A NEW AND COMPACT ALGORITHM FOR SIMULTANEOUSLY MATCHING AND ESTIMATION

Hongdong Li and Richard Hartley

Department of System Engineering , RSISE, Australian National University
Canberra Research Laboratory, National ICT Australia,Ltd.
lihd@zju.edu.cn ,{hongdong.li,richard.hartley}@anu.edu.au

ABSTRACT

Feature matching and transformation estimation are two fundamental problems in computer vision research. These two problems are often related and even interlocked, solving one is solving the other's precondition. Such makes them hard to solve. In order to overcome such difficulty, this paper presents a new and compact algorithm where less than 10 lines of matlab codes suffice. We show that the solutions of correspondence and transformation are merely two factors of two Grammian matrices, and can be worked out with factorization method. A Newton-Schulz numerical iteration algorithm is used for such factorization. The two interlocked problems are solved in an alternate(flip-flop) way. The effectiveness and efficiency are illustrated by experiments on both synthetic and real images. Global and fast convergence attained even start from random chosen initial guesses.

1. INTRODUCTION

In digital video processing and computer vision research, ones often meet the problem of transformation estimation. This problem can be formally stated as: given two feature sets(for example, the interest point sets) drawn from two images, which are related by a certain geometric transformation with unknown parameters (up to some noise), the problem is to estimate these parameters.

Usually, this problem is solved by such scheme: first establish point-wise features correspondences between the two images, then establish a set of equations with those transformation parameters as unknowns, solving them, and you will get the proper answers.

This scheme is simple, straight-forwards, and has been widely adopted. However, finding the correct correspondences is by no means a trivial task [5][7].

It seems to be an interlock relationship between *feature correspondence* and *transformation estimation*. On one side, transformation estimation requires good correspondences given beforehand, while on the other side, many traditional correspondence methods often fail if nothing is known about the transformation. In other words, good knowledge of transformation will reduce the difficulty of correspondence.

This paper proposes a compact and efficient algorithm to unlock this inter-lock. The two problems of correspondence and estimation are solved simultaneously by using less than 10 lines of matlab codes. The main idea is to alternately update their intermediate solutions represented by two matrix factors of two Grammian matrices constructed from the input feature sets. A Newton-Schulz numerical iteration method is used at each iteration step for solving the matrix factorization problem. Both experiments on synthetic and real images have shown good results. Our algorithm is formulated in matrix form, therefore it can be directly applied to higher dimension problems without any modification.

In the sense of using an alternate iteration method to solve the correspondence-and-estimation problem, the proposed algorithm shares some features with *SoftAssign* method[1], *SoftPOSIT* method[2] and *Sparse Correspondence* method[3]. But ours much differs in either problem formulation or solution technique. In their algorithms, concave minimization or deterministic annealing procedures were applied. In addition, experiments showed better performance over the others in the sense of faster and globally convergence.

2. PROBLEM FORMULATION

Given two images, image#1 and image#2, each has a feature(point) sets of size M and N, respectively. Let us for the moment assume that (i) M=N and (ii) the correspondences between the two feature sets exist and is bijective(one-to-one). These assumptions will simplify algorithm explanation, and are not too restrictive because we will relax them later.

Each feature point is represented by a row feature vector of dimension d, here d =2. Stacking all feature vectors of

This work is partially supported by NSFC , Grant No. 60105003.

each image together, we get two *feature matrices* of size N xd.Denoting them *feature matrix* **X** and **Y**, of image#1 and image#2, respectively, we have:

$$X = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \dots & \dots \\ x_{i1} & x_{i2} \\ \dots & \dots \\ x_{N1} & x_{N2} \end{bmatrix}, \ Y = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \\ \dots & \dots \\ y_{j1} & y_{j2} \\ \dots & \dots \\ y_{N1} & y_{N2} \end{bmatrix} \quad (1)$$

Notice that so far we do not know whether or not the index of i of image#1 and the index j of image#2 are corresponding, however, we can express the correspondences by a permutation matrix **P** of size NxN. By definition, a permutation matrix is a square discrete-valued matrix whose entries are either zero or one, and the entries of each row is added up to one, and so as to each columns. In our scenario, the positions of entries 1s in **P** describe the correct correspondences between **X** and **Y**, i.e, **X**↔**PY**.

If a geometric transformation exists between the two feature sets, though with unknown parameters, we can as well represent the transformation by a parametric model. We use a symbol **R** for this transformation. Most commonly used image transformations are linear transformations, such as the Euclidean transformation, the similarity transformation, and the affine transformation, etc. Therefore we consider **R** a linear transformation represented by a non-singular matrix, and we describe the transformation in a linear form, **X**↔**YR** , where **X** and **Y** are matched feature matrices.

Combining both transformation matrix **R** and correspondence matrix **P**, we obtain:

$$\mathbf{X} = \mathbf{PYR} \quad (2)$$

This equation expresses the relationship between two feature matrices under transformation **R** and correspondence **P**. Similar equations have been suggested by [1][2][3][8]. Paper[8]also gives a condition number analysis for such feature point-matching. The **X** and **Y** are two known matrices that can be measured from two input images, the **P** and **R** are two unknowns to be determined. This forms a *matrix factorization* formulation, say, if by some means we can factorize matrix **X** into the product of matrices **P** and **Y** and **R** , where **P** is a NxN permutation and **R** is a 2x2 rotation, then we solve the correspondence-and-estimation problem. However, because there are two unknowns in one equation, such problem is not easy to be solved. Our approach is to construct two Grammian matrices, one outer product matrix and one inner product matrix, denoted by $\mathbf{G}_1$ and $\mathbf{G}_2$, respectively.

$$\mathbf{G_1} = \mathbf{XY^T}, \mathbf{G_2} = \mathbf{Y^T X} \quad (3)$$

Substitute (2) into (3), we obtain,

$$\begin{aligned} \mathbf{G_1} &= \mathbf{XY^T} = \mathbf{PYRY^T}, \\ \mathbf{G_2} &= \mathbf{Y^T X} = \mathbf{Y^T PYR} \end{aligned} \quad (4)$$

Now consider two ideal cases:

(a) No transformation :

In this case, there is no geometric transformation between two feature sets, so we can set **R** = $\mathbf{I}_2$, where $\mathbf{I}_2$ is 2x2 identity matrix. If the two sets are matched, i.e, their coordinates must be precisely superimposed, then we have, (the $\sim$ is used to denote ideal cases.)

$$\mathbf{\tilde{G}_1} = \mathbf{XY^T} = \mathbf{PYY^T} \quad (5)$$

It is obvious that the $\mathbf{YY}^T$ is a symmetric positive-definite matrix, therefore equation (5) is equivalent to the following minimization problem:

$$\underset{P}{min} \left\| \mathbf{\tilde{G}_1} - \mathbf{PYY^T} \right\|_2 = \underset{P}{min} \left\| \mathbf{P} - \mathbf{\tilde{G}_1} \right\|_2 \quad (6)$$

, where $\|\bullet\|_2$is the 2-norm.

Eq.(6) states, if there is no transformation (or approximately say, the transformation has been roughly estimated and has been compensated), then the best correspondence matrix **P** is the *permutation matrix* that is *closest* to $\mathbf{G}_1$. In fact, the form of (5) is a matrix *polar decomposition*[6], says, because $\mathbf{YY}^T$ is symmetric, **P** is the unitary polar factor of $\tilde{\mathbf{G}}_1$.

(b) Known correspondence:

In this case, because the correspondences are already known, we can re-arrange one of the feature matrices (by row reordering) such that X and Y have identical row order. Thus we have **P** = $\mathbf{I}_N$ and in such case let

$$\mathbf{\tilde{G}_2} = \mathbf{Y^T PYR} = \mathbf{Y^T YR} \quad (7)$$

Similar to (a), we know that (7) is equivalent to:

$$\underset{R}{min} \left\| \mathbf{\tilde{G}}_2 - \mathbf{Y^T YR} \right\|_2 = \underset{R}{min} \left\| \mathbf{R} - \mathbf{\tilde{G}_2} \right\|_2 \quad (8)$$

Eq.(8) states, if correspondences are known (or approximately say, has been roughly estimated), then the transformation matrix **R** is the *closest rotation matrix* to $\mathbf{G}_2$. In fact, the form of eq.(7) is a matrix polar decomposition, says, **R** is the unitary polar factor of $\tilde{\mathbf{G}}_2$.

Here, (a) and (b) have shown a symmetry or duality relationship in correspondence matrix **P** and transformation matrix **R**. Solved one of them, the other is easily solved through a polar factorization procedure or closest matrix finding procedure.

However, remember that neither **P** nor **R** is known in practice, and they are interlocked. This makes the problem difficult. To our knowledge, no analytic closed form solution available now. Therefore in this paper, we propose an alternating(flip-flop) iterative algorithm to solve it. The algorithm will be detailed in section-3. Section-4 will give some experimental results to show the effectiveness and efficiency of our algorithm.

3. THE PROPOSED ALGORITHM

For N features, there are in total N! numbers of possible permutations, an exhaust combinatory discrete searching procedure is obvious impractical. The resolvent is to relax the original discrete problem to a continuous equivalent form. There are two distinct continuous equivalent forms that a permutation matrix may take. One is a doubly stochastic form, and the other is an orthogonal form. Paper [1][2][3] employed Lagrange multiplier method to impose the doubly stochastic constraints via a (continues) deterministic annealing procedure. Notice that in general a doubly stochastic matrix may not be orthogonal, while any permutation must hold orthogonality. Therefore, rather than applying the doubly stochastic property, we apply the orthogonal property to guide the searching procedure. More specifically, *a permutation matrix must be an orthogonal matrix,* or,

$$\mathbf{P}\mathbf{P}^{\mathbf{T}} = \mathrm{I}_{\mathbf{N}} \tag{9}$$

We found that this property is so strong that we can directly derive our continuous iteration procedure merely depending upon it. In other words, the permutation matrix that we seek is the square-root-matrix of $\mathbf{I}_N$ that also satisfies (6). Such effort results in an elegant Newton-Schulz iterative root-finding method.

3.1 Newton-Schulz iteration

Higham[4]suggested several numerical iteration methods for *matrix square root problem*, which are simply analogues from the original scalar Newton root–finding method (i.e.,the tangent-line method) to matrix form. Among them, the Newton-Schulz iteration is as follows:

$$\mathbf{P_{k+1}} = \mathbf{P_k}(\mathbf{3}\mathrm{I}_{\mathbf{N}} - \mathbf{P_k^T P_k})/\mathbf{2} \tag{10}$$

This method has good theoretic properties and it converges quadratically for

$$\left\|\mathbf{P_0}\mathrm{P}_{\mathbf{0}}^{\mathbf{T}} - \mathbf{I}\right\|_{\mathbf{2}} < \mathbf{1.0} \tag{11}$$

Thus, the minimization of (6) can be realized by the iteration of (10) with **exp(**$\mathbf{G}_1$) as its starting point. The **exp**() is an exponential function acting on each entries of $\mathbf{G}_1$, whose purpose is to ensure that all the entries are positive,therefore the Newton root-finding algorithm may start from a reasonable point. And, of a little surprise, experiments showed that the final **P** is very likely to have entries that are very close to zero or one and sum to one (in rows and columns), notice that no discrete value constraint has ever been imposed. When the transformation matrix **R** is a rotation (hence orthogonal), it can also be solved by another Newton iteration, as the duality suggested. However, for the sake of computation simplicity, in our implementation, **R** is approximated from the current **P**, i.e., $\mathbf{R} \approx \mathbf{G_2} = \mathbf{Y^T P X}$, rather than Newton_Schulz iteration.

3.2 The proposed algorithm

The whole algorithm is thus summarized as:

Algorithm: Flip-flop Newton-Schulz Algorithm

BEGIN.
Guess R_0
For k = 0 to K, **do**

$$\mathbf{P_k} = \mathbf{exp}(\mathbf{X R_k^T Y^T})$$

For j = 1 to J, **do**

$$\mathbf{P_k} = \mathbf{P_k} / \left\|\mathbf{P_k^T P_k}\right\|_{\mathbf{2}}$$

$$\mathbf{P_{k+1}} = \mathbf{P_k}(\mathbf{3I_N} - \mathbf{P_k^T P_k})/\mathbf{2}$$

End do

$$\mathbf{R_k} = \mathbf{Y^T P_k X}$$

End do
END.

Note, less than 10 lines matlab codes are sufficient to fully implement our algorithm.

In practice, when the two set sizes are unequal,say,**M**≠**N**,or there are unmatchable outliers matrix **P** is no longer a square and orthogonal matrix, but it's columns or rows' orthonormality still hold. In this case, we augmented **P** with one row and one column. The newly added row or column corresponds to a so-called slack feature, a term being used in linear programming techniques, that can match any other features or outliers. Therefore, our algorithm is still applicable.

4. EXPERIMENTS

We have carried out several experiments on both synthetic data and real images to test the performances of our algorithm.

In the first experiment we tested on synthetic datasets. We drew a planar shape that has N =79 discrete points, and used them as feature set#1. Feature set#2 were generated by Gaussian random perturbing and geometric transforming of the coordinates of feature set#1. The perturbations(with standard deviation of 2.0∼3.0 pixel) are relatively large with respect to the minimum distance between un-perturbed feature points (only about 3∼4 pixel). In other words, large shape distortions exist in feature set#2. This is rather challenge for most conventional correspondence methods that are based on local search and correlation.

Figure-1 gives the result on the synthetic data. Note that the estimated parameters are very accurate, and robust to noise. A more important fact is that: for all the experiments we conducted (including those on real imagery), though we did not use any special guess of the parameters as initial condition, the algorithm always converged. Moreover, the converge speed is rather fast. A typical setting is: for outer loop K, 7∼10 is enough; for inner loop J , 30∼50 suffices.

Figure-2 (a) and (b) show the results on real images. We used image pairs of House sequence and Building sequences of CMU-RI VASC dataset. The feature points were detected by a Harris corner detector, and manually selected a subset. Here, the feature vectors are of dimension 11. Besides coordinates, we added the nine gray values of its 3x3 neighborhood of each feature point. The two images were drawn from two different frames of the video sequence; therefore their viewpoints and feature point coordinates are all different. These differences are regarded as noise. Note that the matching results are still very good. Figure1(c) gives the pictures of matrix $\mathbf{G}_1$ before and after iteration (using the House sequence). Note that when the iteration converged, $\mathbf{G}_1$ really becomes a permutation matrix.

5. CONCLUSIONS

We have proposed a compact algorithm that is able to simultaneously match two feature sets and estimate transformation parameters using less than 10 lines of codes. We achieved this by considering two Grammian matrices from two feature sets. A Newton-Schulz numerical algorithm is proposed to solve the problem. Matching and estimation are updated in an alternate way, and finally converge to their correct solutions. The convergence is faster, and more global, in the sense that it is less easily get trapped into local minima. This algorithm can directly be applied to higher-dimensional problems(such as 3D mesh model alignment, molecular structure conformation,etc.) without modification.

Acknowledgements The authors much appreciate K.Yang, R.Orsi,K.Sim for insightful discussion. Thanks given to anonymous reviewers who gave us valuable suggestions.

6. REFERENCES

[1]. 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.

[2]. P.David, D.Dementhon, R.Duraiswami, H.Samet, SoftPOSIT: Simultaneous Pose and correspondence determination, ECCV-2002. Denmark. 2002.

[3]. Maciel and Costeira , A global solution to sparse correspondence problems, T-PAMI, 25-2, 2003

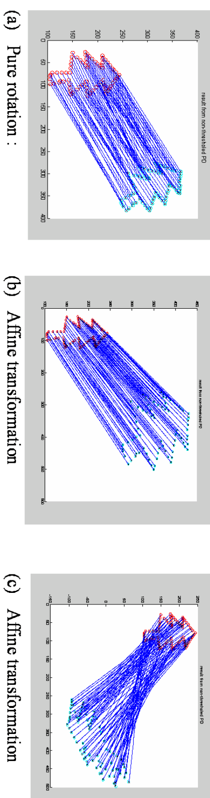
[4]. N. Higham, Stable iteration for the matrix square root, Numerical algorithm 15, 1997, pp 227-242.

[5]. R.Hartley, A.Zisserman, "Multiple view geometry-in computer vision ", Cambridge University Press, 2000.

[6]. Hongdong Li, R.Hartley, Correspondence from polar decomposition, Internal Technical Report, ASST-NICTA, RSISE-ANU. Oct. 2003.

[7]. Carcassoni, M, Hancock, E., Spectral correspondence for point pattern matching, PR(36), No. 1, pp. 193-204. January 2003

[8] Kenney, C.S.; Manjunath, B.S.; Zuliani, M.; Hewer, G.A.; Van Nevel,A condition number for point matching with application to registration and postregistration error estimation A.; IEEE T-PAMI, Vol 25 no.11,Pp.1437 -1454,Nov. 2003

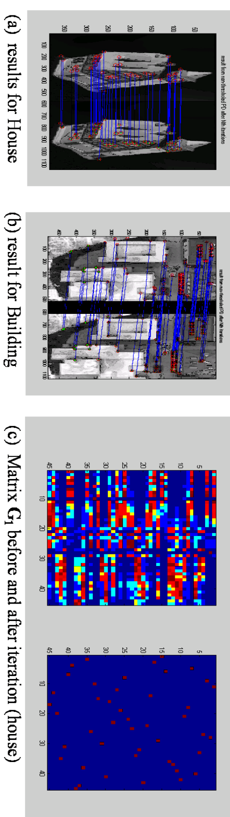


(a) Pure rotation : true parameter : 39.0 degree estimated: 39.01 degree

(b) Affine transformation true parameters: [2.2, -0.8; 0.6, 1.1] ; estimated: [2.1988 -0.7999; 0.6001 1.0998]

(c) Affine transformation true parameters: [2.0, -0.8; 0.6, -1.1] estimated : [2.0112, -0.7983; 0.5978, -1.0996]

Figure-1: matching-and-estimation results on synthetic data



(a) results for House

(b) result for Building

(c) Matrix $\mathbf{G}_1$ before and after iteration (house)

Figure-2 : Matching results on real images.