. We tested the program on many randomly generated data sets and did a case study on a calibration grid. Instead of the LSE method for finding the global affine transformation resp. projective transformation we use the LAD method. As already mentioned the LAD method is more robust against noise and outliers than the LSE method solving the problem B. Because the algorithm is a combination of solving problem A and problem B we have two factors for the robustness of the method:

– How many percent of the correct references can be computed for the correct rotation angle in the case of noise, outliers or a different cardinality of the point sets?
– How many percent of the correct references are necessary for the computation of the correct affine or projective transformation? For this purpose we used the LAD-method in the present paper. However, in a future work we will test accumulation methods (cluster methods, voting methods) to solve problem B.

6.1 Random Point Sets

We tested the program on a large number of randomly generated point sets P. A randomly generated point set has always 100 points in a 400 x 400 pixel image. An affine transformation is generated with the condition that the determinant of the affine matrix is between 0.8 and 1.2. Then we have transformed the generated point set to Q. In our first experiments, in order to make them more realistic, we used normally distributed noise for every coordinates of a point in the point set Q. We will denote the standard deviation of noise by σ. Adding noise to these points we tested our new algorithm. It can be seen in Fig. 1 that the algorithm is very robust against noise.

In a second series of experiments we have investigated the steps of the discretization of the rotation. It is very surprising that the permutation problem is robust against the rotation. Fig. 2 shows the percentage of correct references against the rotation increment in degree. Beginning with an increment of 15 degrees, no correct references occur. For most practical applications a rotation increment of 10 degree is sufficient.

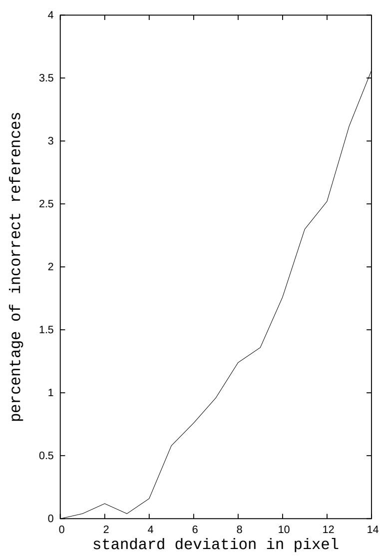

Fig. 1. Alignment error in dependence of noise of all points

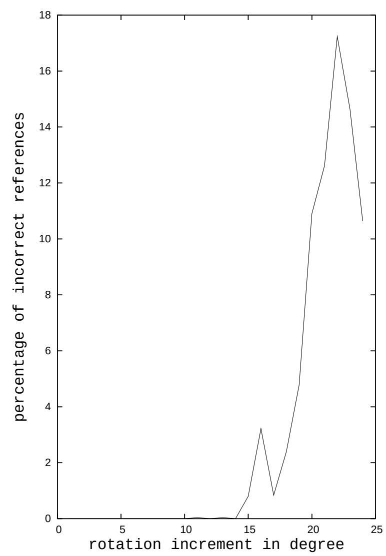

Fig. 2. Alignment error in dependence of discretization of rotation

6.2 Outliers

It is very difficult to give a real evaluation of the algorithm with respect to outliers. The LAD-method for solving problem B basically determines the behavior of the algorithm against outliers. About 5 percent outliers or 5 percent difference in the cardinality of the point sets also lead to a stable result. We have tested the algorithm for point sets with a lot of different levels of the cardinality. The result is very surprising. There are examples where the point set P has only 20 percent of the point set Q, and the algorithm works very well. However, there are also examples with 90 percent and the algorithm fails. It depends apparently on the "order" of the points. This area is our research for the near future.

6.3 Case Study

To test our new algorithm on a real world situation, we applied it to a planar calibration problem to test the influence of (practical) projective transformations. We took a calibration grid of 50 squares printed on a paper. The coordinates are known as the intersection of the two diagonales of each square and form the point set P. These 50 coordinates $(x_i, y_i), i = 1, ..., 50$ are saved in a data structure, (see Fig. 3) . We took also a real image produced with a camera from this printed calibration grid with projective transformations and distortions of the lens. Lenses with a small focal length produce often distortions, (see Fig. 4). This means that our real image can not be approximated by an affine transformation. In our algorithm we have to substitute the affine transformation by a projective mapping. In a first step we are searching the objects of the projectively transformed squares in the image by a contour following process. We fit quadrangles to each object, and compute the intersection of the both diagonales (to use the centroid would be not correct). These points present now the second point set Q, see also Fig. 4.

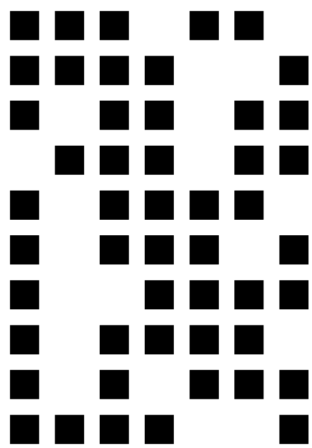

Fig. 3. Calibration grid without markers or landmarks

Even though we have strong projective deformations the algorithm works very well. With the correct references we can projectively transform the image Fig. 4 to the image Fig. 5. In the image Fig. 6 we can see a small occlusion. The algorithm has detected 96 percent of all references as correct references. The projective backtransformation to the image Fig. 7 works very well.

7 Conclusions and Future Work

In the present paper, we have developed a novel theory for affine point pattern matching (APPM). We have constructed a suitable cost function, and by min-

Fig. 4. Image I of the calibration grid

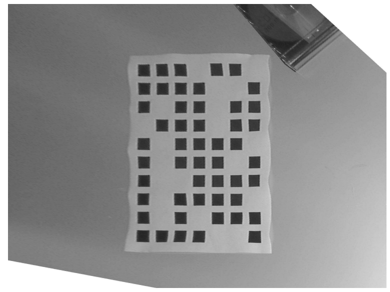

Fig. 5. Projectively transformed image I' with correct references

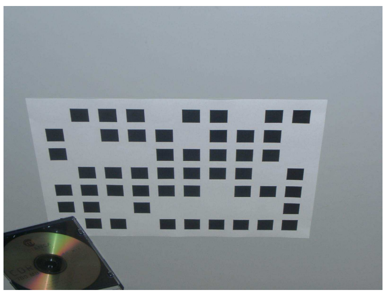

Fig. 6. Image I of the calibration grid with an occlusion

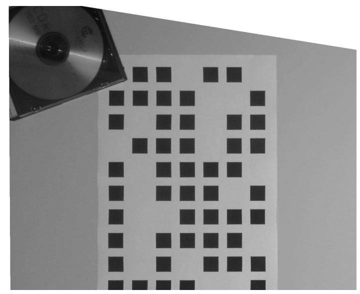

Fig. 7. But for all that it can be computed a correct projectively transformed image I'

imization of the costs we get the optimal permutation of the second point set. The cost for a point pair is the square of the geometric distance of the both points. For the minimization of the cost function we have used the Hungarian method. It is very surprising that the minimization of the costs is associated with an affine transformation of the points. The method is very robust against noise and can also be used for projective transformations appearing in real situations using real cameras. The robustness of the procedure against outliers and a different cardinality of the point sets can be improved by using a method which is more robust against outliers than the LAD method for the determination of the transformation.

On this note, one well known method in APPM is the accumulation-method or cluster-method (voting). In the simplest case, we select from the first point set P three points, and from the second point set Q also three points, compute the affine transformation and accumulate it in the six-dimensional accumulation-space (for projective transformations the accumulation is done in 8 dimensions). This is a robust algorithm, but it has the time-complexity $O(n^6)$. Using barycentric coordinates and geometric hashing the complexity can be reduced to $O(n^4)$, see e.g. [13],[5]. However, if we have solved our assigment problem A, then we can select 3 references, compute the affine transformation and accumulate it in the accumulation space. This algorithm has only the complexity $O(n^3)$. It is the same complexity as in the case of the Hungarian method.

References

1. T.S. Caetano, T. Caelli, D. Schuurmans, D.A.C. Barone: Graphical Models and Point Pattern Matching. PAMI 28, No, 10 (2006) 1646-1670
2. M. Carcassoni, E.R. Hancock: Point pattern matching with robust spectral correspondence. Proceedings CVPR 2000, pp. 649-655.
3. S.H. Chang, F.H. Cheng, W.H. Hsu, G.Z. Wu: Fast algorithm for point pattern-matching: Invariant to translations, rotations and scale changes. Pattern Recognition 30 (1997) 311-320.
4. G.S. Cox, G. DeJager: A survey of point pattern matching techniques and a new approach to point pattern recognition. Proc. of South African Symposium on Communications and Signal Processing 1993, pp. 243-248.
5. M.Costa, R. Haralick, L. Shapiro: Optimal Affine-Invariant Point Matching. Proceedings of the SPIE Conference on Applications of Artificial Intelligence, Orlando, Fl., 1989, pp. 515-530.
6. M. Dubuisson, A. Jain: A Modified Hausdorff Distance for Object Matching, Proc. 12 Int. Conf. on Pattern Recognition, Jerusalem 1994, pp. 566-568.
7. H.N. Gabbow, R.E. Tarjan: Faster scaling algorithms for network problems. SIAM J. Comput. 18 (1989) 1013-1036.
8. S. Gold, A. Rangarajan, C.P. Lu, S. Pappu, E. Mjolsness: New algorithms for 2D and 3D point matching: Pose estimation and correspondence. Pattern Recognition 31 (1998) 1019-1031.
9. C. Gope, N. Kehtarnavaz: Affine invariant comparison of point-sets using convex hulls and hausdorff distances. PR 40 (2007) 309-320.
10. M. Hagedoorn, R.C. Veltkamp: Reliable and efficient pattern matching using an affine invariant metric. IJCV 31 (1999) 203-225.
11. J. Hong, X. Tan: A new approach to point pattern matching. Proc. of 9th International Conference on Pattern Recognition 1988, Vol.1, pp.82-84.
12. H. Karloff: Linear Programming. Birkhauser Verlag, Basel 1991.
13. H.W. Kuhn: The Hungarian Method for the assigment problem. Naval Res. Logist. Quart. 2 (1955) 83-97.
14. Y. Lambdan, H. Wolfson: Geometric Hashing: A General and Efficient Model-Based Recongnition Scheme. Proceedings 2. ICCV, Tarpon Springs, MD, 1988, pp. 238-249.
15. L. Lovasz, M.D. Plummer: Matching theory. Amsterdam, North-Holland 1986.
16. B. Luo, E.R. Hancock: A unified framework for alignment and correspondence. Computer Vision and Image Understanding 92 (2003) 26-55.
17. C.H. Papadimitriou, K. Steiglitz: Combinatorial Optimization, Algorithms and Complexity. Prentice Hall N.J. 1982.
18. S. Ranade, A. Rosenfeld: Point Pattern Matching by relaxation. Pattern Recognition 12 (1980) 269-275.
19. J. Sprinzak, M. Werman: Affine point matching. Pattern Recognition Letters 15 (1994) 337-339.
20. H. Suesse, W. Ortmann: Robust matching of affinely transformed objects. Proc. ICIP 2003 , Barcelona, Proc. Part II, pp. 383-386.

21. K.Voss, H.Suesse: Affine Point Pattern Matching, Lecture Notes in Computer Science 2191, Munich 2001, pp. 155-162.
22. K.Voss, H.Suesse: Apareamiento de puntos y optimizacion lineal. Comunicacion tecnica I-01-06/05-06-2001, CC/CIMAT, Centro de Investigaciones en Matematicas Guanajuato, Mexico, 2001.
23. P.B. Van Wamelen, Z. Li, S.S. Iyengar: A fast expected time algorithm for the 2D-point pattern matching problem. Pattern Recognition 37 (2004) 1699-1711.
24. H.M. Wagner: Linear programming techniques for regression analysis. American Statistical Association Journal, March 1959.
25. Y. Zheng, D. Doermann: Robust Point Matching for Nonrigid Shapes by Preserving Local Neighborhood Structures. PAMI 28, No. 4 (2006) 643-649