

LAMP: LINEAR APPROACH FOR MATCHING POINTS

Manuel Marques and João Costeira

Institute for Systems and Robotics
Instituto Superior Técnico
Av. Rovisco Pais - 1049-001 Lisboa - PORTUGAL
{manuel,jpc}@isr.ist.utl.pt

ABSTRACT

The 3D object recognition from a single or multiple 2D images is a very important problem in the computer vision field with a wide range of real applications. Considering the affine camera model, the main issue in solving this problem is the matching process between the object's 3D points and their 2D projections. In this work, we tackle the 3D-2D matching problem. It is formulated as a finite set of independent linear programs, solved efficiently. The 2D-2D and 3D-3D are also discussed. To show the validity of the proposed method, synthetic and real experiments are performed.

Index Terms Correspondence problem, matching, object recognition

1. INTRODUCTION

Nowadays, recognizing 3D rigid objects from a single or multiple 2D views is a very important topic in the image processing field and a fundamental task for a large number of applications. Assuming an affine camera model, we show how to find, in a very efficient way, a global and exact solution for the matching problem between two point clouds: the object's (3D) and the image's (2D). The solution can be obtained by solving a finite set of linear programs and imposing a known fixed number of matches.

Unlike the recent invariant permutation methods [1, 2] developed for 2D-2D recognition, the correspondence problem becomes the central issue in 3D-2D object recognition tasks.

In general, image based recognition/matching methods can be classified in two different classes: appearance and feature based. The first approach hinges on image correlation [3] and due to this, small motion variations between images should be assumed. To overcome this limitation, a global and more general assumption is used by several algorithms - rigidity. To deal with this intrinsic feature of the model, an optimization problem is formulated and different methods are applied to solve it: graph search [4], randomized search [5], dynamic programming. In [6], the authors propose a global n -frame correspondence solution solving an optimization problem with a non-linear cost function. Such as in [7], convergence is not guaranteed and the obtained solution depends

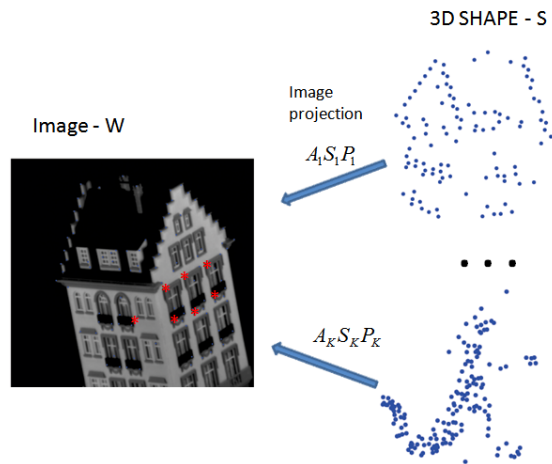


Fig. 1. 3D Object Recognition from 2D views: Image points (\mathbf{W}) result from a projection of a permuted shape matrix. Correct object has minimum projected error $|\mathbf{W} - A_i S_i P_i|$

on the initial point.

In this paper, we define the general principles under which a unique optimal solution is obtainable for the 3D-2D correspondence problems. Unlike [7], these principles allow us to build a convex optimization problem (without local *minima*). The solution has a lower computational cost than [5], because the constraint has linear formulation and the simplex or interior-point methods can be used. At this point we do not deal with outliers. Even though the computational aspect is relevant, the main issue in this paper is the theoretical statement about the solution's uniqueness and optimality character. So far we have only been able to produce such a statement regarding permutation matrices. To consider outliers we have to impose rank constraints on the matrices turning the task much harder (though some algorithmic tweaks could be set).

1.1. Problem Formulation

As figure 1 shows, we know the object's 3D shape and one image with unsorted 2D projections. To recognize the object in this 2D view, we must compute correspondences between the two sets of features (3D object - 2D projections). Even though this figure suggests the existence of a search process (in the shape domain), we concentrate only on the

shape-image (shape-shape or image-image) correspondence.

Given the 3D shape composed by N points, $\mathbf{S} \in \mathbb{R}^{3 \times N}$, and a set of N 2D projections $\mathbf{W} \in \mathbb{R}^{2 \times N}$, resulting from a linear transformation (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

$$\begin{aligned} \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 & x_2 & \cdots & x_N \\ y_1 & y_2 & \cdots & y_N \\ z_1 & z_2 & \cdots & z_N \end{bmatrix}}_S &= \\ = \underbrace{\begin{bmatrix} u_1 & u_2 & \cdots & u_N \\ v_1 & v_2 & \cdots & v_N \end{bmatrix}}_W \mathbf{P} & \quad (1) \end{aligned}$$

where A is an unknown transformation. In other words, matrix \mathbf{P} permutes columns of \mathbf{W} such that \mathbf{WP} lies in the $range(\mathbf{S})$. Equation (1) is equivalent to

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

where \mathbf{S}^\perp represents a base in the orthogonal space to $range(\mathbf{S})$ and $\mathbf{0}$ is a matrix of zeros. In summary, we seek the correct sorting of image points that satisfies the projection model. As we will see in the following section, this problem has straightforward solutions in the noiseless case but leads to combinatorial (very hard) problems if noise is present. In other words, with noisy data, state of the art algorithms either require a close initial estimate or simply do not converge to the global optimum. It turns out, that the global optimum can be computed efficiently. As we will see, the global solution of problem (eq 2) is solved by a linear program (or a finite set of linear programs).

2. FINDING CORRESPONDENCES

In the noiseless case, matrix A (1) is computed knowing 3 correct matches [5]. Through this, matrix \mathbf{P} can be also calculated. The 3 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 solution of problem (1,2) becomes a combinatorial problem. The method presented in [5] finds a local solution depending on 4 chosen points. In [6, 7], a global solution is found, but the convergence of both algorithms is not guaranteed, such as we mentioned before.

To deal with these limitations, we propose a new approach to find a global solution to (1) by a set of N convex programs (without local *minimae*).

2.1. The proposed method

With noisy data equation (2) must be solved in least error sense. One possible solution results from solving the following optimization problem:

Problem 1

$$\begin{aligned} (\hat{\mathbf{P}})^* &= \arg \min_{\mathbf{P}} \|\mathbf{S}^\perp \mathbf{P} \mathbf{W}^\mathbf{T}\|_1 \\ \text{s.t. } & \mathbf{P} \in \mathcal{P} \end{aligned}$$

where \mathcal{P} represents the set of permutation matrices. Problem 1 is an integer optimization program and there is no efficient way to optimally solve it. One way of avoiding brute force is to use a non-optimal method [5] or changing the original problem to a different domain such as the convex-hull of the set of permutation matrices. This is the set of the *doubly-stochastic* matrices \mathcal{DS} [6, 7], defined by the following constraints:

$$\sum_{i=1}^N \mathbf{P}_{ij} = 1, \sum_{j=1}^N \mathbf{P}_{ij} = 1 \quad (3)$$

$$\mathbf{P}_{ij} \geq 0, \forall i = 1, \dots, N, \forall j = 1, \dots, N \quad (4)$$

After some elementary matrices manipulations, the cost function of Problem 1 is rewritten and the relaxed version of this problem is given by

Problem 2

$$\begin{aligned} (\hat{\mathbf{p}})^* &= \arg \min_{\mathbf{P}} \|\mathbf{G} \mathbf{p}\|_1 \\ \text{s.t. } & \mathbf{P} \in \mathcal{DS} \end{aligned}$$

where $\mathbf{G} = \mathbf{S}^\perp \otimes \mathbf{W}$ and \mathbf{p} is $\text{vec}(\mathbf{P})^1$. The symbol \otimes is the Kronecker product. The relaxed version (Problem 3), though nonlinear, can be solved by linear programming recurring to the *epigraph technique* [8], leading to

Problem 3

$$\begin{aligned} (\mathbf{p}, t)^* &= \arg \min_t t \\ & -t \leq \mathbf{G} \mathbf{p} \leq t \\ \text{s.t. } & \mathbf{P} \in \mathcal{DS} \\ & t \geq 0 \end{aligned}$$

The new variable t is called *slack variable*. This last problem is a linear one (linear cost function and linear constrains) and can be solved by linear programming. To compute the solution with standard solvers, we convert the Problem (3) in the canonical form, given by

Problem 4

$$\begin{aligned} (\hat{\mathbf{p}})^* &= \arg \min_x c^\mathbf{T} x \\ \text{s.t. } & \underbrace{\begin{bmatrix} \mathbf{G} & \mathbf{I}_1 & 0 \\ \mathbf{G} & -\mathbf{I}_1 & 0 \\ \mathbf{B} & 0 & \mathbf{I}_2 \\ \mathbf{B} & 0 & -\mathbf{I}_2 \end{bmatrix}}_D \underbrace{\begin{bmatrix} \mathbf{p} \\ t_1 \\ t_2 \end{bmatrix}}_x \leq \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \\ & x \geq 0 \end{aligned}$$

where \mathbf{B} represents equations (3) in matrix form and \mathbf{I}_1 and \mathbf{I}_2 identity matrices, $t = [t_1 \ t_2]$ and cost vector c is given by

$$c = [0 \ c_1 \ c_2]$$

¹ $\text{vec}()$ stacks the columns of its argument into a single column.

where c_1 and c_2 coefficients correspond to the variables t_1 and t_2 , respectively. Note that c_1 and c_2 should be greater than 0.

Even though Problem 4 is in linear form, an integer solution is not guaranteed. In other words, we avoid the combinatorial nature by relaxing to the continuous domain (the convex hull of the set of permutation matrices) but, since the constraints' matrix \mathbf{D} is not unimodular [9] (due to terms $\mathbf{S}^\perp \otimes \mathbf{W}$), the vertices of the polytope are not necessarily integer.

We discovered conditions under which the solution of that same problem is guaranteed to be a permutation matrix and unique. Summarizing the main idea behind the scheme, we have the following general principles:

- In case we know the 3D shape of the object and one 2D image, we will need prior knowledge of one matched 3D-2D pair (or N linear programs) leading to the sought solution.
- In case we have 3D-3D point matching (say two views from a moving laser scanner), or 2D-2D point matching (two images of planar objects) the solution of problem 4 (or a slight modification of it) is guaranteed to be integer and unique.

The solution for Problem 4 is unique and therefore integer, if we know 2 matched pairs (in 3D-2D case) or 1 matched pair (in 3D-3D or 2D-2D cases). In both cases, one of the needed matched pair can be obtained without computational cost using a simple trick: adding new (synthetic) features \mathbf{S}_s to the original shape \mathbf{S} and computing its (synthetic) image projections. In Problem 4, \mathbf{S} and \mathbf{W} are replaced by the following expressions

$$\begin{aligned} \mathbf{S}' &= \begin{bmatrix} \mathbf{S}_s \mathbf{C}_s & \mathbf{S}_s \left(\frac{1}{N_s} \mathbf{1}_{N_s} \mathbf{1}_N^T \right) \\ \mathbf{S} \left(\frac{1}{N} \mathbf{1}_N \mathbf{1}_{N_s}^T \right) & \mathbf{S} \mathbf{C} \end{bmatrix} \quad (5) \\ \mathbf{W}' &= [\mathbf{A}_s \mathbf{A}] \mathbf{S}' \quad (6) \end{aligned}$$

where matrices $\mathbf{C} = \mathbf{I} - \frac{1}{N} \mathbf{1}_N \mathbf{1}_N^T$ and \mathbf{C}_s are *centering* matrices, \mathbf{A} the unknown affine camera model and $\mathbf{1}_m$ is a m column vector of ones. Matrix \mathbf{A}_s , defined by the user, is the synthetic affine transformation, which allows us to obtain the projections of the synthetic shape \mathbf{S}_s .

The success of the approach rests on the fact that the solution of Problem 4 is integer. This is so because the equation system given by

$$\mathbf{S}'^\perp \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_0 \end{bmatrix} \mathbf{W}'^T = \mathbf{0}, \mathbf{P}_0 \in \mathcal{DS} \quad (7)$$

has a unique solution. Of course, we know that if only one solution exists it must be the sought permutation matrix. Note that the Identity matrix (\mathbf{I} in eq. 7) represents the known correspondences.

This statement hinges on the proof that the intersection between Birkhoff polytope and the given affine subspace

(constraints of Problem 4) is one single point (for details, see [10]).

2.2. LAMP - The Algorithm

In this section, we propose a simple way to implementing our solution, consequently easily benchmarked. Due to the intuitive syntax, we use Yalmip interface [11]. To solve the formulated linear program, the user can choose one lp solver for Matlab (*ex lp_solve*). We illustrate here the more complex situation of matching a set of 2D points (image) to a known 3D model (object). In this case it is required the knowledge of one matched point (the first column of \mathbf{W} and the first column of \mathbf{S} should match). Algorithm 1 shows how to implement the solution to the 3D-2D correspondence problem².

Algorithm 1 Implementation of LAMP

Require: Shape matrix \mathbf{S} and Image data \mathbf{W} .

Ensure: The value of \mathbf{P}

- 1: $\mathbf{P} = \text{sdpvar}(N + \text{size}(\mathbf{S}_s, 2), N + \text{size}(\mathbf{S}_s, 2), 'full')$;
 - 2: $\text{constraints} = [\text{sum}(\mathbf{P}, 1) \leq 1,$
 $\text{sum}(\mathbf{P}, 2) \leq 1,$
 $\mathbf{P}(1:(\text{size}(\mathbf{S}_s, 2)+1), 1:(\text{size}(\mathbf{S}_s, 2)+1)) ==$
 $\text{eye}(\text{size}(\mathbf{S}_s, 2)+1),$
 $\mathbf{P}(:) \geq 0]$;
 - 3: $\text{costFunction} = \text{norm}(\text{null}(\mathbf{S}') * \mathbf{P} * \mathbf{W}', 1)$;
 - 4: $\text{solvesdp}(\text{constraints}, \text{costFunction})$
 - 5: $\mathbf{P} = \text{double}(\mathbf{P})$
 - 6: $\hat{\mathbf{P}} = \text{Hungarian}(-\mathbf{P})$
-

In the last step of Algorithm 1, we obtain an integer solution $\hat{\mathbf{P}}$ computing the closest permutation matrix to \mathbf{P} through the Hungarian method (obtainable at Matlabcentral). For low noise, truncating \mathbf{P} with an high threshold is enough to obtain a permutation matrix.

3. EXPERIMENTS

To evaluate the algorithm's performance under noisy conditions, we present here synthetic and real experiments. In the synthetic case, we benchmark LAMP against the algorithm proposed in [5] (SZB), performing 200 independent experiments per each 5 different noise levels. The 3D object is composed of 25 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. The gaussian noise ($\mathcal{N}(0, \sigma^2)$) was added to all points' projections and an higher perturbation ($\mathcal{N}(0, 20\sigma^2)$) to one or two projections. Due to the difference of noise's power, the last points can be called by "very noisy" points. It is important to refer that the affine transformation is computed using 4 correspondences [5] and one of them is the outlier point.

In figure 2, we show the noise standard deviation (σ - X axis) in pixels whereas image size is 1000×1000 . If the noise level is below than 15 pixels, LAMP achieves a better

²To better benchmark our method, the Matlab script is available at <http://users.isr.ist.utl.pt/~manuel>

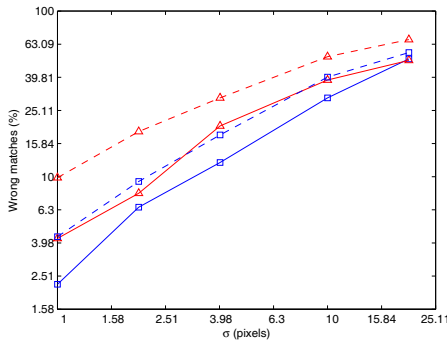


Fig. 2. Synthetic experiments: LAMP(blue squares) Vs. SZB(red triangles). Performance in Solid Line contains one very noisy point in the data set. Dashed line- two very noisy points. Because LAMP acts globally the influence of these points has less impact. In SZB one bad choice can jeopardize the search.

performance than the algorithm suggested in [5]. With an increasing of the projections with a high perturbation, our method keeps a good performance unlike the other algorithm whose results can be catastrophic.

To evaluate our method with real data, we used the Hotel sequence³. In the considered dataset, the 3D object has 106 points and the sequence is composed by 182 2D images.

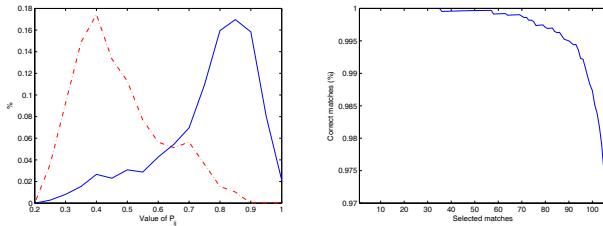


Fig. 3. Left: Histograms of the $(\mathbf{P} \odot \hat{\mathbf{P}})$ values: solid line - correct matches dashed line - wrong matches Right: best k ($k = 1, \dots, 106$) matches Vs. % of correct matches

Using this data, 60 experiments were performed selecting 5 (out of 182) 2D images randomly selected in each experiment. The object's shape (\mathbf{S}^\perp in 2) is computed from 4 images using Tomasi Kanade shape-from-motion algorithm, and the other one is used as a 2D test image (\mathbf{W}).

When the solution is not the desired one (step 6 of Alg. 1), a lot of matches are correct. To avoid wrong matches, the proposed optimization problem (Problem 4) allows us to obtain a suboptimal solution, considering only the higher values of \mathbf{P} (a square partial-permutation solution), instead of calculating the permutation matrix $\hat{\mathbf{P}}$ (see algorithm 1). This situation happens because the values $(\mathbf{P} \odot \hat{\mathbf{P}})_{ij}$ of correct and wrong matches are not uniformly distributed (\odot represents the Schur product), as figure 3 (Left) shows. According

to experiments, the 30 highest values of \mathbf{P} correspond to correct matches (see figure 3 - Right) and due to that reason these matches can be used to estimate the affine transformation A and consequently, matrix \mathbf{P} .

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an efficient way to compute a solution for the 3D-2D correspondence problem, a crucial step in 3D object recognition from 2D views. Imposing a general constraint, the solution is found by solving one (or a finite set) of linear programs. Because of the formulated optimization problem is convex, the proposed method allows us to compute the desired solution and does not need an initialization. On the other hand, handling outliers is still an open issue.

5. REFERENCES

- [1] R. Kondor, A. Howard, and T. Jebara, "Multi-object tracking with representations of the symmetric group," in *Artificial Intelligence and Statistics*, 2007.
- [2] J. J. Rodrigues, P. Aguiar, and J. Xavier, "Ansig an analytic signature for permutation-invariant two-dimensional shape representation," *CVPR*, 2008.
- [3] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, 1981, pp. 674–679.
- [4] V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via graph cuts," in *ECCV*, 2002, pp. 82–96.
- [5] G. Sudhir, A. Zisserman, and S. Banerjee, "Finding point correspondences in motion sequences preserving affine structure," *CVIU*, vol. 68, no. 2, pp. 237–246, 1997.
- [6] J. Maciel and J.P. Costeira, "A global solution to sparse correspondence problems," *TPAMI*, 2003.
- [7] B. Pires, J. F. Moura, and J. Xavier, "Lasic: A model invariant framework for correspondence," *ICIP*, 2008.
- [8] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 2nd edition, 1999.
- [9] L. Wolsey and G. Nehmouser, *Integer and Combinatorial Optimization*, John Wiley & Sons, July 1999.
- [10] M. Marques, M. Stošić, and J. Costeira, "Subspace matching: Unique solution to point matching with geometric constraints," in *ICCV*, 2009.
- [11] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *CCA/ISIC/CACSD*, sep. 2004.

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