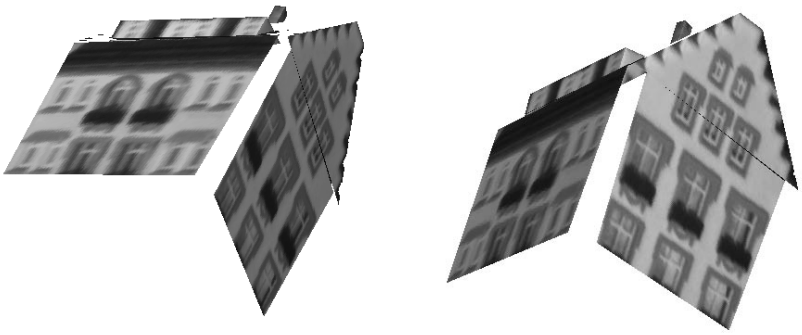
UNIVERSIDADE TÉCNICA DE LISBOA

INSTITUTO SUPERIOR TÉCNICO



Global Matching:
optimal solution to correspondence problems

João Lourenço Teixeira Lopes de Sousa Maciel

(Licenciado)

Dissertação para obtenção do Grau de Doutor em

Engenharia Electrotécnica e de Computadores

Orientador:		Doutor João Paulo Salgado Arriscado Costeira
Júri:	Presidente:	Reitor da Universidade Técnica de Lisboa
	Vogais:	Doutor José Manuel Fonseca de Moura
		Doutor Takeo Kanade
		Doutor João José dos Santos Sentieiro
		Doutor José Alberto Rosado dos Santos Victor
		Doutor João Paulo Salgado Arriscado Costeira
		Doutor José Manuel Bioucas Dias
		Doutor Pedro Manuel Quintas Aguiar

Agosto de 2001

Agradecimentos

Apesar de todos os compromissos a que tem que obedecer, esta tese reflecte a minha presente maneira de ser; por isso agradeço em primeiro lugar às pessoas que, pelos seus ensinamentos, interferiram no processo da minha evolução.

Pouco tempo depois de eu nascer, o meu irmão ensinou-me que a vida é difícil. No mesmo dia, os meus pais mostraram-me que, apesar disso, há sempre gente à nossa volta que fará tudo para nos ajudar, mesmo quando o não percebemos. Durante toda a minha vida tentei pôr esta tese à prova, e sempre pude constatar a sua veracidade. Já em idade adulta aprendi a lição mais elementar de todas e a que, provavelmente, mais fortemente influenciou a minha maneira de ser. A Ana ensinou-me que o racionalismo mais bem ordenado pode sucumbir perante uma ligeira brisa de emotividade.

Esta tese reflecte também a maneira de ser colectiva do ISR. A maior qualidade do ISR é o facto de reunir pessoas que gozam de plena realização profissional. Acredito que isso se deve à contagiante liberdade com que o Prof. João Sentieiro lidera. Espero ter conseguido assimilar uma fracção da sua inegável capacidade para formar equipas e motivá-las.

Agradeço a todos os elementos do ISR com os quais interagi, em especial aos do VisLab. Destaco o César, que foi a fonte necessária de entropia em todo o processo. Tenho a impressão que ainda vamos ter muitas oportunidades para gerar entropia em conjunto; pelo menos assim o espero. Apesar das mudanças que tem estado a sofrer, estou certo que o VisLab manterá o dinamismo incutido pelo incansável José Alberto, que me tem servido de modelo pela sua metódica organização.

Agradeço à Fundação para a Ciência e Tecnologia, que financiou o meu Doutoramento, e à Fundação Luso-Americana para o Desenvolvimento, que financiou parte da minha estadia em Pittsburgh em 1999. Agradeço ainda ao Prof. Fonseca de Moura, pela forma como me acolheu na CMU. Ensinou-me que a capacidade de trabalho pode não ter limites.

Finalmente, certamente não estaria a escrever esta página se o João Paulo não me tivesse dado essa oportunidade há quatro anos. O João Paulo é a pessoa que conheço com maior sentido de altruismo profissional. Diariamente abdica do tempo que for necessário para ajudar os seus alunos, sem deixar de lhes dar toda a liberdade e de neles confiar. Desde o primeiro momento que me sinto profundamente atraído pela sua sincera paixão pela procura e a descoberta. Mais do que um orientador, considero-o o meu verdadeiro mentor.

Lisboa, Julho de 2001.

Resumo

Nesta tese propõe-se uma nova metodologia para resolver o problema de correspondência em sequências de imagens. Este é um problema chave em Visão por Computador e, até à data, não existe nenhum método genérico para o resolver. A metodologia proposta suporta a maioria dos critérios de correspondência, usando um único formalismo. As tarefas de correspondência e eliminação de pontos espúrios são realizadas num único passo, e a solução óptima — global — é encontrada em tempo útil.

As tarefas de selecção de características e sua correspondência são formuladas como um problema de optimização inteira. O custo é transformado numa função côncava e o domínio do problema é relaxado para um conjunto convexo. Este novo problema é equivalente ao primeiro e a sua estrutura especial permite que sejam aplicados algoritmos eficientes que garantem a optimalidade da solução.

Esta formulação permite que sejam utilizados critérios globais, que utilizam todos os pontos e a totalidade do conjunto de possíveis correspondências entre eles, em vez de considerarem pares de pontos isoladamente. Nesta tese alguns critérios globais são apresentados sob a forma explícita de polinómios que servem como função de custo na nossa metodologia.

Um dos critérios em questão consiste na *rigidez*. A rigidez traduz o critério principal utilizado na reconstrução 3D por triangulação, mas raramente é utilizada como critério de emparelhamento; geralmente a correspondência e a reconstrução são encaradas como tarefas distintas. Nesta tese a rigidez é utilizada em conjunto com modelo de câmara ortográfico escalado, em sequências de imagens não calibradas. As correspondências são calculadas por forma a optimizar o mesmo critério que é utilizado no cálculo de forma e movimento pelo método de factorização.

Palavras-chave: Visão por Computador, Problema de Correspondência, Emparelhamento de Características, Reconstrução Tridimensional, Minimização Côncava, Optimização.

Abstract

In this thesis we propose a new methodology to reliably solve the correspondence problem between points on image sequences. 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 formulated as an integer optimization problem. 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 extension assures its equivalence to the original problem, and it can be optimally solved by efficient algorithms that avoid combinatorial search.

This formulation has the advantage of allowing the use of global criteria, which consider the whole set of features and the whole set of possible correspondences, instead of one isolated pair of points at a time. We develop explicit polynomial cost functions for a few global criteria, and use them in our methodology.

One such criterion is rigidity. Rigidity is the main assumption used in 3D reconstruction by triangulation, but it is not often used for correspondence. Shape extraction and point correspondence are treated, usually, as two different computational processes. We consider the rigidity criterion in fully uncalibrated scaled-orthographic image sequences. Correspondences are set such that they optimize the same criterion used to compute shape and motion by the factorization method.

Keywords: Computer Vision, Correspondence Problem, Feature Matching, Structure from Motion, Concave Minimization, Optimization.

Contents

Chapter 1

Introduction

Estimating point correspondences on image sequences 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.

There are three main difficulties involved in this problem. First, there are no general constraints to help reduce ambiguity. Second, it exhibits high complexity due to the huge dimensionality of its 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.

So far, no general method exists to cope with these three difficulties simultaneously. In order to deal with ambiguity, correspondence methods impose domain-specific restrictive assumptions — matching criteria. To reduce problem complexity, most of the existing algorithms establish a similarity measure between feature-pairs in different frames, and assume that matches between different pairs of features are independent of each other. On the other hand, some methods consider the whole set of features and the whole set of possible correspondences, instead of a single pair of points at a time. Though harder to use, these global criteria are much more restrictive, providing the most reliable and ambiguity-free correspondence methods.

One such criterion is rigidity. Rigidity is the main assumption used in 3D reconstruc-

tion by triangulation, but it is not often used in correspondence. The assumptions and models used to match image points are frequently unrelated to those used to estimate their 3D coordinates. Shape extraction and point correspondence are treated, usually, as two different computational processes. On one hand, shape estimation algorithms usually require known correspondences, solving for the unknown shape and motion. On the other hand, feature matching algorithms often disregard the 3D estimation process and require the knowledge of camera parameters or use other assumptions. Furthermore, while matching algorithms tend to rely on local information, shape computation algorithms rely on rigidity as a global scene attribute.

1.1 Thesis overview

In this thesis, we propose a new methodology to reliably solve point correspondence problems. Our methodology is generic, in the sense that it is able to handle most of the commonly used assumptions in a unique formulation. Feature selection and correspondence are both formulated as a single integer optimization problem that considers the whole space of possible point selections and correspondences. We find its global solution avoiding combinatorial search without having to impose additional assumptions. We do so by relaxing the discrete search domain into its convex-hull and finding an equivalent concave cost function. The result is a *concave programming* problem that can be optimally solved by existent efficient algorithms.

We use this methodology with a few global matching criteria, devising different correspondence methods. When using the rigidity criterion, correspondences are set in such a way that they optimize the same criterion used to compute shape and motion. In other words, we link shape computation to image feature matching, by choosing point correspondences that maximize a single global criterion — rigidity. This method is an instance of the original correspondence methodology, consequently it takes feature selection and outlier rejection into account, in a compact and integrated way. Deviations from rigidity under scaled-orthography are represented by a polynomial cost function, for which a concave version can be easily computed.

1.2 Previous Work

In this section we review the most important approaches to the correspondence problem. Comprehensive surveys on the correspondence problem can be found in [109, 33, 106].

Computer vision problems dealing with image sequences are modeled either in continuous-time or in discrete-time. In the first case, structure and motion can be computed directly from image intensities — direct approach [68, 92] — or from their space-time derivatives [24, 71]. In the discrete-time case, image sets are taken with wide-baselines. Apart from some exceptions [3], the first processing stage consists of computing disparity fields, or extracting feature points and putting them in correspondence. Throughout this review we will primarily refer to feature-based wide-baseline uncalibrated situations.

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 [70, 83], epipolar constraint [70, 83] — rigidity as a constraint — uniqueness [31], visibility [89] and proximity. Tracking-like algorithms [52, 85, 96] 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 [101, 52] — image similarity assumption. Other usual choices are point proximity [52, 108] or smoothness of disparity fields [95, 70].

Finally, correspondence algorithms differ also on the computational framework used to solve optimization problems. Dynamic programming [70, 10], graph search [83], bipartite graph matching [25, 18], and convex minimization [52] guarantee optimality. Non-optimal approaches include greedy algorithms [107], simulated annealing [91], relaxation [31], alternating optimization and constraint projection [12], randomized search [95, 101] and voting strategies [20].

In most of the existent methods, domain specific algorithms are developed, select-

ing combinations of appropriate assumptions. The most common combination include epipolar, order, smoothness, proximity and uniqueness [52, 44, 40]. In [81], smoothness of motion, proximity and uniqueness are used in an algorithm able to cope with occlusions. The combination of uniqueness constraints with simple geometric or image-based criteria often results on optimization problems for which highly efficient algorithms are available. For example, in [84, 77, 86] correspondence is formulated as a Procrusterian problem, that can be solved with an SVD decomposition. In [31, 80] a series of matching problems are formulated using uniqueness as a major constraint, considering spurious features. The resulting nonlinear problem is solved with the softassign algorithm [30].

Prior knowledge can also be used to simplify the solution to the correspondence problem, or even turn it trivial. In [98, 99] the knowledge if calibration is used to transform images into a phase-space domain. In [90] a similar representation is used, in which reconstruction and interpolation of views become trivial tasks, even with occlusions.

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 [101] presents a pruning mechanism that performs outlier rejection in sets of previously matched features. Likewise, in [111] classical techniques are used to compute an initial set of matches, which are then used to estimate the epipolar geometry applying a *least median squares* technique while rejecting the outliers.

Rigidity constraint

One of the most useful criteria for correspondence is rigidity. In most situations, objects can be seen as rigid and static. Ultimately any object is rigid when observed at a single time instant. Rigidity has been used before in the correspondence framework [12, 95, 18, 83, 19, 11], though usually in calibrated scenes or in conjunction with other assumptions. In [19], points are observed by a calibrated trinocular system, and distances between projection rays are minimized. The problem is formulated as a maximum-weight bipartite graph matching problem, and solved by an efficient algorithm. In [61] rigidity is used as a constraint to remove optical flow ambiguities.

In [11, 12] rigidity is used to solve the correspondence problem between several projections of a curve in 3D space. The solution results from iterating between shape computations and curve parameterization. In [95], preservation of the affine structure is checked for different correspondences of sets of five points in three images.

The approach in [22] is an example where matching and 3D reconstruction are deeply related. Correspondences, shape and motion are simultaneously optimized by an Expectation Maximization algorithm. Spurious features are not explicitly taken into account.

The use of rigidity constraint and prior knowledge about camera geometry reduce the complexity of matching. Examples are epipolar constraints [110] and multi-view motion constraints [88]. Other approaches use a minimal set of correspondences which help in computing the correspondences for the complete set of features [87]. In [60], a Kalman filter tracks an initial set of points, using an affine structure model. The authors claim that the procedure is robust to occluded and spurious points.

Most of the work using rigidity for correspondence deals with orthographic, scaled-orthographic or paraperspective [2] models. In [8], a closed form solution to a metric that penalizes affine deformations under weak-perspective is used in a image-to-model alignment scheme. A series of algebraic relations expressing rigidity under paraperspective are devised in [7], though no practical matching algorithm is presented. In [51], rigidity and proximity are used to match points under scaled-orthography.

Factorization method

In this thesis, the use of the rigidity criterion — Chapter 3 — is intimately related to the factorization method for structure from motion. Next, we present a short overview of the evolution of this method. A complete survey can be found in [43].

The factorization method was first introduced by Tomasi and Kanade [97]. Its popularity is due to the soundness of its formulation, its robustness and computational simplicity. An observation matrix is factored in an orthographic projection matrix and a shape matrix, using SVD decomposition and solving a simple overconstrained linear system. All the points and all the frames are used simultaneously.

In [1], the observation matrix is modified to accommodate confidence weights for feature trajectories. The computational cost is also reduced because shape and motion are computed by factorization of a different matrix, which is rank-1 without noise. Directional uncertainty is modeled in [41], where the covariance-weighted error of the observation matrix approximation is minimized. Further statistical study of the method is reported in [66]. In [45] different noise models are introduced, and optimal solutions are devised for each one of them.

The method was also extended to handle feature types other than points [65, 79]. In [64] a recursive formulation of the method is developed. Shape is refined as the observations are taken. A recursive formulation that is able to handle occlusions and inclusion of new features is described in [14]. In [21], the factorization framework is used to segment multiple objects moving independently. It uses rigidity as a single criterion, so transparency is considered. The factorization method was also extended to different projection models, namely paraperspective [78, 32] and perspective [94].

1.3 Original Contributions

In this thesis a new methodology to reliably solve point correspondence problems is proposed. The methodology combines the following advantages:

- It is able to handle the usual assumptions in a unique formulation.
- It considers the whole space of possible point selections and correspondences, so outliers are rejected in a compact and integrated way.
- The global optimal solution is found with feasible computation.

The validation of the methodology comprises the proof of integrality of special sets of partial permutation matrices. Under the framework of the methodology, we propose global optimal solutions to several correspondence problems, namely:

- Matching of uncalibrated scaled-orthographic image sequences: correspondences are set such that they minimize the residual of the factorization method.

- Optimal block matching: the sum of image-block correlations is globally maximized; a uniqueness constraint is imposed and spurious points are rejected.
- Matching of a calibrated trinocular perspective system: correspondences are set such that all epipolar constraints are consistent.
- Registration of 2D and 3D point clouds: correspondences are set such that a linear transformation between point sets exists; outliers are rejected.

Most of the contributions reported in this thesis were first published in [56, 57, 58, 59].

1.4 Structure of the Thesis

The thesis is organized as follows:

Chapter 2: correspondence is formulated as a concave programming problem. The required mathematical constructs are first introduced. The methodology is then outlined and, finally, important details are described.

Chapter 3: the rigidity criterion is expressed as an explicit polynomial function, that can be used in the methodology presented in Chapter 2. The link with structure from motion estimation is made, by proving that the factorization method is chained in the optimization problem that is solved. Finally, some practical extensions are introduced in the method.

Chapter 4: the problem of concave function minimization under linear constraints is discussed. A comprehensive bibliographic review is presented. Implementation of three different algorithms is briefly described.

Chapter 5: the methodology is used with rigidity and other criteria. Experiments are described and results discussed.

Chapter 6: the most important features and limitations of the methodology are stated. Some extensions are presented as avenues for future work.

1.5 Notation

$\otimes$	Kronecker product
$\odot$	Hadamard (or Schur) element-wise product
$\mathbf{0}_{[m\times n]}$	$m \times n$ matrix of zeros
$\mathbf{1}_{[m\times n]}$	$m \times n$ matrix of ones
α^f	Scale factor for the scaled-orthographic camera in frame f
α_k	Weights of a convex combination
ϵ	Absolute bound on the entries of noise matrices
ϵ_i	Concavization coefficients of J_ϵ
$\mathbf{\Lambda}$	Matrix of the camera scale factors
λ_4	4-th (smallest) singular value of $\mathbf{W}$
$\mathbf{\Pi}^\perp$	Projection operator onto the null space of $\mathbf{CX}_1$
$\mathbf{\Sigma}$	Matrix of singular values of $\mathbf{W}$
$\mathbf{a}$	Component of $\mathbf{z}$ orthogonal to the row-space of $\mathbf{S}_0$
$\mathbf{A}$	Matrix of the linear constraints
$\mathbf{A}_i,\ \mathbf{b}_i$	Submatrces of $\mathbf{A}$ and $\mathbf{b}$
$\mathbf{A}^f$	Block f of a block-diagonal matrix $\mathbf{A}$
$\mathbf{b}$	Right-hand side of the constraints
$\mathbf{B}$	Row-wise matrix of variable compactation
$\mathbf{B}_i$	Matrices of the quadratic forms that make J_{rig}
$\mathcal{B}$	Unit hypercube in $\mathbb{R}^{p_1 p_2}$
$\mathbf{C}$	Centering matrix
$\mathcal{C}$	A compact convex constraining set
$\mathrm{ds_s}$	Short for *doubly substochastic*
$\mathcal{DS}_\mathrm{s}(p_1, p_2)$	Set of $p_1 \times p_2$ doubly substochastic matrices
$\mathbf{E}^\mathbf{X},\ \mathbf{E}^\mathbf{Y}$	Matrices of additive noise terms of $\mathbf{X}',\ \mathbf{Y}'$
$f,\ g$	Frame number indices
F	Number of frames in the image sequence
$\mathbf{H}$	Hessian of function J
$i,\ j,\ k$	Generic indices

$\mathbf{i}^f, \mathbf{j}^f$	Orthonormal coordinate system for image f, in world coordinates
I	Set of indices of a row selection of A
$I_1,\ I_2$	Partition of I
$\mathbf{I}_{[n]}$	n-dimensional identity matrix
J	Scalar cost function
J_ϵ	Concave equivalent to J
J_{rig}	Polynomial rigidity cost function
J_{seq}	Multi-frame rigidity cost function
$\mathbf{J}$	Matrix of the quadratic part of cost function J
$\mathbf{L}, \mathbf{N}$	Generic matrices
m	Number of constraints of a problem
$\mathbf{m}_3^\top$	Last row of the motion matrix
$\mathbf{M}$	Motion matrix (columns are $\mathbf{i}^f, \mathbf{j}^f$). Also a generic matrix
$\mathbf{M}_0$	First 2 rows of the motion matrix
$\hat{\mathbf{M}}_\mathbf{P}$	Estimate of $\mathbf{M}$ up to a linear transformation of the columns
n	Dimensionality of a problem. Number of ones on a support matrix $\mathbf{S}$
N	Dimensionality of feature representations
$\mathcal{O}_N^{rig}$	Set of the best N solutions to the problem under rigidity criterion
p	Index for feature number
$p_1, p_2, \ldots$	Number of features on the successive images
p_t	Rank of correspondences (number of points put in correspondence)
$\mathrm{p_p}$	Short for *partial permutation*
$\mathbf{P}$	A partial permutation or doubly substochastic matrix variable
$\mathbf{P}^*$	The optimal (exact) solution of the correspondence problem
$\mathbb{P}$	Multi-frame variable. Collection of $\mathbf{P}_f$ matrices
$\mathcal{P}_\mathrm{p}(p_1, p_2)$	Set of $p_1 \times p_2$ partial permutation matrices
$\mathcal{P}_\mathrm{p}^\mathrm{c}(p_1, p_2)$	Set of $p_1 \times p_2$ column-wise partial permutation matrices
$\mathcal{P}_\mathrm{p}^{p_\mathrm{t}}(p_1, p_2)$	Set of $p_1 \times p_2$ partial permutation matrices of rank p_t
$\mathbf{q}$	Vectorized version of variable $\mathbf{P}$
$\mathbf{q}^\mathrm{c}$	Compact version of $\mathbf{q}$

$\mathbf{Q}$	Linear transformation of $\hat{\mathbf{S}}$ and $\hat{\mathbf{M}}$, imposed by metric constraints. Also a shape interaction matrix
$\tilde{\mathbf{R}}$	Observation matrix on the rank-1 formulation
$\tilde{\mathbf{R}}_{\mathbf{P}}$	Observation matrix on the rank-1 formulation, as a function of $\mathbf{P}$
$\mathbf{s}_p$	3D coordinates of the p-th point (p-th row of $\mathbf{S}$)
$\mathrm{s_s}$	Short for *substochastic*
$\mathbf{S}$	Shape matrix (3D coordinates). Also a support matrix
$\mathbf{S}_0$	First two columns of the shape matrix
$\hat{\mathbf{S}}_{\mathbf{P}}$	Estimate of $\mathbf{S}$ up to a linear transformation of the columns
$\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}(p_1, p_2)$	Set of $p_1 \times p_2$ column-wise substochastic matrices
$\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}(p_1, p_2)$	Set of $p_1 \times p_2$ doubly substochastic matrices of rank p_{t}
$\mathbf{t}^f$	Camera translation in frame f
u_p^f, v_p^f	Row/collumn coordinates of the p-th point on the f-th frame
$\mathbf{u'}_f,\ \mathbf{v'}_f$	Vector of row/column coordinates of noise-affected points on frame f
$\mathbf{U},\ \mathbf{V}$	Matrix of left/right singular vectors of $\mathbf{W}$
vec()	Vectorization operator
$\mathbf{W}$	Noise-free matrix of centered observations
$\mathbf{W'}$	Noise-affected matrix of centered observations
$\mathbf{W}_{\mathbf{P}},\ \mathbf{W'}_{\mathbf{P}}$	Observation matrices, as a function of $\mathbf{P}$
$\mathbf{W}^*$	Observation matrix, with the optimal (correct) correspondence
$x_{ij},\ y_{ij}$	j-th coordinate of the i-th feature of the first/second image
x_{ij}^f	j-th coordinate of the i-th feature of the f-th image
$\mathbf{X},\ \mathbf{Y}$	Collection of noise-free observations from the first/second image
$\mathbf{X'},\ \mathbf{Y'}$	Collection of noise-affected observations from the first/second image
$\mathbf{X}_f$	Collection of noise-free observations from the f-th image
$\mathbf{X'}_f$	Collection of noise-affected observations from the f-th image
$\mathbf{z}$	Relative depths (last column of the shape matrix)

Chapter 2

Correspondence as an optimization problem

In this chapter, we formulate correspondence as an optimization problem. This formulation is generic, in the sense that it can handle most of the commonly used assumptions in a unique formulation. Both problems of feature selection and correspondence are formulated as a 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. As shown in Chapters 3 and 5, this formulation has the advantage of allowing the use of global criteria, such as rigidity.

Section 2.1 introduces the problem formulation and its relaxation. The full methodology is outlined in Section 2.2 and the implementation details are presented in the subsequent sections. For the sake of simplicity, we start with the two-image case. Extension to sequences is discussed in Section 2.7.

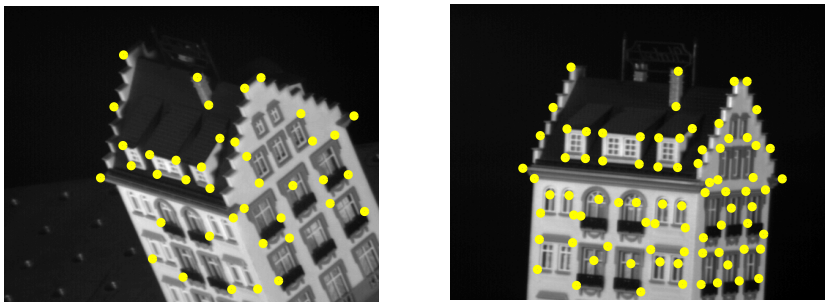


Figure 2.1: Two images from the *Hotel* sequence, with extracted corners.

2.1 Problem formulation

Consider the images of a static scene shown in Figure 2.1[1]. Segment p_1 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

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,N} \\ \vdots & & \vdots \\ x_{p_1,1} & \cdots & x_{p_1,N} \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_{1,1} & \cdots & y_{1,N} \\ \vdots & & \vdots \\ y_{p_2,1} & \cdots & y_{p_2,N} \end{bmatrix} \tag{2.1}$$

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.

Using the previous definitions we formulate the correspondence problem as the integer minimization Problem 1 where J is a scalar objective function.

Problem 1

$$\begin{aligned} \mathbf{P}^* \quad = \quad \arg\min_{\mathbf{P}} \quad & J(\mathbf{X}, \mathbf{Y}, \mathbf{P}) \\ \text{s.t.} \quad & \mathbf{P} \in \mathcal{P}_p(p_1, p_2) \end{aligned}$$

$\mathbf{P}$ is constrained to $\mathcal{P}_{\mathrm{p}}(p_1, p_2)$, the set of $p_1 \times p_2$ *partial permutation matrices* ($\mathrm{p_p}$-matrices). A $\mathrm{p_p}$-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 to correspondence with the rows of $\mathbf{X}$. Each entry $\mathbf{P}_{i,j}$ when set to 1 indicates that

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

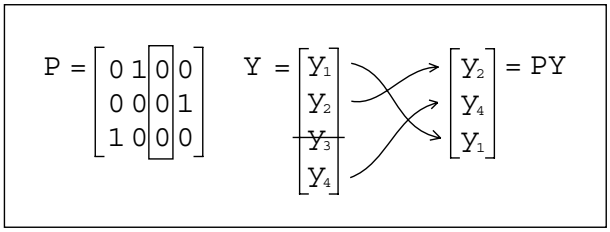


Figure 2.2: A partial permutation matrix representing a particular selection and permutation of rows of $\mathbf{Y}$.

features $\mathbf{X}_{i\cdot}$ (row i of $\mathbf{X}$) and $\mathbf{Y}_{j\cdot}$ (row j of $\mathbf{Y}$) are put to correspondence. Figure 2.2 shows an example.

To guarantee robustness in the presence of outliers, $\mathbf{P}$ must allow some features not to be corresponded, so it cannot be a simple permutation. $\mathrm{p_p}$-matrices represent, at most, one correspondence for each feature, and allow some features not to be matched. If row $\mathbf{P}_{i\cdot}$ is a row of zeros then feature $\mathbf{X}_{i\cdot}$ is not matched. If column $\mathbf{P}_{\cdot j}$ is a column of zeros then feature $\mathbf{Y}_{j\cdot}$ is not matched.

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

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

$$\mathbf{P}_{i,j} \in \{0,1\}\ , \quad \forall i = 1 \dots p_1\ , \quad \forall j = 1 \dots p_2 \tag{2.2}$$

$$\textstyle\sum_{i=1}^{p_1} \mathbf{P}_{i,j} \leq 1\ , \quad \forall j = 1 \dots p_2 \tag{2.3}$$

$$\textstyle\sum_{j=1}^{p_2} \mathbf{P}_{i,j} \leq 1\ , \quad \forall i = 1 \dots p_1 \tag{2.4}$$

To avoid the trivial solution $\mathbf{P}^* = \mathbf{0}$, we establish a fixed number of correspondences $p_{\mathrm{t}} \leq \min(p_1, p_2)$ by considering a slightly different set of matrices $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}(p_1, p_2)$. We call these *rank-$p_t$ partial permutation matrices* (rank-p_{t} $\mathrm{p_p}$-matrices). Constraining the optimization problem to $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}$ leads to a process of picking up just the best p_{t} correspondences. Like in most robust methods [101], p_{t} should be kept near the minimum number of features required by the assumed model or lower than the estimated number of inliers.

The case with $p_{\mathrm{t}} = p_2 \geq p_1$ 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 $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}(p_1, p_2)$, the set of *column-wise partial permutation matrices* (column-wise $\mathrm{p_p}$-matrices). Definitions and properties of $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}$ and $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}$ can be found in Appendix A.1.

2.1.1 Reformulation with a compact convex domain

Problem 1 is a brute force integer minimization problem. In general, there is no efficient way to optimally solve such type of problems. 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

$$\begin{aligned} \mathbf{P}^* \;=\; \arg\min_{\mathbf{P}} \quad & J_\epsilon(\mathbf{X}, \mathbf{PY}) \\ \text{s.t.} \quad & \mathbf{P} \in \mathcal{DS}_s(p_1, p_2) \end{aligned}$$

where J_ϵ is a concave version of J, defined later — equation (2.17). $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$ is the set of *doubly substochastic matrices*, defined by conditions (2.3), (2.4) and:

$$\mathbf{P}_{i,j} \geq 0 \;,\; \forall i = 1 \ldots p_1 \;,\; \forall j = 1 \ldots p_2 \tag{2.5}$$

Problems 1 and 2 can be shown to be equivalent[2] — Section 2.3. The latter belongs to the class of *concave programming* (CP) problems, which is one of the best studied classes of problems in global optimization — Chapter 4. This new formulation has the advantage that several efficient and practical algorithms are available for its resolution. These 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 $\mathrm{p_p}$-matrix, with integer (0-1) entries. The same sort of relaxation can be made to $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}$ and $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}$ — see Appendix A.1.

[2] Two optimization problems are equivalent when they have the same global solution

2.2 Outline of the methodology

Section 2.1 described how to formulate a generic correspondence problem as a concave minimization problem. Matching criteria can be any, as long as features are represented by equal-length vectors. If cost functions are class C^2 — continuous second derivatives — then a concave equivalent can always 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. Furthermore, 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. The whole process is outlined as follows:

1. Extract points of interest and build $\mathbf{X}$, $\mathbf{Y}$ — equation (2.1).

2. Use $\mathbf{X}$, $\mathbf{Y}$ to build a cost function $J(\mathbf{P})$.

3. Build the concave equivalent $J_\epsilon(\mathbf{P})$ — Section 2.5.

4. Write $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$ in canonical form — Section 2.4.

5. Eventually add extra constraints — Section 2.6.

6. Solve Problem 3 using a CP algorithm — Chapter 4.

The remaining sections of this chapter present the details. Section 2.3 presents the fundamental proof of the equivalence between Problems 1 and 2. Section 2.4 presents the canonical form of the relaxed constraints. Section 2.5 describes how the concave equivalent $J_\epsilon(\mathbf{P})$ can be found. Sections 2.6 and 2.7 describe extensions where extra constraints and sequences of images are included.

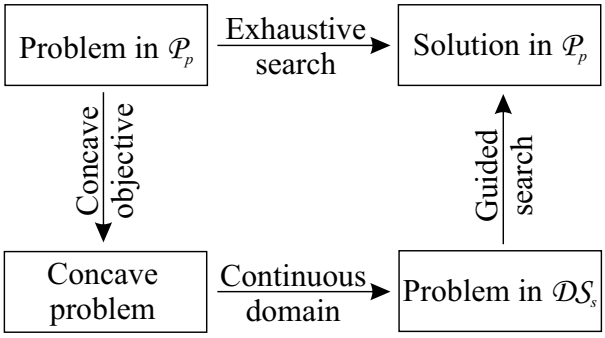


Figure 2.3: Efficient solution to the combinatorial problem.

2.3 Equivalence of Problems 1 and 2

Theorem 1 states the fundamental reason for the equivalence of Problems 1 and 2. In [37] its proof if given.

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

The constraining set of a minimization problem with 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 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.

Finally, 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 $J_\epsilon(\mathbf{P})$. Figure 2.3 summarizes the whole process. It remains valid even 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 [53, 69, 26]. This is particularly useful in problems with constraining integral polytopes, because the relaxation is exact [38, 37, 9].

This means that the optimal solution of the relaxed problem is still integer, so there is no need to project the solution on the neglected 0-1 constraints — *e.g.* rounding.

Exact relaxation is usually treated 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 [35, 36, 13]. It is the convex-hull of the set of *permutation matrices*.

2.4 Constraints in canonical form

Most concave and linear programming algorithms assume that problems have their constraints in canonical form. We will now express the canonical form of the constraints that define $\mathcal{DS}_{\mathrm{s}}$, in the canonical form, that is, we will re-state Problem 2 as

Problem 3

$$\begin{aligned} \mathbf{P}^* \;=\; \arg\min_{\mathbf{P}} \quad & J_\epsilon(\mathbf{X}, \mathbf{PY}) \\ \text{s.t.} \quad & \mathbf{Aq} \leq \mathbf{b}\ ,\ \mathbf{q} \geq \mathbf{0} \\ & \mathbf{q} = \mathit{vec}(\mathbf{P}) \end{aligned}$$

$\mathbf{A}_{[m\times n]}$ and $\mathbf{b}_{[m\times 1]}$ define the intersection of m left half-planes in $\mathbb{R}^n$. 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. Condition (2.3) can now be written as

$$\mathbf{P}.\mathbf{1}_{[p_2\times 1]} \leq \mathbf{1}_{[p_1\times 1]} \tag{2.6}$$

Applying the vec() operator [54] to both sides of this inequality we obtain

$$\left(\mathbf{1}^\top_{[1\times p_2]} \otimes \mathbf{I}_{[p_1]}\right) \mathbf{q} \leq \mathbf{1}_{[p_1\times 1]} \tag{2.7}$$

where $\otimes$ is the Kronecker product, and condition (2.3) becomes equivalent to

$$\mathbf{A}_1\mathbf{q} \leq \mathbf{b}_1 \tag{2.8}$$

with

$$\mathbf{A}_1 \;=\; \mathbf{1}^\top_{[1\times p_2]} \otimes \mathbf{I}_{[p_1]} \quad , \quad \mathbf{b}_1 \;=\; \mathbf{1}_{[p_1\times 1]} \tag{2.9}$$

We can write similar inequalities for condition (2.4) and conditions (A.1) and (A.2) of Appendix A.1, which are used in the definitions of $\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}$ and $\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}$. First define

$$\mathbf{A}_2 = -\mathbf{1}_{[1\times p_2]}^{\top} \otimes \mathbf{I}_{[p_1]} \quad , \quad \mathbf{b}_2 = -\mathbf{1}_{[p_1\times 1]} \tag{2.10}$$

$$\mathbf{A}_3 = \mathbf{I}_{[p_2]} \otimes \mathbf{1}_{[1\times p_1]}^{\top} \quad , \quad \mathbf{b}_3 = \mathbf{1}_{[p_2\times 1]} \tag{2.11}$$

$$\mathbf{A}_4 = \mathbf{1}_{[1\times p_1 p_2]}^{\top} \quad , \quad \mathbf{b}_4 = p_t \tag{2.12}$$

$$\mathbf{A}_5 = -\mathbf{1}_{[1\times p_1 p_2]}^{\top} \quad , \quad \mathbf{b}_5 = -p_t \tag{2.13}$$

The following equations are the canonical constraints that define our sets of interest

$$\mathbf{P} \in \mathcal{DS}_{\mathrm{s}}(p_1,p_2) \quad \Leftrightarrow \quad q_i \in \mathbb{R}_0^+ \ , \ \forall i \quad \wedge \quad \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_3 \end{bmatrix} \mathbf{q} \ \leq \ \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_3 \end{bmatrix} \tag{2.14}$$

$$\mathbf{P} \in \mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}(p_1,p_2) \quad \Leftrightarrow \quad q_i \in \mathbb{R}_0^+ \ , \ \forall i \quad \wedge \quad \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_3 \\ \mathbf{A}_4 \\ \mathbf{A}_5 \end{bmatrix} \mathbf{q} \ \leq \ \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \\ \mathbf{b}_5 \end{bmatrix} \tag{2.15}$$

$$\mathbf{P} \in \mathcal{S}_{\mathrm{s}}^{\mathrm{c}}(p_1,p_2) \quad \Leftrightarrow \quad q_i \in \mathbb{R}_0^+ \ , \ \forall i \quad \wedge \quad \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \end{bmatrix} \mathbf{q} \ \leq \ \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix} \tag{2.16}$$

2.5 Concave equivalent to a class C^2 cost function

This section describes a way of finding a concave function $J_\epsilon : \mathcal{DS}_{\mathrm{s}}(p_1,p_2) \to \mathbb{R}$ having the same values as J at every point of $\mathcal{P}_{\mathrm{p}}(p_1,p_2)$. To be precise, we will only guarantee concavity inside $\mathcal{DS}_{\mathrm{s}}(p_1,p_2)$, not over the entire $\mathbb{R}^{p_1 p_2}$, but this is enough to verify the conditions of Theorem 1.

Consider 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

$$J_\epsilon(\mathbf{q}) = J(\mathbf{q}) - \sum_{i=1}^{p_1 p_2} \epsilon_i q_i^2 + \sum_{i=1}^{p_1 p_2} \epsilon_i q_i \tag{2.17}$$

If $\mathbf{P} \in \mathcal{P}_{\mathrm{p}}$, all entries $\mathbf{P}_{ij}$ are either 0 or 1. In this situation, the extra terms in equation (2.17) cancel, so we conclude that $J_\epsilon(\mathbf{q}) = J(\mathbf{q})$ whenever $\mathbf{q} = \mathrm{vec}(\mathbf{P})$ and $\mathbf{P} \in \mathcal{P}_{\mathrm{p}}$.

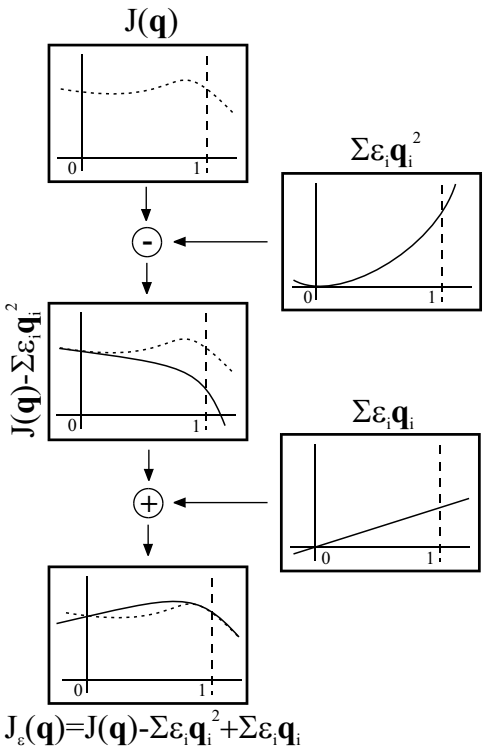


Figure 2.4: Finding $J_\epsilon(\mathbf{q})$, concave in $[0,1]$ and such that $J_\epsilon(\mathbf{q}) = J(\mathbf{q}), \forall q_i \in \{0,1\}$.

On the other hand $\mathcal{P}_{\mathrm{p}}(p_1,p_2)$ is bounded by hypercube $\mathcal{B} = \{\mathbf{q} \in \mathbb{R}^{p_1p_2} : 0 \leq q_i \leq 1,\ \forall i\}$. 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

$$\epsilon_i \geq \frac{1}{2}\left[\max_{\mathbf{q}}\left(\sum_{j=1,j\neq i}^{p_1p_2}|\mathbf{H}_{ij}(\mathbf{q})|\right) + \max_{\mathbf{q}}\left(\mathbf{H}_{ii}(\mathbf{q})\right)\right] \tag{2.18}$$

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

$$\frac{\partial^2 J_\epsilon(q)}{\partial q_i^2} < -\sum_{j=1,j\neq i}^{p_1p_2}\left|\frac{\partial^2 J_\epsilon(q)}{\partial q_i \partial q_j}\right| \quad , \quad \forall i \tag{2.19}$$

A strictly diagonally dominant matrix having only negative elements on its diagonal is strictly negative definite [35], so these values of ϵ_i will guarantee that $J_\epsilon(\mathbf{q})$ is concave for $\mathbf{q} \in \mathcal{B}$ and, therefore, also for $\mathbf{q} \in \mathcal{DS}_{\mathrm{s}}(p_1,p_2)$. The same is true for $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}(p_1,p_2)$ and $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}(p_1,p_2)$ sets. Figure 2.4 illustrates this process for a simple 1D example.

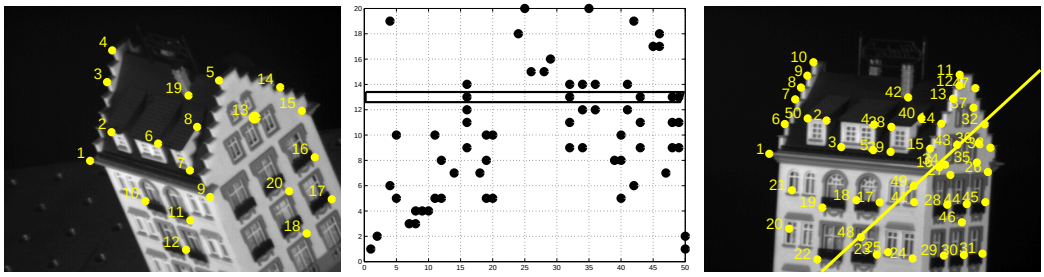


Figure 2.5: An example of a support matrix representing the epipolar constraint

2.6 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.

First express the new constraints by an indicator matrix $\mathbf{S}$ — Figure 2.5. $\mathbf{S}$ 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.

We then *squeeze* our variable vector $\mathbf{q}$, eliminating the entries fixed to 0. Thus we obtain a new variable $\mathbf{q}^{\mathrm{c}}$ of dimension $n = \sum_{i,j} S_{ij}$. This new variable is such that

$$\mathbf{q} = \mathbf{B}\,\mathbf{q}^{\mathrm{c}} \tag{2.20}$$

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 $\mathrm{vec}(\mathbf{S}) = \mathbf{B}.\mathbf{1}^{\top}_{[\mathbf{n}\times\mathbf{1}]}$. Finally, the new constraints are implicitly included in Problem 4 through variable $\mathbf{q}^{\mathrm{c}}$

Problem 4

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

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.

2.7 Handling image sequences

For the sake of simplicity, the descriptions so far were valid for the two-image case only. In this section we extend the methodology for sequences of images.

With F frames, feature-points are extracted and arranged in F matrices

$$\mathbf{X}_f = \begin{bmatrix} x^f_{1,1} & \cdots & x^f_{1,N} \\ \vdots & & \vdots \\ x^f_{p_f,1} & \cdots & x^f_{p_f,N} \end{bmatrix} \quad , \quad f = 1, \ldots, F \tag{2.21}$$

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
$$\begin{aligned} \mathbb{P}^* \;=\; \arg\min_{\mathbb{P}} \quad & J(\mathbf{X}_1, \ldots, \mathbf{X}_F, \mathbb{P}) \\ \text{s.t.} \quad & \mathbf{P}_1, \ldots, \mathbf{P}_{F-1} \in \mathcal{P}_p \end{aligned}$$

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

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{A}^{F-1} \end{bmatrix} \tag{2.22}$$

A block diagonal matrix with TU blocks is also TU — Appendix A.4 — so the relaxation is still exact.

Chapter 3

Maximizing rigidity

This chapter describes a method to solve the correspondence problem between points of a fully uncalibrated scaled-orthographic image sequence. The formulation presented in Chapter 2 is applied, using a single global criterion — rigidity. Among all possible point selections and permutations, the method matches points that maximize rigidity. In other words, correspondences are set such that shape and motion computation is optimal. This way, we link shape computation to image feature correspondence. Furthermore, we show that our criterion is optimal under bounded noise conditions.

The rigidity criterion is inserted in the formulation introduced in Chapter 2, so feature selection and outlier rejection are taken into account in a compact and integrated way.

In Section 3.1 the problem which we propose to solve is precisely stated. Section 3.2 formulates the problem and Section 3.3 outlines the full method. The implementation details are presented in the subsequent sections. For the sake of simplicity, we start with the two-image nondegenerate case. We then extend the method to handle sequences of images — Section 3.6 — and explain how to deal with possible degenerate cases — Section 3.7.

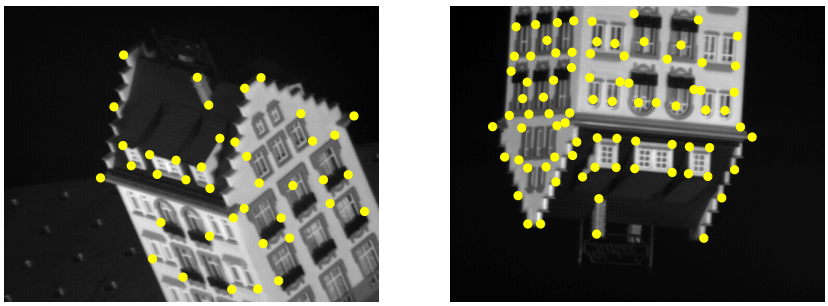


Figure 3.1: Two modified images from the Hotel sequence, with extracted corners.

3.1 Problem statement

Consider the images of Figure 3.1. Segment p_1 points on the first image and $p_2 > p_1$ on the second. Arrange their row and column coordinates u_p and v_p in matrices $\mathbf{X}$ and $\mathbf{Y}$

$$\mathbf{X} = \begin{bmatrix} u_1^1 & v_1^1 \\ \vdots & \vdots \\ u_{p_1}^1 & v_{p_1}^1 \end{bmatrix} \quad , \quad \mathbf{Y} = \begin{bmatrix} u_1^2 & v_1^2 \\ \vdots & \vdots \\ u_{p_2}^2 & v_{p_2}^2 \end{bmatrix} \tag{3.1}$$

Some of these features are projections of the same 3D points. We wish to recover their 3D coordinates assuming no prior knowledge except that the object is rigid and the camera is scaled-orthographic. To do so we apply the *factorization method* [97] to the observations in $\mathbf{X}$ and $\mathbf{Y}$. A selection mechanism must first arrange some of the observed features in a matrix of measurements $\mathbf{W}$. Corresponding features must lie in the same row of $\mathbf{W}$. Note that we make no local image assumptions, no calibration information is known and no constraints on disparity are used.

Without noise $\mathbf{W}$ is, at most, rank-3, even when scale changes. We propose to search for the correspondences that best generate a rank-three $\mathbf{W}$ matrix. In the presence of noise, $\mathbf{W}$ is always full-rank, so we must be able to answer the following questions:

1. Is it possible to generalize the rank criterion in the presence of noise?

2. Can we find the best solution to this problem within reasonable time?

In this chapter we give a positive answer to these questions, by formulating the correspondence problem as an optimization problem with polynomial cost function.

3.2 Optimal matching

Our goal is to determine a matrix $\mathbf{P}^* \in \mathcal{P}_{\mathrm{p}}^{\mathrm{c}}(p_1, p_2)$, such that $\mathbf{X}$ and $\mathbf{P}^*\mathbf{Y}$ have the corresponding features in the same rows. $\mathcal{P}_{\mathrm{p}}^{\mathrm{c}}$ is the set of column-wise partial permutation matrices ($\mathrm{p_p}$-matrices) — see Appendix A. Such a constraint guarantees robustness in the presence of outliers by allowing some features no to be corresponded — Chapter 2.

Each $\mathrm{p_p}$-matrix $\mathbf{P}$ encodes one way of selecting and grouping the measurements in the following matrix of centered observations

$$\mathbf{W_P} \;=\; \mathbf{W}(\mathbf{P}) \;=\; [\mathbf{CX} \mid \mathbf{CPY}]_{[p_1 \times 4]} \tag{3.2}$$

The observations are normalized to zero mean (centroid) with

$$\mathbf{C}_{[p_1 \times p_1]} = \mathbf{I} - \frac{1}{p_1}\mathbf{1}_{[p_1 \times p_1]} \tag{3.3}$$

where $\mathbf{1}_{[p_1 \times p_1]}$ is a square matrix of ones. The optimal correspondence $\mathbf{P}^*$ generates the observation matrix

$$\mathbf{W}^* = \mathbf{W}(\mathbf{P}^*) \tag{3.4}$$

which is the transpose of the usual measurement matrix of the factorization method [97]. This way of grouping the data is particularly useful because $\mathbf{W}^*$ satisfies a simple rank property, known in the literature as the *Rank Theorem*. Theorem 2 is a re-statement of this result for the case of scaled-orthographic cameras under the framework of the correspondence problem. Its proof is presented in Appendix B.

Theorem 2 *A scaled-orthographic camera taking noise-free measurements from non-degenerate — full 3D — objects, provides rank-3 observation matrices* $\mathbf{W}^* = \mathbf{W}(\mathbf{P}^*)$ *whenever* $\mathbf{P}^*$ *is the correct partial permutation.*

A single mismatched point generates a row of $\mathbf{W_P}$ outside of the 3-dimensional row-space of $\mathbf{W}^*$. $\mathbf{W_P}$ will be full-rank even in the absence of noise. This property is exploited in the noise-free correspondence problem formulated as the following optimization problem:

Problem 6

$$\begin{aligned} \mathbf{P}^* &= \arg\min_{\mathbf{P}} \quad rank(\mathbf{W_P}) \\ &\quad \text{s.t.} \quad \mathbf{P} \in \mathcal{P}_p^c(p_1, p_2) \end{aligned}$$

The correct correspondence is a solution to Problem 6. However, Theorem 2 is an implication only, so more than one solution may exist. For example, an object deforming with a 3D linear transformation generates rank-3 $\mathbf{W}^*$ matrices. Permutation of points on the same epipolar line do not change the rank of $\mathbf{W}$ with two images. Appendix C describes degenerate cases in detail. Sections 3.6 and 3.7 describe ways of handling these cases. For now, assume that Problem 6 has a single nondegenerate solution.

3.2.1 Approximate rank

Consider now the case of noisy measurements. The observations in equation (3.1) are now arranged as

$$\mathbf{X}' = \mathbf{X} + \mathbf{E}^X \tag{3.5}$$

$$\mathbf{Y}' = \mathbf{Y} + \mathbf{E}^Y \tag{3.6}$$

where $\mathbf{E}^X$ and $\mathbf{E}^Y$ are additive noise terms. The observation matrix becomes

$$\mathbf{W'_P} = \left[\mathbf{CX}' \mid \mathbf{CPY}'\right] \tag{3.7}$$

Even when $\mathbf{P}$ is the correct correspondence, $\mathbf{W'_P}$ is full rank, because noise affects its largest singular value. The factorization method [97] provides an efficient way of dealing with noise, finding the best rigid interpretation of noisy observations. It finds a shape matrix $\mathbf{S}$ and an orthographic projection matrix $\mathbf{M}$ that solve[1].

$$\min_{\mathbf{M},\mathbf{S}} \left\|\mathbf{W'_P} - \mathbf{SM}\right\|^2 \tag{3.8}$$

To solve this problem, the factorization method finds the rank-3 matrix that best approximates $\mathbf{W'_P}$. In the 2-image case, this is done by neglecting the smallest singular value of the following partition of the SVD decomposition of $\mathbf{W'_P}$

$$\mathbf{W'_P} = \mathbf{U\Sigma V}^\top \tag{3.9}$$

$$= \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^\top + \mathbf{U}_2\lambda_4\mathbf{V}_2^\top \tag{3.10}$$

[1] In our notation, $\mathbf{W}$ is the transpose of the observation matrix in [97]

with

$$\mathbf{U} = [\underbrace{\mathbf{U}_1}_{3} \mid \underbrace{\mathbf{U}_2}_{1}] \} p_1 \tag{3.11}$$

$$\mathbf{\Sigma} = \left[\begin{array}{c|c} \mathbf{\Sigma}_1 & \mathbf{0} \\ \hline \mathbf{0} & \lambda_4 \end{array}\right] \begin{array}{l} \} 3 \\ \} 1 \end{array} \tag{3.12}$$

$$\mathbf{V}^\top = \underbrace{\left[\begin{array}{c} \mathbf{V}_1^\top \\ \hline \mathbf{V}_2^\top \end{array}\right]}_{4} \begin{array}{l} \} 3 \\ \} 1 \end{array} \tag{3.13}$$

In the 2-image case, the approximation error of this rank-3 fitting is measured by $\lambda_4(\mathbf{W'_P})$, the smallest singular value of $\mathbf{W'_P}$.

We use $\lambda_4(\mathbf{W'_P})$ as a generalization of the rank criterion of Problem 6. In the presence of noise, the correspondence problem is formulated as follows

Problem 7

$$\begin{aligned} \mathbf{P}^* = \arg\min_{\mathbf{P}} \quad & \lambda_4(\mathbf{W'_P}) \\ \text{s.t.} \quad & \mathbf{P} \in \mathcal{P}_p^c(p_1, p_2) \end{aligned}$$

Solving this problem corresponds to finding the correspondence that minimizes the approximation error of the solution given by the factorization method, that is

$$\arg\min_{\mathbf{P}} \lambda_4(\mathbf{W'_P}) = \arg\min_{\mathbf{P}} \left(\min_{\mathbf{M},\mathbf{S}} \|\mathbf{W'_P} - \mathbf{SM}\|^2 \right) \tag{3.14}$$

Equation (3.14) expresses the link between correspondence and reconstruction. The inner minimization is implicit in λ_4.

In Section 3.5 we show that, for bounded noise, the λ_4 criterion is optimal, in the sense that the solution of Problem 7 is equal to the solution of Problem 6 when no noise is present. In practice, we use yet another related criterion — Section 3.4 — which allows us to obtain an explicit polynomial cost function. It can also be interpreted as a minimization of the approximation error of the factorization method.

3.3 Outline of the method

In Section 3.2 we described how to use the rigidity criterion in the methodology of Chapter 2. An outlier rejection mechanism is directly embedded in the formulation, and prior knowledge can be easily included. Sections 3.4 and 3.5 present the details. Section 3.6 explains how to extend the method to solve sequences of images. Section 3.7 explains how to deal with the degenerate cases described in Appendix C. The whole process is outlined as follows:

1. Extract interest points and build $\mathbf{X}_1 \dots \mathbf{X}_F$ — equation (2.21).

2. Use $\mathbf{X}_1 \dots \mathbf{X}_F$ to build cost function $J_{rig}(\mathbf{P})$ — equation (3.15).

3. Generate $\mathcal{O}_N^{rig}$ of equation (3.41) — steps **3.** to **6.** of Section 2.2.

4. Solve Problem 9 searching in $\mathcal{O}_N^{rig}$ — Section 3.7.

3.4 Explicit polynomial cost function

Chapter 4 accounts for the practical importance of having a cost function in explicit, low-order polynomial form. In chapter 5 we will present polynomial cost functions for a number of different criteria. In this section we devise an explicit polynomial cost function representing the rigidity criterion.

We will now show that Problem 6 has an equivalent explicit biquadratic polynomial cost function like

$$J_{rig}(\mathbf{P}) = \left(\mathbf{q}^\top \mathbf{B}_1 \mathbf{q}\right)\left(\mathbf{q}^\top \mathbf{B}_2 \mathbf{q}\right) - \left(\mathbf{q}^\top \mathbf{B}_3 \mathbf{q}\right)^2 \tag{3.15}$$

where $\mathbf{B}_i$ are matrices independent of $\mathbf{q} = \mathrm{vec}(\mathbf{P})$.

We start by considering yet another description of the rigidity criterion. If we assume bounded noise, we can show that Problem 6 is equivalent to Problem 8

Problem 8

$$\begin{aligned} \mathbf{P}^* \;=\; \arg\min_{\mathbf{P}} \quad & \det\left(\mathbf{W'}_{\mathbf{P}}^\top \mathbf{W'}_{\mathbf{P}}\right) \\ \text{s.t.} \quad & \mathbf{P} \in \mathcal{P}_p^c(p_1, p_2) \end{aligned}$$

In the absence of noise the result is direct. The full proof is in Section 3.5. The criterion of Problem 8 can be transformed into equation (3.15). Start by using the fact that for any two matrices $\mathbf{M}_{[l\times m]}$ and $\mathbf{N}_{[l\times n]}$, if $\mathbf{L}_{[l\times(m+n)]} = [\mathbf{M} \mid \mathbf{N}]$ and $\mathbf{N}$ is full-rank — see [35, 36, 54] — then

$$\det\left(\mathbf{L}^\top\mathbf{L}\right) = \det\left(\mathbf{M}^\top\mathbf{M}\right)\det\left\{\mathbf{M}^\top\left[\mathbf{I}_{[m]} - \mathbf{N}\left(\mathbf{N}^\top\mathbf{N}\right)^{-1}\mathbf{N}^\top\right]\mathbf{M}\right\} \tag{3.16}$$

With $\mathbf{L} = \mathbf{W'_P} = [\mathbf{CX'} \mid \mathbf{CPY'}]$ and since $\mathbf{C}$ is symmetrical and idempotent, then

$$\det\left(\mathbf{W'}_\mathbf{P}^\top\mathbf{W'_P}\right) = \det\left(\mathbf{X'}^\top\mathbf{CX'}\right)\det\left(\mathbf{Y'}^\top\mathbf{P}^\top\mathbf{\Pi}^\perp\mathbf{PY'}\right) \tag{3.17}$$

with $\mathbf{\Pi}^\perp = \mathbf{C} - \mathbf{CX'}\left(\mathbf{X'}^\top\mathbf{CX'}\right)^{-1}\mathbf{X'}^\top\mathbf{C}$. Since the first determinant in equation (3.17) is positive and independent of $\mathbf{P}$, we can simplify the cost function, stating

$$\arg\min_\mathbf{P}\det\left(\mathbf{W'}_\mathbf{P}^\top\mathbf{W'_P}\right) = \arg\min_\mathbf{P}\det\left(\mathbf{Y'}^\top\mathbf{P}^\top\mathbf{\Pi}^\perp\mathbf{PY'}\right) \tag{3.18}$$

Now define $\mathbf{Y'} = [\mathbf{u'}_2 \mid \mathbf{v'}_2]$, where $\mathbf{u'}_2$ and $\mathbf{v'}_2$ are respectively the row and column coordinates of points on the second image. This leads to

$$\det\left(\mathbf{W'}_\mathbf{P}^\top\mathbf{W'_P}\right) = \det\left[\begin{array}{c|c}\mathbf{u'}_2^\top\mathbf{P}^\top\mathbf{\Pi}^\perp\mathbf{Pu'}_2 & \mathbf{u'}_2^\top\mathbf{P}^\top\mathbf{\Pi}^\perp\mathbf{Pv'}_2 \\ \hline \mathbf{v'}_2^\top\mathbf{P}^\top\mathbf{\Pi}^\perp\mathbf{Pu'}_2 & \mathbf{v'}_2^\top\mathbf{P}^\top\mathbf{\Pi}^\perp\mathbf{Pv'}_2\end{array}\right] \tag{3.19}$$

For any two matrices $\mathbf{L}_{[l\times m]}$ and $\mathbf{M}_{[m\times n]}$

$$\mathrm{vec}(\mathbf{LM}) = \left(\mathbf{M}^\top\otimes\mathbf{I}_{[m]}\right)\mathrm{vec}(\mathbf{L}) \tag{3.20}$$

The observations $\mathbf{u'}_2$ and $\mathbf{v'}_2$ are vectors so, using equation (3.20), we obtain

$$\begin{aligned}\mathbf{u'}_2^\top\mathbf{P}^\top\mathbf{\Pi}^\perp\mathbf{Pv'}_2 &= \mathrm{vec}(\mathbf{Pu'}_2)^\top\mathbf{\Pi}^\perp\mathrm{vec}(\mathbf{Pv'}_2) \\ &= \mathbf{q}^\top\left(\mathbf{u'}_2\otimes\mathbf{I}_{[p_2]}\right)\mathbf{\Pi}^\perp\left(\mathbf{v'}_2^\top\otimes\mathbf{I}_{[p_2]}\right)\mathbf{q}\end{aligned} \tag{3.21}$$

There are similar expressions for the other combinations of $\mathbf{u'}$ and $\mathbf{v'}$, so

$$\arg\min_\mathbf{P}\det\left(\mathbf{W'}_\mathbf{P}^\top\mathbf{W'_P}\right) = \arg\min_\mathbf{P}\left[\left(\mathbf{q}^\top\mathbf{B}_1\mathbf{q}\right)\left(\mathbf{q}^\top\mathbf{B}_2\mathbf{q}\right) - \left(\mathbf{q}^\top\mathbf{B}_3\mathbf{q}\right)^2\right] \tag{3.22}$$

with $\mathbf{q} = \mathrm{vec}(\mathbf{P})$ and

$$\mathbf{B}_1 = \left(\mathbf{u'}_2\otimes\mathbf{I}_{[p_2]}\right)\mathbf{\Pi}^\perp\left(\mathbf{u'}_2^\top\otimes\mathbf{I}_{[p_2]}\right) \tag{3.23}$$

$$\mathbf{B}_2 = \left(\mathbf{v'}_2\otimes\mathbf{I}_{[p_2]}\right)\mathbf{\Pi}^\perp\left(\mathbf{v'}_2^\top\otimes\mathbf{I}_{[p_2]}\right) \tag{3.24}$$

$$\mathbf{B}_3 = \left(\mathbf{u'}_2\otimes\mathbf{I}_{[p_2]}\right)\mathbf{\Pi}^\perp\left(\mathbf{v'}_2^\top\otimes\mathbf{I}_{[p_2]}\right) \tag{3.25}$$

3.4.1 Rank-1 factorization

The rank-1 factorization [1], uses a useful different observation matrix $\tilde{\mathbf{R}}$. In this section we show that, when $\mathbf{P} = \mathbf{P}^*$ is the correct correspondence, the term $\mathbf{\Pi}^{\perp}\mathbf{P}\mathbf{Y}'$ of equation (3.17) is exactly the observation matrix $\tilde{\mathbf{R}}$. Each given $\mathbf{P}$ corresponds to a different observation matrix $\tilde{\mathbf{R}}_{\mathbf{P}}$.

In rank-1 formulation, shape and motion — Section 3.2.1 — are decomposed as

$$\mathbf{S} = [\, \mathbf{S}_0 \mid \mathbf{z} \,] \tag{3.26}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_0 \\ \hline \mathbf{m}_3^{\top} \end{bmatrix} \tag{3.27}$$

$\mathbf{S}_0$ and $\mathbf{M}_0$ can be any, as long as they are consistent with the first observation. In particular we use $\mathbf{S}_0 = \mathbf{C}\mathbf{X}'$, so $\mathbf{z}$ are the relative depths.

Using this notation, the optimization problem solved in equation (3.8) becomes

$$\min_{\mathbf{m}_3,\mathbf{a}} \left\| \tilde{\mathbf{R}}_{\mathbf{P}} - \mathbf{a}\mathbf{m}_3^{\top} \right\|^2 \tag{3.28}$$

where $\mathbf{a}$ is the component of $\mathbf{z}$ orthogonal to $\mathbf{S}_0$, and

$$\tilde{\mathbf{R}}_{\mathbf{P}} = \left[\mathbf{I} - \mathbf{S}_0 \left(\mathbf{S}_0^{\top}\mathbf{S}_0 \right)^{-1} \mathbf{S}_0^{\top} \right] \mathbf{W}'_{\mathbf{P}} \tag{3.29}$$

$$= \mathbf{\Pi}^{\perp}\mathbf{P}\mathbf{Y}' \tag{3.30}$$

$\mathbf{\Pi}^{\perp}$ is the orthogonal projector onto the null space of $\mathbf{C}\mathbf{X}'$. Without noise, $\tilde{\mathbf{R}}^* = \tilde{\mathbf{R}}_{\mathbf{P}^*}$ is rank-1. With noise, equation (3.28) is solved by the rank-1 matrix that best approximates $\tilde{\mathbf{R}}_{\mathbf{P}}$. The approximation error of this fitting is measured by the second singular value of $\tilde{\mathbf{R}}_{\mathbf{P}} = \mathbf{\Pi}^{\perp}\mathbf{P}\mathbf{Y}'$, that is

$$\arg\min_{\mathbf{P}} \lambda_2 \left(\mathbf{\Pi}^{\perp}\mathbf{P}\mathbf{Y}' \right) = \arg\min_{\mathbf{P}} \left(\min_{\mathbf{m}_3,\mathbf{a}} \left\| \tilde{\mathbf{R}}_{\mathbf{P}} - \mathbf{a}\mathbf{m}_3^{\top} \right\|^2 \right) \tag{3.31}$$

For bounded noise, this value varies monotonically with $\det\left(\mathbf{Y}'^{\top}\mathbf{P}^{\top}\mathbf{\Pi}^{\perp}\mathbf{P}\mathbf{Y}' \right)$. To prove this we can use the reasoning of Section 3.5. In conclusion, minimizing equation (3.18) corresponds to finding the correspondence that minimizes the approximation error of the solution given by the rank-1 factorization method.

3.5 Optimality of rigidity cost functions

Proposition 1 states that, when Problem 6 is not degenerate, then both $\lambda_4(\mathbf{W'_P})$ and $\det\left(\mathbf{W'_P}^\top \mathbf{W'_P}\right)$ are, in some sense, optimal criteria.

Proposition 1 *When Problem 6 has a single non-degenerate solution, it is possible to find a scalar $\epsilon > 0$ such that if $\left|\mathbf{E}^X_{(i,j)}\right| < \epsilon$ and $\left|\mathbf{E}^Y_{(i,j)}\right| < \epsilon \ \forall\ i,j$ then the solution of Problems 7 and 8 is exactly $\mathbf{P}^*$, the solution of Problem 6 without noise.*

Proof: A unique $\mathbf{P}^*$ is also solution to Problems 7 and 8 without noise, because

$$\operatorname{rank}(\mathbf{W}^*) = 3 \ \Leftrightarrow\ \lambda_4(\mathbf{W}^*) = 0 \ \Leftrightarrow\ \det\left(\mathbf{W}^{*\top}\mathbf{W}^*\right) = 0 \tag{3.32}$$

Non-degeneracy means that there is a nonzero difference between the best and second best cost values of Problems 7 and 8. This is to say that

$$\exists\ \delta_1 > 0 \ : \quad \lambda_4(\mathbf{W}^*) + \delta_1 < \lambda_4(\mathbf{W_P}) \quad , \ \forall\ \mathbf{P} \neq \mathbf{P}^* \tag{3.33}$$

$$\exists\ \delta_2 > 0 \ : \quad \det\left(\mathbf{W}^{*\top}\mathbf{W}^*\right) + \delta_2 < \det\left(\mathbf{W_P}^\top \mathbf{W_P}\right) \quad , \ \forall\ \mathbf{P} \neq \mathbf{P}^* \tag{3.34}$$

$\lambda_4(\mathbf{W'_P})$ and $\det\left(\mathbf{W'_P}^\top \mathbf{W'_P}\right)$ are continuous functions of the entries of $\mathbf{W'_P}$ so they are also continuous functions of the entries of $\mathbf{E}^X$ and $\mathbf{E}^Y$. By definition of continuity, $\exists\ \epsilon > 0$ such that if $|\mathbf{E}^X_{(i,j)}| < \epsilon$ and $|\mathbf{E}^Y_{(i,j)}| < \epsilon\ \forall\ i,j$ then equations (3.33) and (3.34) still hold for $\mathbf{W'}$. This guarantees that, under these noise constraints, $\mathbf{P}^*$ is still the optimal solution to Problems 7 and 8.

This proof for Proposition 1 does not present a constructive way to compute ϵ, so we did an empirical evaluation of the practical validity of the noise bound. Points were segmented on the images of Figure 3.1. This is an inherently noisy process. A set of randomly generated $\mathbf{P}$ matrices, with controlled number of wrong matches, were used to compute $\lambda_4(\mathbf{W'_P})$ and $\det\left(\mathbf{W'_P}^\top \mathbf{W'_P}\right)$. Figure 3.2 shows their statistics. In both cases, the global minimum is reached for the correct correspondence $\mathbf{P}^*$, even with feature location noise. This shows that the bound ϵ is a realistic one.

Finally note that the average values of the criteria increase monotonically with the number of mismatches. This means that suboptimal solutions with objective values

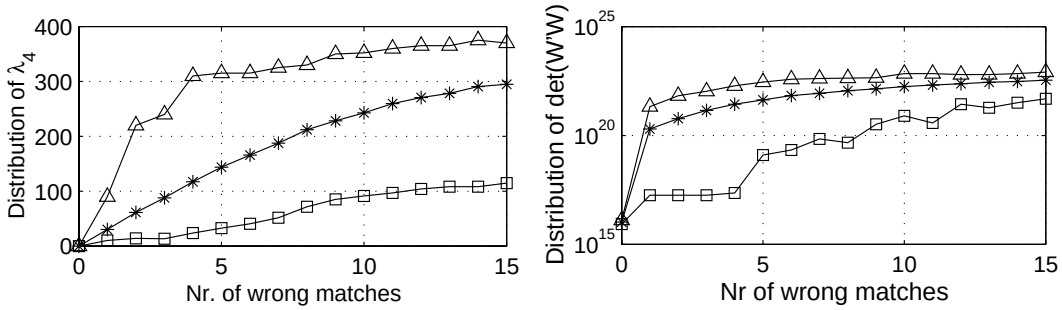


Figure 3.2: Minimum, maximum and average of $\lambda_4(\mathbf{W'_P})$ and $\det\left(\mathbf{W'_P}^\top\mathbf{W'_P}\right)$ as functions of the number of mismatches.

close to optimal will, on average, have a small number of mismatches. This is useful to devise a stopping criterion for our algorithm — Chapter 4. Furthermore, note that the plot for $\det\left(\mathbf{W'_P}^\top\mathbf{W'_P}\right)$ is shown in logarithmic scale. The $\lambda_4 \mapsto \det\left(\mathbf{W'_P}^\top\mathbf{W'_P}\right)$ mapping has the useful effect of making the global minimum *deeper*.

3.6 Multi-frame formulation

In this section we extend the cost function of equation (3.15) to the multi-frame formulation of Section 2.7. With F frames, the original Problem 6 is extended to variable $\mathbb{P} = [\mathbf{P}_1 \mid \cdots \mid \mathbf{P}_{F-1}]$. The new observation matrix is

$$\mathbf{W'}_{\mathbb{P}} = \left[\ \mathbf{C}\mathbf{X'}_1 \mid \mathbf{C}\mathbf{P}_1\mathbf{X'}_2 \mid \cdots \mid \mathbf{C}\mathbf{P}_{F-1}\mathbf{X'}_F\ \right]_{[p_1 \times 2F]} \tag{3.35}$$

The observation matrix of the rank-1 formulation — Section 3.4.1 — is

$$\tilde{\mathbf{R}}_{\mathbb{P}} = \mathbf{\Pi}^\perp\mathbf{C}\left[\ \mathbf{P}_1\mathbf{u'}_2 \mid \mathbf{P}_1\mathbf{v'}_2 \mid \cdots \mid \mathbf{P}_{F-1}\mathbf{u'}_F \mid \mathbf{P}_{F-1}\mathbf{v'}_F\ \right] \tag{3.36}$$

where $\mathbf{u'}_f$ and $\mathbf{v'}_f$ are respectively the row and column coordinates of the points on frame f, and $\mathbf{\Pi}^\perp$ is the orthogonal projector onto the null space of $\mathbf{C}\mathbf{X}_1$. Without noise, and for the correct correspondence, $\operatorname{rank}(\mathbf{W}_{\mathbb{P}^*}) = 3 \Leftrightarrow \operatorname{rank}\left(\tilde{\mathbf{R}}_{\mathbb{P}^*}\right) = 1$. Any selection of two columns of $\tilde{\mathbf{R}}_{\mathbb{P}^*}$ is a rank-1 matrix. We denote each possible pair of columns of $\tilde{\mathbf{R}}_{\mathbb{P}}$ by

$$\tilde{\mathbf{R}}^1_{f,g}(\mathbb{P}) \quad = \quad \mathbf{\Pi}^\perp\mathbf{C}\left[\ \mathbf{P}_{f-1}\mathbf{u'}_f \mid \mathbf{P}_{g-1}\mathbf{u'}_g\ \right] \tag{3.37}$$

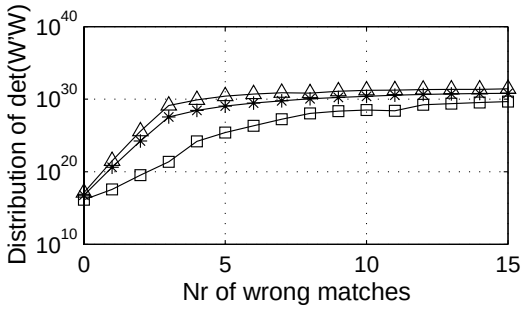


Figure 3.3: Minimum, maximum and average of $J_{seq}(\mathbb{P})$ as functions of the number of mismatches.

$$\tilde{\mathbf{R}}^2_{f,g}(\mathbb{P}) = \mathbf{\Pi}^{\perp}\mathbf{C}\left[\, \mathbf{P}_{f-1}\mathbf{u'}_f \mid \mathbf{P}_{g-1}\mathbf{v'}_g \,\right] \tag{3.38}$$

$$\tilde{\mathbf{R}}^3_{f,g}(\mathbb{P}) = \mathbf{\Pi}^{\perp}\mathbf{C}\left[\, \mathbf{P}_{f-1}\mathbf{v'}_f \mid \mathbf{P}_{g-1}\mathbf{v'}_g \,\right] \tag{3.39}$$

For bounded noise, we can use the arguments of Section 3.5 to show that the criterion $\text{rank}\left(\tilde{\mathbf{R}}_{\mathbb{P}^*}\right) = 1$ is equivalent to the cost function

$$J_{seq}(\mathbb{P}) = \sum_{(i,f,g)\in\mathcal{I}} \det\left(\tilde{\mathbf{R}}^{i\top}_{f,g}(\mathbb{P})\tilde{\mathbf{R}}^{i}_{f,g}(\mathbb{P})\right) \tag{3.40}$$

with $\mathcal{I} = \{(i,f,g) \;:\; i = 1,\ldots,3 \;;\; f,g = 2,\ldots,F\} \setminus \{(1,f,f),(3,f,f) \;:\; f = 2,\ldots,F\}$. $\mathcal{I}$ excludes the cases $\tilde{\mathbf{R}}^1_{f,g}$, $\tilde{\mathbf{R}}^3_{f,g}$ with $f = g$, which are trivially rank-1. Each term of $J_{seq}(\mathbb{P})$ is a biquadratic function like equation (3.15), so the multi-frame rigidity maximization problem has a cost function like equation (3.15) where $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$ are sums of $\left(\frac{3F^2-5F+2}{2}\right)$ terms like those in equations (3.23), (3.24) and (3.25).

Finally, we can experimentally verify that the introduced noise bound is realistic — like in Section 3.5. Points were segmented on three images from the *Hotel* sequence. This is an inherently noisy process. A set of randomly generated $\mathbb{P}$ matrices, with controlled number of wrong matches, were used to compute $J_{seq}(\mathbb{P})$. Figure 3.3 shows their statistics. The global minimum is reached for the correct correspondence $\mathbb{P}^*$, even with feature location noise. This shows that the bound ϵ is a realistic one.

3.7 Using metric constraints to resolve ambiguities

Appendix C describes situations where the solution to Problem 6 is degenerate — multiple global optima. With degeneracy, the demonstration of Section 3.5 is not valid anymore. This means that, with noise, the solution to Problem 8 may not be $\mathbf{P}^*$.

However, $\mathbf{P}^*$ is among the best solutions to Problem 8. We use a procedure that will be described in Section 4.5 to generate a set $\mathcal{O}_N^{rig}$ of the best N solutions, given by

$$\mathcal{O}_N^{rig} = \left\{ \mathbf{P}_k, k = 1 \dots N \ : \ J_{rig}(\mathbf{P}_k) \leq J_{rig}(\mathbf{P}) \ , \ \forall \mathbf{P} \notin \mathcal{O}_N^{rig} \right\} \tag{3.41}$$

We then use a different criterion to choose $\mathbf{P}^*$ among the elements of $\mathcal{O}_N^{rig}$.

When noise is bounded, N does not need to be larger than the number of degenerate solutions to Problem 6. This is precisely stated in Proposition 2.

Proposition 2 *If Problem 6 has M degenerate solutions, then it is possible to find a scalar $\epsilon > 0$ such that if $\left|\mathbf{E}^X_{(i,j)}\right| < \epsilon$ and $\left|\mathbf{E}^Y_{(i,j)}\right| < \epsilon \ \forall \ i, j$ then $\mathbf{P}^* \in \mathcal{O}_N^{rig}$, $\forall \ N \geq M$.*

Proof: Without noise, the solution set $\mathcal{O}_M^{rig}$ of Problem 8 is the same as that of Problem 6, because

$$\text{rank}\left(\mathbf{W}^*\right) = 3 \quad \Leftrightarrow \quad \det\left(\mathbf{W}^{*\top}\mathbf{W}^*\right) = 0 \tag{3.42}$$

Furthermore, there is a nonzero difference between $J_{rig}\left(\mathbf{P}^*\right)$ and the cost value of any non-optimal solution to Problem 8. This is to say that

$$\exists \ \delta > 0 \quad : \quad \det\left(\mathbf{W}^{*\top}\mathbf{W}^*\right) + \delta < \det\left(\mathbf{W}_{\mathbf{P}}^{\top}\mathbf{W}_{\mathbf{P}}\right) \ , \ \forall \ \ \mathbf{P} \notin \mathcal{O}_M^{rig} \tag{3.43}$$

Since $\det\left(\mathbf{W'}_{\mathbf{P}}^{\top}\mathbf{W'}_{\mathbf{P}}\right)$ is a continuous function of the entries of $\mathbf{W'}_{\mathbf{P}}$ then it is also a continuous function of the entries of $\mathbf{E}^X$ and $\mathbf{E}^Y$. By definition of continuity, $\exists \ \epsilon > 0$ such that if $|\mathbf{E}^X_{(i,j)}| < \epsilon$ and $|\mathbf{E}^Y_{(i,j)}| < \epsilon \ \forall \ i, j$ then equation (3.43) still holds for $\mathbf{W'}$. This guarantees that, under these noise constraints, $\mathbf{P}^* \in \mathcal{O}_N^{rig}$, $\forall \ N \geq M$.

Once more, the practical significance of bound ϵ is left for empirical evaluation. The experiments in Chapter 5 show that bound ϵ is realistic. Furthermore N can be choosen small enough to find $\mathbf{P}^*$ by exhaustive search.

The degenerate cases discussed in Appendix C.2 are solved by choosing, among the elements of $\mathcal{O}_N^{rig}$, the one that best agrees with the metric constraints. For each $\mathbf{P} \in \mathcal{O}_N^{rig}$ define

$$\mathbf{W'_P} \approx \hat{\mathbf{S}}_\mathbf{P} \hat{\mathbf{M}}_\mathbf{P} \tag{3.44}$$

estimated using the factorization method. The motion matrix is partitioned in frames

$$\hat{\mathbf{M}}_\mathbf{P} = \left[\hat{\mathbf{M}}_\mathbf{P}^1 \mid \cdots \mid \hat{\mathbf{M}}_\mathbf{P}^F \right] \tag{3.45}$$

The decomposition of equation (3.44) is not unique. Without noise, there is an invertible 3×3 matrix $\mathbf{Q_P}$ that makes

$$\begin{aligned} \hat{\mathbf{M}}_\mathbf{P}^{f\top} \mathbf{Q}_\mathbf{P}^\top \mathbf{Q}_\mathbf{P} \hat{\mathbf{M}}_\mathbf{P}^f &= \mathbf{K}_\mathbf{P}^{f\top} \mathbf{K}_\mathbf{P}^f \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} , \ \forall f = 1 \ldots F \end{aligned} \tag{3.46}$$

With noise, and for $F \geq 3$, an overconstrained system of linear equations can be built to solve for $\mathbf{Q}_\mathbf{P}^\top \mathbf{Q_P}$ in least squares sense. $\mathbf{P}^*$ is the solution with smallest residual in this system. In short, it is the solution to Problem 9

Problem 9

$$\begin{aligned} \mathbf{P}^* = \arg \min_{\mathbf{P}} & \sum_{f=1}^{F} \left\| \mathbf{I}_{[2]} - \mathbf{K}_\mathbf{P}^{f\top} \mathbf{K}_\mathbf{P}^f \right\|^2 \\ \text{s.t.} & \quad \mathbf{P} \in \mathcal{O}_N^{rig} \end{aligned}$$

Choose a small N and solve Problem 9 by exhaustive search. Finally, once $\mathbf{P}^*$ and $\mathbf{Q}_\mathbf{P}^{*\top} \mathbf{Q}_\mathbf{P}^*$ are found, $\mathbf{Q}_\mathbf{P}^*$ is found by SVD

$$\mathbf{Q}_\mathbf{P}^{*\top} \mathbf{Q}_\mathbf{P}^* = \mathbf{U} \mathbf{\Sigma} \mathbf{U}^\top \tag{3.47}$$

$$\mathbf{Q}_\mathbf{P}^* = \mathbf{\Sigma}^{\frac{1}{2}} \mathbf{U}^\top \tag{3.48}$$

Shape and motion are computed by

$$\mathbf{S}^* = \hat{\mathbf{S}}_{\mathbf{P}^*} \mathbf{Q}_{\mathbf{P}^*}^\top \tag{3.49}$$

$$\mathbf{M}^* = \mathbf{Q}_{\mathbf{P}^*} \hat{\mathbf{M}}_{\mathbf{P}^*} \tag{3.50}$$

Chapter 4

Concave programming for $\mathcal{DS}_S$ problems

Concave Programming (CP) problems[1] fall in a broader class of the so called *Global Optimization* problems. The latter are problems with nonconvex cost functions, with or without constraints. Global optimization problems have, in general, several local minima where function values may differ substantially. Usually, there is no local criterion for deciding whether a local solution is global. For this reason, standard (local) nonlinear programming algorithms cannot be applied. The single exception is linear programming. Linear functions are simultaneously concave and convex, so local methods are well suited for linear programming.

In this chapter we deal with the problem of minimizing linearly constrained concave functions. Such problems arise from our formulation of correspondence problems. We start with an overview of the most important methods and algorithms for concave programming. We then describe implementations of three different concave programming algorithms, specialized to handle large-scale, sparse, Totally Unimodular constraints. Finally we explain how the algorithms can be used to generate a list of solutions with close to optimal cost values.

[1] Also called *Concave Minimization* problems

4.1 Overview of CP methods

Most global optimization problems belong to the class of non-polynomial (NP) problems. Nevertheless, there are a few special cases for which it is possible to apply efficient algorithms. Among these, concave minimization is the best studied class of problems [74, 75, 72, 4, 38, 73, 76]. It is characterized by having concave cost function and linear constraints. In its most general form, the CP problem is stated as

Problem 10

$$\begin{aligned} \mathbf{q}^* \;=\; \arg\min_{\mathbf{q}} \quad & J(\mathbf{q}) \\ \text{s.t.} \quad & \mathbf{q} \in \mathcal{C} \end{aligned}$$

where $J : \mathbb{R}^n \rightarrow \mathbb{R}$ is concave and $\mathcal{C}$ is a full dimensional polytope of $\mathbb{R}^n$.

During the last three decades, several efficient and practical algorithms have been devised to solve the CP problem. Though NP in worst case [28], their time complexity is some times heuristically bounded by polynomials [27, 37, 100]. Checking for local optimality in a nonconvex problem is NP-hard, with the exception of concave problems, for which local optimality can be verified in polynomial time [28].

Concave programming approaches are primarily classified as being either deterministic, stochastic or suboptimal (approximate). Deterministic algorithms are those that guarantee the global optimal solution in finite time. Suboptimal algorithms are usually more efficient, but do not guarantee optimality. Stochastic algorithms are suboptimal but also return probabilistic measures of confidence for their answers.

4.1.1 Deterministic approaches to CP

Deterministic algorithms for concave minimization [34, 29] usually combine some of the techniques described next.

Cutting-plane and cone-covering

Most of the classic approaches to solve CP were inspired in the work by Tuy [102]. This cutting-plane method starts at a specific vertex of the constraints and defines a new set of constraints — a cone — that contains the feasible region. Values of a bounding

concave extension of the cost function are used to define cuts that successively reduce the constraining cone, until it reduces to a point.

Subsequent cutting-plane methods [48, 62, 23] basically propose new cuts and new ways of dealing with degeneracy. Among existing optimal methods, cutting-plane and cone-covering provide the most efficient algorithms, but are hard to implement.

Extreme-point enumeration

Enumerative techniques [72, 16, 17, 67, 105] rely on the fact that the solution to a CP program lies at a vertex of the constraining polytope. The basic idea is to start from a vertex and rank the nearby vertices to choose where to move next. Global optimality is tested exploiting cost function concavity. The extreme-point ranking approach provides the simplest and most intuitive method of concave programming, but may degenerate to a complete inspection of all vertices.

Branch-and-bound

Branch-and-bound [26, 69] is one of the most popular techniques for global optimization. It is specially efficient to solve mixed-integer problems, where only a small fraction of the variables is integer. Concave programming algorithms can also be solved by branch-and-bound — ultimately, extreme-point enumeration algorithms perform branch-and-bound — though they may also degenerate to a complete inspection of all vertices.

Interior-point methods

Recent work in [63] describes an exact algorithm combining gradient descent and polytope cuts for minimizing a generic concave function with linear constraints. The paper also presents an interior-point method to find a lower bound to a concave problem in polynomial-time. Good lower bounds can be used to improve the performance of enumeration and branch-and-bound algorithms [16].

Finally, the work reported in [28, 104, 103] proposes a global optimization algorithm for the solution to a large class of nonconvex problems. The algorithm solves the original

problem iteratively through a series of primal and relaxed dual subproblems, which provide upper and lower bounds on the global solution. It combines some of the strategies above, and takes advantage of the existence of linear variables.

4.1.2 Stochastic and approximate methods

Recently, special attention has been paid to suboptimal concave minimization algorithms. There has been a large number of applications requiring the solution to large-scale problems, namely in control theory [4] and computational chemistry [29]. Some highly efficient suboptimal algorithms have been reported as being able to produce *good* solutions [39, 93, 82]. The main research issues are low-complexity bound computation, sparse large-scale representations and algorithm parallelization.

[42, 26] describe implementations of Frank and Wolfe, and Keller algorithms and claim good performances on large-scale sparse problems. These methods find a unique local minimum and do not cycle in concave problems. They are known to provide *good* local solutions with small time complexity. They are good candidates for local algorithm of bootstraping procedures. They can also be extended by adding cuts and perturbations.

Karmakar [46] extended his well known interior-point method to solve nonlinear problems. The algorithm finds *good* local optima and tests for global optimality. It solves a succession of problems using gradient descent constrained to an ellipsoid inscribed in the constraining polytope. Each time a local minimum is found, a new cut is added. Finally, simulated annealing [15] is also experiencing growing popularity.

4.2 An extreme point ranking algorithm for CP

In this section we describe an extension of the exact method of [16] for concave minimization. Its implementation is straightforward. Worst case complexity is non-polynomial but, like the *simplex* algorithm, it typically visits only a fraction of the extreme points.

The algorithm chooses a starting vertex and ranks its neighbors according to the

values of a linear underestimator. The best neighbor is used as starting point for the next iteration. Consider the following nonlinear problem

$$\begin{aligned} \mathbf{q}^* &= \arg \min_{\mathbf{q}} \quad J(\mathbf{q}) \\ &\quad s.t. \quad \mathbf{q} \in \mathcal{C} \end{aligned} \tag{4.1}$$

where $\mathcal{C}$ is a compact nonempty convex subset of $\mathbb{R}^n$ and J is concave. Compute a lower bounding linear function $K(\mathbf{q})$ and define the lower bounding problem

$$\begin{aligned} \mathbf{q}_0 &= \arg \min_{\mathbf{q}} \quad K(\mathbf{q}) \\ &\quad s.t. \quad \mathbf{q} \in \mathcal{C} \end{aligned} \tag{4.2}$$

$J_l = K(\mathbf{q}_0)$ is a lower bound on J^*, and $J_u = J(\mathbf{q}_0)$ is an upper bound on J^*. Define $\{\mathbf{q}_k\}$ as the set of all extreme points of $\mathcal{C}$ such that $K(\mathbf{q}_k) \leq J_u$. The solution to the original problem belongs to this set, that is, $\mathbf{q}^* \in \{\mathbf{q}_k\}$. The following algorithm is based on the previous observations and solves the concave problem of equation (4.1).

0. Find $\mathbf{q}_0$, J_l and J_u. $\mathbf{q}_k^*$ is the *current best*.

1. Let $k \leftarrow 0$. Initialize $\mathbf{q}_0^* \leftarrow \mathbf{q}_0$.

2. Let $k \leftarrow k+1$. Let $\mathbf{q}_k \leftarrow$ *next best* of $K(\mathbf{q})$.

3. **a)** If $K(\mathbf{q}_k) > J_u$ then STOP: $\mathbf{q}^* = \mathbf{q}_k^*$.

b) Otherwise $J_l \leftarrow K(\mathbf{q}_k)$.

4. If $J(\mathbf{q}_k) < J_u$ then $J_u \leftarrow J(\mathbf{q}_k)$ and $\mathbf{q}_k^* \leftarrow \mathbf{q}_k$.

5. Goto 2.

As iterations run, the *current best* solution follows an *ever improving* sequence of extreme points of $\mathcal{C}$, and the bounds on J^* become tighter and tighter. Since $\mathcal{C}$ is a finite-dimensional polytope, it has a finite number of extreme points, so the solution is achieved after a finite number of iterations.

4.2.1 Lower bound for quadratic problems

This method is particularly useful for quadratic problems, because tight bounding functions can be computed by linear programming. Consider the following quadratic problem

$$\begin{array}{lll} \min\limits_{\mathbf{q}} & J(\mathbf{q}) \;=\; \mathbf{c}^\top \mathbf{q} + \mathbf{q}^\top \mathbf{J} \mathbf{q} \\ s.t. & \mathbf{q} \in \mathcal{C} \end{array} \tag{4.3}$$

Denote the ith column of $\mathbf{J}$ by $\mathbf{j}_i$ and define u_i (for $i = 1, \ldots, n$) as

$$\begin{array}{rcll} u_i & = & \min\limits_{\mathbf{q}} & \mathbf{j}_i^\top \mathbf{q} \\ & & s.t. & \mathbf{q} \in \mathcal{C} \end{array} \tag{4.4}$$

Since $\mathcal{C}$ is compact and nonempty, then these n linear problems have finite solution. The computed values are used to compute the following bounding problem

$$\begin{array}{lll} \min\limits_{\mathbf{q}} & K(\mathbf{q}) \;=\; \sum\limits_{i=1}^{n} \left(c_i + u_i\right) q_i \\ s.t. & \mathbf{q} \in \mathcal{C} \end{array} \tag{4.5}$$

This is a lower bounding problem because

$$J(\mathbf{q}) = \sum_{i=1}^{n} c_i q_i + \sum_{i=1}^{n} \left(\mathbf{q}^\top \mathbf{j}_i\right) q_i \tag{4.6}$$

4.2.2 Implementation details

Step **2.** can be implemented using existing efficient methods of ranking the extreme points of a linear program [105, 72, 67].

We implemented a *simplex* algorithm specialized for large-scale, sparse, Totally Unimodular constraints. It deals with redundancy and degeneracy using some tests from [47, 50, 112] and a non-rational base perturbation scheme of [49]. Phase I uses an artificial variable free approach [5]. We used this algorithm to solve all linear problems.

The representations and fundamental operators of this algorithm are the basis for our implementation of the extreme point ranking procedure proposed in [67]. This procedure is used in our implementation of the concave minimization algorithm of [16].

We use a threshold on the difference between bounds as stoping criterion. So, the stopping criterion used in Section 4.2 is replaced by

3. a) If $J_u - J_l \leq t$ then STOP: $\mathbf{q}^* = \mathbf{q}_k^*$.

Cost grows fast with the number of mismatches — Section 3.4 — so this suboptimal strategy returns solutions close to optimal (optimal most of the times) and reduces dramatically the number of iterations.

4.3 A greedy algorithm for CP

We implemented a suboptimal greedy algorithm to solve quadratic CP problems. Consider the quadratic problem of equation (4.3). The algorithm starts with a $p_1 \times p_2$ matrix of zeros. It then successively develops partial solutions by activating — setting to 1 — one variable at each iteration. At iteration k, generated matrices are stored in a set $\mathcal{L}^k$. This set contains rank-k partial permutations, built by activating some entries of the elements of $\mathcal{L}^{k-1}$. Setting variable q_i to 1 represents a cost increase of

$$\mathbf{c}_i + \mathbf{J}(i,i) + 2\sum_{j\in\mathcal{J}} \mathbf{J}(i,j) \tag{4.7}$$

where $\mathcal{J}$ is the index set of active variables of the partial solution under consideration. It also represents a potential cost increase of

$$2\sum_{j\in\overline{\mathcal{J}}} \mathbf{J}(i,j) \tag{4.8}$$

where $\overline{\mathcal{J}}$ is the complement of index set $\mathcal{J}$. At iteration k, the algorithm generates N_k solutions that minimize a weighted sum real and potential costs. N_k follows a predefined increasing succession, so that the algorithm explores only a fraction of the tree of all possible partial solutions.

Rigidity cost functions of Chapter 3 have *deep* global minima. This means that the cost value has large gradients in the neighborhood of its global optimum. In this case, suboptimal solutions with cost value close to optimal cannot differ much from the optimal solution. In other words, local algorithms such as this, should work well.

4.4 A deterministic annealing algorithm for CP

In this section we describe an approximate method to solve concave problems with partial permutation constraints. Using the notation of Problem 10 in Section 4.1 with $\mathbf{q} = \text{vec}(\mathbf{P})$, the algorithm consists of the following steps:

0. Initialize with stationary point $\mathbf{q}^* = \mathbf{q}_0 = \arg\ \max_{\mathbf{q}} J(\mathbf{q})$

1. For $k = 1$ to N, **repeat** bootstrap:

1.1 $\mathbf{q}^* \leftarrow \arg\ \min\left(J(\mathbf{q}_k^*)\ ,\ J(\mathbf{q}^*)\right)$.

1.2 Reinitialize $\mathbf{q}_k \leftarrow \mathbf{q}_0 + \mathbf{e}$ and $\beta \leftarrow \beta_0$.

1.3 While $\beta < \beta_f$ do $\beta \leftarrow \beta\beta_r$ and **repeat** *softassign*:

1.3.1 Unconstrained local step: $\mathbf{q}_k \leftarrow \mathbf{q}_k - \delta\ \nabla_J(\mathbf{q}_k)$.

1.3.2 *Softassign* annealing: $\mathbf{q} \leftarrow \exp(\beta\mathbf{q})$.

1.3.3 Greedy elimination:

a) Cancel $p_1 - p_t$ rows of $\mathbf{P}_k$ with least salient entries.

b) Cancel $p_2 - p_t$ cols of $\mathbf{P}_k$ with least salient entries.

1.3.4 Sinkhorn: **repeat** until $\mathbf{q}_k$ converges to $\mathbf{q}_k^*$:

1.3.4.1 $\forall$ nonzero row i do $\mathbf{P}_k(i,j) \leftarrow \frac{\mathbf{P}_k(i,j)}{\sum_{j=1}^{p_2} \mathbf{P}_k(i,j)}, \forall j$.

1.3.4.2 $\forall$ nonzero column j do $\mathbf{P}_k(i,j) \leftarrow \frac{\mathbf{P}_k(i,j)}{\sum_{i=1}^{p_1} \mathbf{P}_k(i,j)}, \forall i$.

The algorithm consists of thee concatenated loops. The outmost loop performs bootstrap with randomly perturbed initializations. It is responsible for keeping record of the best solution found. The innermost loops are a slight modification of the *softassign* algorithm of [30, 31, 80]. *Softassign* is a deterministic annealing algorithm that iterates between local optimization of the relaxed problem — without any constraints — and a constraint

imposing step. We added the elimination step **1.3.3**, so that the solutions are partial-permutations. The saliency measure used in this step was $\max_j \frac{|\mathbf{P}_k(i,j)|}{\sum_{l\neq j}|\mathbf{P}_k(i,l)|}$ for rows, and transposed for columns. The rational behind this elimination is that rows and columns without a salient entry are *far* from subspace polytope borders of norm 1, so they are projected on zero-norm subspace borders.

In our algorithm, deterministic annealing *temperature* β varies between β_0 and β_f with increase ratio β_r. δ is the step size of the unconstrained local optimization and $\mathbf{e}$ is a random perturbation vector.

The innermost loop — Sinkhorn normalization — is proved to converge to a doubly stochastic matrix. Furthermore, using large β_f, the algorithm returns a 0-1 matrix, so the solution is a $\mathrm{p_p}$-matrix.

4.5 Generating solutions with similar cost

The simplest way of globally generating all N best solutions of a concave problem is to solve the problem N times, each time adding a cut that removes the latest best vertex.

In the framework of Section 4.2, this task is simplified if the generated vertex tree is stored. Adding a cut consists on removing the corresponding node and updating neighborhood relations through the tree. Once this is done, the best solution of the new problem is found by inspection.

In the algorithm of Section 4.4 we avoid the unwanted solutions adding penalty terms to the cost function during the unconstrained local minimization step.

Chapter 5

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.

In Section 5.1 we use the 2-frame rigidity criterion of Chapter 3. In Section 5.2, correlation matching is cast into our optimization formulation. In Section 5.3, correlation and rigidity maximization are combined in a multi-frame, fully automated correspondence method. In Section 5.4 we devise a method that finds correspondences consistent with all epipolar constraints of a calibrated trinocular system. In Sections 5.5 and 5.6 we describe methods of performing registration of 2D and 3D point clouds.

The resulting methods are tested with real images. In some cases, points are extracted and put to correspondence, and shape reconstruction is performed. In other cases, correspondences are used for different tasks. In some examples, robustness is compared with benchmark algorithms.

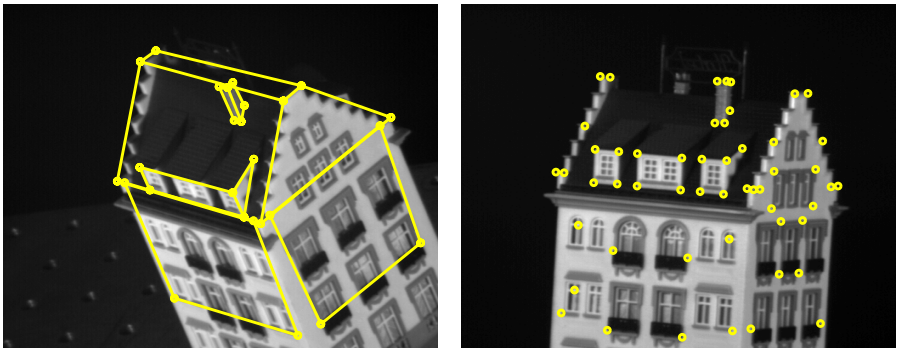


Figure 5.1: Two images from the *Hotel* sequence, with manually segmented points.

5.1 Matching by maximizing 3D rigidity

We implemented a correspondence algorithm that minimizes J_{rig} of equation (3.15), constrained to the set of column-wise partial-permutation matrices. The relaxed problem is solved using the concave programming algorithm described in Section 4.2. Rigidity was the only criterion used. This method is best suited to problems with wide-baseline and little or none *a priori* knowledge about the scene, since no other criterion is used.

Results

Figure 5.1 shows two images of the *Hotel* sequence, having large disparity. No prior knowledge about the scene or the cameras is used, except the fact that the orthographic model is appropriate. No local image assumptions were used. Points were manually selected in both images — 26 points on the first image and 52 on the second. The wireframe is drawn for better perception. The method was applied as described before, using rigidity as the only criterion, and assuming that all the points on the first image had a match on the second. Figure 5.2 shows the reconstruction of the matched points.

Discussion

In this example there were no degenerate situations. The optimal solution is correct — zero wrong matches. The noise bound ϵ is realistic — Section 3.5 — even though feature points were selected manually, with no concern with accuracy.

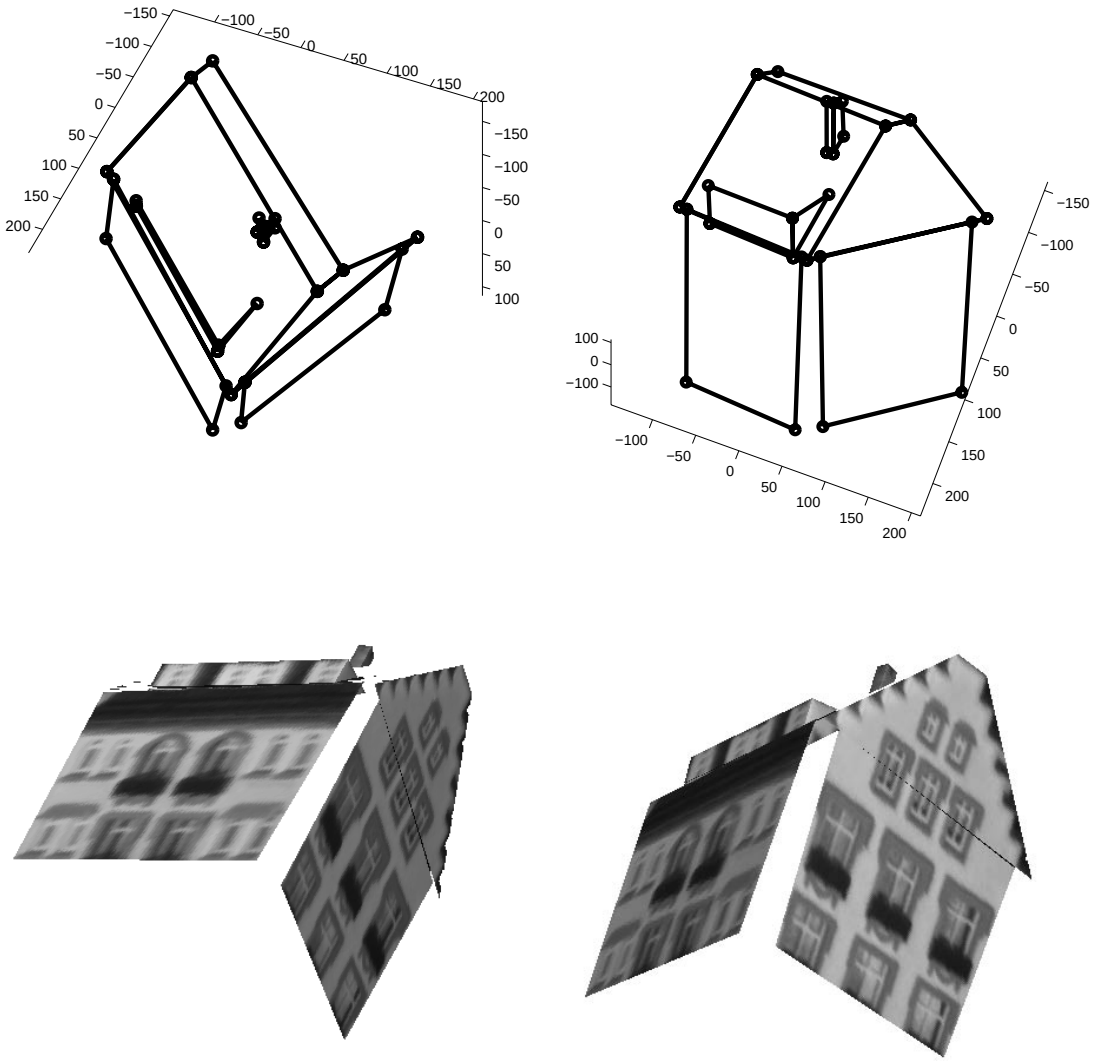


Figure 5.2: Views of a 3D wire-frame and a texture-mapped reconstruction of the Hotel.

This 26 point example required the solution of a biquadratic problem of dimension 1352. The number of feasible solutions is around 10^{70}. The optimal solution was found at iteration 874312, yet the algorithm visited over 10^7 vertices until bounds were closer than a predefined threshold — Section 4.2. The global minimum of $\det\left(\mathbf{W'}^{\top}\mathbf{W'}\right)$ is *deep*, so this stopping is enough to almost guarantee the optimal solution.

The high dimensionality of the resulting problem represented the greatest difficulty. However, this is an extreme situation. With some reasonable assumptions, dimensionality could be reduced drastically. Furthermore, as far as we know, no other method globally optimizes the rigidity criterion.

5.2 Correlation matching

Matching by correlation of image patches is by far 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.

To use this criterion, 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 ith 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

$$J_{corr}(\mathbf{P}) = -\mathrm{tr}\left(\mathbf{P}\hat{\mathbf{Y}}\hat{\mathbf{X}}^{\top}\right) \tag{5.1}$$

Using algebraic properties of the trace operator [54] we get

$$J_{corr}(\mathbf{q}) = -\mathrm{vec}\left(\hat{\mathbf{X}}\hat{\mathbf{Y}}^{\top}\right)^{\top}\mathbf{q} \tag{5.2}$$

which is linear in $\mathbf{q} = \mathrm{vec}(\mathbf{P})$. Problem 3 of Section 2.4 with this linear cost function was

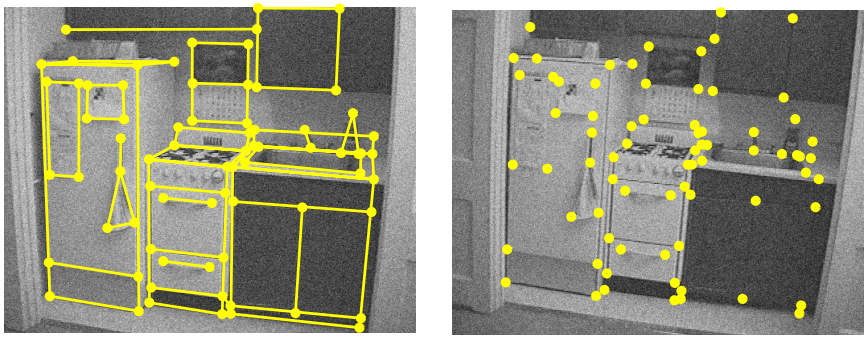


Figure 5.3: Two images — 480×512 pixels, 256 gray levels — from the *Kitchen* sequence, with added noise. Wireframe is for better perception only.

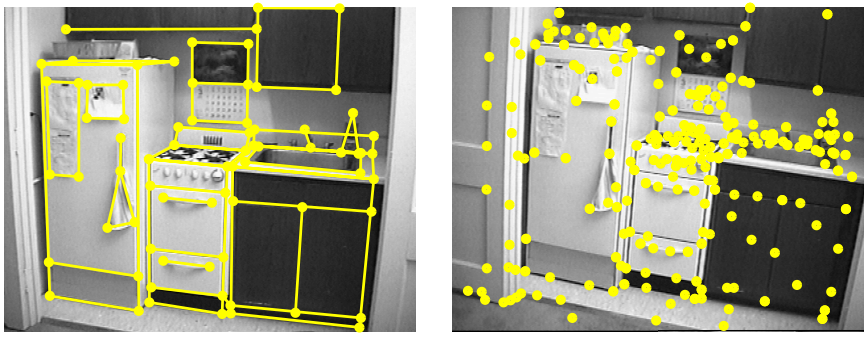


Figure 5.4: Two images — 480×512 pixels, 256 gray levels — from the *Kitchen* sequence, with spurious features. Wireframe is for better perception only.

solved using a *simplex* algorithm. Our implementation takes advantage of the sparse TU structure of the constraints, and deals with degeneracy.

Results

We used the above described method to generate a list of candidate matches. These candidates were pruned for outliers using a random sampling algorithm described in [101], which uses an extra rigidity assumption. This 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 compared the results of this method with those resulting from the application

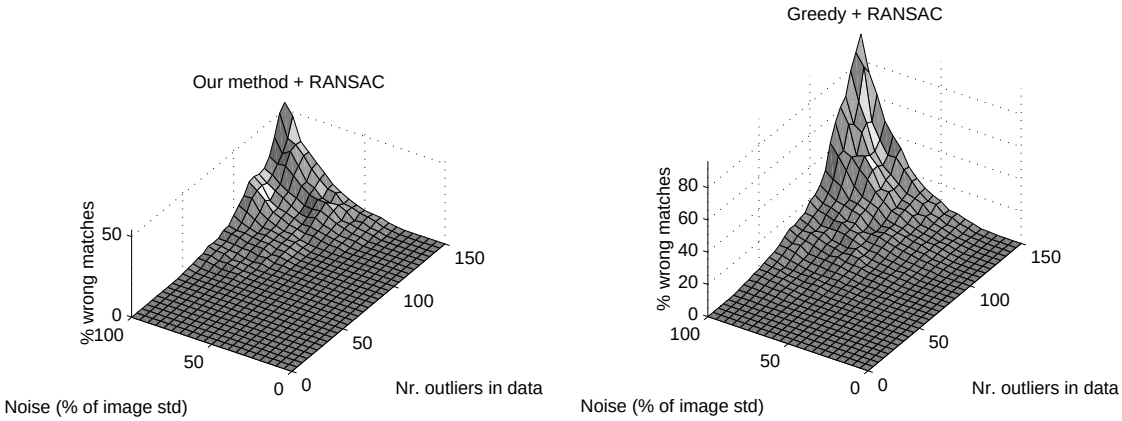


Figure 5.5: Average number of incorrect matches found in 100 trials for increasing levels of data corruption.

of the original procedure of [101], where the initial list of candidate pairs is built by a greedy correlation algorithm.

We selected some image pairs with large disparity from the *Kitchen* sequence[1]. Images were corrupted with zero-mean Gaussian noise, with increasing standard deviation — Figure 5.3. 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 — Figure 5.4. 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 Figures 5.5, 5.6 and 5.7.

Each linear problem was solved in a fraction of a second by a *simplex* algorithm running on a Pentium processor. In the case of 75 features plus 150 outliers, the cardinality of $\mathcal{P}_{\mathrm{p}}(p_1, p_2)$ is roughly 10^{260}. Exhaustive search would be impractical, while the *simplex* algorithm visits less than 500 solutions. A total of 65100 experiments were performed, taking several hours of processing.

[1]Data was provided by the Modeling by Videotaping group in the Robotics Institute, CMU.

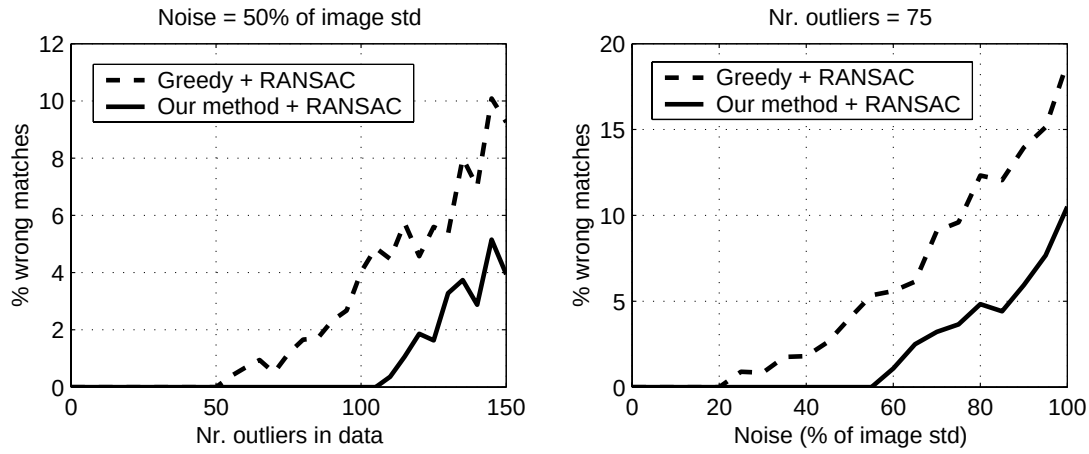


Figure 5.6: Comparison of two profiles from the plots of Figure 5.5.

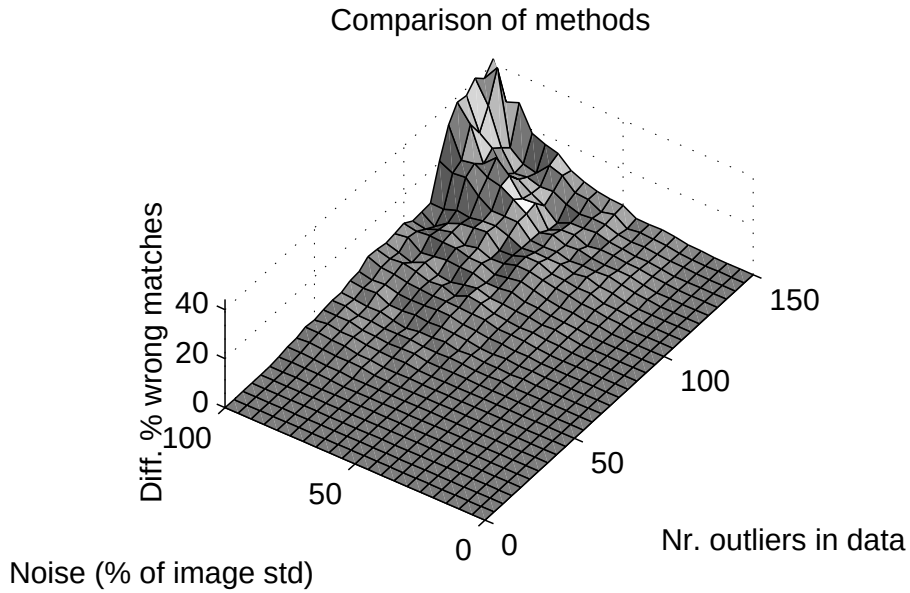


Figure 5.7: Average difference in number of incorrect matches found in two algorithms.

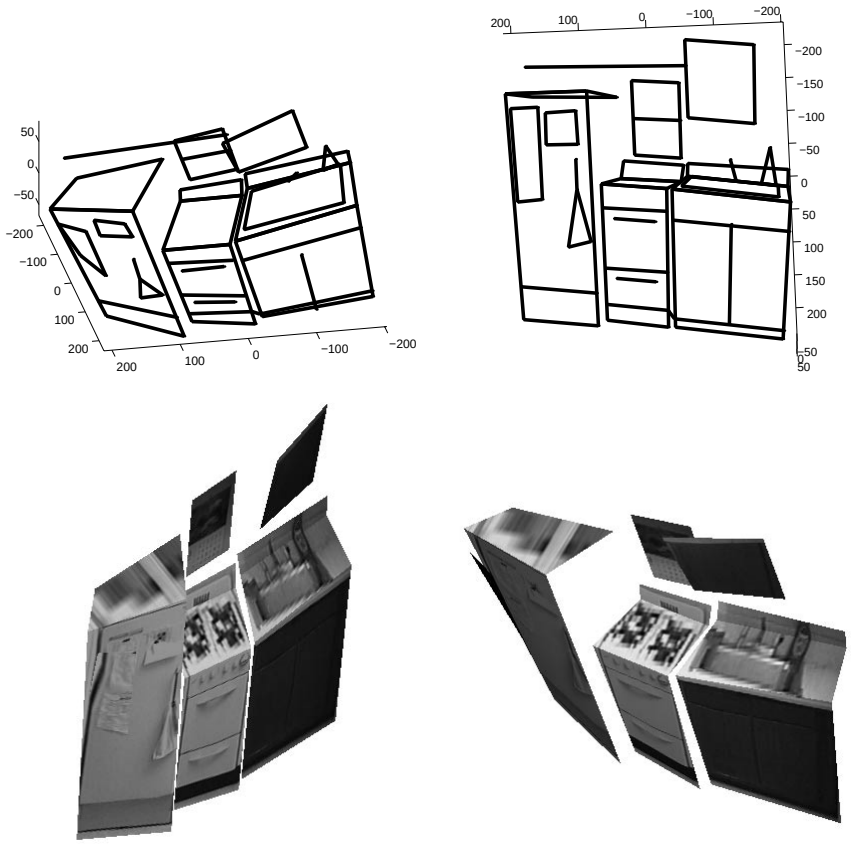


Figure 5.8: Wireframes and texture-mapped VRML reconstructions using 50% of outliers in data, and noise with 50% of images standard deviation.

Discussion

The original method of [101] consistently produced higher number of mismatches, specially when outliers are present. We observed that, when the greedy algorithm returns more than 40% of outliers among the candidates, the validation procedure starts rejecting many good matches. This tends to raise the percentage of wrong matches.

The simultaneous rejection and correspondence of features is a reliable strategy. The reconstruction in Figure 5.8 was obtained in spite of 50% outliers in data, and noise with 50% of the signal standard deviation. With such an amount of image noise, the corner detector returned features with location errors up to 8 pixels. These errors produced a highly distorted reconstruction, but the correspondences were all correct.

5.3 Fully automated reconstruction

We used the methods of Sections 5.1 and 5.2 to produce a fully automated reconstruction. We selected 6 images — fully uncalibrated — in the whole *Hotel* sequence range — Figure 2.1. This image sequence presents two challenges. First, any image pair is related by large disparities, so proximity constraints cannot be used. Second, there are repeated image-patterns, so correlation criterion is highly ambiguous.

We applied a corner detector to smoothed images, extracting 16 points from the first image and 60 points from each one of the subsequent images. We then used the multi-frame formulation of Section 3.6 to compute 16 correspondences across the whole sequence. The nonlinear problem was solved using the algorithm of Section 4.4, which was also used to generate $\mathcal{O}_N^{rig}$ with $N = 100$ — equation (3.41).

We then chose the element of $\mathcal{O}_N^{rig}$ most consistent with the metric constraints (Problem 9 of Section 3.7) recovering motion between each image pair — epipolar constraints. Then, an edge detection algorithm was used to segment 5000 feature-points on each image. The epipolar constraints were used to build support matrices, reducing the dimensionality of the problem. Correspondences between consecutive image pairs were computed using the method of Section 5.2, setting $p_t = 3000$. We then looked for features put to correspondence across all 6 images, and built an observation matrix $\mathbf{W}$ with 1000 observation. 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.

Results

The results of Figure 5.9 were generated without any human intervention or any prior knowledge. We observed that an average of 1.8 wrong matches were computed in the first stage, but the motion estimates were good enough. The full process took around 4 hours of processing. It is hard to find any wrong match among the final 900, because there is no ground truth. Inspection of the images of Figure 5.9 does not reveal any evident spurious point nor major global distortion on the reconstruction.

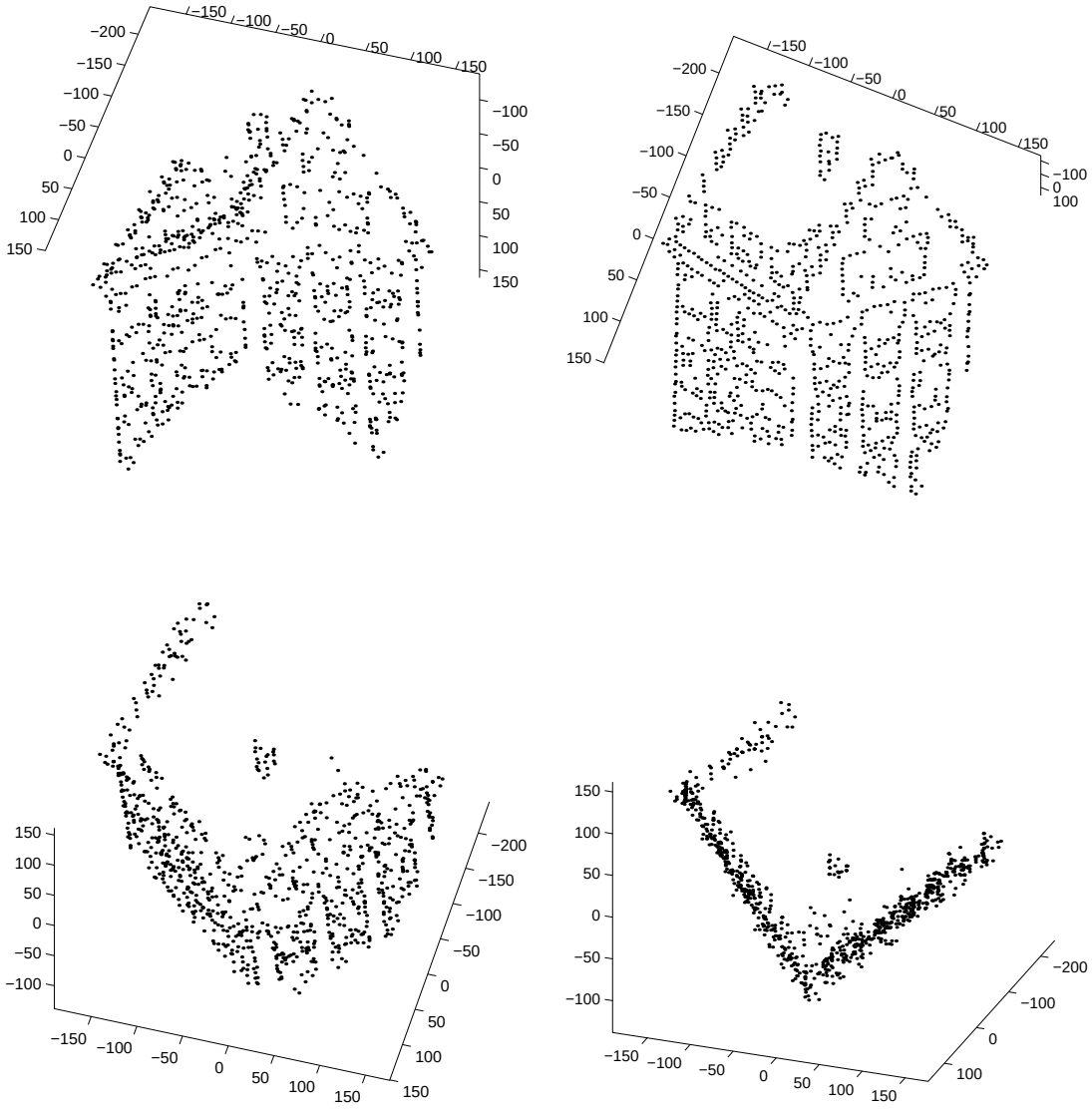


Figure 5.9: Four views of a 3D cloud with 900 points, automatically generated from 6 uncalibrated views.

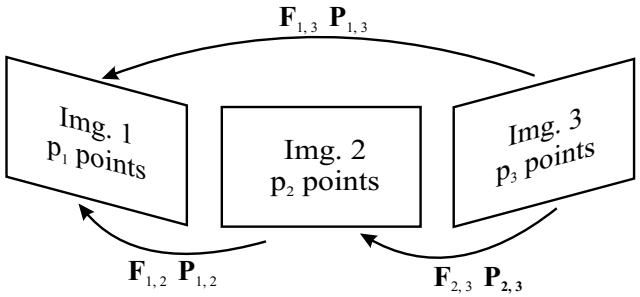


Figure 5.10: Notation for a trinocular system.

5.4 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. Figure 5.10 shows the notation. Each known Fundamental matrix $\mathbf{F}_{k,l}$ defines p_l epipolar lines $\mathcal{L}_{k,l}^m$, $m = 1, \ldots, p_l$ on image k. A point on image k corresponding to the m-th point on image l must lie *close* to $\mathcal{L}_{k,l}^m$. 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 $i = 1, \ldots, p_k$ of image k and the epipolar lines $\mathcal{L}_{k,l}^j$.

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 $\mathbb{P} = \left[\mathbf{P}_{1,2}^\top \mid \mathbf{P}_{2,3}\right]$. We close the loop by estimating the *compound correspondence* $\hat{\mathbf{P}}_{1,3} = \mathbf{P}_{1,2}\mathbf{P}_{2,3}$. The objective function is

$$\begin{aligned} J_{tri} &= \sum_{i=1}^{p_1}\sum_{j=1}^{p_2}\left(\mathbf{P}_{1,2}\odot\mathbf{D}_{1,2} + \mathbf{P}_{2,3}\odot\mathbf{D}_{2,3} + \hat{\mathbf{P}}_{1,3}\odot\mathbf{D}_{1,3}\right) + \\ &\quad + \lambda\left[J_{corr}(\mathbf{P}_{1,2}) + J_{corr}(\mathbf{P}_{2,3}) + J_{corr}(\hat{\mathbf{P}}_{1,3})\right] \end{aligned} \tag{5.3}$$

where $\odot$ is the element-wise product. The addition of correlation terms J_{corr} — Section 5.2 — is used to remove ambiguities. The value of weight λ is chosen experimentally. By algebraic manipulation, we obtain the objective function

$$J_{tri}(\mathbf{q}) = \mathbf{q}^\top\mathbf{J}_{tri}\mathbf{q} + \mathbf{c}_{tri}^\top\mathbf{q} \tag{5.4}$$

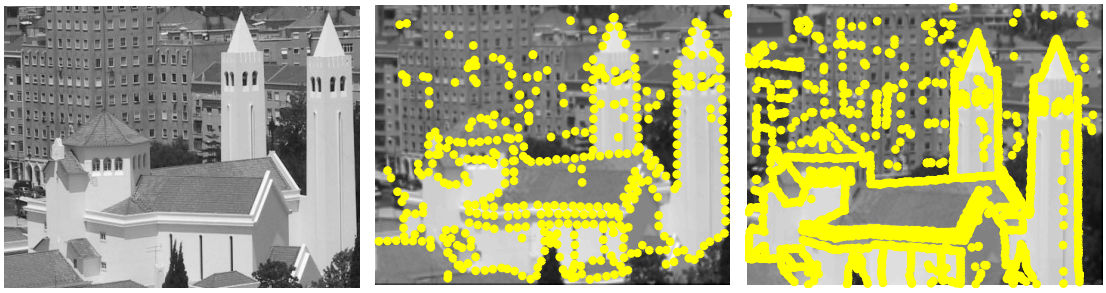


Figure 5.11: *Church* images (384 × 512 pixels) and extracted points.

with $\mathbf{q} = \mathrm{vec}\,(\mathbb{P})$. We reduce the dimensionality of the problem using the support matrices $\mathbf{S}_{k,l}(i,j) = \mathbf{D}_{k,l}(i,j) \leq \delta\ ,\ \forall i,j$.

An entry (i,j) of $\mathbf{S}_{k,l}$ is set to 1 if the i-th point on image k is close to $\mathcal{L}^j_{k,l}$. These constraints are included in the problem as described in Section 2.6. We recover each one of the full variables through $\mathrm{vec}\,(\mathbf{P}_{k,l}) = \mathbf{B}_{k,l}\,\mathbf{p}^c_{k,l}$.

$\mathbf{J}_{tri}$ is, in general, not concave, so a concave version J_ϵ was computed using equations (2.17) and (2.18), before the minimization algorithm was applied.

Results

We applied the described method to the images of Figure 5.11. 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 1500 from the remaining. The second and third images contain, at least, 1000 outliers, so the problem was solved in the presence of more than 65% 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 using the greedy algorithm of Section 4.3. The result was an observation 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.

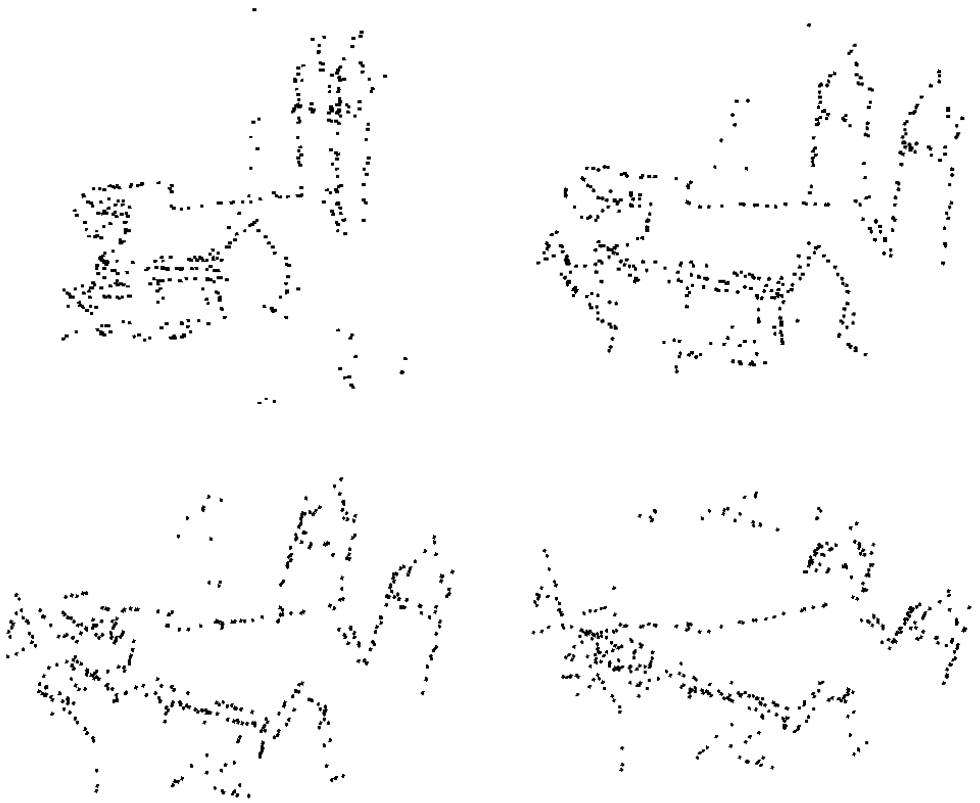


Figure 5.12: Views of a 3D could with 450 points, generated from 3 calibrated views.

This was done to remove existing outliers. Finally, the factorization method was applied to the remaining 450 points. The resulting reconstruction is shown in Figure 5.12.

Discussion

We visually detected around 10% gross mismatches in $\mathbf{P}_{1,2}$ and $\mathbf{P}_{1,3}$. The CP algorithm used is suboptimal, so some of the matches were not consistent with all epipolar constraints. By subspace projection we were able to remove most of these mismatches.

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.

5.5 2D point registration

In this section we develop a method to perform 2D registration of images using an affine 2D model. We propose to search for the correspondences that best fit to a 2D affine transformation of the feature-point set on the first model image. We also describe an application to lip-tracking. Figures 5.13 and 5.14 show an example.

As in Chapter 3, we 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 $\mathbf{CX}$ and $\mathbf{CP}^*\mathbf{Y}$ are best related by a linear transformation. This is cast to the following cascaded optimization problem, where $\mathbf{L}$ is a 2D linear transformation

Problem 11

$$\begin{aligned} \mathbf{P}^* \;=\; \arg\min_{\mathbf{P}} \left(\min_{\mathbf{L}} \|\mathbf{CX} - \mathbf{CPYL}\|^2 \right) \\ \text{s.t.} \quad \mathbf{P} \in \mathcal{P}_p^c(p_1, p_2) \end{aligned}$$

Again, this is equivalent to choose $\mathbf{P}$ such that the observation matrix

$$\mathbf{W_P} = [\mathbf{CX} \mid \mathbf{CPY}] \tag{5.5}$$

is rank-2. This case, however, is easier to solve since $\mathbf{CX}$ spans the whole subspace were the observations must lie. We can *a priori* compute $\mathbf{\Pi}$ — the orthogonal projector on the space of columns of $\mathbf{CX}$ — and use it in the objective function

$$J_{2D}(\mathbf{P}) = \|\mathbf{\Pi CPY}\|^2 \tag{5.6}$$

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

$$\tilde{J}_{2D}(\mathbf{P}) \;=\; |\mathbf{\Pi}_u\mathbf{CPu}| + |\mathbf{\Pi}_u\mathbf{CPv}| + |\mathbf{\Pi}_v\mathbf{CPu}| + |\mathbf{\Pi}_v\mathbf{CPv}| \tag{5.7}$$

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 solve a linear problem with objective function

$$\tilde{J}^i_{2D}(\mathbf{P}) \;=\; \pm(\mathbf{\Pi}_u\mathbf{CPu}) \pm (\mathbf{\Pi}_u\mathbf{CPv}) \pm (\mathbf{\Pi}_v\mathbf{CPu}) \pm (\mathbf{\Pi}_v\mathbf{CPv}) \quad , \quad i = 1, \ldots, 16 \tag{5.8}$$

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

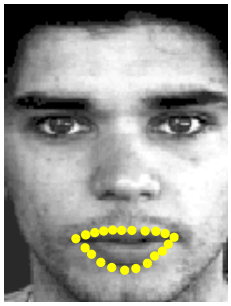


Figure 5.13: Lip tracking template.

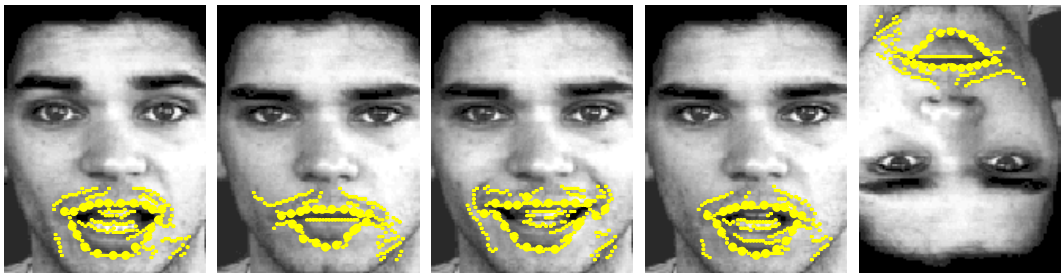


Figure 5.14: Extracted edges on 5 of the 20 test images. Matched edges are marked with bigger dots.

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 — Figure 5.13. 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 equation (5.8). Figure 5.14 shows extracted points on 5 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 11. Figure 5.15 shows results obtained from the application of these linear deformations to the model. 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 of Section 5.1.

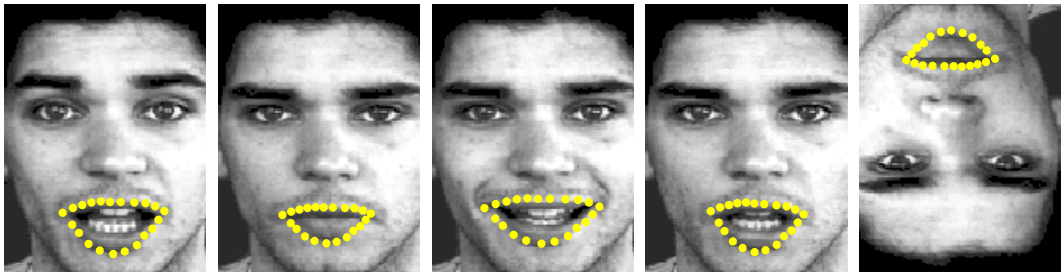


Figure 5.15: Results of lip tracking on 5 of the 20 test images. These point sets are affine transformations of the point set of Figure 5.13.

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) [108]. This method finds closest points 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 [84]. 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

$$\mathbf{G}_{i,j} = e^{-\frac{\left\|\mathbf{x}_i^1 - \mathbf{x}_j^2\right\|^2}{2\sigma^2}} \tag{5.9}$$

where $\mathbf{x}_i^f$ is the coordinate vector of the i-th 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 [86]. It is a modification of the previous method, intended to solve situations with large translation, rotation and scaling. It starts measuring intra-frame point distances, and uses them to

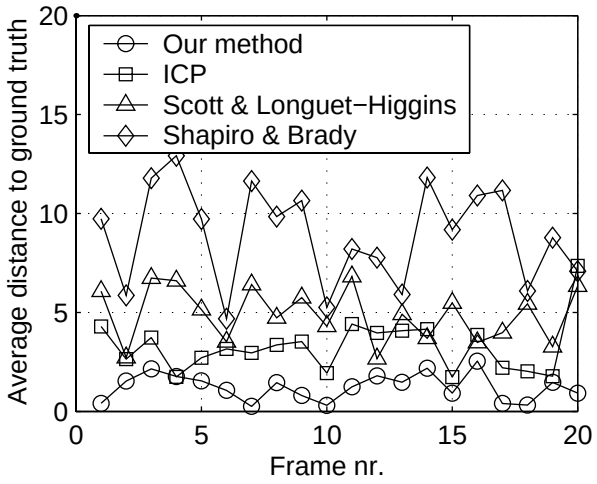


Figure 5.16: Performance of four different lip-tracking algorithms.

compute the symmetric matrix:

$$\mathbf{H}_{i,j}^{f} = e^{-\frac{\left\|\mathbf{x}_i^f - \mathbf{x}_j^f\right\|^2}{2\sigma^2}} \tag{5.10}$$

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 Figure 5.16.

Discussion

Figure 5.16 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

Figure 5.17: Original images of the subjects.

Figure 5.18: Bottom rows are modified according to the examples on top.

rotation is never captured. The last algorithm fails because of the large number of spurious points, which affects the shape modes.

Application

This algorithm can be used to track all facial features through image sequences. These tracks can be used to compute shape and texture representations of face images [55]. This representation allows us to generate sequences of images of a person in action, using a single image of that person.

For example, starting from the images of Figure 5.17, we were able to synthesize all the images on the bottom row of Figure 5.18. The subjects mimic the actions of an exemplifying actor — top row — whose face features were tracked.

5.6 3D point registration

In this section we describe a method of performing 3D point registration, similar to the 2D registration of Section 5.5. Non-contact 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

$$\mathbf{X} = \begin{bmatrix} x_1^1 & y_1^1 & z_1^1 \\ \vdots & \vdots & \vdots \\ x_{p_1}^1 & y_{p_1}^1 & z_{p_1}^1 \end{bmatrix} \quad , \quad \mathbf{Y} = \begin{bmatrix} x_1^2 & y_1^2 & z_1^2 \\ \vdots & \vdots & \vdots \\ x_{p_2}^2 & y_{p_2}^2 & z_{p_2}^2 \end{bmatrix} \tag{5.11}$$

The goal is to find a 3D rigid transformation that aligns a fraction of these points, and reject non-overlapping points. We propose to search for a general linear transformation that aligns subsets of the data, solving Problem 12 where $\mathbf{L}$ is a 3D linear transformation

Problem 12

$$\begin{aligned} \mathbf{P}^* \;=\; \arg\min_{\mathbf{P}} \left(\min_{\mathbf{L}} \|\mathbf{CX} - \mathbf{CPYL}\|^2 \right) \\ \text{s.t.} \quad \mathbf{P} \in \mathcal{P}_p(p_1, p_2) \end{aligned}$$

With 3D data, the approximate linear cost function — equation (5.8) — transforms to

$$\tilde{J}_{3D}^i(\mathbf{P}) \;=\; \pm(\mathbf{\Pi}_x\mathbf{CPx}) \pm (\mathbf{\Pi}_x\mathbf{CPy}) \pm \ldots \pm (\mathbf{\Pi}_z\mathbf{CPz}) \quad , \quad i = 1, \ldots, 512 \tag{5.12}$$

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 $\mathcal{P}_{\mathrm{p}}^{p_{\mathrm{t}}}(p_1, p_2)$ 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% of p_1 and used a *ransac* procedure to find the most consistent points. We solved Problem 12 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.

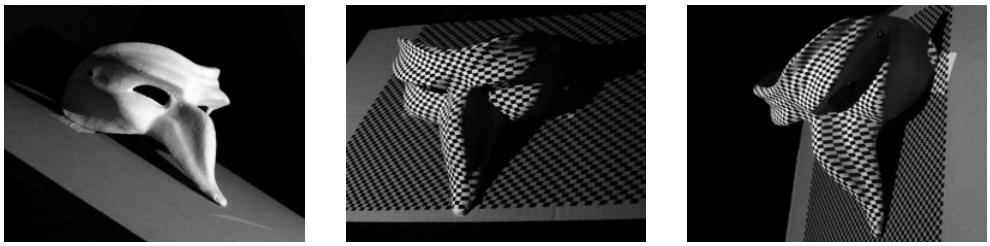


Figure 5.19: Three views of a *mask*, with projected structured-light.

Figure 5.20: Three point sets to be registered.

Results

We used a structured-light 3D reconstruction system on the object of Figure 5.19, building 3 clouds of 3D points — Figure 5.20. The set in the middle is reference $\mathbf{X}$.

Each set contains around 1600 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 6000. 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% smallest distances. Figure 5.21 shows the registration of the 3 data sets.

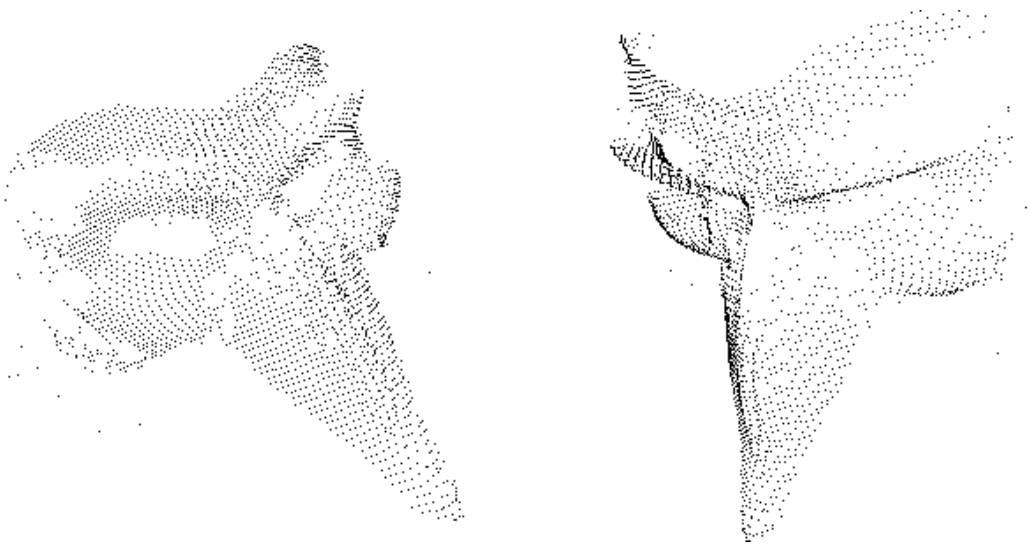


Figure 5.21: Two views of the registered 3D data sets.

Discussion

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.

Chapter 6

Conclusion

In this thesis, we proposed a new methodology to solve correspondence problems. Our methodology handles different feature representations and different assumptions using a unique formulation, in the form of a concave minimization problem. Correspondence and outlier rejection are both taken into account, in a compact and integrated way.

In order to reduce the dimensionality of the resulting concave problems, extra assumptions can be introduced in the form of support constraints. In some experiments, we used the epipolar constraint and bounds of disparity as extra constraints.

Several matching criteria were cast to polynomial cost functions that were conveniently made concave. Uniqueness constraint and feature rejection are represented as sets of linear constraints. The concave cost functions together with linear constraints form *concave programming* problems that were solved by three different algorithms that avoid combinatorial search.

Global matching criteria provide the most reliable correspondence methods, because the whole set of features and the whole set of possible correspondences are considered. Our methodology is able to handle global criteria, as long as a class C^2 cost function can be found. We used correlation, epipolar coherence, affine shape models and uncalibrated 3D rigidity under orthography as matching criteria. When using the rigidity criterion, correspondences were set such that they optimized the residual of the factorization method for shape and motion computation.

Experiments showed that the methods can handle outliers, even in situations where other robust methods fail. We have also shown that the methodology avoids the inclusion of unwanted assumptions by requiring their explicit statement.

Developed methods were tested in real images and compared with benchmark algorithms. Combining rigidity and correlation matching, we were able to perform a fully automated 3D reconstruction of the *Hotel* sequence, without any prior calibration and without using proximity constraints. This allows us to find correspondences in situations that would be practically impossible to solve using other methods.

6.1 Limitations

The most important limitation of the methodology is the dimensionality of the optimization problems. This constrains the practical size of data sets, specially when the objective function is a high-order polynomial. We implemented the simplest available optimal algorithm, so we were only able to deal with large data sets using either linear cost functions — correlation matching and point registration — or suboptimal concave programming algorithms. The addition of *a priori* constraints, in the form of support matrices, eases this problem — Section 2.6.

Another important limitation of the methodology is the nonexistence of a general way to perform feature selection on all images. In a general case, features on the first and second images can be rejected using $\mathbf{P}\mathbf{P}^\top\mathbf{X}$ and $\mathbf{P}\mathbf{Y}$. 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 produces cost functions of higher degree.

Finally, each application example has its own specific limitations. The correspondence method for calibrated trinocular systems does not correctly solve ambiguities. Correlation is not a good criterion to match straight edges because of directional uncertainty. The 2D registration method is unable to handle nonlinear transformations, which may occur in facial deformations. The 3D registration method searches for general linear transformations, and hence ambiguous solutions are likely to occur.

6.2 Future work

Future work will be conducted having in mind to overcome the limitations of the methodology and extend its application. Next, we present the most important issues.

Implementation of efficient CP algorithms for high-order problems.

We plan to implement faster algorithms by exploiting the special structure of the problems. For example, the biquadratic rigidity cost function is the difference of two convex functions. We have been considering a special class of algorithms — DC programming algorithms — that deals efficiently with this type of cost functions.

Development of approximated lower-order cost functions

Low order cost functions allow us to use larger data sets. In Sections 5.5 and 5.6 we described low-order approximate cost functions for 2D and 3D point registration. In the near future we will search for lower-order approximations to the rigidity cost function.

Inclusion of new constraints

The methodology would benefit from the inclusion of new constraints, since ambiguous cases would be reduced. For example, visibility is a generic and highly constraining geometric relation. In [89], visibility is expressed as a set linear constraints, but the integral property of the constraining polytope is destroyed.

Application to other problems

We are considering the application of the methodology to model matching problems. Model matching is the comparison of a previously obtained 3D model with a partial 2D view of the object. This problem is solved with the rigidity constraint once an efficient mechanism to reject points from all images is devised. The model can also be recursively updated and refined, as long as new images are observed and matched.

The methodology is also applicable to problems other than matching — Section 6.3.

6.3 Close range topics

In this final section we briefly describe possible ways of tackling some of the problems listed in Section 6.2.

Successive approximation CP algorithm

We have been developing an algorithm that reduces the CP problem to a system of nonlinear equations. The algorithm searches for a level set of the cost function that intersects the constraining polytope at a single point. It iteratively approximates a pair of bounds k_l^i and k_u^i by bisection, until they converge.

Consider the notation of the generic concave Problem 10 — Chapter 4. The algorithm consists of the following steps:

1. Let $i \leftarrow 0$. Initialize $k_l^0 < J(\mathbf{q}^*)$ and $k_u^0 > J(\mathbf{q}^*)$.

2. Let $i \leftarrow i+1$. Compute $k^i = \frac{1}{2}\left(k_l^{i-1} + k_u^{i-1}\right)$

3. Define a level set $\mathcal{L}^i \equiv \{\mathbf{q} : J(\mathbf{q}) = k^i\}$.

4. **a)** If $\mathcal{L}^i \cap \mathcal{C} = \emptyset$ then $k_l^i \leftarrow k^i$.

b) Otherwise $k_u^i \leftarrow k^i$.

5. **a)** If $k_u^i - k^i < t$ the bounds collapsed, goto 6.

b) Otherwise goto 2.

6. Define $i^* \leftarrow i$ and $\mathcal{L}^* = \mathcal{L}^{i^*}$. Solve $\mathcal{L}^* = \mathcal{C}$.

If $J(\mathbf{q})$ is polynomial, step 4. can be solved using the technique of [6] that checks feasibility of systems of polynomial equations. Bounds get closer at each iteration and, for small enough t, $\mathcal{L}^* \cap \mathcal{C}$ contain a single — possibly degenerate — extreme point of $\mathcal{C}$. That extreme point is the solution to the concave problem $\mathbf{q}^*$. It is found solving the system of equations of 6. Note that all but one equations of the system are linear.

Application to 3D point triangulation

Triangulation of 3D point clouds is another challenging problem in Computer Vision. Triangulation can be used directly for rendering, and it provides valuable topological information for posterior curve and surface interpolation.

Next, we describe a tentative formulation of this problem, using our methodology. First build a list with all possible $m = \frac{n^2+n}{2}$ pairs of points — edges. A triangle is represented by a set of 3 different edges. Register the area of all valid triangles in an $m \times m \times m$ 3-way array $\mathbf{D}$. The full triangulation is represented by an indicator 3-way array $\mathbf{P}$ that selects entries of $\mathbf{D}$. A proper triangulation must meet these conditions:

1. Only sets of edges with 3 common points represent valid triangles.
2. Only a tetrahedron of $\mathbf{P}$ if valid — uniqueness of triangle representation.
3. Edges cannot belong to more than 2 triangles — avoid surface intersections.
4. Edges must belong at least to 1 triangle — avoid isolated edges.

Conditions 1. and 2. are represented by a support 3-way array $\mathbf{S}$. Conditions 3. and 4. are represented by a set of linear inequalities on the entries of $\mathbf{P}$. Using this notation, triangulation can be cast to a problem with Totally Unimodular constraints and the following linear cost function, that measures total surface area:

$$J_{tri}(\mathbf{P}) = \sum \sum \mathbf{P} \odot \mathbf{D} \tag{6.1}$$

This cost function introduces a regularization effect. The rank of $\mathbf{P}$ — number of ones — is constrained to p_{t}. Again, prior knowledge is used to build support constraints that reduce problem dimensionality.

The weakness of this formulation lies in its high sensitivity to the value of p_{t}. If p_{t} is too low, the result is a set of unconnected small triangles. If p_{t} is too high, the problem will be infeasible. If the highest feasible p_{t} is chosen, outlier rejection will not be effective. In conclusion, this formulation lacks an extra constraint or regularization term to assure triangle connectivity with low values of p_{t}.

Application to multi-body segmentation

We also plan to formulate the multi-body segmentation problem [21] using our methodology. Suppose that an orthographic camera observes an unknown number of independently moving objects. Assume that we were able to track feature points belonging to these objects but we do not yet know how to group the features into objects. Among all possible row permutations of the observation matrix ($\mathbf{W_P} = \mathbf{PW}$) we wish to find which permutation has features from the same object in contiguous rows. Such a sorted matrix $\mathbf{W}^* = \mathbf{W}_{\mathbf{P}^*}$ is said to be in canonical form.

The method in [21] solves this problem using the rigidity constraint. It uses an invariant square matrix $\mathbf{Q} = \mathbf{U}\mathbf{U}^\top$ where $\mathbf{U}$ are the left singular vectors of the observation matrix $\mathbf{W}$. $\mathbf{Q}$ is called the *shape interaction matrix*. Without noise, entry $\mathbf{Q}(i,j)$ is zero whenever features i and j belong to different objects. Each row permutation of $\mathbf{W}$ corresponds to a row and column permutation $\mathbf{Q_P} = \mathbf{PQP}^\top$ of the shape interaction matrix. A correctly sorted shape interaction matrix $\mathbf{Q}^* = \mathbf{Q}_{\mathbf{P}^*}$ has block diagonal structure, therefore the problem of segmenting multiple objects boils down to permuting rows and columns of $\mathbf{Q}$ until it becomes block diagonal.

With noise, a pair of features i and j from different objects may exhibit a small nonzero value of $\mathbf{Q}(i,j)$, so in [21] the energy of off-diagonal blocks is minimized. The challenge consists on expressing this energy criterion as an explicit polynomial function J_{mb} on the entries of the observation matrix. Once this is done, the multi-body segmentation problem is written as

$$\begin{aligned} \mathbf{P}^* \;=\; \arg\min_{\mathbf{P}} \quad & J_{mb}(\mathbf{PW}) \\ s.t. \quad & \mathbf{P} \in \mathcal{P}_{\mathrm{p}}^{\mathrm{c}} \end{aligned} \tag{6.2}$$

$\mathbf{P}^*$ not only sorts the rows of $\mathbf{W}$ in the correct way, but is also eliminates spurious features.

Appendix

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 non-square matrices, defining them using conditions (2.2), (2.3) and (2.4).

A $[p_1 \times p_2]$ $\mathrm{p_p}$-matrix $\mathbf{P}$ is rank-p_t *iff* it also complies with condition (A.1)

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

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 conditions (2.4) and (A.1) can be changed to a single condition (A.2):

$$\sum_{j=1}^{p_2}\mathbf{P}_{i,j} = 1 \ , \ \forall i = 1 \ldots p_1 \tag{A.2}$$

The resulting set of matrices $\mathcal{P}_\mathrm{p}^\mathrm{c}(p_1, p_2)$ is denoted the set of *column-wise partial permutation matrices*. Finally, rank can be bounded if condition (A.1) is changed to

$$\sum_{i=1}^{p_1}\sum_{j=1}^{p_2}\mathbf{P}_{i,j} \leq p_\mathrm{t} \tag{A.3}$$

A.1.1 Relaxations

As stated in Section 2.1.1, we define $\mathcal{DS}_{\mathrm{s}}$ — the set of *doubly substochastic matrices* — by conditions (2.3), (2.4) and (2.5). Section 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 conditions (2.3), (2.4), (2.5) and (A.1). As shown in Section 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 conditions (2.3), (2.5) and (A.2). In Section 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 3 *For given p_1, p_2, the elements of $\mathcal{P}_p(p_1, p_2)$ are the vertices of $\mathcal{DS}_s(p_1, p_2)$.*

Proposition 4 *For given p_1, p_2, the elements of $\mathcal{P}_p^{p_t}(p_1, p_2)$ are the vertices of $\mathcal{S}_s^{p_t}(p_1, p_2)$.*

Proposition 5 *For given p_1, p_2, the elements of $\mathcal{P}_p^{c}(p_1, p_2)$ are the vertices of $\mathcal{S}_s^{c}(p_1, p_2)$.*

These results are generalizations of Birkhoff's theorem — see [35, 13, 69] — 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 [69].

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 Section 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 proved in Section 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 condition (2.2) 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

$$\alpha_1 \mathbf{P}_1 + \alpha_2 \mathbf{P}_2 = \mathbf{P} \tag{A.4}$$

Non-negativity 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. In conclusion, all elements of $\mathcal{P}_{\mathrm{p}}(p_1, p_2)$ are extreme points of $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$.

A.2.2 Necessity

We now show that the constraint matrices of equations (2.14), (2.15) and (2.16), 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 3 — proved in [69].

Theorem 3 *If* $\mathbf{A}$ *is an* $m \times n$ *Totally Unimodular (TU) matrix, then the polytope* $\mathcal{C} = \{\mathbf{q} \in \mathbb{N}^n \ : \ \mathbf{Aq} \leq \mathbf{b}\}$ *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 proved in [69].

Theorem 4 $\mathbf{A}$ *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*

$$\left| \sum_{i \in I_1} a_{ij} - \sum_{i \in I_2} a_{ij} \right| \leq 1 \ , \ \ \forall j = 1, \ldots, n \tag{A.5}$$

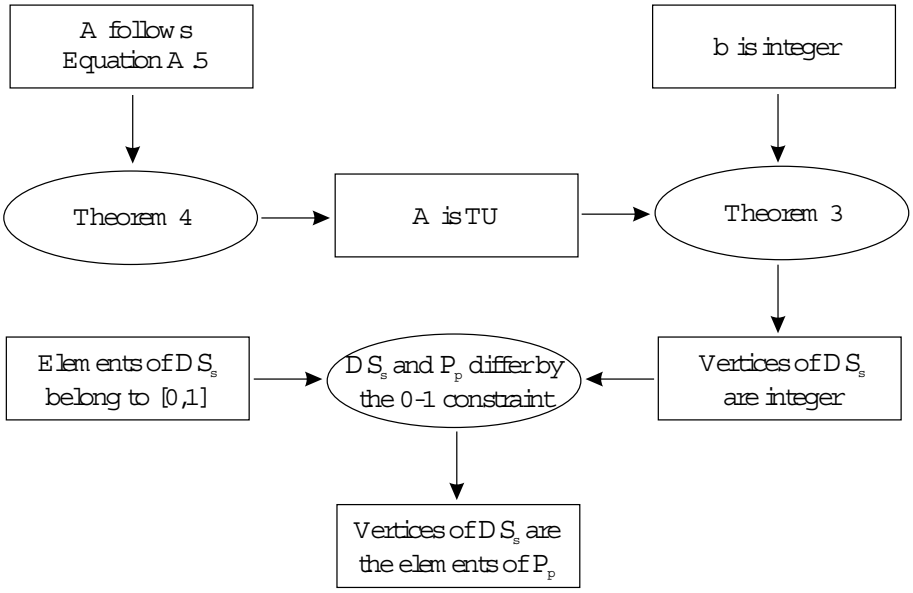


Figure A.1: Outline of the proof for necessity in Proposition 3

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 — equations (2.9), (2.10), (2.11), (2.12) and (2.13). Therefore, showing that their $\mathbf{A}$ matrices satisfy equation (A.5) ensures that the conditions of Theorem 3 are met and, therefore, that $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$, $\mathcal{S}_{\mathrm{s}}^{p_{\mathrm{t}}}(p_1, p_2)$ and $\mathcal{S}_{\mathrm{s}}^{\mathrm{c}}(p_1, p_2)$ are integral polytopes. Figure A.1 shows an outline of the entire demonstration.

Integral property of $\mathcal{DS}_{\mathbf{s}}$

The matrix $\mathbf{A}$ of equation (2.14) satisfies the conditions of equation (A.5) 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 condition (A.5) always holds. Since $\mathbf{A}$ is TU, $\mathbf{b}$ is integer and $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$ is not empty then we conclude that all the vertices of $\mathcal{DS}_{\mathrm{s}}(p_1, p_2)$ are integer.

Integral property of $\mathcal{S}_{\mathbf{s}}^{p_{\mathbf{t}}}$

The matrix $\mathbf{A}$ of equation (2.15) satisfies the conditions of equation (A.5) if, for each row selection I, we choose a partition in the following way:

If I does not include neither $\mathbf{A}_4$ nor $\mathbf{A}_5$: Build the partition as in Section A.2.2, that is, I_1 should include only rows from block $\mathbf{A}_1$ and I_2 should include only rows from block $\mathbf{A}_3$. Each block contains at most one nonzero element per column, so condition (A.5) always holds.

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 $\mathbf{A}_5$. Every column contains entries 1 and -1 which will cancel out, so column sums will still be constrained to $\{-1, 0, 1\}$.

If I includes $\mathbf{A}_4$ but not $\mathbf{A}_5$: Put all elements from $\mathbf{A}_1$ 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. In short, column sums will be constrained to $\{-1, 0, 1\}$ as desired.

If I includes $\mathbf{A}_5$ but not $\mathbf{A}_4$: Put all rows in I_1 and leave I_2 empty. Column sums will be, once more, constrained to $\{-1, 0, 1\}$.

Integral property for bounded rank

The 0-1 relaxation is still valid if $\mathrm{p_p}$-matrices have bounded rank — condition (A.1) changed by condition (A.3), so $\mathbf{A}_5$ does not appear in the constraints. The demonstration is similar to that of Section A.2.2, but cases with $\mathbf{A}_5$ can be discarded.

Integral property of $\mathcal{S}_\mathrm{s}^\mathbf{c}$

The matrix $\mathbf{A}$ of equation (2.16) is in the conditions of equation (A.5). For each row selection I, choose a partition in the following way:

If I does not include $\mathbf{A}_2$: Choose a partition just 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 2.6 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 Section 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. This result is used in Section 2.7.

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 4 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 equation (A.5) holds. We known this because Theorem 4 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

$$I_1 = \bigcup_{f=1}^{F} I_1^f \quad , \quad I_2 = \bigcup_{f=1}^{F} I_2^f \tag{A.6}$$

The nonzero entries of each column of $\mathbf{A}$ belong to a single block — $\mathbf{A}$ is block-diagonal — so equation (A.5) holds for all columns of $\mathbf{A}$. We can apply the sufficiency of Theorem 4 and conclude that $\mathbf{A}$ is TU.

Appendix B

Rank Theorem in scaled-orthography

Theorem 2 is a generalization of the Rank Theorem of [97], and our proof follows their approach. Consider the image coordinates of extracted feature-points, grouped together in a matrix of centered observations

$$\mathbf{W}_{[p_1 \times 2F]} \;=\; [\mathbf{CX} \mid \mathbf{CPY}] \tag{B.1}$$

where $\mathbf{P}$ is a $\mathcal{P}^{\mathrm{c}}_{\mathrm{p}}(p_1, p_2)$ matrix, $\mathbf{C}$ is the centering matrix of equation (3.3), and the measurements are not corrupted by noise. Refer to Figure B.1, where the origin $\mathbf{o}$ of the world coordinate system is at the centroid of the object points $\mathbf{s}_p = (x_p, y_p, z_p)$. In scaled-orthographic model, the projection of a centroid is the centroid of the projections. We consider $\mathbf{s}_p$ and $\mathbf{t}^f$ as row vectors, in order to simplify the notation. The projection $\left(u_p^f, v_p^f\right)$ of each point $\mathbf{s}_p$ onto frame f is given by

$$u_p^f \;=\; \alpha^f \left(\mathbf{s}_p - \mathbf{t}^f\right) \mathbf{i}^f \tag{B.2}$$

$$v_p^f \;=\; \alpha^f \left(\mathbf{s}_p - \mathbf{t}^f\right) \mathbf{j}^f \tag{B.3}$$

Under scaled-orthography, points are projected along $\mathbf{k}^f = \mathbf{i}^f \times \mathbf{j}^f$ and scaled with factors α^f. When $\mathbf{P} = \mathbf{P}^*$ is correct then $\mathbf{PY}$ has the features placed in the same order

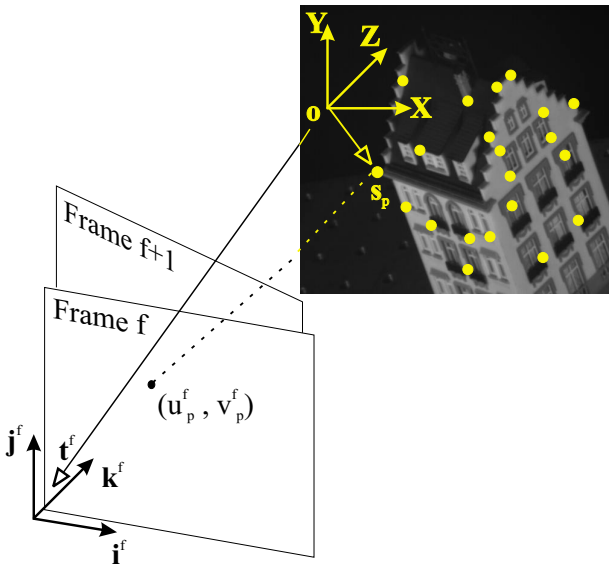


Figure B.1: The camera, the object and their coordinate systems.

as those of $\mathbf{X}$, so the centered observation matrices yield

$$\begin{aligned}\tilde{u}_p^f &= u_p^f - \frac{1}{p_1}\sum_{p=1}^{p_1} u_p^f \\ &= \alpha^f\left(\mathbf{s}_p - \mathbf{t}^f\right)\mathbf{i}^f - \frac{1}{p_1}\sum_{p=1}^{p_1}\alpha^f\left(\mathbf{s}_p - \mathbf{t}^f\right)\mathbf{i}^f \\ &= \alpha^f\left(\mathbf{s}_p - \frac{1}{p_1}\sum_{p=1}^{p_1}\mathbf{s}_p\right)\mathbf{i}^f \end{aligned} \tag{B.4}$$

The origin of the world coordinate system is at the centroid of the object, therefore $\sum_{p=1}^{p_1}\mathbf{s}_p = \mathbf{0}$, and so $\tilde{u}_p^f = \alpha^f\mathbf{s}_p\mathbf{i}^f$. Similarly $\tilde{v}_p^f = \alpha^f\mathbf{s}_p\mathbf{j}^f$ so $\mathbf{W}$ can be written as

$$\mathbf{W} = \mathbf{SM\Lambda} \tag{B.5}$$

where $\mathbf{S}$ — shape matrix — gathers the $\mathbf{s}_p$ vectors, $\mathbf{M} = \left[\mathbf{i}^1\ \mathbf{j}^1 \cdots \mathbf{i}^f\ \mathbf{j}^f\right]$ represents the camera rotation, and $\mathbf{\Lambda}$ contains the scale factors organized in the following way

$$\mathbf{\Lambda} = \begin{bmatrix} \alpha^1 & & \\ & \ddots & \\ & & \alpha^{p_1} \end{bmatrix} \otimes \mathbf{I}_{[2]} \tag{B.6}$$

Since $\mathbf{S}$ is $p_1 \times 3$ and $\mathbf{M}$ is $3 \times 2F$, then matrix $\mathbf{W}$ is, at most, rank three.

Appendix C

Maximizing rigidity: degenerate cases

In Theorem 2 we only state a necessary condition for rigidity. If spurious points exist, it is possible that different point selection and permutations generate different rank-3 observation matrices and, consequently, different possible object reconstructions. In this appendix we present two different types of degenerate cases, and explain how they can be handled.

C.1 Multiple candidates

Consider the situation depicted in Figure C.1, where motion is assumed to be known *a priori* and feature coordinates are exact — no noise. The epipolar line crosses edges in two different locations. Both locations are consistent with the motion so there are at least two possible optimal solutions for Problem 6. With noise, any one of them — not necessarily the correct — can become the optimal. This situation some times occurs in practice, if some features are to be rejected. They are well handled when p_t is close to p_2 — uniqueness constraint. Anyway, all the solutions share a common row space, so the estimated motion will always be correct.

Using more than two images decreases the number of degenerate cases dramatically.

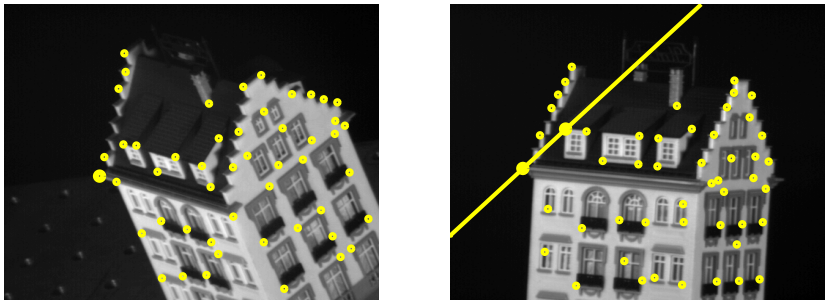


Figure C.1: A scaled-orthographic stereo pair with known motion. The epipolar line corresponds to the highlighted point

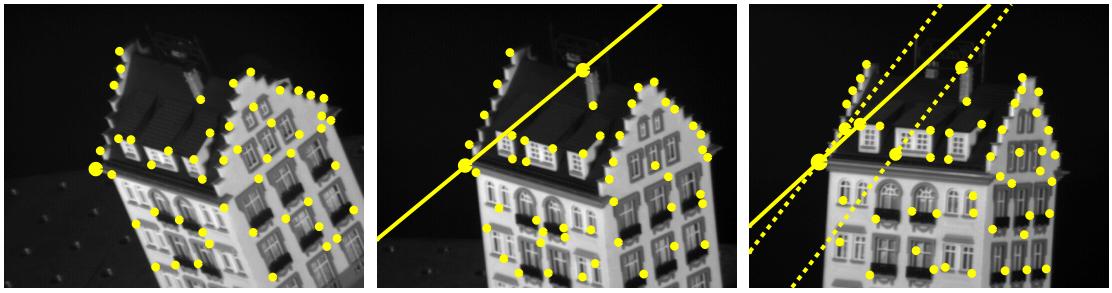


Figure C.2: A sequence of 3 scaled-orthographic images with known motion. Epipolar lines in full correspond to the highlighted point. Dashed epipolar lines correspond to each one of the candidate matches.

Add a third image to the observations, with known motion — Figure C.2. The highlighted point generates an epipolar line on each one of the subsequent images. Each candidate generates its own epipolar line on the third image, and each one of these has their own candidates matches. However there is one single correspondence solution among the three images that is consistent with all epipolar lines. In practice it is hard to find multi-image examples with multiple solutions.

In conclusion, finding a rank-3 correspondence between points of two or more images is equivalent to finding set of point correspondences for which a consistent set of epipolar constraints can be found. Using longer image sequences decreases the number of solutions with consistent motion. The best practice is, therefore, to use the largest possible number of images, in the multi-image framework of Section 3.6.

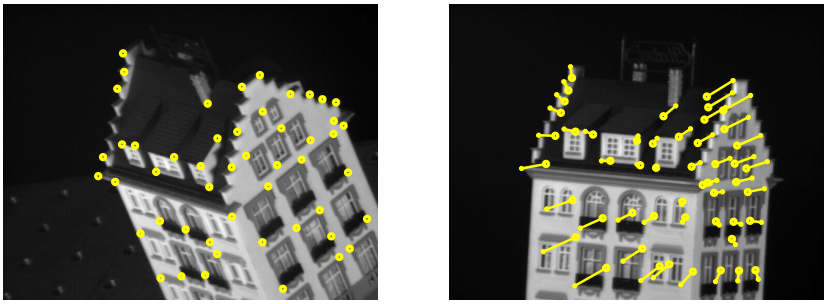


Figure C.3: A stereo pair with candidate matches with linearly related coordinates.

C.2 Linearly-dependent observations

In this section we describe a degenerate situation that can occur independently of the number of frames. A method to solve it is presented in Section 3.7.

Consider again the example in Figure C.1. Assume that the points were segmented without noise, and that correspondences were correctly established. Such measurements generate a rank-3 observation matrix

$$\mathbf{W}_1 = [\, \mathbf{CX} \mid \mathbf{CP}^*\mathbf{Y} \,] \tag{C.1}$$

For the correct $\mathbf{P}^*$, $\mathbf{W}_1$ is rank-3. Now apply a 2D linear transformation $\mathbf{L}$ to the coordinates of the rightmost observations.

$$\mathbf{W}_2 = [\, \mathbf{CX} \mid \mathbf{CP}^*\mathbf{YL} \,] \tag{C.2}$$

These are depicted in Figure C.3, superimposed with the original ones. $\mathbf{W}_2$ is obviously still rank-3. In conclusion, if the p_2 points include those of Figure C.3, then there will be at least two exact solutions to this noise-free correspondence problem. In the presence of noise, the cost values of these solutions are perturbed and the emergent optimal solution is randomly chosen. The best practice consists of generating the *best* N solutions — with cost value close to optimal — and choose the one most consistent with the metric constraints — Section 3.7.

Bibliography

Bibliography

[1] P. Aguiar and J. Moura. Factorization as a rank-1 problem. In *Proc. CVPR*, pages 1063–1069, June 1999.

[2] J. Aloimonos. Perspective approximations. *Image and Vision Computing*, 8(3):179–192, Aug. 1990.

[3] J. Aloimonos and J. Hervé. Correspondenceless stereo and motion: Planar surfaces. *IEEE Trans. PAMI*, 12(5):504–510, May 1990.

[4] P. Apkarian and H. Tuan. Robust control via concave minimization – local and global algorithms. In *Proc. Conference on Decision and Control*, Tampa, Florida, 1998.

[5] H. Arsham. Initialization of the simplex algorithm: an artificial-free approach. *SIAM Rev.*, 39(4):736–744, Dec. 1997.

[6] Barvinok. Feasibility testing for systems of real quadratic equations. *Discrete and Computational Geometry*, 10:1–13, 1993.

[7] R. Basri. Paraperspective ≡ affine. *IJCV*, 19(2):169–179, Aug. 1996.

[8] R. Basri and D. Weinshall. Distance metric between 3d models and 3d images for recognition and classification. Technical Report AIM 1373, MIT, AI Lab, Jul. 1992.

[9] M. Bazaraa, J. Jarvis, and H. Sherali. *Linear Programming and Network Flows*. Wiley, second edition, 1990.

[10] A. Bensrhair and P. Debrie. Fast and automatic stereo vision matching algorithm based on dynamic programming method. *Pattern Recognition Letters*, 17:457–466, 1996.

[11] R. Berthilsson and K. Astrom. Reconstruction of 3D-curves from 2D-images using affine methods for curves. In *Proc. CVPR*, 1997.

[12] R. Berthilsson, K. Astrom, and A. Heyden. Projective reconstruction of 3d-curves from its 2d-images using error models and bundle adjustments. In *Scandinavian Conference on Image Analysis*, 1997.

[13] R. Bhatia. Matrix analysis. In S. Alex, editor, *Graduate Texts in Mathematics*. Springer, 1997.

[14] C. Branco and J. Costeira. A 3d image mosaiching system using the factorization method. In *IEEE ISIE*, Pretoria, South Africa, July 1998.

[15] R. Burkard, E. Çela, and Klinz B. On the biquadratic assignment problem. pages 117–146, May 1993.

[16] A. Cabot and R. Francis. Solving certain nonconvex quadratic minimization problems by ranking the extreme points. *Operations Research*, 18(1):82–86, Feb 1970.

[17] J. Carillo. A relaxation for the minimization of a quasiconcave function on a convex polyhedron. *Mathematical Programming*, 13:69–80, 1977.

[18] Y. Cheng, R. Collins, A. Hanson, and E. Riseman. Triangulation without correspondences. In *Proc. ARPA IUW*, Monteray, CA, 1994.

[19] Y. Cheng, V. Wu, R. Collins, A. Hanson, and E. Riseman. Maximum-weight bipartite matching technique and its application in image feature matching. In *Proc. of the SPIE Conf. on Visual Communication and Image Processing*, Orlando, FL, 1996.

[20] R. Collins. A space-sweep approach to true multi-image matching. In *Proc. CVPR*, San Francisco, CA, 1998.

[21] J. Costeira and T. Kanade. A multibody factorization method for independent moving objects. *IJCV*, 29(3):159–179, 1998.

[22] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. Structure from motion without correspondence. In *Proc. CVPR*. IEEE Press, June 2000.

[23] J. Falk and K. Hoffman. A successive underestimation method for concave minimization problems. *Mathematics of Operations Research*, 1(3):251–259, August 1976.

[24] O. Faugeras and T. Papadopoulo. A theory of the motion field of curves. *IJCV*, 10(2):125–156, 1993.

[25] G. Fielding and M. Kam. Weighted matchings for dense stereo correspondence. *Pattern Recognition*, 33:1511–1524, 2000.

[26] R. Fletcher. *Practical methods of optimization*. Wiley, second edition, 1987.

[27] C. Floudas and P. Pardalos. *Recent Advances in Global Optimization*. Princeton, 1992.

[28] C. Floudas and V. Visweswaran. Quadratic optimization. In *Handbook of global optimization*, pages 217–269. Kluwer, 1995.

[29] Christodoulos A. Floudas. Deterministic global optimization in design, control, and computational chemistry. In *IMA Proceedings: Large Scale Optimization with Applications. Part II: Optimal Design and Control*, pages 129–184. Springer-Verlag, 1997.

[30] S. Gold and A. Rangarajan. Softassign versus softmax. In *Advances in Neural Information Processing Systems*, volume 8, pages 626–632, 1996.

[31] S. Gold, A. Rangarajan, C. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2D and 3D point matching. *Pattern Recognition*, 31(8):1019–1031, 1998.

[32] E. Grossman and J. Santos-Victor. A closed-form solution for paraperspective reconstruction. In *Proc. ICPR*, Barcelona, Spain, September 2000.

[33] R. Haralick and L. Shapiro. *Computer and Robot Vision*. Addison-Wesley, 1993.

[34] R. Hoarst and H. Tuy. *Global Optimization: deterministic approaches*. 1996.

[35] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge U. Press, 1985.

[36] R. Horn and C. Johnson. *Topics in Matrix Analysis*. Cambridge U. Press, 1991.

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

[38] R. Horst, P. Pardalos, and N. Thoai. *Introduction to Global Optimization*, volume 3 of *Nonconvex Optimization and Its Appl.* Wiley, 1990.

[39] R. Horst and N. Thoai. Modification, implementation and comparison of three algorithms for globally solving linearly constrained concave minimization problems. *Computing*, 42:271–289, 1989.

[40] X. Hu and N. Ahuja. Matching point features with ordered geometric, rigidity and disparity constraints. *IEEE Trans. PAMI*, 16(10):1041–1049, 1994.

[41] M. Irani and P. Anandan. Factorization with uncertainty. In *Proc. ECCV*, 2000.

[42] J. Júdice and A. Faustino. Solution of the concave linear complementary problem. In *Recent Advances in Global Optimization*, pages 76–101, 1992.

[43] T. Kanade and D. Morris. Factorization methods for structure from motion. *Phil. Trans. R. Soc. London, Series A*, 356(1740):1153–1173, 1998.

[44] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window. *IEEE Trans. PAMI*, 16(9):920–932, Sep 1994.

[45] K. Kanatani. Factorization without factorization: Statistical analysis. Technical report, Dept. Computer Science, Gunma University, Japan, 1998.

[46] N. Karmakar, M. Resende, and K. Ramakrishnan. An interior point algorithm to solve computationally difficult set covering problems. *Mathematical Programming*, 52:597–618, 1991.

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

[48] H. Konno. Maximization of a convex quadratic function under linear constraints. *Mathematical Programming*, 11:117–127, 1976.

[49] A. Krishna. Note on degeneracy. Technical Report SOL 89-4, Systems Optimization Laboratory, Dept. of Operations Research, Stanford University, April 1989.

[50] H. Kruse. *Degeneracy Graphs and the Neighbourhood Problem*. Springer-Verlag, 1986.

[51] C. Lee and T. Huang. Finding point correspondences and determining motion of a rigid object from two weak perspective views. *Computer Vision and Image Processing*, 52:309–327, 1990.

[52] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of the 7th International Joint Conference on AI*, 1981.

[53] D. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, second edition, 1989.

[54] H. Lutkepohl. *Handbook of Matrices*. Wiley, 1996.

[55] J. Maciel and J. Costeira. Holistic synthesis of human face images. In *Proc. of the IEEE Int. Conf. Acoustics Speech Signal Proc.*, Phoenix, AZ, 1999.

[56] 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.

[57] J. Maciel and J. Costeira. Robust point correspondence by concave minimization. In *Proc. of the 11th BMVC*, pages 626–635, Bristol, UK, 2000.

[58] J. Maciel and J. Costeira. Robust point correspondence by concave minimization. *to appear in BVMC'2000 special issue IVC*, 2001.

[59] J. Maciel and J. Costeira. Towards a global solution of the correspondence problem. *submitted to IEEE Trans. PAMI*, 2001.

[60] G. Manku, P. Jain, A. Aggarwal, and L. Kumar. Object tracking using affine structure for point correspondences. In *Proc. CVPR*, pages 704–709, 1997.

[61] J. Mendelsohn, E. Simoncelli, and R. Bajcsy. Discrete-time rigidity-constrained optical flow. In *Proc. CAIP*, Sep. 1997.

[62] C. Meyer. A simple finite cone covering algorithm for concave minimization. Technical Report 193, Karl-Franzens-Universitat Graz, Apr. 2000.

[63] J. Mitchell, P. Pardalos, and M. Resende. Interior point methods for combinatorial optimization. In *Handbook of global optimization*. Kluwer, 1995.

[64] T. Morita and T. Kanade. A sequential factorization method for recovering shape and motion from image streams. In *Proc. of the ARPA Image Understanding Workshop*, pages 1177–1187, Monterrey, 1994.

[65] D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Proc. ICCV*, pages 696–702, Bombay, India, Jan. 1998.

[66] D. Morris, K. Kanatani, and T. Kanade. Uncertainty modeling for optimal structure from motion. In *IEEE Workshop on Vision Algorithms: Theory and Practice*, pages 33–40, Corfu, Greece, Sep. 1999.

[67] K. Murty. Solving the fixed charge problem by ranking the extreme points. *Operations Research*, 17:268–279, 1969.

[68] S. Negahdaripour and B. Horn. Direct passive navigation. *IEEE Trans. PAMI*, 9(1):168–176, January 1987.

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

[70] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans. PAMI*, 7(2):139–154, March 1985.

[71] T. Papadopoulo and O. Faugeras. Computing structure and motion of general 3d rigid curves from monocular sequences of perspective images. Technical Report RR-2765, INRIA, 1995.

[72] P. Pardalos. Enumerative techniques for solving some nonconvex global optimization problems. Technical Report CS–86–33, Dept. Computer Science, The Pennsylvania State University, Nov. 1986.

[73] P. Pardalos, F. Rendl, and H. Wolkowicz. The quadratic assignment problem: a survey and recent developments. In *Proc. of the DIMACS Workshop on Quadratic Assignment and Related Problems*, pages 1–42, May 1993.

[74] P. Pardalos and J. Rosen. Constrained global optimization: References. Technical Report CS–86–02, Dept. Computer Science, The Pennsylvania State University, Jan. 1986.

[75] P. Pardalos and J. Rosen. Methods for global concave minimization: A bibliographic survey. *SIAM Review*, 28(3):367–379, Sep 1986.

[76] P. Pardalos and J. Rosen. *Constrained Global Optimization: Algorithms and Applications*. Springer-Verlag, 1987.

[77] M. Pilu. A direct method for stereo correspondence based on singular value decomposition. In *Proc. CVPR*, Puerto Rico, 1997.

[78] C. Poelman and T. Kanade. A paraperspective factorization method for shape and recovery. In *Proc. ECCV*, volume 2, pages 97–108, Stockholm, 1994.

[79] L. Quan and T. Kanade. A factorization method for affine structure from line correspondences. In *Proc. CVPR*, pages 803–808, San Francisco, June 1996.

[80] A. Rangarajan, H. Chui, and F. Bookstein. The softassign procrustes matching algorithm. In *Proc. ICIAP*, 1997.

[81] K. Rangarajan and M. Shah. Establishing motion correspondence. *CVGIP: Image Understanding*, 54(1):56–73, 1991.

[82] J. Rosen and P. Pardalos. Global minimization of large-scale constrained concave quadratic problems by separable programming. *Mathematical Programming*, 34:163–174, 1986.

[83] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proc. ICCV*, 1997.

[84] G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two patterns. In *Proc. Royal Society of London, B*, volume 244, pages 21–26, 1991.

[85] I. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Trans. PAMI*, 9(1):56–73, 1987.

[86] L. Shapiro and J. Brady. Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, 10(5):283–288, June 1992.

[87] A. Shashua. Correspondence and affine shape from two orthographic views. Technical Report AIM 1327, MIT, AI Lab, 1991.

[88] A. Shashua. Trilinear tensor: the fundamental construct of multi-view geometry and its applications. In *Int. Workshop on Algebraic Frames for The Perception Action Cycle*, Kiel, Germany, September 1997.

[89] C. Silva. *3D Motion and Dense Structure Estimation: Representations for Visual Perception and the Interpretation of Occlusions.* PhD thesis, Instituto Superior Técnico, 2001.

[90] C. Silva and J. Santos-Victor. Intrinsic images for dense stereo matching with occlusions. In *Proc. ECCV*, pages 100–114, 2000.

[91] J. Starink and E. Backer. Finding point correspondences using simulated annealing. *Pattern Recognition*, 28(2):231–240, 1995.

[92] G. Stein and A. Shashua. Direct methods for estimation of structure and motion from three views. Technical Report AIM 1594, MIT, AI Lab, 1996.

[93] E. Stuart and A. Phillips. Fast approximate solution of large-scale constrained global optimization problems. Technical Report TR 88–9, Computer Science Dept., Institute of Technology, University of Minnesota, Jan. 1988.

[94] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *ECCV*, volume II, pages 709–720, 1996.

[95] G. Sudhir, S. Banerjee, and A. Zisserman. Finding point correspondences in motion sequences preserving affine structure. *CVIU*, 68(2):237–246, November 1997.

[96] C. Tomasi. *Shape and Motion from Image Streams: a Factorization Method*. PhD thesis, Carnegie Mellon University, Pittsburgh, 1991.

[97] C. Tomasi and T. Kanade. Shape from motion from image streams under orthography: a factorization method. *IJCV*, 9(2):137–154, November 1992.

[98] C. Tomasi and R. Manduchi. Stereo without search. In *Proc. ECCV*, volume I, pages 452–465, April 1996.

[99] C. Tomasi and R. Manduchi. Stereo matching as a nearest-neighbour problem. *IEEE Trans. PAMI*, 20(3):333–340, March 1998.

[100] A. Torn and A. Zilinskas. *Global Optimization*. Springer-Verlag, 1989.

[101] P. Torr. *Motion Segmentation and Outlier Detection*. PhD thesis, Dept. Engineering Science, U. Oxford, 1995.

[102] H. Tui. Concave programming under linear constraints. *Soviet Mathematics*, 5:1437–1440, 1964.

[103] V. Visweswaran and C. Floudas. Computational results for an efficient implementation of the gop algorithm and its variants. In *Global Optimization in Chemical Engineering*, pages 111–153. Kluwer, 1996.

[104] V. Visweswaran and C. Floudas. New formulations and branching strategies for the gop algorithm. In *Global Optimization in Chemical Engineering*, pages 75–100. Kluwer, 1996.

[105] S. Wallace. Pivoting rules and redundancy schemes. *BIT*, 25:274–280, 1985.

[106] J. Weng, N. Ahuja, and T. Huang. Matching two perspective views. *IEEE Trans. PAMI*, 14(8):806–825, 1992.

[107] M. Wu and J. Leou. A bipartite matching approach to feature correspondence in stereo vision. *Pattern Recognition Letters*, 16:23–31, January 1995.

[108] Z. Zhang. Iterative point matching for registration of free-form curves. Technical Report RR-1658, INRIA, 1992.

[109] Z. Zhang. Le problème de la mise en correspondance: L'état de l'art. Technical Report RR-2146, INRIA, 1993.

[110] Z. Zhang. Determining the epipolar geometry and its uncertainty - a review. *IJCV*, 27(2):161–195, 1998.

[111] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Technical Report RR-2273, INRIA, 1994.

[112] P. Zörnig. *Degeneracy Graphs and Simplex Cycling*. Springer-Verlag, 1991.