A Global Solution to Sparse Correspondence Problems

João Maciel and João P. Costeira

Abstract—We propose a new methodology for reliably solving the correspondence problem between sparse sets of points of two or more images. This is a key step in most problems of computer vision and, so far, no general method exists to solve it. Our methodology is able to handle most of the commonly used assumptions in a unique formulation, independent of the domain of application and type of features. It performs correspondence and outlier rejection in a single step and achieves global optimality with feasible computation. Feature selection and correspondence are first formulated as an integer optimization problem. This is a blunt formulation, which considers the whole combinatorial space of possible point selections and correspondences. To find its global optimal solution, we build a concave objective function and relax the search domain into its convex-hull. The special structure of this extended problem assures its equivalence to the original one, but it can be optimally solved by efficient algorithms that avoid combinatorial search. This methodology can use any criterion provided it can be translated into cost functions with continuous second derivatives.

Index Terms—Correspondence problem, linear and concave programming, sparse stereo.

1 Introduction

Estimating feature correspondences between two or more images is a long standing fundamental problem in computer vision. Most methods for 3D reconstruction, object recognition, and camera self-calibration start by assuming that image feature-points were extracted and put to correspondence. This is a key problem and, so far, no general reliable method exists to solve it. There are three main difficulties associated with this problem. First, there are no general constraints to reduce its ambiguity. Second, it suffers from high complexity due to the huge dimensionality of the combinatorial search space. Finally, the existence of outliers must be considered since features can be missing or added through a sequence of images, due to occlusions and errors of the feature extraction procedure.

1.1 Overview of Correspondence Methods

Correspondence can be interpreted as an optimization problem. Each method translates the assumptions into an objective function—criterion—and a set of constraints.

Constraints are conditions that must be strictly met. Examples are order [19], [22], epipolar constraint [19], [22]—rigidity as a constraint—uniqueness [7], visibility [25], and proximity. Tracking-like algorithms [13] impose strict proximity constraints so they should be considered as continuous-time methods. The objective function reflects a condition that can be relaxed, but which value should be optimized. The most commonly used objective function is image correlation [28], [13], [16]—image similarity assumption. Other usual choices are point proximity [13], [30] or smoothness of disparity fields [19]. Finally, correspondence algorithms differ also in the computational framework used to solve optimization problems. Dynamic programming [19], graph search [22], bipartite graph matching [6], and convex minimization [13] guarantee optimality. Nonoptimal approaches include greedy algorithms [29], simulated annealing [26], relaxation [7], alternating optimization and constraint projection [1], and randomized search [28].

Vision systems often have to deal with the existence of spurious features and occlusions. Algorithms that explicitly handle these situations are more likely to behave robustly. The work in [28] presents a pruning mechanism that performs outlier rejection in sets of previously matched features.

2 Correspondence as an Optimization Problem

We formulate the correspondence problem as an integer optimization problem in a generic sense. In other words, it can handle most of the commonly used assumptions using one single formalism. Both problems of feature selection and correspondence were designed as one single optimization problem so both tasks are performed in an integrated way. Furthermore, its global solution can be found avoiding combinatorial search without having to impose additional assumptions. We do so by relaxing the discrete search domain into its convex-hull. The special structure of the constraints and objective function assure that the relaxation is exact so the result is an equivalent problem that can be optimally solved by efficient algorithms. For the sake of simplicity, we start with the two-image case; however, the extension to sequences is discussed in Section 2.10.

2.1 Problem Formulation

Consider the images of a static scene shown in Fig. 1.[1] Segment p_1 represents feature-points on the first image and p_2 on the second—the white dots. Some of these are projections of the same 3D points. Arrange their representations in two matrices $\mathbf{X}$ and $\mathbf{Y}$ as

• The authors are with the Instituto de Sistemas e Robótica–Instituto Superior Técnico, Av. Rovisco Pais, 1049-001 Lisboa, Portugal. E-mail: {maciel, jpc}@isr.ist.utl.pt.

Manuscript received 29 Nov. 2000; revised 30 Nov. 2001; accepted 10 May 2002.
Recommended for acceptance by D. Fleet.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 113212.

1. Data was provided by the Modeling by Video group in the Robotics Institute at CMU.

Fig. 1. Two images from the *Hotel* sequence, with extracted corners.

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,N} \\ \vdots & & \vdots \\ x_{p_1,1} & \cdots & x_{p_1,N} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} y_{1,1} & \cdots & y_{1,N} \\ \vdots & & \vdots \\ y_{p_2,1} & \cdots & y_{p_2,N} \end{bmatrix}. \quad (1)$$

The N-dimensional features can represent image coordinates of feature-points or any image-related quantity like intensities of neighboring pixels. The type of information conveyed by the features does not affect our formulation. The goal is to find a correspondence between rows of $\mathbf{X}$ and $\mathbf{Y}$.

Using the previous definitions, we formulate the correspondence problem as the integer constrained minimization Problem 1:

Problem 1.

$$\mathbf{P}^* = \arg \min_{\mathbf{P}} \quad J(\mathbf{X}, \mathbf{Y}, \mathbf{P}) \\ s.t. \quad \mathbf{P} \in \mathcal{P}_{\rm p}(p_1, p_2).$$

J can be almost any scalar objective function—Section 2.2. $\mathbf{P}$ is constrained to $\mathcal{P}_{\rm p}(p_1, p_2)$, the set of $p_1 \times p_2$ *partial permutation matrices* (${\rm p_p}$-matrices). A ${\rm p_p}$-matrix is a permutation matrix with added columns and rows of zeros. The optimal $\mathbf{P}^*$ is a zero-one variable that selects and sorts some rows of $\mathbf{Y}$, putting them in correspondence with the rows of $\mathbf{X}$. For each entry, $\mathbf{P}_{i,j}$ when set to 1 indicates that features $\mathbf{X}_{i\cdot}$ (row i of $\mathbf{X}$) and $\mathbf{Y}_{j\cdot}$ (row j of $\mathbf{Y}$) are put in correspondence. Fig. 2 shows an example. To guarantee robustness in the presence of outliers, $\mathbf{P}$ must also represent unmatched features so it cannot be a simple permutation. ${\rm p_p}$-matrices represent, at most, one correspondence for each feature. If row $\mathbf{P}_{i\cdot}$ is a row of zeros, then feature $\mathbf{X}_{i\cdot}$ does not have matching feature in $\mathbf{Y}$. If column $\mathbf{P}_{\cdot j}$ is a column of zeros, then feature $\mathbf{Y}_{j\cdot}$ does not have a matching row in $\mathbf{X}$.

Both correspondence and outlier rejection are intrinsic to this formulation because each element of $\mathcal{P}_{\rm p}(p_1, p_2)$ permutes only a subset of all features. The global optimal solution to Problem 1 is the best among all possible point samples and permutations.

We generalize the usual definition of ${\rm p_p}$-matrices to nonsquare matrices, saying that any $p_1 \times p_2$ real matrix $\mathbf{P}$ is a ${\rm p_p}$-matrix *iff* it complies with the following conditions:

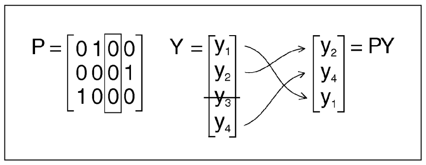

Fig. 2. A partial permutation matrix representing a particular selection and permutation of rows of $\mathbf{Y}$.

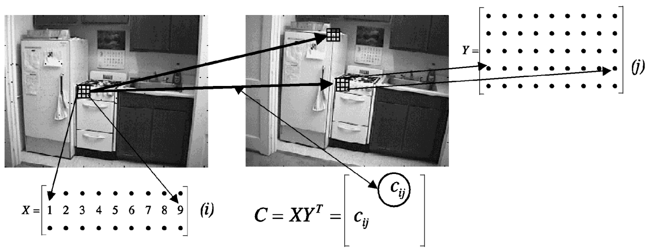

Fig. 3. Data matrices for correlation matching criterion. Each 3×3 window is lined one row of $\mathbf{X}$ and $\mathbf{Y}$. Matrix C represents the cost of each matching.

$$\text{zero-one}: \quad \mathbf{P}_{i,j} \in \{0, 1\}, \quad \forall i \le p_1, \forall j \le p_2, \quad (2)$$

$$\text{row-sum}: \quad \sum_{i=1}^{p_1} \mathbf{P}_{i,j} \le 1, \quad \forall j \le p_2, \quad (3)$$

$$\text{col-sum}: \quad \sum_{j=1}^{p_2} \mathbf{P}_{i,j} \le 1, \quad \forall i \le p_1. \quad (4)$$

To avoid the trivial solution $\mathbf{P}^* = \mathbf{0}$, we establish a fixed number of correspondences $p_{\rm t} \le \min(p_1, p_2)$ by considering a slightly different set of matrices $\mathcal{P}_{\rm p}^{p_{\rm t}}(p_1, p_2)$, the set of *rank-$p_{\rm t}$ partial permutation matrices* (rank-$p_{\rm t}$ ${\rm p_p}$-matrices). This set is constructed by adding (5)

$$\text{rank-}p_{\rm t}: \sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \mathbf{P}_{i,j} = p_{\rm t} \quad (5)$$

to the three previous conditions. Constraining the optimization problem to $\mathcal{P}_{\rm p}^{p_{\rm t}}$ leads to a process of picking up just the best $p_{\rm t}$ correspondences. The case with $p_{\rm t} = p_2 \ge p_1$ yields a very simple formulation which is particularly useful when very few reliable features are extracted from the first image, while the second image is densely sampled. We refer to the resulting set of matrices by $\mathcal{P}_{\rm p}^{\rm c}(p_1, p_2)$, the set of *column-wise partial permutation matrices* (column-wise ${\rm p_p}$-matrices). Definitions and properties of $\mathcal{P}_{\rm p}^{p_{\rm t}}$ and $\mathcal{P}_{\rm p}^{\rm c}$ can be found in Appendix A.1.

2.2 Matching Criteria

Each correspondence method depends upon particular instances of $J()$ and particular representations of $\mathbf{X}$ and $\mathbf{Y}$. Image correlation is one of the most popular criterion in the literature. The goal is to determine the correspondences that maximize the similarity between image patches on two frames. In our case, pixel (brightness) values of $N \times N$ windows are mapped into rows of matrix X—Fig. 3. Likewise, rows of Y contain the window elements of the second frame. With such a representation, any two window correlation is just the dot product between correspondent rows of $\mathbf{X}$ and $\mathbf{Y}$. Thus, matrix $\mathbf{C} = \mathbf{Y}\mathbf{X}^T$ represents all possible window correlation values. The global problem of maximizing all possible correlation is then to permute rows in the second data matrix ($\mathbf{Y}$) such that the sum of the diagonal elements is maximum. In other words, the cost function is represented by $J(\mathbf{X}, \mathbf{Y}, \mathbf{P}) = -trace(\mathbf{P}\mathbf{Y}\mathbf{X}^T)$. This criterion leads to a linear cost function on the variables $\mathbf{P}_{ij}$.

Not all matching problems are modeled by linear functions. In Section 4, we show two second order polynomials which model calibrated stereo matching and 3D point registration. In fact, as long as it has a continuous second derivative, any function can be used as criterion.

2.3 Reformulation with a Compact Convex Domain

Problem 1 is a brute force integer minimization problem. In general, there is no efficient way of finding its optimal solution. Nonetheless, there is a related class of optimization problems for which there are efficient, optimal algorithms. Such a class can be defined as Problem 2.

Problem 2.

$$\mathbf{P}^* = \arg\min_{\mathbf{P}} \quad J_\epsilon(\mathbf{X}, \mathbf{P}, \mathbf{Y})$$
$$s.t. \quad \mathbf{P} \in \mathcal{DS}_{\mathrm{s}}(p_1, p_2),$$

where J_ϵ is a concave[2] version of J, (to be defined later—(17)). $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$ is the set of *doubly substochastic matrices* defined by the row-sum (3), col-sum (4), and a new condition (6):

$$\begin{aligned} \text{row-sum}: &\quad \sum_{i=1}^{p_1} \mathbf{P}_{i,j} \leq 1, &&\forall j \leq p_2,\\ \text{col-sum}: &\quad \sum_{j=1}^{p_2} \mathbf{P}_{i,j} \leq 1, &&\forall i \leq p_1, \\ \text{real in zero-one}: &\quad \mathbf{P}_{i,j} \geq 0, &&\forall i \leq p_1, \forall j \leq p_2. \end{aligned} \tag{6}$$

Set $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$ is the convex hull of $\mathcal{P}_{\mathrm{p}}(p_1, p_2)$, a real compact convex set. In other words, this set results from relaxing the integer domain of $\mathcal{P}_{\mathrm{p}}(p_1, p_2)$ into the continuous domain of its polytope.

Problems 1 and 2 can be shown to be equivalent[3] —Section 2.5. The latter belongs to the class of *concave programming* (CP) problems, which is one of the best-studied classes of problems in global optimization—Section 3. This new formulation has the advantage that several efficient and practical algorithms are available for its resolution. For example, when cost function is linear, the *simplex* algorithm can be used. Most CP algorithms take advantage of the linearity of the constraints and the concavity of the cost function. Their efficiency is also improved if the constraints are written in canonical form and the cost function is an explicit polynomial.

The equivalence of these problems means that we can guarantee that the solution of the relaxed problem is still a p-matrix, with integer (0-1) entries. The same relaxation can be made to $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}$ and $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}$—see Appendix A.1.

2.4 Outline of the Methodology

As explained before, feature point correspondences are determined by the solution of a constrained integer optimization problem, which is very hard to solve. On the other hand, if the optimizing function is concave, the solution found with relaxed constraints is the same, and this we know how to solve efficiently. Furthermore, matching criteria can be any, as long as features are represented by equal-length vectors and cost functions are class C^2—continuous second derivatives—so that a concave equivalent can be found. Each choice of criterion produces a particular correspondence method, for which the global optimal solution is guaranteed to be found with feasible computation.

An outlier rejection mechanism is directly embedded in the formulation. Finally, prior knowledge can be included in the form of extra *support* constraints that cannot be expressed as linear equations of the variables. We use these extra constraints to reduce the dimensionality of the problem, while keeping the special structure of the linear constraints. In stereo matching, for example, limiting the maximum allowed disparity can decrease the search space dramatically. The whole process is outlined as follows:

1. Extract points of interest and build $\mathbf{X}$, $\mathbf{Y}$ — Eq. (1).
2. Use $\mathbf{X}$, $\mathbf{Y}$ to build a cost function $J(\mathbf{P})$.
3. Build the concave equivalent $J_\epsilon(\mathbf{P})$ — Section 2.7.
4. Write $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$ in canonical form — Section 2.6.
5. Eventually add extra constraints — Section 2.8.
6. Solve Problem 3 using a CP algorithm — Section 3.

In the remaining sections, we will present the details. Also, two important issues must be addressed: showing that the integer problem (1) is equivalent to the relaxed problem (2)—Section 2.5—and extending the formulation to multiview—Section 2.10.

2.5 Equivalence of Problems 1 and 2

Theorem 1 states the fundamental reason for the equivalence of Problems 1 and 2.

Theorem 1. *A strictly concave function $J : \mathcal{C} \rightarrow \mathbb{R}$ attains its global minimum over a compact convex set $\mathcal{C} \subset \mathbb{R}^n$ at an extreme point of $\mathcal{C}$.*

In [10], a proof is presented. The constraining set of a minimization problem with a concave objective function can be replaced by its convex-hull, provided that all the points in the original set are extreme points of the new compact set. This is what happens in $\mathcal{DS}_{\mathrm{s}}$. In Appendix A.2, we prove that, for a given p_1 and p_2, $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$ is the convex-hull of $\mathcal{P}_{\mathrm{p}}(p_1, p_2)$ and the set of vertices of $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$ is exactly $\mathcal{P}_{\mathrm{p}}(p_1, p_2)$. A crucial part of this demonstration consists on showing that $\mathcal{DS}_{\mathrm{s}}$ is an integral polytope—all vertices have integer coordinates.

Note that the cost function $J(\mathbf{P})$ needs not to be concave by construction, since we also present a way of building a concave equivalent $J_\epsilon(\mathbf{P})$. Fig. 4 summarizes the whole process. It remains valid in the presence of the rank-fixing constraint because the vertices of $\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}(p_1, p_2)$ are exactly the elements of $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}(p_1, p_2)$ and the vertices of $\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}(p_1, p_2)$ are the elements of $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}(p_1, p_2)$—see Appendix A.2.

Relaxing the constraints of a combinatorial problem into its convex-hull is a well-known method of simplifying its solution [18]. This is particularly useful in problems with constraining integral polytopes because the relaxation is exact

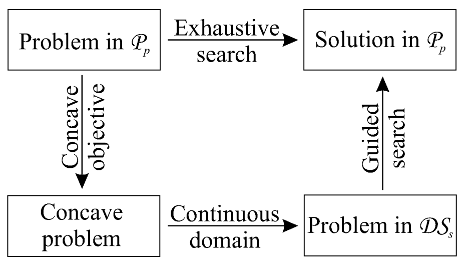
Problem in $\mathcal{P}_p$
Exhaustive search
Solution in $\mathcal{P}_p$
Concave objective
Guided search
Concave problem
Continuous domain
Problem in $\mathcal{DS}_s$

Fig. 4. Efficient solution to the combinatorial problem.

2. Symmetric of a convex.
3. Two optimization problems are equivalent when they have the same global solution.

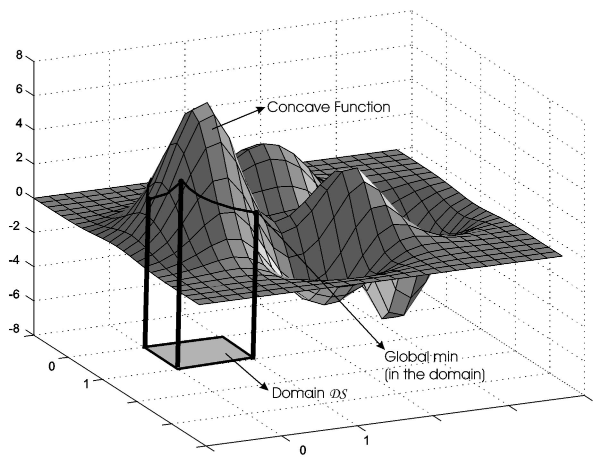

Fig. 5. Minimum of a concave function over a convex compact set is always a vertex.

[10]. This means that the optimal solution of the relaxed problem is still an integer, so there is no need to project the solution on the neglected 0-1 constraints—e.g., rounding. Fig. 5 shows one example.

Exact relaxation is often regarded as an academic exercise because useful constraining polytopes are seldom integral. A classical exception is the set of *doubly stochastic matrices*, which is an integral polytope [9], [2]. It is the convex-hull of the set of *permutation matrices*.

2.6 Constraints in Canonical Form

Most concave and linear programming algorithms assume that problems have their constraints in a special way, called canonical form. We will now express the constraints that define the hypercube $\mathcal{DS}_\mathrm{s}$, in the canonical form. We restate Problem 2 as

Problem 3.

$$\begin{aligned} \mathbf{P}^* = \arg\min_{\mathbf{P}} \quad & J_\epsilon(\mathbf{X},\mathbf{P},\mathbf{Y}) \\ s.t. \quad & \mathbf{A}\mathbf{q} \le \mathbf{b},\ \mathbf{q} \ge \mathbf{0} \\ & \mathbf{q} = \mathrm{vec}(\mathbf{P}). \end{aligned}$$

The natural layout of the variables is a matrix $\mathbf{P}$, so we use $\mathbf{q} = \mathrm{vec}(\mathbf{P})$, where vec() stacks the columns of its argument into a column vector. The row-sum condition (3) can now be written as

$$\sum_{i=1}^{p_1} \mathbf{P}_{i,j} \le 1 \ \Leftrightarrow \mathbf{P}.\mathbf{1}_{[p_2\times 1]} \le 1_{[p_1\times 1]}. \tag{7}$$

Applying the vec() operator [9] to both sides of this inequality, we obtain

$$\left(\mathbf{1}^\top_{[1\times p_2]} \otimes \mathbf{I}_{[p_1]}\right)\mathbf{q} \le \mathbf{1}_{[p_1\times 1]}, \tag{8}$$

where $\otimes$ is the Kronecker product and (3) becomes equivalent to $\mathbf{A}_1\mathbf{q} \le \mathbf{b}_1$ with

$$\mathbf{A}_1 = \mathbf{1}^\top_{[1\times p_2]} \otimes \mathbf{I}_{[p_1]}, \quad \mathbf{b}_1 = 1_{[p_1\times 1]}. \tag{9}$$

We can write similar inequalities for (4) and (5) and (32) of Appendix A.1, which are used in the definitions of $\mathcal{S}_\mathrm{s}^{p_t}$ and $\mathcal{S}_\mathrm{s}^\mathrm{c}$. In conclusion, we define

$$\mathbf{P} \in \mathcal{DS}_\mathrm{s} \Leftrightarrow q_i \in \mathbb{R}_0^+, \forall i \wedge \begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_3\end{bmatrix}\mathbf{q} \le \begin{bmatrix}\mathbf{b}_1\\ \mathbf{b}_3\end{bmatrix}, \tag{10}$$

$$\mathbf{P} \in \mathcal{S}_\mathrm{s}^{p_t} \Leftrightarrow q_i \in \mathbb{R}_0^+, \forall i \wedge \begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_3\\ \mathbf{A}_4\\ \mathbf{A}_5\end{bmatrix}\mathbf{q} \le \begin{bmatrix}\mathbf{b}_1\\ \mathbf{b}_3\\ \mathbf{b}_4\\ \mathbf{b}_5\end{bmatrix}, \tag{11}$$

$$\mathbf{P} \in \mathcal{S}_\mathrm{s}^\mathrm{c} \Leftrightarrow q_i \in \mathbb{R}_0^+, \forall i \wedge \begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\\ \mathbf{A}_3\end{bmatrix}\mathbf{q} \le \begin{bmatrix}\mathbf{b}_1\\ \mathbf{b}_2\\ \mathbf{b}_3\end{bmatrix}. \tag{12}$$

using

$$\mathbf{A}_2 = -\mathbf{1}^\top_{[1\times p_2]} \otimes \mathbf{I}_{[p_1]}, \quad \mathbf{b}_2 = -\mathbf{1}_{[p_1\times 1]}, \tag{13}$$

$$\mathbf{A}_3 = \mathbf{I}_{[p_2]} \otimes \mathbf{1}^\top_{[1\times p_1]}, \quad \mathbf{b}_3 = \mathbf{1}_{[p_2\times 1]}, \tag{14}$$

$$\mathbf{A}_4 = \mathbf{1}^\top_{[1\times p_1p_2]}, \quad \mathbf{b}_4 = p_t, \tag{15}$$

$$\mathbf{A}_5 = -\mathbf{1}^\top_{[1\times p_1p_2]}, \quad \mathbf{b}_5 = -p_t. \tag{16}$$

2.7 Concave Equivalent to a Class C^2 Cost Function

We will now describe how to find a concave function $J_\epsilon : \mathcal{DS}_\mathrm{s}(p_1,p_2) \to \mathbb{R}$ having the same values of the original $J()$ at every point of $\mathcal{P}_\mathrm{p}(p_1,p_2)$. We will only guarantee concavity inside the polytope $\mathcal{DS}_\mathrm{s}(p_1,p_2)$, not over the entire $\mathbb{R}^{p_1p_2}$, but this is enough to verify the conditions of Theorem 1.

Consider our canonical optimization problem (Problem 3), where $J(\mathbf{q})$ is a class C^2 scalar function. Each entry of its Hessian is a continuous function $\mathbf{H}_{ij}(\mathbf{q})$. The concave version of $J(\mathbf{q})$ is

$$J_\epsilon(\mathbf{q}) = J(\mathbf{q}) - \sum_{i=1}^{p_1p_2} \epsilon_i q_i^2 + \sum_{i=1}^{p_1p_2} \epsilon_i q_i. \tag{17}$$

Since $\mathbf{P} \in \mathcal{P}_\mathrm{p}$, all entries $\mathbf{P}_{ij}$ are either 0 or 1. The two extra terms in (17) cancel, so we conclude that $J_\epsilon(\mathbf{q}) = J(\mathbf{q})$ whenever $\mathbf{q} = \mathrm{vec}(\mathbf{P})$ and $\mathbf{P} \in \mathcal{P}_\mathrm{p}$. On the other hand, $\mathcal{P}_\mathrm{p}(p_1,p_2)$ is bounded by hypercube

$$\mathcal{B} = \{\mathbf{q} \in \mathbb{R}^{p_1p_2} : 0 \le q_i \le 1,\ \forall i\}.$$

All $\mathbf{H}_{ij}(\mathbf{q})$ are continuous functions, so they are bounded for $\mathbf{q} \in \mathcal{B}$—Weierstrass' theorem. This means that we can always choose a set of finite values ϵ_i, defined by

$$\epsilon_i \ge \frac{1}{2}\left[\max_{\mathbf{q}}\left(\sum_{j=1,j\ne i}^{p_1p_2} \left|\mathbf{H}_{ij}(\mathbf{q})\right|\right) + \max_{\mathbf{q}}(\mathbf{H}_{ii}(\mathbf{q}))\right], \tag{18}$$

which impose a negative strictly dominant diagonal to the Hessian of J_ϵ, that is, to say,

$$\frac{\partial^2 J_\epsilon(q)}{\partial q_i^2} < -\sum_{j=1,j\ne i}^{p_1p_2}\left|\frac{\partial^2 J_\epsilon(q)}{\partial q_i \partial q_j}\right|, \ \forall i. \tag{19}$$

A strictly diagonally dominant matrix having only negative elements on its diagonal is strictly negative definite [9], so these values of ϵ_i will guarantee that $J_\epsilon(\mathbf{q})$ is concave for $\mathbf{q} \in \mathcal{B}$ and, consequently, also for $\mathbf{q} \in \mathcal{DS}_\mathrm{s}(p_1,p_2)$. The same is true for $\mathcal{P}_\mathrm{p}^{p_t}(p_1,p_2)$ and $\mathcal{P}_\mathrm{p}^\mathrm{c}(p_1,p_2)$ sets. Fig. 6 illustrates this process for a simple 1D example.

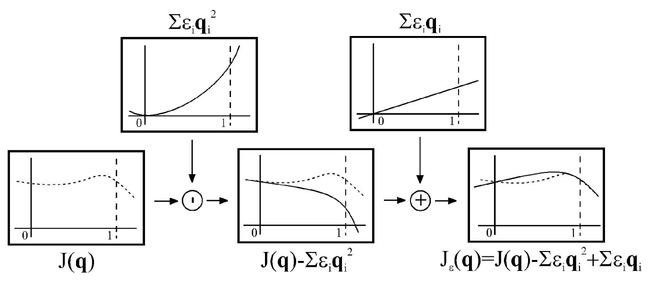

Fig. 6. Finding $J_\epsilon(\mathbf{q})$, concave in $[0,1]$ and such that $J_\epsilon(\mathbf{q}) = J(\mathbf{q}), \forall q_i \in \{0,1\}$.

2.8 Inclusion of Other Constraints

In this section, we describe how to complement Problem 3 with constraints that cannot be expressed as linear equations on the variables. The use of a priori conservative constraints—like epipolar or bounds on the disparity— reduces the dimensionality of the problem and the number of ambiguous solutions. As an example, with 50 points in each image, searching for the 20 best correspondences creates a search space with size 10^{33}. Imposing constraints on the possible candidates in the example of Fig. 7, the search space decreases to 10^9.

This type of a priori knowledge can be represented within our framework. First, express the new constraints by an indicator matrix $\mathbf{S}$—Fig. 7. $\mathbf{S}$ is the support of solution $\mathbf{P}^*$. If entry (i,j) of $\mathbf{S}$ is set to 0, then entry (i,j) of variable $\mathbf{P}$ is permanently set to 0. This means that point i on the first image cannot correspond to point j on the second image. On the other hand, entry (i,j) of $\mathbf{S}$ is set to 1 if entry (i,j) of $\mathbf{P}$ should remain as a variable.

Then, we *squeeze* vector $\mathbf{q}$, eliminating all entries set to 0. Thus, we obtain a new variable $\mathbf{q}^c$ of dimension $n = \sum_{i,j} S_{ij}$. This new variable is such that $\mathbf{q} = \mathbf{B}\,\mathbf{q}^c$, where $\mathbf{B}$ is $[p_1 p_2 \times n]$ *row-wise* p_p-matrix—the transpose of a column-wise $\mathrm{p_p}$-matrix—such that $\mathrm{vec}(\mathbf{S}) = \mathbf{B}.1^\top_{[\mathbf{n}\times\mathbf{1}]}$. Finally, the new constraints are implicitly included in Problem 4 through variable $\mathbf{q}^c$

Problem 4.

$$\mathbf{q}^{c^*} = \arg\min_{\mathbf{q}^c} \quad J_\epsilon(\mathbf{X}, \mathbf{Y}, \mathbf{B}\,\mathbf{q}^c)$$
$$s.t. \quad \mathbf{A}\mathbf{B}\,\mathbf{q}^c \leq \mathbf{b},\ \mathbf{q}^c \geq \mathbf{0}.$$

In Appendix A.3, we show that these constraints also define an integral polytope—vertices remain integer—so that the 0-1 relaxation is still valid.

2.9 Feature Rejection in All Images

In a general case, features on the first image can be rejected using $\mathbf{P}\mathbf{P}^\top\mathbf{X}$. If $\mathbf{P}$ is a fixed-rank $\mathrm{p_p}$-matrix, then $\mathbf{P}\mathbf{P}^\top$ is an identity matrix with some zeros on the diagonal, so points on the first image are rejected wherever $\mathbf{P}$ has a row of zeros. Though effective, this rejection mechanism most times produces cost functions of higher degree. One exception is the linear cost function of Section 4.1.

2.10 Handling Image Sequences

With F frames, feature-points are extracted and arranged in F matrices $\mathbf{X}_f,\ f = 1,\ldots,F$. Correspondences are represented by a set of $F-1$ $\mathrm{p_p}$-matrices collected in variable $\mathbb{P} = [\mathbf{P}_1 \mid \cdots \mid \mathbf{P}_{F-1}]$ and Problem 1 is extended to

Problem 5.

$$\mathbb{P}^* = \arg\min_{\mathbb{P}} \quad J(\mathbf{X}_1,\ldots,\mathbf{X}_F,\mathbb{P})$$
$$s.t. \quad \mathbf{P}_1,\ldots,\mathbf{P}_{F-1} \in \mathcal{P}_\mathrm{p}.$$

The obvious consequence is an increase on the dimensionality and number of constraints. Furthermore, putting the cost function in explicit polynomial form may become even harder. The relaxation to $\mathcal{DS}_\mathrm{s}$ constraints is straightforward. The new vectorized variable is $\mathbf{q} = \mathrm{vec}(\mathbb{P})$, so the canonical constraint matrix $\mathbf{A}$ is block-diagonal with blocks $\mathbf{A}^1,\ldots,\mathbf{A}^{F-1}$. A block diagonal matrix with TU blocks is also TU—Appendix A.4—so the relaxation is still exact.

3 Concave Programming for $\mathcal{DS}_s$ Problems

To minimize nonlinear concave cost functions constrained to convex sets, we cannot rely on local methods because many local minima may occur. Instead, we apply *global optimization algorithms* that exploit both the concavity of the cost function and the convexity of the constraining set. In [14], we give detailed descriptions of three such algorithms which were used in our experiments.

Concave programming is the best studied class of problems in global optimization [10], [21], so our formulation has the advantage that several efficient and practical algorithms are available for its resolution. Among existing optimal methods, cutting-plane, and cone-covering [17] provide the most efficient algorithms, but these are usually hard to implement. Enumerative techniques [20] are the most popular, mainly because their implementation is straightforward.

Recently, special attention has been paid to suboptimal concave minimization algorithms. Júdice and Faustino [11] describe implementations of Frank and Wolfe and Keller algorithms and claims good performances in large-scale sparse problems. Simulated Annealing [3] is also having growing popularity.

We implemented the method of [4]. As iterations run, the *current best* solution follows an ever improving sequence of extreme points of the constraining polytope. On each iteration, global optimality is tested and a pair of upper and lower bounds are updated. Worst case complexity is nonpolynomial but, like the *simplex* algorithm, it typically visits only a fraction of the extreme points. Our implementation takes advantage of the sparse structure of the constraints and deals with redundancy and degeneracy using the techniques of [12].

4 Experiments

In this section, we consider some of the most frequently used assumptions for correspondence and cast them in our global correspondence framework. Each main assumption results on a particular correspondence method, suitable for a given application. For each considered assumption, we develop an explicit cost function and describe the details of the resulting method. Finally, the implementation difficulties are discussed.[4]

4. Code available at http://www.isr.ist.utl.pt/~maciel/code.html.

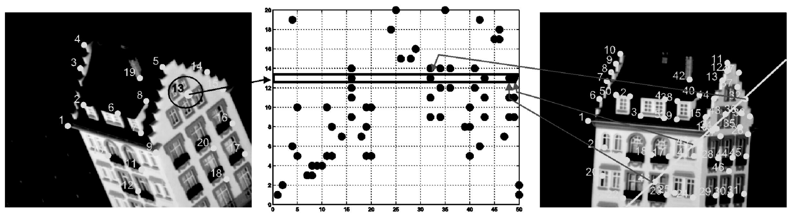

Fig. 7. An example of a support matrix representing the epipolar constraint. Feature 13 on the left produces the right epipolar line. All candidate matching points must be within a certain threshold distance from the line. All other possibilities are not allowed.

4.1 Correlation Matching

Matching by correlation of image patches is the most popular method for stereo correspondence. It is well-suited to solve situations with short baselines and small photometric distortion. This criterion provides the simplest formulation of the correspondence problem so it can be efficiently used to match large numbers of features. However, other assumptions must be used in order to solve emerging ambiguities and perform outlier rejection. Our formulation solves both problems in a natural way.

Features consist of image patches with N pixels centered around the previously segmented points of interest. Row i of $\mathbf{X}$ (and $\mathbf{Y}$) is the row vectorization of a patch around the $i\mathrm{th}$ feature-point of the first (and second) image. We normalize the rows of $\mathbf{X}$ and $\mathbf{Y}$ to zero mean and unit norm, producing matrices $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$. The sum of the correlation coefficients of the rows of $\mathbf{X}$ and $\mathbf{Y}$ is given by the matrix inner product of $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$. The objective function of this method is

$$J_{corr}(\mathbf{P}) = -\mathrm{tr}\big(\mathbf{P}\hat{\mathbf{Y}}\hat{\mathbf{X}}^\top\big). \tag{20}$$

Using algebraic properties of the trace operator [9], we obtain

$$J_{corr}(\mathbf{q}) = -\mathrm{vec}\big(\hat{\mathbf{X}}\hat{\mathbf{Y}}^\top\big)^\top \mathbf{q} \tag{21}$$

which is linear in $\mathbf{q} = \mathrm{vec}(\mathbf{P})$. Problem 3 of Section 2.6 with this linear cost function was solved using a *simplex* algorithm. Our implementation takes advantage of the sparse TU structure of the constraints and deals with degeneracy.

Note that this problem cannot be directly solved using bipartite matching. Weighted bipartite matching [8] finds only perfect matches ($p_t = min(p_1, p_2)$). In unweighted bipartite matching, feature rejection hinges on the right threshold setting [5] and cannot optimize the criterion of (21).

4.1.1 Experiment with Real Data

In this first example, we illustrate the whole extent of the methodology using a linear cost function. In particular, we start with a set of automatically selected features, perform feature rejection in all images, and use prior knowledge to reduce the search space (support constraints).

We selected six images from the *Hotel* sequence—Fig. 1. This image sequence presents two challenges: First, all image pairs have high disparity so proximity constraints cannot be used. Second, there are repeated image-patterns so correlation criterion is highly ambiguous.

The experiment assumes we know a coarse estimate of the Fundamental matrix between consecutive views—epipolar constraints. An edge detection algorithm was used to segment 5,000 feature-points on each image. Support matrices were built which eliminate candidates farther than seven pixels from the epipolar lines, thus reducing the dimensionality of the global problem. Correspondences between consecutive image pairs were computed by optimizing the correlation cost function of (21) using the *simplex* algorithm. p_t was set to 3,000. We then looked for features with correspondence across all six images and built a feature track matrix $\mathbf{W}$ suitable for Tomasi-Kanade's factorization method [27] with 1,000 observations. The row space of $\mathbf{W}$ was computed by SVD and the 100 points most distant to its rank-3 subspace were removed. This was done to remove possible outliers. Finally, the factorization method was applied to the remaining 900 points.

The results in Fig. 8 show the reconstruction from the computed matches. Each linear problem was solved in a fraction of a second by a *simplex* algorithm running on a Pentium processor. Inspection of the images of Fig. 8 does not reveal any evident spurious point nor major global distortion on the reconstruction. In the next sections, we evaluate the performance of this method qualitatively.

4.1.2 Performance Evaluation

We compared the performance of this method against the procedure of [28]. This benchmarking method builds an initial list of candidate pairs using a greedy correlation algorithm. These candidates are pruned for outliers using a random sampling algorithm with an extra rigidity assumption. The outlier rejection algorithm randomly chooses small sets of feature pairs to estimate Fundamental matrices. It then computes the median distances between points and corresponding epipolar lines and chooses the Fundamental matrix that minimizes this criterion. We propose to replace the initial greedy stage by our optimal method and measure the gain in robustness.

We selected one image pairs with large disparity from the *Kitchen* sequence.[5] Images were corrupted with zero-mean Gaussian noise with increasing standard deviation. We then applied a corner detector that locally tuned the position of a set of 75 manually segmented feature points. A second data set was built adding spurious points—Fig. 9. In this section, we impose exactly one match for each point on the first image ($p_t = p_1$). Later on, we study how the method behaves with varying p_t.

We measured the number of incorrect matches returned by the two algorithms in repeated experiments performed on data with increasing levels of noise. The results are summarized on Figs. 10 and 11. In the case of 75 features plus 150 outliers, the cardinality of $\mathcal{P}_\mathrm{p}(p_1, p_2)$ is roughly

5. Data was provided by the Modeling by Videotaping group in the Robotics Institute, CMU.

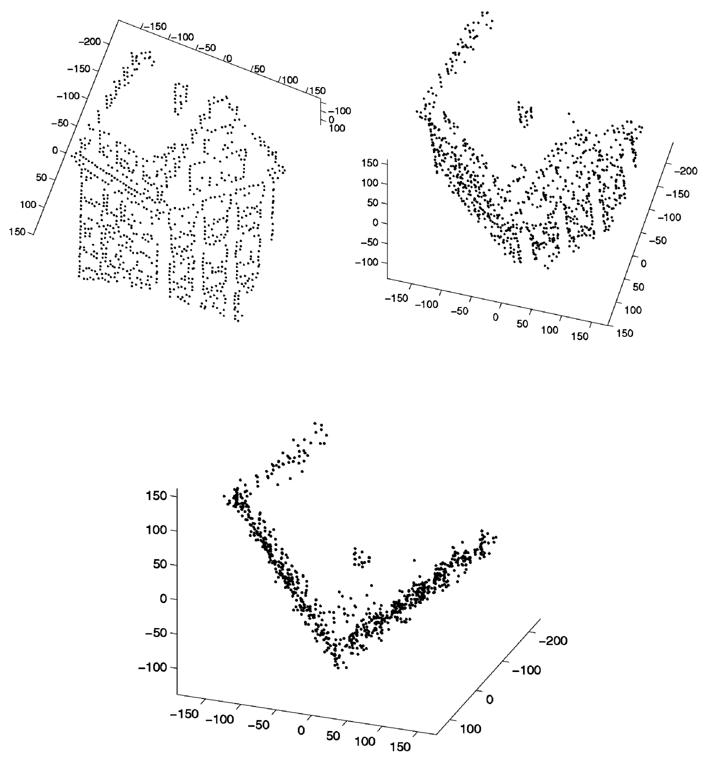

Fig. 8. Three views of a 3D cloud with 900 points.

10^{260}. Exhaustive search would be impractical, while the *simplex* algorithm visits less than 500 solutions. A total of 65,100 experiments were performed.

The original method of [28] consistently produced higher number of mismatches, especially when outliers are present. We observed that, when the greedy algorithm returns more than 40 percent of outliers among the candidates, the validation procedure starts rejecting many good matches. This tends to raise the percentage of wrong matches.

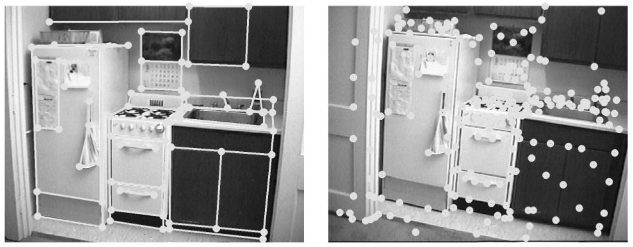

Fig. 9. Two images—480×512 pixels, 256 gray levels—from the *Kitchen* sequence with spurious features. Wireframe is for better perception only.

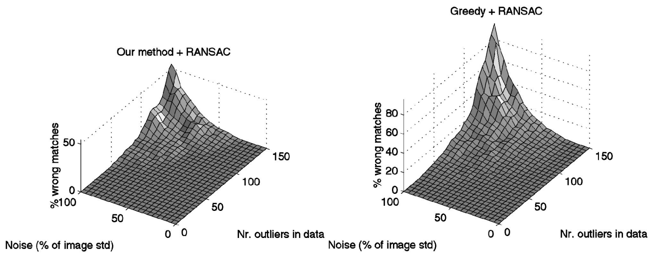

Fig. 10. The average number of incorrect matches found in 100 trials for increasing levels of data corruption.

The simultaneous rejection and correspondence of features is a reliable strategy. The reconstruction in Fig. 12 was obtained in spite of 50 percent outliers in data and noise with 50 percent of the signal standard deviation. With such an amount of image noise, the corner detector returned features with location errors up to eight pixels. These errors produced a highly distorted reconstruction but the correspondences were all correct.

4.1.3 Sensitivity to the Number of Selected Features

The theory of this paper revolves around the idea of finding an optimal subset of correspondences from the set of all possible candidates. However, nothing has been said about the size of this subset. In other words, one must know what is a "reasonable" value for p_t. To have a more precise estimate, we ran a sensitivity test to evaluate how well the method performs as a function of the number of required matches.

Fig. 13 shows the percentage of wrong matches (number of wrong matches $/p_t$) *versus* p_t. The experiment was done with images shown in the previous section. For each value of p_t, the graph shows the average number of mismatches in

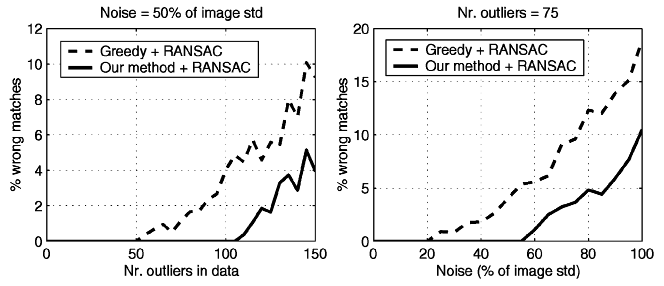

Fig. 11. Comparison of two profiles from the plots of Fig. 10.

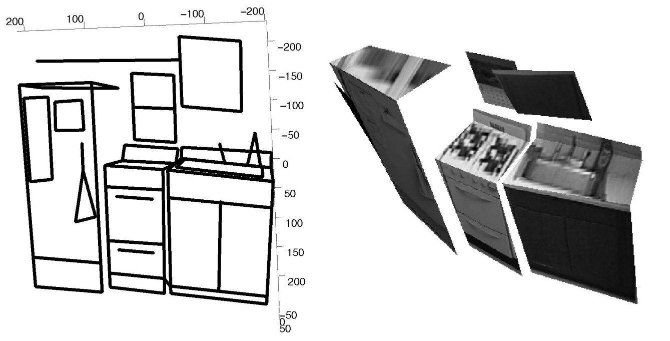

Fig. 12. Wireframe and texture-mapped VRML reconstructions using 50 percent of outliers in data, and noise with 50 percent of images standard deviation.

80 runs with different noise sample and outlier features. The important factors are also depicted in Fig. 13. There are zero errors even when p_t is close to the maximum number of available correct features. The number of wrong matches increases when there are no more correct features available for matching. For small values of p_t, the number of mismatches also grows. Though image dependent, the errors grow for fewer required matches. This is due to an increase in ambiguity since there are many features that look alike for the few required correspondences. In other words, if the number of candidates is large and the required number of matches is small, there are a lot more possibilities of finding a wrong match with high correlation for every feature. The strong constraint of global matching does not make much of a difference here. Under these circumstances, the limitation is more on the criterion itself that does not discriminate features then on the method.

4.2 Matching in a Calibrated Trinocular System

Consider a trinocular system in generic configuration—-focal points are not colinear—for which we know all Fundamental Matrices. Fig. 14 shows the notation. Each Fundamental matrix $\mathbf{F}_{k,l}$ defines p_l epipolar lines $\mathcal{L}^m_{k,l}, m = 1, \dots, p_l$ on image k. A point on image k corresponding to the mth point on image l must lie *close* to $\mathcal{L}^m_{k,l}$. We arrange the distances between every possible pair of point and the epipolar line in matrices $\mathbf{D}_{1,2}$, $\mathbf{D}_{2,3}$, and $\mathbf{D}_{1,3}$. $\mathbf{D}_{k,l}(i,j)$ contains the distances between points $i = 1, \dots, p_k$ of image k and the epipolar lines $\mathcal{L}^j_{k,l}$.

We want to compute a set of correspondences that minimize the sum of distances between each point and the corresponding epipolar line. The variable of this problem is $\mathbb{P} = [\mathbf{P}^\top_{1,2} \mid \mathbf{P}_{2,3}]$. We close the loop by estimating the

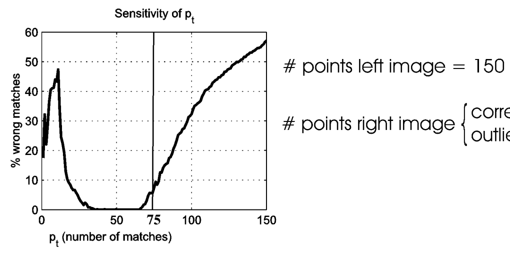

Fig. 13. Testing the senstivity of p_t. The vertical line represents the maximum number of correct canditates available.

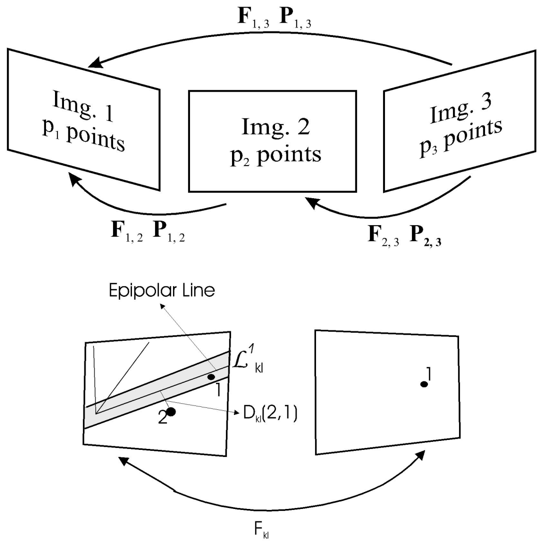
$\mathbf{F}_{1,3}$ $\mathbf{P}_{1,3}$
Img. 1
p_1 points
Img. 2
p_2 points
Img. 3
p_3 points
$\mathbf{F}_{1,2}$ $\mathbf{P}_{1,2}$
$\mathbf{F}_{2,3}$ $\mathbf{P}_{2,3}$
Epipolar Line
$\mathcal{L}^1_{kl}$
1
2
$D_{kl}(2,1)$
1
F_{kl}

Fig. 14. Notation for a trinocular system.

compound correspondence $\hat{\mathbf{P}}_{1,3} = \mathbf{P}_{1,2}\mathbf{P}_{2,3}$. The objective function is

$$J_{tri} = \sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \big(\mathbf{P}_{1,2} \odot \mathbf{D}_{1,2} + \mathbf{P}_{2,3} \odot \mathbf{D}_{2,3} + \hat{\mathbf{P}}_{1,3} \odot \mathbf{D}_{1,3}\big) + \lambda\big[J_{corr}(\mathbf{P}_{1,2}) + J_{corr}(\mathbf{P}_{2,3}) + J_{corr}(\hat{\mathbf{P}}_{1,3})\big],$$

where $\odot$ is the element-wise product. The addition of correlation terms J_{corr}—Section 4.1—is used to remove ambiguities. The value of weight λ is chosen experimentally. By algebraic manipulation, we obtain the quadratic objective function

$$J_{tri}(\mathbf{q}) = \mathbf{q}^\top \mathbf{J}_{tri}\mathbf{q} + \mathbf{c}^\top_{tri}\mathbf{q} \tag{22}$$

with $\mathbf{q} = \mathrm{vec}(\mathbb{P})$. We reduce the dimensionality of the problem using the support matrices $\mathbf{S}_{k,l}(i,j) = \mathbf{D}_{k,l}(i,j) \le \delta, \; \forall i,j$ (see Fig. 14). An entry (i,j) of $\mathbf{S}_{k,l}$ is set to 1 if the ith point on image k is close to $\mathcal{L}^j_{k,l}$. These constraints are included in the problem as described in Section 2.8. We recover each one of the full variables through $\mathrm{vec}\,(\mathbf{P}_{k,l}) = \mathbf{B}_{k,l}\,\mathbf{p}^c_{k,l}$.

J_{tri} is, in general, not concave, so a concave version J_ϵ was computed using (17) and (18) before the minimization algorithm was applied.

4.2.1 Results

We applied the described method to the images of Fig. 15. Points were extracted by an edge detector with a bucketing procedure to increase feature sparsity. A total of 500 points were extracted from the first image and 1,500 from the remaining. The second and third images contain, at least, 1,000 outliers, so the problem was solved in the presence of more than 65 precent of outliers in the data.

We then manually segmented 20 points from each image, and used them to compute fundamental matrices between each pair of images. The width of the epipolar bands was set to 15 pixels because of large errors on the fundamental matrices and large distances between consecutive points on an edge. Maximum disparity was set to 50 pixels.

The quadratic problem was solved, taking more than 30 minutes of CPU time. The result was an observation

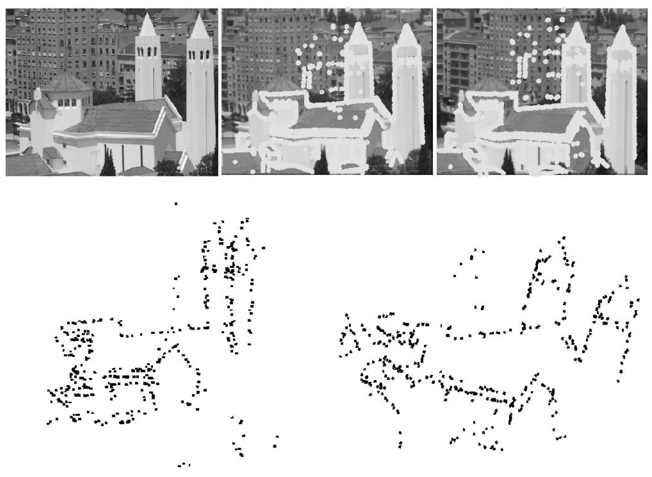

Fig. 15. *Church* images, extracted points and views of the generated 3D could with 450 points.

matrix $\mathbf{W}$ with 500 observations. The row space of $\mathbf{W}$ was computed by SVD, and the 50 points most distant to its rank-3 subspace were removed. This was done to remove existing outliers. Finally, the factorization method was applied to the remaining 450 points. The resulting reconstruction is shown in Fig. 15.

We could find some wrong matches. Given the scene complexity it is hard to fully evaluate its performance. There is also a large amount of noise on the reconstructed points. When several candidates exist on the intersection of two epipolar bands, they are disambiguated using the correlation term of the cost function. Correlation is not a good criterion to match edges because of directional uncertainty. To correct this, either better estimation of the epipolar geometry or a different disambiguating criterion should be used.

4.3 2D Point Registration

In this section, we describe one method to performing 2D image registration with affine models. We propose to search in correspondences that best 2D affine transformation of the first image feature-point set. We also describe a lip-tracking application. Figs. 16 and 17 show one example. Also, this experiment reveals the importance of a good design of $J()$. In many situations, we can solve the same problem with much simpler models which, in the computational domain, can make quite a difference.

Fig. 16 illustrates how to represent row and column coordinates of feature points in matrices $\mathbf{X}$ (and $\mathbf{Y}$). Our goal is to find a $\mathrm{p_p}$-matrix $\mathbf{P}^*$ such that $\mathbf{CX}$ [6] $\mathbf{CP}^*\mathbf{Y}$ are related by a linear transformation (in the LSE sense). This is translated into the following cascaded optimization problem, where $\mathbf{L}$ is the 2D linear transformation:

Problem 6.

$$\mathbf{P}^* = \arg \min_{\mathbf{P}} \left(\min_{\mathbf{L}} \|\mathbf{CX} - \mathbf{CPYL}\|^2 \right)$$
$$s.t. \quad \mathbf{P} \in \mathcal{P}_{\mathrm{p}}^{\mathrm{c}}(p_1, p_2).$$

This problem is equivalent to choosing $\mathbf{P}$ such that the observation matrix

6. Matrix $C_{p_1 \times p_1} = I - \frac{1}{p_1} 1_{p_1 \times p_1}$ centers feature coordinates to zero mean (centroid). The translation component is eliminated.

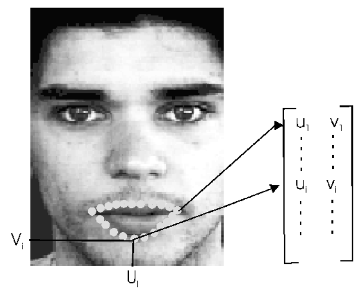

Fig. 16. Lip tracking template.

$$\mathbf{W_P} = [\mathbf{CX} \mid \mathbf{CPY}] \tag{23}$$

is rank-2. Matrix $\mathbf{CX}$ spans the whole subspace where the observations lie. We can a priori compute $\mathbf{\Pi}$—the orthogonal projector[7] onto the column space of $\mathbf{CX}$—and use it in the objective function

$$J_{2D}(\mathbf{P}) = \|\mathbf{\Pi CPY}\|^2. \tag{24}$$

The maximum of this second order polynomial solves the registration problem. Since there is no rejection of model points, this problem is equivalent to minimizing the projection on the null space of $\mathbf{CX}$. In practice, we use the following approximation

$$\tilde{J}_{2D}(\mathbf{P}) = |\mathbf{\Pi}_u\mathbf{CPu}| + |\mathbf{\Pi}_u\mathbf{CPv}| + |\mathbf{\Pi}_v\mathbf{CPu}| + |\mathbf{\Pi}_v\mathbf{CPv}|, \tag{25}$$

where $\mathbf{\Pi}_u$ and $\mathbf{\Pi}_v$ are basis vectors of the column space of $\mathbf{CX}$, and $\mathbf{u}$ and $\mathbf{v}$ are the two columns of $\mathbf{CY}$. This is equivalent to solving a linear problem with objective function

$$\tilde{J}^i_{2D}(\mathbf{P}) = \pm(\mathbf{\Pi}_u\mathbf{CPu}) \pm (\mathbf{\Pi}_u\mathbf{CPv}) \pm (\mathbf{\Pi}_v\mathbf{CPu}) \pm (\mathbf{\Pi}_v\mathbf{CPv}) \tag{26}$$

for all 16 possible sign combinations. We solve all 16 linear problems, set the signs that make all terms positive and choose the solution with largest objective.

4.3.1 Results

We applied the described method to a sequence of images of a talking person. A set of 20 feature points was manually extracted on the first image—Fig. 16. This set of points is the shape model. On each subsequent image, 130 edge points were automatically extracted around the mouth area, and used to build the objective function of (26). Fig. 17 shows extracted points on five of the 20 test images. Matched points are marked with bigger dots. The last image was artificially rotated, to test situations with large deformations and disparities. The linear problems were solved using a *simplex* algorithm and the proper solution chosen. Finally, matched points were used to estimate the linear deformation $\mathbf{L}$ of Problem 6. Each frame took around 8 seconds of processing time on a Pentium processor. This is dramatically shorter time than what it takes for the higher-order problems like trinocular stereo.

We compared the performance of this method with three well-known, simple and effective methods for matching 2D point patterns and curves. The first method is Iterative Closest Point (ICP) [30]. This method finds the closest points

7. $\mathbf{\Pi} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ projects vectors on the $span(\mathbf{X})$

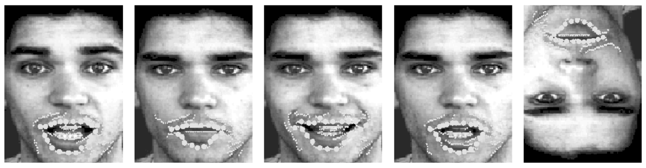
Fig. 17. Extracted edges on five of the 20 test images. Matched edges are marked with bigger dots.

between the two images satisfying a maximum tolerance D_{max} for distance. It then removes outliers through a statistical analysis of distances. Rigid motion is then computed solving a system of equations in least-squares sense. We computed a full affine transformation of the centered data, instead. The computed transformation is then applied to all points. The procedure is iterated until it converges.

The second benchmark method was proposed by Scott and Longuet-Higgins [23]. This method uses the principles of proximity and exclusion (one-to-one match). It minimizes the sum of squared distances between matched points. It starts computing a Gaussian-weighted distance matrix

$$G_{i,j} = e^{-\frac{\left\|\mathbf{x}^1_i - \mathbf{x}^2_j\right\|^2}{2\sigma^2}}, \tag{27}$$

where $\mathbf{x}^f_i$ is the coordinate vector of the ith feature from frame f. An affinity matrix is then computed by $\mathbf{P} = \mathbf{U}\mathbf{V}^\top$, where columns of $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors of $\mathbf{G}$. Large entries of $\mathbf{P}$ indicate strongly coupled features, so they are considered good matches, as long as the uniqueness constraint is held.

The third and final method was proposed by Shapiro and Brady [24]. It is a modification of the previous method, intended to solve situations with large translation, rotation, and scaling. It starts measuring intraframe point distances and uses them to compute the symmetric matrix:

$$\mathbf{H}^f_{i,j} = e^{-\frac{\left\|\mathbf{x}^f_i - \mathbf{x}^f_j\right\|^2}{2\sigma^2}}. \tag{28}$$

Each frame is represented by matrix $\mathbf{U}^f$ with the singular vectors of $\mathbf{H}^f$. The first elements of each row of $\mathbf{U}$ are the coordinates of a point in the reference system of the principal modes of the point set. Correlation between all possible pairs of rows of $\mathbf{U}$ is computed and the results are stored in an affinity matrix $\mathbf{P}$ that is used like in the method of Scott and Longuet-Higgins.

To compare the performance of all four algorithms, we computed the average distance between computed matches and manually segmented ground truth points. The results are shown in Fig. 18.

Fig. 18 clearly shows that our method is more robust. In return, it requires slightly heavier computation. Among all the benchmark algorithms, ICP is the best suited for this application. Both the first and second benchmark methods use proximity as the major criterion. They clearly fail on the last image because they get stuck on local minima. The large rotation is never captured. The last algorithm fails because of the large number of spurious points, which affects the shape modes.

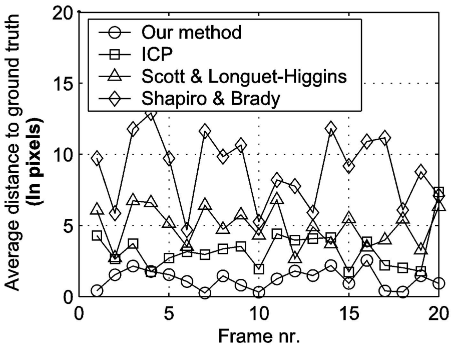

Fig. 18. Performance of four different lip-tracking algorithms.

4.4 3D Point Registration

In this section, we describe a method of performing 3D point registration, similar to the 2D registration of Section 4.3. Noncontact 3D reconstruction systems provide only partial views of the objects that must be combined in complete descriptions.

We group the 3D point coordinates in matrices

$$\mathbf{X} = \begin{bmatrix} x^1_1 & y^1_1 & z^1_1 \\ \vdots & \vdots & \vdots \\ x^1_{p_1} & y^1_{p_1} & z^1_{p_1} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} x^2_1 & y^2_1 & z^2_1 \\ \vdots & \vdots & \vdots \\ x^2_{p_2} & y^2_{p_2} & z^2_{p_2} \end{bmatrix}. \tag{29}$$

The goal is to find a 3D rigid transformation that aligns a fraction of these points, and reject nonoverlapping points. We propose to search for a general linear transformation that aligns subsets of the data, solving Problem 7, where $\mathbf{L}$ is a 3D linear transformation.

Problem 7.

$$\mathbf{P}^* = \arg\min_{\mathbf{P}} \left(\min_{\mathbf{L}} \|\mathbf{CX} - \mathbf{CPYL}\|^2 \right)$$
$$s.t. \quad \mathbf{P} \in \mathcal{P}_{\mathrm{p}}(p_1, p_2).$$

With 3D data, the approximate linear cost function—(26)—transforms to

$$\tilde{J}^i_{3D}(\mathbf{P}) = \pm(\mathbf{\Pi}_x\mathbf{CPx}) \pm (\mathbf{\Pi}_x\mathbf{CPy}) \pm \ldots \pm (\mathbf{\Pi}_z\mathbf{CPz}), \tag{30}$$

where $\mathbf{\Pi}_x$, $\mathbf{\Pi}_y$, and $\mathbf{\Pi}_z$ are basis vectors of the column space of $\mathbf{CX}$, and $\mathbf{x}, \mathbf{y}, \mathbf{z}$ refer to the second set. There are 512 possible combinations of signs. Again, we solve all the 512 linear problems, set the signs that make all terms positive, and choose the solution with largest objective.

In order to eliminate points on both sets, we use $\mathcal{P}^{p_t}_{\mathrm{p}}(p_1, p_2)$ constraints. Points with large coordinate values are priviledged, so wrong solutions appear when p_t is small. On the other hand, when p_t is close to p_1, some spurious points are not eliminated.

To overcome this difficulty, we set p_t to 90 percent of p_1 and used a *ransac* procedure to find the most consistent points. We solved Problem 7 using several random samples of the first data set. The resulting transformations were applied to all the data and points with close matches were voted. After a number of trials, points with highest scores were used to compute a final rigid transformation.

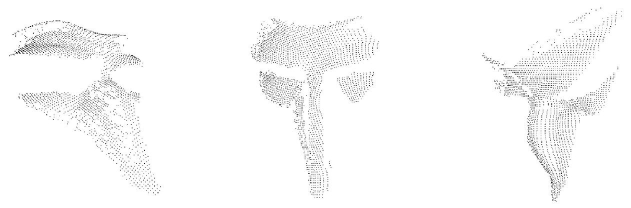
Fig. 19. Three point sets to be registered.

4.4.1 Results

We used a structured-light 3D reconstruction system to build three clouds of 3D points—Fig. 19. The set in the middle is reference $\mathbf{X}$.

Each set contains around 1,600 points, from which 300 were randomly selected. The *ransac* procedure consisted of 30 trials, each randomly choosing $p_1 = 100$ points from the reference set. Support constraints for disparity bounds were added to reduce problem dimensions down to 6,000. Linear problems were solved in approximately 1.2 seconds. On each one of the 30 trials, the best 500 points were voted. At the end, points with more than 10 votes were used in a final problem. The resulting matches were used to compute the desired rigid transformation. Finally, an ICP-like algorithm was used to refine the registration. It measures distances between points in one set and linearly interpolated surface patches on the other. It uses local least-squares to compute the rigid transformation that minimizes the sum of the 30 percent smallest distances. Fig. 20 shows the registration of the three data sets.

When point sets are related by large transformations and contain many spurious points, ICP and similar methods fail, except if a good initial guess is provided. The proposed method is suitable to automatically provide such initial guesses, but requires heavy computation.

The biggest practical difficulty in 3D registration is the fact that, usually, many spurious points must be eliminated from all data sets. Furthermore, there is usually no exact match between data sets. To solve such situations with precision, point sets should first be interpolated and cost functions should be computed using interpolated values.

Finally, performing surface interpolation on registered point clouds becomes a new problem because topological information is lost.

5 Conclusion

We have shown a methodology to solve the correspondence problem, which avoids unwanted assumptions by requiring their explicit statement. Furthermore, it reliably handles outliers, even in situations where other robust methods fail.

The most important limitation of the methodology is the dimensionality of the optimization problems, especially when the objective functions are high-order polynomials. A practical way of minimizing this is the inclusion of additional a priori constraints, with minor changes to the underlying formulation of the problem. Ongoing work is being conducted on the implementation of an efficient algorithm for high-order polynomial problems. These will allow us to deal with the assumption of rigidity under various camera models—see [15].

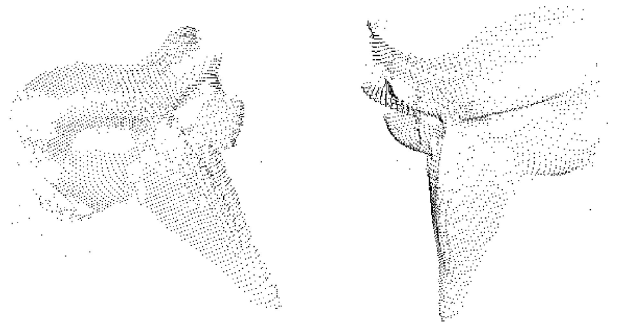
Fig. 20. Two views of the registered 3D data sets.

Appendix A
Partial-Permutation and Related Matrices

A.1 Definitions

As stated in Section 2.1, we generalize the usual definition of $\mathrm{p_p}$-matrices to nonsquare matrices, defining them using (2), (3), and (4).

A $[p_1 \times p_2]$ $\mathrm{p_p}$-matrix $\mathbf{P}$ is rank-p_{t} *iff* it also complies with (31)

$$\sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \mathbf{P}_{i,j} = p_{\mathrm{t}}. \tag{31}$$

The set of rank-p_{t} $\mathrm{p_p}$-matrices of dimension $p_1 \times p_2$ is denoted by $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}(p_1, p_2)$. The case $p_{\mathrm{t}} = p_2 \geq p_1$ is simpler because (4) and (31) can be changed to a single (32):

$$\sum_{j=1}^{p_2} \mathbf{P}_{ij} = 1 \ , \ \forall i = 1 \dots p_1. \tag{32}$$

The resulting set of matrices $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}(p_1, p_2)$ is denoted as the set of *column-wise partial permutation matrices*.

As stated in Section 2.3, we define $\mathcal{DS}_{\mathrm{s}}$—the set of *doubly substochastic matrices*—by (3), (4), and (6). Appendix A.2 shows that, for given p_1 and p_2, $\mathcal{DS}_{\mathrm{s}}$ is the convex-hull of $\mathcal{P}_{\mathrm{p}}$, and that every element of $\mathcal{P}_{\mathrm{p}}$ is a vertex of $\mathcal{DS}_{\mathrm{s}}$.

The convex-hull of $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}$ is $\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}$—*rank p_{t} doubly substochastic matrices*. It is defined by (3), 4), (6), and (5). As shown in Appendix A.2, for given p_1, p_2, and p_{t}, the vertices of $\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}$ belong to $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}$ and every element of $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}$ is a vertex of $\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}$.

Finally, $\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}$ stands for the set of *column-wise substochastic matrices*. It is the convex-hull of $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}$, and is defined by (3), (4), and (6). In Appendix A.2, it is shown that $\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}$ is the convex-hull of $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}$, and that every element of $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}$ is a vertex of $\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}$.

A.2 Integral Property of $\mathcal{DS}_{\mathrm{s}}$ and Related Sets

In this section, we prove the following propositions

Proposition 1. *For given p_1, p_2, the elements of $\mathcal{P}_{\mathrm{p}}(p_1, p_2)$ are the vertices of $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$.*

Proposition 2. *For given p_1, p_2, the elements of $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}(p_1, p_2)$ are the vertices of $\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}(p_1, p_2)$.*

Proposition 3. *For given p_1, p_2, the elements of $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}(p_1, p_2)$ are the vertices of $\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}(p_1, p_2)$.*

These results are generalizations of Birkhoff's theorem—see [9], [2], [18]—which states that the set of $n \times n$ *doubly stochastic* matrices is a compact convex set whose extreme points are permutation matrices. Our proofs are inspired in the approach of [18].

All three results comprise a necessity and a sufficiency condition. We need to prove that being an element of $\mathcal{P}_{\mathrm{p}}(p_1, p_2)$ is both a sufficient and a necessary condition for a $p_1 \times p_2$ matrix to be an extreme point of $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$. In Appendix A.2.1, we prove that every element of $\mathcal{P}_{\mathrm{p}}$ is an extreme point of $\mathcal{DS}_{\mathrm{s}}$—sufficiency—showing how to write any $\mathrm{ds_s}$-matrix as a convex combination of $\mathrm{p_p}$-matrices. This also proves that $\mathcal{DS}_{\mathrm{s}}$ is a bounded convex polytope. By the same token, we can prove sufficiency for $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}$ and $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}$, so we omit this step. Necessity is proven in Appendix A.2.2, showing that vertices of $\mathcal{DS}_{\mathrm{s}}$, $\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}$, and $\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}$ have integer coordinates. Since they are constrained to the interval $[0, 1]$, then they can only be either 0 or 1 so (2) is satisfied by the vertices.

A.2.1 Sufficiency

Consider that $\mathbf{P}$ is a $[p_1 \times p_2]$ $\mathrm{p_p}$-matrix. If we suppose that $\mathbf{P}$ is not an extreme point of $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$, then it will be possible to find two matrices $\mathbf{P}_1, \mathbf{P}_2 \in \mathcal{DS}_{\mathrm{s}}(p_1, p_2)$ such that $\mathbf{P}_1 \neq \mathbf{P}_2 \neq \mathbf{P}$ and two positive scalars α_1, α_2 with $\alpha_1 + \alpha_2 = 1$ such that $\alpha_1\mathbf{P}_1 + \alpha_2\mathbf{P}_2 = \mathbf{P}$. Nonnegativity conditions assure that the entries of $\mathbf{P}_1$ and $\mathbf{P}_2$ that correspond to zeros of $\mathbf{P}$ are zero. Since $\mathbf{P}$ has, at most, one nonzero entry per row and per column, then the same happens to $\mathbf{P}_1$ and $\mathbf{P}_2$. Their row and column sums are lower or equal than 1—they belong to $\mathcal{DS}_{\mathrm{s}}$—so every nonzero entries must be equal or smaller than 1. The only solution left is $\mathbf{P}_1 = \mathbf{P}_2 = \mathbf{P}$ which contradicts the initial assumption.

A.2.2 Necessity

We now show that the constraint matrices of (10), (11), and (12), that define $\mathcal{DS}_{\mathrm{s}}$, $\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}$, and $\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}$, satisfy the conditions of Theorem 2—proven in [18].

Theorem 2. *If* $\mathbf{A}$ *is a Totally Unimodular (TU) matrix of size* $m \times n$, *then the polytope* $\mathcal{C} = \{\mathbf{q} \in \mathbb{N}^n : \mathbf{A}\mathbf{q} \leq \mathbf{b}\}$ *is integral for all* $\mathbf{b} \in \mathbb{Z}^m$ *for which* $\mathcal{C}$ *is not empty.*

A matrix $\mathbf{A}$ is TU if the determinant of every square submatrix of $\mathbf{A}$ is 0, 1 or -1. To show that a given $\mathbf{A}$ matrix is TU, we use the following result, also proven in [18].

Theorem 3. $\mathbf{A}$ *is an* $m \times n$ *TU matrix iff for each of its row selections using row index set* $I \subseteq \{1, \ldots, m\}$, *there exists a partition* I_1, I_2 *of* I *such that*

$$\left| \sum_{i \in I_1} a_{ij} - \sum_{i \in I_2} a_{ij} \right| \leq 1, \quad \forall j = 1, \ldots, n \tag{33}$$

We know in advance that $\mathcal{DS}_{\mathrm{s}}$, $\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}$, and $\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}$ are not empty. Also, the definitions of these polytopes use integer $\mathbf{b}$ vectors —(9), (13), (14), (15), and (16). Therefore, showing that their $\mathbf{A}$ matrices satisfy (33) ensures that the conditions of Theorem 2 are met and, therefore, that $\mathcal{DS}_{\mathrm{s}}$, $\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}$, and $\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}$ are integral polytopes.

A.2.3 Integral Property of $\mathcal{DS}_{\mathrm{s}}$

The matrix $\mathbf{A}$ of (10) satisfies the conditions of (33) if, for every row selection I, we choose a partition so that I_1 selects only rows from block $\mathbf{A}_1$ and I_2 selects only rows from block $\mathbf{A}_3$. Each block contains at most one nonzero element per column, so (33) always holds. Since $\mathbf{A}$ is TU, b is an integer, and $\mathcal{DS}_{\mathrm{s}}$ is not empty, then all the vertices of $\mathcal{DS}_{\mathrm{s}}$ are integer.

A.2.4 Integral Property of $\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}$

Matrix $\mathbf{A}$ of (11) satisfies the conditions of (33) if, for each row selection I, we choose a partition so that column sums will be constrained to $\{-1, 0, 1\}$. We do so the following way:

If I does not include neither $\mathbf{A}_4$ *nor* $\mathbf{A}_5$: Build the partition as in Appendix A.2.3, that is, I_1 should include only rows from block $\mathbf{A}_1$ and I_2 should include only rows from block $\mathbf{A}_3$.

If I includes both $\mathbf{A}_4$ and $\mathbf{A}_5$ *simultaneously*: Put all rows from $\mathbf{A}_1$ in I_1 and the remaining in I_2—from $\mathbf{A}_3$, $\mathbf{A}_4$, and $\mathbf{A}_5$. Note that every column contains entries 1 and -1 that cancel out.

If I includes $\mathbf{A}_4$ *but not* $\mathbf{A}_5$: Put $\mathbf{A}_1$ and $\mathbf{A}_3$ in I_1. I_2 will indicate a single row $\mathbf{A}_4$. The I_1 part of each column will sum either 0, 1, or 2, and every column will be subtracted by an entry 1 from I_2.

If I includes $\mathbf{A}_5$ *but not* $\mathbf{A}_4$: Put all rows in I_1 and leave I_2 empty.

A.2.5 Integral Property of $\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}$

The matrix $\mathbf{A}$ of (12) is in the conditions of (33). For each row selection I, choose a partition in the following way:

If I does not include $\mathbf{A}_2$: Choose a partition like for $\mathcal{DS}_{\mathrm{s}}$.

If I includes both $\mathbf{A}_1$ *and* $\mathbf{A}_2$: Choose a partition like for $\mathcal{DS}_{\mathrm{s}}$ and put all the rows of $\mathbf{A}_2$ in the same partition as those of $\mathbf{A}_1$.

If I includes $\mathbf{A}_2$ *but not* $\mathbf{A}_1$: Choose a partition like for $\mathcal{DS}_{\mathrm{s}}$ and put all the rows of $\mathbf{A}_2$ in the same partition as those of $\mathbf{A}_3$.

A.3 $\mathcal{DS}_{\mathrm{s}}$ with Support Constraints Remains Integral

In this section, we show that the constraints of Problem 4 in Section 2.8 still define an integral polytope. This is enough to show that the solution of Problem 4 is an element of $\mathcal{P}_{\mathrm{p}}$—or $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}$ or $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}$. In Appendix A.2.2, we show that it is only required that the constraint matrix $\mathbf{AB}$ remains Totally Unimodular (TU). Note that $\mathbf{B}$ is a submatrix—a column selection—of a $[p_1p_2 \times p_1p_2]$ identity matrix, so $\mathbf{AB}$ is a submatrix of $\mathbf{A}$. Since $\mathbf{A}$ is TU—Section A.2.2—and since, by definition, any submatrix of a TU matrix is also TU, then $\mathbf{AB}$ is necessarily TU, so Problem 4 has an integer solution.

A.4 Block-Diagonal TU Matrices

In this section, we prove that, if a block-diagonal matrix $\mathbf{A}$ has TU blocks, then $\mathbf{A}$ is also TU.

Consider that $\mathbf{A}$ has F blocks ($\mathbf{A}^f$ with $f = 1, \ldots, F$). Each block $\mathbf{A}^f$ has dimension $m_f \times n_f$, so the total dimension of $\mathbf{A}$ is $m \times n$ with $m = \sum_{f=1}^{F} m_f$ and $n = \sum_{f=1}^{F} n_f$. Recall Theorem 3 that states a necessary and sufficient condition for a matrix to be TU. Each row selection of $\mathbf{A}$ using row index set $I \subseteq \{1, \ldots, m\}$ corresponds to a certain row selection I^f of each block $\mathbf{A}^f$, for which it is possible to build partitions I_1^f, I_2^f that will ensure that (33) holds. We known this because Theorem 3 is a necessary condition for TU, so, since each block $\mathbf{A}^f$ is TU by definition, then they must satisfy the conditions of the theorem. We can, therefore, build the partition

$$I_1 = \bigcup_{f=1}^{F} I_1^f, I_2 = \bigcup_{f=1}^{F} I_2^f. \quad (34)$$

The nonzero entries of each column of $\mathbf{A}$ belong to a single block, so (33) holds for all columns of $\mathbf{A}$. We can apply the sufficiency of Theorem 3 and conclude that $\mathbf{A}$ is TU.

Acknowledgments

J. Maciel was supported by Subprograma Ciência e Technologia, 2° Quandro Comunitário de Apoio.

References

[1] R. Berthilsson, K. Astrom, and A. Heyden, "Projective Reconstruction of 3D-Curves from Its 2D-Images Using Error Models and Bundle Adjustments," *Proc. Scandinavian Conf. Image Analysis,* 1997.

[2] R. Bhatia, "Matrix Analysis," *Graduate Texts in Mathematics,* S. Alex, ed., Springer, 1997.

[3] R. Burkard, E. Çela, and B. Klinz, "On the Biquadratic Assignment Problem," *Proc. DIMACS Workshop Quadratic Assignment and Related Problems,* pp. 117-146, May 1993.

[4] A. Cabot and R. Francis, "Solving Certain Nonconvex Quadratic Minimization Problems by Ranking the Extreme Points," *Operations Research,* vol. 18, no. 1, pp. 82-86, Feb. 1970.

[5] Y. Cheng, V. Wu, R. Collins, A. Hanson, and E. Riseman, "Maximum-Weight Bipartite Matching Technique and Its Application in Image Feature Matching," *Proc. SPIE Conf. Visual Comm. and Image Processing,* 1996.

[6] G. Fielding and M. Kam, "Weighted Matchings for Dense Stereo Correspondence," *Pattern Recognition,* vol. 33, pp. 1511-1524, 2000.

[7] S. Gold, A. Rangarajan, C. Lu, S. Pappu, and E. Mjolsness, "New Algorithms for 2D and 3D Point Matching," *Pattern Recognition,* vol. 31, no. 8, pp. 1019-1031, 1998.

[8] A.V. Goldberg, S.A. Plotkin, and P.M. Vaidya, "Sublinear-Time Parallel Algorithms for Matching and Related Problems," *Proc. IEEE Symp. Foundations of Computer Science,* pp. 174-185, 1988.

[9] R. Horn and C. Johnson, *Matrix Analysis.* Cambridge Univ. Press, 1985.

[10] *Handbook of Global Optimization,* R. Horst and P. Pardalos, eds., Kluwer, 1995.

[11] J. Júdice and A. Faustino, "Solution of the Concave Linear Complementary Problem," *Recent Advances in Global Optimization,* pp. 76-101, 1992.

[12] M. Karwan, V. Lotfi, J. Telgen, and S. Zionts, *Redundancy in Mathematical Programming.* Springer-Verlag, 1983.

[13] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. Seventh Int'l Joint Conf. Artifical Inteligence,* 1981.

[14] J. Maciel, "Global Matching: Optimal Solutions for Correspondence Problems," PhD thesis, Instituto Superior Téecnico 2001. http://www.isr.ist.utl.pt/vislab/thesis/macielPhD.pdf.

[15] J. Maciel and J. Costeira, "Stereo Matching as a Concave Minimization Problem," Technical Report VISLAB-TR 11/99, Instituto de Sistemas e Robótica, IST, 1999.

[16] J. Maciel and J. Costeira, "Robust Point Correspondence by Concave Minimization," *Proc. 11th British Machine Vision Conf.,* pp. 626-635, 2000.

[17] C. Meyer, "A Simple Finite Cone Covering Algorithm for Concave Minimization," Technical Report 193, Karl-Franzens-Universitat Graz, Apr. 2000.

[18] G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization.* Wiley, 1988.

[19] Y. Ohta and T. Kanade, "Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, no. 2, pp. 139-154, Mar. 1985.

[20] P. Pardalos, "Enumerative Techniques for Solving some Nonconvex Global Optimization Problems," Technical Report CS-86-33, Dept. of Computer Science, The Pennsylvania State Univ., Nov. 1986.

[21] P. Pardalos and J. Rosen, "Methods for Global Concave Minimization: A Bibliographic Survey," *SIAM Rev.,* vol. 28, no. 3, pp. 367-379, Sept. 1986.

[22] S. Roy and I. Cox, "A Maximum-Flow Formulation of the n-Camera Stereo Correspondence Problem," *Proc. Int'l Conf. Computer Vision,* 1997.

[23] G. Scott and H. Longuet-Higgins, "An Algorithm for Associating the Features of Two Patterns," *Proc. Royal Soc. London, B,* vol. 244, pp. 21-26, 1991.

[24] L. Shapiro and J. Brady, "Feature-Based Correspondence: An Eigenvector Approach," *Image and Vision Computing,* vol. 10, no. 5, pp. 283-288, June 1992.

[25] C. Silva, "3D Motion and Dense Structure Estimation: Representations for Visual Perception and the Interpretation of Occlusions," PhD thesis, Instituto Superior Técnico, 2001. http://www.isr.ist.utl.pt/vislab/thesis/CesarPhD.pdf.

[26] J. Starink and E. Backer, "Finding Point Correspondences Using Simulated Annealing," *Pattern Recognition,* vol. 28, no. 2, pp. 231-240, 1995.

[27] C. Tomasi and T. Kanade, "Shape from Motion from Image Streams under Orthography: A Factorization Method," *Int'l J. Computer Vision,* vol. 9, no. 2, pp. 137-154, Nov. 1992.

[28] P. Torr, "Motion Segmentation and Outlier Detection," PhD thesis, Dept. of Eng. Science, Univ. Oxford, 1995.

[29] M. Wu and J. Leou, "A Bipartite Matching Approach to Feature Correspondence in Stereo Vision," *Pattern Recognition Letters,* vol. 16, pp. 23-31, Jan. 1995.

[30] Z. Zhang, "Iterative Point Matching for Registration of Free-Form Curves," Technical Report RR-1658, INRIA, 1992.

João Maciel received the Licenciatura degree and the PhD degree in electrical engineering from Instituto Superior Técnico, Lisbon, Portugal in 1997 and 2001, respectively. He was with the Computer Vision group at Instituto de Sistemas e Robótica (Lisbon Pole) from 1996 to 2002. He was awarded IBM (Portugal) Science Prize. Currently, he is with Reverse Engineering Corp., a startup company of which he is a cofounder.

João P. Costeira received the Licenciatura degree, MS degree, and PhD degree in electrical and computer engineering from Instituto Superior Técnico (IST), Lisbon, Portugal in 1985, 1989, and 1995, respectively. He was a visiting scientist at the Robotics Institute, Carnegie Mellon University between 1992 and 1995, where he developed work in the area of multiple motion segmentation. Since 1995, he has been an assistant professor at IST and a researcher at Instituto de Sistemas e Robótica working in the area of computer vision.