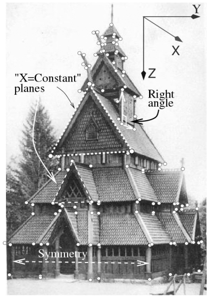
Universidade Técnica de Lisboa
Instituto Superior Técnico



Maximum Likelihood 3D Reconstruction From One or More Uncalibrated Views Under Geometric Constraints

Etienne Grossmann

Dissertação para a obtenção do Grau de Doutor em
Engenharia Electrotécnica e de Computadores

Orientador: Doutor José Alberto Rosado dos Santos Victor

Júri:

Presidente:	Reitor da Universidade Técnica de Lisboa
Vogais:	Doutor Rachid Deriche
	Doutor Jorge Manuel Miranda Dias
	Doutor João José dos Santos Sentieiro
	Doutor Mário Alexandre Teles de Figueiredo
	Doutor José Alberto Rosado dos Santos Victor
	Doutor Pedro Manuel Quintas Aguiar

Outubro de 2002

RESUMO

Considera-se o problema da reconstrução tridimensional obtida a partir de uma ou mais imagens, quando as projecções perspectivas dos pontos 3D de interesse são conhecidas, e também algumas propriedades geométricas, tal como planaridade, colinearidade, simetria, ângulos entre direcções etc. Assim, o método proposto aplica-se sobretudo à reconstrução de ambientes e objectos artificiais, onde estas propriedades geométricas são comuns.

O método decompõe-se em duas fases. Na primeira, o problema da reconstrução é transformado num problema de álgebra linear cujas soluções são identificadas com as do problema inicial. Além disso, examinando a dimensão do espaço das soluções do problema linear, é possível determinar se os dados são suficientes para definir uma reconstrução única.

Na segunda fase, a reconstrução de máxima verosimilhança é obtida. O problema de reconstrução é transformado num problema de optimização sem restrições em que é usada uma parametrização diferenciável dos pontos tridimensionais sujeitos às restrições geométricas.

Estas duas técnicas combinam-se num método de reconstrução que apresenta melhorias respeito a métodos previamente publicados, quer por oferecer uma grande flexibilidade de uso, quer por dar um resultado cuja precisão pode ser caracterizada estatisticamente. O método é avaliado usando dados sintéticos e reais.

PALAVRAS-CHAVE

Visão por computadores, fotogrametria, reconstrução 3D, reconstrução a partir de uma imagem, máxima verosimilhança, restrições geométricas.

ABSTRACT

We consider the problem of tridimensional reconstruction obtained from one or more images, when the 2D perspective projections of 3D points of interest are available, together with some geometric properties, such as planarities, alignments, symmetries, known angles between directions etc. Because these geometric properties occur mainly in man-made environments and objects, the presented method applies mostly to these cases.

The method has two phases. In the first, the reconstruction problem is transformed into one of linear algebra, and the solutions to the initial problem are identified with that of the second. Thus, examining the dimension the space of solutions allows to determine whether the provided information is sufficient to uniquely define a reconstruction.

In the second phase, the maximum likelihood reconstruction is obtained. The reconstruction problem is transformed into a problem of unconstrained optimization by using a differential parameterization of the 3D points subject to geometric constrained.

These two techniques combine into a reconstruction method that improves over the current state-of-the-art by offering a great flexibility of use and by providing a reconstruction that is statistically characterized. The method is benchmarked using synthetic and real-world data.

KEYWORDS

Computer vision, photogrammetry, 3D reconstruction, single-view reconstruction, maximum likelihood, geometric constraints.

Agradecimentos

Chegou o fim da tese, chegou a hora dos agradecimentos. Peço já desculpa aos que vou esquecer. Para começar, eu queria agradecer aos meus pais, e à restante família, irmã, irmão, sobrinhos ... Também não posso esquecer o apoio que recebi no mundo académico, dos Prof. José Santos Victor, João Sentieiro, Giulio Sandini, Vitor Barroso e Isabel Ribeiro, e também de Filomena Viegas e Nuno Sena. Queria agradecer a todos os amigos que eu encontrei em Lisboa, no ISR, a todos os membros do Vislab e não só, César, Cecília, Alexandre e Lena, Gaspar, Nuno, Maria João, Patricia, Cecília, Carlos, João Xavier, António, João Pedro, Carlos Carreira, João Maciel, João Paulo, Raquel, Zedú, Claudia, Sjoerd, Sónia, Roger, Anastácia, Maria, Klaudia, Lenildo, Gauthier, Matteo, Marco, Carlo, Diego, Javier, Eval, Niall, Sajjad, Jorge, Manuel, Ricardo, Freek, Vitor, Francisco, Pedro, Miloud, Loïc, Lionel, Luis, Manuel, Pedro, Ángela, Sébastien, Dejan, Dusan, Rodrigo, Marko, Emanuel, Reza e Franck. Se não há dúvida que esqueci alguém da pletora que precede, que frequentei durante os últimos anos, é ainda mais certo que eu vou esquecer ainda mais pessoas dentre os amigos de Marselha, Snoop, Sabine, Laurent, Gilles, Delphine, Marie-Paule, Francette, Yves, Scotto, Manu, Lionel, Hussin, Sophie-Charlotte, Guillhem, Yvon, Antoine, Béa, Momo, Serge, Sylvie, Ludovic, et qui d'autre? Também lá, no mundo académico, há quem -acho eu- influenciou o meu trabalho, por exemplo Henri Chaix e Thierry Viéville. Há também muitas pessoas que eu nunca vi, que contribuíram a esta tese, tal como os que disponibilizaram software que foi de grande ajuda na escrita, nas computações e na visualização de dados. Entre eles, eu queria agradecer, misturando pessoas e projectos, LyX, LaTeX, Debian, John W. Eaton, Paul Kienzle, John Stewart, Octave e FreeWRL.

Finalmente, tenho que agradecer às instituições sem as quais não teria sido possível realizar este trabalho, o Instituto de Sistemas e Robótica que me acolheu, o Instituto Superior Técnico de que estou estudante, ao programa COPERNICUS que financou o projecto 960174-VIRTUOUS que apoiou o inicio da tese e a Fundação para a Ciência e Tecnologia que me concedeu a bolsa PRAXIS XXI Grant BD / 19594 / 99.

Contents

Chapter 1

Introduction

In a picture taken by a camera, depth information about the scene is lost, but most human beings can interpret the image and somehow recover the original shape. In contrast, we are presently unable of programming a computer to do the same thing, except in limited cases. For the time being, the human visual system is in most respects superior to computer vision algorithms, which explains why many computer vision systems still rely on a human operator to do part of the work.

By observing an image like that in Figure 1.1 (left), a human being gets a good feeling of the shape of the viewed object, but cannot determine precise metric properties of the scene, such as angles or ratios of lengths. For example, it is hard to accurately determine the ratio of the height to the width of the building. Tools from geometry, on the other hand, allow us to relate 2D measurements to 3D points and, using some algebra and estimation theory, it is possible to calculate some information about the original 3D object. A computer is best adapted to implement these numerical calculations. For example, Figure 1.1 (right) shows a model computed from the image on the left, with the help of a human operator. In this model, measurements of angles and ratios of distances can be done, and the precision of the obtained figures can be estimated.

This thesis proposes a methodology to recover the 3D structure of objects from one or more images, exploiting auxiliary geometric information provided by a user. We combine the advantages of a human operator and that of a computer and develop a method for measuring precisely 3D objects observed in one or more 2D images. We improve over the state-of-the art by using geometric properties of scenes that were not exploited previously, such as the symmetry in certain objects. Also, we arguably claim that the available information is treated in a more appropriate way, which results not only in effective algorithms, but also allows to give a precise probabilistic meaning to the estimated quantities.

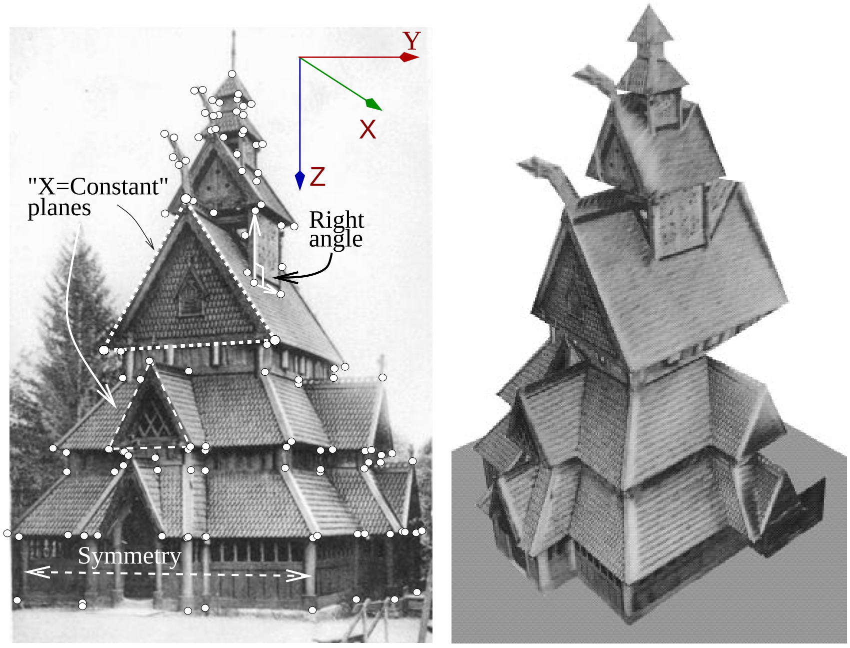


Figure 1.1: **Left:** A human being observing this picture is able to decompose this scene in triangles, and other planar surfaces and in prisms, parallelograms and pyramids.
Typical man-made scenes are rich in planarities, right angles and symmetry : the outlined triangular façades lie on vertical planes, the panel on the right side is rectangular, and the building has vertical planes of symmetry passing through the topmost point.
Right: By using 2D points (white dots) localized in the image at left and some geometric information known a priori, it is possible to compute this reconstruction. The positions of the 3D points (white dots at left) are estimated, and textured planar surfaces are used to improve the visual aspect.

The "structured scenes" that we consider are sets of 3D points that verify certain geometric properties such as planarity, parallelism and other forms of geometric regularity. The reconstruction method that we develop is well adapted architectural scenes, but others can be treated, as long as sufficient geometric "structure" can be found. These geometric properties are easily identified in an image by a human being, whereas a computer cannot, as of today, be programmed to do the same.

The task required from the human operator, the "user", thus consists in defining a polyhedral representation of the reconstructed object and giving some information about the orientation of the facets. For the user, it is relatively simple to choose what part of the scene should be reconstructed, select a meaningful set of planar surfaces in the scene and determine the orientation of these facets.

The following example gives a more explicit idea of the information provided by the user. First, he or she will identify the 2D projections of 3D points of interest, which form the edges of a polyhedral object that approximates the scene. In Figure 1.1 (left), these points are identified by white dots. Then, some "X", "Y" and "Z" directions, and possibly others are chosen, which are "aligned" with the features of the object. For example, the "Z" direction typically is taken to be vertical, while the "X" and "Y" directions are horizontal, perpendicular to each other, and parallel to some edges of the object. The user will then indicate some geometric properties involving the identified points and these directions. For example, the user could have indicated that the two triangles outlined in Figure 1.1(left) are contained in planes orthogonal to the "X" direction. Also, it is clear that the object is symmetric with respect to a vertical plane orthogonal to the "Y" direction, passing through the top of the object. The user could indicate this property by identifying a point that belongs to the plane of symmetry and pairs of points that are symmetric with respect to that plane. As one sees, a human operator can easily identify the desired information.

Once this collection of 2D points and geometric properties are given, the problem of 3D reconstruction consists in finding 3D points such that their perspective projection (Figure 1.2) best coincides with the 2D points. This is a problem of geometry and, because there are errors in the observations, one of probability. The bulk of this thesis is concerned with finding an appropriate solution.

Before going into the details, we show that solving the proposed problem has is some practical utility. Then, we examine the previously published solutions, and put in evidence strong and weak points. In the last section of this chapter, we outline the method that is the subject of this thesis

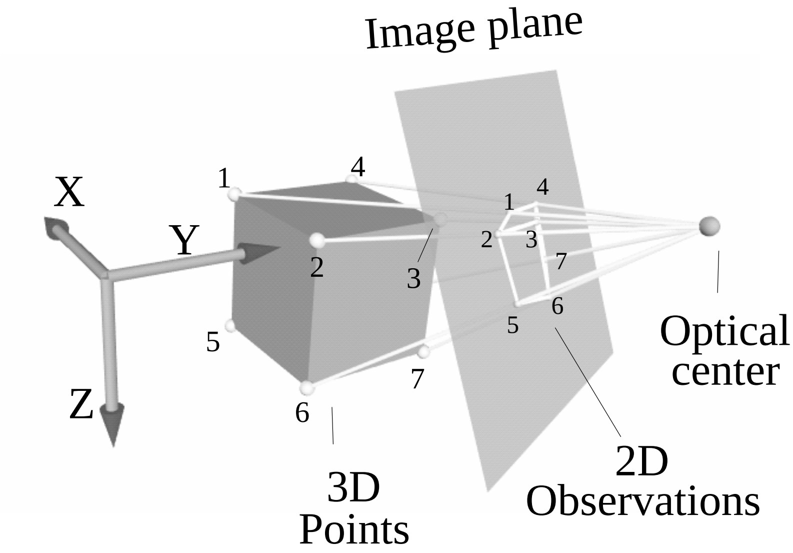


Figure 1.2: **Geometry of the Perspective Projection model.** The observed 2D points are obtained by perspective projection. That is, the original 3D point lies on the line defined by the optical center and the observed 2D point. In practice, however, small errors occur in the observations, and the three points are not exactly aligned.

and briefly explain how it addresses shortcomings of other methods.

1.1 Motivation

Man-made environments and images thereof are very common, and there are many situations in which metric information about the original scene is sought, but only one image is available, so that a classical reconstruction method [55] cannot be used. Having metric information allows to perform measurements of 3D objects using images.

Architecture, biometrics and forensic science are possible recipients of such measurement methods [13]. Metric information is also needed to obtain virtual models of buildings or indoor scenes, which are easier to visualize than simple measurements. Urbanists, architects, real-estate companies, movie-makers and producers of computer games could all be interested in obtaining models of buildings [19] from few images. In some cases, buildings that have disappeared can be modeled from as little as a single image, for example, an old photograph or a painting [13].

Other types of scenes that verify some geometric regularity can also be analyzed. For example, [57, 69] analyze images of sports events. In [57], the 3D posture of a sportsman is reconstructed from a single image, by exploiting known ratios of lengths in human limbs. By generating novel views of the sportsman, a better understanding of the situation on the field of game can then be gained. In [57], the authors estimate, using a probabilistic framework, the position of a soccer ball in the field, and determine whether a goal has or not been scored.

Although some of these applications are accessible to general reconstruction methods [71, 55], others, most notably those in which a single image is available, are possible only because some techniques were developed that exploit specific geometric properties of the scenes that are being treated.

Also, in the case of architectural scenes, it has been noticed [6] that the quality of the obtained 3D measurements can be greatly improved if some geometric properties of the scene are taken into account, such as planarities, orthogonalities, parallelism etc. Because classical method do not take advantage of this fact, new techniques have been proposed and are still being developed.

Finally, the evolution of computers seems an invitation to reconstruction systems in which part of the information is provided by a user, while numerical computations are executed automatically. Not only has the computational power of computers increased, the interface with the user has also become much more intuitive, allowing him or her to enter information and commands more conveniently.

The conjunction of many applications, new methods and ergonomic user interfaces has rendered reconstruction methods accessible and attractive beyond the circle of researchers and students who developed them.

Having given motivation to solve the problem of reconstructing structured scenes, we now see how previous work has addressed it, and then indicate how this thesis improves on the actual state-of-the-art.

1.2 Related work on the reconstruction of structured scenes

The methods that can be found in the literature can be roughly classified as “model-based” (or “CAD-based” [75]) and “constraint-based”. In the former, a model object is fitted to image features [49, 20, 42], while the latter [13, 48, 60, 62, 67] rely on geometric properties, such as planarity, alignment etc, to obtain a reconstruction. Some techniques share aspects of both constraint and model-based methods [76].

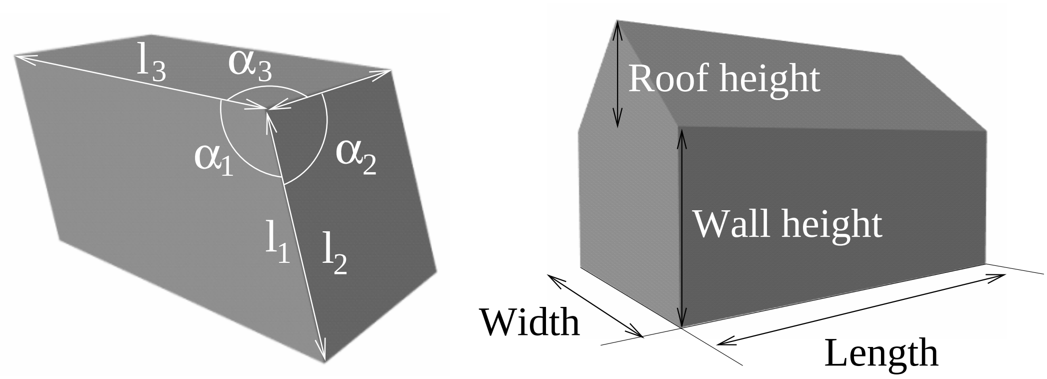


Figure 1.3: Parameterized models provide a convenient way of representing common shapes. **Left:** A parallelogram can be represented by the lengths l_1, l_2, l_3, and the angles α_1, α_2 and α_2. **Right:** A house-like object can be represented by the four lengths shown in the figure.

1.2.1 Model-based approaches

Rectangles, parallelograms, circles, prisms are commonly found in man-made environments and many objects can be modeled as a combination of these shapes. The main idea behind model-based approaches is to obtain the reconstruction as an assemblage of "primitive shapes" that best fits the image data. Typically, the user specifies one [41, 69] or more models [49, 19] and localizes them approximately in the image. The model is then fitted to the image.

One reason for which it is convenient to use such combinations of primitive shapes is that each one can be conveniently characterized by a small collection of numbers. For example, a rectangle parallelogram is uniquely defined by three lengths, or a general parallelogram, by three lengths and three angles (Figure 1.3, left). Other objects such as the house in Fig. (1.3, right) can also be concisely represented. This representation is relatively simple in mathematical terms and is easy to use in computer algorithms.

In methods that use an assemblage of many shapes, the relative position of objects can also be defined conveniently. For example, it is possible to say that one shape is "on top of" another, or that they lie on the same plane etc. In mathematical terms, the referentials in which different objects are represented are subject to some constraints, and parameterized accordingly.

As a result of this modeling process, the collection of 3D points that define the object, for example some of its edges or corners, are represented as a function of some parameters. In other terms, the object is parameterized by a "shape function" :

$$\mathbf{X} = \mathcal{M}\left(\Theta\right), \tag{1.1}$$

where $\mathbf{X}$ is a vector holding all the coordinates of 3D points, Θ is a real-valued vector holding all the parameters and $\mathcal{M}$ is the function that associates the 3D points to the parameters.

The reconstruction is obtained by minimizing a cost function that measures discrepancy between the 2D observations and the projections of the model :

$$\text{Cost} = \sum_{i=1}^{N} \left\| \mathbf{x}_i - \mathcal{P}_i \left(\mathcal{M} \left(\Theta \right), R, \mathbf{T}, K \right) \right\|^2 \tag{1.2}$$

where $\mathbf{x}_1, ..., \mathbf{x}_N$ are the image observations, $\mathcal{P}_i$ is a "projection" function and R, $\mathbf{T}$ and K represent the orientation(s), position(s) and calibration(s) of the camera(s). Of course, the actual cost function is usually not written in this exact way, but it has similar properties.

One of the most useful characteristic of this cost function is its differentiability, which facilitates the localization of its minimum [56]. For this property to hold, it is necessary that the function in Eq. (1.1) be differentiable, which can easily be guaranteed by construction. Another noteworthy characteristic of the cost function in Eq. (1.2) is that it may involve observations (the $\mathbf{x}_i$) in many images. As a consequence, single and multi-view datasets can be treated in similar ways. Finally, the cost function can often be given a probabilistic interpretation, so that the reconstruction is actually a maximum likelihood estimator and it is possible to analyze its precision [34]. From this exposition, is clear that model-based methods are well-behaved mathematically and that they present many interesting properties.

The main limitation of model-based methods is that they can only reconstruct scenes for which a model is available, or which can be decomposed into simpler shapes. In order to extend the generality of the method, there are basically two solutions. One is to add primitive shapes, which requires that the producer of the method works out the details of some "shape functions" Eq. (1.1). Another is to decompose the scene in many objects, which requires more work from the user, and is not always possible. Model-based methods are thus well adapted to some sorts of common objects, while some others require more work to be reconstructed.

Model-based methods have also been developed in the photogrammetry community, where they are called "CAD-based", because they are often integrated to CAD software [75]. Also, in the context of artificial intelligence, model-based methods of one form or another are commonly found in fully automatic reconstruction and recognition systems [8].

In summary, the main characteristic of model-based methods is that they require the reconstructed scene to be decomposed into some geometric models, which somewhat limits their field of application. Past this disadvantage, they are adapted to least-squares estimation and work with single and multiple views alike.

1.2.2 Constraint-based methods

Other methods, which we call “constraint-based” rely on more generic properties of the scene, such as planarity, parallelism, known angles etc. A great variety of such methods have been developed for the single- [57, 62, 13, 67, 76] and multi-view [5, 11, 68, 65, 3] cases, and we now review their underlying principles and the characteristics that distinguish one from another.

Whereas model-based methods rely on the same principles in the single and multi-view cases, this is most often not the case in constraint-based methods. Indeed, the multi-view methods are usually derived from general reconstruction methods, while single-view methods are obtained by exploiting known geometric properties in a more direct manner. We begin this section by examining how constraint-based methods differ in their use of geometric information.

Then, these methods are gauged according to other important characteristics, the first of which being whether the obtained reconstruction actually verifies the geometric constraints specified by the user. The second discriminating criterion is whether the method checks whether the user gave input data that is coherent and sufficient to define a unique reconstruction.

We will see below that vanishing point estimation and camera calibration play a crucial role in constraint-based methods, so that this section ends with a discussion on these topics.

i) A-priori geometric information

All constraint-based methods [5, 3, 65, 68, 11, 62, 13, 67] use known planarities, and in some cases [68], this is the only used geometric property. Also, known angles are often used. Finally, there have been attempts at using more general polynomial constraints [3]. We first consider how multi-view methods have exploited geometric information and then continue with single-view methods.

Multiview constraint-based methods share many aspects of general reconstruction methods [71, 55]. Like these, they take as input the coordinates of 2D points that have been tracked along many images. However, the input data also includes some geometric properties of the reconstructed scene. The computational procedure of the constraint-based methods can often be likened to a general method which has been modified so that geometric information be taken into account.

In the multiple-view case, Szeliski and Torr [68] propose many ad-hoc ways to exploit planarities, e.g. to reduce the number of observations needed to obtain a reconstruction. Also, a specialized

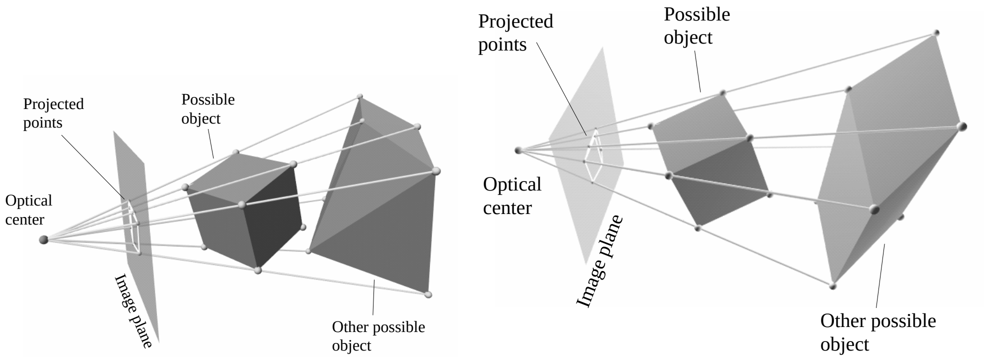


Figure 1.4: **Left:** If only 2D points are available and no other assumption is made, there is no unique reconstruction : the vertices of the cube and the polyhedron both project to the same 2D points. Planarity relations that exist for vertices of the cube are not verified in the object at the right.
Right: If 2D points and planarity information are available and no other assumption is made, the reconstruction is still not uniquely defined : the vertices of the two objects project to the same 2D points. The vertices of the rightmost object verifies the same planarity relations as that of the cube, but parallelism between some segments is not preserved.

optimization method is proposed, which successively estimates the camera orientation and position, estimates the 3D points and projects these points onto planes. This last step consists in fitting a plane to points and replacing them by their projection on the plane. However, it is not clear what happens when points belong to more than one plane. The authors report that known planarities alone "do not significantly reduce the reconstruction error" [68, p. 182].

Sparr [65], still with multiple views, also proposes an iterative scheme in which reconstruction steps alternate with projection steps that enforce known planarities and parallelisms. However, it is again unclear whether all these properties are always verified by the final reconstruction.

Boufama et al [5] assume that not only some planarity relations are known, but also that the orientation of the planes is known and that the normal to each plane is one of three orthogonal directions, labeled "X", "Y" and "Z". Likewise, only alignments that are parallel to one of these directions are considered. These geometric properties are used within a least-squares approach in which the reconstruction problem is transformed in an optimization problem. While, without the knowledge of orthogonality relations, it would only be possible to obtain a projective reconstruction [24], this system obtains a Euclidean reconstruction from two uncalibrated views.

There are thus many ways of exploiting known geometric properties when performing multi-view reconstruction.

From a single-view, the information that is available is fundamentally different. There is no such thing as "tracking" of points along many view, and there is no way to estimate the depth of points from parallax. It is clear that, if only 2D points are available, then the depths of 3D points may be arbitrarily set. Figure 1.4 (left) shows how two distinct object can project to the same image. When a single view is available, it is thus necessary to use other clues to obtain a reconstruction.

Knowing some planarity relations reduces the ambiguity, but is usually insufficient to define a unique reconstruction [37] (Figure 1.4, right). Parallelism between planes can suffice to define a reconstruction that is unique up to an affine transformation. Finally, known angles between plane normals can be exploited to obtain a reconstruction up to a Euclidean transformation. In the single view-case, it is thus necessary that the user gives a priori knowledge of, at least, parallelism and preferably also angles. This contrasts with some multi-view methods [68, 65, 3] which need less extended information.

The most commonly encountered situation, in the single-view case, is when some plane normals are known to form three orthogonal directions [5, 20, 11, 67]. It is then possible to fix these directions to some canonical values, such as [1; 0; 0], [0; 1; 0] and [0; 0; 1], and determine the coordinates of these vectors in the camera frame [13, 62]. This is usually done by using the vanishing point of the plane normals or the vanishing line of the planes [9]. We now explain how the knowledge of plane normals can be used.

It is well known that points $\mathbf{X}_1, ..., \mathbf{X}_N \in \mathbb{R}^3$ belong to a plane with normal $\mathbf{v} \in \mathbb{R}^3$ if and only if there exists a number $d \in \mathbb{R}$ such that, for all i in $1..N$:

$$\mathbf{v}^\top \mathbf{X}_i = d. \tag{1.3}$$

If $\mathbf{v}$ is known, then one has N linear equations that constrain the unknowns $\mathbf{X}_1, ..., \mathbf{X}_N$ and d. In contrast, if $\mathbf{v}$ were unknown, the constraint on the unknowns $\mathbf{X}_1, ..., \mathbf{X}_N$, $\mathbf{v}$ and d would be *bilinear* and thus much harder to use than a linear equation.

In a man-made scene, the number of directions that are needed to describe the scene is typically quite small. For example, three mutually orthogonal directions are sufficient to express the properties of the building in Figure 1.1. Although other directions are present -the slanted edges of the roofs- the three directions "X", "Y", "Z" are sufficient for the purpose of describing the properties of the object in such a way that a reconstruction can be obtained. Because of the special role played by these directions, we will call them the "*dominant directions*" of the scene.

Most single-view constraint-based methods obtain the reconstruction [13, 11, 62, 67] by combining many planarities with the information given by 2D points.

Shum et al [62, 63] solve a linear system consisting of planarity constraints obtained from Eq. (1.3) and constraints obtained from the 2D observations. The resulting system is overconstrained and the user is asked to choose some constraints that will be exactly verified, while those remaining will be solved in the least-squares sense.

Sturm and Maybank [67, 66] use a different system of linear equations, that involves the "height" of each 3D plane (variable d in Eq. (1.3)) and the distance from each point to the optical center (its "depth"). A first reconstruction is obtained by solving this system. Other points, plane heights and plane normals can subsequently be estimated by combining 2D information and previously estimated quantities.

In Criminisi et al [17, 13, 14, 15, 16], three independent directions are used, two being parallel to a "reference" plane. Ratios of lengths of segments parallel to one of the directions within the plane can be measured, as well as ratios of lengths of segments parallel to the third direction and originating on the plane. By using different reference planes, and assembling the measurements together, a reconstruction is obtained. With this method, the scene must be decomposable into parallel planes linked together by segments, which somewhat restricts the application of the method. Criminisi's method for reconstruction is thus based on assembling measurements taken along different directions.

As can be seen from these three examples, there exist many ways to obtain a reconstruction from 2D observations, geometric information and known plane normals. There is a certain number of characteristics that can be taken into account when comparing these methods. For example, [62] obtain the whole reconstruction at once, whereas [67] build a first reconstruction to which points are later added, and [13] obtain a reconstruction by combining measurements taken along different 3D directions. Also, [67] can estimate new plane normals during the reconstruction process, while [13, 62] require that all normals be estimated beforehand. Note, however, that all methods, even [67], require that initially, at least three independent but not necessarily orthogonal 3D directions be known. All these methods are thus dependent on a previous step in which vanishing points (and maybe calibration) are estimated, so that some plane normals can be considered as known.

The single-view algorithms that we have considered so far exploit known planarities, parallelism and orthogonalities. Other types of constraints have been used to a lesser extent, which we now

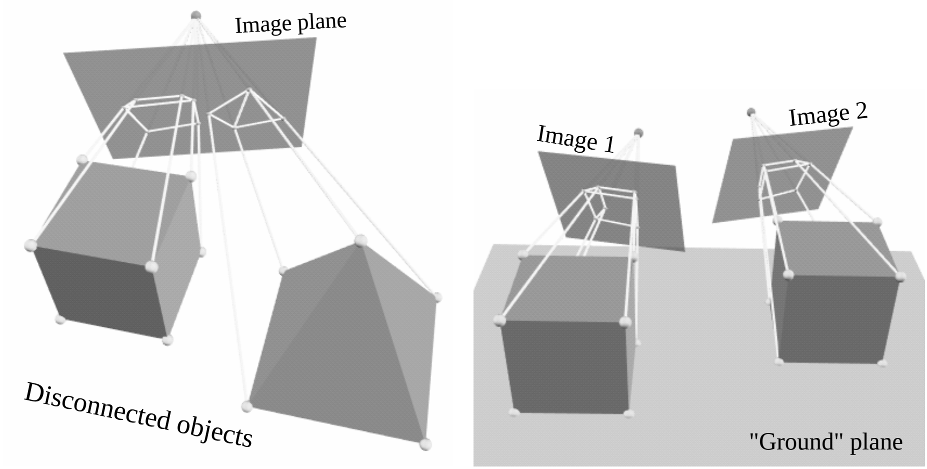


Figure 1.5: **Left:** The relative scale of the two objects cannot be determined unless one assumes e.g. that they lie on a same “ground” plane.
Right: The relative position of the two objects-camera pairs cannot be determined, despite both objects being known to lie on the same ground plane.

detail.

First, since a reconstruction can only be obtained up to a translation, rotation and scaling, some point coordinates can be arbitrarily fixed, which reduces the number of unknowns to be estimated [62] and simplifies the analytical study of the problem [13]. These practices are mostly done for convenience and, although they may actually influence the outcome of the method, they do not change the capabilities of the method.

In the multi-view context, Bondyfalat et al [4, 3] propose to use polynomial equations in the coordinates of the 3D points to express more general geometric properties. In this way, planarity can be expressed even if the plane normal is not known a priori, and other properties may be used as well, e.g. known distances or ratios of distances. The method relies on manipulating polynomials symbolically, eliminating variables until a subset thereof is left that parameterizes the whole reconstructed object. The heavy computation that may be needed constitutes a serious drawback. Maybe for this reason, only planarity, alignment and orthogonality relations are used [3, p. 227]. Although this method shows some promise, it appears impractical in its present state.

In summary, planarity, parallelism and orthogonality are the most important properties used in constraint-based methods. Others can and have been used, but in a more limited fashion.

ii) Sufficient user information

An issue which has received very little attention is that of determining whether the geometric information provided by the user is sufficient to define a unique reconstruction, and whether this information is coherent. This question does not concern model-based methods, which usually have an interactive user interface that guides the user so that sufficient information is always given. Also, multi-view constraint-based methods that are derived from general reconstruction algorithms [3, 65, 68] either do not require a "minimum" amount of geometric information, or it is trivial to determine whether this amount has been given [5]. This issue is thus mostly relevant to constraint-based single-view methods, which cannot obtain a unique reconstruction without a substantial amount geometric information.

Even in these methods, the issue has barely been addressed because they use a limited type of geometric information : planarities, parallelism and angles. In this case, the only possible consequence of insufficient information, illustrated in Figure 1.5 (left) is the presence of many "unconnected" objects whose reconstruction can be obtained independently, but whose relative scale cannot be determined. While some methods detect this situation [62, 67], it is usually implicitly assumed that a single object is being reconstructed.

Another possible problem occurs when the user mistakingly gives geometric constraints incorrectly. For example, if the user confuses "X" and "Y" and says that a "X=Constant" plane should be reconstructed as a "Y=constant" plane, then the reconstruction will be "flattened" and have an aspect totally different from that expected. This case is never mentioned in literature, where it is assumed that the geometric information given by the user is correct. On a related issue, [62] mention the possibility of the constraints being altogether impossible to satisfy.

It thus appears that, except Shum et al [62], previous work considers that the user provided sufficient and coherent information. Otherwise, (s)he shall get what (s)he deserves, whether a computer error or a strange-looking reconstruction. This attitude is justified for the previously published systems, in which the risk of an error occurring is either inexistent by design (model-based methods) or relatively small (constraint-based methods).

However, we shall see later in this thesis that, if more general types of geometric information are used, then the possibility increases that a constraint-based method will be given insufficient or incoherent information. This may happen for example when symmetry information is given, or when a single-view methods, that does not use matched points, is extended for the multi-view case. For example, Figure 1.5 (right) shows two objects viewed separately by two cameras, with no overlap, but with the knowledge that both cubes lie on the same plane. It is then possible to position arbitrarily the two camera-object pairs on the plane and even swap one pair to the other side of the plane. Only Shum et al[62] consider the extension of their method to the multi-view case, and require that some points be tracked between views, which eliminates the possibility of this ambiguity.

The notion of sufficient geometric information applies to whole datasets, rather than to any particular element. If we take the point of view of counting the number of "independent parameters" that define the reconstruction, and the number effectively given by the data, the difficulty appears when counting the number of "independent parameters" provided by the geometric information. For example, how many parameters are eliminated by the knowledge that a parallelepiped is as wide as it is high? As soon as more general geometric constraints are used, simple arguments are not satisfactory to determine whether a dataset is sufficient.

In summary, testing whether the data given by the user is sufficient and coherent has not been a major issue. Model-based methods appear immune to the problem, while constraint-based methods were limited to cases in which the question is easily resolved. However, if these methods are to be extended, more subtle ambiguities and incoherences may appear and it is important to be able to detect these cases.

iii) Compliant reconstruction

Whether the obtained reconstruction actually verifies the geometric constraints that the user specified is another important issue. Some methods, from the way the reconstruction is obtained, do not guarantee that all constraints are met. Moreover, this issue is rarely addressed in the literature, and authors usually do not explicitly state whether this may happen or not. One reason for this situation may be that, when the method succeeds, the reconstruction should be nearly compliant with all the constraints, so that, visually, one is not likely to note the discrepancy. In this section, we review some reconstruction methods according to their treatment of the question.

First, some methods always return a compliant reconstruction. This is the case, understandably, in model-based methods, where all 3D points belong to an object. This is also the case with some constraint-based [5, 13, 3] methods, either because the reconstruction is represented in such a way that all the constraints are always verified [3], or as a consequence of the reconstruction procedure

[13].

Many other methods [64, 62, 68, 67, 60] do not guarantee that all geometric constraints are verified by the reconstruction, for example the single-view methods [62, 67, 60] that work by solving an over-constrained system of equations obtained from the observations and from the a priori geometric knowledge, like Eq. (10). Because this system is over-constrained, it is solved in the least-squares sense, and the reconstruction does not necessarily verify all the geometric properties provided by the user. The problem is partially solved in [62], by asking the user to choose what constraints will be exactly verified[1] and which will only be approximately met.

Another case in which geometric constraints may not be verified is when the constraints are enforced one after the other, and enforcing one may break another. This happens in the multi-view system [11], where an unconstrained reconstruction is obtained, which is later made to approximately verify the geometric properties. In [64, 68], where reconstruction steps alternate with steps that enforce the geometric properties, it appears that enforcing one constraint may break another one, so that it is unclear whether the reconstruction is guaranteed to verify all the constraints.

Pose and calibration from a single view

The availability of camera calibration parameters is important for 3D reconstruction, because it determines the type of information that can be obtained about the observed scene.

When a camera is calibrated, the 2D pixel coordinates can be related to metric measurements, whereas otherwise pixel coordinates are given in an unknown referential of the image plane. In the first case, pixel coordinates can be likened to physical measurements. For example, one typically assumes that 2D coordinates are given in an orthogonal 2D frame centered on the principal point. Also, because the focal length is known, it is possible to measure the angle between 2D points given as pixel coordinates. In contrast, uncalibrated coordinates are given in a referential whose origin and axes are unknown. Not surprisingly, more 3D information can be obtained from calibrated 2D observations.

Without calibration, only an affine reconstruction is obtained from a single view. As a consequence, the angles between the vectors that form the basis in which the reconstruction is obtained are not known. It is still possible, however, to compare distance measurements taken along a given direction [57, 13]. When calibration is known, the reconstruction is obtained up to a rigid transformation and it is thus possible to measure angles and arbitrary lengths. It is thus important to

[1] These are called “hard” and “soft” constraints.

estimate the calibration of the camera, and fortunately, it is possible to do so in many cases.

Caprile and Torre [9] detail the relation between the vanishing points and the pose and calibration of the camera[2]. Despite being mainly concerned with stereo, that article clearly exposes the geometrical issues that arise and it is considered to be a stepping-stone in the problem of single-view calibration. In that article, it is shown that, from the vanishing points of three orthogonal 3D directions, at most three intrinsic parameters[3] can be estimated. The priority is given to the estimation of the parameters that are most likely to vary, the focal length and principal point, and it is shown that this task is impossible for some configurations of the camera.

Also, it is shown that, once the camera is calibrated, the vanishing point of a 3D direction can be identified with the coordinates, in the camera frame, of that direction. Thus, estimating the vanishing point of a 3D direction, is equivalent to estimating the 3D direction itself. For this reason, most reconstruction methods first estimate the vanishing points and then consider that the 3D directions are known, which simplifies the reconstruction problem, as explained in Section 1.2.2.

Because the method of Caprile and Torre appeared as the most practical solution, it was studied in detail and extended in various ways. For example, it has been shown how to use known angles that are not right angles and how to use vanishing points of more than three pairs of directions [35]. Kanatani [43] performs a sensitivity analysis of the estimation of the focal length, based on the sensitivity analysis of a vanishing point from 2D segments. An analytic expression yields an estimate of the focal length from a pair of vanishing points corresponding to orthogonal directions. This study gives insight on the influence of 3D line configuration on the precision, but unfortunately ignores the effects of unknown skew, aspect ratio and principal point. This line of research, however, requires the previous estimation of vanishing points of the 3D directions of interest.

Using vanishing points, although the most popular, is not the only way of obtaining calibration and camera orientation from a single image. A known 3D object has often been used [47], but, this is unpractical because most images do not contain an object for which measurements are available. A more practical approach uses any six vertices of a generic parallelepiped. Chen et al [10] use the projection of a parallelepiped whose angles are known, and give an analytic calibration method. Another methods is presented by Wilczkowiak et al [76], who show how known ratios of length of parallelepiped edges can be used. This method is thus able to exploit not only known angles, but

[2] This result seems to have been known earlier in the photogrammetry community : Hartley and Zisserman [35, p. 213] allude to the same result in [27]. Also, [73, p. 81] says that this result can be found in [78].

[3] Here, we only consider the focal length, skew, aspect ratio and principal point, which are defined on p. 25. The effects of radial and tangential distortion[47] are ignored.

also known ratios of distances, which cannot be used when calibrating from vanishing points.

However, these methods can hardly be considered to be more general than those based on vanishing points. First, they require that 2D projections of vertices of a parallelepiped be identified, which is usually sufficient to localize the vanishing points in the image. Moreover, it is often the case that the edges of a parallelogram are partially visible, enough to compute vanishing points, but that the vertices are occluded, so that the method cannot be applied directly. Although these methods show an interesting direction for research, methods based on vanishing points are nowadays more common.

1.2.3 Performance analysis

Having introduced the principles used in reconstruction methods, we now address the question of their precision. As can be expected, all the methods yield exact reconstructions if the input data is noiseless. But, in the presence of noise, the output will be corrupted. All methods are not equally sensitive to noise and it is important to determine how noise affects the quality.

Many multi-view methods claim that using geometric constraints improves the precision [3, 68], but it is exceptional that a publication gives some quantitative information on the error in the reconstruction. Most often, the quality of a method is assessed by showing some good-looking reconstructions [20, 3, 62, 67, 12]. While this shows that the method works in some cases, and suggests that it is not badly flawed, this is an insufficient warrant of quality.

A better approach to performance evaluation is benchmarking on datasets for which the ground-truth is known. This method gives insight that is useful, but does not extend beyond the setups that were tested. Despite being relatively easy to do, benchmarking is usually not performed at all [20, 62, 67], or only on parts of the algorithm [11].

Knowing the order of magnitude of the error in real-world cases, when the ground-truth is unknown, is of great importance for some applications. This is sometimes possible [57, 13], when one makes some assumptions on the error in the observations, and when one is able to propagate [34, 46] it to the output. For this last condition to hold, it must be possible to define, at least locally, the output as a differentiable function of the input. This is possible when the output is computed analytically from the input, and also when the output is defined as the minimum of a function, as in Eq. (1.2) on page 7. A method that is defined in either of these ways thus lends itself well to performance analysis. Having a differentiable functional relation between the input and the output is thus important for performance analysis.

In the following sections, we will see how performance analysis is done in model and constraint-based methods and discuss the assumptions that authors make about the error in the input data.

Model-based methods

Most model-based methods [49, 20, 41] are, in theory, well adapted to performance analysis, because they are obtained by minimizing a differentiable function and can be interpreted as maximum likelihood estimators, under the assumption of additive Gaussian independent errors in the observations.

In practice, however, we are aware of no model-based method developed in the computer vision community that actually does this error analysis. It should be noted that the “ShapeQuest” commercial package [1] claims optimality and gives an estimate of the precision, but we cannot confirm this because a precise description of the method is lacking.

Constraint-based methods

Constraint-based methods that obtain the reconstruction by estimating intermediate quantities (vanishing points etc) and solving systems of equations [62, 65, 68, 67] do not provide an obvious way to find a functional relation between the input and output. Although some methods [62, 67] may do least-squares fitting in the final stage of the algorithm, it is practically impossible to characterize the error in the reconstruction, because the output is not defined as a differentiable function of the input.

Some publications, however, give a big importance to precision. Reid and Zisserman [57] and Criminisi [13] take a probabilistic approach, and propagate the error from the observations to the intermediate quantities and to the reconstruction. But even then, [13, p. 100] the study is done under simplifying assumptions and does not concern the full reconstruction, but rather individual measurements.

Finally, multi-view constraint-based methods in which a well-defined cost function is optimized [5, 3] could be studied using the framework of estimation theory [34]. However the issue of precision was not addressed in a quantitative way in these publications.

The assumption of Gaussian errors in the observations

Finally, we give a remark on the probabilistic foundation of methods that adopt a least-squares approach. The usual justification for doing least-squares fitting is that the obtained reconstruction is then a maximum likelihood estimate, under the assumption that the errors in the observations are Gaussian, independent and identically distributed. Most often, this assumption, which we will call the "Gaussian hypothesis", is not mentioned, and the reader is left to guess that it is taken implicitly. If justifying the least-squares approach is seldom done, we are aware of only Torr et al [72] that actually justifies the Gaussian hypothesis itself, and in a slightly different context. In that article, the empirical probability density function of the residues in automatically matched points is studied, whereas in the present thesis, we mostly consider points that have been identified in a single image. Because of the scarceness of experimental evidence, the Gaussian assumption should be treated with some caution.

1.2.4 Summary of previous work

We now conclude our review of reconstruction methods that exploit the geometric properties of structured scenes. In summary, there are two main approaches to the reconstruction of structured scenes, the "constraint-based" and the "model-based". Model-based approaches can be used for reconstructing specific objects [49] that cannot be conveniently described by geometric properties alone. However, a model must be given for each type of reconstructed object, a disadvantage that is partially removed by combining many objects together [20]. Constraint-based approaches can treat more general scenes, but they are limited by the small collection of geometric properties on which they rely. Also, model-based techniques appear better adapted to precision analysis, because estimation theory can be used to characterize their output.

1.3 Method overview and original contributions

Having described the state of the art in the reconstruction of structured scenes, we now start presenting the method that is developed in this thesis. We first list the chronological steps of the reconstruction procedure, and outline the specificity of our approach :

Input: A user provides some 2D points and some geometric information, in the form of planarities, alignments and known angles. In addition, it is possible to indicate known ratios of distances between parallel planes. Many real-world scenes display this last type of property, but we are aware of no method that fully exploits it.

Vanishing points and calibration: Maximum likelihood estimates of the vanishing points are computed, from which, if a right trihedron is known, calibration is computed.

Geometric information as linear constraints: All the geometric information is summarized into a system of linear equations that will be solved exactly. The geometric properties given by the user are thus exactly verified.

Reconstruction as a linear problem: A linear system is built, whose rank determines whether the input data defines a unique reconstruction. If this is not the case, the algorithm stops, because the input data is insufficient. Otherwise, the 2D observations are used to form a third system of equations, and a non-optimal reconstruction is obtained by solving it in the least-squares sense.

Maximum likelihood reconstruction: Assuming that the errors in the input 2D points are independent, identically distributed and Gaussian, the likelihood function takes the form of a sum of squares. By defining a differentiable parameterization of the estimated quantities, the likelihood function is itself differentiable and the maximum likelihood estimate can be obtained by using a slightly modified Levenberg-Marquardt algorithm. Also, an approximation of the covariance matrix of the estimator is computed.

As can be seen, a constraint-based technique has been developed, which addresses the main shortcomings of these methods. Most notably the user may specify extended the geometric properties, the reconstruction is optimal in a well-defined probabilistic sense and its precision is approximately known. We now detail these points.

Extended geometric information : known ratios of distances between planes

The method we propose is more general than most other constraint-based methods[4] because it can exploit one more type of geometric constraints. Whereas constraint-based methods rely exclusively on planarity, alignment and known angles, we add the possibility of expressing other geometric properties, such as symmetry and periodicity, which are very common in real-world scenes.

[4] There exist some cases in which [67] may be used, but our method cannot, namely, situations in which they may estimate plane normals from previously reconstructed points, whereas our method needs all the normals to be initially estimated from vanishing points.

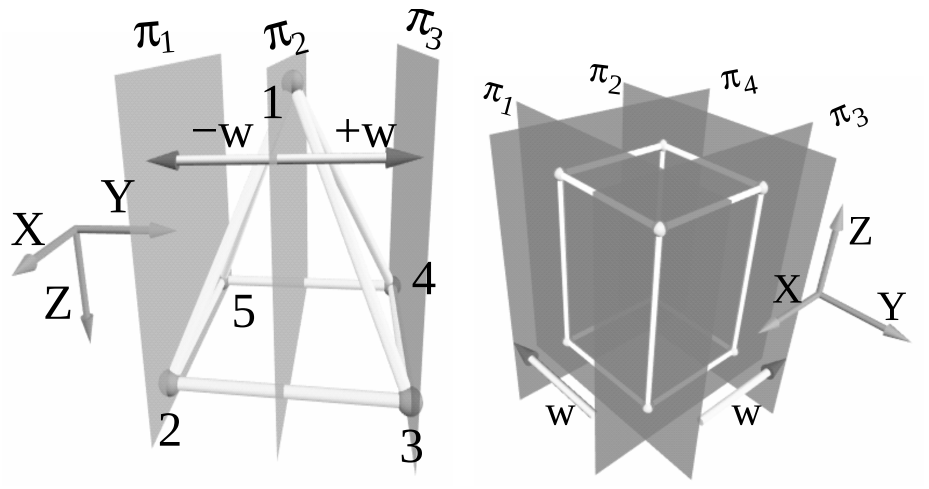


Figure 1.6: **Left:** In this 5-point dataset, it is possible to express the symmetry of the pyramid with respect to "Y=Constant" plane passing through point 1, by saying that the signed distances from planes π_2 to π_1 and from planes π_2 to π_3 are opposite one of another.
Right: It is also possible to indicate that the parallelepiped is as long as it is wide, by saying that the (signed) distances from π_1 to π_2 and from π_3 to π_4 are equal.

This is done by saying that the ratio of distances between some parallel planes are known. Figure 1.6 shows two examples of how this type of information can be used. It should be noted that the pyramid on the left could not be reconstructed from a single view if it were not for the knowledge of symmetry. Indeed, the topmost point is isolated from the others and its distance to the camera could be arbitrarily set, if it were not constrained to lie on a plane π_2 that is midway between π_1 and π_3. Other examples are given in Section. 6, where most real-world datasets could not be obtained by other published methods.

Known ratios of distances between parallel planes are easily expressed in mathematical terms, in a way similar to Eq. (1.3). As a consequence, all the geometric information is used in a homogeneous way, and this extra type of information does not imply any complication of the algorithm. Moreover, the uniform treatment given to the geometric information allows to devise a simple test on the sufficiency and coherence of the geometric information provided by the user.

Also, it has been noticed that multiple views can be treated in almost the same conditions as the single-view case. The presented method can thus be used with multiple views, even when no point is tracked between views.

In summary, this type of geometric information allows to extend the reach of constraint-based methods in multiple ways : more diverse objects can be reconstructed, and any number of views can be exploited without substantially changing the method.

Maximum likelihood estimation and performance analysis

From the review on performance evaluation, in Section 1.2.3, it appeared that constraint-based methods do not lend themselves well to either optimal estimation or sensitivity analysis. We show that this is not necessarily the case, by presenting a method for maximum likelihood reconstruction which also allows to determine the precision with which the reconstruction is obtained.

In summary, the principal claims about the method developed in this thesis can be listed in the following way :

1. Augmenting the field of application of constraint-based methods.
2. Justifying the least-squares approach by validating the Gaussian hypothesis.
3. Implementing a maximum likelihood estimator.
4. Benchmarking it on synthetic data.
5. Being able to estimate the precision of reconstructions obtained without ground truth.

The contribution of this thesis lies not only in the reconstruction method that is presented, but also in the validation of the methodology, which is done both by verifying that the least-squares approach is appropriate, and by benchmarking the reconstruction method.

1.3.1 Thesis structure

Having presented the problem that we wish to address, reviewed the current state-of-the-art, and indicated the improvements that we intend to bring, we may now go into the detail of our approach. To begin with, we introduce the vocabulary and the mathematical notation that will be used to describe the problem. Then, in Chapter 3, a sub-problem is treated, namely the estimation of vanishing points and of the calibration of the camera. With these estimates available, we develop in Chapter 4 an algebraic method of reconstruction. Then, in Chapter 5, the maximum likelihood reconstruction method is presented. Benchmarking of these methods is performed in Chapter 6, where some real-world datasets are presented. Finally, Chapter 7 presents some conclusions on this work and suggests some possible extensions.

Chapter 2

Problem formulation

In this section, the reconstruction problem is formally defined, so that it can be studied using mathematical tools in the following chapters. We introduce the notation and vocabulary that will be used throughout the thesis and delineate the problem by specifying the quantities that we want to estimate, the observation model that relates these quantities to the 2D observations, and the geometric information that is given by the user.

2.1 Estimated quantities

Problems of 3D reconstruction are primarily concerned with the estimation of the positions of 3D points. In the present case, we will estimate a discrete set of points of interest $\mathbf{X}_1, \ldots, \mathbf{X}_N \in \mathbb{R}^3$, usually the vertices of a polyhedron that approximates the scene. The positions and orientations of the optical center(s) of one or more cameras are also estimated and will be written $\mathbf{T}_1, ..., \mathbf{T}_F$ and $R_1, ..., R_F$, the orientations being represented by 3×3 rotation matrices. We assume that not all 3D points are equal, and that no point coincides with the optical center of a camera.

As is customary, the 3D points $\mathbf{X}_m$ and camera positions $\mathbf{T}_f$ are identified with their coordinates in a referential formed by a basis $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ of $\mathbb{R}^3$ and a point of origin. The physical meaning of these coordinates depends on whether a Euclidean or an affine reconstruction is sought. This in turn depends on whether the input data allows to calibrate the camera(s), and will be discussed in Section 2.3.

In the case of Euclidean reconstruction, these estimated coordinates will be related to the "true" values by an unknown rigid transformation, that is, the composition of a scaling, rotation and translation. Physical quantities that are invariant by this type of transformation, such as angles and ratios of distances, can be measured in the reconstruction.

In the case of an affine reconstruction, the estimated and true coordinates are related by an unknown affine transformation, that is, a translation and right multiplication by an arbitrary nonsingular 3×3 matrix, which depends on the geometry of the camera. It is thus not possible to measure angles, which are not invariant to affine transformations, or to measure the ratio of two distances that are taken along different directions.

2.2 Observation model

We now define in mathematical terms how these estimated quantities are observed. We assume the 2D observations $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^2$ are the perspective projections of the 3D points $\mathbf{X}_1, ..., \mathbf{X}_N$, corrupted by additive noise because the camera has optical imperfections, finite resolution and because the user is likely to commit errors when localizing the 2D points in the image. Assuming $\mathbf{x}_m$ has been observed in image f, one has [35, 9] :

$$\lambda_m \begin{bmatrix} \mathbf{x}_m \\ 1 \end{bmatrix} = K \underbrace{\left[\mathbf{r}_1^f \, \mathbf{r}_2^f \, \mathbf{r}_3^f\right]}_{R_f} (\mathbf{X}_m - \mathbf{T}_f) + \begin{bmatrix} \varepsilon_m \\ 0 \end{bmatrix}, \tag{2.1}$$

where λ_m is the "depth", $\mathbf{T}_f$ is the position of the camera in world coordinates and K is the matrix of intrinsic parameters, which are assumed not to vary when many images are available. The errors in the observations are modeled by the terms ε_m, which are realizations of identically distributed, independent Gaussian variables with covariance

$$\operatorname{cov}(\varepsilon_m) = \sigma^2 I_2,$$

where I_2 is the 2×2 identity matrix and σ is the unknown standard deviation of the error. The individual elements of the rotation matrices R_f be written :

$$R_f = \left[\mathbf{r}_1^f \, \mathbf{r}_2^f \, \mathbf{r}_3^f\right] = \begin{bmatrix} r_{11}^f & r_{21}^f & r_{31}^f \\ r_{12}^f & r_{22}^f & r_{32}^f \\ r_{13}^f & r_{23}^f & r_{33}^f \end{bmatrix},$$

so that the depth is :

$$\lambda_m = \sum_{i=1}^{3} r_{i3}^f (X_{mi} - T_{fi}). \tag{2.2}$$

Finally, the matrix of intrinsic parameters, or “calibration matrix”, is assumed to be of the form

$$K = \begin{bmatrix} \rho & 0 & u_0 \\ 0 & \rho & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.3}$$

where ρ is the *focal length* of the camera and $[u_0\, v_0]^\top$ is the *principal point*. In other work [23, 35], the matrix of intrinsic parameters often assumes the more general form :

$$K = \begin{bmatrix} \rho_1 & \alpha & u_0 \\ 0 & \rho_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.4}$$

where ρ_1/ρ_2 is the *aspect ratio* of the camera and α is its *skew*. We prefer the simpler model in Eq. 2.3 because, in the most common case (when only three dominant directions are present), only three intrinsic parameters can be estimated (Section 3.2). Also, because the aspect ratio and skew, for a given sensor, do not vary with time, whereas the focal length and principal point are known to vary when a camera focuses or zooms [47]. In some cases, namely if one or more vanishing point is at infinity, it is not possible to estimate the principal point reliably [9], and we will use the simpler camera model :

$$K = \begin{bmatrix} \rho & 0 & 0 \\ 0 & \rho & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.5}$$

Moreover, if two vanishing points are at infinity, the focal length cannot be estimated either and it is given an arbitrary value.

In the following chapters of this thesis, it will be convenient to have a function $\mathcal{P}$ that associates to a 3D point $\mathbf{X}$, a rotation matrix R, a camera position $\mathbf{T}$ and camera calibration matrix K, the perspective projection of $\mathbf{X}$,

$$\mathbf{x} = \mathcal{P}\left(\mathbf{X}, R, \mathbf{T}, K\right) = \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} KR\left(\mathbf{X} - \mathbf{T}\right), \tag{2.6}$$

where λ is defined as in Eq. (2.2). We call this function the *observation function*. For convenience, we shall group all the parameters $\mathbf{X}_m$, R_f, $\mathbf{T}_f$ and K into a single entity

$$\mathcal{X} = (\mathbf{X}_1, ...\mathbf{X}_N, R_1, ..., R_F, \mathbf{T}_1, ..., \mathbf{T}_F, K)\,, \tag{2.7}$$

and define the m^{th} *observation function* $\mathcal{P}_m$ that associate the m^{th} observation to the parameters $\mathcal{X}$. If, e.g., the m^{th} observation comes from the f^{th} image, one would have :

$$\mathcal{P}_m(\mathcal{X}) = \mathcal{P}(\mathbf{X}_m, R_f, \mathbf{T}_f, K)\,. \tag{2.8}$$

Finally, we introduce another function with the same name $\mathcal{P}$:

$$\mathcal{P}(\mathcal{X}) = \begin{bmatrix} \mathcal{P}_1(\mathcal{X}) \\ \vdots \\ \mathcal{P}_N(\mathcal{X}) \end{bmatrix}. \tag{2.9}$$

This function can easily be disambiguated from that in Eq. (2.6) by the number of arguments and the context.

It should be noted that $\mathcal{P}$ is everywhere many-to-one. One easily sees that $\mathcal{P}(\mathbf{X}, R, \mathbf{T}, K)$ is invariant if $\mathbf{X}$ and $\mathbf{T}$ are translated and scaled arbitrarily. Also, the observations are invariant if $\mathbf{X}$ and $\mathbf{T}$ are transformed by an arbitrary rotation R_0 while the inverse rotation is applied to the camera. In other terms, for any scalar ν, vector $\Delta\mathbf{X} \in \mathbb{R}^3$ and rotation matrix R_0, one has :

$$\mathcal{P}(\mathbf{X}, R, \mathbf{T} + \Delta\mathbf{X}, K) = \mathcal{P}\left(\nu R_0(\mathbf{X} + \Delta\mathbf{X}), R R_0^\top, \nu R_0(\mathbf{T} + \Delta\mathbf{X}), K\right). \tag{2.10}$$

This same invariance can be extended to the function in Eq. (2.6).

2.3 Input data

Finally, we precisely define the input data from which the estimated quantities will be computed. This includes the 2D pixel coordinates $\mathbf{x}_1, \ldots, \mathbf{x}_N$, and when many images are used, one knows in which image each $\mathbf{x}_m$ is observed. Moreover, it is assumed that the original image size (w, h) is known, and the input 2D points have been shifted and scaled so that the $\mathbf{x}_m$ belong to $[-1,\, 1] \times [-1,\, 1]$ rather than $[0,\, w] \times [0,\, h]$. With this transformation, the center of the image has coordinates $[0, 0]$ (expressed in pixels). It is well known [36] that normalization of the input data can greatly improve the conditioning of numerical computations done using these coordinates.

2.3.1 Planarity, alignment and parallelism

In addition to the 2D observations, some *geometric information* is given concerning the 3D points $\mathbf{X}_1, \ldots, \mathbf{X}_N \in \mathbb{R}^3$. Amongst other things, information about planarity and alignment is given : subsets of 2D points are known to be the projections of coplanar or collinear 3D points.

Moreover, some relations of parallelism are known between planes and lines. One may thus group together planes and lines that are parallel and consider the 3D directions that characterize their orientations. Because the resulting 3D directions play a particularly important role in single-view reconstruction, we will call them the "*dominant directions*" of the considered scene. Typically, at least three independent dominant directions are used, which most often are orthogonal two-by-two and called "X", "Y" and "Z". These directions will be written $\mathbf{v}_1, ..., \mathbf{v}_D$. Because the first three directions will be used as a basis in which the reconstruction is obtained, we assume that they are independent. Just like the 3D points $\mathbf{X}_m$, the dominant directions $\mathbf{v}_i$ will be identified with their coordinates in the basis $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$, so that $\mathbf{v}_1 = [1\,0\,0]^\top$, $\mathbf{v}_2 = [0\,1\,0]^\top$ and $\mathbf{v}_3 = [0\,0\,1]^\top$.

There are two natural ways to define the orientation of a plane, either with a direction that is orthogonal to the plane, or by two directions that are parallel to it. The first form cannot be used if an affine reconstruction is sought, because the orthogonality relation on which it relies is not preserved by affine transformations. It is thus necessary, in that case, to express planarity by two directions parallel to the plane, rather than by its normal.

Also, one should note that alignment information can be dispensed with altogether, because each line can be defined as the intersection of two planes. For example, saying that some points belong to a line parallel to the "X" direction is equivalent to saying that they belong to a plane parallel to the "X" and "Y" directions, and that they also belong to a plane parallel to the "X" and "Z" directions. Because this simplifies the treatment of geometric information, we assume that all alignment information is represented as planarity information.

To give a better idea of how planes are used, one should say that a plane does not necessarily correspond to a facet of the object and that a plane may be defined to contain as little as a single point. For example, going back to the example in Figure 1.6, on page 21, the plane π_2 contains a single point. Moreover, when multiple views are being used, one plane may contain points that are observed in different images. In fact, in this thesis, it is in this way that points are tracked from one image to another. Indeed, saying that a 2D point $\mathbf{x}_m$, in one image, and point $\mathbf{x}_n$, in another image, correspond to a same 3D point, is equivalent to saying that the original 3D points $\mathbf{X}_m$

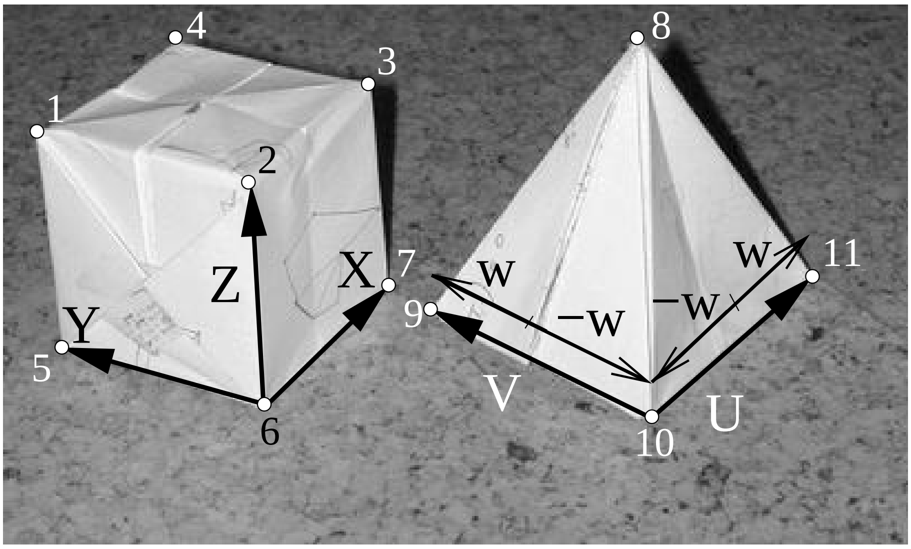


Figure 2.1: This simple dataset illustrates the type of known ratios of distances between parallel planes that are commonly found in real-world situations. Saying that the pyramid is symmetric with respect to a "U-Z" plane passing through its top is equivalent to saying that the distance from point 8 to point 9 along the "V" direction is equal to minus one times the distance from point 8 to point 10 along that direction. This can be written as $\overline{8,9}^V = -\overline{8,10}^V$, where $\overline{8,9}^V$ is the distance between points 8 and 9 along the "V" direction. Also, because the pyramid is symmetric around a vertical line passing through its top, one has $\overline{8,9}^V = \overline{8,11}^U$, and because the parallelepiped 1-7 is a cube, $\overline{1,2}^Y = -\overline{1,4}^X = \overline{1,5}^Z$.

and $\mathbf{X}_n$ are equal. This equality will be expressed by defining three planes, e.g. "X=Constant", "Y=Constant" and "Z=Constant" that contain points $\mathbf{X}_m$ and $\mathbf{X}_n$. Since the intersection of these three planes is a single point, $\mathbf{X}_m$ and $\mathbf{X}_n$ are necessarily equal. Planarity information is thus not limited to expressing the flatness of object faces.

2.3.2 Known ratios of signed distances between parallel planes

Other geometric properties are commonly found in man-made scenes, such as symmetry and repetition. These properties can be expressed using distances between 3D points. In this work, they are expressed in such a way that they can be translated into linear constraints. This is made possible by asking the user to specify the direction along which the distances are measured.

Figure 1.6(right) (page 21) and the associated text on page. 21 showed how known ratios of distances between parallel planes could be used for that purpose, and we now illustrate how this is done in practice.

Consider, for example, that we are interested in a Euclidean reconstruction of the pyramid in

Figure 2.1. This object is symmetric with respect to a “U-Z” plane passing through its top, which is equivalent to saying that the distance from point 8 to point 9 along the “V” direction is equal to minus one times the distance from point 8 to point 10 along the “V” direction. We may write this as $\overline{8,9}^V = -\overline{8,10}^V$, where $\overline{8,9}^V$ is the distance between points 8 and 9 along the “V” direction. Likewise, the fact that the pyramid has a square base can be written $\overline{9,10}^V = \overline{11,10}^U$ (note that one distance is measured along the “V” directions and the other along the “U” direction). In the present work, the user specifies this type of geometric information by giving the indices of pairs of points and directions that are involved.

In more precise terms, the relation which we wrote $\overline{8,9}^V = -\overline{8,10}^V$can be expressed by the equation

$$\mathbf{v}_5^\top (\mathbf{X}_9 - \mathbf{X}_8) = -\mathbf{v}_5^\top (\mathbf{X}_{10} - \mathbf{X}_8), \tag{2.11}$$

where $\mathbf{v}_5$ is the vector of coordinates of the “V” direction and $\mathbf{X}_i$ holds the coordinates of the i^{th} point. One easily notes that, in this equation, one could substitute $\mathbf{X}_{10}$ by $\mathbf{X}_{11}$, or by any other point $\mathbf{X}_i$ such that $\mathbf{v}_5^\top (\mathbf{X}_{10} - \mathbf{X}_i)$. That is, any point belonging to the plane orthogonal to $\mathbf{v}_5$ and passing through $\mathbf{X}_{10}$ could be used instead of $\mathbf{X}_{10}$. This equation clearly shows that distances between planes are being considered, rather than between points.

As said above, in the case of affine reconstruction, the orientation of planes should be specified by two linearly independent directions that are parallel to the plane. Looking again at the relation, written $\overline{8,9}^V = -\overline{8,10}^V$in the Euclidean case, one sees that the planes that it involves are parallel to the “U” and “Z” directions, so that it could be written $\overline{8,9}^{U,Z} = -\overline{8,10}^{U,Z}$ in shorthand notation. The converse of Eq. (29) would then be :

$$(\mathbf{v}_3 \times \mathbf{v}_4)^\top (\mathbf{X}_9 - \mathbf{X}_8) = -(\mathbf{v}_3 \times \mathbf{v}_4)^\top (\mathbf{X}_{10} - \mathbf{X}_8).$$

Also, it should be noted that, in the affine case, one cannot relate distances taken along different directions, because they are not invariant to affine transformation. It is thus not possible to express that the base of the pyramid is square (or even that it is a rectangle, since the angle between its edges are not estimated). The use of known ratios of signed distances between pairs of parallel planes is thus more limited in the affine case than in the Euclidean case.

In some cases, however, a single dominant direction can be used to specify the direction in which the distance is taken, even when an affine reconstruction is sought. This can be done when the two considered points lie on a segment that is parallel to the dominant direction. For example, the distance $\overline{8,9}^V$ could be used to express a known ratio of signed distances taken along a line[1].

Going back to the Euclidean case, an issue arises when the user says that the base of the pyramid is a square e.g. with the relation $\overline{10,9}^V = \overline{10,11}^U$. Indeed, if the direction of the "U" axis is inverted, then distances measured along this direction are also inverted, so that the correct relation would be $\overline{10,9}^V = -\overline{10,11}^U$. When this problem occurs, it is solved by asking the user to specify in the image plane the direction along which distances are positive.

This concludes the discussion on how known ratios of distances can be specified by the user and on related geometric issues. Using this type of information is important because the geometric properties that are considered are quite common in man-made scenes, and often allow to obtain a reconstruction in situations that no other presently published method could treat.

2.3.3 Known angles between dominant directions

Finally, after information on points and planes, the user may also indicate constraints on the dominant directions. First, some directions may be fixed, which is equivalent to choosing the orientation of the frame in which the reconstruction is obtained. Also, the user may indicate known angles as well coplanarity relations between directions. In order to benefit from simplifications in the implementation, some restrictions are set on the usage of these constraints, which are detailed in Section 5.2.1.

As we have said in the introduction, known angles can be used to obtain the calibration of a camera from a single view, and it is the possibility of calibrating the camera that determines whether a Euclidean or affine reconstruction is obtained. This type of information is thus very important for the reconstruction problem.

We take the most commonly used approach, which is to estimate three or less intrinsic parameters from three mutually orthogonal vanishing points. Although more general methods could be considered, this is the only calibration method that is used in this thesis. As a consequence, we will obtain Euclidean reconstruction only when three dominant directions are known to be orthogonal, and an affine reconstruction otherwise.

Because the vanishing points are needed in the initial steps of the reconstruction, we assume that enough geometric information has been provided to estimate the vanishing point of each dominant direction. The vanishing points are estimated from the intersections of two or more 2D

[1] No useful relationship of this kind appears in this dataset.

lines, which consist in the 2D projections of two or more 3D points that belong to a 3D line parallel to the considered direction. The alignment of 3D points is deduced from the intersection of distinct planes that contain the points. In order to estimate the vanishing points, it is thus necessary to assume that sufficient planarity information has been provided.

2.3.4 Summary of geometric information

In summary, the geometric information consists of the following elements :

- *Planarity information*: Subsets of 2D points whose corresponding 3D points are known to belong to a 3D plane parallel to two of the dominant directions, which are also given. One can specify the 2D points by a set of indices included in $\{1, ..., N\}$, and the dominant directions by a pair of indices in $\{1, ..., D\}$. If a Euclidean (rather than affine) reconstruction is sought, the orientation of a plane may be defined by a single dominant direction to which it is orthogonal.

- *Known ratios of signed distances between parallel planes* are represented by two pairs of parallel planes $(\mathcal{P}, \mathcal{P}')$ and $(\mathcal{Q}, \mathcal{Q}')$ and the knowledge of the ratio α of the distances between them. In the case of an affine reconstruction, one must furthermore assume that all four planes are orthogonal, while, in the case of a Euclidean reconstruction, and if the four planes are not parallel, it is necessary to indicate the direction of positive distances for the involved directions.

- *Known angles* between pairs of dominant directions and *coplanarities* between triplets of dominant directions. Information of this type is not absolutely necessary. If given at all, we assume that the first three dominant directions form a right trihedron, from which calibration will be estimated. In this last case, a Euclidean reconstruction is obtained rather than an affine reconstruction.

Chapter 3

Estimating vanishing points, calibration and dominant directions

In this chapter, we estimate the principal directions $\mathbf{v}_1, ..., \mathbf{v}_D$ which will be used by the reconstruction method presented in the next chapter. This is done by estimating the corresponding vanishing points and, if possible, using a method similar to that of Caprile and Torre [9] to calibrate the camera.

Image information and geometric information are used to compute as many vanishing points as possible, from which the relative orientation of the cameras is computed. The vanishing points are identified with the coordinates of the principal directions, in the basis associated to the sensor, which is not necessarily orthogonal. If possible, the principal directions will be represented in an orthogonal basis, so as to later obtain a Euclidean reconstruction. More precisely, if the three first principal directions are known to form a right trihedron, then the corresponding vanishing points are used to compute the camera calibration.

Because the problem of estimating the principal directions is somewhat marginal to this thesis, the reader may skip this chapter in a first reading. In the first two sections, the vanishing points and calibration are computed. Because the same treatment is done on the data of each image, we may consider that a single image is present and drop the image number f from our notation. The principal directions are estimated in the last section, using the information previously obtained from all the available images.

3.1 Maximum likelihood vanishing point estimation from 2D lines

In this section, we show how the geometric info allows to identify projections of parallel 3D lines and how to locate their intersection, which is the vanishing point of that direction. For each image and principal direction $\mathbf{v}_i$, the geometric information allows to identify collections of 2D points that lie on the projection of a 3D line parallel to $\mathbf{v}_i$, by finding the intersections of planes that define a line parallel to $\mathbf{v}_i$.

Each 2D line consists of a subset of the points $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$. If it consists in $\mu \in \mathbb{N}$ points, it is defined by the indices $\{m_1, ..., m_\mu\} \subset \{1, ..., N\}$ of the points it contains, and we may identify the line with this set of indices. If two lines $\{m_1^1, ..., m_{\mu_1}^1\}$, $\{m_1^2, ..., m_{\mu_2}^2\}$ are found, it is possible to find their intersection, which is the vanishing point of the dominant direction $\mathbf{v}_i$. If a single line is found and the vanishing line of a plane parallel to $\mathbf{v}_i$ is known (e.g. from two vanishing points that lie on it), it is still possible to estimate the vanishing point of $\mathbf{v}_i$ as the intersection of the observed line and of the vanishing line of the plane [62].

The main difficulty in estimating vanishing points appears when three or more lines are present, and, because of noise in the observations, do not intersect in a single point. The remaining of this section gives the details on how to compute the maximum likelihood estimate of a vanishing point.

We consider the vanishing point $\mathbf{g}_i$ of the principal direction $\mathbf{v}_i$ and assume that Q line segments are observed. Each segment is defined by the 2D points it contains, and we write $\mathcal{I}_1 = (m_1^1, ..., m_{\mu_1}^1), \ldots, \mathcal{I}_Q = (m_1^Q, ..., m_{\mu_Q}^Q)$ the lists of indices of points that belong to these segment. With this notation, segment number j consists of points $\mathbf{x}_{m_1^j}, ..., \mathbf{x}_{m_{\mu_j}^j}$, where μ_j is the number of points it contains.

As said in Section 2.2 (p. 24), we assume that the error terms in the observations are independent, Gaussian and have covariance $\sigma^2 I_2$, for some unknown σ. The maximum likelihood estimate of the vanishing point is the intersection $\mathbf{g}$ of lines $\mathbf{l}_1, ..., \mathbf{l}_Q$ that best fit the 2D data. More precisely, $\mathbf{g}$ can be defined by

$$\begin{aligned} \mathbf{g} &\sim \mathbf{l}_1 \times \mathbf{l}_2, \text{ where} \\ (\mathbf{l}_1, ..., \mathbf{l}_Q) &= \arg\min_{\mathbf{l}_1, \ldots, \mathbf{l}_Q} \sum_{j=1}^{Q} \sum_{k=1}^{\mu_j} d\left(\mathbf{l}_j, \mathbf{x}_{m_k^j}\right)^2, \\ & \text{under the constraint that all } \mathbf{l}_j \text{ intersect in } \mathbf{g}. \end{aligned}$$

In this equation, $\sim$ denotes equality up to a scale factor and $d(\mathbf{l}, \mathbf{x})$ is the Euclidean distance between the line $\mathbf{l}$ and the point $\mathbf{x}$. Image lines are represented by a 3×1 vector $\mathbf{l}$ and the set of 2D points contained in this line is $\left\{\mathbf{x} \in \mathbb{R}^2 | \left[\mathbf{x}^\top\, 1\right] \mathbf{l} = 0\right\}$. With this representation, $\mathbf{l}_1 \times \mathbf{l}_2$ represents the intersection of $\mathbf{l}_1$ and $\mathbf{l}_2$.

We show in Appendix A that, for a given $\mathbf{g}$, the optimal $\mathbf{l}_j$ are easily found. For any j, the line $\mathbf{l}_j$ passing through $\mathbf{g}$ that minimizes

$$\sum_{k=1}^{\mu_j} d\left(\mathbf{l}_j, \mathbf{x}_{m_k^j}\right)^2$$

is either the line $\hat{\mathbf{l}}_j$ (Figure 3.1, left) that passes through $\mathbf{g}$ and $\bar{\mathbf{x}}_j = \sum_k \mathbf{x}_{m_k^j} / \mu_j$ (the centroid of the set of points), $\hat{\mathbf{l}}_j = \mathbf{g} \times \bar{\mathbf{x}}_j$, or the line $\hat{\mathbf{l}}'_j$ (Figure 3.1, right) orthogonal to $\hat{\mathbf{l}}_j$ and passing through $\mathbf{g}$. One then has to minimize a function of $\mathbf{g}$ alone, defined by :

$$q(\mathbf{g}) = \sum_{j=1}^{Q} \min \left\{ \sum_{k=1}^{\mu_j} d\left(\hat{\mathbf{l}}_j, \mathbf{x}_{m_k^j}\right)^2, \sum_{k=1}^{\mu_j} d\left(\hat{\mathbf{l}}'_j, \mathbf{x}_{m_k^j}\right)^2 \right\}. \tag{3.1}$$

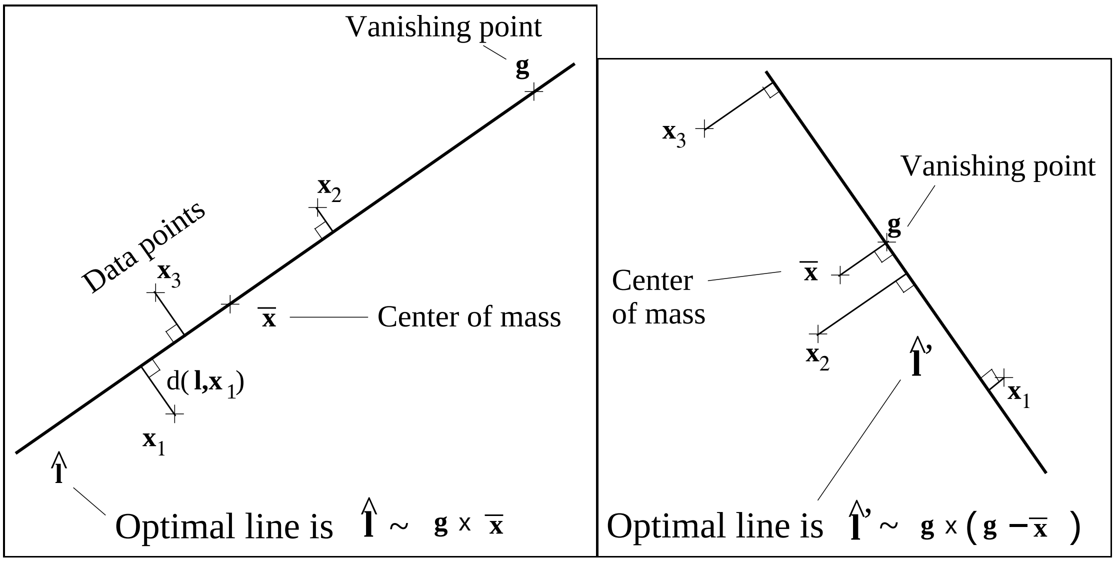


Figure 3.1: The line passing through $\mathbf{g}$ that minimizes the sum of distances to the $\mathbf{x}_i$ is either $\hat{\mathbf{l}}$ that passes through the centroid $\bar{\mathbf{x}}$ of the $\mathbf{x}_i$ (left), or $\hat{\mathbf{l}}'$ that is orthogonal to $\hat{\mathbf{l}}$ (right).

In practice, Eq. (3.1) can be computed without doing a sum over k at each time. Moreover the likelihood function can be extended to the whole projective plane, including points at infinity (Appendix A.3). This function is continuous, and we use the Nelder-Mead algorithm [56] to locate its minimum. In order to avoid falling in local minimum, the initial position is chosen amongst 626 points nearly uniformly located on the half-sphere $\left\{\mathbf{g} \in \mathbb{R}^3 | \, ||\mathbf{g}|| = 1, \, [0\,0\,1]^\top \mathbf{g} \geq 0\right\}$.

3.1.1 Experimental results

We now study experimentally the precision of the vanishing point estimation method.

Mean absolute error v.s. noise We first consider the sensitivity to noise in the observations. Two setups are used, one with four segments (Figure 3.2 left) spanning $\pi/3$ radians, consisting of four points each, the other with two segments, $\pi/3$ radians apart, consisting of two points each. The point on the segment closest to the vanishing point is two times closer than the farthest. The points are translated so that their center of mass coincides with the origin of coordinates and scaled so that the field of view corresponds to approximately 90 degrees.

Gaussian independent and identically distributed (i.i.d.) noise terms are added to these 2D points, with a standard deviation between 0.1% and 3.1% of the mean distance of the 2D points to the center of mass of the 2D points. This range of noise levels is wider than that encountered in real-world situations, when error lies between 0.3 and 1% (see Section 6) of the standard deviation of the data. The maximum likelihood vanishing point is estimated and the error between the estimated and true vanishing point is measured by the angle between them.

This process is repeated 30 times. The mean absolute error is taken for each error level and plotted in Figure 3.2 (right). The smooth curve plots the error obtained from four segments of four points, whereas the dashed curve plots the error obtained from two segments of two points. This figure shows that the error evolves approximately linearly with the error level in the observations, indicating that the maximum likelihood estimator is behaves well at the considered noise levels.

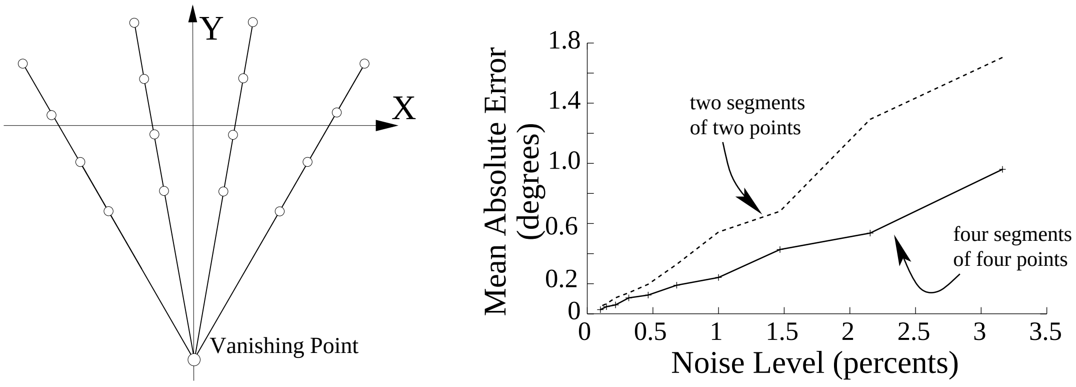


Figure 3.2: **Left:** Setup consisting in four line segments, each defined by four 2D points. **Right:** Effect of noise on the error in vanishing points estimated from a setup similar to that on the right and from a setup consisting of just two segments of two points. The curves show that the mean absolute error increases approximately linearly with the amplitude of the noise.

Mean absolute error v.s. angle between segments We now study how the error in the estimated vanishing points varies when the vanishing point moves to infinity. The setup in this experiment has two segments consisting of four points each. The distances from the 2D points to the vanishing point are in the interval $[1, 2]$. The angle between the segments is made to vary between 0 radians (parallel segments) and $\pi/2$ radians (orthogonal segments). Gaussian i.i.d noise terms are added to the 2D points, whose standard deviation is 1% of that of the 2D points. The maximum likelihood vanishing point is estimated, and the error between the estimated and true vanishing point is measured by the angle between them.

This process is repeated 30 times. The mean absolute error is taken for each angle and plotted in Figure 3.3 (right), which shows that the error is, to a large extent, independent of the angle between the segments.

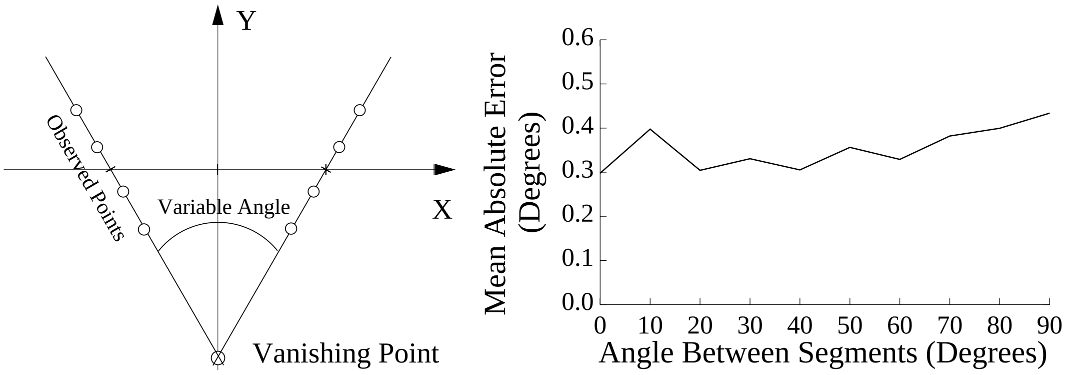


Figure 3.3: **Left:** Setup consisting in two line segments, each defined by four 2D points. **Right:** Effect of angle between segments on the error in vanishing points estimated from the setup on the left. The curve shows that the mean absolute error in estimating the vanishing point does not vary very much with the angle between the two intersecting lines.

3.2 Calibration from vanishing points

If the three first dominant directions form a right trihedron, and K has the form in Eq (2.3) or (2.5) (page 25), then it is well known [9] that ρ, u_0 and v_0 can be estimated from the vanishing points $\mathbf{g}_1$, $\mathbf{g}_2$ and $\mathbf{g}_3$, except in the following cases :

- If a single vanishing point is at infinity[1], the principal point $[u_0\, v_0\; 1]^\top$ cannot be uniquely determined. In that case, we assume $u_0 = v_0 = 0$, and estimate ρ only.

[1] A point on the projective plane, with homogeneous coordinates $[x_1\, x_2\, x_3]$ is said to be "at infinity", or "on the line at infinity", if and only if $x_3 = 0$.

- If two vanishing points are on the line at infinity, ρ cannot be estimated either and we will obtain an affine reconstruction.

3.2.1 Details of calibration

We now detail how the calibration is obtained from three vanishing points $\mathbf{g}_1$, $\mathbf{g}_2$ and $\mathbf{g}_3$, corresponding to orthogonal 3D directions. From the observation model (Eq. 2.1), one has :

$$[\mathbf{g}_1\,\mathbf{g}_2\,\mathbf{g}_3]\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{bmatrix} = KR,$$

where R is the orthogonal matrix representing the orientation of the camera and the λ_i are unknown scale factors. From this equation, one easily deduces :

$$\begin{bmatrix}\lambda_1^2 & 0 & 0\\ 0 & \lambda_2^2 & 0\\ 0 & 0 & \lambda_3^2\end{bmatrix} = [\mathbf{g}_1\,\mathbf{g}_2\,\mathbf{g}_3]^{-1} KK^\top [\mathbf{g}_1\,\mathbf{g}_2\,\mathbf{g}_3]^{-\top} . \qquad (3.2)$$

We take as estimates of ρ, u_0 and v_0, the values that minimize the sum of squares of off-diagonal terms of the right hand side of Eq. (3.2). These terms are easily seen to be differentiable functions of ρ, u_0 and v_0, so that one has to minimize a differentiable function. This minimization is done using the Nelder-Mead [56] algorithm.

3.2.2 Experimental results

We have benchmarked the process of calibration from the vanishing points of three orthogonal directions. We focus on the effect of one or two vanishing points going to infinity, when the principal point or focal length (respectively) cannot be estimated.

We designed the setup so that, by varying the angle τ between the world "Z" axis and the optical axis (the normal to the image plane), from 0 to $\pi/2$, one goes from one critical setup (one vanishing point at infinity) to the other (two vanishing points at infinity), while passing by setups that are adequate for calibration. Figure 3.4 shows how the dominant directions are oriented with respect to the optical axis.

- The "X" and "Y" axes are symmetric with respect to the plane containing the "Z" axis and the optical axis.

- The vectors $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$, representing the "X", "Y" and "Z" axes, are the columns of the 3×3 matrix corresponding to a rotation around the axis $[-1\ 1\ 0]$, with angle τ, where τ is the angle between the "Z" axis and the optical axis (Figure 3.4).

Thus, when $\tau = 0$, two vanishing points (that of "X" and "Y") are at infinity. When $\tau = \pi/2$, one vanishing point (that of "Z") is at infinity and for intermediate values of τ, all vanishing points are finite.

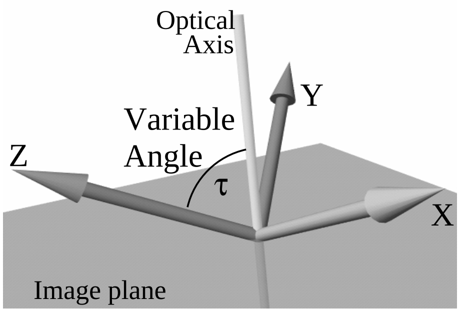


Figure 3.4: Setup used for benchmarking the calibration process : the "Z" axis forms an angle from 0 to $\pi/2$ radians with the image plane, while the "X" and "Y" axes point upward.

The vanishing point $\mathbf{g}_i$ are obtained by (see Equation 2.1) :

$$\mathbf{g}_{\tilde{i}} \sim \begin{bmatrix} \rho & 0 & u_0 \\ 0 & \rho & v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{v}_i + \text{Error},$$

where $\sim$ denotes equality up to a scale factor, ρ is uniformly distributed in $[1, 2]$ (not an uncommon value, in real-world situations, when the pixel coordinates are normalized in $[\pm 1] \times [\pm 1]$) and u_0, v_0 are i.i.d Gaussian variables with mean 0 and variance 0.05; these values are in accordance with that mentioned in [47]. The "Error" term is Gaussian, with amplitude chosen in such a way to correspond to an angular error of either 0.2 degrees (Figure 3.5, top) or 0.4 degrees (Figure 3.5, bottom). These values correspond approximately to lower and upper bounds for errors, as shown in Section 3.1.1.

The calibration parameters are estimated in two distinct cases : in one (dashed curves), only the focal length ρ is estimated, whereas in the other, ρ, u_0 and v_0 are estimated. In both cases, estimates are obtained by minimizing the sum of squared off-diagonal terms in Eq. (3.2).

The error in focal length is measured by the absolute value of the difference between true and estimated values. The error in the principal point is measured by the norm of the difference between true and estimated values. The estimation process is repeated 50 times for each value of the angle τ between the "Z" axis and the image plane, and the mean error is plotted.

Figure 3.5 shows the error as a function of the angle between the "Z" axis and the image plane, when the error in the vanishing points is 0.2 (top) or 0.4 (bottom) degrees respectively. The dashed curves plot the error obtained when only the focal length ρ is estimated (and one assumes $u_0 = v_0 = 0$), while the continuous curves plot the error obtained when ρ, u_0 and v_0 are estimated. The error in the principal point is plotted on the left (a,c) while the error in the focal length is plotted on the right (b,d).

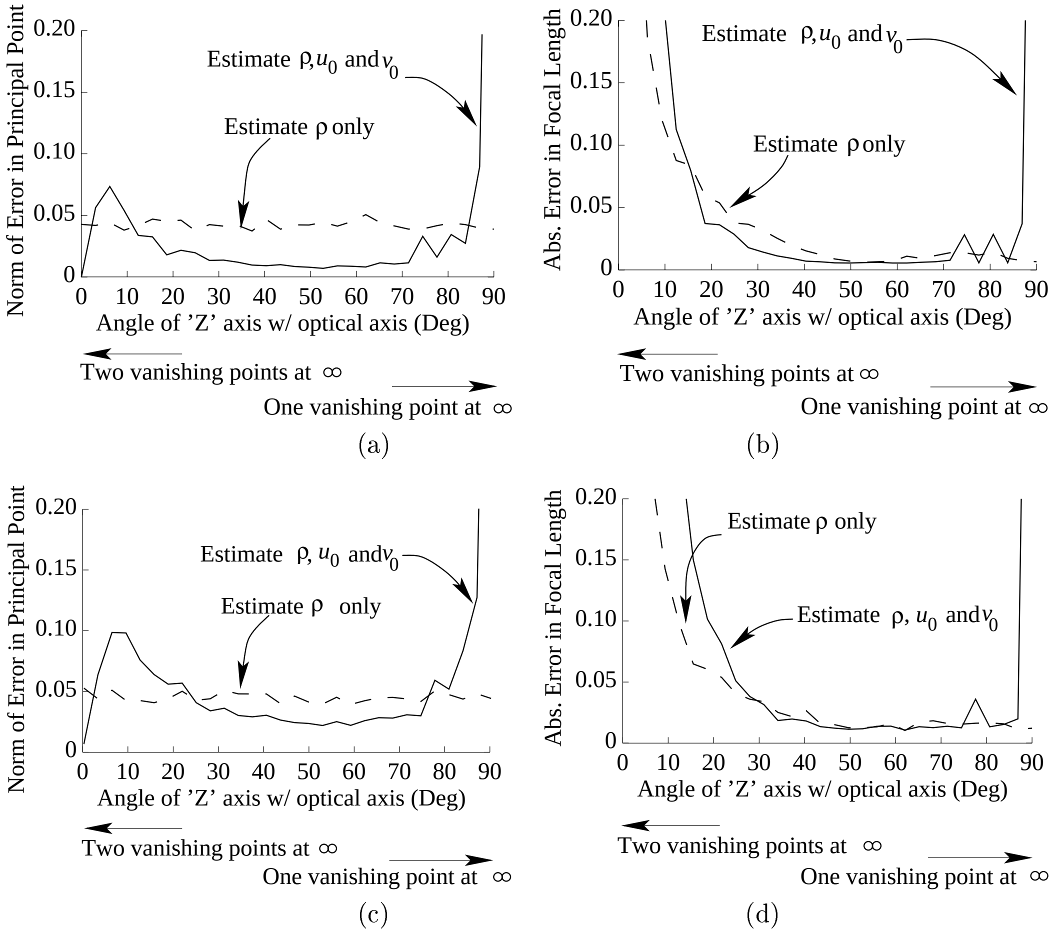


Figure 3.5: Error when calibrating from vanishing points subject to 0.2 (**top** a,b) or 0.4 (**bottom** b,c) degrees of error. **Left** (a,c) : Mean error in the estimated principal point vs. angle between "Z" and the optical axis. **Right** (c,d): Mean error in the estimated focal length vs. angle between "Z" and the optical axis.

The principal observations that arise from these curves are :

- The continuous curve in Figure 3.5 (bottom left) shows that, when ρ, u_0 and v_0 are estimated, the principal point (u_0, v_0) is not estimated reliably : except for angles below 5 degrees, the error is never less than half the amplitude of the estimated quantity itself, and outside of the range of $[25, 75]$ degrees, the error is often greater than that of the estimated quantity.
- The same curve in Figure 3.5 (top left) shows that, when the error is 0.2 degrees, the principal point can be more reliably estimated, on a wider range of angles.
- When two vanishing points are near the line at infinity ($\tau < 20^o$), the absolute error in the estimated focal length increases quickly, as is expected and was noted in [67].
- The estimation of the focal length never improves much from estimating the principal point, and degrades a lot at the extremes of the angle range.

From these remarks, we conclude that, unless one is sure that the error level in the vanishing points is small, and that no vanishing point is within 15 degrees from the line of infinity, it is better to only estimate the focal length. This result is in agreement with similar claims in the literature [67, Section 3].

3.3 Estimation of principal directions

In the basis formed by the first three dominant directions, the coordinates of these directions are $\mathbf{v}_1 = [1\,0\,0]^\top$, $\mathbf{v}_2 = [0\,1\,0]^\top$ and $\mathbf{v}_3 = [0\,0\,1]^\top$. The coordinates of the other dominant directions $\mathbf{v}_4, \ldots$ (if any) can then be estimated by :

$$\mathbf{v}_i = \left[\mathbf{g}_1^f\,\mathbf{g}_2^f\,\mathbf{g}_3^f\right]^{-1}\mathbf{g}_i^f .$$

If many (for many f) estimates of $\mathbf{v}_i$ are available, the mean is taken and normalized to have unit norm. Of course, if only three dominant directions are present, no estimation is needed, since their coordinates are simply $[1\,0\,0]^\top$, $[0\,1\,0]^\top$ and $[0\,0\,1]^\top$.

3.4 Summary of algorithm

This section provides a brief summary of how the techniques presented in this chapter are used to obtain estimates of the vanishing points, of the dominant directions (those that are not fixed) and -if possible- of the calibration. This is done in the following steps :

1. Determine in each image what 3D directions are projected in two or more lines and compute the ML estimate of the corresponding vanishing points.

2. If possible, estimate other vanishing points from a single line and the vanishing line of a plane parallel to that direction.

3. If the three first dominant directions are known to form a right trihedron,

   (a) Estimate calibration matrix K.

   (b) Transform observations $\begin{bmatrix} \mathbf{x}_m \\ 1 \end{bmatrix}$ into $K^{-1} \begin{bmatrix} \mathbf{x}_m \\ 1 \end{bmatrix}$ and the vanishing points $\mathbf{g}_i^f$ into $K^{-1}\mathbf{g}_i^f$ so that all coordinates are expressed in an orthogonal basis.

4. Estimate the dominant directions (expressed as their coordinates in the basis formed by $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$).

Chapter 4

Reconstruction as a linear equality problem

In this chapter, we show that, once dominant directions have been estimated, the reconstruction problem is equivalent to solving a system of linear equations. In the absence of errors in the observations, the set of solutions to the reconstruction problem can be identified with that of the linear problem and the dimension of the later indicates whether the solution is unique up to a scale factor and translation.

In the presence of errors in the observations, the rank of the system is altered, so that it cannot be used to determine the unicity of the solution. This difficulty can be overcome by noticing that it is always possible to build a "twin" linear system, whose rank does not depend on the error in the observations and whose corank is equal to four if and only if the dataset defines a reconstruction up to scale and translation.

4.1 Geometric constraints

We now show how the geometric information -coplanarity and known ratios of lengths- can be expressed as a system of linear equations on the coordinates of the 3D points. All solutions to the reconstruction problem will verify exactly these equalities. We build a basis for the set of coordinates (a linear subspace) that verify these equalities. Examining this basis, one checks whether the user provided coherent geometric information.

Planarity information Two 3D points m, n belong to a plane parallel to the directions $\mathbf{v}_i$, $\mathbf{v}_j$, if and only if their coordinates $\mathbf{X}_m$, $\mathbf{X}_n$ verify the equation :

$$\left(\mathbf{v}_i \times \mathbf{v}_j\right)^\top \left(\mathbf{X}_m - \mathbf{X}_n\right) = 0. \tag{4.1}$$

Metric information Saying that points (m,n) (resp. (p,q)) lie on a pair of parallel planes $\mathcal{P}$, $\mathcal{P}'$ (resp. $\mathcal{Q}$, $\mathcal{Q}'$) with normal $\mathbf{v}$ (resp. $\mathbf{w}$) and that one knows the ratio α of the distances from $\mathcal{P}$ to $\mathcal{P}'$ and from $\mathcal{Q}$ to $\mathcal{Q}'$, is equivalent to saying that :

$$\mathbf{v}^\top \left(\mathbf{X}_m - \mathbf{X}_n\right) = \alpha \mathbf{w}^\top \left(\mathbf{X}_p - \mathbf{X}_q\right). \tag{4.2}$$

The normals may be specified as the cross product of two dominant directions, or, if an Euclidean reconstruction is sought, by a dominant direction (see Section 2.3, page 27).

By grouping all the equations (4.1) and (4.2) provided by the geometric information, one gets a system of equations, the *geometric constraints* :

$$B\left(\mathbf{v}_1, ..., \mathbf{v}_D\right) \mathbf{X} = \mathbb{0}, \tag{4.3}$$

where $\mathbf{X} = \left[\mathbf{X}_1^\top, \ldots, \mathbf{X}_N^\top\right]^\top$ is the $3N \times 1$ vector holding all the point coordinates and $B\left(\mathbf{v}_1, ..., \mathbf{v}_D\right)$ is a $P \times 3N$ matrix holding the coefficients in equations (4.1) and (4.2). The notation $B\left(\mathbf{v}_1, ..., \mathbf{v}_D\right)$ is used to emphasize that this matrix only depends on the directions $\left(\mathbf{v}_1, ..., \mathbf{v}_D\right) \in \mathbb{R}^{3N}$ and will often be replaced by B for brevity.

It is clear that Eq. (4.3) always has nonzero solutions, since, if one choose arbitrarily $\mathbf{X}_1 = ... = \mathbf{X}_N \in \mathbb{R}^3 \setminus \{\mathbb{0}\}$, all the constraints Eq. (4.2) and (4.1) are necessarily verified. The dimension M of the nullspace of B is thus greater or equal to 3 and there exists a $3N \times M$ matrix U whose columns form an orthonormal basis of this nullspace. In practice, this matrix is computed by doing the singular value decomposition [26] of B.

The matrix U thus forms a basis of the set of (collections of) 3D points that verify all the geometric constraints, since all the solutions to Eq. (4.3) are of the form

$$\mathbf{X} = U\left(\mathbf{v}_1, ..., \mathbf{v}_D\right) \mathbf{V} \tag{4.4}$$

for some $\mathbf{V} \in \mathbb{R}^M$. Again, the notation $U\left(\mathbf{v}_1, ..., \mathbf{v}_D\right)$ is used to emphasize the fact that this matrix depends only on the dominant directions, but U will be used for brevity. Each triplet of

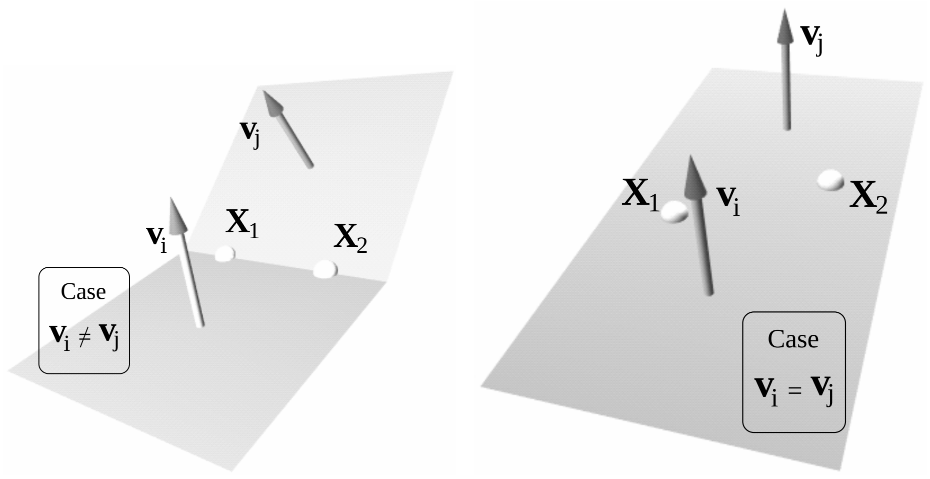


Figure 4.1: **Left :** If the dominant directions $\mathbf{v}_i$ and $\mathbf{v}_j$ are different, then points $\mathbf{X}_1$ and $\mathbf{X}_2$ that belong simultaneously to a plane with normal $\mathbf{v}_i$ and a plane with normal $\mathbf{v}_j$ are constrained to belong to a line. **Right :** If the dominant directions $\mathbf{v}_i$ and $\mathbf{v}_j$ are parallel, the same planarity properties only constrain the points $\mathbf{X}_1$ and $\mathbf{X}_2$ belong to a plane, and there is one extra degree of freedom.

rows numbered $\{3m-2, 3m-1, 3m\}$ of U correspond to the coordinates $\mathbf{X}_m$. In intuitive terms, M (the corank of B) is the number of "degrees of freedom" of the object formed by the 3D points subject to the geometric constraints.

Examining U allows us to perform a check on the coherence of the geometric information : if any triplets $\{3m-2, 3m-1, 3m\}$ and $\{3n-2, 3n-1, 3n\}$ $(m \neq n)$ of rows of U, corresponding to observations in the same image, are equal, this indicates that all solutions to Eq. (4.2) verify $\mathbf{X}_m = \mathbf{X}_n$. That is, a single 3D point has two distinct observations $\mathbf{x}_m$ and $\mathbf{x}_n$ in the same image, which is likely to indicate an error in the geometric information.

This situation may occur when the user adds some spurious points to a plane, when (s)he does a mistake in the orientation of a plane or in the sign of some ratios of distances. Such errors become more likely when dozens of points and planes are present, in which case it becomes more important detect these situations.

Assumptions on B We now consider an assumption on the matrix B that is important for the study of the algebraic properties of the reconstruction problem. We assume that, for all the values $\mathbf{v}_i$ that we will be considering, the rank of B does not vary when the $\mathbf{v}_i$ are slightly perturbed. Geometrically, this means that the restrictions given to the 3D points do not change in nature when the $\mathbf{v}_i$ are slightly changed, meaning that we are away from critical configuration (some $\mathbf{v}_i$

almost parallel etc).

Figure 4.1 illustrates the distance between critical and non-critical situations. At the left, if $\mathbf{v}_i$ and $\mathbf{v}_j$ are slightly altered, the points $\mathbf{X}_1$ and $\mathbf{X}_2$ will still be constrained to lie on a line perpendicular to both $\mathbf{v}_i$ and $\mathbf{v}_j$. On the right is a setup forbidden by our assumption on B, where the points $\mathbf{X}_1$ and $\mathbf{X}_2$ are constrained to belong to a plane, but, if $\mathbf{v}_i$ is slightly changed, then the points will be constrained to belong to a line.

It can be shown that the rank of $B(\mathbf{v}_1, ..., \mathbf{v}_D)$ reaches its maximum for almost all values of the $\mathbf{v}_i$. This is easily done by noting that the minors of $B(\mathbf{v}_1, ..., \mathbf{v}_D)$ are polynomial functions of the $\mathbf{v}_i$'s coordinates and thus either are identically zero, or have isolated zeros. If M is the highest integer such that there exists a nonzero minor of B (for some $\mathbf{v}_1, ..., \mathbf{v}_D$), then the maximum rank of B is M and B has this rank for almost all values of $\mathbf{v}_1, ..., \mathbf{v}_D$ because the corresponding minor is almost always nonzero.

The assumption stated at the beginning of this section says that for all values of $\mathbf{v}_i$ that we will be considering, the rank of B will be M. In practice, this means that the rank of B is not altered by the errors in the estimated $\mathbf{v}_i$ and that U is always $3N \times M$. This assumption does not constitute a limitation of any kind because it is verified in all practical situations.

4.2 Observation constraints

Having seen how the geometric information is used to constrain the 3D points $\mathbf{X}$, a linear system is now built from the 2D observations, that further constrains the coordinates of the reconstructed 3D points. Each observed 2D point $\mathbf{x}_m$ defines a 3D line to which the corresponding 3D point $\mathbf{X}_m$ must belong, which will be expressed by a system of affine equations that the coordinates $\mathbf{X}$ should verify.

Since $\left[\mathbf{x}_m^\top \, 1\right]^\top$ was obtained by perspective projection

$$\lambda \begin{bmatrix} \mathbf{x}_m \\ 1 \end{bmatrix} = KR_f \left(\mathbf{X}_m - \mathbf{T}_f\right) = \begin{bmatrix} \mathbf{g}_1^f & \mathbf{g}_2^f & \mathbf{g}_3^f \end{bmatrix} \left(\mathbf{X}_m - \mathbf{T}_f\right), \tag{4.5}$$

and recalling that two 3D vectors (here the left and right hand sides of this equation) are collinear if and only if their cross product is zero, one necessarily has

$$\underbrace{\begin{bmatrix} 0 & -1 & x_{m2} \\ 1 & 0 & -x_{m1} \\ -x_{m2} & x_{m1} & 0 \end{bmatrix}}_{S_m} \begin{bmatrix} \mathbf{g}_1^f & \mathbf{g}_2^f & \mathbf{g}_3^f \end{bmatrix} (\mathbf{X}_m - \mathbf{T}_f) = \mathbb{0}_{3\times 1}, \tag{4.6}$$

where the product by S_m is the Rodrigues[1] matrix of $\left[\mathbf{x}_m^\top \, 1\right]^\top$. Because this matrix has rank 2, the system of equations in Eq. (4.6) has rank 2, which expresses that only two scalars x_{m1} and x_{m2} are observed to constrain the three coordinates of $\mathbf{X}_m$.

By concatenating the equations Eq. (4.6) obtained from each point $\mathbf{x}_1, ..., \mathbf{x}_N$, one obtains a linear system of *observation constraints :*

$$A\mathbf{X} + L\mathbf{T} = \mathbb{0}_{3N\times 1}, \tag{4.7}$$

where $\mathbf{X} = \left[\mathbf{X}_1^\top \ldots \mathbf{X}_N^\top\right]^\top$, $\mathbf{T} = \left[\mathbf{T}_1^\top \ldots \mathbf{T}_F^\top\right]^\top$ and A and L are $3N \times 3N$ and $3N \times 3F$ matrices holding the coefficients that multiply elements of $\mathbf{X}$ and $\mathbf{T}$, respectively. The matrix A is block-diagonal of the form :

$$A = \begin{bmatrix} S_1 & & \\ & \ddots & \\ & & S_N \end{bmatrix} \in \mathbb{R}^{3N\times 3N},$$

where S_m is the Rodrigues matrix of $\left[\mathbf{x}_m^\top \, 1\right]^\top$, and L has the form :

$$L = \begin{bmatrix} -\mathbf{e}_{\varphi_1}^\top \otimes S_1 \\ \vdots \\ -\mathbf{e}_{\varphi_N}^\top \otimes S_N \end{bmatrix} \in \mathbb{R}^{3N\times 3F},$$

where $\varphi_m \in \{1, ..., F\}$ is the index of the image in which point $\mathbf{x}_m$ is observed and $\mathbf{e}_{\varphi_m}$ is a $F \times 1$ vector whose elements are all zero, except that in position φ_m, which is equal to one.

[1] The Rodrigues matrix of a vector $\mathbf{v} = [v_1, v_2, v_3] \in \mathbb{R}^3$ is the 3×3 matrix

$$S_\mathbf{v} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

such that for all $\mathbf{w} \in \mathbb{R}^3$, $\mathbf{v} \times \mathbf{w} = S_\mathbf{v}\mathbf{w}$.

Solving simultaneously Eqs. (4.3) and Eq. (4.7) is thus equivalent to solving :

$$AU\mathbf{V} + L\mathbf{T} = [AU \,|\, L] \begin{bmatrix} \mathbf{V} \\ \mathbf{T} \end{bmatrix} = \mathbb{0}_{3N\times 1}, \tag{4.8}$$

where $\mathbf{X} = U\mathbf{V}$, which enforces that the geometric constraints are all verified.

Although we now have a linear system whose solutions verify both the observation and geometric constraints, two important questions remain :

- Does this system define a solution that is unique up to scale and translation?
- Do errors in the observations change the nature of the set of solutions to Eq. (4.8)?

These questions are answered in the following section.

4.3 Test of unicity of the solution

As was said in the introduction, it is important to determine whether the dataset provided by the user determines a reconstruction that is unique, up to scale and translation. In this section, we give a mathematical definition for a *unique reconstruction* and show how to determine whether a given dataset defines a unique reconstruction, in a way that is insensitive to the noise in the 2D observations of the dataset.

4.3.1 Noiseless case

Before treating the real-world situation, in which the observations are subject to errors, we consider the noiseless case, in which the unicity of a reconstruction can be defined and tested in a natural way.

Definition 1 *A dataset is said to define a* unique reconstruction *if and only if there exist vectors* $\mathbf{X}^* = [\mathbf{X}_1^*; \ldots; \mathbf{X}_N^*] \in \mathbb{R}^{3N}$ *and* $\mathbf{T}^* = [\mathbf{T}_1^*; \ldots; \mathbf{T}_F^*] \in \mathbb{R}^{3F}$ *such that, for all* $\mathbf{X}$ *and* $\mathbf{T}$ *that verify simultaneously Eqs. (4.3) and (4.7), there exists a scale factor* $\lambda \in \mathbb{R}$ *and a vector* $\Delta\mathbf{T} \in \mathbb{R}^3$ *such that :*

$$\begin{array}{rcll} \mathbf{X}_m & = & \lambda\left(\mathbf{X}_m^* + \mathbf{T}_{\varphi_m}^*\right) + \Delta\mathbf{T} & (\forall m \in \{1...N\}) \textit{ and} \\ \mathbf{T}_f & = & \lambda\mathbf{T}_f^* + \Delta\mathbf{T} & (\forall f \in \{1...F\}) \end{array} \tag{4.9}$$

where $\varphi_m \in \{1...F\}$ *is the index of the image in which* $\mathbf{x}_m$ *is observed.*

In this definition, λ is the scale and $\Delta\mathbf{T}$ the translation. Note that one could have used Eq. (4.8) rather than Eqs. (4.3) and (4.7) in the definition of a unique reconstruction.

If a dataset defines a unique solution, then the existence of a unique reconstruction is equivalent to saying that all simultaneous solutions to Eqs. (4.3) and (4.7) are of the form :

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{T} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^* & | & \mathbb{1}_{N\times 1} \otimes I_3 \\ \mathbf{T}^* & | & \mathbb{1}_{F\times 1} \otimes I_3 \end{bmatrix} \begin{bmatrix} \lambda \\ \Delta\mathbf{T} \end{bmatrix},$$

where $\mathbb{1}_{N\times 1} \in \mathbb{R}^N$ is a vector whose elements are all equal to one.

If one calls $\mathbf{V}^*$ the (unique) vector of $\mathbb{R}^M$ such that $\mathbf{X}^* = U\mathbf{V}^*$, then the existence of a unique reconstruction is also equivalent to saying that all the solutions to Eq. (4.8) have the form :

$$\begin{bmatrix} \mathbf{V} \\ \mathbf{T} \end{bmatrix} = \begin{bmatrix} \mathbf{V}^* & | & \mathbb{1}_{N\times 1} \otimes I_3 \\ \mathbf{T}^* & | & \mathbb{1}_{F\times 1} \otimes I_3 \end{bmatrix} \begin{bmatrix} \lambda \\ \Delta\mathbf{T} \end{bmatrix}.$$

Since the set of the $\left[\mathbf{V}^\top \, \mathbf{T}^\top\right]^\top$ defined by this equation, for all $\lambda \in \mathbb{R}$ and $\Delta\mathbf{T} \in \mathbb{R}^3$ is a linear subspace of dimension four of $\mathbb{R}^{3N+3F}$, and since this subspace coincides with the nullspace of $[AU \,|\, L]$, the existence of a unique reconstruction implies that $[AU \,|\, L]$ has corank four. In Appendix B, we show that the converse is also true, so that, if the corank this matrix is four, then all the simultaneous solutions to Eqs. (4.3) and (4.7) have the form in Eq. (4.9). In other words, one has the following proposition :

Proposition 1 *There is a unique reconstruction if and only if the matrix $[AU \,|\, L]$ has corank four.*

This proposition affirms that, if $[AU \,|\, L]$ has corank four, then the solution is necessarily unique. That is, it cannot happen that this matrix has corank four, but the solutions to Eqs. (4.3) and (4.7) have another form than that of Eq. (4.8).

Because the rank of a matrix can be computed reliably [26], this proposition provides an easy way of determining whether a dataset defines a unique reconstruction, in the absence of noise.

If there is noise in the observations $\mathbf{x}_m$, Eqs. (4.3) and (4.7) may have no nontrivial simultaneous solution, in which case the definition of a unique reconstruction becomes useless. It is thus necessary to give a definition of the unicity of a reconstruction that is valid even in the presence of noise in the observations.

A natural way of doing so is to say that a dataset defines a unique reconstruction if, assuming the observations were obtained without noise, a unique reconstruction were defined, in the sense of Definition 1. Unfortunately, this is not of direct use, because, in order to verify whether a dataset defines a unique reconstruction, is is necessary to know the noiseless observations.

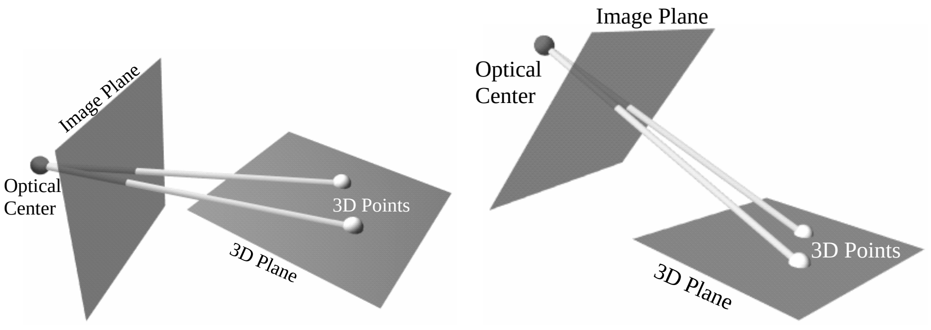


Figure 4.2: **Left:** When the optical center is contained in one of the reconstructed planes, observation of 3D points and the knowledge that they belong to the plane are not sufficient to determine their position. **Right:** If the optical center is not contained in that plane, the same information uniquely defines the positions of the points.

4.3.2 "Twin" matrices : dealing with noisy data

However, it is possible to use the new definition, at the price of a little extra work. Indeed, one can build matrices $\tilde{A}$ and $\tilde{L}$ such that $\left[\tilde{A}U \mid \tilde{L}\right]$ has the rank that $[AU \mid L]$ would have if there were no noise in the observations. Because of this property, $\left[\tilde{A}U \mid \tilde{L}\right]$ is called the "twin matrix" and is used to determine whether a unique reconstruction is defined. In this section, we explain how this matrix is built and indicate the assumptions under which this test is guaranteed to be valid.

The idea of the test comes from noting that, to a large extent, the unicity of a reconstruction does not depend on the actual positions of the 3D points and cameras or on the camera orientation and calibration. By building a collection of 3D points that verify the geometric properties and projecting these points using Eq. (4.5) (or Eq. 2.1), one obtains a collection of *noiseless observations*. These observations are then used to build matrices $\tilde{A}$ and $\tilde{L}$ in the exact same way that A and L were built in Section 4.2.

Going more in the details of the procedure, by generating a random vector $\tilde{\mathbf{V}} \in \mathbb{R}^M$ and defining $\tilde{\mathbf{X}} = U\tilde{\mathbf{V}}$, one sees that $\tilde{\mathbf{X}}$ is a vector of coordinates of 3D points that verify all the geometric constraints. This is a consequence of the columns of U forming a basis of the space of coordinates of 3D points that verify all the geometric constraints. Although U depends on the estimated dominant directions $\mathbf{v}_1, ..., \mathbf{v}_D$, we have assumed in Section 4.1 (page 45) that the configuration of these vectors is such that the errors in their estimated values do not affect the size of U.

Finally, by generating random camera positions $\tilde{\mathbf{T}}_1, ...\tilde{\mathbf{T}}_F$, all the elements are available to compute *noiseless observations* $\tilde{\mathbf{x}}_m$, using Eq. (4.5). From these observations, matrices $\tilde{A}$ and $\tilde{L}$ are obtained as in Section 4.2.

Just as we had assumed that the dominant directions do not influence the rank of B, we must now assume that the value of $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{T}}_1, ...\tilde{\mathbf{T}}_F$ do not influence the rank of $\left[\tilde{A}U \mid \tilde{L}\right]$. In geometric terms, we are excluding the situations in which small variations in $\tilde{\mathbf{X}}$ or in the $\tilde{\mathbf{T}}_f$ change the nature of the reconstruction problem. Such a situation is shown in Figure 4.2 (left). Such a situation may also occur when two cameras have the same position, while one assumes that they are distinct, e.g. because the position of a 3D point is determined only by information from stereo, i.e. from the intersection of two distinct optical rays.

Outside of these critical situations, $\left[\tilde{A}U \mid \tilde{L}\right]$ and $[AU \mid L]$ have, in the absence of noise, the same rank, and the following affirmation holds :

Proposition 2 *There is a unique reconstruction if and only if the "twin matrix"* $\left[\tilde{A}U \mid \tilde{L}\right]$ *has corank four.*

Since the rank of the twin matrix does not depend on the noisy observations, this criterion is immune to noise in the observations or in the vanishing points.

4.4 Computing a solution

We now assume the dataset defines a unique reconstruction, and will show how to obtain the coordinates of the reconstructed 3D points and camera positions.

We first obtain the reconstruction as the four-dimensional space of all reconstruction that are equal, up to scale and translation. This space cannot be obtained directly as the nullspace of $[AU \mid L]$, since, because of noise in the observations, this matrix does not have corank four[2]. Instead, we use the nullspace of the matrix with corank four that is closest, for the Frobenius norm, to $[AU \mid L]$. The space of solutions thus obtained is :

$$\left\{ \begin{bmatrix} \mathbf{V} \\ \mathbf{T} \end{bmatrix} = H\mathbf{w} \right\}_{\mathbf{w} \in \mathbb{R}^4} \tag{4.10}$$

where the columns of H are the right singular vectors of $[AU \mid L]$ corresponding to the four smallest singular values. It is often more convenient to represent the solution space by :

$$\left\{ \begin{bmatrix} \mathbf{X} \\ \mathbf{T} \end{bmatrix} = \begin{bmatrix} UH_1 \\ H_2 \end{bmatrix} \mathbf{w} \right\}_{\mathbf{w} \in \mathbb{R}^4},$$

[2] By construction, it has corank at least three, since for any $\Delta\mathbf{T} \in \mathbb{R}^3$, $[AU \mid L] \left(\mathbb{1}_{(N+F)\times 1} \otimes \Delta\mathbf{T}\right) = \mathbb{0}_{3N\times 1}$. In the presence of noise, $[AU \mid L]$ has corank exactly three.

where $H = \left[H_1^\top \, H_2^\top\right]^\top$ and H_1, H_2 have M and $3F$ rows respectively (corresponding to $\mathbf{V}$ and $\mathbf{T}$).

Until now, we have identified a reconstruction with the four-dimensional space of all reconstruction that are equal, up to scale and translation. We now show how a particular reconstruction can be obtained by imposing four additional constraints to $\mathbf{X}$ and/or $\mathbf{T}$. It is clear that there are many ways of doing so. For example, we choose the center of mass of the $\mathbf{X}_m$ to coincide with the origin of coordinates and the mean value of the norms $\|\mathbf{X}_m\|$ to be equal to one. This constraint is expressed as :

$$\underbrace{[\mathbb{1}_{1,N} \otimes I_3]\, U H_1}_{G}\, \mathbf{w} = \mathbb{0}_3, \tag{4.11}$$

where $\mathbb{1}_{1,N}$ is a row vector of ones with size $1 \times N$, I_3 is the 3×3 identity matrix and $\otimes$ is the Kronecker product. The 3×4 matrix G has rank three and all solutions to Eq. (4.11) are of the form $\lambda \mathbf{w}_0$, where λ is an arbitrary scale factor and $\mathbf{w}_0$ is any nonzero vector in the nullspace of G.

Finally, the constraint on the norm is expressed as $\lambda^2 \mathbf{w}_0^\top H_1^\top H_1 \mathbf{w}_0 = 1$, which is easily solved for λ. The desired reconstruction is then :

$$\begin{aligned} \mathbf{X} &= \lambda U H_1 \mathbf{w}_0 \quad \text{and} \\ \mathbf{T} &= \lambda H_2 \mathbf{w}_0. \end{aligned} \tag{4.12}$$

4.5 Summary of algorithm

The procedure used to determine the coherence and sufficiency of the available data, and to obtain a reconstruction can be summarized as :

1. Build the matrix B from the vanishing points $\mathbf{g}_i^f$, the planarity information and the metric information.

2. Determine a basis U for the nullspace of B.

3. Verify that no 3D point is observed twice in the same image (stop otherwise).

4. Build the matrix A from the vanishing points $\mathbf{g}_i^f$ and the observations $\mathbf{x}_m$.

5. Build $\tilde{A}$, $\tilde{L}$ and the twin matrix $\left[\tilde{A}U \,|\, \tilde{L}\right]$. Verify that it has corank equal to four (stop otherwise).

6. Compute the four-dimensional space of solutions by computing the singular vectors of $[AU \mid L]$ corresponding to the smallest singular values.

7. Compute a particular solution using other constraints.

The reconstruction method is benchmarked in Chapter 6, where its performance will be compared with that of the optimal method described in the next chapter.

Chapter 5

Maximum likelihood 3D reconstruction

Up to now, we have presented a reconstruction method that, in the absence of error in the observations, recovers exactly the original scene -up to a scale factor and a translation. However, in real-world situations, there is some error in the observations and as a consequence the reconstruction also has some error. Two important goals are to confine this error as much as possible and to gauge its amplitude. While the algebraic method presented in the last chapter is convenient for studying geometric aspects of the reconstruction problem, it is not the best for these two last requirements.

In this chapter, we define and implement the maximum likelihood method, which can be characterized in probabilistic terms and whose precision can be assessed. We first set the probabilistic framework and define the likelihood function in Section 5.1, where it is assumed that the reader is familiar with estimation theory. The maximum likelihood estimate is obtained by maximizing the likelihood function, that is, by solving an optimization problem. This is a problem of *constrained optimization*, because the estimated quantities are subject to the geometric constraints specified by the user. We choose to transform this problem into a smaller problem of unconstrained optimization by defining a differential parameterization of the feasible set.

The optimization procedure is outlined in Section 5.2 and the study of the performance of the maximum likelihood estimator is presented in Section 5.3.

5.1 Likelihood function

The maximum likelihood estimate is defined as the 3D reconstruction and camera parameters that are most likely to have produced the given observations. The exact form of the likelihood function derives from the probabilistic assumptions about the errors in the observations -the ϵ_m in the projection equation Eq. (2.1) :

$$\lambda_m \begin{bmatrix} \mathbf{x}_m \\ 1 \end{bmatrix} = KR_f \left(\mathbf{X}_m - \mathbf{T}_f \right) + \begin{bmatrix} \varepsilon_m \\ 0 \end{bmatrix}.$$

In our case, these terms are assumed to be Gaussian, independent, with zero mean and covariance :

$$\operatorname{cov}\left(\varepsilon_m\right) = \sigma^2 I_2.$$

We verify empirically this hypothesis in Section 6.1 and thus legitimize the adopted estimation approach.

With this assumption, the probability of observing points $\mathbf{x} = \left[\mathbf{x}_1^\top, ..., \mathbf{x}_N^\top\right]^\top$ (obtained by Eq. (2.1)) is :

$$P\left(\mathbf{x} \mid \bar{\mathcal{X}}\right) = \prod_{m=1}^{N} \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}\left\|\mathbf{x}_m - \mathcal{P}_m\left(\bar{\mathcal{X}}\right)\right\|^2}, \tag{5.1}$$

where $\bar{\mathcal{X}}$ (See Eq. (2.7), p. 26) holds all the true parameters $\bar{\mathbf{X}}_m$, $\bar{R}_f$, $\bar{\mathbf{T}}_f$ and $\bar{K}$, $\mathcal{P}_m(.)$ is the m^{th} observation function, defined in Eq. (2.8), and $\|.\|$ is the 2-norm [7]. This equation can also be written :

$$P\left(\mathbf{x} \mid \bar{\mathcal{X}}\right) = \left(2\pi\sigma^2\right)^{-N} e^{-\frac{1}{2\sigma^2}\left\|\mathbf{x} - \mathcal{P}(\mathcal{X})\right\|^2}, \tag{5.2}$$

where $\mathcal{P}(\mathcal{X})$ is the observation function introduced in Eq. (2.9). The mapping that associates to a given $\mathcal{X}$, $P(\mathbf{x} \mid \mathcal{X})$, is called the *likelihood function*.

The maximum likelihood estimator is the function that associates, to a given observation $\mathbf{x} \in \mathbb{R}^{2N}$, the vector $\hat{\mathcal{X}}$ that maximizes $P(\mathbf{x} \mid \mathcal{X})$ while verifying the geometric constraints. As is most often the case, we choose to minimize the inverse logarithm of $P(\mathbf{x} \mid \mathcal{X})$, which takes the more convenient form :

$$Q\left(\mathbf{x}, \mathcal{X}\right) = \frac{1}{2\sigma^2} \quad \left\|\mathbf{x} - \mathcal{P}(\mathcal{X})\right\|^2 + \text{Constant}, \tag{5.3}$$

where all the terms that do not depend on $\mathcal{X}$ are grouped in the "Constant" term. It is clear that the location of the minimum of $Q(\mathbf{x}, \mathcal{X})$ does not depend on σ, which needs not be known for the purpose of estimating $\mathcal{X}$. The maximum likelihood estimator is thus defined by :

$$\hat{\mathcal{X}}(\mathbf{x}) = \arg\min_{\mathcal{X}} Q(\mathbf{x}, \mathcal{X}) \quad \text{Subject to the geometric constraints,} \tag{5.4}$$

where the geometric constraints are the planarities, parallelism, ratios of lengths, known angles etc, defined in Sections 2.3.1, 2.3.2 and 2.3.3. The *feasible set* of this optimization problem thus consists of configurations

$$\mathcal{X} = (\mathbf{X}_1, ..., \mathbf{X}_N, R_1, ..., R_F, \mathbf{T}_1, ..., \mathbf{T}_F, K) \in \mathbb{R}^{3N+12F+9}$$

where the 3D points verify the geometric constraints, the R_f are rotation matrices and K is a calibration matrix. By counting three parameters for each 3D point and camera position and nine for each matrix R_f and for K, the feasible set is seen to be a subset of $\mathbb{R}^{3N+12F+9}$.

5.2 Solving the optimization problem

Some aspects of the optimization problem are particularly important. First, there are typically hundreds of variables to be estimated, which could lead to very high computational costs. Fortunately, the cost function is differentiable, which allows to use efficient optimization algorithms [56]. Finally, the parameters are subject to the geometric constraints specified by the user, which brings some extra difficulty.

There exist many possibilities for solving equality constrained optimization problems. One is to parameterize the estimated quantities in such a way that the mapping between the parameters and the quantities be defined on an open set, yielding an equivalent unconstrained problem that can be solved using well-known optimization tools [56]. The parameterization usually complicates the computation of the cost function, but it also reduces the dimension of the problem. Other possibilities include the "exact penalty function" method or a "feasible direction" method [59]. In the former, the solution is obtained as the limit of a sequence of solutions of optimization problems. The method of feasible directions is based on constraining the search to the directions contained in the tangent space to the feasible set, and may give a solution that is outside of the feasible set.

In the present work, it is chosen to parameterize the feasible set by a differentiable function $\mathcal{X}(\Theta)$ defined on $\mathbb{R}^P$ (for some P). For all $\Theta \in \mathbb{R}^P$, the associated configuration $\mathcal{X}(\Theta)$ belongs to the feasible set, and any configuration in the feasible set can be reached by the parameterization function. With this parameterization, the negative log-likelihood function is written :

$$Q(\mathbf{x},\Theta) \quad = \frac{1}{2\sigma^2} \quad \|\mathbf{x} - \mathcal{P}(\mathcal{X}(\Theta))\|^2 + \text{Constant}. \tag{5.5}$$

Because the function $\mathcal{X}(\Theta)$ is a surjection onto the feasible set, the original problem of finding the optimal $\mathcal{X}$ is clearly equivalent to that of finding the optimal $\Theta \in \mathbb{R}^P$. Because the function $\mathcal{X}(\Theta)$ is defined on $\mathbb{R}^P$, the reconstruction problem is transformed into an *unconstrained optimization* problem.

The question is now to find the function $\mathcal{X}(\Theta)$. This is done in the next two sections by first defining a parameterization of the 3D points and then of the camera positions, orientations and calibration.

5.2.1 Parameterization of the 3D points

The 3D parameterization of the 3D points $\mathbf{X}_1, ..., \mathbf{X}_N$ subject to the geometric constraints is to some extent analogous to the computations described in Chapter 4. First, the dominant directions are computed from some parameters $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D$:

$$\begin{array}{rcl} \mathbf{v}_1 & = & \mathbf{v}_1(\boldsymbol{\theta}_1) \\ \vdots & & \vdots \\ \mathbf{v}_D & = & \mathbf{v}_D(\boldsymbol{\theta}_D). \end{array} \tag{5.6}$$

Then, a $3N \times M$ matrix $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$ is computed from the $\mathbf{v}_i$ (and thus from the $\boldsymbol{\theta}_i$), whose columns form a basis of the space of 3D coordinates of points that verify all the geometric constraints (i.e. a basis of the nullspace of the matrix B of Chapter 4). Finally, the 3D points are obtained from this matrix and from a vector of parameters $\mathbf{V} \in \mathbb{R}^M$.

$$\mathbf{X} \quad = \quad U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)\,\mathbf{V}. \tag{5.7}$$

The main difference with respect to Chapter 4 is that, now, the dominant directions $\mathbf{v}_i$ are not fixed and $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$ should be defined as a differentiable function. The remainder of this section defines the $\mathbf{v}_i(\boldsymbol{\theta}_i)$ and $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$.

Dominant directions:

The dominant directions are computed in such a way that the geometric constraints imposed to them by the user (Section 2.3.3, p. 30) are verified. We assume that the directions are numbered so that they can be computed sequentially, each one being defined by one of the following rules :

Known direction: $\mathbf{v}_i = \text{Known}$. E.g. $\mathbf{v}_1 = [1, 0, 0]^\top$. No parameter is needed.

Arbitrary direction: $\mathbf{v}_i = \boldsymbol{\theta}_i / \|\boldsymbol{\theta}_i\|$ for some $\boldsymbol{\theta}_i \in \mathbb{R}^3 \setminus \{\mathbb{O}\}$.

Fixed angle: $\mathbf{v}_i$ is constrained to form a known angle α with some previously computed $\mathbf{v}_j$ (i.e. $j < i$). The direction is computed from a vector $\boldsymbol{\theta} \in \mathbb{R}^3 \setminus \{\lambda \mathbf{v}_j \mid \lambda \in \mathbb{R}\}$, which is projected orthogonally onto the 3D circle $\left\{\mathbf{v} \mid \mathbf{v}^\top \mathbf{v}_j = \cos(\alpha), \|\mathbf{v}\| = 1\right\}$.

Coplanarity: $\mathbf{v}_i$ is coplanar to some previously computed $\mathbf{v}_j$ and $\mathbf{v}_k$ ($j < i$, $k < i$ and $j \neq k$). This is the same as fixing the angle between $\mathbf{v}_i$ and $\mathbf{v}_j \times \mathbf{v}_k$ to $\pi/2$.

Two orthogonalities: $\mathbf{v}_i$ is constrained to be orthogonal to two non-collinear directions $\mathbf{v}_j$ and $\mathbf{v}_k$ ($j < i$, $k < i$ and $j \neq k$) and can thus be computed by $\mathbf{v}_i = \mathbf{v}_j \times \mathbf{v}_k / \|\mathbf{v}_j \times \mathbf{v}_k\|$. No parameter is thus required.

One first notes that no parameter $\boldsymbol{\theta}_i$ is needed for known directions and for directions defined by a cross-product, so that these parameters can be omitted from the implementation. For simplicity[1], we keep these useless $\boldsymbol{\theta}_i$ in our notation and assume e.g. that they belong to a single-element set $\{0\}$. Also, since a direction $\mathbf{v}_i$ may depend on the previously computed directions $\mathbf{v}_j$, $j < i$, it is more rigorous to write $\mathbf{v}_i(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_i)$.

Some details of the computation of the $\mathbf{v}_i(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_i)$, most notably that of a direction forming a fixed angle with another direction, are given in Appendix C.1. The differentials $\frac{\partial}{\partial \boldsymbol{\theta}_j} \mathbf{v}_i(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_i)$ (for $1 \leq j \leq i$), are also given there.

Beyond being differentiable, one should note that the $\mathbf{v}_i(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_i)$ are not in general injective functions, since distinct parameters $\boldsymbol{\theta}_i$ may yield the same directions $\mathbf{v}_i$. For example, if $\mathbf{v}_i(\boldsymbol{\theta}_i) = \boldsymbol{\theta}_i / \|\boldsymbol{\theta}_i\|$, is an arbitrary direction, then for all $\boldsymbol{\theta}_i \in \mathbb{R}^3 \setminus \{\mathbb{O}\}$ and all $\lambda \in \mathbb{R} \setminus \{0\}$, one has $\mathbf{v}_i(\boldsymbol{\theta}_i) = \mathbf{v}_i(\lambda \boldsymbol{\theta}_i)$.

[1] Alternatively, if $D' \leq D$ direction parameters are actually used, one could consider the parameters $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{D'}$, where $\boldsymbol{\theta}_i$ is the parameter that defines the direction $\mathbf{v}_{\mu_i}$ and, for all $i \in \{1, ..., D'\}$, $\mu_i \in \{1, ..., D\}$ is the index of the direction defined by $\boldsymbol{\theta}_i$. Keeping all the $\boldsymbol{\theta}_i$ in our notation avoids having to define an extra sequence μ_i.

Finally, the initial values for the $\boldsymbol{\theta}_i$, required by the optimization algorithm, are given by the dominant directions obtained in Chapter 3.

Constrained points:

Having parameterized the dominant directions $\mathbf{v}_i$, we only need to define $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$ before using Eq. (58) to obtain the 3D points. The computation of the $3N \times M$ matrix $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$ is done in essence like in Chapter 4, where the matrix of geometric constraints B (Eq. (44)) is built from the $\mathbf{v}_i$ (and thus from the $\boldsymbol{\theta}_i$) and $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$ is taken to be an orthonormal basis of its nullspace. However, some provisions must be taken to ensure that the resulting function is differentiable, and that its differential can in practice be computed.

The main difficulty in defining $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$ comes from the fact that the nullspace of B usually has infinitely many orthonormal bases. Indeed, if U is a $3N \times M$ matrix whose columns form an orthonormal base of the nullspace of B and O is an arbitrary $M \times M$ orthogonal matrix, then the columns of UO also form an orthonormal base of the nullspace of B.

In order to uniquely define a function $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$, its value at some point $\left(\boldsymbol{\theta}_1^0, ..., \boldsymbol{\theta}_D^0\right)$ is fixed to an arbitrary matrix U_0 whose columns form an orthonormal base of the nullspace of $B\left(\boldsymbol{\theta}_1^0, ..., \boldsymbol{\theta}_D^0\right)$ and the value of $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$ at other points is defined by :

$$U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D) = \arg\min_U \left\{ \|U - U_0\|_F \mid U \text{ forms an orthonormal basis of } \mathcal{N}ull\left(B(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)\right) \right\}, \tag{5.8}$$

where $\|.\|_F$ is the Frobenius norm. The proof that this procedure defines a differentiable function on a neighborhood of $\left(\boldsymbol{\theta}_1^0, ..., \boldsymbol{\theta}_D^0\right)$ is given in Appendix E, together with practical means to compute $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$ and its differential. One aspect that will be important later, is that the differentials $\frac{\partial}{\partial \boldsymbol{\theta}_i} U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$ are simpler to compute at $\left(\boldsymbol{\theta}_1^0, ..., \boldsymbol{\theta}_D^0\right)$ than at other points.

As a final note, one must indicate how to choose U_0. Since any matrix whose columns form an orthonormal base of the nullspace of $B\left(\boldsymbol{\theta}_1^0, ..., \boldsymbol{\theta}_D^0\right)$ is adequate, we use the one returned by the `null()` function of “Octave” [22], the matrix computation language that we use.

To summarize the two last sections, we have presented a differentiable parameterization of a collection of 3D points subject to geometric constraints of the type defined in Section 2.3 and may now turn to the parameterization of the other estimated quantities.

5.2.2 Parameterization of the camera positions, orientations and calibration

In this section, we outline the parameterization of camera orientations, positions and calibration by differentiable mappings.

The orientation R_f of each camera is parameterized separately by a mapping $R_f\left(\varphi_f\right)$ defined on $\mathbb{R}^3$. There exist many possible ways to represent a rotation matrix [70] and we use a vector $\varphi_f \in \mathbb{R}^3$ that is collinear to the axis of the rotation, with norm equal to its amplitude in radians :

$$R_f\left(\varphi_f\right) = \tilde{\varphi}_f\tilde{\varphi}_f^\top + \cos\left(\tau_f\right)\left(I_3 - \tilde{\varphi}_f\tilde{\varphi}_f^\top\right) - \sin\left(\tau_f\right)S_{\tilde{\varphi}_f}, \tag{5.9}$$

where $\tau_f = \left\|\varphi_f\right\|$, $\tilde{\varphi}_f = \varphi_f / \left\|\varphi_f\right\|$ and $S_{\tilde{\varphi}_f}$ is the Rodrigues matrix of $\tilde{\varphi}_f$. Since Eq. (5.9) is not defined at $\varphi_f = \mathbb{0}$, we take $R_f\left(\mathbb{0}\right) = I_3$ and show in Appendix C.2 that the resulting mapping is differentiable.

It is also shown in that appendix that the differential of this function takes a particularly simple form in $\varphi_f = \mathbb{0}$, and for this reason we take a provision to ensure that we never need to compute the differential except in that case. Instead of using Eq. (5.9) directly, we left-multiply this expression by a previous estimate of the rotation matrix, called R_0^f. The actual mapping is thus :

$$\begin{array}{rcll} R_f\left(\mathbb{0}\right) & = & R_0^f & \\ R_f\left(\varphi_f\right) & = & R_0^f\left(\tilde{\varphi}_f\tilde{\varphi}_f^\top + \cos\left(\tau_f\right)\left(I_3 - \tilde{\varphi}_f\tilde{\varphi}_f^\top\right) - \sin\left(\tau_f\right)S_{\tilde{\varphi}_f}\right) & \text{if}\, \varphi_f \neq \mathbb{0}, \end{array} \tag{5.10}$$

where $\tau_f = \left\|\varphi_f\right\|$, $\tilde{\varphi}_f = \varphi_f / \left\|\varphi_f\right\|$ and $S_{\tilde{\varphi}_f}$ is the Rodrigues matrix of $\tilde{\varphi}_f$. Having parameterized the camera orientations, the F camera positions are and the camera calibration are parameterized by :

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 \\ \vdots \\ \mathbf{T}_F \end{bmatrix} \in \mathbb{R}^{3F} \text{ and } \mathbf{K} = \begin{bmatrix} \log\left(\rho\right) \\ u_0 \\ v_0 \end{bmatrix} \in \mathbb{R}^3, \tag{5.11}$$

where $[u_0, v_0]$ is the principal point of the camera and ρ its focal length, which is thus guaranteed to be positive. The differentials of these two mappings are trivial.

By concatenating the parameters of the mappings above in a single vector $\Theta = [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D, \mathbf{V}, \varphi_1, ..., \varphi_F, \mathbf{T}, \mathbf{K}] \in \mathbb{R}^P$ and by joining the mappings defined in Eqs. (5.6-5.8, 5.10 and 5.11), one obtains a parameterization $\mathcal{X}\left(\Theta\right)$ of the feasible set of the original optimization problem. This mapping, being built from differentiable mappings, is differentiable and its differential can be obtained by using the chain rule.

Also of importance, the mapping $\mathcal{X}(\Theta)$ depends on some fixed quantities U_0 and $R_0^1, ..., R_0^F$ that will be set so that computing the differential of $\mathcal{X}(\Theta)$ is facilitated.

5.2.3 Modified Levenberg-Marquardt algorithm

We assume that the reader is familiar with the Levenberg-Marquardt algorithm, whose details can be found in many textbooks [59, 56]. The listing below outlines the algorithm, to which a step -called "re-parameterize"- has been added. During that step, the parameterization of the feasible set is changed by setting the values of R_0^f and U_0 to the current best values, and $\boldsymbol{\varphi}_f$ are set to $\mathbb{0}$.

```
Initialization: Initialize $\Theta$, $U_0$ and the $R_0^f$ and evaluate $\mathcal{P}(\Theta)$.

Outer loop: While some stopping condition is not met :

    Evaluate derivatives: Evaluate $\frac{\partial}{\partial\Theta}\mathcal{P}(\Theta)$ at the current best $\Theta$.

    Inner loop: While some stopping condition is not met, search amongst a set of possible $\Theta$ :

        Evaluate function: One evaluation of $\mathcal{P}(\Theta)$ is needed for each evaluation of $Q(\mathbf{x}, \Theta)$.

    Re-parameterize: Set $U_0 := U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$ and $R_0^f := R_f(\boldsymbol{\varphi}_f)$ where the $\boldsymbol{\theta}_i$ and $\boldsymbol{\varphi}_f$ are
        the current best values, and then set $\boldsymbol{\varphi}_f := \mathbb{0}$.
```

One can check that these steps do not change the values of the $R_f(\boldsymbol{\varphi}_f)$ or of $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$, so that the value of $Q(\mathbf{x}, \Theta)$ is left unchanged. Since the differential of the $R_f(\boldsymbol{\varphi}_f)$ and of $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$ take simplified forms at $\boldsymbol{\varphi}_f = \mathbb{0}$ and $U(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D) = U_0$, and the change of parameterization occurs just before the differential of $\mathcal{X}(\Theta)$ is be computed, only the simplified forms of the differentials need to be evaluated.

The innocuity of changing the parameterization of the feasible set at each outer loop comes from the fact that Levenberg-Marquardt algorithm discards at each step all previous information about the minimized function except the current best parameters and the value of the cost function at that point. At the end of each iteration of the outer loop, we are thus free to change the optimized function, since only the information from the new function will be used during the next iteration.

Having presented the main practical aspects of the implementation of the maximum likelihood estimator, we turn to the theoretical study of its precision.

5.3 Analytical covariance of the ML estimator

It is shown in [34] that if the function $Q(\mathbf{x}, \Theta)$ has isolated minima, then one can get analytical expressions for the covariance of the estimator. This is also the case when the minima $Q(\mathbf{x}, \Theta)$ are not isolated, but one has a differentiable function of constraints $S(\Theta)$, such that the minima of $Q(\mathbf{x}, \Theta)$ in the set $\{\Theta \mid S(\Theta) = \mathbb{0}\}$ are isolated.

As we already noted (Section 2.2) that $\mathcal{P}(\Theta)$ remain unchanged if the camera positions $\mathbf{T}$ and 3D points $\mathbf{X}$ are translated and/or scaled and that the parameterization of the dominant directions (Section 5.2.1) is not injective, a suitable constraint function $S(\Theta)$ is needed, which we define below. Restricting the study to the set $\{\Theta \mid S(\Theta) = \mathbb{0}\}$ in effect constitutes a *normalization* of the parameters.

Then, in Section 5.3.2, expressions for the covariance are given, which are derived in Section D or, in a slightly different form, in [34].

5.3.1 Parameter normalization

In the present work, the 3D points are normalized so that their center of mass coincides with the origin of coordinates and their average norm $\|\mathbf{X}_m\|$ is one. These requirements are expressed by the equations :

$$\begin{aligned} \left(\mathbb{1}_{1\times N} \otimes I_3\right)\mathbf{X} &= \mathbb{0}_{3\times 1} \text{ and} && (5.12) \\ \mathbf{X}^\top \mathbf{X} - N &= 0. && (5.13) \end{aligned}$$

In order to render the functions $\mathbf{v}_i(\boldsymbol{\theta}_i)$ injective, their domains are restricted to coincide with their images. Only the $\mathbf{v}_i(\boldsymbol{\theta}_i)$ that actually require a parameter are considered, i.e. those that define an arbitrary direction, or that forms a fixed angle α with another direction $\mathbf{v}_j$. In the first case, its domain of definition is defined on the unit sphere $\left\{\boldsymbol{\theta}_i \in \mathbb{R}^3 \mid \boldsymbol{\theta}_i^\top \boldsymbol{\theta}_i = 1\right\}$, and in the second case, on the circle $\left\{\boldsymbol{\theta}_i \in \mathbb{R}^3 \mid \boldsymbol{\theta}_i^\top \boldsymbol{\theta}_i = 1,\, \boldsymbol{\theta}_i^\top \mathbf{v}_j = \cos(\alpha)\right\}$[2]. In both cases, the parameter $\boldsymbol{\theta}_i$ belongs to the respective domain of definition if and only if it verifies the equation :

$$\boldsymbol{\theta}_i = \mathbf{v}_i. \qquad (5.14)$$

[2] Note that this restriction of the domain of definition is done only in order to allow the theoretical study of the estimator. For the purpose of actually optimizing the cost function, the original domain of definition can still be used.

By joining Eqs. 5.12, 5.13 and the Eqs. 5.14 (if any) arising from parameterized directions, one obtains a system of equations of the form $S(\Theta) = \mathbb{0}$ such that the function $\mathcal{P}(\Theta)$ restricted to the set $\left\{\Theta \in \mathbb{R}^P \mid S(\Theta) = \mathbb{0}\right\}$ is injective. Because the function $S(\Theta)$ is moreover differentiable, it is possible to study analytically the covariance of the maximum likelihood estimator using the tools presented by Haralick, e.g. in [34]. The rest of this chapter outlines this study, which is detailed in Appendix D.

5.3.2 Analytical expressions of the covariance

We assume that there exists a set of "true" parameters $\bar{\Theta}$ such that the observations $\mathbf{x}$ were obtained by Eq. (2.1) and that the maximum likelihood estimate $\hat{\Theta}$ was obtained by minimizing $Q(\mathbf{x}, \Theta)$, defined in Eq. (5.5). By doing some first-order approximations (Appendix D), and calling $\Delta\Theta = \bar{\Theta} - \hat{\Theta}$ the error in the estimate , one obtains :

$$\Delta\Theta \simeq \mathcal{U}\left(\mathcal{U}^\top H \mathcal{U}\right)^{-1} \mathcal{U}^\top F \varepsilon \,, \tag{5.15}$$

where $\mathcal{U}$ is an orthogonal basis of $\mathcal{N}ull\left(\frac{\partial}{\partial\Theta} S\left(\hat{\Theta}\right)\right)$, $F = \frac{\partial}{\partial\Theta}\mathcal{P}\left(\hat{\Theta}\right)/\sigma^2$ and $H = \frac{\partial^2}{\partial\Theta\Theta} Q\left(\mathbf{x}, \hat{\Theta}\right)$. The derivation of this result can be found in Appendix D, p. 92. This expression shows that, in a first approximation, the expectation of $\Delta\Theta$ is zero; that is, the maximum likelihood estimator is unbiased. From Eq. (5.15), one immediately gets :

$$\operatorname{cov}(\Delta\Theta) \simeq \hat{\Sigma} = \mathcal{U}\left(\mathcal{U}^\top H \mathcal{U}\right)^{-1} \mathcal{U}^\top, \tag{5.16}$$

The two expressions above are obtained under the assumption of small error in the observation, and are otherwise valid independently of the error model on the observations, which need not be Gaussian.

Chapter 6

Experimental results

This chapter describes the experiments that sustain the theoretical study of the two previous chapters. First (Section 6.1), the assumption that, in real-world situations, the errors introduced in the observations $\mathbf{x}_m$ are Gaussian is verified. Then (Section 6.2) the validity of the theoretical study of the covariance of the maximum likelihood estimator is verified empirically, using simulation. Section 6.3 then benchmarks the algebraic reconstruction method defined in Chapter 4 and the maximum likelihood method of Chapter 5. Finally, Section 6.4 displays some real-world reconstructions and the precision obtained in these cases.

6.1 Nature of the error in the observations

In this section, the assumption that the error in the observations is approximately Gaussian is verified experimentally by inspecting the histogram of the residues obtained from real-world reconstructions.

Probability model for the residues: At the maximum likelihood estimate $\hat{\Theta}$, the residue $\mathbf{x}_m - \mathcal{P}_m\left(\hat{\Theta}\right)$ can be approximated by :

$$\begin{aligned}\mathbf{x}_m - \mathcal{P}_m\left(\hat{\Theta}\right) &\simeq \bar{\mathbf{x}}_m + \varepsilon_m - \left(\bar{\mathbf{x}}_m + \frac{\partial}{\partial\Theta}\mathcal{P}_m\left(\hat{\Theta}\right)\Delta\Theta\right)\\ &= \varepsilon_m - \frac{\partial}{\partial\Theta}\mathcal{P}_m\left(\hat{\Theta}\right)\Delta\Theta,\end{aligned}$$

where $\Delta\Theta = \bar{\Theta} - \hat{\Theta}$ is the error in the estimate and $\bar{\mathbf{x}}_m = \mathcal{P}_m\left(\bar{\Theta}\right)$ is the noiseless projection of the "true" n^{th} point. Grouping all the $\mathbf{x}_m$ and using Eq. 5.15, one gets :

$$\mathbf{x} - \mathcal{P}\left(\hat{\Theta}\right) \simeq \varepsilon - F\mathcal{U}\left(\mathcal{U}^\top H\mathcal{U}\right)^{-1}\mathcal{U}^\top F\varepsilon$$

$$= \underbrace{\left(I - F\mathcal{U}\left(\mathcal{U}^\top H \mathcal{U}\right)^{-1}\mathcal{U}^\top\right)}_{\Pi_{F^\top}} F\varepsilon.$$

Noting that $\Pi_{F^\top}$ is a projector, the "light" singular value decomposition of this matrix can be written :

$$\Pi_{F^\top} = \mathcal{V}\mathcal{V}^\top,$$

where $\mathcal{V}$ is a rectangular unitary matrix.

The mean and covariance of the "*normalized residues*" $\mathbf{y} = \mathcal{V}^\top \left(\mathbf{x} - \mathcal{P}\left(\hat{\Theta}\right)\right)$ are easily shown to be :

$$\begin{aligned} E(\mathbf{y}) &= \mathbb{O} \\ \operatorname{cov}(\mathbf{y}) &\simeq \sigma^2 I, \end{aligned}$$

so that each element of $\mathbf{y}$ is an independent normal Gaussian random variable.

In order to gather residues from real-world situations, each of the setups that will be presented in Section 6.4 were used. The normalized residues $\mathbf{y}$ were computed and the covariance estimated by $\hat{\sigma}^2 = N^{-1}\|\mathbf{y}\|^2$. The elements of $\hat{\sigma}^{-1}\mathbf{y}$ are thus, if our model is correct, normal Gaussian random variables.

From these elements, the empirical probability density function can be obtained by doing a histogram. The histogram, obtained by grouping the 347 scaled residues in 24 bins (Figure 6.1, left), shows that the empirical probability function is approximately normal. The rather good fit between the empirical and assumed probability models justify the choice of the Gaussian probability model. Most notably, the absence of outliers suggests that the sum-of-squares function Q (Eq. 5.3, p. 56) is appropriate. If this were not the case, it would have been necessary to use a robust [39, 61] version of Q.

6.2 Theoretical and empirical covariance of estimators

Since some approximation was involved in the derivation of Eq. (5.16), we must verify the validity of this expression. In this section, using synthetic data for which the ground truth is known, we compare the empirical error and that predicted by Eq. (5.16).

A setup consisting of 27 points disposed on a $3 \times 3 \times 3$ irregular grid is built (Figure 6.2). The camera orientation is given by a rotation matrix whose axis is is given by a $N(\mathbb{O}, I_3)$ random

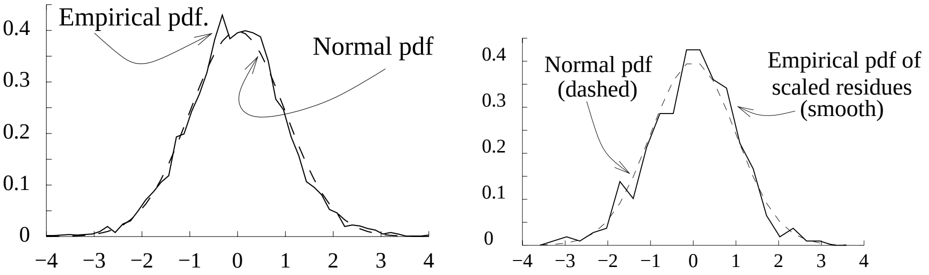


Figure 6.1: **Left:** Histogram of scaled residues obtained from the real-world reconstructions in Section 6.4. The good correspondence between the histogram and the Gaussian pdf shows that the error, is approximately Gaussian.
Right: Histogram of the normalized errors in the parameters obtained from simulations in which the errors on the observations are Gaussian.

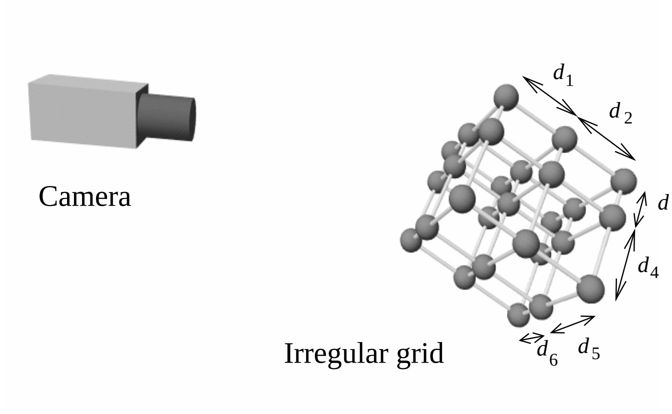


Figure 6.2: Points on an irregular grid and camera position. The lengths d_i of the edges of the grid are uniform random variables in $[0.5, 1]$. The object is then translated and scaled so that its center of mass be $[0\,0\,0]^\top$ and the mean norm of its vertices be 1. The world origin has coordinates $[T_1\,T_2\,5]^\top$ in the camera frame, where the T_i are $N\,(0, 0.5)$ random variables.

vector, whose norm provides the angle of the rotation. The observations are obtained using a focal-length-only camera (Eq. (2.5)). The natural logarithm of the focal length is a $N(0, 0.1)$ random variable. In all, the vector of parameters has 16 elements : nine for the 3D points, three for the camera orientation, three for the camera position and one calibration parameter.

Gaussian white noise of 1% (40 dB) is added to the observations. The maximum likelihood estimator is started from the vector of true parameters to which 3.2% (30 dB) of Gaussian white noise is added. The covariance of the estimator is estimated using Eq. (5.16). Assuming this approximation is valid, one should have $\Delta\Theta \sim N(\mathbb{O}, \hat{\Sigma})$ and thus the random variable $\Psi = \left(\hat{\Sigma}^{+}\right)^{\frac{1}{2}} \Delta\Theta$ should be $\Psi \sim N(\mathbb{O}, I)$, that is, each element of Ψ is a $N(0,1)$ random variable. We call Ψ the vector of *normalized errors*.

Four hundred (400) such experiments were done. The 6400 (400×16) resulting normalized errors were observed to have mean 0.0212 and covariance 1.05. Figure 6.1 shows the histogram of the normalized errors with the superposed probability density function of a $N(0,1)$ variable. The good correspondence of the theoretical and empirical probability density functions indicates that the approximations that have been used to obtain Eq. (5.16) are valid for the considered noise levels. Also, they confirm that the implementation of the maximum likelihood estimator is correct.

6.3 Maximum likelihood vs. algebraic method

In this section, we compare the precision of the maximum likelihood estimator and the algebraic method, using synthetic data. The same setup as in Figure 6.2 (Section 6.2, page on page 66) was used. The level of error in the observations was made to vary between 0.0 percent (no noise) and 3.12 percent (30dB). In real-world situations, the noise levels are usually between 0.3 percent (50dB) and 1.0 percent (40dB). We prefer to measure the error in percents (%) or decibels, rather than in pixels, so that these figures be independent of the size of the image.

For each error level, the algebraic and maximum likelihood reconstruction methods were run 50 times and the mean error in the results is considered. Figure 6.3 shows the curves of the error in the estimated 3D points (left), camera orientation (middle) and position (right). The error in the 3D points and camera positions is measured in percents of the mean amplitude of these quantities. The choice of reporting errors as percentage rather than absolute value allows these results to be invariant to the scale of the reconstruction. The error in the camera orientation is measured by the mean angle, in degrees, between the columns of the true and the estimated orientation matrices

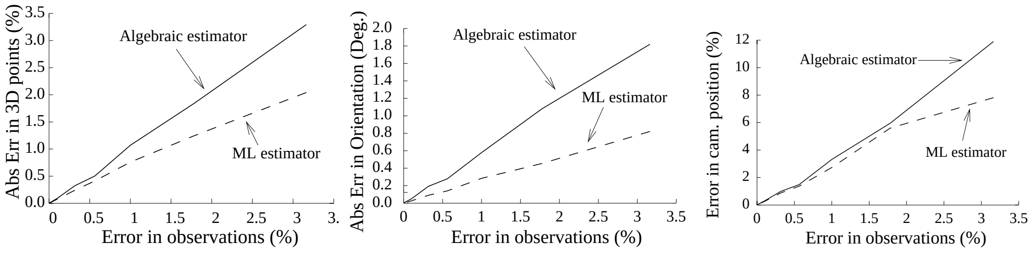


Figure 6.3: Error in the reconstruction process as a function of the error in the observations : error in the estimated 3D points (**left**), the orientation of the cameras (**middle**, measured in degrees) and the position of the cameras (**right**).

Quantity (unit)	**X** (%)	Orient. (deg.)	**T** (%)	$\log(\rho)$ (%)	Residue (%)	N	P
Mont Saint-Michel	5.49	0.49	13.4	3.6	0.86	114	180
Tour Eiffel	0.52	0.21	30.1	4.03	0.52	70	56
Folkemuseum	1.53	0.31	21.4	3.84	0.88	122	131
Conciergerie	2.05	0.26	2.90	1.68	0.62	72	119
Hall	3.01	0.12	0.30		1.23	61	78

Table 6.1: Theoretical error in the real-world reconstructions. The first column identifies the dataset, and the remaining ones contain the standard deviation of the error in the 3D points, camera orientations, positions and focal length.

(R_f, in Eq. (2.1), p. 24).

Figure 6.2 shows that both estimators behave well when noise becomes very high and give exact results in the absence of noise. Also, the precision of the maximum likelihood estimator, except for camera location, is clearly superior to that of the algebraic method. The relatively high error in the location of the camera is in accordance with our previous findings [32, 33] on the precision of 3D reconstruction of general scenes.

6.4 Architectural examples

In this section, we present some results obtained from real-world images. The figures display maximum likelihood reconstructions , showing the aspect of reconstructions; the amplitude of the residues is indicated in the text, giving an idea of the quality of the fit and justifying the assertion that the error level in the observations lies in the range 40-50 dB (1.00-0.32%). In most cases, at least one vanishing point is close to infinity, so that we can at best estimate the focal length.

For each dataset in this section, the covariance of the estimator is estimated using Eq. (5.16), p. 64. Table 6.1 shows the standard deviation of the error in the reconstructions, for the 3D points, camera orientation, position and focal length.

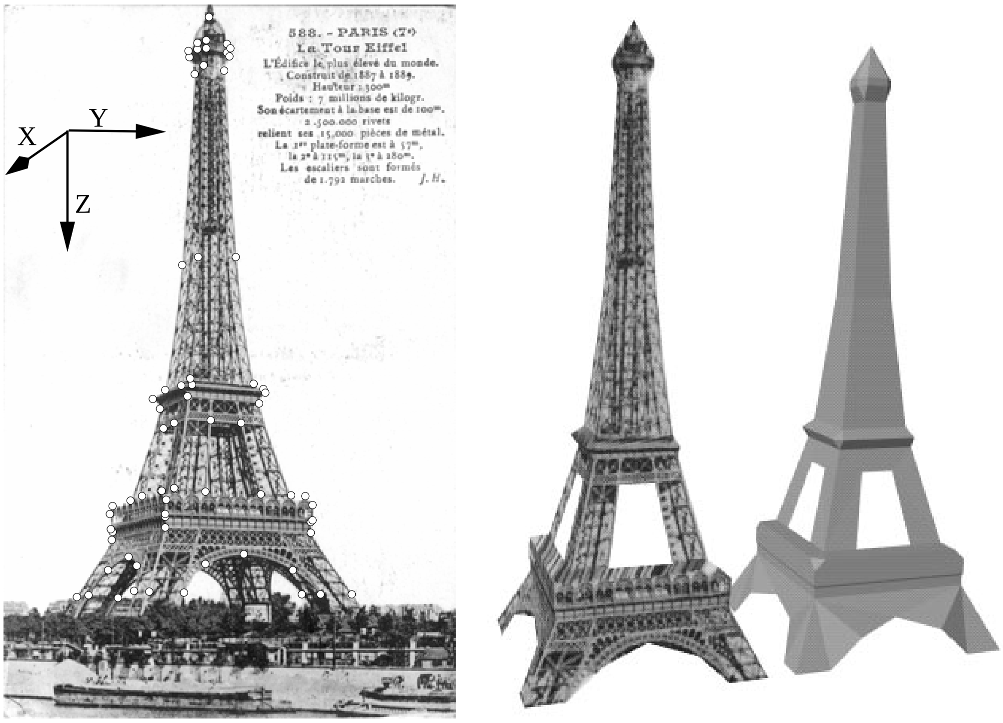

588. - PARIS (7e)
La Tour Eiffel
L'Édifice le plus élevé du monde.
Construit de 1887 à 1889.
Hauteur : 300m
Poids : 7 millions de kilogr.
Son écartement à la base est de 100m.
2.500.000 rivets
relient ses 15,000 pièces de métal.
La 1re plate-forme est à 57m,
la 2e à 115m, la 3e à 180m.
Les escaliers sont formés
de 1.792 marches. J. H.
X
Y
Z

Figure 6.4: Outdoors scene (**left**) and reconstruction (**right**). 70 image features were identified. Geometric information consists in 56 planes and 43 distance equalities. Some distances are equal due to the symmetry of the object around a vertical line passing through the topmost point. In all, there the object has 49 degrees of freedom.

6.4.1 Tour Eiffel

This example illustrates how symmetry can be used to obtain a uniquely defined solution that would not be achievable otherwise. Figure 6.4 (left) shows an image with superposed image features. Horizontal coplanarity relations are easily identified, together with vertical coplanarities amongst points on the same horizontal plane.

Few vertical are present, so that points at different heights cannot be connected. In order to obtain the reconstruction, it is necessary to use the symmetry of the object.

Altogether, there are 77 points, 56 planes and 45 known length ratios; the object has 49 degrees of freedom. Figure 6.4 (right) shows the reconstruction, with and without texture. The amplitude of the residues is 39.5 dB (1.05%) in the algebraic reconstruction and 45.7 dB (0.52%) in the ML reconstruction.

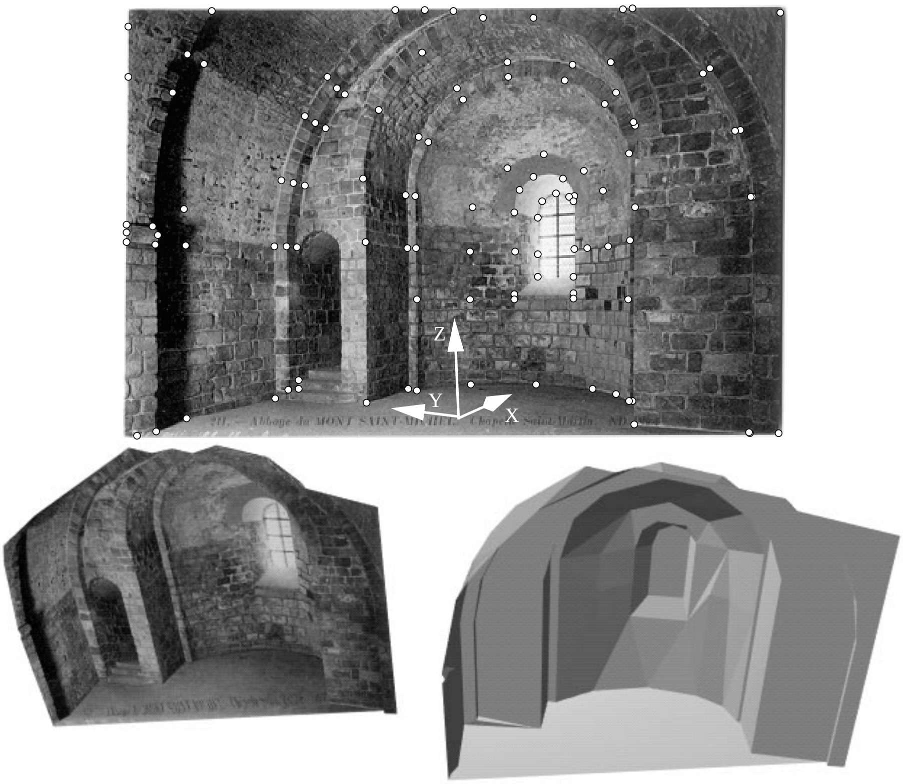


Figure 6.5: **Top:** Indoor scene with 114 identified points. **Bottom:** Reconstruction obtained from these points and 39 known planes; shown with and without texture.

6.4.2 Mont Saint-Michel

Figure 6.5 (top) shows an indoor image with superposed image features. Horizontal coplanarity relations hold for points on the floor, for points on the wall that can be connected along a horizontal row of bricks and for some points on the window. Vertical planes are identified on the walls and arcades. Around the window some coplanarity were tentatively identified. Altogether, 114 points and 39 planes were identified. No known ratios of distance between pairs of planes were used. The scene has 173 degrees of freedom. Figure 6.5 (bottom) shows the maximum likelihood reconstruction obtained from this data. In this model, the principal point of the camera was estimated. The residues have an amplitude of 44 dB (0.63%) in the algebraic reconstruction and 47.5 dB (0.42%) in the maximum likelihood reconstruction.

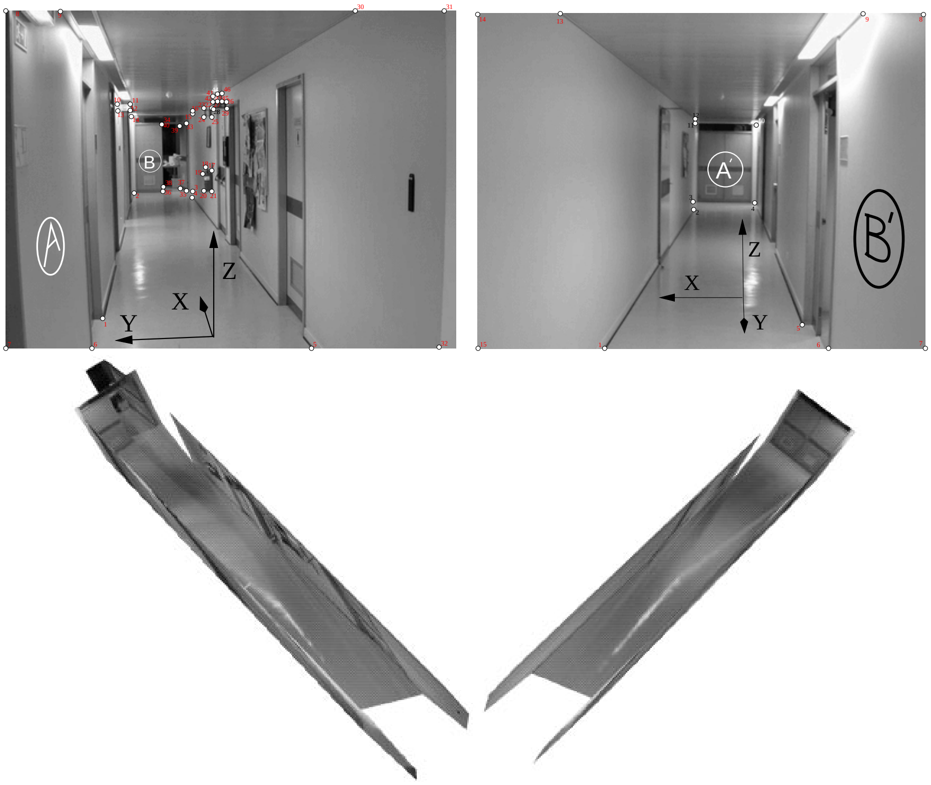


Figure 6.6: Left: Two indoor images. No 2D point is tracked between them Right : reconstruction.

6.4.3 Folkemuseum

Figure 1.1 (left), on page 2 shows an image in which 122 points were identified. The geometric information consists in 75 planes and 26 known length ratios. The model has 124 degrees of freedom. The resulting maximum likelihood reconstruction is shown in Figure 1.1 (right); the residues in the image of the reprojection of the algebraic and maximum likelihood reconstructions have amplitudes of 40.8 dB (0.91%) and 41.1 dB (0.88%) respectively.

6.4.4 Multiple-view indoors

Figure 6.6 shows two indoor images taken from approximately the same point, but in nearly perpendicular directions. The input consists in 61 points, 35 planes and one known ratio of lengths : the distance from the point marked “A” in the first image to that marked “A-prime” in the second image is equal to that from point “B” (first image) to point “B-prime” second image. Note that the dataset defines a unique reconstruction even though no 3D point being visible in both images. No intrinsic parameter can be reliably estimated because two vanishing points are almost at infinity,

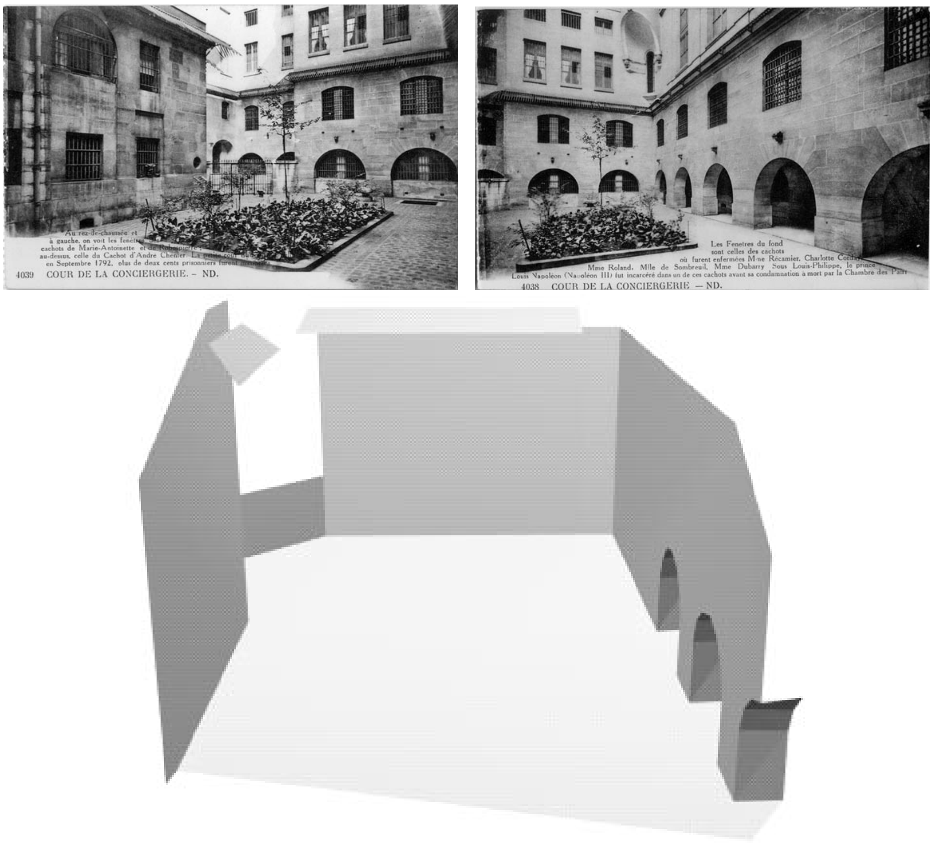

4039 COUR DE LA CONCIERGERIE. - ND.
4038 COUR DE LA CONCIERGERIE — ND.

Figure 6.7: Top: Two outdoor images. Bottom : Untextured reconstruction

so that the reconstruction is defined up to an affine transformation. The reconstruction shown in Figure 6.6 was obtained by assuming that the focal length is one. The error level of reprojected points with respect to observed points are 41.9 (0.81 %) for the algebraic estimator and 38.2 dB (1.23 %) for the maximum likelihood estimators. The greater residues in that last case are due to the treatment of the vanishing points, which are constrained to be mutually orthogonal in the maximum likelihood estimator, whereas the vanishing points are used unchanged in the original algebraic reconstruction.

6.4.5 Multiple-view outdoors

Figure 6.7 (top) shows two outdoors image with some overlap. Seventy-two (24 in the first image, 48 in the second) points and 21 planes are identified; two known length ratios are given, to express that the spikes on the walls stick out by the same amount on the left and front wall (without this information, the reconstruction would not be unique). The reconstruction is shown in the bottom.

The error levels of reprojected points with respect to observed points are 46.9 dB (0.45%) and 44.1 dB (0.62%) for the algebraic and the maximum likelihood reconstructions.

Chapter 7

Conclusions

We have addressed the problem of reconstruction of structured scenes from image points when geometric information is available a priori. Two complementary reconstruction methods were developed, that use the same 2D points and geometric information as input. The algebraic method identifies the reconstruction problem with a problem of linear algebra and the maximum likelihood method computes the "best possible" reconstruction from the available data

We have shown that linear constraints can express geometric constraints of a more general type than was used previously. By summarizing all the available information in a single linear system, a clear link is made between the original reconstruction problem and a problem of linear algebra. By building a "twin" linear system that is not altered by noise in the observations, we showed that it is possible to determine whether a dataset defines a unique reconstruction. Finally, it was shown possible to implement a constraint-based method whose output, not only is the maximum likelihood reconstruction, but also has a known precision.

The proposed reconstruction method was tested in real-world situations. The assumption of Gaussian noise in the observations was experimentally confirmed and the reconstruction was benchmarked using synthetic data. Finally, the precision obtained in real-world situations was shown.

Many extensions of the presented reconstruction method appear possible. Some have already been explored [29, 31, 30] and were not included in this thesis because their interest is somewhat peripheral :

- Line observations can be used in addition to 2D points. This is very simple in the algebraic method [29, 31] because a line observation is easily transformed into a linear constraint.

However, using image edges rather than points in the maximum likelihood method requires a different error model [21, 74], and a change in the cost function.

- It is possible to detect the presence of many unconnected objects [31, 30] whose relative scale cannot be determined (Figure 1.5, left).
- It should be possible to give geometric information about the camera, e.g. by saying that the camera is at a constant height, has constant velocity or that it is watching an object on a turntable.
- The computational cost could be reduced. Indeed, the size of the optimization problem can be reduced without changing the final result.
- Finally, the method could benefit from a more general calibration procedure, such as that proposed in [76], or one based on matched points along images [35].

Another possible application of this thesis concerns the parameterization of constrained 3D points that was developed in Section 5.2.1. It should be noted that this parameterization is independent of the reconstruction process, since it only depends on the geometric constraints. It could thus be used to define parameterized shapes on the fly from geometric constraints, to be integrated in a model-based reconstruction system, with the aim of giving it the flexibility of constraint-based systems.

In conclusion, we have developed a reconstruction method based on 2D observations and geometric information whose output is the maximum likelihood reconstruction and has a known precision, and can thus be used as a statistically characterized measuring tool.

Appendix A

Maximum likelihood vanishing point estimation

In this section, we study the maximum likelihood estimator of a vanishing point when sets of aligned points are given. It is assumed that independent identically distributed Gaussian noise is added to these observations. We first define the likelihood function over $\mathbb{R}^2$, then extend it to the projective plane (including points at infinity) and finally show how to find the maximum likelihood estimate.

A.1 Input data and assumptions

We assume the input consists in M lines; line number i is formed by points $\mathcal{X} = \left(\mathbf{x}_1^i, \ldots, \mathbf{x}_{N_i}^i\right)$. One assumes that observations are subject to noise : $\mathbf{x}_j^i = \mathbf{x}_j^{*i} + \epsilon_j^i$ where the ϵ_j^i are i.i.d. Gaussian variable with zero mean and covariance $\sigma^2 \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}$ for some σ, and $\mathbf{x}_j^{*i}$ are the noiseless observations. The noiseless lines all intersect in the true vanishing point $\mathbf{g}^*$.

A.2 Likelihood function

From the observation model, the maximum likelihood estimate of $\mathbf{g}^*$, given the observation, is the point $\mathbf{g}$ such that there exist M lines $\mathbf{l}_1, \ldots, \mathbf{l}_M$ that contain $\mathbf{g}$, that minimize the negative log-likelihood

$$q(\mathbf{g}, \mathcal{X}) = \sum_{i=1}^{M} \min_{\mathbf{l}_j} \sum_{j=1}^{N_1} d\left(\mathbf{l}_i, \mathbf{x}_j^i\right)^2 .$$

Finding the minimum of this function is simplified by the fact that, for all $\mathbf{g}$ and all sets of points $\mathbf{x}_1, \ldots, \mathbf{x}_N$, the line passing through $\mathbf{g}$ that minimizes the sum of squared distances

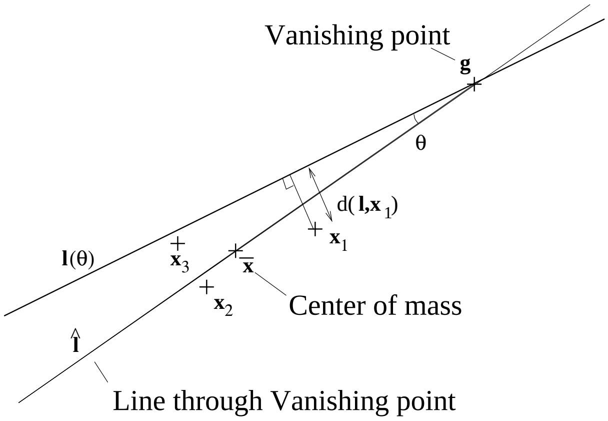


Figure A.1: Data points $\mathbf{x}_i$, their center of mass $\bar{\mathbf{x}}$, proposed vanishing point $\mathbf{g}$, line $\mathbf{l} = \mathbf{g} \times \bar{\mathbf{x}}$ passing through $\mathbf{g}$ and $\bar{\mathbf{x}}$ and line $\mathbf{l}(\theta)$ forming an angle $-\theta$ with $\mathbf{l}$. The line $\mathbf{l}(\theta)$ that minimizes the sum of squared distances $d(\mathbf{l}(\theta), \mathbf{x}_i)$ is either $\mathbf{l}(0)$ or $\mathbf{l}(\pi/2)$.

$\sum_{j=1}^{N} d\left(\mathbf{l}, \mathbf{x}_j\right)^2$is either the line $\mathbf{l} = \mathbf{g} \times \bar{\mathbf{x}}$, where $\bar{\mathbf{x}}$ is the centroid of the $\mathbf{x}_i$, or the line $\mathbf{l}'$ passing through $\mathbf{g}$ and orthogonal to $\mathbf{l}$.

This property can be proven by considering the function

$$f(\theta) = \sum_{j=1}^{N} d\left(\mathbf{l}(\theta), \mathbf{x}_j\right)^2$$

where $\mathbf{l}(\theta)$ is the line (see Figure A.1) passing through $\mathbf{g}$ and forming an angle θ with the line $\mathbf{g} \times \bar{\mathbf{x}}$. This function is periodic and it is sufficient to show that it is minimized at $\theta \in \{k\pi/2 | k \in \mathbb{N}\}$. The squared distance from the j^{th} point $\mathbf{x}_j$ to $\mathbf{l}(\theta)$ is :

$$\begin{aligned} d\left(\mathbf{l}(\theta), \mathbf{x}_j\right)^2 &= K^2 \left(\underbrace{(\mathbf{g} - \mathbf{x}_j)^\top}_{\mathbf{w}_j} \underbrace{\begin{bmatrix} & -1 \\ 1 & \end{bmatrix}}_{E} \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \underbrace{(\mathbf{g} - \bar{\mathbf{x}})}_{\bar{\mathbf{w}}} \right)^2 \\ &= K^2 \left(c\mathbf{w}_j^\top E \bar{\mathbf{w}} - s\mathbf{w}_j^\top \bar{\mathbf{w}} \right)^2 \end{aligned}$$

where $c = \cos(\theta)$, $s = \sin(\theta)$ and $K = ||\bar{\mathbf{x}} - \mathbf{g}||^{-1}$. The derivative of f, scaled by K^{-2} is then :

$$\begin{aligned} K^{-2} f'(\theta) &= 2 \textstyle\sum_j \left(-s\mathbf{w}_j^\top E \bar{\mathbf{w}} - c\mathbf{w}_j^\top \bar{\mathbf{w}} \right) \left(c\mathbf{w}_j^\top E \bar{\mathbf{w}} - s\mathbf{w}_j^\top \bar{\mathbf{w}} \right) \\ &= \sin(2\theta) \textstyle\sum_j \left(\left(\mathbf{w}_j^\top \bar{\mathbf{w}}\right)^2 - \left(\mathbf{w}_j^\top E \bar{\mathbf{w}}\right)^2 \right) + \\ & \quad 2\cos(2\theta) \, \underbrace{\mathrm{Tr} \left(\sum_j \left(\mathbf{w}_j \mathbf{w}_j^\top \right) \bar{\mathbf{w}} \bar{\mathbf{w}}^\top E \right)}_{0} \end{aligned}$$

The coefficient of $\cos(2\theta)$ is seen to be zero by noting that $\sum_j \left(\mathbf{w}_j\mathbf{w}_j^\top\right)\bar{\mathbf{w}}\bar{\mathbf{w}}^\top$ is symmetric and that for any symmetric matrix $\left[\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right]$ one has $\mathrm{Tr}\left(\left[\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right]\left[\begin{smallmatrix} & -1 \\ 1 & \end{smallmatrix}\right]\right) = b - b = 0$.

The minimum of f is thus for $\theta \in \{k\pi/2\}_{k\in}$. The corresponding values of f are thus

$$\begin{array}{rclcl} f(0) & = & f(\pi) & = & K^2 \sum_i \left(\mathbf{w}_j^\top E\bar{\mathbf{w}}\right)^2 \\ & & & = & K^2 N \bar{\mathbf{w}}^\top E^\top \bar{V}_x E \bar{\mathbf{w}} \end{array} \tag{A.1}$$

and

$$\begin{array}{rclcl} f\left(\frac{\pi}{2}\right) & = & f\left(\frac{3\pi}{2}\right) & = & K^2 \sum_i \left(\mathbf{w}_j^\top \bar{\mathbf{w}}\right)^2 \\ & & & = & K^2 N \left(\left(\bar{\mathbf{w}}^\top\bar{\mathbf{w}}\right)^2 + \bar{\mathbf{w}}^\top \bar{V}_x \bar{\mathbf{w}}\right). \end{array} \tag{A.2}$$

where $\bar{V}_x = N^{-1}\sum_j (\bar{\mathbf{x}} - \mathbf{x}_j)(\bar{\mathbf{x}} - \mathbf{x}_j)^\top$. In order to determine which value is smallest, one may evaluate the two expression Eq. (A.1) and (A.2). Also, one may note that $f(\frac{\pi}{2}) > f(0)$ if $\|\bar{\mathbf{w}}\|^2 > \left\|E^\top \bar{V}_x E\right\|$. The second norm is itself majored by $v_{xx} + v_{yy} + |v_{xy}|$, where v_{xx}, v_{yy} and v_{xy} are the elements of $\bar{V}_x = \begin{bmatrix} v_{xx} & v_{xy} \\ v_{xy} & v_{yy} \end{bmatrix}$.

Finally, it should be noted that if at the optimal vanishing point, obtained from two or more segments, the minimum at $\theta = \pi/2$ for one or more segments, this corresponds to an abnormal situation. Indeed, in that case, the centroid of the segment is near the optimal vanishing point, which usually indicates a local minimum has been reached or that some of the input 2D points do not correspond to a 3D segment.

Going back to the original problem of minimizing q, it is now clear that q can be written as a function of the two coordinates of $\mathbf{g}$ only.

$$q(\mathbf{g}, \mathcal{X}) = \sum_{i=1}^{M} \min\left\{f_i(0), f_i\left(\frac{\pi}{2}\right)\right\}$$

where $f_i(.)$ is the function defined from $\mathbf{g}$ and the points $\mathbf{x}_1^i, \ldots, \mathbf{x}_{N_i}^i$, like f above.

A.3 Extension of Q to the projective plane

We now extend the domain of the function $q(., \mathcal{X}) : \mathbb{R}^2 \to \mathbb{R}^+$ to $\mathbb{R}^3 \setminus \{\mathbb{0}_{3\times1}\}$, where $\mathbb{R}^3 \setminus \{\mathbb{0}_{3\times1}\}$ represents points in the projective plane. This offers the advantage that finite and infinite vanishing points are represented alike.. One defines :

$$\begin{array}{rcll} q'\left([v_1\, v_2\, v_3]^\top, \mathcal{X}\right) & = & q\left(v_3^{-1}[v_1\, v_2]^\top, \mathcal{X}\right) & \text{if}\, v_3 \neq 0, \\ & = & \sum_{i=1}^{M}\sum_{j=1}^{N_i}\left([v_1\, v_2]\, E\left(\mathbf{x}_j^i - \bar{\mathbf{x}}_i\right)\right)^2 \|[v_1\, v_2]\|^{-1} & \text{if}\, v_3 = 0, \end{array}$$

where $[v_1\, v_2\, v_3] \in \mathbb{R}^3 \setminus \{\mathbb{0}_{3\times 1}\}$ represents a point in the projective point plane. One easily verifies that this function is continuous on its domain.

A.4 Optimization issues

Finding the maximum likelihood vanishing points amounts to finding $\mathbf{g}' = [v_1\, v_2\, v_3] \in \mathbb{R}^3 \setminus \{\mathbb{0}_{3\times 1}\}$ that minimizes $q'(\mathbf{g}', \mathcal{X})$. There are many possible ways of doing so. Although globally convergent [38] or numerically efficient methods could have been chosen, we preferred the Nelder-Mead [56] algorithm, which is well-known and easily implemented.

In order to avoid falling in a minimum other than the global, algorithm, the starting point $\mathbf{g}_0$ is chosen amongst 626 points spread on half the unit sphere, in such a way that all points are within 5.75 degrees from a proposed starting point. Assuming that the optimal vanishing point is not altogether perpendicular to the starting point, we use a local parameterization centered on the starting point. The minimized function is defined on the tangent plane to the unit sphere in $\mathbf{g}_0$, defined by :

$$q''([\mu_1\, \mu_2]) = q'(\mathbf{g}_0 + \mu_1 \mathbf{g}_1 + \mu_2 \mathbf{g}_2, \mathcal{X}),$$

where $\mathbf{g}_1$, $\mathbf{g}_2$ are unit vectors such that $\mathbf{g}_0^\top \mathbf{g}_1 = \mathbf{g}_0^\top \mathbf{g}_2 = \mathbf{g}_1^\top \mathbf{g}_2 = 0$. Minimizing this function with the Nelder-Mead algorithm proved entirely satisfactory.

The precision of this method for estimating vanishing points is studied in Section 3.1.1, where benchmarking is done using synthetic data.

Appendix B

Algebraic method

In this chapter, we prove of Properties 1 and 2. We call $\bar{\mathbf{X}}_m$ the true 3D points, $\bar{\mathbf{v}}_i$ the true dominant directions, $\bar{R}_f$, $\bar{\mathbf{T}}_f$ and $\bar{K}$ the true camera orientations, positions and calibration, and $\bar{\mathbf{x}}_m$ the noiseless observations. The matrices of observation and geometric constraints obtained from these noiseless quantities will likewise be written $\bar{A}$, $\bar{L}$ and $\bar{B}$, and $\bar{U}$ will denote an orthonormal matrix whose columns form a basis of the nullspace of $\bar{B}$. The letters A, B etc designate, as in Chapter 4, matrices obtained from possibly (but not necessarily) noisy observations and dominant directions.

B.1 Demonstration of Proposition 1

In this section, we prove Proposition 1, from Section 4. In short, we show that there is a rigid reconstruction if and only if the matrix $\begin{bmatrix} \bar{A} & \bar{L} \\ \bar{B} & \mathbb{O} \end{bmatrix}$ has same corank four, and then that this matrix has same corank as $\left[\bar{A}\bar{U} \mid \bar{L}\right]$, so that there is a rigid reconstruction if and only if this last matrix has corank four.

One easily sees that Definition 1 is equivalent to the following :

Definition 1' *A dataset defines a rigid reconstruction if and only if there exist vectors* $\mathbf{X}^* = \left[\mathbf{X}_1^{*\top} \ldots \mathbf{X}_N^{*\top}\right]^\top \in \mathbb{R}^{3N}$ *and* $\mathbf{T}^* = \left[\mathbf{T}_1^{*\top}, \ldots, \mathbf{T}_F^{*\top}\right] \in \mathbb{R}^{3F}$ *such that :*

$$\mathcal{N}ull \begin{bmatrix} \bar{A} & \bar{L} \\ \bar{B} & \mathbb{O} \end{bmatrix} = \left\{ \begin{bmatrix} \mathbf{X} \\ \mathbf{T} \end{bmatrix} \middle| \exists \lambda \in \mathbb{R}, \exists \Delta\mathbf{T} \in \mathbb{R}^3, \begin{bmatrix} \mathbf{X} \\ \mathbf{T} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{X}^* \\ \mathbf{T}^* \end{bmatrix} + \begin{bmatrix} \Delta\mathbf{T} \\ \vdots \\ \Delta\mathbf{T} \end{bmatrix} \right\}. \tag{B.1}$$

This second form clearly implies that, if there is a rigid reconstruction, then $\begin{bmatrix} \bar{A} & \bar{L} \\ \bar{B} & \mathbb{0} \end{bmatrix}$ has corank four. We now show that the converse is also true; that is, if the matrix has corank four, then its nullspace has the particular form of the right hand side of Eq. (B.1).

We show that, if one takes $\mathbf{X}^* = \bar{\mathbf{X}}$ and $\mathbf{T}^*$ and $\bar{\mathbf{T}}$, where $\bar{\mathbf{X}}$ and $\bar{\mathbf{T}}$ are the true coordinates of 3D points and cameras, respectively, then the set

$$\mathcal{S} = \left\{ \begin{bmatrix} \mathbf{X} \\ \mathbf{T} \end{bmatrix} \middle| \exists \lambda \in \mathbb{R}, \exists \Delta\mathbf{T} \in \mathbb{R}^3, \begin{bmatrix} \mathbf{X} \\ \mathbf{T} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{X}^* \\ \mathbf{T}^* \end{bmatrix} + \mathbb{1}_{(N+F)\times 1} \otimes \Delta\mathbf{T} \right\}$$

has dimension four, and that it is always included in $\mathcal{N}ull \begin{bmatrix} \bar{A} & \bar{L} \\ \bar{B} & \mathbb{0} \end{bmatrix}$.

First, since the true 3D points and camera positions verify the observation and geometric constraints, it is clear that

$$\begin{bmatrix} \bar{A} & \bar{L} \\ \bar{B} & \mathbb{0} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{X}} \\ \bar{\mathbf{T}} \end{bmatrix} = \mathbb{0}.$$

Then, one notes that the coplanarity and ratios of lengths are invariant by translation. That is, the geometric constraints, Eq. (4.1) and (4.2), do not change if $\mathbf{X}$ is translated. For any $\Delta\mathbf{T} \in \mathbb{R}^3$ and any $\mathbf{X}$ that verifies $\bar{B}\mathbf{X} = \mathbb{0}$, one has

$$\bar{B} \left(\mathbf{X} + \begin{bmatrix} \Delta\mathbf{T} \\ \vdots \\ \Delta\mathbf{T} \end{bmatrix} \right) = \mathbb{0}.$$

Note that this holds not only for the noiseless matrix $\bar{B}$, so that one has :

$$B \begin{bmatrix} \Delta\mathbf{T} \\ \vdots \\ \Delta\mathbf{T} \end{bmatrix} = B \left(\mathbb{1}_{N\times 1} \otimes I_3 \right) \Delta\mathbf{T} = \mathbb{0}. \tag{B.2}$$

Likewise, the projection of 3D points is invariant by simultaneous translation of camera(s) and points. Thus the observation constraints, Eq. (4.7) are invariant by simultaneous translation of $\mathbf{X}$ and $\mathbf{T}$, so that, for any $\Delta\mathbf{T} \in \mathbb{R}^3$ and any $\mathbf{X}$ and $\mathbf{T}$ that verify $\bar{A}\mathbf{X} + \bar{L}\mathbf{T} = \mathbb{0}$, one has

$$\bar{A} \left(\mathbf{X} + \left(\mathbb{1}_{N\times 1} \otimes \Delta\mathbf{T} \right) \right) + \bar{L} \left(\mathbf{T} + \left(\mathbb{1}_{F\times 1} \otimes \Delta\mathbf{T} \right) \right) = \mathbb{0}.$$

Again, this holds not only for the noiseless matrices $\bar{A}$ and $\bar{L}$ and one has :

$$[A\,|\,L]\begin{bmatrix}\Delta\mathbf{T}\\ \vdots\\ \Delta\mathbf{T}\end{bmatrix}=[A\,|\,L]\left(\mathbb{1}_{(N+F)\times 1}\otimes\Delta\mathbf{T}\right)=\mathbb{0}. \tag{B.3}$$

From Eqs. (B.2) and (B.3), it is clear that the set

$$\bar{\mathcal{S}}=\left\{\begin{bmatrix}\mathbf{X}\\ \mathbf{T}\end{bmatrix}\;\middle|\;\exists\lambda\in\mathbb{R},\,\exists\Delta\mathbf{T}\in\mathbb{R}^3,\,\begin{bmatrix}\mathbf{X}\\ \mathbf{T}\end{bmatrix}=\lambda\begin{bmatrix}\bar{\mathbf{X}}\\ \bar{\mathbf{T}}\end{bmatrix}+\left(\mathbb{1}_{(N+F)\times 1}\otimes\Delta\mathbf{T}\right)\right\}$$

is included in $\mathcal{N}ull\begin{bmatrix}\bar{A}&\bar{L}\\ \bar{B}&\mathbb{0}\end{bmatrix}$. Moreover, this set has dimension four, because it contains $\mathcal{S}pan\,\left(\mathbb{1}_{(N+F)\times 1}\otimes I_3\right)$[1], which has dimension three, and because

$$\begin{bmatrix}\bar{\mathbf{X}}\\ \bar{\mathbf{T}}\end{bmatrix}\notin\mathcal{S}pan\left(\mathbb{1}_{(N+F)\times 1}\otimes I_3\right),$$

since this would imply that $\bar{\mathbf{X}}_1=...=\bar{\mathbf{X}}_N=\bar{\mathbf{T}}_1=...=\bar{\mathbf{T}}_F$, which we assumed (Section 2.3) not to be the case.

This achieves the proof that $\bar{\mathcal{S}}\subset\mathcal{N}ull\begin{bmatrix}\bar{A}&\bar{L}\\ \bar{B}&\mathbb{0}\end{bmatrix}$ and that $\bar{\mathcal{S}}$ has dimension four. Thus, if $\mathcal{N}ull\begin{bmatrix}\bar{A}&\bar{L}\\ \bar{B}&\mathbb{0}\end{bmatrix}$ has dimension four, this set necessarily coincides with $\bar{\mathcal{S}}$, which has the form needed in Eq. (B.1). □

B.2 Demonstration of Proposition 2

Finally, we show that $\mathcal{N}ull\begin{bmatrix}\bar{A}&\bar{L}\\ \bar{B}&\mathbb{0}\end{bmatrix}$ and $\mathcal{N}ull\left[\bar{A}\bar{U}\,|\,\bar{L}\right]$ have same dimension, by showing that there exists a linear bijection from one to the other : consider the application

$$\begin{bmatrix}\mathbf{V}\\ \mathbf{T}\end{bmatrix}\in\mathcal{N}ull\left[\bar{A}\bar{U}\,|\,\bar{L}\right]\longrightarrow\begin{bmatrix}\bar{U}\mathbf{V}\\ \mathbf{T}\end{bmatrix}\in\mathcal{N}ull\begin{bmatrix}\bar{A}&\bar{L}\\ \bar{B}&\mathbb{0}\end{bmatrix}.$$

[1] Since it contains $\mathbb{1}_{(N+F)\times 1}\otimes\Delta\mathbf{T}$ for all $\Delta\mathbf{T}\in\mathbb{R}^3$.

This function is well defined because, if $\begin{bmatrix} \mathbf{V} \\ \mathbf{T} \end{bmatrix} \in \mathcal{N}ull\left[\bar{A}\bar{U} \,|\, \bar{L}\right]$, then $\begin{bmatrix} \bar{U}\mathbf{V} \\ \mathbf{T} \end{bmatrix} \in \mathcal{N}ull \begin{bmatrix} \bar{A} & \bar{L} \\ \bar{B} & \mathbb{0} \end{bmatrix}$. Also, this function is injective because the columns of $\bar{U}$ are linearly independent. Finally, it is surjective because any $\mathbf{X} \in \mathcal{N}ull\left(\bar{B}\right)$ can be written as $\mathbf{X} = \bar{U}\mathbf{V}$ for $\mathbf{V} = \bar{U}^\top \mathbf{X}$, and one has

$$\begin{aligned}
\mathbb{0} = \begin{bmatrix} \bar{A} & \bar{L} \\ \bar{B} & \mathbb{0} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{T} \end{bmatrix} \quad &\Longleftrightarrow \quad \mathbb{0} = \begin{bmatrix} \bar{A} & \bar{L} \\ \bar{B} & \mathbb{0} \end{bmatrix} \begin{bmatrix} \bar{U}\bar{U}^\top \mathbf{X} \\ \mathbf{T} \end{bmatrix} \\
&\Longleftrightarrow \quad \mathbb{0} = \begin{bmatrix} \bar{A}\bar{U} & \bar{L} \\ \bar{B}\bar{U} & \mathbb{0} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{T} \end{bmatrix} \\
&\Longleftrightarrow \quad \mathbb{0} = \begin{bmatrix} \bar{A}\bar{U} & \bar{L} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{T} \end{bmatrix}, \qquad \text{(By def. of } \bar{U}\text{)}
\end{aligned}$$

so that any vector $\left[\mathbf{X}^\top \, \mathbf{T}^\top\right]^\top$ in $\mathcal{N}ull \begin{bmatrix} \bar{A} & \bar{L} \\ \bar{B} & \mathbb{0} \end{bmatrix}$ has an inverse image in $\mathcal{N}ull\left[\bar{A}\bar{U} \,|\, \bar{L}\right]$. □

This achieves the proof of Proposition 2.

Appendix C

Parameterization of the estimated quantities

This appendix gives some details on the parameterization of the estimated quantities defined in Sections 5.2.1 and 5.2.2. The parameterization of the dominant directions $\mathbf{v}_i$ is treated in Appendix C.1, and that of the rotation matrices R_f in Appendix. C.2.

C.1 Parameterization of dominant directions

In this section, we complete the description of the parameterization of a collection of constrained directions. Section 5.2.1 presents in sufficient detail the computational procedure except for the case of a direction $\mathbf{v}_i$ constrained to have a fixed angle α with another direction $\mathbf{v}_j$, which will be treated now. Then, in Appendix C.1.2, the differential of all directions with respect to the parameters $\boldsymbol{\theta}_i$ is derived.

As was said in Section 5.2.1, a vector $\mathbf{v}_i$ verifying $\mathbf{v}_j^\top \mathbf{v}_i = \cos(\alpha)$ and $\mathbf{v}_i^\top \mathbf{v}_i = 1$ is computed from a vector $\boldsymbol{\theta}_i \in \mathbb{R}^3 \backslash \{\lambda \mathbf{v}_j \mid \lambda \in \mathbb{R}\}$ by projecting $\boldsymbol{\theta}_i$ onto the 3D circle $\mathcal{C} = \left\{\mathbf{v} \mid \mathbf{v}_j^\top \mathbf{v} = \cos(\alpha), \mathbf{v}^\top \mathbf{v} = 1\right\}$ in such a way that the distance $\|\mathbf{v}_i - \boldsymbol{\theta}_i\|$ is minimized. The exact procedure is :

1. Set $\mathbf{u}_1 := \left(I_3 - \mathbf{v}_j \mathbf{v}_j^\top\right) \boldsymbol{\theta}_i$.
2. Set $\mathbf{u}_2 := \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}$.
3. Set $\mathbf{v}_i := \cos(\alpha)\, \mathbf{v}_j + \sin(\alpha)\, \mathbf{u}_2$.

The resulting vector $\mathbf{v}_i$ verifies $\mathbf{v}_i^\top \mathbf{v}_i = 1$ and $\mathbf{v}_j^\top \mathbf{v}_i = \cos(\alpha)$, and can be shown to minimize the distance to $\boldsymbol{\theta}_i$ amongst all the points on $\mathcal{C}$.

C.1.1 Parameter normalization

If the vector θ that parameterizes the dominant directions has nonzero dimension, the parameterization given above is everywhere many-to-one : the $\mathbf{v}_i$ are unchanged if the $\boldsymbol{\theta}_l$ are scaled; in the case of directions forming a fixed angle with a direction $\mathbf{v}_j$, $\mathbf{v}_i$ depends only on the plane formed by $\mathbf{v}_j$ and $\boldsymbol{\theta}_l$, not on the location of $\boldsymbol{\theta}_l$ in this plane.

We will need in Section 5.3 to define a function $S_d(\boldsymbol{\theta})$ such that $h(\boldsymbol{\theta})$ is one-to-one on the set $\{\boldsymbol{\theta} \,|\, S_d(\boldsymbol{\theta}) = \mathbb{O}\}$. A natural choice for $S_d(\boldsymbol{\theta})$ is

$$S_d(\boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{\theta}_1 - \mathbf{v}_{i_1} \\ \vdots \\ \boldsymbol{\theta}_{D'} - \mathbf{v}_{i_{D'}} \end{bmatrix},$$

where, for any $l \in \{1, ..., D'\}$, i_l is the smallest index of a direction $\mathbf{v}_{i_l}$ in whose computation $\boldsymbol{\theta}_l$ is used. We will say that parameters $\boldsymbol{\theta}$ that verify $S_d(\boldsymbol{\theta}) = \mathbb{O}$ are *normalized.*

C.1.2 Differential of the $\mathbf{v}_i(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_i)$

We now indicate how the differentials $\frac{\partial}{\partial \boldsymbol{\theta}_j} \mathbf{v}_i(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_i)$ are computed.

First, the differential $\frac{\partial}{\partial \boldsymbol{\theta}_i} \mathbf{v}_i$ is defined for the directions in which a parameter $\boldsymbol{\theta}_i$ is needed, i.e. the arbitrary directions and those that must form a fixed angle with a previously defined direction. For an arbitrary direction $\mathbf{v}_i = \boldsymbol{\theta}_i \|\boldsymbol{\theta}_i\|^{-1}$ defined by a parameter $\boldsymbol{\theta}_i \in \mathbb{R}^3 \setminus \{\mathbb{O}\}$, one has :

$$\frac{\partial}{\partial \boldsymbol{\theta}_i} \left(\frac{\boldsymbol{\theta}_i}{\|\boldsymbol{\theta}_i\|} \right) = \frac{1}{\|\boldsymbol{\theta}_i\|} \left(I - \frac{1}{\|\boldsymbol{\theta}_i\|^2} \boldsymbol{\theta}_i \boldsymbol{\theta}_i^\top \right). \tag{C.1}$$

If the parameter $\boldsymbol{\theta}_i$ is normalized (Sec. C.1.1), i.e. if $\boldsymbol{\theta}_i = \mathbf{v}_i$, and $\|\boldsymbol{\theta}_i\| = 1$, this expression becomes :

$$\frac{\partial}{\partial \boldsymbol{\theta}_i} \mathbf{v}_i = I - \mathbf{v}_i \mathbf{v}_i^\top.$$

For a direction $\mathbf{v}_i$ forming an angle α with $\mathbf{v}_j$, one has ,

$$\frac{\partial}{\partial \boldsymbol{\theta}_i} \mathbf{v}_i = \frac{\sin(\alpha)}{\|\mathbf{u}_1\|} \left(I - \mathbf{u}_2 \mathbf{u}_2^\top - \mathbf{v}_j \mathbf{v}_j^\top \right). \tag{C.2}$$

where the $\mathbf{u}_i$ are those defined at the beginning of Section C.1. If $\boldsymbol{\theta}_i$ is normalized, this expression simplifies to :

$$\frac{\partial}{\partial \boldsymbol{\theta}_i} \mathbf{v}_i = \left(I - \mathbf{u}_2 \mathbf{u}_2^\top - \mathbf{v}_j \mathbf{v}_j^\top \right).$$

For a direction $\mathbf{v}_i$ that forms a fixed angle with a direction $\mathbf{v}_j$, or is defined by the cross product of two directions $\mathbf{v}_j$ and $\mathbf{v}_k$, the derivatives of $\mathbf{v}_i$ with respect to the other parameters $\boldsymbol{\theta}_j$ for $1 \leq j < i$ are not all zero. These are computed by considering $\mathbf{v}_i$ as a function of $\mathbf{v}_j$ (and eventually $\mathbf{v}_k$), computing $\frac{\partial}{\partial \mathbf{v}_j}\mathbf{v}_i$ (and eventually $\frac{\partial}{\partial \mathbf{v}_k}\mathbf{v}_i$) and using the chain rule to obtain the $\frac{\partial}{\partial \boldsymbol{\theta}_j}\mathbf{v}_i$ from the $\frac{\partial}{\partial \boldsymbol{\theta}_l}\mathbf{v}_j$ $(1 \leq l < j)$ and eventually the $\frac{\partial}{\partial \boldsymbol{\theta}_l}\mathbf{v}_k$ $(1 \leq l < k)$.

For a direction $\mathbf{v}_i$ forming an angle α with $\mathbf{v}_j$,

$$\frac{\partial}{\partial \mathbf{v}_j}\mathbf{v}_i = \cos(\alpha) I - \frac{\sin(\alpha)}{\|\mathbf{u}_1\|}\left(\left(\mathbf{v}_j^\top \boldsymbol{\theta}_i\right)\left(I - \mathbf{u}_2 \mathbf{u}_2^\top\right) + \mathbf{v}_j \boldsymbol{\theta}_i^\top\right) \tag{C.3}$$

where the $\mathbf{u}_i$ are defined as above. If the parameters are normalized, this expression simplifies to :

$$\frac{\partial}{\partial \mathbf{v}_j}\mathbf{v}_i = \cos(\tau)\, \mathbf{u}_2 \mathbf{u}_2^\top.$$

In the case of a direction $\mathbf{v}_i$ defined by the cross product of directions $\mathbf{v}_j$ and $\mathbf{v}_k$, one has :

$$\begin{aligned} \frac{\partial}{\partial \mathbf{v}_j}\mathbf{v}_i &= -\frac{1}{\sin(\mathbf{v}_j, \mathbf{v}_k)}\left(I - \mathbf{v}_i \mathbf{v}_i^\top\right) S_{\mathbf{v}_k} \quad \text{and} \\ \frac{\partial}{\partial \mathbf{v}_k}\mathbf{v}_i &= \frac{1}{\sin(\mathbf{v}_j, \mathbf{v}_k)}\left(I - \mathbf{v}_i \mathbf{v}_i^\top\right) S_{\mathbf{v}_j}., \end{aligned} \tag{C.4}$$

where $S_{\mathbf{v}_j}$ is the Rodrigues matrix and $\sin(\mathbf{v}_j, \mathbf{v}_k)$ is the sine of the (positive) angle formed by $\mathbf{v}_j$ and $\mathbf{v}_k$. These expressions are obtained with the help of the equalities :

$$\frac{\partial}{\partial \boldsymbol{\theta}}\|\boldsymbol{\theta}\| = \boldsymbol{\theta}^\top$$

and that :

$$\frac{\partial}{\partial \boldsymbol{\theta}}\left(\boldsymbol{\theta}\boldsymbol{\theta}^\top \mathbf{v}\right) = \boldsymbol{\theta}^\top \mathbf{v} I - \boldsymbol{\theta}\mathbf{v}^\top.$$

C.2 Parameterization of rotation matrices

We now define the differential of the parameterization of the rotation matrices defined in Eq. (5.9), on page 61, from which the differential of the function defined in Eq. (5.10) is easily obtained. We thus consider the mapping :

$$R(\varphi) = \tilde{\varphi}\tilde{\varphi}^\top + \cos(\tau)\left(I_3 - \tilde{\varphi}\tilde{\varphi}^\top\right) - \sin(\tau) S_{\tilde{\varphi}}, \text{ for } \varphi \neq \mathbb{0}, \tag{C.5}$$

where $\tau = \|\varphi\|$, $\tilde{\varphi}_f = \varphi / \|\varphi\|$ and $S_{\tilde{\varphi}}$ is the Rodrigues matrix of $\tilde{\varphi}$. The domain of $R(\varphi)$ is extended to $\mathbb{R}^3$ by defining $R(\mathbb{0}) = I_3$. The continuity of $R(\varphi)$ in 0 and elsewhere is easily shown.

We saw in Section 5.2.3 that only the differential of $R(\boldsymbol{\varphi})$ in $\mathbb{0}$ is needed. All that is needed to show that $R(\boldsymbol{\varphi})$ is differentiable in 0 [2] is the existence of a linear function $\mathcal{D}R(\mathbb{0})$ such that :

$$R(\boldsymbol{\theta}) = R(\mathbb{0}) + \mathcal{D}R(\mathbb{0})(\boldsymbol{\theta}) + O\left(\|\boldsymbol{\theta}\|^2\right). \tag{C.6}$$

We now show that taking $\mathcal{D}R(\mathbb{0})(\boldsymbol{\theta}) = -S_{\boldsymbol{\theta}}$, i.e. a linear function of $\boldsymbol{\theta}$, satisfies this requirement. First, $R(\boldsymbol{\theta})$ can be written

$$R(\boldsymbol{\theta}) = \cos(\|\boldsymbol{\theta}\|)\, I - \frac{\sin(\|\boldsymbol{\theta}\|)}{\|\boldsymbol{\theta}\|} S_{\boldsymbol{\theta}} + \frac{\cos(\|\boldsymbol{\theta}\|) - 1}{\|\boldsymbol{\theta}\|^2} \boldsymbol{\theta}\boldsymbol{\theta}^\top.$$

Then, using the Taylor expansion of $\sin(\|\boldsymbol{\theta}\|)$ and $\cos(\|\boldsymbol{\theta}\|)$, one obtains Eq. (C.6).

In order to present $DR(\boldsymbol{\theta})$ in matrix notation, we consider the differential of $\text{vec}(R(\boldsymbol{\theta}))$ in $\mathbb{0}$, which is :

$$\frac{\partial}{\partial \boldsymbol{\theta}} \text{vec}(R(\mathbb{0})) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top. \tag{C.7}$$

The differential of $R(\boldsymbol{\theta})$ can also be found for nonzero values of $\boldsymbol{\theta}$, by differentiation of Eq. (C.5), but the resulting expression is less simple than Eq. (C.7).

Appendix D

Covariance of maximum likelihood estimators

We show how the covariance matrix of a maximum-likelihood estimator can be derived analytically, simply from its definition. This result, which can be found in [34], applies for any error model that results in a twice differentiable likelihood function. We then specialize the expressions to the case of independent identically distributed Gaussian error terms, when the (negative log-) likelihood function is a sum of squares. This is precisely the situation in Section 5.

It is often the case that the maximum of the likelihood function is not isolated, in which case the estimator is not uniquely defined. This difficulty is overcome by constraining the estimates to lie in a subset in which the maximum of the likelihood function is isolated. More precisely, the estimate is restricted to a set of the form

$$\{\Theta \,|\, S(\Theta) = \mathbb{0}_{P\times 1}\}$$

where $S(.)$ is a differentiable function chosen in such a way that the maximum likelihood estimator is well defined.

While this function can be used to express any type of constraints, including all the geometric constraints specified by the user (Chapter 2.3), in the present work it is used to remove the usual scale-and-translation ambiguity and to enforce that the parameters that define the dominant directions are normalized as in Appendix C.1.

The exact definition of the estimator is :

Definition : The maximum likelihood estimator is the function that associates to a set of observations $\mathbf{x}$, the maximum likelihood estimate $\hat{\Theta}$, defined by :

$$\hat{\Theta} = \arg\min_{\Theta} Q(\mathbf{x}, \Theta) \quad \text{subject to } S(\Theta) = 0_{P\times 1}. \tag{D.1}$$

D.1 Notations

We shall write Θ^* the "true" parameters, $\hat{\Theta}$ the ML estimate; their difference, the error, is $\Delta\Theta = \hat{\Theta} - \Theta^*$. We will write $Q^* = Q(\mathbf{x}^*, \Theta^*)$, $\hat{Q} = Q(\mathbf{x}, \hat{\Theta})$ and likewise $\hat{S} = S(\hat{\Theta})$. In general, an asterisk will denote a function evaluated in Θ^* or $(\mathbf{x}^*, \Theta^*)$, and a hat will denote evaluation at $\hat{\Theta}$ or $(\mathbf{x}, \hat{\Theta})$. We will write D_Θ and $D^2_{\Theta\Theta}$ the operators of first and second differentiation with respect to Θ. Finally, the following abbreviations are used:

$$\begin{array}{rcccl} D_\Theta Q^* & = & \frac{\partial Q}{\partial \Theta}(\mathbf{x}^*, \Theta^*) & & \\ D_{\Theta\Theta} Q^* & = & \frac{\partial^2 Q}{\partial \Theta^2}(\mathbf{x}^*, \Theta^*) & & \\ D_\Theta \hat{Q} & = & \frac{\partial Q}{\partial \Theta}(\hat{\mathbf{x}}, \hat{\Theta}) & = & F \\ D^2_{\Theta\Theta} \hat{Q} & = & \frac{\partial^2 Q}{\partial \Theta^2}(\hat{\mathbf{x}}, \hat{\Theta}) & = & H \\ D_\Theta \hat{S} & = & \frac{\partial^2 S}{\partial \Theta^2}(\hat{\Theta}) & = & G \end{array}$$

D.2 Derivation of the covariance of the estimator

A well-known property is that, at the minimum $\hat{\Theta}$, the derivative of Q is a linear combination of the derivatives of constraints. That is, there exist a (row) vector Λ such that :

$$D_\Theta \hat{Q} + \Lambda D_\Theta \hat{S} = 0_{1\times \text{size}(\Theta)} \tag{D.2}$$

These are the so-called "*normal equations*". The first-order Taylor series of $D_\Theta Q$ at $(\hat{\mathbf{x}}, \hat{\Theta})$, yields the following approximation :

$$D_\Theta Q^* \simeq D_\Theta \hat{Q} - D^2_{\Theta\Theta} \hat{Q} \Delta\Theta - D^2_{\Theta\mathbf{x}} \hat{Q} \varepsilon. \tag{D.3}$$

It is easy to see that $D_\Theta Q^* = 0$: since $Q^* = 0$, and, for all $(\mathbf{x}, \Theta)$, $Q(\mathbf{x}, \Theta) \geq 0$, then $(\mathbf{x}^*, \Theta^*)$ is a global minimum of Q (regardless of constraints). This implies that $D_\Theta Q^* = 0$. Thus one has :

$$D^2_{\Theta\Theta} \hat{Q} \Delta\Theta + D^2_{\Theta\mathbf{x}} \hat{Q} \varepsilon + \Lambda D_\Theta \hat{S} \simeq 0.$$

Likewise, since $S^* = 0 = \hat{S}$, and $\hat{S} \simeq \hat{S} - D_\Theta \hat{S} \Delta\Theta$, one has : $D_\Theta \hat{S} \Delta\Theta \simeq 0$. Replacing approximations by equalities, and writing in matrix form, one has:

$$\begin{bmatrix} D^2_{\Theta\Theta}\hat{Q} & D_\Theta \hat{S}^\top \\ D_\Theta \hat{S} & 0 \end{bmatrix} \begin{bmatrix} \Delta\Theta \\ \Lambda^\top \end{bmatrix} = \begin{bmatrix} -D^2_{\Theta\mathbf{x}}\hat{Q}\varepsilon \\ 0 \end{bmatrix},$$

which can be abbreviated as :

$$\begin{bmatrix} H & G^\top \\ G & 0 \end{bmatrix} \begin{bmatrix} \Delta\Theta \\ \Lambda^\top \end{bmatrix} = \begin{bmatrix} -F\varepsilon \\ 0 \end{bmatrix}. \tag{D.4}$$

The vector $[\Delta\Theta^\top, \Lambda]$ is thus *a linear combination of* ε. Its covariance is :

$$\operatorname{cov} \begin{bmatrix} \Delta\Theta \\ \Lambda^\top \end{bmatrix} = \begin{bmatrix} H & G^\top \\ G & 0 \end{bmatrix}^{-1} \begin{bmatrix} \sigma^2 F F^\top & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} H & G^\top \\ G & 0 \end{bmatrix}^{-1} \tag{D.5}$$

D.3 Specialization to the problem of reconstruction :

The above formulas hold for any ML estimator. We now specialize them to the case of Gaussian noise, when the log-likelihood is a sum of squared differences between observations and predictions, of the form of Eq. 5.5 :

$$\begin{aligned} Q &= \frac{1}{2\sigma^2}\left(\mathbf{f}(\Theta) - \mathbf{x}\right)^\top \left(\mathbf{f}(\Theta) - \mathbf{x}\right) \\ D_\Theta Q &= \frac{1}{\sigma^2}\left(\mathbf{f}(\Theta) - \mathbf{x}\right)^\top D_\Theta \mathbf{f}(\Theta) \\ D^2_{\Theta\Theta} Q &= \tfrac{1}{\sigma^2} D_\Theta \mathbf{f}(\Theta)^\top D_\Theta \mathbf{f}(\Theta) + \\ & \quad \frac{1}{\sigma^2} \textstyle\sum_{m=1}^N \sum_{i=1}^2 \left(f_{mi}(\Theta) - x_{mi}\right) D^2_{\Theta\Theta} f_{mi}(\Theta) \\ D^2_{\Theta\mathbf{x}} Q &= \frac{1}{\sigma^2} D_\Theta \mathbf{f}(\Theta) \end{aligned} \tag{D.6}$$

Practical consideration : One can eliminate the need of knowing the second derivatives $D^2_{\Theta\Theta} f_{mi}(\Theta)$ when computing covariance matrices, because, at $(\Theta^*, \mathbf{x}^*)$, one has $f_{mo} = x_{mi}$, and thus the second order terms are eliminated in Eq. (D.6). In what follows, we assume that, at $\hat{\Theta}$, the residuals $(f_{mi}(\Theta) - x_{mi})$ are small, so that the second order terms in Eq. (D.6) are small compared to the first order terms. With that assumption, the approximation :

$$D^2_{\Theta\Theta} Q \simeq \frac{1}{\sigma^2} D_\Theta \mathbf{f}(\Theta)^\top D_\Theta \mathbf{f}(\Theta)$$

is legitimate, and from here on, we will use it.

Covariance of the ML estimator : Because we consider that noise terms ε_{npi} are independent and have same variance σ^2, one has : $F\Sigma F^\top = \sigma^2 D_\Theta Q^\top D_\Theta Q = H$. Replacing in Eqs. (D.5) yields.

$$\mathrm{Cov}\begin{bmatrix}\Delta\Theta \\ \Lambda\end{bmatrix} = \begin{bmatrix} H & G^\top \\ G & 0\end{bmatrix}^{-1}\begin{bmatrix} H & 0 \\ 0 & 0\end{bmatrix}\begin{bmatrix} H & G^\top \\ G & 0\end{bmatrix}^{-1} \tag{D.7}$$

Expressions without Λ It is also possible to find expressions for $\Delta\Theta$ and its covariance in which Λ does not appear : one easily sees that :

Proposition D-3 *If there is a solution to :*

$$\begin{bmatrix} H & G^\top \\ G & \mathbb{0}\end{bmatrix}\begin{bmatrix} X \\ Y\end{bmatrix} = \begin{bmatrix} U \\ V\end{bmatrix},$$

then

$$X = \left(\Pi_{G^\perp}H + \Pi_G\right)^+ \left(\Pi_{G^\perp}U + G\left(G^\top G\right)^{-1}V\right),$$

where Π_G *is the orthogonal projector onto* $\mathcal{S}pan\left(G^\top\right)$*,* $\Pi_G = G^\top\left(GG^\top\right)^{-1}G$ *and* $\Pi_{G^\perp}$ *is the orthogonal projector onto* $\mathcal{N}ull\left(G\right)$*,* $\Pi_{G^\perp} = I - G^\top\left(GG^\top\right)^{-1}G$*.*

From this proposition and Eq. (D.5), one has

$$\Delta\Theta = \left(\Pi_{G^\perp}H + \Pi_G\right)^{-1}\Pi_{G^\perp}F\varepsilon. \tag{D.8}$$

By definition of $\Pi_{G^\perp}$ and Π_G, one may write these matrices $\Pi_{G^\perp} = U_1U_1^\top$ and $\Pi_G = U_2U_2^\top$, where U_1 and U_2 are $M \times (M-P)$ and $M \times P$ unitary matrices such that $U_1^\top U_1 = I_{M-P}$, $U_2^\top U_2 = I_P$ and $U_2^\top U_1 = \mathbb{0}_{P\times(M-P)}$.

One easily shows the following :

Proposition D-4 *If*

$$\begin{bmatrix} U_1^\top \\ U_2^\top\end{bmatrix}\begin{bmatrix} U_1 & U_2\end{bmatrix} = I$$

and $\mathcal{S}pan\left(U_2\right) \subset \mathcal{N}ull\left(H\right)$*, then one has :*

$$\left(U_1U_1^\top H + U_2U_2^\top\right)^{-1} = \left(U_1\left(U_1^\top HU_1\right)^{-1}U_1^\top - U_1\left(U_1^\top HU_1\right)_1^{-1}\left(U_1^\top HU_2\right)U_2 + U_2U_2^\top\right).$$

Using this proposition, one gets from Eq. (D.8) :

$$\Delta\Theta = U_1\left(U_1^\top HU_1\right)^{-1}U_1F\varepsilon. \tag{D.9}$$

The, one gets the expression for the covariance :

$$\operatorname{cov}(\Delta\Theta) = \sigma^2 U_1 \left(U_1^\top H U_1\right)^{-1} U_1 F F^\top U_1^\top \left(U_1^\top H U_1\right)^{-1} U_1^\top . \tag{D.10}$$

Finally, the counterpart of Eq. (D.7) is :

$$\operatorname{cov}(\Delta\Theta) = U_1 \left(U_1^\top H U_1\right)^{-1} U_1^\top . \tag{D.11}$$

Appendix E

Differentiation of the nullspace of a matrix-valued function

In this appendix is proved the correctness of Eq. (5.8) for defining a matrix-valued function whose columns form an orthonormal basis of the nullspace of a matrix-valued function of constant corank. We also show that the resulting function is differentiable and give means to compute its differential. In order to simplify the notation in this appendix, we will write $B(\boldsymbol{\theta})$ (instead of $B(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D)$) the original function, assume it has size $P \times N$ (instead of $P \times 3N$).

We consider a differentiable matrix-valued function $B(\boldsymbol{\theta})$ defined on an open set $\mathcal{D}$ of $\mathbb{R}^Q$ such that the dimension of its nullspace $\mathcal{N}ull(B(\boldsymbol{\theta}))$[1] is constant for all $\boldsymbol{\theta}$, and define a differentiable matrix-valued function $U(\boldsymbol{\theta})$ such that the columns of $U(\boldsymbol{\theta})$ form an orthonormal basis of $\mathcal{N}ull(B(\boldsymbol{\theta}))$. This requirement alone, as was noted in Sec. 5.2.1 (page 60), does not define uniquely a function $U(\boldsymbol{\theta})$, so that other constraints should be imposed. The ambiguity if *partially* removed by choosing a point $\boldsymbol{\theta}_0$ and fixing the value of U in $\boldsymbol{\theta}_0$. However, this is in general still insufficient to uniquely define $U(\boldsymbol{\theta})$ in points other than $\boldsymbol{\theta}_0$, as shown by a counter-example in Section E.6.

For this reason, we impose the extra constraint that $U(\boldsymbol{\theta})$ minimizes, amongst all unitary matrices whose columns form a basis of $\mathcal{N}ull(B(\boldsymbol{\theta}))$, the distance $\|U(\boldsymbol{\theta}_0) - U(\boldsymbol{\theta})\|_F$, where $\|.\|_F$ is the Frobenius norm, which is the condition given in Eq. (5.8).

In this appendix, we show that the definition given in Eq. (5.8) uniquely defines a function $U(\boldsymbol{\theta})$. The following proposition states formally the conditions under which it can be used.

Proposition E-2 *If one is given a differentiable function $B(\theta)$ defined on an open set $\mathcal{D} \subset \mathbb{R}^Q$ whose images are $P \times N$ real matrices, with rank L and corank equal to some $M > 0$ for all θ, and*

[1] The nullspace of a matrix B, defined as the vectorial subspace of $\mathbb{R}^N$ consisting of all vectors $\mathbf{b}$ such that $B\mathbf{b} = \mathbb{0}_{P\times 1}$, is written $\mathcal{N}ull(B)$.

one is given a point $\theta_0 \in \mathcal{D}$ and a unitary matrix U_0 whose columns form an orthonormal basis of $\mathcal{N}ull\left(B\left(\theta_0\right)\right)$,

then there exists a neighborhood $\mathcal{F}$ of θ_0 and a $N \times M$ matrix-valued differentiable function $U\left(\theta\right)$ defined on $\mathcal{F}$, with image in $\mathbb{R}^{N\times M}$, such that, for all $\theta \in \mathcal{F}$, one has :

$$\begin{aligned} B\left(\theta\right) U\left(\theta\right) &= \mathbb{0}_{P\times M}, & \text{(E.1)} \\ U\left(\theta\right)^\top U\left(\theta\right) &= I_M \textit{ and} & \text{(E.2)} \\ U\left(\theta\right) &= \textit{arg}\min_U \left\{ \left\| U_0 - U \right\|_F \,\middle|\, U \textit{ verifies} \left(2,3\right) \right\}. & \text{(E.3)} \end{aligned}$$

The function $U\left(\theta\right)$ is called the nullspace function *of (the function) $B\left(\theta\right)$. Moreover, $U\left(\theta\right)$ can be computed using Equation (E.9), its differential can be computed by Equations (E.10) and (E.14), and the differential in θ_0 takes the simpler form in Equation (E.13)..*

We show, using the implicit function theorem[2], that the above constraints are sufficient to define a differentiable function U on a neighborhood of $\boldsymbol{\theta}_0$. Also, we show how to efficiently implement such a function in a programming language such as "Octave" [22] or "Matlab" [51][3].

The demonstration leading to this proposition is given in the following sections, together with methods for computing $U\left(\boldsymbol{\theta}\right)$ (Eq. (E.9)) and $\frac{\partial}{\partial\boldsymbol{\theta}}U\left(\boldsymbol{\theta}_0\right)$ (Eq. (E.13) or Eqs. (E.10) and (E.14)).

Results related to that of this appendix have been known for some time. The differentiation of the eigenvalues and eigenvectors of square matrices are studied in [45, 44, 25], and previous work which could not be consulted is sometimes cited [40, 58]. Computational procedures are known for simple [52] and multiple [53, 18] eigenvalues. The main difference between previously published work and this appendix lies in the choice of the normalization used to ensure the unicity of a basis of the nullspace, in the method used to prove the existence and unicity of the "nullspace function" and the possibility of computing the value and derivatives of this mapping in points other than $\boldsymbol{\theta}_0$.

This appendix has the following plan : Theorem E-2 is proven for a special case in Section E.1. Computation methods are given in Sections E.2 and E.3 and the general case is treated in Section E.4.

E.1 Existence and differentiability

We will rephrase (E.1-E.3) in the form $h\left(\boldsymbol{\theta}, U\right) = \mathbb{0}$ in such a way that the implicit function theorem may be applied. We start by finding an equivalent set of non-redundant equations that is equivalent to (E.1-E.3).

[2] Credit must be given to João Xavier for suggesting the usage of the implicit function theorem for showing the differentiability of U.

[3] Octave is a numerical language comparable to "Matlab". In this appendix, only features common to the two languages are considered.

Characterization The system of equations (in U) $B(\boldsymbol{\theta}_0)U = \mathbb{0}_{P\times M}$ has rank LM and, unless $L = P$, it is redundant. We assume in this section that $P = L$ and consider again the general case in Sections E.2, E.3 and E.4.

Since $U^\top U$ is symmetric, Equation (E.2) is redundant. A non-redundant formulation of this equation is $D_M^\top \mathrm{vec}\left(U^\top U - I_M\right) = \mathbb{0}_{(M(M+1)/2)\times 1}$, where D_M is the $M^2 \times M(M+1)/2$ duplication matrix [28, 50].

Moreover, one shows in Section E.6 that a matrix U, verifying (E.1) and (E.2), minimizes $\|\mathrm{U}_0 - U\|_F$ if and only if $U_0^\top U$ is symmetric, that is, if $U_0^\top U - U^\top U_0 = \mathbb{0}_{M\times M}$. This can be expressed as the vectorial equation

$$\mathrm{vec}\left(U_0^\top U - U^\top U_0\right) = \mathbb{0}_{M^2}. \tag{E.4}$$

However, since $U_0^\top U - U^\top U_0$ is skew-symmetric, some of these equations are identical; the $M(M-1)/2$ distinct equations can be written in the form $W_M^+ \mathrm{vec}\left(U_0^\top U - U^\top U_0\right) = \mathbb{0}$ where W_M^+ is the $M^2 \times (M(M-1)/2)$ selects the sub-diagonal elements : for all $M \times M$ skew-symmetric matrix A, one has :

$$\mathrm{vec}(A) = W_M \mathrm{vecl}(A),$$

where $\mathrm{vecl}(A) = [A_{2,1}, ..., A_{M,1}, A_{3,2}, ..., A_{MM-1}]^\top$ is the vector of sub-diagonal elements of A. This matrix is the counterpart, for skew-symmetric matrices, of the "duplication matrix" .

Altogether, (E.1-E.3) can be expressed as $h(\boldsymbol{\theta}, U) = \mathbb{0}_{MN\times 1}$, where h is defined by :

$$h(\boldsymbol{\theta}, U) = \left.\begin{bmatrix} \mathrm{vec}\left(B(\boldsymbol{\theta})U\right) \\ D_M^\top \mathrm{vec}\left(U^\top U - I_M\right) \\ W_M^\top \mathrm{vec}\left(U_0^\top U - U^\top U_0\right) \end{bmatrix}\right\} MN \tag{E.5}$$

The three components in (E.5) have length MP, $M(M-1)/2$ and $M(M+1)/2$ respectively, which sum up to MN. It is clear that $h(\boldsymbol{\theta}, U)$ is differentiable in both $\boldsymbol{\theta}$ and U.

Existence and differentiability In order to apply the implicit function theorem, one must show that $\frac{\partial}{\partial U} h$ is bijective. Representing this differential as a $MN \times MN$ matrix, one has :

$$\frac{\partial}{\partial U} h(\boldsymbol{\theta}, U) = \begin{bmatrix} I_M \otimes B(\boldsymbol{\theta}) \\ D_M^\top \left(K_M + I_{M^2}\right)\left(I_M \otimes U^\top\right) \\ W_M^\top \left(K_M - I_{M^2}\right)\left(I_M \otimes U_0^\top\right) \end{bmatrix}$$

where K_M is the commutation matrix [28, 50] of size M^2. Noting that $D_M^\top (K_M + I_{M^2}) = 2D_M^\top$ and $W_M^\top (K_M - I_{M^2}) = 2W_M^\top$, this expression can be simplified to :

$$\frac{\partial}{\partial U} h(\boldsymbol{\theta}, U) = \begin{bmatrix} I_M \otimes B(\boldsymbol{\theta}) \\ 2D_M^\top \left(I_M \otimes U^\top\right) \\ 2W_M^\top \left(I_M \otimes U_0^\top\right) \end{bmatrix} \tag{E.6}$$

This matrix is shown to be invertible in Section E.6. According to the implicit function theorem, there exist a neighborhood $\mathcal{F}$ and a differentiable function $U(\boldsymbol{\theta})$ defined on $\mathcal{F}$ such that for all $\boldsymbol{\theta} \in \mathcal{F}$, one has $h(\boldsymbol{\theta}, U(\boldsymbol{\theta})) = \mathbb{0}$. Moreover, the differential of $U(\boldsymbol{\theta})$ is given by :

$$\frac{\partial}{\partial \boldsymbol{\theta}} U(\boldsymbol{\theta}) = -\frac{\partial}{\partial U} h(\boldsymbol{\theta}, U)^{-1} \frac{\partial}{\partial \boldsymbol{\theta}} h(\boldsymbol{\theta}, U). \tag{E.7}$$

and the differential of h with respect to $\boldsymbol{\theta}$ is equal to :

$$\frac{\partial}{\partial \boldsymbol{\theta}} h(\boldsymbol{\theta}, U) = \begin{bmatrix} K_{M,P} \left(I_P \otimes U^\top\right) \\ \mathbb{0}_{M^2 \times NP} \end{bmatrix} \frac{\partial}{\partial \boldsymbol{\theta}} \mathrm{vec}\left(B(\boldsymbol{\theta})\right). \tag{E.8}$$

In conclusion, we have just proven Theorem E-2 in the special case of $\mathrm{rank}(B(\boldsymbol{\theta})) = P$.

E.2 Computing $U(\boldsymbol{\theta})$

As indicated in Section E.6, the unitary matrix U whose columns form a basis of $\mathcal{N}ull(B(\boldsymbol{\theta}))$ and minimizes $\|U - U_0\|_F$ is given by :

$$U = U_1 \mathcal{U} \mathcal{V}^\top \tag{E.9}$$

where U_1 can be any unitary matrix whose columns form a basis of $\mathcal{N}ull(B(\boldsymbol{\theta}))$, and $\mathcal{U}$, $\mathcal{V}$ are given by the SVD decomposition of $U_1^\top U_0$: $U_1^\top U_0 \overset{\mathrm{SVD}}{=} \mathcal{U} D \mathcal{V}^\top$. In practice, when using the Octave [22] language, U_1 is obtained with the `null()` function, and $\mathcal{U}$ and $\mathcal{V}$ by the `svd()` function.

Note that $U(\boldsymbol{\theta})$ can be computed using (E.9), even if $\mathrm{rank}(B(\boldsymbol{\theta})) = L < P$.

E.3 Computing $\frac{\partial}{\partial \boldsymbol{\theta}} U(\boldsymbol{\theta})$

Equation (E.7) provides a straightforward way of computing the differential of U, but we shall see that this computation can be done in a way that is less costly and that does not require that $B(\boldsymbol{\theta})$ have rank P. The computation is made possible by identifying necessary conditions on the partial derivatives of $U(\boldsymbol{\theta})$ that completely characterize these partial derivatives. This method is inspired by that found in e.g. [50, 54] for computing the partial derivatives of the eigen- and singular values and vectors of matrices.

For convenience, we will write the partial derivatives of $B(\boldsymbol{\theta})$ and $U(\boldsymbol{\theta})$ with respect to the i^{th} component of $\boldsymbol{\theta}$

$$B_i' = \frac{\partial}{\partial \boldsymbol{\theta}_i} B(\boldsymbol{\theta}) \text{ and } U_i' = \frac{\partial}{\partial \boldsymbol{\theta}_i} U(\boldsymbol{\theta}).$$

Computing the derivative of (E.1) with respect to θ_i, one gets :

$$B_i'U + AB_i' = \mathbb{0}_{N \times Q},$$

which in turn implies that :

$$U_i' = -B^{+}B_i'U + UC \tag{E.10}$$

for some $M \times M$ matrix C. Here, B^{+} is the pseudo-inverse of B.

The derivative of (E.2) with respect to θ_i, yields :

$$U_i'^{\top}U + U^{\top}U_i' = \mathbb{0}_{M \times M}. \tag{E.11}$$

Replacing (E.10) into (E.11), one gets : $C = -C^{\top}$, that is, C is skew symmetric.

Finally, the derivative of (E.4) with respect to θ_i is :

$$U_0^{\top}U_i' - U_i'^{\top}U_0 = \mathbb{0}_{M \times M}\,.$$

Using (E.10) in this expression, one obtains the following constraint on C :

$$X + Y^{\top}C - C^{\top}Y = \mathbb{0}_{M \times M}\,, \tag{E.12}$$

where $X = -U_0^{\top}B^{+}B_i'U + U^{\top}B_i'^{\top}B^{+\top}U_0$ and $Y = U_0^{\top}U$. Note that, when $U = U_0$, this equation reduces to $C = \mathbb{0}$, so that, at $U = U_0$, one has :

$$U_i' = -B^{+}B_i'U. \tag{E.13}$$

This case is of importance in practice, because this expression is simpler than the general case and because the optimization algorithm used in Chapter 5 only requires derivatives of this form.

It is shown in Section E.6 that the only skew-symmetric C that solves (E.12) is

$$\operatorname{vecl}(C) = \left(W_M^{\top}\left(I_M \otimes Y^{\top}\right)W_M\right)^{-1}\operatorname{vecl}(X)\,. \tag{E.14}$$

The partial derivative U_i' is then given by (E.10).

E.4 Generalization to rank-deficient $B(\boldsymbol{\theta})$

It can be seen that (E.10) and (E.13) do not require that $B(\boldsymbol{\theta})$ have rank P, contrarily to (E.7).

We now show that it is valid to use expressions (E.10) and (E.13) when $\operatorname{rank}(B(\boldsymbol{\theta})) = L < P$.

If we are given a function $B(\boldsymbol{\theta})$ such that (E.10) and (E.13), and if there exists a differentiable function $\tilde{B}(\boldsymbol{\theta})$ defined on $\mathcal{D}$ whose images are $L \times N$ matrices and such that $\mathcal{S}pan\left(\tilde{B}(\boldsymbol{\theta})^\top\right) = \mathcal{S}pan\left(B(\boldsymbol{\theta})^\top\right)$ for all $\boldsymbol{\theta}$ (and thus $\tilde{B}(\boldsymbol{\theta})$ has rank L), then $B(\boldsymbol{\theta})$ and $\tilde{B}(\boldsymbol{\theta})$ have same null space for all $\boldsymbol{\theta}$. As a consequence, their nullspace functions $U(\boldsymbol{\theta})$ and $\tilde{U}(\boldsymbol{\theta})$, if they exist, are equal.

The existence of $\tilde{U}(\boldsymbol{\theta})$ is given by Theorem E-2 and thus $U(\boldsymbol{\theta})$ also exists. Since the computations of $\frac{\partial}{\partial\boldsymbol{\theta}}U(\boldsymbol{\theta})$ given in (E.13), (E.10) and (E.14) are based only on necessary conditions that $\frac{\partial}{\partial\boldsymbol{\theta}}U(\boldsymbol{\theta})$ must verify, and can be done even if $B(\boldsymbol{\theta})$ is not full rank, their results are correct in that case too.

It is thus possible to compute $U(\boldsymbol{\theta})$ and $\frac{\partial}{\partial\boldsymbol{\theta}}U(\boldsymbol{\theta})$ even if $\operatorname{rank}(B(\boldsymbol{\theta})) < N$, without having to compute $\tilde{B}(\boldsymbol{\theta})$ or even knowing $\operatorname{rank}(B(\boldsymbol{\theta}))$ exactly.

All that is needed now is to prove the existence of a function $\tilde{B}(\theta) \in \mathbb{R}^{L\times N}$. We first define $\tilde{B}(\theta)$ locally, e.g. in a neighborhood of any $\theta_1 \in \mathcal{D}$. First, there exists a subset of L independent rows of $B(\theta_1)$ and there thus exists a $L \times N$ matrix of zeros and ones, S_1, that selects these rows, so that $\tilde{B}_1(\theta_1) = S_1 B(\theta_1)$ has rank L. Because $\tilde{B}_1(\theta)$ is a continuous function of θ, and the smallest singular value of $\tilde{B}_1(\theta)$ is a continuous function of $\boldsymbol{\theta}$ [77, Part 2][4], there exists an open neighborhood of θ_1 on which the rank of $\tilde{B}_1(\theta)$ does not vary. Then, $\mathcal{D}$ can be covered by such overlapping neighborhoods, so that a $L \times P$ function $\tilde{B}(\theta)$ is defined. Note also that only finitely many neighborhoods are needed, since there are finitely many selection matrices. Although $\tilde{B}(\theta)$ is not continuous (because the subset of selected rows changes), it is differentiable almost everywhere (everywhere except on the borders of the neighborhoods) and its nullspace varies in a "continuous" fashion. As a consequence, the nullspace function $U(\theta)$ can be defined everywhere and is differentiable.

This completes the proof of Theorem E-2, and of the correctness of (E.9), (E.10), (E.14) and (E.13) for computing $U(\theta)$ and its differentials.

[4] Wilkinson considers the eigenvalues. We can apply his results to $\tilde{B}_1(\theta)\,\tilde{B}_1(\theta)^\top$, whose eigenvalues are the squares of the singular values of $\tilde{B}_1(\theta)$.

E.5 Summary

We have shown that it is possible to build a differentiable function $U(\boldsymbol{\theta})$ whose columns form a basis of the nullspace of a differentiable function $B(\boldsymbol{\theta})$. Many such functions exist; a particular function is chosen by fixing a particular $\boldsymbol{\theta}_0$ and a particular matrix U_0 whose columns form an orthonormal basis of $\mathcal{N}ull(B(\boldsymbol{\theta}))$. The function $U(\boldsymbol{\theta})$ can be computed using Eqs. (E.9), while the differential can be computed using Eq. (E.10) and (E.14) or (E.13). These expressions can be computed for all $\boldsymbol{\theta} \in \mathcal{D}$, but the existence and differentiability of $U(\boldsymbol{\theta})$ is only guaranteed on a neighborhood of $\boldsymbol{\theta}_0$.

E.6 Detailed demonstrations

Demonstrations of properties used in the previous sections are found in this section.

Non-unicity of $U(\boldsymbol{\theta})$ when $U(\boldsymbol{\theta}_0)$ is fixed We show, by an example, that fixing the value of $U(\boldsymbol{\theta})$ in a point $\boldsymbol{\theta}_0$ is not sufficient to uniquely define the function $U(\boldsymbol{\theta})$.

Consider the function $B(\boldsymbol{\theta}) = [\cos(\boldsymbol{\theta})\ \sin(\boldsymbol{\theta})\ 0] \in \mathbb{R}^{1\times 3}$, defined on $\mathbb{R}$ and take $\boldsymbol{\theta}_0 = 0$ and $U_0 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^\top$, a basis of $\mathcal{N}ull(B(\boldsymbol{\theta}_0))$. Define $U_1(\boldsymbol{\theta}) = \begin{bmatrix} -\sin(\boldsymbol{\theta}) & \cos(\boldsymbol{\theta}) & 0 \\ 0 & 0 & 1 \end{bmatrix}^\top$ and $U_2(\boldsymbol{\theta}) = U_1(\boldsymbol{\theta}) \begin{bmatrix} \cos(\boldsymbol{\theta}) & -\sin(\boldsymbol{\theta}) \\ \sin(\boldsymbol{\theta}) & \cos(\boldsymbol{\theta}) \end{bmatrix}$. Both functions verify $U_i(\boldsymbol{\theta}_0) = U_0$, but one does not have in general $U_1(\boldsymbol{\theta}) = U_2(\boldsymbol{\theta})$.

Condition on the minimizing U It is well known [26, p. 601] that, given $N \times M$ matrices U_0 and U_1, the orthogonal matrix R that minimizes $\|U_0 - U_1 R\|_F$ is $R^* = \mathcal{U}\mathcal{V}^\top$, where $\mathcal{U}$ and $\mathcal{V}$ are given by the singular value decomposition of $U_0^\top U_1 \overset{\text{SVD}}{=} \mathcal{U}D\mathcal{V}^\top$. Moreover, one sees that $U_0^\top U_1 R^*$ is symmetric by computing $U_0^\top U_1 R^* - R^{*\top} U_1^\top U_0$.

Differentials of $f(\mathbf{u},\mathbf{v}) = \mathbf{vec}\left(F^\top G\right)$ Let $\mathbf{u} \in \mathbb{R}^{PM}$, $\mathbf{v} \in \mathbb{R}^{PN}$ be vectors, $F = [\mathbf{u}_1 \ldots \mathbf{u}_M] = \mathrm{mat}_{P,M}(\mathbf{u})$ and $G = [\mathbf{v}_1 \ldots \mathbf{v}_N] = \mathrm{mat}_{P,N}(\mathbf{v})$ be $P \times M$ and $P \times N$ matrices, and define $f(\mathbf{u},\mathbf{v}) =$

$\text{vec}\left(F^\top G\right) \in \mathbb{R}^{MN}$. The differentials of $f(\mathbf{u},\mathbf{v})$ with respect to $\mathbf{u}$ and $\mathbf{v}$ are :

$$\frac{\partial}{\partial \mathbf{v}} f(\mathbf{u},\mathbf{v}) = \left(I_N \times F^\top\right) \qquad \text{because} \quad f(\mathbf{u},\mathbf{v}) = \begin{bmatrix} \mathbf{u}_1^\top \mathbf{v}_1 \\ \vdots \\ \mathbf{u}_M^\top \mathbf{v}_1 \\ \vdots \\ \mathbf{u}_M^\top \mathbf{v}_N \end{bmatrix} = \left(I_N \otimes F^\top\right)\mathbf{v}, \text{ and}$$

$$\frac{\partial}{\partial \mathbf{u}} f(\mathbf{u},\mathbf{v}) = K_{M,N}\left(I_M \otimes G^\top\right) \quad \text{because} \quad f(\mathbf{u},\mathbf{v}) = \begin{bmatrix} \mathbf{v}_1^\top \mathbf{u}_1 \\ \vdots \\ \mathbf{v}_1^\top \mathbf{u}_M \\ \vdots \\ \mathbf{v}_M^\top \mathbf{u}_N \end{bmatrix} = K_{M,N}\left(I_M \otimes G^\top\right)\mathbf{u}.$$

Finally, if one defines $g(\mathbf{u}) = f(\mathbf{u},\mathbf{u})$, one has $\frac{\partial}{\partial \mathbf{u}} g(\mathbf{u}) = \left(I_{M^2} + K_M\right)\left(I_M \otimes F^\top\right)$.

Invertibility of $\frac{\partial}{\partial U} h$ We now show, that $\frac{\partial}{\partial U} h$ is invertible, by showing that any vector $\mathbf{w}^\top = \text{vec}(W) \in \mathbb{R}^{MN}$ such that :

$$\begin{bmatrix} I_M \otimes B(\boldsymbol{\theta}) \\ D_M^\top\left(I_M \otimes U^\top\right) \\ W_M^\top\left(I_M \otimes U_0^\top\right) \end{bmatrix} \mathbf{w} = \mathbb{0}_{MN\times 1}.$$

is necessarily zero (we have suppressed the needless factor 2 in the lower blocks).

The equation $\left(I_M \otimes B(\boldsymbol{\theta})\right)\mathbf{w} = \mathbb{0}_{MP\times 1}$ is equivalent to $B(\boldsymbol{\theta})W = \mathbb{0}_{N\times Q}$, which implies that

$$W = UV, \tag{E.15}$$

(by definition of U) for some nonzero $M \times M$ matrix V. Then, the equation $D_M^\top\left(I_M \otimes U^\top\right)\mathbf{w} = \mathbb{0}$ simply says that $U^\top W$ is skew-symmetric, i.e., by Eq. (E.15), that V is skew-symmetric. Finally, the equation $W_M^\top\left(I_M \otimes U^\top\right)\mathbf{w} = \mathbb{0}$ is equivalent to saying that $U_0^\top W = U_0^\top UV$ is symmetric, i.e. $U_0^\top UV - V^\top U^\top U_0 = \mathbb{0}_{M\times M}$ Since V is necessarily skew-symmetric, one has

$$U_0^\top UV + VU^\top U_0 = \mathbb{0}_{M\times M}.$$

In order to show that $\frac{\partial}{\partial U} h$ is invertible, all one needs to show is that this equation has no nonzero solutions. This is the case, e.g. if $U_0^\top U$ has no negative eigenvalues [28], which we now show to be the case : let's assume that there exists a nonzero $\mathbf{v} \in \mathbb{R}^M$ such that $U_0^\top U\mathbf{v} = \mathbb{0}_{M\times 1}$, so that $\mathcal{N}ull\left(U_0^\top\right) \cap \mathcal{S}pan\left(U\right) \neq \{\mathbb{0}_{N\times 1}\}$, or, equivalently, $\mathcal{S}pan\left(B_0^\top\right) \cap \mathcal{N}ull\left(B\right) \neq \{\mathbb{0}_{N\times 1}\}$, which is also equivalent to saying that there exists a nonzero $\mathbf{w} \in \mathbb{R}^P$ such that $BB_0^\top\mathbf{w} = \mathbb{0}_{P\times 1}$ (remember that $B_0^\top$, having full row rank, has rank P).

Now, $B\left(\boldsymbol{\theta}\right)B_0^\top$ is a continuous function of $\boldsymbol{\theta}$, and, in $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, $B\left(\boldsymbol{\theta}_0\right)B_0^\top$ has only positive eigenvalues, because B_0 is full-rank. Since the smallest eigenvalue of a matrix can be shown to be a continuous function of the elements of the matrix [77, 44, Part 2], the smallest eigenvalue of $B\left(\boldsymbol{\theta}\right)B_0^\top$ is a continuous function of Θ, and, on a neighborhood of $\boldsymbol{\theta}_0$, the smallest eigenvalue of this matrix is necessarily positive. □

Solving $X + Y^\top C - C^\top Y = \mathbb{0}$ with skew-symmetric C and X In this section, we give a computationally efficient way of solving

$$X + Y^\top C - C^\top Y = \mathbb{0}_{M\times M} \tag{E.16}$$

when X and C are skew-symmetric and Y is a symmetric $M \times M$ matrix. Note that, since $C = -C^\top$, any solution to (E.16) is also a solution to :

$$X + Y^\top C + CY = \mathbb{0}_{M\times M}. \tag{E.17}$$

A solution to (E.17) is ([28, p. 38]) $\mathbf{c} = -\left(I_M \otimes Y^\top + Y^\top \otimes I_M\right)^{-1}\mathbf{x}$, where $\mathbf{c} = \mathrm{vec}\left(C\right)$ and $\mathbf{x} = \mathrm{vec}\left(X\right)$. The computation cost is (approximately) that of a $M^2 \times M^2$ matrix inversion, that is, in the order $\left(M^2\right)^3$.

We now show that C can be computed approximately at the cost of the inversion of a $M\left(M-1\right)/2\times M\left(M-1\right)/2$ matrix, that is, in the order of $M^6/8$. We first note, as in Section E.6, that (E.16) is equivalent to :

$$W_M\tilde{\mathbf{x}} + \left(I_{M^2} - K_M\right)\left(I_M \otimes Y^\top\right)W_M\tilde{\mathbf{c}} = \mathbb{0},$$

where $\tilde{\mathbf{x}} = \mathrm{vecl}\left(X\right)$ and $\tilde{\mathbf{c}} = \mathrm{vecl}\left(X\right)$ are vectors holding the $M\left(M-1\right)/2$ distinct off-diagonal elements of the skew-symmetric matrices X and C : $\tilde{\mathbf{x}} = \left[X_{2,1},\ldots,X_{M,1},X_{3,2},\ldots,X_{M,M-1}\right]^\top$, and W_M is the $M^2 \times M\left(M-1\right)/2$ matrix such that $\mathrm{vec}\left(X\right) = W_M\tilde{\mathbf{x}}$. Then, noting that $W_M^+W_M\tilde{\boldsymbol{\theta}} =$

$\boldsymbol{I}\tilde{\boldsymbol{\theta}}$, that $W_M^+ K_M = -W_M^+$ and that $W_M^+ = \frac{1}{2} W_M^\top$, one has :

$$\tilde{\boldsymbol{\theta}} + W_M^\top \left(I_M \otimes Y^\top \right) W_M \tilde{\mathbf{c}} = \mathbb{0}.$$

It is thus possible to compute $\tilde{\mathbf{c}} = \left(W_M^\top \left(I_M \otimes Y^\top \right) W_M \right)^{-1} \tilde{\boldsymbol{\theta}}$, by inverting a $M (M-1)/2 \times M (M-1)/2$ matrix.

Bibliography

[1] http://www.shapecapture.com.

[2] M. Berger and B. Gostiaux. *Differentiable geometry : Manifolds, Curves and Surfaces.* Springer-Verlag, 1988.

[3] D. Bondyfalat and S. Bougnoux. Imposing euclidean constraint during self-calibration processes. In *Proc. SMILE Workshop*, pages 224–235, Freiburg, Germany, 1998.

[4] D. Bondyfalat, B. Mourrain, and T. Papadopoulo. An application of automatic theorem proving in computer vision. In *Automated Deduction in Geometry*, pages 207–231, 1998.

[5] B. Boufama, R. Mohr, and F. Veillon. Euclidean constraints for uncalibrated reconstruction. In *Proc. ICCV*, pages 466–470, Berlin, Germany, May 1993.

[6] Sylvain Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *Proceedings of the 6th ICCV*, pages 790–796, Jan 1998.

[7] Ray M. Bowen and C.-C. Wang. *Introduction to vectors and tensors*, volume 1. Plenum Press, 1976,1980.

[8] R. A. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, (17):285–348, 1981.

[9] B. Caprile and V. Torre. Using vanishing points for camera calibration. *IJCV*, 4:127–140, 1990.

[10] Chu-Song Chen, Chi-Kuo Yu, and Yi-Ping Hung. New calibration-free approach for augmented reality based on parameterized cuboid structure. In *ICCV*, volume 1, pages 30–37, 1999.

[11] R. Cipolla and E. Boyer. 3D model acquisition from uncalibrated images. In *IAPR Workshop on Machine Vision Applications*, pages 559–568, 1998.

[12] Roberto Cipolla, Duncan Robertson, and Edmond Boyer. Photobuilder - 3D models of architectural scenes from uncalibrated images. In *International Conference on Multimedia Computing and Systems,Firenze*, volume 1, pages 25–31, 1999.

[13] A. Criminisi. *Accurate visual metrology from single and multiple uncalibrated images.* PhD thesis, University of Oxford, 1999.

[14] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. In *Proc. ICCV*, pages 434–441, Corfu, Greece, 1999.

[15] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *IJCV*, 40(2):123–148, 2000.

[16] A. Criminisi and A. Zisserman. Shape from texture : homogeneity revisited. In *Proc. BMVC*, pages 82–91, 2000.

[17] A. Criminisi, A. Zisserman, L. Van Gool, Bramble S., and D. Compton. A new approach to obtain height measurements from video. In *Proc. of SPIE, Boston, MA, USA*, volume 3576, 1-6 November 1998.

[18] R. Lane Dailey. Eigenvector derivatives with repeated eigenvalues. *AIAA Journal*, 27(4):486–491, 1989.

[19] Paul E. Debevec. *Modeling and Rendering Architecture from Photographs.* PhD thesis, University of California at Berkeley, Computer Science Division, Berkeley CA, 1996.

[20] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs : A hybrid geometry- and image-based approach. Technical Report CSD/96-893, UCB, 1996.

[21] Rachid Deriche, Régis Vaillant, and Olivier Faugeras. From noisy edge points to 3D reconstruction of a scene: A robust approach and its uncertainty analysis. In *Scandinavian Conference on Image Analysis*, pages 225–232, 1991.

[22] J. W. Eaton. Octave—a high-level interactive language for numerical computations. Technical report, The University of Texas Center for Control and Systems Research, Austin, Texas, 1993. http://www.octave.org/.

[23] O. Faugeras. *Three Dimensional Computer Vision.* MIT Press, 1993.

[24] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *Proc. of the 2nd. European Conference on Computer Vision*, Lecture Notes in Computer Science, pages 563–578, Santa Margherita, Italy, 1992. Springer-Verlag.

[25] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials.* Academic Press, 1982.

[26] G. H. Golub and C. F. Van Loan. *Matrix computations.* Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, Baltimore, MD, USA, third edition, 1996.

[27] G. Gracie. Analytical photogrammetry applied to single terrestrial photograph mensuration. In *Proc XIth Intl. Conf. of Photogrammetry*, Lausanne, Switzerland, July 1968.

[28] Alexander Graham. *Kronecker products and matrix calculus with applications.* Wiley, 1981.

[29] Etienne Grossmann, Diego Ortin, and José Santos-Victor. Algebraic aspects of reconstruction of structured scenes from one or more views.

[30] Etienne Grossmann, Diego Ortin, and José Santos-Victor. Algebraic aspects of reconstruction of structured scenes from one or more views. In *Proc. BMVC*, volume 2, pages 633–642, 2001.

[31] Etienne Grossmann, Diego Ortin, and José Santos-Victor. Reconstruction of structured scenes from one or more views : Nature of solutions. *Technical Report*, 2001. Available upon demand.

[32] Etienne Grossmann and José Santos-Victor. The precision of 3D reconstruction from uncalibrated views. In Paul H. Lewis and Mark S. Nixon, editors, *proc. British Machine Vision Conference*, volume 1, pages 115–124. British Machine Vision Association, September 1998.

[33] Etienne Grossmann and José Santos-Victor. Uncertainty analysis of 3D reconstruction from uncalibrated views. *Image and Vision Computing*, 18:685–696, May 2000.

[34] R.M. Haralick. Propagating covariance in computer vision. In *Proc. Workshop on Performance Characteristics of Vision Algorithms*, pages 1–12, Cambridge, 1996.

[35] R. Hartley and A. Zissermann. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[36] Richard I. Hartley. In defense of the 8-point algorithm. In IEEE, editor, *ICCV*, pages 1064–1070, Cambridge, MA., June 1995.

[37] Anders Heyden. On the consistency of line-drawings, obtained by projections of piecewise planar objects. *Journal of Mathematical Imaging and Vision*, 6(4):393–412, 1996.

[38] R. Horst, P.M. Pardalos, and N.V. Thoai. *Introduction to global optimization.* Kluwer, 1995.

[39] Peter J. Huber. *Robust statistics.* John Wiley and sons, 1981.

[40] K.G.J. Jacobi. Uber ein leichtes verfahren die in der theorie der saecularstoerungen vorkommenden gleichungen numerisch aufzuloesen. *Zeitrschriift fur Reine und Angewandte Mathematik*, 30:51–59, 1846. Also, NASA TT.F-13,666 June 1971.

[41] David Jelinek and C. J. Taylor. Reconstruction of linearly parameterized models from single images with a camera of unknown focal length. In *Proc. CVPR*, pages 346–352, 1999.

[42] David Jelinek and C. J. Taylor. Reconstruction of linearly parameterized models from single images with a camera of unknown focal length. *Trans. PAMI*, 23(7):767–774, July 2001.

[43] Kenichi Kanatani. Statistical analysis of focal-length calibration using vanishing points. *IEEE Transactions on Robotics and Automation*, 8(6):767–775, December 1992.

[44] Tosio Kato. *Perturbation Theory for Linear Operators.* Springer, 1966.

[45] P. Lancaster. On eigenvalues of matrices dependent on a parameter. *Numerische Mathematik*, 6:377–387, 1964.

[46] E.L. Lehmann. *Theory of point estimation.* ??, 1983.

[47] Reimar K. Lenz and Roger Y. Tsai. Techniques for calibration of the scale factor and image center for high-accuracy 3D machine vision metrology. *IEEE Trans PAMI*, 10(5):713–720, Sept 88.

[48] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Proc. EuroGraphics*, volume 18, pages 39–50, September 1999.

[49] David G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. PAMI*, 14(5):441–450, May 1991.

[50] Jan R. Magnus and Heinz Neudecker. *Matrix differential calculus with applications in statistics and econometrics.* John Wiley & Sons, 1999.

[51] The MathWorks. *Matlab Reference Guide.* The MathWorks, Inc, Natick MA, 1992. http://www.mathworks.com/.

[52] Richard B. Nelson. Simplified calculation of eigenvectors derivatives. *AIAA Journal*, 14(9):1201–1205, 1976.

[53] I.U. Ojalvo. Efficient computation of modal sensitivities for systems with repeated frequencies. *AIAA Journal*, 26(3):361–366, 1988.

[54] T. Papadopoulo and M.I.A. Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. In *ECCV (1)*, pages 554–570, 2000.

[55] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proceedings of the 6th ICCV*, pages 90–95, Jan 1998.

[56] William H. Press, Saul A. Teutolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes, The Art of Scientific Computing, 2nd edition.* Cambridge University Press, 1992.

[57] Ian Reid and Andrew Zisserman. Goal-directed video metrology. In *Proc. ECCV*, volume II, pages 647–658, 1996.

[58] F. Rellich. Störungstheorie der spektralzerlegung, i-v. *Math. Ann.*, (113,113,116,117,118):600–619,677–685,555–570,356–382,462–484, 1937-1942.

[59] R.Fletcher. *Practical methods of optimization.* John Wiley and Sons, 2 edition, 1982.

[60] D. Robertson and R. Cipolla. An interactive system for constraint-based modelling. In *BMVC2000*, volume 2, pages 536–545, 2000.

[61] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection.* John Wiley & Sons, 1987.

[62] Heung-Yeung Shum, Mei Han, and Richard Szeliski. Interactive construction of 3D models from panoramic mosaics. In *CVPR*, pages 427–433, 1998.

[63] Heung-Yeung Shum, Richard Szeliski, Simon Baker, Mei Han, and P. Anandan. Interactive 3D modeling from multiple images using scene regularities. In *Proc. SMILE Workshop*, pages 236–252, Freiburg, Germany, June 1998.

[64] G. Sparr. Depth computations from polyhedral images. In *Proc. ECCV*, pages 378–386, 1992.

[65] Gunnar Sparr. Euclidean and affine structure/motion for uncalibrated cameras from affine shape and subsidiary information. In *Proc. SMILE Workshop*, pages 187–207, Freiburg, Germany, June 1998.

[66] Peter Sturm. A method for 3D reconstruction of piecewise planar objects from single panoramic images. In *Proc. Workshop on Omnidirectional Vision*, pages 119–126, Hilton Head Island, 2000.

[67] P.F. Sturm and S.J. Maybank. A method for interactive 3D reconstruction of piecewise planar objects from single views. In *Proc. BMVC*, pages 265–274, 1999.

[68] R. Szeliski and P.H.S Torr. Geometrically constrained structure from motion: points on planes. In *Proc. SMILE Workshop*, pages 171–186, Freiburg, Germany, 1998.

[69] Camillo J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *CVIU*, 80(349-363), 2000.

[70] Menno Tienstra. *A method for the calculation of orthogonal transformation matrices, and its application to photogrammetry and other disciplines.* Waltman Delft, 1969.

[71] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams : a factorization method. Technical Report TR 92-1270 and CMU-CS-92-104, Cornell University and Carnegie Mellon University, 1992.

[72] P. H. S. Torr, A. Zisserman, and S. Maybank. Robust detection of degenerate configurations for the fundamental matrix. *CVIU*, 71(3):312–333, 1998.

[73] F.A. van den Heuvel. Vanishing point detection for architectural photogrammetry. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXII, pages 652–659. ISPRS, 1998.

[74] F.A. van den Heuvel. A line-photogrammetric mathematical model for the reconstruction of polyhedral objects. In Sabry F. El-Hakim, editor, *Videometrics VI, Proc. of SPIE*, volume 3641, pages 60–71, 1999. ISBN 0-8194-3112-5.

[75] F.A. van den Heuvel. Trends in cad-based photogrammetric measurement. *International Archives of Photogrammetry and Remote Sensing*, 33(5/2):852–863, 2000.

[76] Marta Wilczkowiak, Edmond Boyer, and Peter Sturm. Camera calibration and 3D reconstruction from single images using parallelepipeds. In *ICCV 2001*, volume 1, pages 142–148, 2001.

[77] J. H. Wilkinson. *The algebraic eigenvalue problem.* Oxford: Clarendon, 1965,1988.

[78] J.R. Williamson and M.H. Brill. *Dimensional Analysis through Perspective A Reference Manual.* American Society for Photogrammetry and Remote Sensing, 1990. ISBN 0-8403-5673-0.