Subspace matching: Unique solution to point matching with geometric constraints

Manuel Marques, Marko Stošić and João Costeira

Institute for Systems and Robotics - Instituto Superior Técnico
Av. Rovisco Pais, 1, 1049-001 Lisboa PORTUGAL

{manuel,mstosic,jpc}@isr.ist.utl.pt

Abstract

Finding correspondences between feature points is one of the most relevant problems in the whole set of visual tasks. In this paper we address the problem of matching a feature vector (or a matrix) to a given subspace. Given any vector base of such a subspace, we observe a linear combination of its elements with all entries swapped by an unknown permutation. We prove that such a computationally hard integer problem is uniquely solved in a convex set resulting from relaxing the original problem. Also, if noise is present, based on this result, we provide a robust estimate recurring to a linear programming-based algorithm. We use structure-from-motion and object recognition as motivating examples.

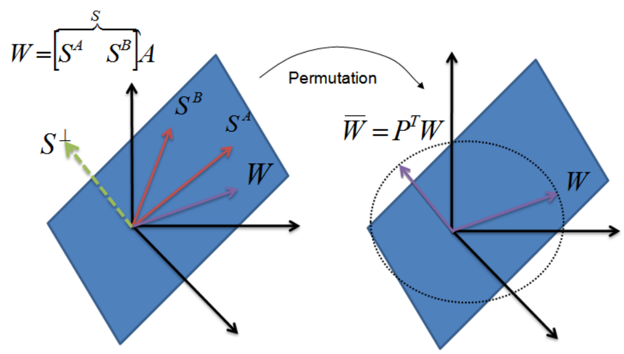

Figure 1. Feature matching constrained to a linear subspace: Feature vector $\mathbf{W}$ is constrained to the subspace generated by $\mathbf{S}^A$ and $\mathbf{S}^B$. The observation results from a linear combination of this base followed by a permutation of its entries. The objective is to infer the sorting knowing the original subspace.

1. Introduction

It is common knowledge that visual tasks like object recognition, shape reconstruction or image and motion segmentation may become from very hard to compute to very simple, by finding the adequate representation for the data or model. Adjoint to this, assigning correspondences between several images or between images and models is essential to implementing those tasks, as in locating objects in images, computing depth, tracking and recognizing objects in images.

Representing images and objects by linear subspaces, or using them as constraints, is almost "omnipresent" in computer vision: Eigenfaces, shape-from-motion, shape from shading, sound, illumination, style and content and optical flow estimation, to list a few fields of application, where linear (bilinear) models are extensively used with great success.

This work was partially supported by the Fundacao para a Ciencia e Tecnologia (ISR/IST pluriannual funding) through the POSC Program that includes FEDER funds and J. Costeira is also supported by project FCT - PTDC/EEA-ACR/72201/2006.

In this article we tackle the problem of finding the correspondences between points clouds (or feature vectors) constrained to lie on a subspace. As illustrated in figure 1, the problem here is to sort entries of a vector (or columns of a matrix) $\overline{\mathbf{W}}$, such that the permuted vector lies on the linear subspace resulting from the span of $\mathbf{S}$. In other words, observations result from a linear map of $\mathbf{S}$ followed by a permutation of its coordinates. Note that this map can be rank deficient (like a camera projection).

In the absence of noise, this problem can be solved in polynomial time [15]. However, with noise (or data which deviates from the model), it becomes intractable. In this paper we reveal a striking fact: The combinatorial (hard) problem of finding a permutation that "best" fits one vector with unsorted coordinates into its underlying subspace, can be obtained by minimizing a convex function over a convex set, which solution is a permutation. The problem complexity will depend on the relative dimensions between image and model, but will always lead to the execution of one, or a finite number of linear programs.

Even though this is a general result, we will focus, in the

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

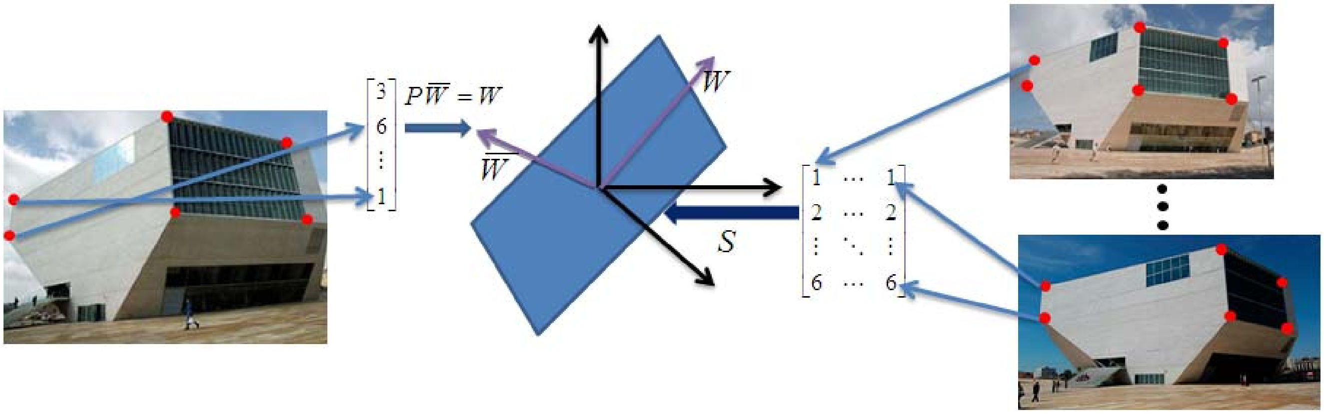

Figure 2. Subspace matching problem: Finding the correct permutation matrix $\mathbf{P}$ such that the resorted observed feature vector $\mathbf{P}\overline{\mathbf{W}}$ lies on the subspace $\mathbf{S}$ generated by the model. Note that the subspace can be computed from the 3D shape or from several image correspondences.

fields of structure-from-motion and object recognition with affine cameras as an application framework.

1.1. Related Work

Looking at the aforementioned subjects, several algorithms have been used to solve the object recognition problem from single images - feature/appereance and geometric approaches. Unlike 2D-2D recognition problem [14], a common strategy to recognize a 3D object from one single 2D image is solving a matching problem between two point clouds, because there are no general invariants between subspaces of different dimensions, e.g. 3D-2D [4, 10]. In [1], the *shape context* descriptors (orientation and distance histograms) are used to compute the two sets correspondence, using linear programming. The work presented in [2] allows us to compute the match between two 2D views of a 3D object using appearance information (feature descriptors) and geometric constraints, e.g. proximity between points. Smooth motion is required and the algorithm's convergence is not guaranteed because the solution is found by solving a linearized version of an integer quadratic problem. In [16], the authors build a cost function based on geometric and appearance information and find the solution through graph-cuts.

Based on geometric constraints, the correspondences between two 2D and 3D sets is computed by an highly nonconvex optimization problem, which is solved using deterministic annealing [7]. This approach requires a rough alignment between the two sets. In [6], the 3D-3D matching points is estimated by ICP-like algorithms. The 2D case is also discussed and the Levenberg-Marquardt algorithm is used to find the solution. An EM-type algorithm is proposed in [5] but, such as [6], the convergence is not guaranteed and some solutions can correspondent to local *minimae*. A quadratic optimization problem is suggested in [13] to find the 2D-2D matching solution. The authors use a newton-based algorithm to minimize the cost function, but a global minimum may not be found. A global solution for the 3D-3D correspondence can be found through a *branch-and-bound* algorithm [9], a similar strategy used in [3, 12]. In [12], the proposed RANSAC-based approach is a computationally expensive method. Imposing rank constrains through a linear program [11] is another approach to reach the global optimum. However, like some of the mentioned methods, the solution given by the algorithm depends critically on the initial estimate. The algorithm proposed in [17] uses a QR factorization approach to compute the correspondence between two subspaces. In this case, the subspaces' dimension must be equal, e.g. 2D-2D and 3D-3D.

In sum, the state of the art methods do not guarantee global convergence [5, 6, 11, 13], global optimum solution [15] or if they satisfy these two requirements are computationally heavy [3, 9, 12]. As in [9, 13, 17], our approach does not deal with outliers, however we provide conditions and an explicit solution that is always attained and a quite efficient algorithm to compute it.

2. Problem formulation

As said before and shown in figure 2, we consider the case of an object characterized by a 3D point cloud, observed by an affine camera. Recognizing the object in this 2D view requires the correspondences between the two sets of features (3D object - 2D projections).

So, given the known 3D shape $\mathbf{S}$, composed by N points $\mathbf{S} \in \mathbb{R}^{N\times 3}$, and a set of N 2D unsorted projections $\overline{\mathbf{W}} \in \mathbb{R}^{N\times 2}$, resulting from a linear map of $\mathbf{S}$ (translation can be removed by centering the data), the correspondence problem can be solved by computing the $N \times N$ permutation matrix $\mathbf{P}$ such that

$$\overbrace{\begin{bmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & z_N \end{bmatrix}}^{\mathbf{S}} \overbrace{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}}^{A} = \mathbf{P} \overbrace{\begin{bmatrix} u_1 & v_1 \\ \vdots & \vdots \\ u_N & v_N \end{bmatrix}}^{\overline{\mathbf{W}}} \quad (1)$$

where A is an unknown (camera) transformation. Note that the subspace of interest is the column space of $\mathbf{S}$ (we use the transposed notation introduced by Tomasi&Kanade). Thus, matrix $\mathbf{P}$ permutes columns of $\overline{\mathbf{W}}$[1] such that $\mathbf{P}\overline{\mathbf{W}}$ lies in $range(\mathbf{S})$. Equation (1) is equivalent to

$$\overline{\mathbf{W}}^{\mathbf{T}} \mathbf{P}^T \mathbf{S}^{\perp} = \underline{\mathbf{0}} \quad (2)$$

where $\mathbf{S}^{\perp}$ represents a base in the orthogonal space to $range(\mathbf{S})$ and $\underline{\mathbf{0}}$ is a matrix of zeros. We seek the correct sorting of image points that satisfies the projection model. In the noiseless case, matrix A is computed knowing 3 correct matches [15]. Knowing A, permutation $\mathbf{P}$ can be calculated easily: The 3 initial matches are found by exhaustive search in $\mathcal{O}(N^3)$, and the other assignments through a linear program solved in $\mathcal{O}(N^3)$ (ex the Hungarian method). However, in a real case (noisy data and approximate camera model), finding the globally optimal constrained to (1,2) becomes a (very hard) combinatorial problem. State of the art algorithms either require a close initial estimate or simply do not guarantee convergence to the global optimum. It turns out, that the global optimum can be computed efficiently with some prior knowledge.

3. A unique solution for the subspace matching problem

3.1. Assumptions and constraints for unicity

In noisy conditions, equation (2) will not be satisfied in general. The intuitive way of coping with this, is to define some criterion by which that equality is approximated, for example, in the least square error sense. Then a solution is found by searching for the minimum of the error $\epsilon^2(P) = ||\mathbf{W}^{\mathbf{T}}\mathbf{P}^T\mathbf{S}^{\perp}||_2^2$ in the set of permutation matrices. This set is defined as the set of matrices with elements 0 or 1, and which rows and columns sum to 1, that is:

$$\mathbf{P} \in \mathcal{P} \Leftrightarrow \quad \sum_{i=1}^{N} \mathbf{P}_{ij} = 1 \quad (3)$$

$$\sum_{j=1}^{N} \mathbf{P}_{ij} = 1 \quad (4)$$

$$\mathbf{P}_{ij} \in \{0, 1\} \quad (5)$$

The combinatorial nature of the problem can be circumvented by relaxing the domain ($\mathcal{P}$), to its convex hull, the set of doubly stochastic matrices ($\mathcal{D}$). Besides compact, this set is convex thus suggesting a better way to designing efficient algorithms to seek the optimum. Mathematically this set is obtained by replacing the non-convex constraint 5 in the above definition by the "convex" one $\mathbf{P}_{ij} \geq 0$. In simple words we relax the {0,1} constraint. The main problem here is that the solution is not guaranteed to be unique, and possible solutions may not even be permutations. In fact there are infinite many solutions, resulting from the intersection of the polytope represented by equations (3,4) and $\mathbf{P}_{ij} \geq 0$, known as the Birkhoff polytope [18], and the linear subspace spanned by the shape matrix $\mathbf{S}$. This intuitive notion is rigorously handled in the appendix.

The surprising fact is that there exist conditions under which the above relaxation can be done, leading to a unique solution, that is to say, leading to the correct permutation. This fact converts what we thought to be a very hard problem into a clearly solvable and computationally simple one.

For clarity purposes we will be using current notation and focus on the particular case of 3D-2D matching. Considering the shape matrix $\mathbf{S} \in \mathbb{R}^{N \times 3}$ and the image points (observation matrix) $\overline{\mathbf{W}} \in \mathbb{R}^{N \times 2}$ as in equation (1,2) we state the following

Proposition 1 *If there are at least two known correspondences between 3D points and their 2D image projections, which for simplicity we define to be the first 2 coordinates, the solution of*

$$\overline{\mathbf{W}}^{\mathbf{T}} \begin{bmatrix} I_{2\times 2} & \underline{\mathbf{0}} \\ \underline{\mathbf{0}} & \mathbf{P}_0 \end{bmatrix} \mathbf{S}^{\perp} = \underline{\mathbf{0}} \quad (6)$$

with $\mathbf{P}_0 \in \mathcal{D}$*, is unique and consequently* $\mathbf{P}_0$ *is a permutation matrix.*

This proposition is formally presented and generalized in the appendix through the statement of theorem 1, together with a sketch of its proof. Intuitively speaking, the known points change the linear space spanned by the shape matrix ($\mathbf{S}$) into an affine subspace (revealed algebraically by the 2×2 identity matrix). The intersection of such an affine subspace with the Birkhoff polytope is proved to be one vertex which is, by definition, a permutation matrix.

In general, if the model has dimension $N \times r$ and observations have dimension $N \times k$ it is required the knowledge of $r - k + 1$ correspondences. We will introduce one simple "trick" by which one known correspondence can be dropped. For the particular case illustrated here, we need to know only one correspondence. Alternatively, we can simply solve N equations such as (6).

Finally remark that some degenerate cases exist, mostly inherent to the representation. In particular, if one row of $\mathbf{S}$ is expressed by a convex linear combination of 2 others there will be a doubly stochastic row which also generates the image point. However, most notably in the noisy case,

[1] Note that, in equation (1), indexes $1, \cdots, N$ in $\overline{\mathbf{W}}$ do not correspond to indexes $1, \cdots, N$ in $\mathbf{S}$

these degenerate cases do not affect the solution globally. In other words, if a small set of entries are degenerate this only affects the corresponding elements of the permutation matrix.

3.2. Approximate solutions using linear programming

The facts stated above are true in the noiseless case. As said before, equation (6) is not verified if both $\mathbf{S}$ and $\overline{\mathbf{W}}$ are noisy. So, we need some criterion to approximate it. Among all possible criteria, we seek one that exhibits robustness and computational efficiency. Note that $\mathbf{P}$ can be of large dimension. We chose the minimization of the ℓ_1 norm which can be calculated using linear programming (any other norm would do the job since the problem remains convex)[2]. So, interpreting equation (6) in the least-error-sense it boils down to the following optimization problem:

Problem 1

$$(\widehat{\mathbf{P}})^* = arg\min_{\mathbf{P}} \quad \left|\left|\overline{\mathbf{W}}^{\mathbf{T}}\mathbf{P}^T\mathbf{S}^{\perp}\right|\right|_1$$
$$s.t. \quad \begin{array}{l} \mathbf{P} \in \mathcal{D}, \\ \mathbf{P}_{i_1,j_1} = 1, \mathbf{P}_{i_2,j_2} = 1 \end{array}$$

where $\mathbf{P}_{i_1,j_1}$ and $\mathbf{P}_{i_2,j_2}$ are the two known correspondences.

Though nonlinear, this problem can be solved by linear programming recurring to the well-known *epigraph technique*. Using the same technique, the ℓ_∞ norm minimization can be performed also by linear programming.

4. Experiments

To evaluate the algorithm's performance under noisy conditions, in this section we present synthetic and real experiments. We will show the algorithm's behavior in two different real scenarios: matching between points of two 2D images and recognition of a 3D object from one 2D image. In fact, given the assumptions and the theoretical proof of the uniqueness of the solution the result stands by itself. However the effect of the affine camera approximation and image noise require an experimental evaluation since it leads to some noticeable errors.

We created 3 data sets: One with synthetic data following the whole sequence of the article (matching 2D image points to 3D object points) to test robustness to noise and two real scenarios, as we mentioned before.

4.1. Synthetic data

In the synthetic case, we defined 8 different noise levels, and for each of them, 200 independent experiments were run. The 3D object is composed of 22 randomly generated points. In each experiment, we used two different sets: a 3D object ($\mathbf{S}$) and one 2D image ($\mathbf{W}$), generated by an orthographic projection and added gaussian noise ($\mathcal{N}(0, \sigma^2)$).

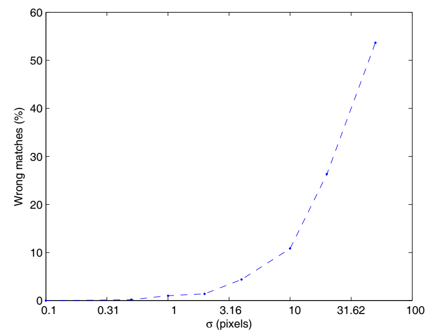

Figure 3. Synthetic experiments

In figure 3, we show the noise standard deviation (σ - X axis) in pixels whereas image size is 1000×1000. See that, if $\sigma = 3$ pixels ($3\sigma = 9$) the percentage of wrong matches is below 5%. For higher noise levels (ex. $\sigma = 10$), we obtained 90% of correct matches. Note that with such level of noise, 99% of the points may have up to 100 pixels deviation in an image of size 1000×1000.

4.2. Image to shape matching

To evaluate our method with real data, we used the Hotel sequence[3]. The 3D object has 106 points and the sequence is composed by 182 2D images (figure 4 - Up Left).

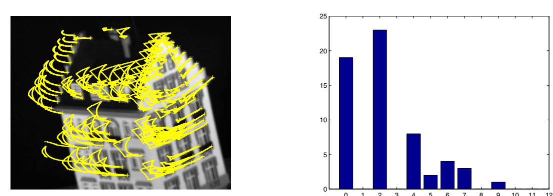

Figure 4. Real data - Left: Hotel's sequence with image tracks. Right: Histogram of wrong matches

Using this data, 60 experiments were performed randomly selecting 5 (out of 182) 2D images in each experiment. The object's shape ($\mathbf{S}^{\perp}$) is computed from 4 images using Tomasi-Kanade shape-from-motion algorithm, and the other one is used as a test image ($\overline{\mathbf{W}}$). By observing figure 4 (Right), we can see that the algorithm obtained the correct solution in almost 20% and switched 2 points in 23% of cases. Note that the maximum number of wrong matches in all 60 experiments was 9 (less than 10% of 106 features).

[2] Matlab code using Yalmip software is available at http://users.isr.ist.utl.pt/~manuel

[3] http://vasc.ri.cmu.edu/idb/html/motion/long-hotel/index.html

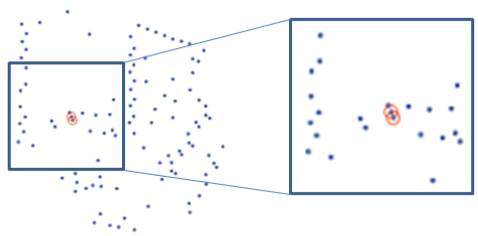

Figure 5. A common wrong match

Although the percentage of "completely" correct solutions is not high, a wrong solution does not imply a gross mistake. As shown in figure 5, common mistakes occur because neighboring points are switched. Also, sometimes the images used to compute the shape are almost degenerate (very close frames in time).

4.3. Face recognition: a 2D-2D example

Figure 6. Face images with strong pose

As referred before, the main goal of our method is recognizing a 3D object from one 2D image. But, we can apply this approach to find the correspondence between two n-dimensional sets, such as [8]. In this section, we present the matching results between 2D points from two images, an useful task to several face recognition methods.

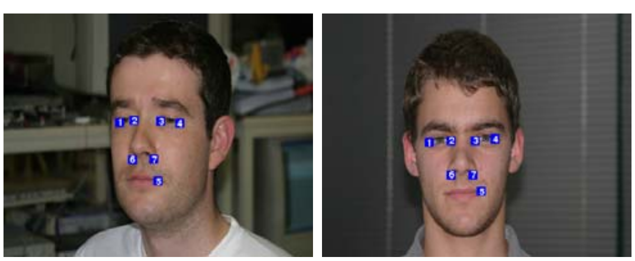

Figure 7. Frontal face images

To evaluate our method, we use 145 2D images of 20 different people and each face is represented by 7 features points (see figure 6)[4]. The images were captured by a regular digital camera (3072×2048 pixels) and there were no constraints on the distance between the camera and the subject (scale factor).

In first experiment, face images with strong pose are used (figure 6). The results presented in table were obtained from 100 experiments. Each one consists of randomly selecting one image and computing the correspondence between the 2D points of this image and the points of each other images. This means that, in each experiment, 144 permutations matrices were calculated. According to the obtained results, the highest percentage of correct matches occurs between eye features.

Feature n.	1	2	3	4	5	6	7
1	98	1	-	-	-	-	1
2	1	96	1	-	-	-	2
3	-	2	94	-	1	2	1
4	-	-	2	96	1	1	-
5	-	-	-	2	98	-	-
6	1	1	2	-	-	94	2
7	-	-	1	2	-	3	94

Table 1. Confusion matrix for first face matching experiment. Entries $\{i,j\}$ represent the percentage of correspondences between feature i and j.

By quick inspection of the tables we see that frontal face images produce less error. This is due to the wider spread of feature points (fig. 7). In figure 6 it is noticeable that features 1 and 2 have a very close projection. Thus, it is possible that the minimum error solution switches these two points. This is the same source of error described in the previous experiment (fig. 5).

Feature n.	1	2	3	4	5	6	7
1	99	1	-	-	-	-	-
2	1	99	-	-	-	-	-
3	-	-	98	1	-	1	-
4	-	-	1	99	-	-	-
5	-	-	-	-	100	-	-
6	-	-	1	-	-	99	-
7	-	-	-	-	-	-	100

Table 2. Confusion matrix for frontal image matching experiment.

5. Conclusions and future work

This article tackles the problem of finding correspondences between feature vectors when prior knowledge exists about the geometry of the data. In particular, if data is constrained to a linear subspace, feature matching is the process by which points in the feature vector are sorted, such that it lies on the known subspace. Under noisy conditions, this is a very hard combinatorial problem to solve.

The complexity of the proposed algorithm does not depend on the rank of the subspaces, but rather on the differ-

[4]The whole faces' database is available at http://users.isr.ist.utl.pt/~manuel/faceRecog/facesDatabase.zip

ence of rank between them. Note that the number of variables (entries of the permutation matrix $\mathbf{P}$) increases with N^2.

First we prove that the original integer problem has unique solution in a convex set. Second, in the noisy case, we compute a robust estimate with a linear programming-based algorithm. Finally, we show experiments demonstrating its adequacy to real world situations.

Major extensions include the ability to deal with outliers keeping low complexity. This implies using non-squared permutation matrices introducing extra degrees of freedom (which feature to eliminate).

Acknowledgements: Authors wish to thank to João Xavier for fruitful ideas and discussions.

6. Appendix

This appendix proves the major contribution of this article, introduced and explained intuitively in section 3.

Let N, r and k be positive integers, such that $r \geq k$, and $N \geq r + k$. Let

$$C = \begin{bmatrix} a^1 & a^2 & \cdots & a^k \\ X^1 & X^2 & \cdots & X^k \end{bmatrix} \in \mathbb{R}^{N \times k},$$

be a matrix where $a^i \in \mathbb{R}^{(r-k+1)\times 1}$ and $X^i \in \mathbb{R}^{(N-r+k-1)\times 1}$, $i = 1, \ldots, k$ are given vectors. Also let S be a prescribed r-dimensional vector subspace of $\mathbb{R}^N$, such that the columns of C belong to S.

Then we have the following:

Theorem 1 *For generic matrix C and subspace S we have the following: if $M \in \mathbb{R}^{(N-r+k-1)\times(N-r+k-1)}$ is a doubly-stochastic matrix such that all vectors $\begin{bmatrix} a^i \\ MX^i \end{bmatrix}$, $i = 1, \ldots, k$, belong to S, then M is the identity matrix.*

In the case we are interested in, we have $k = 2$ and $r = 2$, or $k = 2$ and $r = 3$, while N is large number. Here we just sketch the main ideas of the proof for $k = 2$ - the general case is done analogously.

Sketch of the proof for $k = 2$:

Our main goal is to find a vector v which is a linear combination of the vectors X^1 and X^2 such that

$$Mv = v. \tag{7}$$

Having this relation we use the doubly-stochastic property of M, to obtain strong restrictions on v. This relies heavily on the Perron-Frobenius theorem for nonnegative matrices.

First of all, there exists a permutation matrix P such that PMP^T has the block matrix form:

$$PMP^T = \begin{bmatrix} M_1 & 0 & \cdots & 0 \\ M_{21} & M_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ M_{l1} & M_{l2} & \cdots & M_l \end{bmatrix},$$

where matrices $M_1, \ldots, M_l$, are irreducible, i.e. cannot be further splitted in this way. Moreover, since M is a nonnegative matrix, with row and column sums equal to 1, we have that all off-diagonal blocks are equal to zero. So, from now on, we may assume that is in the block-diagonal form:

$$PMP^T = \begin{bmatrix} M_1 & 0 & \cdots & 0 \\ 0 & M_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & M_l \end{bmatrix} \tag{8}$$

with all blocks being irreducible. Moreover, we have the following lemma:

Lemma 1 *If Q is an irreducible doubly-stochastic matrix, and w is a vector such that $Qw = w$, then all entries of w are equal, i.e. there exists $c \in \mathbb{R}$, such that $w = c[1, \ldots, 1]^T$.*

Proof of lemma:

Every row-stochastic matrix has as an eigenvector the vector $[1, \ldots, 1]^T$, with the eigenvalue 1. On the other hand, by Geršgorin theorem (see below), all real eigenvalues of row-stochastic matrix are less or equal than 1. Since Q is an irreducible nonnegative matrix, by Perron-Frobenius theorem (see below) the multiplicity of its dominant eigenvalue (which as we have proved is 1), is equal to 1, i.e. it has only one linearly independent eigenvector corresponding to the eigenvalue 1. As we saw, this one is $[1, \ldots, 1]^T$, as wanted.

Geršgorin theorem: Let $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ be a square matrix. For every $i = 1, \ldots, n$, denote by d_i the sum of absolute values of all non-diagonal entries of the i-th row, i.e.

$$d_i = \sum_{j \neq i} |a_{ij}|.$$

Then all (complex) eigenvalues of A lie in the union of discs with centres in a_{ii} with radius d_i, for all $i = 1, \ldots, n$.

(Part of) Perron-Frobenius theorem: Let $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ be a nonnegative square matrix, i.e. such that $a_{ij} \geq 0$, for every i and j. Then A has a real nonnegative eigenvalue λ corresponding to the eigenvector with all entries nonnegative, and such that all other eigenvalues μ of the matrix A, are such that $|\mu| \leq \lambda$.

If in addition A is irreducible, then λ has multiplicity 1.

Now, to find a vector v that satisfies (7) we do the following: Denote by Σ_i, $i = 1, 2$, the space of all vectors of the form $\begin{bmatrix} a^i \\ Y^i \end{bmatrix}$, where Y^i runs through $\mathbb{R}^{(N-r+1)\times 1}$, which belong to S. Since the space of all vectors $\begin{bmatrix} a^i \\ Y^i \end{bmatrix}$ has the dimension $N - r + 1$ and S has the dimension r, then their intersection, Σ_i, (in generic case) has dimension 1. Moreover, Σ_1 and Σ_2 are parallel lines, and hence they determine a plane Σ.

Denote by $p \subset \Sigma$ the line determined by $\begin{bmatrix} a^1 \\ X^1 \end{bmatrix}$ and $\begin{bmatrix} a^2 \\ X^2 \end{bmatrix}$, and by $q \subset \Sigma$ the line determined by $\begin{bmatrix} a^1 \\ MX^1 \end{bmatrix}$ and $\begin{bmatrix} a^2 \\ MX^2 \end{bmatrix}$. If they are parallel, then we have that $MX^2 - MX^1 = X^2 - X^1$, i.e. we can take $v = X^2 - X^1$.

Otherwise they intersect at some point T, and since Σ_1 and Σ_2 are parallel, we have that for some $\lambda \in \mathbb{R}$ we have the following:

$$\begin{aligned} T &= (1-\lambda)\begin{bmatrix} a^1 \\ X^1 \end{bmatrix} + \lambda \begin{bmatrix} a^2 \\ X^2 \end{bmatrix}, \\ T &= (1-\lambda)\begin{bmatrix} a^1 \\ MX^1 \end{bmatrix} + \lambda \begin{bmatrix} a^2 \\ MX^2 \end{bmatrix}, \end{aligned}$$

and so in this case we have $Mv = v$, for $v = (1-\lambda)X^1 + \lambda X^2$.

Finally, with the obtained tools, we can prove that $M = I_{N-r+1}$. First of all, put M in the block-diagonal form (8), with all blocks M_i being irreducible of size d_i. If all d_i are equal to 1, we are done. Otherwise, split the vectors PX^1, PX^2 and Pv in the blocks of the corresponding dimensions d_i: $PX^i = [x_1^i, \ldots, x_l^i]^T$, $Pv = [v_1, \ldots, v_l]^T$. Then we have that for all $i = 1, \ldots, l$:

$$M_i v_i = v_i,$$

and so by Lemma 1 there exists $c_i \in \mathbb{R}$ such that $v_i = c_i[1, \ldots, 1]^T \in \mathbb{R}^{1\times d_i}$. Thus we would have that the vectors x_i^1, x_i^2 and $[1, \ldots, 1]^T$ are linearly dependent, which generically is not satisfied.

Thus $M = I$, as wanted.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):509–522, Apr 2002.

[2] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. *IEEE Conf. on Computer Vision and Pattern Recognition*, 1:26–33 vol. 1, June 2005.

[3] T. M. Breuel. Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding*, 90(3):258–294, 2003.

[4] D. Clemens and D. Jacobs. Model group indexing for recognition. *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 4–9, June 1991.

[5] P. David, D. Dementhon, R. Duraiswami, and H. Samet. Softposit: Simultaneous pose and correspondence determination. *Intern. Journal of Computer Vision*, 59(3):259–284, 2004.

[6] A. W. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(13-14):1145 – 1153, 2003.

[7] S. Gold, A. Rangarajan, L. Chien-Ping, S. Pappu, and E. Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern Recognition*, 31(8):1019–1031, 1998.

[8] H. Jiang, M. S. Drew, and Z.-N. Li. Matching by linear programming and successive convexification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(6):959–975, 2007.

[9] H. Li and R. Hartley. The 3d-3d registration problem revisited. *IEEE Intern. Conf. on Computer Vision*, Oct. 2007.

[10] Y. Moses and S. Ullman. Limitations of non model-based recognition schemes. In *European Conf. on Computer Vision*, pages 820–828. Springer Verlag, 1992.

[11] R. Oliveira, J. Costeira, and J. Xavier. Optimal point correspondence through the use of rank constraints. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2:1016–1021 vol. 2, June 2005.

[12] C. Olsson, O. Enqvist, and F. Kahl. A polynomial-time bound for matching and registration with outliers. *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[13] B. Pires, J. F. Moura, and J. Xavier. Lasic: A model invariant framework for correspondence. *IEEE Intern. Conf. on Image Processing*, pages 2356–2359, Oct. 2008.

[14] J. J. Rodrigues, P. Aguiar, and J. Xavier. Ansig an analytic signature for permutation-invariant two-dimensional shape representation. *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2008.

[15] G. Sudhir, A. Zisserman, and S. Banerjee. Finding point correspondences in motion sequences preserving affine structure. *Computer Vision and Image Understanding*, 68(2):237–246, 1997.

[16] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *European Conf. on Computer Vision*, pages II: 596–609, 2008.

[17] Z. Wang and H. Xiao. Dimension-free affine shape matching through subspace invariance. *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2009.

[18] L. Wolsey and G. Nehmauser. *Integer and Combinatorial Optimization*. John Wiley & Sons, July 1999.