

Feature Matching and Pose Estimation Using Newton Iteration

Hongdong Li and Richard Hartley

Research School of Information Sciences and Engineering,
The Australian National University,
ASSeT, Canberra Research Labs, National ICT Australia

Abstract. Feature matching and pose estimation are two crucial tasks in computer vision. The widely adopted scheme is first find the correct matches then estimate the transformation parameters. Unfortunately, such simple scheme does not work well sometimes, because these two tasks of matching and estimation are mutually interlocked. This paper proposes a new method that is able to estimate the transformation and find the correct matches simultaneously. The above interlock is disentangled by an alternating Newton iteration method. We formulate the problem as a nearest-matrix problem, and provide a different numerical technique. Experiments on both synthetic and real images gave good results. Fast global convergence was obtained without the need of good initial guess.

1 Introduction

In many vision applications, one often encounters the problem of estimating the relative geometric transformation (i.e., pose) between two feature point sets in two images. The most straightforward and widely-adopted way of doing it is first establish point-wise feature matches (correspondences) between the two feature sets and then estimate the transformation parameters. However, finding good correspondences is not easy, especially when two images are linked by some geometric transformations with unknown parameters.

Matching and estimation are two mutually interlocked problems: finding one is often the precondition for finding the other. On one hand, if the transformation is roughly known, then matching becomes easier. The local-correlation-based matching method will generally not work if no information about the transformation is given; on the other hand the transformation parameter can only be precisely estimated when correct matches are given. Thus matching and estimation form a chicken-and-egg dilemma.

This paper proposes a *global* method that is able to estimate the transformation parameters and find correct matches simultaneously. We formulate the matching-and-estimation problem as a *nearest matrix problem*, in particular to find an orthogonal *polar factor* matrix [8]. This compact formulation allows for numerically stable methods. We suggest a very neat *Newton root-finding method*, (the well known *tangent-line method*), which is well understood and has good

theoretical properties. Experimental results on both synthetic feature sets and real imagery gave encouraging results.

2 Matching as a Nearest Matrix Problem

2.1 Global Matching Methods

The majority of existing matching methods relies on local similarity of the feature points. One example is the correlation technique. These methods are well understood and widely adopted. However, the main disadvantages of these methods are that they are very sensitive to noise, and some crucial parameters (such as the size of neighborhood and correlation threshold) must be empirically chosen. Researchers have noted that global properties such as rigidity and exclusiveness are very important for matching. A number of global methods have been proposed. Among them, the spectral graph theory methods [5,3,4,9], global cost minimization methods [2,6], the epipolar constraint method [12], are some nice examples. Such global matching methods seem to be more promising than local methods. We propose a global matching method in this paper. By *global*, we mean that the best correspondences are the ones that minimize a global cost function. Encoded in this cost function are many global properties within and between the two feature sets. The matching problem is cast as a *nearest matrix approximation* problem. The special form of this nearest matrix encapsulates the global properties.

2.2 Nearest Matrix Problem

Suppose we are given two discrete feature sets in two images. For the moment let us assume that these two sets have an identical size of N , and that all feature points have matches. This restriction will be relaxed later.

We assign to each feature an attribute vector, which could be its pixel coordinates $[u, v]$ and/or the gray values within a local neighborhood $[g_1, g_2, \dots]$. Let it be **row vector** of dimension \mathbf{d} . Stacking all the vectors of each image together, we thus obtain two *feature matrices*, X and Y , of size $\mathbf{N} \times \mathbf{d}$, which describe the feature points in image 1 and image 2, respectively. Let us, for the moment, further assume that there is no significant geometric transformation between two feature sets, except for some jitters in coordinates. We use a permutation matrix P of size $\mathbf{N} \times \mathbf{N}$ to express the feature matching relations. A permutation matrix is a square matrix whose entries are either zero or one, and the entries of each row and of each column add up to one.

In the matching problem, the positions of one elements in P describe the correct matching between X and Y , that is, $X \leftrightarrow PY$. So, when all the correct matches are found, the following function J should reach its minimum:

$$J = \|X - PY\|_F^2 \rightarrow \min \quad (1)$$

This J is the global cost function for the matching problem. The particular P corresponding to the minimum value of J gives the correct matches. In fact,

such J is a distance measurement describing the proximity of two feature sets, and is known as the *standard Procrustes distance* [11]. From (1) and by some linear algebra manipulations, we have

$$\begin{aligned}
 \min \| X - PY \|_F^2 &\Leftrightarrow \min Tr((X - PY)^T(X - PY)) \\
 &\Leftrightarrow Tr(X^T PY) \\
 &\Leftrightarrow Tr(PYX^T) \\
 &\Leftrightarrow \min \| P - XY^T \|_F^2
 \end{aligned}
 \tag{2}$$

Defining $A = XY^T$, this problem becomes one of finding a real permutation matrix P such that

$$P = \operatorname{argmin} \| P - A \|_F^2
 \tag{3}$$

This is a typical nearest matrix problem. The formulation is more compact than (1) and has many benefits which will be explained later. Now, we reach the following result: *Given matrix A defined as above, find a permutation matrix P which is nearest to A. Then this P gives the best correspondences.*

Many methods use the doubly stochastic (\mathcal{DS}) property of the permutation matrix. By contrast, this paper places more emphasis on the orthogonality of P . That is, a permutation matrix must be an orthogonal matrix: $PP^T = I_n$.

Similar idea has been used by the spectral matching approaches such as [4] [9], but we find the orthogonality property is so strong that an iteration algorithm can be derived directly from it. This results in a very neat Newton root-finding method. Specifically, the best P is no other than the nearest (with respect to A) square root matrix of the identity matrix I . Moreover, according to our experiments, we found that the resultant matrix P is close enough to the ideal zero-one permutation, even when other constraints of a permutation matrix have not been imposed.

3 Matching and Estimation at the Same Time

When there are geometric transformations between the two feature sets, formula (1) no longer hold. However, we can easily modify it by accounting such transformation. We confine the transformation to be linear only. This is a limitation of our method, but not too serious, because it covers most commonly encountered transformations, such as Euclidean motion, or affine transformation. Denote the linear transformation by a matrix R , Equation (1) then becomes

$$J = \| X - PYT \|_F^2 \rightarrow \min
 \tag{4}$$

where $T_{d \times d} = \begin{bmatrix} R & 0 \\ 0 & I_{d-2} \end{bmatrix}$. The sub-identity matrix I in T is used for those non-coordinate attribute components of X and Y . Following similar derivations as in (2), equation (3) is modified to :

$$P_{n \times n} = \operatorname{argmin} \| P - XT^T Y^T \|_F^2
 \tag{5}$$

If the transformation T is known in advance, the matching problem can be solved by finding the nearest permutation matrix to the matrix P . On the other hand, if we already know the correspondences P , then the transformation T can be estimated by solving another nearest matrix problem: $T_{d \times d} = \operatorname{argmin} \|T - Y^T P^T X\|_F^2$, where T is subject to some special constraints that correspond to the particular transformation model. For example, in the case of a 2d rotation, R should be an orthogonal matrix with unit positive determinant. For the sake of computational simplicity, T can also be approximated as $T \approx Y^T P^T X$ or $T \approx Y^\dagger P^T (X^T)^\dagger$, where \dagger denotes the pseudo-inverse. In our problem, both the transformation and correspondences are not known beforehand. In order to solve this dilemma, we use an alternating iteration approach.

4 The Proposed Newton Iteration Algorithm

The structure of the alternating iteration algorithm for simultaneously solving P and T is given as follows:

Begin

Initialize T_0 .

For $k = 0, 1, 2, 3, \dots$, do the following until convergence:

$$P_{k+1} = \operatorname{argmin} \|P_k - X^T T_k^T Y\|_F^2 \tag{6}$$

$$T_{k+1} = \operatorname{argmin} \|T_k - Y^T P_k^T X\|_F^2 \tag{7}$$

End.

At each iteration step, both P and T should also be adjusted to comply with certain constraints. Since (6) and (7) generate a non-increasing sequence, convergence is ensured. There is actually no preference as to which one is performed before the other, however, by considering our application, the number of feature points is often greater than the number of parameters of T . Thus it is more reasonable to make an initial guess of T and first solve for P .

Equation (7) can be directly solved as a minimization problem, as in [6]. Alternatively, we can simply apply the approximations of the last section. The latter method is simpler and more effective. In our implementation, the orthogonal condition is realized by a matrix version Newton root-finding algorithm. Besides the nearest matrix formulation, this Newton matrix algorithm is another distinct feature of our method.

4.1 Matrix Root Problem and Matrix Newton Method

Consider the problem of finding the real roots of a nonlinear scalar equation $f(x) := x^2 - 1 = 0$. Newton's tangent line method, $x_{k+1} = x_k + f(x_k)/f'(x_k)$, gives the iteration formula $x_{k+1} = (x_k + x_k^{-1})/2$.

In our problem, P is a square root matrix of the identity matrix I . To solve such *matrix root problem*, Higham [8] proposed a numerical iteration method by mapping the original scalar form Newton root-finding method to a matrix form. Analogously, we obtain $P_{k+1} = (P_k + P_k^{-T})/2$.

Of course, for rigorous theoretic justification of this formula, the reader is referred to [8], which has also proven its quadratic convergence. Now the minimization of (6) is realized by the above Newton iteration with $P_0 = \mathbf{exp}(X^T T^T Y)$ as the starting point. Here the function \mathbf{exp} can be thought of a nonlinear kernel function, one of which purposes is to ensure the positivity of the resulting entries. As it may be a singular matrix we actually used pseudo-inverse instead. Experiments showed that the final P was very likely to have entries that were very close to zero or one. Notice that we have not enforced any doubly stochastic constraints upon P .

Similarly, when the transformation is simply a 2D Euclidean motion, e.g, a rotation matrix R , it can also be solved by such Newton iteration, namely, $R_{k+1} = (R_k + R_k^{-T})/2$. Besides, we enforce a constraint $\mathbf{det}(R) = +1$ to make it a proper rotation matrix. For this purpose, we use a simple trick by swapping any two rows of R when needed. Our experiments showed that by this constraint we are able to reach the global optimum even from arbitrary initial guesses.

4.2 Some Practical Considerations

Compare our algorithm with the annealing-based softAssign or SoftPOSIT method. If we roughly think of the orthogonal matrix P_k as a high-dimensional rotation, then the effect of each iteration is to gradually rotate the correspondences towards a consistent configuration. On the other hand, the doubly stochastic annealing strategy consists of first preferring a uniformly distribution of correspondence credits, and then choosing one winner that has the most response according to a winner-take-all consensus.

In practice, we often meet cases where the sizes of the two feature sets are not equal, or where some feature points are occluded or fail detected (outliers). These cases will not degrade our method, because to deal with them we need only make small modifications of the proposed algorithm.

When set sizes are unequal, matrix P is no longer square and no longer an orthogonal matrix, but it is still an orthonormal matrix. Therefore Newton method is still applicable. When there are outliers, we can augment P with one row and one column. The newly added row or column corresponds to a so-called *slack feature* that is able to match any point.

5 Experiments and Results

We carried out several experiments on both synthetic data and real images. The first experiment tested the matching performance with noisy data. We generated a discrete point set with $N = 79$ feature points along the contour of a pine tree shape. This is taken as the model set. The other sets were generated by randomly

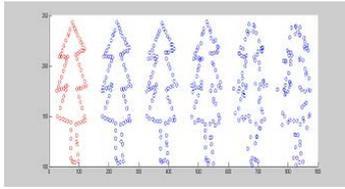


Fig. 1. Pine model set, and data sets with various noise levels, ($\sigma = 0.5, 1.0, 2.0, 3.0, 4.0$)

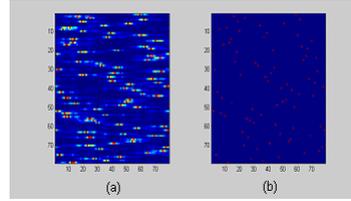


Fig. 2. The correlation matrix A before (left), and after Newton iteration (right)

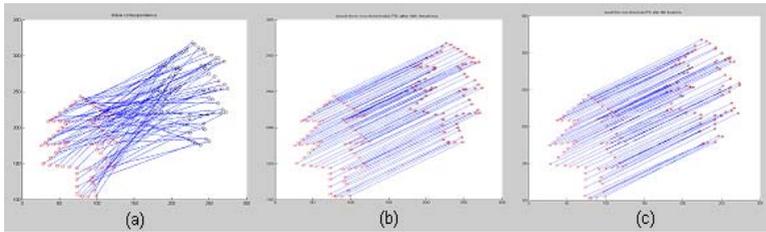


Fig. 3. Feature matching results (a) random initial correspondences; Results (b) $\sigma = 1.0$ (c) $\sigma = 4.0$

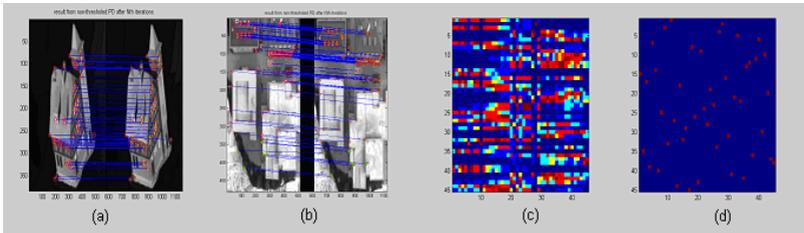


Fig. 4. Matching results on real images

perturbing the coordinates of the model set. Both are shown in figure-1. These perturbations were drawn from a zero-mean isotropic Gaussian distribution with standard deviation σ in the range $[0.5 \ 4.0 \text{ pixels}]$. Note that in particular the perturbations are rather large with respect to the minimum distance between model points (only about 3-4 pixels). In other words, the perturbations caused large shape distortions in the dense model features. This is a challenging situation for local correlation based matching algorithms.

Figure-2 (left) shows (in pseudo-color graph) an initial matrix A corresponding to a random initial correspondence. Matrix A is the input to our Newton algorithm. The resultant nearest permutation matrix P is depicted in figure(2)-(right). Using this P , we obtained the matching results shown in figure-3. All the results are fairly good. No serious bad matches were found. Figures-4 (a) and (b)

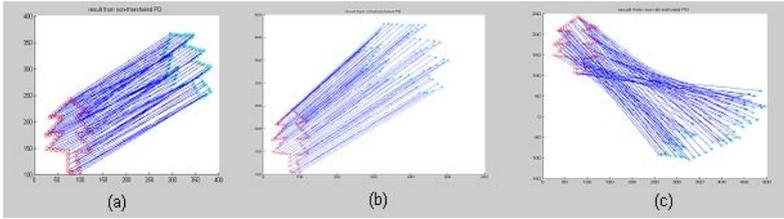


Fig. 5. Simultaneous matching and transformation estimation results; (a) rotation: real 39.0 degree, estimated 39.01 degree; (b) affine (only test 4-parameters): real [2.2, -0.8; 0.6, 1.1], estimated [2.1988 -0.7999; 0.6001 1.0998]; (c) affine (only test 4-parameters): real [+2.2, -0.8; +0.6, -1.1], estimated [2.2012, -0.7993; 0.5978, -1.0996]

show two test results on real images. Feature points were detected using a Harris corner detector. A subset was then manually selected. The attribute vector of each point consisted of its coordinates and the gray values of a 3×3 neighborhood. Also note that the feature number is rather large (Building: 121 points; House: 45 points), and are densely located. Remember that we actually used two different images that were drawn from a video sequence. Therefore their viewpoints and feature point coordinates are all different. We consider these differences as noise, and note that the matching results are still very satisfactory and encouraging. Figure 4(c) illustrates the initial values of entries of matrix A for the house images, and figure-4(d) shows the resulting matrix A after the Newton iteration. It is seen that this matrix has converged to a binary 0-1 matrix. In the second set of experiments, we tested the performance of simultaneous matching and transformation estimation. Figure 5 gives the matching and transformation estimation result on the synthetic graphs. For all the experiments conducted in figure 5, we did not make any initial guess for the transformation, except for knowing their models (Euclidean or affine). Our method has shown to be more robust, in the sense of global convergence, it seems less likely to be trapped in a local minima. In all experiments, the convergence were rather fast. Actually, the total convergence rate of our algorithm is no longer quadratic, but linear because of the alternative operation.

6 Discussion and Conclusions

Feature matching is no more than finding a permutation matrix. Many existing algorithms (such as the SoftAssign method and SoftPOSIT method) implement this by finding a doubly-stochastic matrix as an approximation. For $N \times N$ real matrix, there are $N!$ possible choices. In order to find the one that best explains the input data, they often adopt an annealing procedure. The major difficulty is that the convergence is relatively slow and yet may get trapped in local minima. Notice that a doubly stochastic matrix in general is not an orthogonal matrix. Rather than using the doubly stochastic property in each search step, this paper uses a much stronger condition of the orthogonal property.

The proposed algorithm can be thought of *projecting* a matrix into the nearest *orthogonal manifold*. Because for $N \times N$ real matrix there are at most N orthogonal matrices, in turn, the chance to quickly find the optimal solution increases. This idea was used by many spectral graph matching algorithms[4][5][12], where they apply the SVD factorization. But here we suggest using Newton root-finding method, which has better numerical properties, and is essentially equivalent to finding the matrix **polar factor** only ([9]), therefore saves many computations. In other words, a full SVD decomposition is no need here.

Our method is a *global* matching method because many global Gestalt factors, such as proximity, similarity and exclusiveness, have been encoded in the nearest matrix formulation in a concise way. For cases where mismatching points exist, we consider using EM-like algorithm to resolve it. By applying a specific parametric model this method could be easily extended to the matching of non-rigid deformation objects. These two topics will be investigated in the future.

Acknowledgements. National ICT Australia is funded through the Australian Government's Backing_Australia's_Ability_Initiative, in part through the Australian Research Council. The real images were taken from CMU-RI website. The authors are grateful to the anonymous reviewers for very valuable suggestions.

References

1. P.David, D. Dementhon, R.Duraiswami, H.Samet, SoftPOSIT: Simultaneous Pose and correspondence determination, ECCV-2002. Denmark. 2002.
2. S.Gold, S.Pappu, C.Lu, A.Rangarajan, E.Maolsness, New algorithm for 2D and 3D point matching: Pose estimation and correspondence, PR(31), pp1019-1031, 1998.
3. Scott, G.L., and Longuet-Higgins, H.C., An Algorithm for Associating the Features of Two Images, Proc. Royal, Soci. London B-244, 1991, pp. 21-26.
4. Shapiro, L.S., and Brady, J.M., Feature-Based Correspondence: An Eigenvector Approach, IVC (10), No. 5, June 1992, pp. 283-288.
5. S.Scalaroff, A. Pentland, Modal Matching for correspondence and recognition, T-PAMI 17(6), 1995, pp545-561.
6. Maciel and Costeira , A global solution to sparse correspondence problems, T-PAMI, 25-2, 2003
7. J.Kosowsky,a.Yuille, The invisible hand algorithm: solving the assignment problem with statistical physics, Neural networks, vol-7,1994, pp477-490.
8. N. Higham, Stable iteration for the matrix square root, Numerical algorithm 15, 1997, pp 227-242.
9. Carcassoni, M, Hancock, E., Spectral correspondence for point pattern matching, PR(36), No. 1, pp. 193-204. January 2003,
10. Anand Rangarajan, Haili Chui and Fred L. Bookstein, The Softassign Procrustes Matching Algorithm, James Duncan and Gene Gindi, editors , Information Processing in Medical Imaging, 1997.
11. D.Kendall.Shape Manifolds, Procrustean metrics and complex projective spaces, Bulet. London.Math.Society, 16:81-121,1984.
12. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd edition, Cambridge University Press, 2004.