Generalizations of the Projective Reconstruction Theorem

Behrooz Nasihatkon

A thesis submitted for the degree of
Doctor of Philosophy,
The Australian National University

July 2014

Declaration

The contents of this thesis are mainly extracted from the following papers:

- Behrooz Nasihatkon, Richard Hartley and Jochen Trumpf, "On Projective Reconstruction In Arbitrary Dimensions" **submitted** to CVPR 2014.
- Behrooz Nasihatkon, Richard Hartley and Jochen Trumpf, "A Generalized Projective Reconstruction Theorem" **submitted** to the International Journal of Computer Vision (IJCV).

In addition to the above, the author has produced the following papers during his PhD studies.

- Behrooz Nasihatkon and Richard Hartley, "Move-Based Algorithms for the Optimization of an Isotropic Gradient MRF Model," International Conference on Digital Image Computing Techniques and Applications (DICTA), 2012.
- Behrooz Nasihatkon and Richard Hartley "Graph connectivity in sparse subspace clustering," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

Except where otherwise indicated, this thesis is my own original work.

B. Nasihatkon

Behrooz Nasihatkon
25 July 2014

to my parents and my wife...

Acknowledgments

I had the great opportunity to work under the supervision of Professor Richard Hartley. I like to thank him for his supportive attitude, superb guidance and helpful advice. I learnt from him how to think analytically and systematically when dealing with research problems, and how to merge intellect and intuition to tackle them. His mathematical comprehension, immense knowledge, clarity of thought and vision made my PhD studies a great experience.

I also like give my special thanks to Dr. Jochen Trumpf, my co-supervisor, for his guidance and help, and for the valuable discussions we had during my PhD studies. His mathematical expertise, brilliant questions, invaluable comments, and inspiring tips and suggestions have significantly improved the quality of my PhD.

I like to thank Professor Rene Vidal for offering me a visiting research scholar position at the Computer Vision Lab in the Johns Hopkins University, and also for his help and support, and the insightful discussions we had during my visit. I also thank Dr. Hongdong Li for his feedback and comments on this thesis.

I would like to acknowledge the academic, technical and financial support of the Australian National University and National ICT Asutralia. I also want to thank my fellow labmates Khurrum, Samunda, Mohammad, Sara, Dana, Adnan, Ahmed, Cong, Lin and others for their kindness and the friendly atmosphere they contributed to. In particular, I like to express my gratitude to my close friend Khurrum Aftab for his caring attitude and cheerful character both inside and outside the Lab.

I consider myself incredibly lucky to have been surrounded by a fabulous group of friends who made my time at the ANU enjoyable and with whom I share many cherished memories. I like to thank my good friends Morteza, Mohammad Esmaeilzadeh, Hamid, Mohammad Najafi, Ehsan, Mohammad Saadatfar, Mehdi, Mohmmadreza, Alireza, and their family for the good times we had together. Especially, I am grateful to Mostafa Moghaddam, Mohsen Zamani and Ehsan Abbasnejad for their friendship and care. I appreciate the help and support of my good friends Mohammad Deghat, Alireza Motevalian and Zahra Zamani who helped me get settled in Canberra and whose friendship I have enjoyed to this day. Particularly, I like to thank my close friend Mohammad Deghat for his nice personality and helpful attitude.

I am very grateful to my wife, Fatemeh for all the sacrifices she made to help me finish my PhD. I thank her for her love, patience and caring. My deepest gratitude belongs to my parents for their love, encouragement and support throughout all stages of my life.

Abstract

We present generalizations of the classic theorem of projective reconstruction as a tool for the design and analysis of the projective reconstruction algorithms. Our main focus is algorithms such as bundle adjustment and factorization-based techniques, which try to solve the projective equations directly for the structure points and projection matrices, rather than the so called tensor-based approaches. First, we consider the classic case of 3D to 2D projections. Our new theorem shows that projective reconstruction is possible under a much weaker restriction than requiring, a priori, that all estimated projective depths are nonzero. By completely specifying possible forms of wrong configurations when some of the projective depths are allowed to be zero, the theory enables us to present a class of depth constraints under which any reconstruction of cameras and points projecting into given image points is projectively equivalent to the true camera-point configuration. This is very useful for the design and analysis of different factorization-based algorithms. Here, we analyse several constraints used in the literature using our theory, and also demonstrate how our theory can be used for the design of new constraints with desirable properties.

The next part of the thesis is devoted to projective reconstruction in arbitrary dimensions, which is important due to its applications in the analysis of dynamical scenes. The current theory, due to Hartley and Schaffalitzky, is based on the Grassmann tensor, generalizing the notions of Fundamental matrix, trifocal tensor and quardifocal tensor used for 3D to 2D projections. We extend their work by giving a theory whose point of departure is the projective equations rather than the Grassmann tensor. First, we prove the *uniqueness* of the Grassmann tensor corresponding to each set of image points, a question that remained open in the work of Hartley and Schaffalitzky. Then, we show that projective equivalence follows from the set of projective equations, provided that the depths are all nonzero. Finally, we classify possible wrong solutions to the *projective factorization* problem, where not all the projective depths are restricted to be nonzero.

We test our theory experimentally by running the factorization based algorithms for rigid structure and motion in the case of 3D to 2D projections. We further run simulations for projections from higher dimensions. In each case, we present examples demonstrating how the algorithm can converge to the degenerate solutions introduced in the earlier chapters. We also show how the use of proper constraints can result in a better performance in terms of finding a correct solution.

Contents

List of Figures

Introduction

1.1 Thesis Statement

The subject of this thesis is generalizations to the Theorem of Projective Reconstruction with the purpose of providing a theoretical basis for a wider range of projective reconstruction algorithms, including projective factoirization. We investigate the classic case of 3D to 2D projections in detail, and further, extend the theory to the general case of arbitrary dimensions.

1.2 Introduction

The main purpose of this thesis is to extend the theory of projective reconstruction for multiple projections of a set of scene points. A set of such projections can be represented as

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j \tag{1.1}$$

for $i = 1,\ldots,m$ and $j = 1,\ldots,n$, where $\mathbf{X}_j \in \mathbb{R}^r$ are high-dimensional (HD) points, $\mathtt{P}_i \in \mathbb{R}^{s_i \times r}$ are projection matrices, $\mathbf{x}_{ij} \in \mathbb{R}^{s_i}$ are image points and λ_{ij}-s are nonzero scalars known as *projective depths*. Each point $\mathbf{X}_j \in \mathbb{R}^r$ is a certain representation of a projective point in $\mathbb{P}^{r-1}$ in homogeneous coordinates. Similarly, each $\mathbf{x}_{ij} \in \mathbb{R}^{s_i}$ represents a point in $\mathbb{P}^{s_i-1}$. In the classic case of 3D to 2D projections we have $r = 4$ and $s_i = 3$ for all i. The problem of *projective reconstruction* is to obtain the projection matrices $\mathtt{P}_i$, the HD points $\mathbf{X}_j$ and the projective depths λ_{ij}, up to a projective ambiguity, given the image points $\mathbf{x}_{ij}$.

The relations (1.1) can be looked at from a *factorization* point of view. By writing (1.1) in matrix form we have

$$\Lambda \odot [\mathbf{x}_{ij}] = \mathtt{P}\,\mathtt{X}, \tag{1.2}$$

where the operator "$\odot$" multiplies each element λ_{ij} of the depth matrix Λ by its corresponding image point $\mathbf{x}_{ij}$, that is $\Lambda \odot [\mathbf{x}_{ij}] = [\lambda_{ij}\mathbf{x}_{ij}]$, the matrix $\mathtt{P} = \text{stack}(\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_n) \in \mathbb{R}^{(\sum_i s_i)\times r}$ is the vertical stack of the projection matrices $\mathtt{P}_i$, and $\mathtt{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n] \in \mathbb{R}^{r\times n}$ is the horizontal concatenation of the HD points $\mathbf{X}_j$. This

relation expresses the idea behind the *factorization-based* approaches to projective reconstruction: find Λ such that $\Lambda \odot [\mathbf{x}_{ij}]$ can be factorized as the product of a $(\sum_i s_i) \times r$ matrix $\mathtt{P}$ by an $r \times n$ matrix $\mathtt{X}$, or equivalently, the rank of $\Lambda \odot [\mathbf{x}_{ij}]$ is less than or equal to r.

Tensor-based techniques The conventional way of dealing with the projective reconstruction problem is using the tensor-based approaches. In such approaches, first a specific tensor is estimated from image point correspondences in a subset of views. The projection matrices then are extracted from the tensor. Having the projection matrices, the points can be estimated through a triangulation procedure. In 3D to 2D projections, the possible tensors are the bifocal tensor (fundamental matrix), trifocal tensor and quadrifocal tensor which are respectively created from the point correspondences among pairs, triples and quadruples of images [Hartley and Zisserman, 2004]. Similarly, other types of tensors can be used for projections in other dimensions. Hartley and Schaffalitzky [2004] unify different types of tensors used for different dimensions under the concept of the Grassmann tensor.

Tensor-based projective reconstruction is sometimes not accurate enough, especially in the presence of noise. One problem is imposing necessary nonlinear restrictions on the form of the tensor in the course of its computation from image point correspondences. As a simple example, the fundamental matrix (bifocal tensor) needs to be of rank 2. This is the only required constraint. The number of such *internal constraints* increases dramatically with the dimensionality of the multi-view tensor. For example, the trifocal tensor is known to have 8 internal constraints. For the quadrifocal tensor this number is 51 (see [Hartley and Zisserman, 2004, Sect. 17.5]). Another issue is that for projections from $\mathbb{P}^{r-1}$, at most r views can contribute to the computation of each tensor. For example, for 3D to 2D projections, a tensor can be defined only for up to four views. This prevents us from making use of the whole set of image points from all views to reduce the estimation error. This has led to the use of other approaches such as bundle adjustment [Triggs et al., 2000] and projective factorization [Sturm and Triggs, 1996; Triggs, 1996; Mahamud et al., 2001; Oliensis and Hartley, 2007], in which the projection equations (1.1) are directly solved for projection matrices $\mathtt{P}_i$, HD points $\mathbf{X}_j$ and projective depths λ_{ij}. Analysing such methods requires a theory which derives the projective reconstruction from the projection equations (1.1), rather than from the Grassmann tensor. Providing such a theory is the main object of this thesis.

Projective Factorization We consider, in detail, the classic case of 3D to 2D projections from a projective factorization point of view illustrated in (1.2). Many factorization-based approaches have been suggested to solve (1.2) [Sturm and Triggs, 1996; Triggs, 1996; Ueshiba and Tomita, 1998; Heyden et al., 1999; Mahamud et al., 2001; Oliensis and Hartley, 2007; Dai et al., 2013]. However, in such algorithms, it is hard to impose the geometric constraints such as full-row-rank camera matrices $\mathtt{P}_i$ and all nonzero projective depths λ_{ij}. Completely neglecting such constraints, however, allows wrong solutions to (1.2) which are not projectively equivalent to the

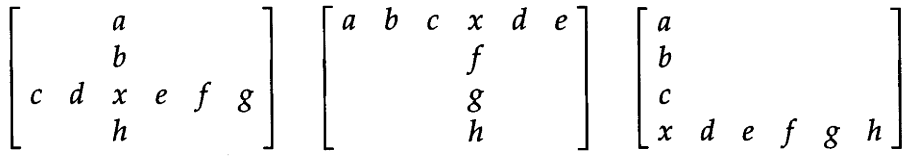


Figure 1.1: Examples of 4×6 cross-shaped matrices. In cross-shaped matrices all elements of the matrix are zero, except those belonging to a special row r or a special column c of the matrix. The elements of the r-th row and the c-th column are all nonzero, except possibly the central element located at position (r, c). In the above examples, the blank parts of the matrices are zero. The elements $a, b, \ldots, h$ are all nonzero, while x can have any value (zero or nonzero). We will show that one class of degenerate solutions to the projective factorization problem (1.2) happens when the estimated depth matrix $\hat{\Lambda}$ takes a cross-shaped form.

true configuration of camera matrices and points. Therefore, without putting extra constraints on the depth matrix the above problem can lead to false solutions.

Degenerate solutions The main source of the false solutions in the factorization-based methods is the possibility of having zero-elements in Λ. One can simply see that setting Λ, $\mathtt{P}$ and $\mathtt{X}$ all equal to zero provides a solution to (1.2). Another trivial solution, as noted by Oliensis and Hartley [2007], occurs when Λ has all but four zero columns. In general, it has been noticed that false solutions to (1.2) can happen when some rows or some columns of the depth matrix are zero. There has been no research, however, specifying all possible false solutions to the factorization equation (1.2). Here, in addition to the cases where the estimated depth matrix has some zero rows or some zero columns, we present a less trivial class of false solutions where the depth matrix has a cross-shaped structure (see Fig. 1.1). We shall further show that all possible false solutions to the projective factorization problem (1.2) are confined to the above cases, namey when

1. the depth matrix $\hat{\Lambda}$ has one or more zero rows,
2. the depth matrix $\hat{\Lambda}$ has one or more zero columns,
3. the depth matrix $\hat{\Lambda}$ is cross-shaped.

Therefore, by adding to (1.2) a constraint on the depth matrix which allows at least one correct solution, and excludes the three cases above, any solution to the factorization problem (1.2) is a correct projective reconstruction.

Constraining projective depths Here, we do not thoroughly deal with the question of how to solve (1.2) and are mostly concerned about the classification of its false solutions, and the constraints which can avoid them. However, we have to be realistic about choosing proper constraints. The constraints have to possess some desirable properties to make possible the design of efficient and effective algorithms for solving (1.2). As a trivial example it is essential for many iterative algorithms that the

$$\begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & 1 & & \\ & & & 1 & 1 & \\ & & & & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 & & & \\ & & & 1 & & \\ & & & 1 & & \\ & & & 1 & 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & & & & & \\ 1 & & & & & \\ 1 & & & & & \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(a) (b) (c)

Figure 1.2: Examples of 4×6 step-like mask matrices. Blank parts of the matrices indicate zero values. A step-like matrix contains a chain of ones, starting from its upper left corner and ending at its lower right corner, made by making rightward and downward moves only. An exclusive step-like mask is one which is not cross-shaped. In the above, (a) and (b) are samples of an exclusive step-like mask while (c) is a nonexclusive one. Associated with an $m \times n$ step-like mask $\mathtt{M}$, one can put a constraint on an $m \times n$ depth matrix $\hat{\Lambda}$ in the form of fixing the elements of $\hat{\Lambda}$ to 1 (or some nonzero values) at sites where $\mathtt{M}$ has ones. For an exclusive step-like mask, this type of constraint rules out all the wrong solutions to the factorization-based problems.

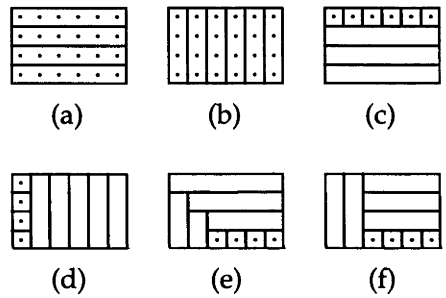


Figure 1.3: Examples of tiling a 4×6 depth matrix with row and column vectors. The associated constraint is to force every tile of the depth matrix to have a unit (or a fixed) norm. This gives a compact constraint space. (More details in Sect. 3.3.1.4.)

constraint space is closed. As nearly all factorization-based algorithms are solved iteratively, this can guarantee that the algorithm does not converge to something outside the constraint space.

Linear equality constraints A major class of desirable constraints for projective factorization problems consists of linear equality constraints. The corresponding affine constraint space is both closed and convex, and usually leads to less complex iterations. We shall show that the linear equality constraints that are used so far in factorization-based reconstruction allow for cross-shaped depth matrices and hence cannot rule out false solutions. We shall further introduce *step-like constraints*, a class of linear equality constraints of a form fixing certain elements of the depth matrix, which provably avoid all the degenerate cases in the factorization problem (see Fig. 1.2). The element-wise nature of these constraints makes the implementation of

the associated factorization-based algorithms very simple.

Compact constraints Another desirable property for the constraint space, which is mutually exclusive with being an affine subspace, is compactness. The importance of a compact constraint space is that certain convergence properties can be proved for a large class of iterative descent algorithms when the sequence of solutions lie inside a compact set. One can think of many compact constraints, however, the important issue is that the constraint needs to be efficiently implementable with a factorization algorithm. Two examples of such constraints are presented in [Heyden et al., 1999] and [Mahamud et al., 2001], in which, respectively, all rows and all columns of the depth matrix are forced to have a fixed (weighted) l^2-norm. In each case, every iteration of the factorization algorithm requires solving a number of eigenvalue problems. Mahamud et al. [2001] prove the convergence of their algorithm to local minima using the General Convergence Theorem [Zangwill, 1969; Luenberger, 1984]. However, these constraints allow zero columns or zero rows in the depth matrix, as well as cross-shaped structures. In this thesis, we combine the constraints used in [Heyden et al., 1999] and [Mahamud et al., 2001], in the sense of *tiling* the matrix with row and column vectors and requiring each tile to have a unit (or fixed) norm (see Fig. 1.3). With a proper tiling, convergence to configurations with zero rows and zero columns is ruled out. Such tilings still allow for cross-shaped structures, however, as shown in Fig. 1.3, the number of possible cross-shaped structures is limited.

Arbitrary dimensional projections The rest of the thesis is devoted to the projections in arbitrary dimensions. The job is harder in this case because the theory has not been developed to the extent it has been for 3D to 2D projections.

The need for projective reconstruction in higher dimensions comes from the applications in the analysis of dynamic scenes, when the motion in the scene is not globally rigid. Wolf and Shashua [2002] consider a number of different structure and motion problems in which the scene observed by a perspective camera is non-rigid. They show that all the given problems can be modeled as projections from a higher-dimensional projective space $\mathbb{P}^k$ into $\mathbb{P}^2$ for $k = 3,4,5,6$. They use tensorial approaches to address each of the problems. Xiao and Kanade [2005], Vidal and Abretske [2006] and Hartley and Vidal [2008] considered the problem of perspective nonrigid deformation, assuming that the scene deforms as a linear combination of k different linearly independent basis shapes. They show that the problem can be modeled as projections from $\mathbb{P}^{3k}$ to $\mathbb{P}^2$.

Such applications demonstrate the need for a general theory of projective reconstruction for arbitrary dimensional spaces. Hartley and Schaffalitzky [2004] present a novel theory to address the projective reconstruction for general projections. Their theory unifies the previous work by introducing the Grassmann tensor, which generalizes the concepts of bifocal, trifocal and quadrifocal tensors used in $\mathbb{P}^3 \rightarrow \mathbb{P}^2$ projections, and other tensors used for special cases in other dimensions. The central theorem in Hartley and Schaffalitzky [2004] suggests that the projection matrices can

be obtained up to projectivity from the corresponding Grassmann tensor. As we discussed, the tensor methods sometimes have problems with accuracy which leads to the use of other methods such as bundle adjustment and projective factorization, in which the projection equations (1.1) are directly solved for projection matrices $\mathtt{P}_i$, HD points $\mathbf{X}_j$ and projective depths λ_{ij}. The current theory of projective reconstruction, however, is not sufficient for the analysis of such methods.

Here, we give a theory which deduces projective reconstruction from the set of equations (1.1). As a first step, we need to answer a question which is left open in [Hartley and Schaffalitzky, 2004], namely whether, for a general setup, the set of image points $\mathbf{x}_{ij}$ uniquely determine the Grassmann tensor, up to a scaling factor. Notice that this is important even for tensor-based projective reconstruction. Our theory in section 4.2.1 gives a positive answer to this question.

The second question is whether all configurations of projective matrices and HD points projecting into the same image points $\mathbf{x}_{ij}$ (all satisfying (1.1) with nonzero depths λ_{ij}) are projectively equivalent. This is important for the analysis of bundle adjustment as well as factorization-based approaches. Answering such a simple question is by no means trivial. Notice that the uniqueness of the Grassmann tensor is not sufficient for proving this, as it does not rule out the existence of degenerate solutions $\{\mathtt{P}_i\}$ whose corresponding Grassmann tensor is zero. This thesis gives a positive answer to this question as well, as a consequence of the theory presented in section 4.3.

The last issue, which only concerns the factorization-based approaches, is classifying all the degenerate solutions to the projective factorization equation (1.2). The factorization-based approaches has been used for higher dimensional projections, for example, for the recovery of nonrigid deformations [Xiao and Kanade, 2005]. Being aware of possible degenerate solutions can help us with the design of the reconstruction algorithms which are able to avoid such solutions. It turns out that the wrong solutions for arbitrary dimensional spaces can be much more complex compared to the case of 3D to 2D projections. We analyse such degenerate solutions in Sect. 4.4.

1.3 Thesis Outline

The thesis continues with Chapter 2 which gives the reader the required background, including the previous work, motivation and a more detailed explanation of the need for a generalized theory and a review of the theory and algorithms of projective reconstruction. In chapter 3 we give our theorem for the special case of 3D to 2D projections, and demonstrate how the theory can be used for the design and analysis of factorization-based projective reconstruction algorithms. Chapter 4 considers the general case of projections in arbitrary dimensional spaces. We extend the current theory on this subject and also show how some results from 3D to 2D projections follow as special cases of our theory for arbitrary dimensions. In chapter 5 we present some of the applications of higher-dimensional projections, including motion segmentation, non-rigid motion recovery and correspondence-free structure from mo-

tion. Chapter 6 contains the experimental results, where we study the application of factorization-based algorithms for the case of 3D to 2D projections for the recovery of rigid structure and motion. We also run experiments on higher-dimensional projections, and demonstrate how degenerate solutions can occur using the projective factorization algorithms.

Chapter 2

Background and Related Work

The aim of this section is to provide readers with the required background on projective reconstruction, help them with the conventions used in the thesis and make clear the importance of the research done. A review of the previous work is done in different occasions throughout the chapter.

2.1 Conventions and problem formulation

2.1.1 Notation

We use typewriter letters ($\mathtt{A}$) for matrices, bold letters ($\mathbf{a}, \mathbf{A}$) for vectors, normal letters (a, A) for scalars and upper-case normal letters (A) for sets, except for special sets like the real space $\mathbb{R}$ and the projective space $\mathbb{P}$. We use calligraphic letters ($\mathcal{A}$) for both tensors and mappings (functions). To refer to the column space, row space and null space of a matrix $\mathtt{A}$ we respectively use $\mathcal{C}(\mathtt{A})$, $\mathcal{R}(\mathtt{A})$ and $\mathcal{N}(\mathtt{A})$. The vertical concatenation of a set of matrices $\mathtt{A}_1, \mathtt{A}_2, \ldots, \mathtt{A}_m$ with compatible size is denoted by $\text{stack}(\mathtt{A}_1, \ldots, \mathtt{A}_m)$.

2.1.2 Genericity

We make use of the terms "generic" and "in general position" for entities such as points, matrices and subspaces. By this term we mean that they belong to an open and dense subset of their ambient space. This *generic* subset in some occasions are explicitly determined using a set of generic properties, and in some cases, we just use the term generic without mentioning any properties. In such cases the generic subset is implicitly determined from the properties assumed as a consequence of genericity in our proofs.

2.1.3 The projection-point setup

Here, we are dealing with multiple projection from a higher-dimensional space $\mathbb{P}^{r-1}$ to lower-dimensional spaces $\mathbb{P}^{s_i-1}$. More precisely, we have a set of n higher-dimensional (HD) projective points $\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2, \ldots, \tilde{\mathbf{X}}_n \in \mathbb{P}^{r-1}$ and a set of m projective

transformations $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_m$ with $\mathcal{P}_i \colon \mathbb{P}^{r-1} \to \mathbb{P}^{s_i-1}$. Each point $\tilde{\mathbf{X}}_j$ is mapped by each projection $\mathcal{P}_i$ to a lower-dimensional projective point $\tilde{\mathbf{x}}_{ij} \in \mathbb{P}^{s_i-1}$, that is

$$\tilde{\mathbf{x}}_{ij} = \mathcal{P}_i(\tilde{\mathbf{X}}_j). \tag{2.1}$$

The problem of *projective reconstruction* is to recover the projective maps $\mathcal{P}_i$ and HD points $\tilde{\mathbf{X}}_j$ given the projected points $\tilde{\mathbf{x}}_{ij}$. Obviously, the best we can do given only $\tilde{\mathbf{x}}_{ij}$-s is the recovery of $\mathcal{P}_i$-s and $\tilde{\mathbf{X}}_j$-s up to a projective ambiguity, as one can write

$$\tilde{\mathbf{x}}_{ij} = \mathcal{P}_i(\tilde{\mathbf{X}}_j) = \mathcal{P}_i(\mathcal{H}(\mathcal{H}^{-1}(\tilde{\mathbf{X}}_j))) \tag{2.2}$$

for any invertible projective transformation $\mathcal{H} \colon \mathbb{P}^{r-1} \to \mathbb{P}^{r-1}$. Therefore, if $(\{\mathcal{P}_i\}, \{\tilde{\mathbf{X}}_j\})$ is one possible solution to projective reconstruction, so is $(\{\mathcal{P}_i\mathcal{H}\}, \{\mathcal{H}^{-1}(\tilde{\mathbf{X}}_j)\})$.

To deal with the projections algebraically, we use homogeneous coordinates representing the projective points $\tilde{\mathbf{X}}_j \in \mathbb{P}^{r-1}$ and $\tilde{\mathbf{x}}_{ij} \in \mathbb{P}^{s_i-1}$ by the real vector $\mathbf{X}_j \in \mathbb{R}^r$ and $\mathbf{x}_{ij} \in \mathbb{R}^{s_i}$ respectively. We also represent each projective transformation $\mathcal{P}_i \colon \mathbb{P}^{r-1} \to \mathbb{P}^{s_i-1}$ by an $s_i \times r$ matrix $\mathtt{P}_i$. The projection relations (2.1) can then be represented as

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j \tag{2.3}$$

for nonzero scalars λ_{ij} called the *projective depths*. The task of projective reconstruction can be restated as recovering the HD points $\mathbf{X}_j$, the projection matrices $\mathtt{P}_i$ and the projective depths λ_{ij}, up to a projective ambiguity, from the image points $\mathbf{x}_{ij}$ (see Sect. (2.5) for a formal definition of projective ambiguity).

Here, the setup $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ is usually referred to as *the true configuration* or *the ground truth*. We sometimes use a second setup of projection matrices and points $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$. This new setup, denoted by *hatted* quantities, in most occasions is referred to as the *estimated* configuration, meaning that it is an estimation of the true setup, usually achieved by some algorithm. The object of our main theorems here is to show that if the setup $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ projects into the same set of image points $\mathbf{x}_{ij}$ introduced in (2.3), that is

$$\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathtt{P}}_i\hat{\mathbf{X}}_j, \tag{2.4}$$

then $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ and $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ are projectively equivalent.

The reader must keep in mind that, here, the projection matrices $\mathtt{P}_i, \hat{\mathtt{P}}_i$, HD points $\mathbf{X}_j, \hat{\mathbf{X}}_j$ and image points $\mathbf{x}_{ij}$ are treated as members of a real vector space, even though they might *represent* quantities in a projective space. The equality sign "=" here is strict and never implies equality up to scale.

2.2 Projective Reconstruction Algorithms

2.2.1 Tensor-Based Algorithms

Perhaps the most widely used example of multi-view tensors is the *fundamental matrix* [Faugeras, 1992; Hartley et al., 1992; Hartley and Zisserman, 2004] used in epipolar (two-view projective) geometry. Consider the classic case of 3D to 2D projections with two views. If each scene point $\mathbf{X}_j \in \mathbb{R}^4$ is viewed by two cameras with camera matrices $\mathtt{P}_1, \mathtt{P}_2 \in \mathbb{R}^{3\times 4}$ as image points $\mathbf{x}_{1j}, \mathbf{x}_{2j} \in \mathbb{R}^3$, then we have

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j \tag{2.5}$$

for $i = 1,2$ and nonzero scalars λ_{ij}. One can show that the above induces a bilinear relation between the corresponding image points $\mathbf{x}_{1j}$ and $\mathbf{x}_{2j}$:

$$\mathbf{x}_{2j}^T \mathtt{F}\, \mathbf{x}_{1j} = 0, \tag{2.6}$$

such that the 3×3 matrix $\mathtt{F}$, known as the *fundamental matrix* only depends on the camera matrices $\mathtt{P}_1$ and $\mathtt{P}_2$. The relation (2.6) defines a linear relation on the elements of $\mathtt{F}$. It can be shown that having images $\mathbf{x}_{ij}$ of sufficient number of scene points $\mathbf{X}_j$ in general location, the relations (2.6) determine the fundamental matrix $\mathtt{F}$ uniquely up to scale [Hartley and Zisserman, 2004].

Given the fundamental matrix, the projection matrices $\mathtt{P}_1$ and $\mathtt{P}_2$ can be obtained up to a projective ambiguity. Having the camera matrices $\mathtt{P}_1$ and $\mathtt{P}_2$, the scene points $\mathbf{X}_j$ can be determined, up to scale, by triangulation.

The tensor-based projective reconstruction involving more than two views, or dealing with projections in other dimensions, more or less follows a similar procedure. A tensor is made from the point correspondences between a subset of views, camera matrices are extracted from the tensor and the HD points are constructed by triangulation.

For 3D to 2D projections, only two other types of tensors exist, namely the trifocal tensor and quadrifocal tensor, representing multilinear relations between triples and quadruples of image point correspondences. For three views indexed by 1, 2 and 3, the following relation holds for each triple of point correspondences $\mathbf{x}_{1j}, \mathbf{x}_{2j}, \mathbf{x}_{3j}$

$$\mathcal{T}(\mathbf{x}_{1j}, \mathbf{l}_{2j}, \mathbf{l}_{3j}) = 0 \tag{2.7}$$

where $\mathbf{l}_{2j}$ and $\mathbf{l}_{3j}$ represent any projective lines passing through $\mathbf{x}_{2j}$ and $\mathbf{x}_{3j}$ respectively, and $\mathcal{T}$ is a trilinear mapping known as the trifocal tensor. One can write the above in tensor notation

$$x_{1j,p}\; l_{2j}^q\; l_{3j}^r\; \mathcal{T}_{qr}^p = 0 \tag{2.8}$$

where $x_{1j,p}$ represents the p-th entry of $\mathbf{x}_{1j}$, l_{2j}^q and l_{3j}^r respectively represent the q-th and r-th entries of $\mathbf{l}_{2j}$ and $\mathbf{l}_{3j}$, and $\mathcal{T}_{qr}^p$ represents the pqr-th element of the trifocal tensor $\mathcal{T}$.

Notice that unlike the case of fundamental matrix, the trifocal tensor is not directly defined as a relation on the entries of points, but rather as a relation among points and lines[1]. As more than one line can pass through each point, for each triple of point correspondences one can have more than one relation in the form of 2.8. Again, each relation (2.8) gives a linear equation on the elements of the tensor. With sufficient point correspondences the tensor can be determined up to a scaling factor. In the same way, the quadrifocal tensor defines a quadrilinear relation among quadruples of views. There are no higher order multilinear relations between correspondences of views.

The tensor methods can be used for projections in other dimensions. The comprehensive work of Hartley and Schaffalitzky [2004] gives a general theory for tensor-based projective reconstruction in arbitrary dimensions. They show that multilinear relations exist for point, line or subspace correspondences among subsets of views, described by the so-called Grassmann tensor. The Grassmann tensor can be obtained linearly using the multilinear relations between the Grassmann coordinates of subspaces passing through the corresponding points in different views. Hartley and Schaffalitzky [2004] give a proof for the uniqueness of the reconstruction of the projective matrices, up to a projective ambiguity, given the Grassmann tensor. Using the procedure explained in their constructive proof, one can reconstruct the projective matrices from the Grassmann tensor.

2.2.2 Bundle Adjustment

In bundle adjustment given the image points $\mathbf{x}_{ij}$, one finds an estimate $(\{\hat{\mathrm{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ of projection matrices P_i and HD points $\mathbf{X}_j$ by minimizing the following target function

$$\sum_{i,j} \mathcal{D}(\mathbf{x}_{ij}, \hat{\mathrm{P}}_i \hat{\mathbf{X}}_j) \tag{2.9}$$

where $\mathcal{D}$ is a distance function. The question is what is a proper choice for $\mathcal{D}$. Considering the relation $\lambda_{ij}\mathbf{x}_{ij} = \mathrm{P}_i\mathbf{X}_j$, one might choose $\mathcal{D}$ as $\mathcal{D}(\mathbf{x}, \mathbf{y}) = \min_{\hat{\lambda}} \|\mathbf{x} - \mathbf{y}/\hat{\lambda}\|$. However, a proper choice of $\mathcal{D}$ is problem dependent. One should consider the physical phenomenon behind the projection model and the nature of the noise process. For example, for the common 3D to 2D perspective projections with a Gaussian noise on the 2D images, the optimal choice of $\mathcal{D}$ in the sense of Maximum Likelihood is

$$\sum_{i,j} \mathcal{D}(\mathbf{x}, \mathbf{y}) = (x_1/x_3 - y_1/y_3)^2 + (x_2/x_3 - y_2/y_3)^2, \tag{2.10}$$

defined over the pair of vectors with a nonzero last entry. Bundle adjustment is usually used as a post processing stage for fine tuning given an initial solution obtained from other reconstruction algorithms.

Besides targeting the Maximum Likelihood cost function, bundle adjustment has

[1] It is however possible to write tensor relations directly on the entries of points, with more than one relation for each point correspondence.

the advantage of handling missing data. One issue with bundle adjustment is that it can fall in local minima, and therefore, it requires good initialization. Another issue is that the associated optimization problem gets very large when large numbers of cameras and points are involved. Several solutions have been proposed to address the scalability problem. We refer the reader to [Hartley and Zisserman, 2004, sections 18, A6] and also [Agarwal et al., 2010] for further information.

2.2.3 Projective Factorization

Consider the projection equation

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j \tag{2.11}$$

for m projection matrices $\mathtt{P}_i \in \mathbb{R}^{s_i\times r}$ and n points $\mathbf{X}_j \in \mathbb{R}^r$. The projective depths $\lambda_{ij} \in \mathbb{R}^{s_i}$, $i = 1,\dots,m$, $j = 1,\dots,n$, can be arranged as an $m\times n$ array to form the *depth matrix* $\Lambda = [\lambda_{ij}]$. Similarly, the image data $\{\mathbf{x}_{ij}\}$ can be arranged as a $(\sum_i s_i)\times n$ matrix $[\mathbf{x}_{ij}]$ called here the *data matrix*. In this way, the above equation can be written in the matrix form

$$\Lambda \odot [\mathbf{x}_{ij}] = \mathtt{P}\,\mathtt{X}, \tag{2.12}$$

where $\mathtt{P} = \mathrm{stack}(\mathtt{P}_1,\mathtt{P}_2,\cdots,\mathtt{P}_m)$ is the vertical concatenation of the camera matrices, $\mathtt{X} = [\mathbf{X}_1\mathbf{X}_2\cdots\mathbf{X}_n]$ and $\Lambda \odot [\mathbf{x}_{ij}] = [\lambda_{ij}\mathbf{x}_{ij}]$, that is the operator $\odot$ multiplies each element λ_{ij} of Λ by the corresponding $s_i\times 1$ block $\mathbf{x}_{ij}$ of the matrix $[\mathbf{x}_{ij}]$. From (2.12) it is obvious that having the true depth matrix Λ, the weighted data matrix $\Lambda \odot [\mathbf{x}_{ij}] = [\lambda_{ij}\mathbf{x}_{ij}]$ can be factored as the product of a $(\sum_i s_i)\times r$ matrix $\mathtt{P}$ by an $r\times n$ matrix $\mathtt{X}$. Equivalently, the matrix $\Lambda \odot [\mathbf{x}_{ij}]$ has rank r or less. This is where the underlying idea of factorization-based algorithms comes from. These algorithms try to find an estimation $\hat{\Lambda}$ of the depth matrix for which the matrix $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ has rank r or less, and thus, can be factored as the product of $(\sum_i s_i)\times r$ and $r\times n$ matrices $\hat{\mathtt{P}}$ and $\hat{\mathtt{X}}$:

$$\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathtt{P}}\,\hat{\mathtt{X}}. \tag{2.13}$$

One hopes that by solving the above problem, dividing $\hat{\mathtt{P}}$ into blocks $\hat{\mathtt{P}}_i \in \mathbb{R}^{s_i\times r}$ as $\hat{\mathtt{P}} = \mathrm{stack}(\hat{\mathtt{P}}_1,\hat{\mathtt{P}}_2,\cdots,\hat{\mathtt{P}}_m)$ and letting $\hat{\mathbf{X}}_j$ be the j-th column of $\hat{\mathtt{X}}$, the camera-point configuration $(\{\hat{\mathtt{P}}_i\},\{\hat{\mathbf{X}}_j\})$ is equal to the true configuration $(\{\mathtt{P}_i\},\{\mathbf{X}_j\})$ up to a projective ambiguity. However, it is obvious that given the data matrix $[\mathbf{x}_{ij}]$ not every solution to (2.13) gives a true reconstruction. A simple reason is the existence of trivial solutions, such as $\hat{\Lambda} = 0$, $\hat{\mathtt{P}} = 0$, $\hat{\mathtt{X}} = 0$, or when $\hat{\Lambda}$ has all but r nonzero columns (see [Oliensis and Hartley, 2007] for $r = 4$). In the latter case it is obvious that $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ can be factored as (2.13) as it has a rank of at most r. This is why we see that in almost all projective factorization algorithms the depth matrix $\hat{\Lambda}$ is somehow restricted to some constraint space. The constraints are used with the hope of preventing the algorithm from ending up in wrong solutions, for which (2.13) is satisfied, but $(\{\hat{\mathtt{P}}_i\},\{\hat{\mathbf{X}}_j\})$ is not projectively equivalent to $(\{\mathtt{P}_i\},\{\mathbf{X}_j\})$. Most of the

constraints used in the literature can prevent at least the trivial examples of wrong solutions where the depth matrix has zero columns or zero rows. However, preventing all types of wrong solutions requires more investigation. In Chapter 3, we will show that for 3D to 2D projections, besides the case of zero columns or zero rows in the depth matrix, there exists a third class of wrong solutions when the depth matrix has a cross-shaped structure. The concept of a cross-shaped matrix was described in Fig. 1.1 of the Introduction chapter. We refer the reader to Fig. 3.3 in Sect. 3.2.5 for a simple example demonstrating how a cross-shaped solution can happen. The core contribution of Chapter 3 is showing that wrong solutions to (2.13) are confined to these three cases, namely where the estimated depth matrix $\hat{\Lambda}$ has zero rows, has zero columns, or is cross-shaped. To give the reader a better understanding, we state the main theorem of Chapter 3 here

Theorem 2.1. *Consider a set of $m \geq 2$ generic camera matrices $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m \in \mathbb{R}^{3\times 4}$ and $n \geq 8$ points $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^4$ in general position, projecting into a set of image points $\{\mathbf{x}_{ij}\}$ according to $\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j/\lambda_{ij}$ for nonzero projective depths λ_{ij}. Now, for any other configuration of m camera matrices $\{\hat{\mathtt{P}}_i\}$, n points $\{\hat{\mathbf{X}}_j\}$ and mn depths $\{\hat{\lambda}_{ij}\}$ related to the same image data $\{\mathbf{x}_{ij}\}$ by*

$$\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathtt{P}}_i\hat{\mathbf{X}}_j, \tag{2.14}$$

if the depth matrix $\hat{\Lambda} = [\hat{\lambda}_{ij}]$ satisfies the following

(D1) $\hat{\Lambda}$ has no zero columns,

(D2) $\hat{\Lambda}$ has no zero rows, and

(D3) $\hat{\Lambda}$ is not cross-shaped,

then the camera-point configuration $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ is projectively equivalent to $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$.

The above can help us with the design of proper depth constraints for the factorization-based algorithms dealing with 3D to 2D projections. This will be discussed in detail in Sect. 2.3. Moving from $\mathbb{P}^3 \rightarrow \mathbb{P}^2$ projections to the more general case of arbitrary dimensional projections, the wrong solutions can be much more complex, as we will show in Chapter 4.

Here, we review different types of projective factorization algorithms proposed in the literature classified by the constraints they use. All these algorithms are suggested for 3D to 2D projections ($r = 4$). Therefore, our discussions for the rest of the section is in the context of 3D to 2D projections.

Sturm-Triggs Factorization The link between projective depth estimation and projective reconstruction of cameras and points was noted by Sturm and Triggs [1996], where it is shown that given the true projective depths, camera matrices and points can be found from the factorization of the data matrix weighted by the depths. However, to estimate the projective depths Sturm and Triggs make use of fundamental

matrices estimated from pairwise image correspondences. Several papers have proposed that the Sturm-Triggs method can be extended to iteratively estimate the depth matrix $\hat{\Lambda}$ and camera-point configuration $\hat{P}$ and $\hat{X}$ [Triggs, 1996; Ueshiba and Tomita, 1998; Heyden et al., 1999; Mahamud et al., 2001; Hartley and Zisserman, 2004]. It has been noted that without constraining or normalizing the depths, such algorithms can converge to false solutions. Especially, Oliensis and Hartley [2007] show that the basic iterative generalization of the Sturm-Triggs factorization algorithm can converge to trivial false solutions, and that in the presence of the slightest amount of noise, it generally does not converge to a correct solution.

Unit Row Norm Constraint Heyden et al. [1999] estimate the camera-point configuration and the projective depths alternatingly, under the constraint that every row of the depth matrix has unit l^2-norm. They also suggest a normalization step which scales each column of the depth matrix to make the first row of the matrix have all unit elements. However, they do not use this normalization step in their experiments, reporting better convergence properties in its absence. It is clear that by just requiring rows to have unit norm, we allow zero columns in the depth matrix as well as cross-shaped configurations. If all rows except the first are required to have unit norm, and the first row is constrained to have all unit elements, then having zero columns is not possible, but still a cross-shaped depth matrix is allowed. We refer the reader to Sect. 6.2 for experiments on this constraint.

Unit Column Norm Constraint Mahamud et al. [2001] propose an algorithms which is in some ways similar to that of Heyden et al. [1999]. Again, the depths and camera-point configuration are alternatingly estimated, but under the constraint that each column of the weighted data matrix has a unit l^2-norm. The convergence to a local minimum is proved, but no theoretical guarantee is given for not converging to a wrong solution. In fact, the above constraint can allow zero rows in the depth matrix in addition to cross-shaped depth matrices.

Fixed Row and Column Norms Triggs [1996] suggests that the process of estimating depths and camera-point structure in the Sturm-Triggs algorithm can be done iteratively in an alternating fashion. He also suggests a depth balancing stage after the depth estimation phase, in which it is sought to rescale rows and columns of the depth matrix such that all rows have the same Euclidean length and similarly all columns have a common length. The same balancing scheme has been suggested by Hartley and Zisserman [2004]. The normalization step is in the form of rescaling rows to have similar norm and then doing the same to columns. At each iteration, this can either be done once each, or in a repeated iterative fashion. If an l^p-norm is used for this procedure, alternatingly balancing rows and columns is the same as applying Sinkhorn's algorithm [Sinkhorn, 1964, 1967] to a matrix whose elements are $|\hat{\lambda}_{ij}|^p$ and thereby forcing all rows of the depth matrix to eventually have the same norm, and similarly all columns to have the same norm. In Sect. 3.3 we will show

that forcing the matrix to have equal nonzero column norms and equal nonzero row norms will prevent all types of false solutions to the factorization-based algorithm for 3D to 2D projections. However, the direct implementation of this constraint is difficult. Implementing it as a balancing stage after every iteration can prevent descent steps in the algorithm. Oliensis and Hartley [2007] report that the normalization step can lead to bad convergence properties.

CIESTA Oliensis and Hartley [2007] prove that if the basic iterative factorization is done without putting any constraint on the depth matrix (except possibly retaining a global scale), it can converge to trivial false solutions. More interestingly, they show that in the presence of noise it always converges to a wrong solution. They also argue that many variants of the algorithm, including [Mahamud et al., 2001] and [Hartley and Zisserman, 2004] either are likely to converge to false solutions or can exhibit undesirable convergence behavior. They propose a new algorithm, called CIESTA, which minimizes a regularized target function. Although some convergence properties have been proved for CIESTA, the solution is biased as it favors projective depths that are close to 1. For this choice, even when there is no noise present, the correct solution does not generally coincide with the global minimum of the CIESTA target function. Here, we do not deal with such approaches.

Fixing Elements of a Row and a Column Ueshiba and Tomita [1998] suggest estimating the projective depths through a conjugate gradient optimization process seeking to make the final singular values of the weighted image data matrix small, thus making it close to a rank-four matrix. To avoid having multiple solutions due to the ambiguity associated with the projective depths, the algorithm constrains the depth matrix to have all elements of the r-th row and the c-th column equal to one for some choice of r and c, that is $\hat{\lambda}_{ij} = 1$ when $i = r$ or $j = c$. This constraint can lead to cross-shaped configurations, although there is only one possible location for the centre of cross, namely (r, c).

2.2.4 Rank Minimization

The rank minimization approach is actually a variant of the factorization-based approach. In this approach instead of finding $\hat{\Lambda}$ such that $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ has rank r or less, one tries to find $\hat{\Lambda}$ so as to minimize the rank of $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$. Again, the rank minimization must be done subject to some constraints to avoid false solutions like $\hat{\Lambda} = 0$. Here, we review some of such methods, again classified by the constraint employed. Like the previous section, statements made here are for the case of 3D to 2D projections.

Transportation Polytope Constraint Dai et al. [2010, 2013] note that for any solution to the factorization-based problems, the weighted data matrix $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ is restricted to have rank four or less. They formulated the problem as a rank minimization approach, where one seeks to minimize the rank of $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ subject to some constraints. As a constraint, they require the depth matrix $\hat{\Lambda}$ to have fixed row and column sums.

In addition, this approach also enforces the constraint $\hat{\lambda}_{ij} \geq 0$, that is the projective depths are all nonnegative[2]. In [Angst et al., 2011] it has been noted that the corresponding constraint space is known as the Transportation Polytope.

Dai et al. [2010, 2013] solve the rank minimization problem by using the trace norm as a convex surrogate for the rank function. The relaxed optimization problem can be recast as a semi-definite program. One drawback of this approach is the use of inequality constraints, preventing it from taking advantage of the fast rank minimization techniques for large scale data such as [Lin et al., 2010; Yang and Yuan, 2013]. The same idea is used in [Angst et al., 2011], however, a generalized trace norm target function is exploited to approximate the rank. While Angst et al. [2011] mention the transportation polytope constraint space, for implementation they just fix the global scale of the depth matrix. As this constraint is prone to giving degenerate trivial solutions, the authors add inequality constraints whenever necessary. In Sect. 3.3 we shall show that for 3D to 2D projections the transportation polytope constraint avoids false solutions to the factorization methods if the marginal values to which rows and columns must sum up are chosen properly.

Fixed Row and Column Sums As noted before, the inequality constraint used in [Dai et al., 2010, 2013] can prevent the design of fast algorithms. This might be the reason why, when it comes to introducing scalable algorithms in [Dai et al., 2013], the inequality constraint has been neglected. We will show that neglecting the inequality constraint and just constraining rows and columns of $\hat{\Lambda}$ to have specified sums always allows for cross-shaped structures and thus for false solutions. However, as discussed in Sect. 3.3, it is difficult to converge to such a structure starting from a sensible initial solution.

2.3 Motivation

2.3.1 Issues with the tensor-based approaches and theorems

In Sect. 2.2.1 we had a quick review of the tensor-based approaches. As briefly discussed in the Introduction, tensor-based approaches have some limitations. One issue is that a multi-view tensor can be defined only for up to a limited number of views. For example for 3D to 2D projections, only up to four views can be analysed with a tensor [Hartley and Zisserman, 2004]. In general, for multiple projections from $\mathbb{P}^{r-1}$, at most r views can be involved in multilinear relations corresponding to a single tensor [Hartley and Schaffalitzky, 2004]. This can prevents us from having more exact estimations by considering the projected data from all views when having a large number of views. This can be a problem especially in the presence of noise. There are other issues as well, such as imposing certain *internal* constraints on the tensors. This is because the actual dimensionality or degrees of freedom of a

[2]Actually, in [Dai et al., 2010, 2013] the constraint is given as imposing strictly positive depths: $\hat{\lambda}_{ij} > 0$, giving a non-closed constraint space. However, what can be implemented in practice using semi-definite programming or other iterative methods is non-strict inequalities like $\hat{\lambda}_{ij} \geq 0$ or $\hat{\lambda}_{ij} \geq \delta$.

multi-view tensor is less than its number of elements minus one. The "minus one" here is due to the fact the tensor is determined up to a scaling factor. For example, it is known that the 3×3 fundamental matrix (bifocal tensor) has rank 2, and thus, a zero determinant. This imposes a polynomial constraint on its elements, which is the only required constraint. As a 3×3 matrix defined up to scale has $9-1=8$ degrees of freedom, the fundamental matrix, has 7 degrees of freedom. The number of internal constraints grows rapidly with the dimensionality of the tensor. For example, it is known that the 3×3×3 trifocal tensor has only 18 degrees of freedom. This gives $3^3-1-18=8$ internal constraints. The quadrifocal tensor has $3^4=81$ elements. However, it only has 29 degrees of freedom, giving 51 internal constraints (we refer the reader to [Hartley and Zisserman, 2004, Sect. 17.5] and [Heinrich and Snyder, 2011] for more details). As the tensors are usually estimated linearly, imposing such constraints can be an issue when data is noisy. Because of such issues, other projective reconstruction algorithms are used either with conjunction with the tensor-based methods or independently. These algorithms usually either fall in the category of *Bundle Adjustment* [Triggs et al., 2000] and or *Projective Factorization* [Sturm and Triggs, 1996; Triggs, 1996; Mahamud et al., 2001; Oliensis and Hartley, 2007]. As we saw in Sect. 2.2, these methods try to solve the projection equations

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j \tag{2.15}$$

directly for projection matrices $\mathtt{P}_i$, points $\mathbf{X}_i$ and projective depths λ_{ij}. Analysing such methods requires a theory which derives the projective reconstruction from the projection equations (2.15), rather than from the multi-view tensor. The object of this work is to provide a theoretical basis for the analysis of such reconstruction algorithms. To see why such a theorem is needed, let us have a look at the present theorems of projective reconstruction, both in the case of 3D to 2D projections and arbitrary dimensional projections. We make minor changes to the statements of the theorems to make them compatible with the conventions used here.

First, we consider the Projective Reconstruction Theorem stated in [Hartley and Zisserman, 2004, Sect. 10.3] for 3D to 2D projections in two views. One can extend the theorem to arbitrary number of views, for example, by considering different pairs of views and stitching the reconstructions together.

Theorem 2.2 (Projective Reconstruction Theorem [Hartley and Zisserman, 2004]). *Suppose that* $\mathbf{x}_{1j} \leftrightarrow \mathbf{x}_{2j}$ *is a set of correspondences between points in two images and that the fundamental matrix* $\mathtt{F}$ *is uniquely determined by the condition* $\mathbf{x}_{2j}^T\mathtt{F}\mathbf{x}_{1j}=0$ *for all* j. *Let* $(\mathtt{P}_1,\mathtt{P}_2,\{\mathbf{X}_j\})$ *and* $(\hat{\mathtt{P}}_1,\hat{\mathtt{P}}_2,\{\hat{\mathbf{X}}_j\})$ *be two reconstructions of the correspondences* $\mathbf{x}_{1j} \leftrightarrow \mathbf{x}_{2j}$, *which means*

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j$$
$$\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathtt{P}}_i\hat{\mathbf{X}}_j$$

for $i=1,2$ *and* $j=1,2,\ldots,n$ *with nonzero scalars* λ_{ij} *and* $\hat{\lambda}_{ij}$. *Then there exists nonzero*

scalars τ_1, τ_2 *and* $\nu_1, \nu_2, \ldots, \nu_n$ *and a non-singular matrix* $\mathtt{H}$ *such that*

$$\hat{\mathtt{P}}_1 = \tau_1 \mathtt{P}_1 \mathtt{H} \tag{2.16}$$

$$\hat{\mathtt{P}}_2 = \tau_2 \mathtt{P}_2 \mathtt{H} \tag{2.17}$$

$$\hat{\mathbf{X}}_j = \nu_j \mathtt{H}^{-1} \mathbf{X}_j \tag{2.18}$$

except for those j *such that* $\mathtt{F}\,\mathbf{x}_{1j} = \mathtt{F}^T\mathbf{x}_{2j} = \mathbf{0}$.

Notice that $\mathtt{F}\mathbf{x}_{1j} = \mathtt{F}^T\mathbf{x}_{2j} = \mathbf{0}$ occurs when the $\mathbf{x}_{1j}$ and $\mathtt{F}^T\mathbf{x}_{2j}$ are images of a 3D point lying on the projective line connecting the centres of the two cameras. This is known as a triangulation ambiguity.

The next theorem given by [Hartley and Schaffalitzky, 2004] deals with the case of projections in arbitrary dimensions. The basic finding is that the camera matrices can be obtained up to projectivity from the corresponding multi-view (Grassmann) tensor.

Theorem 2.3 (Hartley and Schaffalitzky [2004]). *Consider a set of* m generic *projection matrices* $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m$, *with* $\mathtt{P}_i \in \mathbb{R}^{s_i \times r}$, *such that* $m \leq r \leq \sum_i s_i - m$, *and an* m*-tuple* $(\alpha_1, \alpha_2, \ldots, \alpha_m)$ *of integers* α_i *such that* $1 \leq \alpha_i \leq m-1$ *for all* i *and* $\sum_{i=1}^m \alpha_i = r$. *Then if at least for one* i *we have* $s_i \geq 3$, *the matrices* $\mathtt{P}_i$ *are determined up to a projective ambiguity from the set of minors of the matrix* $\mathtt{P} = \mathrm{stack}(\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m)$ *chosen with* α_i *rows from each* $\mathtt{P}_i$ *(that is the elements of the Grassmann tensor). If* $s_i = 2$ *for all* i, *there are two equivalence classes of solutions.*

We see that in these theorems, the main focus is on the uniqueness of the reconstruction given the multi-view tensor. This can be particularly an issue for the case of arbitrary dimensional projections for which the theory has not been developed to the extent it has for 3D to 2D projections. We argue that the current theorems are not sufficient for the analysis of algorithms like bundle adjustment and projective factorization whose aim is to directly solve the set of projective equations

$$\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathtt{P}}_i\hat{\mathbf{X}}_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \tag{2.19}$$

for camera matrices $\hat{\mathtt{P}}_i$, HD points $\hat{\mathbf{X}}_j$ and projective depths $\hat{\lambda}_{ij}$. The obstacles for getting from the above theorems to the point we can analyse such algorithms are as follows:

1. Proving that the multi-view tensor is uniquely determined from the image data $\mathbf{x}_{ij}$, in a generic configuration with sufficiently many points.
2. Proving that there is no solution $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\}, \{\hat{\lambda}_{ij}\})$ to (2.19) for which the multi-view tensor corresponding to $\{\hat{\mathtt{P}}_i\}$ is zero.
3. If some of the estimated projective depths $\hat{\lambda}_{ij}$ are not restricted to be nonzero, what types of degenerate solutions to (2.19) can happen.

The third issue above is especially needed for the projective factorization algorithms for which it is inefficient to enforce nonzero constraints on all projective

depths $\hat{\lambda}_{ij}$. This has been considered in detail in the next subsection. After that, in Sect. 2.3.3 we elaborate on all the above three issues for the case of arbitrary dimensional projections.

2.3.2 Projective Factorization Algorithms

In Sect. 2.2.3 we discussed that in the factorization problem one tries to solve

$$\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathrm{P}}\,\hat{\mathrm{X}}. \tag{2.20}$$

where the image points $\mathbf{x}_{ij}$ are obtained through a projection process $\mathbf{x}_{ij} = \frac{1}{\lambda_{ij}}\mathrm{P}_i\mathbf{X}_j$. We also argued that without the use of proper constraints, some solutions to (2.20) are not projectively equivalent to the true camera-point configuration. By reviewing the literature in Sect. 2.2.3, we observed that all of the current methods, either implicitly or explicitly, try to solve the above equation subject to some constraint on $\hat{\Lambda}$. We gave examples of the so-called trivial solutions, such as $\hat{\Lambda} = 0$, $\hat{\mathrm{P}} = 0$, $\hat{\mathrm{X}} = 0$, or when $\hat{\Lambda}$ has all but r zero columns. One can also easily show the existence of false solutions in which one or more rows of $\hat{\Lambda}$ or one or more of its columns are zero. For example, by setting $\hat{\lambda}_{ij} = \lambda_{ij}$ for $i = 2,3,\ldots,m$ and all j, $\hat{\lambda}_{1j} = 0$ for all j, $\hat{\mathrm{P}}_i = \mathrm{P}_i$ for $i = 2,3,\ldots,m$, $\hat{\mathrm{P}}_1 = 0$ and $\hat{\mathbf{X}}_j = \mathbf{X}_j$ for all j, we have a wrong solution satisfying (2.20) for which the first row of $\hat{\Lambda} = [\hat{\lambda}_{ij}]$ is zero.

It is not obvious, however, (and we shall prove it false) if possible false solutions to (2.20) are restricted to these trivial cases. Therefore, factorization-based algorithms lack a proper theoretical basis for finding possible false solutions allowed by given constraints or to determine what constraints on the depth matrix make every solution to (2.20) projectively equivalent to the ground truth.

For 3D to 2D projections, the main theoretical basis for the analysis of projective reconstruction are theorems like the Projective Reconstruction Theorem [Hartley and Zisserman, 2004] discussed briefly in Sect. 2.3.1. It says that, under certain generic conditions, all configurations of camera matrices and 3D points yielding a common set of 2D image points are equal up to a projective ambiguity. This theorem is derived from a geometric perspective and therefore presumes assumptions like the estimated camera matrices $\hat{\mathrm{P}}_i$ having full row rank and all the estimated projective depths $\hat{\lambda}_{ij}$ being nonzero. While these are useful enough for the so-called tensor-based reconstruction approaches, they are not a good fit for the analysis of algebraic algorithms, especially projective factorization. Obviously, these geometric assumptions can be reasonably assumed for the true set of depths $\{\lambda_{ij}\}$ and the true camera-point configuration $(\{\mathrm{P}_i\},\{\mathbf{X}_j\})$. However, for most of the factorization-based algorithms, at least in the case of large-scale problems, it is hard to impose these constraints on the estimated depths $\{\hat{\lambda}_{ij}\}$ and camera-point configuration $(\{\hat{\mathrm{P}}_i\},\{\hat{\mathbf{X}}_j\})$ a priori, during the estimation process.

For 3D to 2D projections, one can show that the basic assumption for the proof of the classic Projective Reconstruction Theorem [Hartley and Zisserman, 2004] is that the estimated depths $\hat{\lambda}_{ij}$ are all nonzero. Other geometric assumptions like full-row-

rank estimated camera matrices $\hat{\mathbf{P}}_i$ follow from this assumption under reasonable conditions. Therefore, one might like to enforce $\hat{\lambda}_{ij} \neq 0$ as a constraint for any algorithm for solving (2.20), and make use of this theorem to show that the algorithm avoids false solutions. However, this type of constraint space cannot be easily implemented in most of the iterative algorithms. Since this constraint space is not closed, it is possible for the procedure to converge to a solution outside the constraint space, even if all iterations lie inside the constraint space. This means that some of the projective depths can converge to zero, resulting in a degenerate solution. Making use of the scale ambiguity of the projective depths, the constraint space can be made closed by using $|\hat{\lambda}_{ij}| \geq \delta$ for some positive number δ rather than $\hat{\lambda}_{ij} \neq 0$. However, this non-connected constraint space again cannot be easily handled by many of the iteration based algorithms. Actually, in practice, when there is no missing data, it is usually the case that all true depths λ_{ij} are positive, as all the 3D points are in front of the cameras. In this case, we can have a convex constraint space by forcing all-positive depths, that is $\hat{\lambda}_{ij} > 0$. Obviously, due to the scale ambiguity, the constraint space can be made closed by using $\hat{\lambda}_{ij} \geq \delta$ instead, for some $\delta > 0$. This gives a set of linear inequalities.

One problem with the inequality constraints is that they are hard to implement for fast and efficient factorization-based algorithms, especially for large-scale problems. Thus, we seek even simpler constraints making the optimization-based techniques more efficient and easier to solve. For example, linear equality constraints, which are easier to handle and for which usually much faster algorithms exist compared to inequality constraints. This can be seen, for example, in state-of-the-art algorithms designed for the convex relaxation of large scale rank minimization problems which work with linear equality constraints [Lin et al., 2010; Yang and Yuan, 2013]. We observed the use of linear equality constraints in papers like [Ueshiba and Tomita, 1998] (by fixing special elements of the depth matrix $\hat{\Lambda}$) and also [Dai et al., 2010, 2013] (by fixing the row and column sums of $\hat{\Lambda}$) when it comes to large scale problems. We also observed other examples of constraints like requiring rows of $\hat{\Lambda}$ [Heyden et al., 1999], or columns of $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ [Mahamud et al., 2001] to have a unit l^2-norm, which allowed for efficient factorization-based algorithms. However, as these constraints, per se, are unable to guarantee all depths to be nonzero or strictly positive, we cannot take advantage of the classic theorem of projective reconstruction to analyse their effectiveness. This shows the need to finding weaker conditions under which projective reconstruction succeeds. The new conditions must allow the verification of the constraints that fit the factorization-based algorithms. We will introduce such a theorem for 3D to 2D projections in Sect. 4.2. The case of arbitrary dimensional projections is discussed in the next subsection.

2.3.3 Arbitrary Dimensional Projections

A major application of projective reconstruction in higher dimensions is the analysis of dynamic scene problems such as motion segmentation [Wolf and Shashua, 2002] and non-rigid deformation recovery [Xiao and Kanade, 2005; Vidal and Abretske,

2006; Hartley and Vidal, 2008]. These problems can be modeled as projections from higher-dimensional projective spaces to $\mathbb{P}^2$. Such applications illustrate the need for developing the theory and algorithms of projective reconstruction in higher dimensions. The first comprehensive study of projective reconstruction for general projections in arbitrary dimensions is due to Hartley and Schaffalitzky [2004]. They introduce the Grassmann tensor as a generalization of the concepts of bifocal, trifocal and quadrifocal tensor used for 3D to 2D projections, and also special cases of multi-view tensors introduced for projections from other dimensions. As discussed in Sect. 2.3.1, their main result is a theorem asserting that the projection matrices can be uniquely determined up to projectivity from the corresponding Grassmann tensor.

As discussed earlier, the tensor methods suffer from some issues such as limited number of views handled by each tensor and the internal constraints of the tensors. Especially, for higher-dimensional projective spaces, the number of internal constraints of the multi-view tensors becomes very large. Such problems encourage the use of other techniques such as bundle adjustment and projective factorization, in which the projection equations

$$\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathrm{P}}_i\hat{\mathbf{X}}_j, \quad i = 1,\ldots,m, \quad j = 1,\ldots,n \tag{2.21}$$

are directly solved for projection matrices $\hat{\mathrm{P}}_i$, HD points $\hat{\mathbf{X}}_j$ and projective depths $\hat{\lambda}_{ij}$. To analyse such methods one needs to further develop the current theory of projective reconstruction. In this thesis, we present an extended theory which deduces projective reconstruction from the set of equations (2.21), rather than the multi-view tensor.

A number of obstacles must be tackled to give a theory for analysing such algorithms. First, we need to prove that sufficiently many image points $\mathbf{x}_{ij}$ obtained from a generic projection-point configuration uniquely determine the Grassmann tensor. While this fact is known for 3D to 2D projections, no proof has yet been given for arbitrary dimensional projections. We will give a proof in Sect. 4.2.1. Notice that this result is important even for tensor-based methods.

The second problem is to show that if a second configuration of projection matrices and points project into the same image points $\mathbf{x}_{ij}$ (with nonzero depths $\hat{\lambda}_{ij}$) it is projectively equivalent to the true configuration form which the image points are created. Besides projective factorization, this result is also important for the analysis of bundle adjustment. To prove this, in addition to the uniqueness of the Grassmann tensor up to scale, one has to show that the Grassmann tensor corresponding to the second set of camera matrices is nonzero. This will be proved in Sect. 4.3.

Finally, to be able to analyse the projective factorization problem

$$\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathrm{P}}\,\hat{\mathrm{X}}, \tag{2.22}$$

one has to understand the nature of the wrong solutions which can happen and classify them. This helps to properly constrain the depth matrix in projective fac-

torization algorithms, and also enables us to verify the final solution given by such algorithms. We mentioned that for the case of 3D to 2D projections, except the trivial solutions where $\hat{\Lambda}$ has zero rows or zero columns, the only possible false solution happens when $\hat{\Lambda}$ is cross-shaped. As we will show, the wrong solutions in arbitrary dimensional case can be in general much more complicated. The classification of such degenerate solutions is done in Sect. 4.4.

The rest of this section reviews some of the applications of higher-dimensional projective reconstruction in the literature.

2.3.3.1 Points moving with constant velocity

Wolf and Shashua [2002] consider the following cases in which points moving with a constant velocity are seen by perspective cameras:

2D constant velocity Points moving independently within a 2D plane, each with a constant velocity along a straight line. They show that this problem can be modeled with projections $\mathbb{P}^4 \to \mathbb{P}^2$.

3D constant collinear velocity Each point moves with a constant velocity along a straight line. All line trajectories are parallel. They demonstrate that this can be modeled as projections $\mathbb{P}^4 \to \mathbb{P}^2$.

3D constant coplanar velocity Each point moves with a constant velocity along a straight line. The velocity vectors are coplanar. It is shown that this can be generally modeled as projections $\mathbb{P}^5 \to \mathbb{P}^2$.

3D constant velocity Each point moves with a constant velocity along a straight line. It is shown that, generically, this can be modeled as projections $\mathbb{P}^6 \to \mathbb{P}^2$.

2.3.3.2 Motion Segmentation

Wolf and Shashua [2002] consider a configuration of 3D points consisting of two rigid bodies whose relative motion to each other consists only of pure translation, that is the rotation in two objects is the same. They show that this can be modeled as projections $\mathbb{P}^4 \to \mathbb{P}^2$. This approach can be generalized to the case of more general types of motion and more than two rigid bodies. We will discuss this further in Sect. 5.1.

2.3.3.3 Nonrigid Motion

Hartley and Vidal [2008] consider the problem of perspective nonrigid deformation. They show that nonrigid deformations can be modeled as linear combinations of a number of rigid prototype shapes. They demonstrate that this problem can be modeled as projections from $\mathbb{P}^{3k}$ to $\mathbb{P}^2$, where k is the number of prototype basis

shapes. Using this fact, they gave a solution to the problem of perspective nonrigid motion recovery using a tensor-based approach. We give more details on this in Sect. 5.2.

2.4 Correspondence Free Structure from Motion

Angst and Pollefeys [2013] study a configuration of multiple cameras which are all fixed in their place, or undergo a global rigid motion. Each camera observes a subset of scene points, producing tracks of image points over time. The proposed algorithm recovers the structure and motion of the scene using the image point tracks given by the cameras. However, no knowledge about the point correspondences between different cameras are required. In fact, the cameras may observe non-overlapping portions of the scene. What links the data obtained by different cameras is the fact that they are all *observing a common rigid motion.* The proposed algorithm assumes an *affine camera* model. Particularly, they show that, assuming affine cameras, the image point tracks lie on a 13-dimensional subspace when the scene undergoes a general rigid motion. If the motion is *planar,* it has been shown that the tracks lie on a 5-dimensional subspace. The proposed algorithm involves a rank-13 (or rank-5) factorization of the image data matrix to decouple the motion from the camera-point setup.

This idea can be generalized to projective cameras. One can show that the recovery of the 3D structure and motion involves a projective reconstruction for projections $\mathbb{P}^{12} \to \mathbb{P}^2$ (or $\mathbb{P}^4 \to \mathbb{P}^2$ for planar motion). We will talk more about this in Sect. 5.3.

2.5 Projective Equivalence and the Depth Matrix

As was stated before, for a set of projection matrices $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m$ with $\mathtt{P}_i \in \mathbb{R}^{s_i \times r}$, a set of points $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ in $\mathbb{R}^r$, and a set of image data $\mathbf{x}_{ij} \in \mathbb{R}^{s_i}$ formed according to the projection relation

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j \tag{2.23}$$

with nonzero projective depths $\lambda_{ij} \neq 0$, projective reconstruction (finding $\mathtt{P}_i$-s and $\mathbf{X}_j$-s) given only the image points $\mathbf{x}_{ij}$ is possible only up to a projective ambiguity. This means that the solution is in the form of a projective equivalence class. Here, we formalize the concept of projective equivalence in the context of the formulation used here. Readers can refer to [Hartley and Zisserman, 2004] for more details.

Definition 2.1. *Two sets of projection matrices* $\{\mathtt{P}_i\}$ *and* $\{\hat{\mathtt{P}}_i\}$, *with* $\mathtt{P}_i, \hat{\mathtt{P}}_i \in \mathbf{R}^{s_i \times r}$ *for* $i = 1, 2, \ldots, m$ *are projectively equivalent if there exist* nonzero *scalars* $\tau_1, \tau_2, \ldots, \tau_m$ *and an* $r \times r$ non-singular *matrix* $\mathtt{H}$ *such that*

$$\hat{\mathtt{P}}_i = \tau_i \mathtt{P}_i \mathtt{H}, \quad i = 1, 2, \ldots, m. \tag{2.24}$$

Two sets of points $\{\mathbf{X}_j\}$ *and* $\{\hat{\mathbf{X}}_j\}$ *with* $\mathbf{X}_j, \hat{\mathbf{X}}_j \in \mathbb{R}^r$ *for* $j = 1,2,\ldots,n$, *are projectively equivalent if there exist* nonzero *scalars* $\nu_1, \nu_2, \ldots, \nu_n$ *and a non-singular* $r \times r$ *matrix* $\mathtt{G}$ *such that*

$$\hat{\mathbf{X}}_j = \nu_j \mathtt{G} \mathbf{X}_j, \quad j = 1,2,\ldots,n. \tag{2.25}$$

Two setups $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ *and* $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ *are projectively equivalent if both* (2.24) *and* (2.25) *hold, and furthermore* $\mathtt{G} = \mathtt{H}^{-1}$. *In other words, there exist nonzero scalars* $\tau_1, \tau_2, \ldots, \tau_m$ *and* $\nu_1, \nu_2, \ldots, \nu_n$, *and an invertible matrix* $\mathtt{H}$ *such that*

$$\hat{\mathtt{P}}_i = \tau_i \mathtt{P}_i \mathtt{H}, \qquad i = 1,2,\ldots,m. \tag{2.26}$$

$$\hat{\mathbf{X}}_j = \nu_j \mathtt{H}^{-1} \mathbf{X}_j, \qquad j = 1,2,\ldots,n. \tag{2.27}$$

2.5.1 Equivalence of Points

The following lemma about the projective equivalence of the points will be used later on in the thesis.

Lemma 2.1. *Consider a set of points* $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^r$ *with* $n > r$ *with the following generic properties*

(P1) $\operatorname{span}(\mathbf{X}_1, \ldots, \mathbf{X}_n) = \mathbb{R}^r$, *and*

(P2) the set of points $\{\mathbf{X}_i\}$ *cannot be partitioned into* $p \geq 2$ *nonempty subsets, such that subspaces defined as the span of each subset are independent*[3].

Now, for any set of points $\{\hat{\mathbf{X}}_i\}$ *projectively equivalent to* $\{\mathbf{X}_i\}$, *the matrix* $\mathtt{G}$ *and scalars* ν_j *defined in* (2.25) *are* unique *up to a scale ambiguity of the form* $(\beta\mathtt{G}, \{\nu_j/\beta\})$ *for any nonzero scalar* β.

Notice that (P2) is generic only when $n > r$, as for $n \leq r$ the set of points $\mathbf{X}_1, \ldots, \mathbf{X}_n$ always can be split such that the spans of the partitions form independent linear subspaces. For example, if $\mathbf{X}_j$-s are linearly independent, then the subspaces $\operatorname{span}(\mathbf{X}_1), \operatorname{span}(\mathbf{X}_2), \ldots, \operatorname{span}(\mathbf{X}_n)$ form independent subspaces. This lemma will be used to prove projective equivalence for the whole set of views given projective equivalence for subsets of views.

Proof of Lemma 2.1. Assume there are two sets of nonzero scalars $\{\nu_j\}$ and $\{\nu'_j\}$ and two invertible matrices $\mathtt{G}$ and $\mathtt{G}'$ such that

$$\hat{\mathbf{X}}_j = \nu_j \mathtt{G} \mathbf{X}_j, \tag{2.28}$$

$$\hat{\mathbf{X}}_j = \nu'_j \mathtt{G}' \mathbf{X}_j. \tag{2.29}$$

[3]Subspaces $U_1, \ldots, U_p$ are independent if $\dim(\sum_{j=1}^p U_j) = \sum_{j=1}^p \dim(U_j)$, where $\sum_{j=1}^p U_j = \{\sum_{j=1}^p \mathbf{u}_j \mid \mathbf{u}_j \in U_j\}$.

This gives

$$\mathtt{R}\mathbf{X}_j = \beta_j \mathbf{X}_j, \tag{2.30}$$

where $\mathtt{R} = \mathtt{G}^{-1}\mathtt{G}'$ and $\beta_j = \nu_j/\nu_j'$. Thus, $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ are the eigenvectors of $\mathtt{R}$ with the corresponding eigenvalues $\beta_1, \beta_2, \ldots, \beta_n$. As an $r \times r$ matrix can have at most r eigenvalues, the set of indices $\{1, 2, \ldots, n\}$ can be partitioned into p nonempty subsets $J_1, J_2, \ldots, J_p$ such that for each subset J_k, the corresponding eigenvalues β_j are equal to a common value $\beta^{(k)}$. Moreover, for each k, the subspace $U_k = \text{span}(\{\mathbf{X}_j\}_{j \in J_k})$ is a subset of the corresponding eigenspace of the eigenvalue $\beta^{(k)}$. It is known that the sum of eigenspaces corresponding to different eigenvalues of a matrix is a direct sum. This means that the eigenspaces are independent. As each U_k is a subset of one eigenspace, the subspaces $U_1, U_2, \ldots, U_p$ are also independent. Now, according to the condition (P2), we must have $p = 1$, and therefore, all eigenvalues $\beta^{(1)}, \beta^{(2)}, \ldots, \beta^{(p)}$, and thus $\beta_1, \ldots, \beta_n$ have a common value, name it β. The corresponding eigenspace of β is $\text{span}(\mathbf{X}_1, \ldots, \mathbf{X}_n)$ which is equal to the whole ambient space $\mathbb{R}^r$ according to (P1). This means that $\mathtt{R} = \beta\mathtt{I}$, where $\mathtt{I}$ is the identity matrix. Now, from the definition of $\mathtt{R}$ and β_j $(= \beta)$ in (2.30) we get $\mathtt{G}' = \beta\mathtt{G}$ and $\nu_j' = \nu_j/\beta$ for all j. Notice that β is nonzero, as ν_j and ν_j' are both nonzero and $\beta = \beta_j = \nu_j/\nu_j'$ □

2.5.2 The depth matrix

We will need to know the implications of projective equivalence of $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ on the depth matrices $\Lambda = [\lambda_{ij}]$ and $\hat{\Lambda} = [\hat{\lambda}_{ij}]$. First, we define the concept of *diagonal equivalence* for matrices:

Definition 2.2. *Two $m \times n$ matrices Λ and $\hat{\Lambda}$ are diagonally equivalent if there exist* nonzero *scalars $\tau_1, \tau_2, \ldots, \tau_m$ and $\nu_1, \nu_2, \ldots, \nu_n$ such that*

$$\hat{\Lambda} = \text{diag}(\boldsymbol{\tau})\, \Lambda\, \text{diag}(\boldsymbol{\nu}) \tag{2.31}$$

where $\boldsymbol{\tau} = [\tau_1, \tau_2, \ldots, \tau_m]^T$, $\boldsymbol{\nu} = [\nu_1, \nu_2, \ldots, \nu_n]^T$ and diag$(\cdot)$ *arranges the entries of a vector on the diagonal of a diagonal matrix.*

The concepts of projective equivalence of projections and points and diagonal equivalence of depth matrices are related by the following lemma

Lemma 2.2. *Consider two configurations of m projection matrices and n points $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$, with $\mathtt{P}_i, \hat{\mathtt{P}}_i \in \mathbb{R}^{s_i \times r}$ and $\mathbf{X}_j, \hat{\mathbf{X}}_j \in \mathbb{R}^r$, such that*

(i) $\mathtt{P}_i\mathbf{X}_j \neq \mathbf{0}$ *for all i, j,*

(ii) $\text{span}(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n) = \mathbb{R}^r$, *and*

(iii) $\mathtt{P} = \text{stack}(\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m)$ *has rank r (full column rank).*

Also, consider two $m \times n$ matrices $\Lambda = [\lambda_{ij}]$ and $\hat{\Lambda} = [\hat{\lambda}_{ij}]$. If the relations

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j \tag{2.32}$$

$$\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathtt{P}}_i\hat{\mathbf{X}}_j \tag{2.33}$$

hold for all $i = 1,\ldots,m$ and $j = 1,\ldots,n$, then $(\{\mathtt{P}_i\},\{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\},\{\hat{\mathbf{X}}_j\})$ are projectively equivalent if and only if the matrices Λ and $\hat{\Lambda}$ are diagonally equivalent.

Proof. First, assume that $(\{\mathtt{P}_i\},\{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\},\{\hat{\mathbf{X}}_j\})$ are projectively equivalent. Then, there exist nonzero scalars $\tau_1,\tau_2,\ldots,\tau_m$ and $\nu_1,\nu_2,\ldots,\nu_n$ and an invertible matrix $\mathtt{H}$ such that (2.26) and (2.27) hold. Therefore we have

$$\begin{aligned}\hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j &= \hat{\lambda}_{ij}\lambda_{ij}\mathbf{x}_{ij} = \lambda_{ij}\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j \\ &= \lambda_{ij}\nu_j\tau_i\,\mathtt{P}_i\,\mathtt{H}\mathtt{H}^{-1}\,\mathbf{X}_j = \lambda_{ij}\nu_j\tau_i\,\mathtt{P}_i\,\mathbf{X}_j.\end{aligned}$$

where the first, second and third equations above hold respectively from (2.32), (2.33) and (2.26, 2.27). By condition (i) in the lemma, that is $\mathtt{P}_i\,\mathbf{X}_j \neq \mathbf{0}$, it follows from the above that $\hat{\lambda}_{ij} = \lambda_{ij}\nu_j\tau_i$ for all i and j. This is equivalent to (2.31) and hence Λ and $\hat{\Lambda}$ are diagonally equivalent.

To prove the other direction, assume that Λ and $\hat{\Lambda}$ are diagonally equivalent. Then from (2.31) we have $\hat{\lambda}_{ij} = \lambda_{ij}\nu_j\tau_i$. This along with (2.32) and (2.33) gives

$$\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathbf{x}_{ij} = \lambda_{ij}\nu_j\tau_i\mathbf{x}_{ij} = \tau_i\nu_j\mathtt{P}_i\mathbf{X}_j = (\tau_i\mathtt{P}_i)(\nu_j\mathbf{X}_j) \tag{2.34}$$

for $i = 1,\ldots,m$ and $j = 1,\ldots,n$. Let $\mathtt{Q}_i = \tau_i\mathtt{P}_i$ and $\mathbf{Y}_j = \nu_j\mathbf{X}_j$, so we have $\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \mathtt{Q}_i\mathbf{Y}_j$. Denote by $\mathtt{Q}$ and $\hat{\mathtt{P}}$ the vertical concatenations of $\mathtt{Q}_i$-s and $\hat{\mathtt{P}}_i$-s respectively and denote by $\mathtt{Y}$ and $\hat{\mathbf{X}}$ respectively the horizontal concatenations of $\mathbf{Y}_j$-s and $\hat{\mathbf{X}}_j$-s. From $\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \mathtt{Q}_i\mathbf{Y}_j$ we have

$$\hat{\mathtt{P}}\hat{\mathbf{X}} = \mathtt{Q}\mathtt{Y} \stackrel{\text{def}}{=} \mathtt{A}. \tag{2.35}$$

From conditions (ii) and (iii) in the lemma along with the fact that τ_i and ν_j are nonzero, we can conclude that $\mathtt{Q}$ has full column rank and $\mathtt{Y}$ has full row rank. Therefore, $\mathtt{A} \stackrel{\text{def}}{=} \mathtt{Q}\mathtt{Y}$ has rank r, and hence, the matrices $\hat{\mathtt{P}}$ and $\hat{\mathbf{X}}$ must both have maximal rank r. As $\mathtt{Q}\mathtt{Y}$ and $\hat{\mathtt{P}}\hat{\mathbf{X}}$ are two rank-r factorizations of $\mathtt{A}$, having $\hat{\mathtt{P}} = \mathtt{Q}\,\mathtt{H}$ and $\hat{\mathbf{X}} = \mathtt{H}^{-1}\mathtt{Y}$ for some invertible matrix $\mathtt{H}$ is the only possibility[4]. This is the same thing as

$$\hat{\mathtt{P}}_i = \mathtt{Q}_i\mathtt{H} = \tau_i\mathtt{P}_i\mathtt{H} \tag{2.36}$$

$$\hat{\mathbf{X}}_j = \mathtt{H}^{-1}\mathbf{Y}_j = \nu_j\mathtt{H}^{-1}\mathbf{X}_j \tag{2.37}$$

[4]The proof is quite simple: The column space of $\mathtt{Q}$, $\hat{\mathtt{P}}$ and $\mathtt{A}$ must be equal and therefore we have $\hat{\mathtt{P}} = \mathtt{Q}\mathtt{H}$ for some invertible 4×4 matrix $\mathtt{H}$. Similarly, we can argue that $\hat{\mathbf{X}} = \mathtt{G}\mathtt{Y}$ for some invertible $\mathtt{G}$. Therefore, we have $\mathtt{Q}\,\mathtt{Y} = \mathtt{Q}\,\mathtt{H}\,\mathtt{G}\,\mathtt{Y}$. As $\mathtt{Q}$ has full column rank and $\mathtt{Y}$ has full row rank, the above implies $\mathtt{H}\,\mathtt{G} = \mathtt{I}$ and hence, $\mathtt{G} = \mathtt{H}^{-1}$.

Therefore, $(\{P_i\}, \{\mathbf{X}_j\})$ and $(\{\hat{P}_i\}, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent. □

2.6 Summary

In this section, we gave the reader a general background for understanding this thesis. We also showed the need for generalizing the current theory of projective reconstruction, both in the case of 3D to 2D projections and arbitrary dimensional projections. We also have a separate background section at the beginning of each of our main chapters which is specific to that chapter. The next chapter, Chapter 3, presents a generalized theory for projections from 3D to 2D. Chapter 4 studies arbitrary dimensional projections.

A Generalized Theorem for 3D to 2D Projections

In this chapter we consider the popular case of $\mathbb{P}^3 \to \mathbb{P}^2$ projections. Therefore, during the whole chapter it is assumed $\mathbf{X}_j \in \mathbb{R}^4$, $\mathbf{x}_{ij} \in \mathbb{R}^3$ and $\mathrm{P}_i \in \mathbb{R}^{3\times 4}$ for all i, j.

3.1 Background

3.1.1 The Fundamental Matrix

An important entity used in this chapter is the *fundamental matrix*. For two cameras, the fundamental matrix gives a bilinear relation between pairs of corresponding image points. To see the existence of a bilinear relation consider the projection relation for two views $i = 1, 2$

$$\lambda_{1j}\mathbf{x}_{1j} = \mathrm{P}_1\mathbf{X}_j \tag{3.1}$$
$$\lambda_{2j}\mathbf{x}_{2j} = \mathrm{P}_2\mathbf{X}_j \tag{3.2}$$

for nonzero projective depths λ_{1j} and λ_{2j}. One can write the above in matrix form:

$$\begin{bmatrix} \mathbf{x}_{1j} & \mathbf{0} & \mathrm{P}_1 \\ \mathbf{0} & \mathbf{x}_{2j} & \mathrm{P}_2 \end{bmatrix} \begin{pmatrix} \lambda_{1j} \\ \lambda_{2j} \\ -\mathbf{X}_j \end{pmatrix} = \mathbf{0}. \tag{3.3}$$

As λ_{1j} and λ_{2j} are nonzero, the above implies that the 6×6 matrix on the left hand side has a nonzero null vector, and therefore, a zero determinant:

$$\det \begin{bmatrix} \mathbf{x}_{1j} & \mathbf{0} & \mathrm{P}_1 \\ \mathbf{0} & \mathbf{x}_{2j} & \mathrm{P}_2 \end{bmatrix} = 0. \tag{3.4}$$

This implies a bilinear relation between $\mathbf{x}_{1j}$ and $\mathbf{x}_{2j}$ in the form of

$$\mathbf{x}_{2j}^T \mathrm{F}\, \mathbf{x}_{1j} = 0, \tag{3.5}$$

in which the ik-th element of the 3×3 matrix $\mathtt{F}$ is

$$f_{ki} = (-1)^{i+k} \det \begin{bmatrix} \mathtt{P}_{1,-i} \\ \mathtt{P}_{2,-k} \end{bmatrix} \tag{3.6}$$

where $\mathtt{P}_{1,-i} \in \mathbb{R}^{2\times4}$ is formed by removing the i-th row of $\mathtt{P}_1$, and similarly, $\mathtt{P}_{2,-k} \in \mathbb{R}^{2\times4}$ is the matrix $\mathtt{P}_2$ with its k-th row removed. The matrix $\mathtt{F}$ is called the *fundamental matrix* corresponding to camera matrices $\mathtt{P}_1$ and $\mathtt{P}_2$. Here, we use a function $\mathcal{F}\colon \mathbb{R}^{3\times4} \times \mathbb{R}^{3\times4} \to \mathbb{R}^{3\times3}$ to show the mapping (3.6) between the camera matrices and the fundamental matrix, that is $\mathtt{F} = \mathcal{F}(\mathtt{P}_1, \mathtt{P}_2)$.

Definition 3.1. *For two* 3×4 *matrices* $\mathtt{Q}$ *and* $\mathtt{R}$*, the fundamental matrix represented by* $\mathcal{F}(\mathtt{Q},\mathtt{R})$*, is defined as*

$$[\mathcal{F}(\mathtt{Q},\mathtt{R})]_{ki} = (-1)^{i+k} \det \begin{bmatrix} \mathtt{Q}_{-i} \\ \mathtt{R}_{-k} \end{bmatrix} \tag{3.7}$$

where $\mathtt{Q}_{-i} \in \mathbb{R}^{2\times4}$ *is formed by removing the* i*-th row of* $\mathtt{Q}$ *and* $\mathtt{R}_{-k}$ *is defined similarly.*

For more details on this definition we refer the reader to [Hartley and Zisserman, 2004, Sect. 17.1]. Notice that in (3.7) the fundamental matrix is the output of the function $\mathcal{F}$ applied to $\mathtt{Q}$ and $\mathtt{R}$ and not the mapping $\mathcal{F}$ itself. One of the advantages of using the above definition for fundamental matrix is that it is not restricted to the case of proper full-rank camera matrices. It can be defined for any pair of 3×4 matrices. Also, the reader must keep in mind that, like other entities in this thesis, the fundamental matrix here is treated as a member of $\mathbb{R}^{3\times3}$, not as an up-to-scale equivalence class of matrices. Basically, the above definition says that the elements of the fundamental matrix of two matrices $\mathtt{Q},\mathtt{R} \in \mathbb{R}^{3\times4}$ are minors of $\mathrm{stack}(\mathtt{Q},\mathtt{R})$ made by choosing two rows from $\mathtt{Q}$ and two rows from $\mathtt{R}$. This gives the following lemma

Lemma 3.1. *For two* 3×4 *matrices* $\mathtt{Q}$ *and* $\mathtt{R}$*, the fundamental matrix* $\mathcal{F}(\mathtt{Q},\mathtt{R})$ *is nonzero if and only if there exists a* non-singular 4×4 *submatrix of* $\mathrm{stack}(\mathtt{Q},\mathtt{R})$ *made by choosing two rows from* $\mathtt{Q}$ *and two rows from* $\mathtt{R}$*.*

The next two lemmas about the fundamental matrix will be used later on in this chapter.

Lemma 3.2 ([Hartley and Zisserman, 2004])**.** *Consider two pairs of camera matrices* $\mathtt{Q},\mathtt{R}$ *and* $\hat{\mathtt{Q}},\hat{\mathtt{R}}$ *such that* $\mathtt{Q}$ *and* $\mathtt{R}$ *both have full row rank and also have distinct null spaces, that is* $\mathcal{N}(\mathtt{Q}) \neq \mathcal{N}(\mathtt{R})$*. Then* $(\mathtt{Q},\mathtt{R})$ *and* $(\hat{\mathtt{Q}},\hat{\mathtt{R}})$ *are projectively equivalent according to Definition 2.1 if and only if* $\mathcal{F}(\mathtt{Q},\mathtt{R})$ *and* $\mathcal{F}(\hat{\mathtt{Q}},\hat{\mathtt{R}})$ *are equal up to a* nonzero *scaling factor.*

Notice that, unlike $(\mathtt{Q},\mathtt{R})$, no assumptions are made in the above about $(\hat{\mathtt{Q}},\hat{\mathtt{R}})$.

Lemma 3.3 ([Hartley and Zisserman, 2004])**.** *Consider two full-row-rank matrices* $\mathtt{Q}$ *and* $\mathtt{R}$ *such that* $\mathcal{N}(\mathtt{Q}) \neq \mathcal{N}(\mathtt{R})$*. If for a matrix* $\mathtt{F} \in \mathbb{R}^{3\times3}$ *the relation*

$$\mathtt{Q}^T\mathtt{F}\mathtt{R} + \mathtt{R}^T\mathtt{F}^T\mathtt{Q} = \mathtt{O}_{4\times4}$$

holds (or equivalently $\mathbf{X}^T(\mathtt{Q}^T\mathtt{F}\mathtt{R})\mathbf{X} = 0$ *holds for all* $\mathbf{X} \in \mathbb{R}^4$*), then* $\mathtt{F}$ *is equal to* $\mathcal{F}(\mathtt{Q},\mathtt{R})$ *up to a scaling factor.*

3.1.2 The Triangulation Problem

Triangulation is the process of determining the location of a 3D point given its images in two or more cameras with known camera matrices. The following lemma states that the solution to triangulation is unique in generic cases:

Lemma 3.4 (Triangulation). *Consider two full-row-rank camera matrices* $\mathtt{P}_1, \mathtt{P}_2 \in \mathbb{R}^{3\times 4}$*, two points* $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^4$*, and scalars* $\hat{\lambda}_1$ *and* $\hat{\lambda}_2$ *satisfying*

$$\mathtt{P}_1\mathbf{Y} = \hat{\lambda}_1\mathtt{P}_1\mathbf{X}, \tag{3.8}$$

$$\mathtt{P}_2\mathbf{Y} = \hat{\lambda}_2\mathtt{P}_2\mathbf{X}. \tag{3.9}$$

Take nonzero vectors $\mathbf{C}_1 \in \mathcal{N}(\mathtt{P}_1)$ *and* $\mathbf{C}_2 \in \mathcal{N}(\mathtt{P}_2)$*. If the three vectors* $\mathbf{C}_1$*,* $\mathbf{C}_2$ *and* $\mathbf{X}$ *are linearly independent, then* $\mathbf{Y}$ *is equal to* $\mathbf{X}$ *up to a scaling factor.*

Notice that the condition of $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{X}$ being linearly independent means that the two camera centres are distinct and $\mathbf{X}$ does not lie on the projective line joining them[1]. A geometric proof of this is given in [Hartley and Zisserman, 2004, Theorem 10.1]. Here, we give an algebraic proof as one might argue that [Hartley and Zisserman, 2004] has used projective equality relations which cannot be fully translated to our affine space equations since we do not assume that $\hat{\lambda}_1$ and $\hat{\lambda}_2$ are nonzero in (3.8) and (3.9).

Proof. Since $\mathtt{P}_1$ and $\mathtt{P}_2$ have full row rank they have a 1D null space. Thus, relations (3.8) and (3.9) respectively imply

$$\mathbf{Y} = \alpha_1\mathbf{C}_1 + \hat{\lambda}_1\mathbf{X}, \tag{3.10}$$

$$\mathbf{Y} = \alpha_2\mathbf{C}_2 + \hat{\lambda}_2\mathbf{X}, \tag{3.11}$$

for some scalars α_1 and α_2. These give $\alpha_1\mathbf{C}_1 + \hat{\lambda}_1\mathbf{X} = \alpha_2\mathbf{C}_2 + \hat{\lambda}_2\mathbf{X}$ or

$$\alpha_1\mathbf{C}_1 - \alpha_2\mathbf{C}_2 + (\hat{\lambda}_1 - \hat{\lambda}_2)\mathbf{X} = 0 \tag{3.12}$$

As the three vectors $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{X}$ are linearly independent, (3.12) implies that $\alpha_1 = 0$, $\alpha_2 = 0$ and $\hat{\lambda}_1 = \hat{\lambda}_2$. Define $\nu \stackrel{\text{def}}{=} \hat{\lambda}_1 = \hat{\lambda}_2$. Then, from (3.10) we have $\mathbf{Y} = \nu\mathbf{X}$. □

3.1.3 The Camera Resectioning Problem

Camera resectioning is the task of computing camera parameters given the 3D points and their images. It can be shown that with sufficient 3D points in general locations,

[1] For simplicity of notation, we are being a bit sloppy here about the projective entities like projective lines, quadric surfaces and twisted cubics. The reader must understand that when talking about a point $\mathbf{X} \in \mathbb{R}^4$ lying on a projective entity, what we really mean is that the projective point in $\mathbb{P}^3$ represented by $\mathbf{X}$ in homogeneous coordinates lies on it.

the camera matrix can be uniquely determined up to scale [Hartley and Zisserman, 2004]. Here, we consider a slightly revised version of this problem, which fits our case where the projective depths are not necessarily all nonzero and the second (estimated) set of camera matrices need not be assumed to have full rank, as stated in the following lemma:

Lemma 3.5 (Resectioning). *Consider a* 3×4 *matrix* $\mathtt{Q}$ *of rank 3 and a set of points* $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p$ *such that for a nonzero vector* $\mathbf{C} \in \mathcal{N}(\mathtt{Q})$ *we have*

(C1) Any four vectors among $\mathbf{C}, \mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p$ *are linearly independent, and*

(C2) the set of points $\{\mathbf{C}, \mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\}$ *do not lie on a twisted cubic (see footnote 1) or any of the degenerate critical sets resulting in a resection ambiguity (set out in [Hartley and Zisserman, 2004, Sect. 22.1]).*

Now, for any $\hat{\mathtt{Q}} \in \mathbb{R}^{3\times4}$ *if we have*

$$\alpha_j \mathtt{Q}\mathbf{X}_j = \beta_j \hat{\mathtt{Q}}\mathbf{X}_j \tag{3.13}$$

for all $j = 1, 2, \ldots, p$ *where scalars* α_j *and* β_j *are such that the vector* (α_j, β_j) *is nonzero for all* j*, then* $\hat{\mathtt{Q}} = a\mathtt{Q}$ *for some scalar* a*.*

Proof. First, since 6 points in general position completely specify a twisted cubic [Semple and Kneebone, 1952], (C2) implies that $p+1 \geq 7$, or $p \geq 6$.

If $\hat{\mathtt{Q}} = \mathtt{0}$, then $\hat{\mathtt{Q}} = a\mathtt{Q}$ with $a = 0$, proving the claim of the lemma. Thus, in what follows we only consider the case of $\hat{\mathtt{Q}} \neq \mathtt{0}$.

By (C1), for all j we have $\mathtt{Q}\mathbf{X}_j \neq \mathbf{0}$. Therefore, $\beta_j \neq 0$, as otherwise if $\beta_j = 0$ from $(\alpha_j, \beta_j)^T \neq \mathbf{0}$ we would have $\alpha \neq 0$ and therefore $\mathbf{0} = \beta_j \hat{\mathtt{Q}}\mathbf{X}_j = \alpha_j \mathtt{Q}\mathbf{X}_j \neq \mathbf{0}$, which is a contradiction. From $\beta_j \neq 0$ and (3.13) it follows that if $\alpha_j = 0$ for some j, then $\mathbf{X}_j \in \mathcal{N}(\hat{\mathtt{Q}})$. Now, if for 4 indices j we have $\alpha_j = 0$, from (C1) it follows that $\hat{\mathtt{Q}}$ has a 4D null space, or equivalently $\hat{\mathtt{Q}} = \mathtt{0}$. Since we excluded this case, we conclude that there are less than 4 zero-valued α_j-s. As $p \geq 6$, it follows that there are at least three nonzero α_j-s, namely α_{j_1}, α_{j_2} and α_{j_3}. Since β_j-s are all nonzero, $\alpha_j \neq 0$ along with (3.13) implies that $\mathtt{Q}\mathbf{X}_j$ is in $\mathcal{C}(\hat{\mathtt{Q}})$, the column space of $\hat{\mathtt{Q}}$. Therefore, we have $\mathrm{span}(\mathtt{Q}\mathbf{X}_{j_1}, \mathtt{Q}\mathbf{X}_{j_2}, \mathtt{Q}\mathbf{X}_{j_3}) \subseteq \mathcal{C}(\hat{\mathtt{Q}})$. From (C1) we know that $\mathrm{span}(\mathbf{X}_{j_1}, \mathbf{X}_{j_2}, \mathbf{X}_{j_3})$ is 3-dimensional and does not contain the null space of $\mathtt{Q}$. Therefore, $\mathrm{span}(\mathtt{Q}\mathbf{X}_{j_1}, \mathtt{Q}\mathbf{X}_{j_2}, \mathtt{Q}\mathbf{X}_{j_3})$ is also 3-dimensional. From $\mathrm{span}(\mathtt{Q}\mathbf{X}_{j_1}, \mathtt{Q}\mathbf{X}_{j_2}, \mathtt{Q}\mathbf{X}_{j_3}) \subseteq \mathcal{C}(\hat{\mathtt{Q}})$ then we conclude that $\hat{\mathtt{Q}}$ has full row rank.

As $\mathcal{R}\mathrm{ank}(\hat{\mathtt{Q}}) = 3$, we can consider it as a proper camera matrix in multiple view geometry, talking about its camera centre represented by its null space. Therefore, for two camera matrices $\mathtt{Q}$ and $\hat{\mathtt{Q}}$ and all the points $\mathbf{X}_j$ for which $\alpha_j \neq 0$ we can apply the results of the classic camera resectioning problem: It is known that for two (up to scale) distinct camera matrices $\mathtt{Q}$ and $\hat{\mathtt{Q}}$ to see the points $\mathbf{X}_j$ equally up to a possible nonzero scaling factor, the points $\mathbf{X}_j$ and the camera centres must lie on a common twisted cubic (or possibly some other specific degenerate sets, see [Hartley and Zisserman, 2004; Buchanan, 1988]).

$$\begin{bmatrix} & & a & & & \\ & & b & & & \\ c & d & x & e & f & g \\ & & h & & & \end{bmatrix} \quad \begin{bmatrix} a & b & c & x & d & e \\ & & & f & & \\ & & & g & & \\ & & & h & & \end{bmatrix} \quad \begin{bmatrix} a & & & & & \\ b & & & & & \\ c & & & & & \\ x & d & e & f & g & h \end{bmatrix}$$

Figure 3.1: Examples of 4×6 cross-shaped matrices. In cross-shaped matrices all elements of the matrix are zero, except those belonging to a special row r or a special column c of the matrix. The elements of the r-th row and the c-th column are all nonzero, except possibly the central element located at position (r,c). In the above examples, the blank parts of the matrices are zero. The elements $a,b,\ldots,h$ are all nonzero, while x can have any value (zero or nonzero).

Notice that, as $\mathcal{R}\text{ank}(\hat{\mathtt{Q}}) = 3$, (C1) implies that among the points $\mathbf{X}_j$ at most one lies on the null-space of $\hat{\mathtt{Q}}$ and therefore, by (3.13) we can say that at most one α_j can be zero. By possibly relabeling the points we assume that $\alpha_1,\ldots,\alpha_{p-1}$ are all nonzero.

Now to get a contradiction, assume that there is a resection ambiguity. We consider two cases namely $\alpha_p \neq 0$ and $\alpha_p = 0$. If $\alpha_p \neq 0$ then by $\alpha_j \mathtt{Q}\mathbf{X}_j = \beta_j \hat{\mathtt{Q}}\mathbf{X}_j$ we know that $\mathbf{X}_1,\ldots,\mathbf{X}_p$ are viewed equally up to scale by both $\mathtt{Q}$ and $\hat{\mathtt{Q}}$ and thus $\mathbf{X}_1,\ldots,\mathbf{X}_6$ along with the camera centre of $\mathtt{Q}$ must lie on a twisted cubic (or other degenerate sets leading to a resection ambiguity), which is impossible due to (C2). If $\alpha_6 = 0$, implying $\mathbf{X}_6 \in \mathcal{N}(\hat{\mathtt{Q}})$, then again the camera center of $\mathtt{Q}$, $\mathbf{X}_1,\ldots,\mathbf{X}_5$ and $\mathbf{X}_6$ (this time as the camera centre of $\hat{\mathtt{Q}}$) must lie on a twisted cubic (or the degenerate sets), contradicting with (C2). Hence there can be no resection ambiguity and $\mathtt{Q}$ and $\hat{\mathtt{Q}}$ must be equal up to a scaling factor. □

3.1.4 Cross-shaped Matrices

The concept of *cross-shaped* matrices is important for the statement of our main theorem and the characterization of false solutions to the projective factorization problem.

Definition 3.2. *A matrix* $\mathtt{A} = [a_{ij}]$ *is said to be* cross-shaped, *if it has a row* r *and a column* c *for which*

$$\begin{cases} a_{ij} = 0 & i \neq r, j \neq c, \\ a_{ij} \neq 0 & i = r, j \neq c, \\ a_{ij} \neq 0 & i \neq r, j = c. \end{cases} \tag{3.14}$$

The pair of indices (r,c) *is called the* centre *of a cross-shaped matrix and* a_{rc} *is called its* central element, *which can be either zero or nonzero. A cross-shaped matrix can be* zero-centred *or* nonzero-centred *depending on whether the central element* a_{rc} *is zero or nonzero.*

A cross-shaped matrix has all of its elements equal to zero except the elements of a certain row r and a certain column c. The r-th row and the c-th column have all nonzero elements, except at their junction where the element can be zero or nonzero.

Examples of cross-shaped matrices are depicted in Fig. 3.1. Notice that any permutation to rows and columns of a cross-shaped matrix results in another cross-shaped matrix.

Lemma 3.6. *(i) Any two $m \times n$ nonzero-centred cross-shaped matrices with a common centre (r, c) are diagonally equivalent. (ii) any two $m \times n$ zero-centred cross-shaped matrices with a common centre (r, c) are diagonally equivalent.*

Proof. Consider two $m \times n$ cross-shaped matrices $\mathtt{A} = [a_{ij}]$ and $\mathtt{B} = [b_{ij}]$ with a common centre (r, c). According to Definition 2.2, to prove diagonal equivalence we need to show that $\mathtt{B} = \mathrm{diag}(\boldsymbol{\tau})\, \mathtt{A}\, \mathrm{diag}(\boldsymbol{\nu})$ for some vectors $\boldsymbol{\tau}$ and $\boldsymbol{\nu}$ with all nonzero entries. If $\mathtt{A}$ and $\mathtt{B}$ are both zero-centred, that is $a_{rc} = b_{rc} = 0$, then we choose the vectors $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_m)^T$ and $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_n)^T$, such that $\tau_r = \nu_c = 1$, $\tau_i = b_{ic}/a_{ic}$ for $i \neq r$, and $\nu_j = b_{rj}/a_{rj}$ for $j \neq c$. If $\mathtt{A}$ and $\mathtt{B}$ are both nonzero-centred, that is $a_{rc} \neq 0$ and $b_{rc} \neq 0$, then the vectors $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_m)^T$ and $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_n)^T$ are chosen such that $\tau_i = b_{ic}/a_{ic}$ for $i = 1, \ldots, m$, $\nu_c = 1$, and $\nu_j = b_{rj}/(a_{rj}\tau_r)$ for $j \neq c$. In either cases, one can easily check that $\boldsymbol{\tau}$ and $\boldsymbol{\nu}$ have all-nonzero entries and $\mathtt{B} = \mathrm{diag}(\boldsymbol{\tau})\, \mathtt{A}\, \mathrm{diag}(\boldsymbol{\nu})$. □

Now, we have the required tools to state our main theorem on projective reconstruction.

3.2 A General Projective Reconstruction Theorem

Here, we give a projective reconstruction theorem which is more general than the classic theorem in the sense that it does not assume, a priori, that the estimated depths $\hat{\lambda}_{ij}$ are all nonzero. This provides significantly more flexibility in the choice of depth constraints for the projective depth estimation algorithms.

Theorem 3.1. *Consider a set of $m \geq 2$ camera matrices $\{\mathtt{P}_i\}$ and $n \geq 8$ points $\{\mathbf{X}_j\}$ which are generic in the sense of conditions (G1-G4) which will be introduced later, and project into a set of image points $\{\mathbf{x}_{ij}\}$ according to*

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j, \tag{3.15}$$

for nonzero depths $\lambda_{ij} \neq 0$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Now, consider any other configuration of m camera matrices $\{\hat{\mathtt{P}}_i\}$, n points $\{\hat{\mathbf{X}}_j\}$ and mn depths $\{\hat{\lambda}_{ij}\}$ related to the same image data $\{\mathbf{x}_{ij}\}$ by

$$\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathtt{P}}_i\hat{\mathbf{X}}_j. \tag{3.16}$$

If the depth matrix $\hat{\Lambda} = [\hat{\lambda}_{ij}]$ satisfies the following conditions

(D1) $\hat{\Lambda}$ has no zero rows,

(D2) $\hat{\Lambda}$ has no zero columns, and

(D3) $\hat{\Lambda}$ *is not a cross-shaped matrix (see Definition 3.2),*

then the camera-point configuration $(\{\hat{P}_i\}, \{\hat{X}_j\})$ *is projectively equivalent to* $(\{P_i\}, \{X_j\})$.

Loosely speaking, by true camera matrices P_i and points X_j being generic, we mean that the camera matrices have full row rank and the points and camera centres are in general position. In Sect. 3.2.1 we will be more specific about the required genericity conditions and mention four generic properties (G1-G4) under which Theorem 3.1 is true. To understand the results, it is essential to notice that the genericity assumptions only apply to the ground truth data $(\{P_i\}, \{X_j\})$. No assumption is made about the estimated (hatted) quantities $\hat{P}_i$ and $\hat{X}_j$ except the relation $\hat{\lambda}_{ij}x_{ij} = \hat{P}_i\hat{X}_j$. We do not a priori rule out the possibility that $\hat{P}_i$-s or $\hat{X}_j$-s belong to some non-generic set. Referring to $\hat{P}_i$-s as camera matrices carries no implications about them whatsoever other than that they are 3×4 real matrices. They can be rank-deficient or even zero unless the opposite is proven.

At a first glance, theorem (3.1) might seem contradictory, as it says that only some small subset of the elements of $\hat{\Lambda} = [\hat{\lambda}_{ij}]$ being nonzero is sufficient for $(\{P_i\}, \{X_j\})$ and $(\{\hat{P}_i\}, \{\hat{X}_j\})$ being projectively equivalent. On the other hand, from Lemma 2.2 we know that if $(\{P_i\}, \{X_j\})$ and $(\{\hat{P}_i\}, \{\hat{X}_j\})$ are projectively equivalent, then $\hat{\Lambda}$ must be diagonally equivalent to Λ and hence have all nonzero elements. The matter is that one has to distinguish between the implications of depth assumptions (D1-D3) in their own rights and their implications combined with the relations $\hat{\lambda}_{ij}x_{ij} = \hat{P}_i\hat{X}_j$. Theorem 3.1, therefore, implies that if a special subset of depths $\{\hat{\lambda}_{ij}\}$ are known to be nonzero, then all of them are. This provides a sound theoretical base for choosing and analysing depth constraints for factorization-based projective reconstruction.

Here, we state the general outline of the proof. Each part of the proof will then be demonstrated in a separate subsection.

Sketch of the Proof for Theorem 3.1. Under the theorem's assumptions, we shall show the following:

- There exist at least two views k and l for which the fundamental matrix $\mathcal{F}(\hat{P}_k, \hat{P}_l)$ is nonzero (section 3.2.2).
- If $\mathcal{F}(\hat{P}_k, \hat{P}_l) \neq 0$ then the two configurations $(P_k, P_l, \{X_j\})$ and $(\hat{P}_k, \hat{P}_l, \{\hat{X}_j\})$ are projectively equivalent (section 3.2.3).
- If for two views k and l, $(P_k, P_l, \{X_j\})$ and $(\hat{P}_k, \hat{P}_l, \{\hat{X}_j\})$ are projectively equivalent, then $(\{P_i\}, \{X_j\})$ and $(\{\hat{P}_i\}, \{\hat{X}_j\})$ are projectively equivalent (section 3.2.4).

This completes the proof. □

Furthermore, we shall show in Sect. 3.2.5 that if any of the depth assumptions (D1), (D2) or (D3) is relaxed, it allows the existence of a configuration $(\{\hat{P}_i\}, \{\hat{X}_j\})$, satisfying the relations $\hat{\lambda}_{ij}x_{ij} = \hat{P}_i\hat{X}_j$ and projectively non-equivalent to $(\{P_i\}, \{X_j\})$. The reader can jump to Sect. 3.2.5 if they are not interested in the details of the proof.

Before stating the different parts of the proof, it is worth mentioning that for proving Theorem 3.1 one may simply assume that the set of true depths λ_{ij} are all equal to one. This can be seen by a simple change of variables $\mathbf{x}'_{ij} = \lambda_{ij}\mathbf{x}_{ij}$, $\lambda'_{ij} = 1$ and $\hat{\lambda}'_{ij} = \hat{\lambda}_{ij}/\lambda_{ij}$, implying $\lambda'_{ij}\mathbf{x}'_{ij} = \mathbf{x}'_{ij} = \mathtt{P}_i\mathbf{X}_j$ and $\hat{\lambda}'_{ij}\mathbf{x}'_{ij} = \hat{\mathtt{P}}_i\hat{\mathbf{X}}_j$. Notice that $\hat{\lambda}'_{ij} = \hat{\lambda}_{ij}/\lambda_{ij}$ is zero if and only if $\hat{\lambda}_{ij}$ is zero. Therefore, (D1-D3) are true for the $\hat{\lambda}'_{ij}$-s if and only if they hold for the $\hat{\lambda}_{ij}$-s. This change of variables requires $\lambda_{ij} \neq 0$ which was among the assumptions of the theorem (and even if it was not, it would follow as a simple consequence of $\mathtt{P}_i\mathbf{X}_j \neq \mathbf{0}$ from (G2-1) below and the relations $\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j$). Throughout the proof of Theorem 3.1, we assume $\lambda_{ij} = 1$. With this assumption, the equations (3.15) and (3.16) are combined into

$$\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j. \tag{3.17}$$

Theorem 3.1 is proved as a conjunction of several lemmas. Therefore, to avoid redundancy, we assume the following assumptions throughout all steps of the proof:

There exist $m \geq 2$ camera matrices $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m \in \mathbb{R}^{3\times 4}$ and $n \geq 8$ points $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^4$ (called the *true* sets of camera matrices and points, or the *ground truth*), and an *estimated* setup of m camera matrices and n points $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$, related by (3.17) for a set of scalars $\{\hat{\lambda}_{ij}\}$.

Each of the genericity assumptions (G1-G4) about the ground truth $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ and the depth assumptions (D1-D3) about the estimated depths $\{\hat{\lambda}_{ij}\}$ will be mentioned explicitly whenever needed.

3.2.1 The Generic Camera-Point Setup

It is known that projective reconstruction from image data can be problematic if the (true) camera matrices and points belong to special degenerate setups [Hartley and Kahl, 2007]. The Projective Reconstruction Theorem is then said to be generically true, meaning that is can be proved under some generic assumptions about how the ground truth is configured. Here, we list the generic assumptions made about the ground truth for the proof of our theorem.

We assume that there exist $m \geq 2$ camera matrices $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m \in \mathbb{R}^{3\times 4}$ and $n \geq 8$ points $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ in $\mathbb{R}^4$. They are generically configured in the following sense:

(G1) All camera matrices $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m \in \mathbb{R}^{3\times 4}$ have full row rank.

(G2) Taking any two views i and k, and two nonzero vectors $\mathbf{C}_i \in \mathcal{N}(\mathtt{P}_i)$ and $\mathbf{C}_k \in \mathcal{N}(\mathtt{P}_k)$, any four vectors among $\mathbf{C}_i, \mathbf{C}_k, \mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$, are linearly independent.

(G3) For any view i, and a nonzero vector $\mathbf{C}_i \in \mathcal{N}(\mathtt{P}_i)$, no n points among $\mathbf{C}_i, \mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ lie on a twisted cubic (see footnote 1), or any of the degenerate critical sets resulting in a resection ambiguity, (see [Hartley and Zisserman, 2004, Sect. 22.1] and [Hartley and Kahl, 2007]).

(G4) For any two views i and k, and two nonzero vectors $\mathbf{C}_i \in \mathcal{N}(\mathtt{P}_i)$ and $\mathbf{C}_k \in \mathcal{N}(\mathtt{P}_k)$, the points $\{\mathbf{C}_i, \mathbf{C}_k\} \cup \{\mathbf{X}_j\}_{j=1,\ldots,n}$ do not all lie on any (proper or degen-

erate) ruled quadric surface (see [Hartley and Zisserman, 2004, Sect. 22.2] and [Hartley and Kahl, 2007], also look at footnote 1).

Obviously, condition (G1) makes the choice of $\mathbf{C}_i$ and $\mathbf{C}_k$ in conditions (G2-G4) unique up to scale. It implies that that any nonzero $\mathbf{C}_i \in \mathcal{N}(\mathtt{P}_i)$ represents the camera centre of $\mathtt{P}_i$. Notice that conditions (G3) and (G4) are generic for $n \geq 8$, because of the facts that 6 points in general position completely specify a twisted cubic and 9 points in general position determine a quadric surface [Semple and Kneebone, 1952]. Condition (G1-G4) are not tight for the proof of Theorem 3.1. One might find tighter generic conditions under which our projective reconstruction theorem is still true. However, we avoid doing this as it unnecessarily complicates the proofs.

Condition (G2) has many implications when combined with (G1). Here, we list the ones needed in the proofs:

(G2-1) For all i and j we have $\mathtt{P}_i\mathbf{X}_j \neq \mathbf{0}$ (as for any nonzero $\mathbf{C}_i \in \mathcal{N}(\mathtt{P}_i)$, $\mathbf{C}_i$ and $\mathbf{X}_j$ are linearly independent). Geometrically, $\mathbf{X}_j$ does not coincide with the camera centre of $\mathtt{P}_i$.

(G2-2) For any two views i, k we have $\mathcal{N}(\mathtt{P}_i) \neq \mathcal{N}(\mathtt{P}_k)$, and hence, no pair of cameras share a common camera centre.

(G2-3) For any two views i, k, $\text{stack}(\mathtt{P}_i, \mathtt{P}_k)$ has full row rank, and therefore, so does $\mathtt{P} = \text{stack}(\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m)$.

(G2-4) For any two views i, k, and any point $\mathbf{X}_j$, the three nonzero vectors $\mathbf{C}_i$, $\mathbf{C}_k$ and $\mathbf{X}_j$ are linearly independent and therefore, $\mathbf{X}_j$ does not lie on the projective line (see footnote 1) joining the camera centres of $\mathtt{P}_i$ and $\mathtt{P}_k$.

(G2-5) For any view i, any three vectors among $\mathtt{P}_i\mathbf{X}_1, \mathtt{P}_i\mathbf{X}_2, \ldots, \mathtt{P}_i\mathbf{X}_n$ are linearly independent (as $\mathbf{C}_i \notin \text{span}(\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3)$ for any three distinct vectors $\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3 \in \{\mathbf{X}_j\}$ and any nonzero vector $\mathbf{C}_i \in \mathcal{N}(\mathtt{P}_i)$).

3.2.2 The Existence of a Nonzero Fundamental Matrix

The object of this section is to prove the following lemma:

Lemma 3.7. *If the genericity assumptions (G1-G4) hold for $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$, and depth assumptions (D1-D3) hold for $\{\hat{\lambda}_{ij}\}$, there exist two views k and l such that the corresponding fundamental matrix $\mathcal{F}(\hat{\mathtt{P}}_k, \hat{\mathtt{P}}_l)$ is nonzero.*

We remind the reader that, as mentioned at the beginning of this section, all the lemmas here are under the assumption that there exist two sets of camera-point configurations $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ with $m \geq 2$ views and $n \geq 8$ points both projecting into the same image points $\{\mathbf{x}_{ij}\}$ through $\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j$ and $\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathtt{P}}_i\hat{\mathbf{X}}_j$ for all i and j.

Using Lemma 3.1, one can say that what is claimed in Lemma 3.7 is equivalent to the existence of an invertible 4×4 submatrix of $\text{stack}(\hat{\mathtt{P}}_k, \hat{\mathtt{P}}_l)$ for some views k and l,

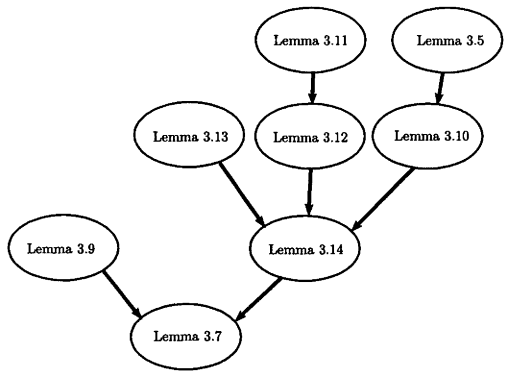


Figure 3.2: The inference graph for the proof of Lemma 3.7. Lemma 3.8 has been omitted due to its frequent use.

made by choosing two rows from $\hat{\mathtt{P}}_k$ and two rows from $\hat{\mathtt{P}}_l$. This lemma is essential for the proof of our last theorem. One reason is that the case of zero fundamental matrices for all pairs of views happens in the cross-shaped degenerate solutions. We will see later in section 3.2.5 that a cross-shaped depth matrix $\hat{\Lambda}$ happens when for one special view r we have $\mathcal{R}\text{ank}(\hat{\mathtt{P}}_r) = 3$ and $\mathcal{R}\text{ank}(\hat{\mathtt{P}}_i) = 1$ for all other views $i \neq r$. One can easily see from Lemma 3.1 that in this case all pairwise fundamental matrices are zero.

Surprisingly, Lemma 3.7 is the hardest step in the proof of Theorem 3.1. We prove this lemma as a consequence of a series of lemmas. Fig. 3.2 can help the reader to keep track of the inference process. The reader might notice that there are different ways of proving some of the lemmas here. Part of this is because the genericity conditions (G1-G4) are not tight. First, we state a lemma giving some simple facts about the second configuration of cameras, points and depths $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\}, \{\hat{\lambda}_{ij}\})$.

Lemma 3.8. *Under (G1, G2) and (D1, D2) The following hold*

(i) For all j we have $\hat{\mathbf{X}}_j \neq \mathbf{0}$, and for all i we have $\hat{\mathtt{P}}_i \neq 0$,

(ii) $\hat{\lambda}_{ij} = 0$ if and only if $\hat{\mathbf{X}}_j \in \mathcal{N}(\hat{\mathtt{P}}_i)$, where $\mathcal{N}(\hat{\mathtt{P}}_i)$ is the null space of $\hat{\mathtt{P}}_i$.

(iii) $\mathcal{R}\text{ank}(\hat{\mathtt{P}}_i) \geq \min(3, n_i)$, where n_i is the number of nonzero elements among $\hat{\lambda}_{i1}, \hat{\lambda}_{i2}, \ldots, \hat{\lambda}_{in}$,

(iv) If $\mathcal{R}\text{ank}(\hat{\mathtt{P}}_i) = 3$, then for any other view $k \neq i$, either the matrix $\text{stack}(\hat{\mathtt{P}}_i, \hat{\mathtt{P}}_k)$ has full column rank or for all j, $\hat{\lambda}_{ij} = 0$ implies $\hat{\lambda}_{ik} = 0$.

(v) If $\mathcal{R}\text{ank}(\hat{\mathtt{P}}_i) = 3$, all the points $\hat{\mathbf{X}}_j$ for which $\hat{\lambda}_{ij} = 0$ are equal up to a nonzero scaling factor.

Proof. To see (i), notice that for any i and j if we have $\hat{\lambda}_{ij} \neq 0$, then from $\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j$ and $\mathtt{P}_i\mathbf{X}_j \neq \mathbf{0}$ (G2-1) we conclude that $\hat{\mathbf{X}}_j \neq \mathbf{0}$ and $\hat{\mathtt{P}}_i \neq 0$. Then (i) follows from the fact that at each row and each column of $\hat{\Lambda} = [\hat{\lambda}_{ij}]$ there exists at least one nonzero element due to (D1, D2).

(ii) is obvious by $\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j$ from (3.17) and the fact that $\mathtt{P}_i\mathbf{X}_j \neq \mathbf{0}$ from (G2-1).

To prove (iii), notice that if $\hat{\lambda}_{ij}$ is nonzero for some i and j, from $\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j$ we conclude that $\mathtt{P}_i\mathbf{X}_j \in \mathcal{C}(\hat{\mathtt{P}}_i)$, where $\mathcal{C}(\hat{\mathtt{P}}_i)$ denotes the column space of $\hat{\mathtt{P}}_i$. Now, if there are n_i nonzero $\hat{\lambda}_{ij}$-s for view i, which (by a possible relabeling) we assume they are $\hat{\lambda}_{i1}, \hat{\lambda}_{i2}, \ldots, \hat{\lambda}_{in_i}$, then $\text{span}(\mathtt{P}_i\mathbf{X}_1, \mathtt{P}_i\mathbf{X}_2, \ldots, \mathtt{P}_i\mathbf{X}_{n_i}) \subseteq \mathcal{C}(\hat{\mathtt{P}}_i)$. By (G2-5) then we have $\min(3, n_i) = \dim\left(\text{span}(\mathtt{P}_i\mathbf{X}_1, \mathtt{P}_i\mathbf{X}_2, \ldots, \mathtt{P}_i\mathbf{X}_{n_i})\right) \leq \dim(\mathcal{C}(\hat{\mathtt{P}}_i)) = \mathcal{R}\text{ank}(\hat{\mathtt{P}}_i)$.

To see (iv), notice that as $\mathcal{R}\text{ank}(\hat{\mathtt{P}}_i) = 3$, if the matrix $\text{stack}(\hat{\mathtt{P}}_i, \hat{\mathtt{P}}_k)$ has a rank of less than 4, the row space of $\hat{\mathtt{P}}_i$ includes that of $\hat{\mathtt{P}}_k$, that is $\mathcal{R}(\hat{\mathtt{P}}_k) \subseteq \mathcal{R}(\hat{\mathtt{P}}_i)$, and thus $\mathcal{N}(\hat{\mathtt{P}}_i) \subseteq \mathcal{N}(\hat{\mathtt{P}}_k)$. Hence, from part (ii) of the lemma we have $\hat{\lambda}_{ij} = 0 \Leftrightarrow \mathbf{X}_j \in \mathcal{N}(\hat{\mathtt{P}}_i) \Rightarrow \mathbf{X}_j \in \mathcal{N}(\hat{\mathtt{P}}_k) \Leftrightarrow \hat{\lambda}_{ik} = 0$.

(v) simply follows from parts (i) and (ii) of this lemma and the fact that a $\hat{\mathtt{P}}_i$ of rank 3 has a 1D null space.

□

We make extensive use of Lemma 3.8 in what comes next. The reader might want to keep sight of it while reading this section.

Lemma 3.9. *Consider two* 3×4 *matrices* $\mathtt{Q}$ *and* $\mathtt{R}$ *such that* $\mathcal{R}\text{ank}(\mathtt{Q}) \geq 2$ *and* $\mathcal{R}\text{ank}(\mathtt{R}) \geq 2$. *Then* $\mathcal{F}(\mathtt{Q}, \mathtt{R}) \neq 0$ *if and only if* $\text{stack}(\mathtt{Q}, \mathtt{R})$ *has rank 4.*

Proof. Assume $\text{stack}(\mathtt{Q}, \mathtt{R})$ has rank 4. If $\mathtt{R}$ and $\mathtt{Q}$ have both rank 3, then $\text{stack}(\mathtt{Q}, \mathtt{R})$ having rank 4 means $\mathcal{N}(\mathtt{R}) \neq \mathcal{N}(\mathtt{Q})$. Geometrically, it means that $\mathtt{R}$ and $\mathtt{Q}$ are two rank-3 camera matrices with different camera centres. It is well known that in this case the fundamental matrix $\mathcal{F}(\mathtt{Q}, \mathtt{R})$ is nonzero [Hartley and Zisserman, 2004].

If $\mathtt{R}$ has rank 2, it has two rows $\mathbf{r}_i^T$ and $\mathbf{r}_j^T$ spanning its row space, that is $\text{span}(\mathbf{r}_i, \mathbf{r}_j) = \mathcal{R}(\mathtt{R})$. Further, as $\text{stack}(\mathtt{Q}, \mathtt{R})$ has rank 4, there exist at least two rows $\mathbf{q}_k^T$ and $\mathbf{q}_l^T$ of $\mathtt{Q}$ such that $\dim(\text{span}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{q}_k, \mathbf{q}_l)) = 4$. The two rows $\mathbf{q}_k$ and $\mathbf{q}_l$ can be chosen by taking the set $\{\mathbf{r}_i, \mathbf{r}_j\}$, adding rows of $\mathtt{Q}$, one by one, to this set, and choose the two rows whose addition leads to a jump in the dimension the span of the vectors in the set. As, the 4×4 matrix $\text{stack}(\mathbf{r}_i^T, \mathbf{r}_j^T, \mathbf{q}_k^T, \mathbf{q}_l^T)$ has rank 4, Lemma 3.1 suggests that $\mathcal{F}(\mathtt{Q}, \mathtt{R}) \neq 0$.

The other direction of the lemma is proved immediately from Lemma 3.1. □

Lemma 3.9 shows that to prove the main Lemma 3.7, it is sufficient to find two camera matrices both of rank 2 or more, whose vertical concatenation gives a matrix of rank 4. We will show in Lemma 3.14 that this is possible. But, to get there we need two extra lemmas. The next lemma relies on the Camera Resectioning Lemma discussed in Sect. 3.1.3.

Lemma 3.10. *Under (G1-G3), if for two distinct views* k *and* l, *there are at least* $n-1$ *indices* j *among the point indices* $1, 2, \ldots, n$, *for which the vector* $(\hat{\lambda}_{kj}, \hat{\lambda}_{lj})$ *is nonzero, we cannot have* $\mathcal{R}(\hat{\mathtt{P}}_l) \subseteq \mathcal{R}(\hat{\mathtt{P}}_k)$, *where* $\mathcal{R}$ *denotes the row space of a matrix.*

Proof. To get a contradiction, assume $\mathcal{R}(\hat{\mathtt{P}}_l) \subseteq \mathcal{R}(\hat{\mathtt{P}}_k)$. Then there must exist a 3×3 matrix $\mathtt{H}$ such that $\hat{\mathtt{P}}_l = \mathtt{H}\hat{\mathtt{P}}_k$. Therefore, for all j we have $\hat{\mathtt{P}}_l\hat{\mathbf{X}}_j = \mathtt{H}\hat{\mathtt{P}}_k\hat{\mathbf{X}}_j$ and by (3.17), that is $\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j$, we get $\hat{\lambda}_{lj}\mathtt{P}_l\mathbf{X}_j = \hat{\lambda}_{kj}\mathtt{H}\mathtt{P}_k\mathbf{X}_j$ for all j. Now, we can apply

Lemma 3.5 on Camera Resectioning (see Appendix 3.1.3) as $(\hat{\lambda}_{kj}, \hat{\lambda}_{lj})$ is nonzero for at least $n-1$ indices j and (G1-G3) hold[2]. By applying Lemma 3.5 we get

$$\mathtt{H}\mathtt{P}_k = a\,\mathtt{P}_l. \tag{3.18}$$

for some scalar a. Now notice that $\mathtt{H} \neq 0$, as otherwise from $\hat{\mathtt{P}}_l = \mathtt{H}\hat{\mathtt{P}}_k$ we have $\hat{\mathtt{P}}_l = 0$, which is excluded due to Lemma 3.8(i). As $\mathtt{H} \neq 0$ and $\mathtt{P}_k$ has full row rank according to (G1), then the scalar a in (3.18) cannot be zero. Therefore, we have

$$\mathtt{P}_l = \frac{1}{a}\,\mathtt{H}\mathtt{P}_k \tag{3.19}$$

meaning $\mathcal{R}(\mathtt{P}_l) \subseteq \mathcal{R}(\mathtt{P}_k)$. This possibility is excluded by (G1, G2-2) and hence we get a contradiction. This completes the proof. □

Lemma 3.11. *If (D1, D2) and (G1, G2) hold, then for at least one view i we have* $\mathcal{R}\text{ank}(\hat{\mathtt{P}}_i) \geq 2$.

Proof. To get a contradiction, assume that no matrix $\hat{\mathtt{P}}_i$ has rank 2 or more. As $\hat{\mathtt{P}}_i$-s are nonzero (Lemma 3.8(i)), we conclude that all $\hat{\mathtt{P}}_i$-s have rank 1. By (D1) and Lemma 3.8(iii) then each row of $\hat{\Lambda}$ must have exactly one nonzero element. Moreover, according to (D2), all columns of $\hat{\Lambda}$ have at least one nonzero element. These two facts imply that $m \geq n$ and that (by a possible relabeling of the views) rows of $\hat{\Lambda}$ can be permuted such that its top $n \times n$ block is a diagonal matrix $\mathtt{D}_{n\times n}$ with all nonzero diagonal elements, that is

$$\hat{\Lambda} = \begin{bmatrix} \mathtt{D}_{n\times n} \\ \mathtt{A} \end{bmatrix} \tag{3.20}$$

where $\mathtt{D}_{n\times n} = \mathrm{diag}(\hat{\lambda}_{11}, \hat{\lambda}_{22}, \ldots, \hat{\lambda}_{nn})$ and $\hat{\lambda}_{jj} \neq 0$ for all $j = 1, \ldots, n$. Using the relations $\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j$, the above gives

$$\begin{bmatrix} \hat{\mathtt{P}}_1 \\ \hat{\mathtt{P}}_2 \\ \vdots \\ \hat{\mathtt{P}}_n \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}}_1 \hat{\mathbf{X}}_2 \ldots \hat{\mathbf{X}}_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 & & & \\ & \mathbf{v}_2 & & \\ & & \ddots & \\ & & & \mathbf{v}_n \end{bmatrix} \tag{3.21}$$

where the $3m \times n$ matrix on the right hand side is block-diagonal with nonzero diagonal blocks $\mathbf{v}_j = \hat{\lambda}_{jj}\mathtt{P}_j\mathbf{X}_j \neq 0$ (as $\hat{\lambda}_{jj} \neq 0$ and $\mathtt{P}_j\mathbf{X}_j \neq 0$ due to (G2-1)). This suggests that on the right hand side there is a matrix of rank n. On the other hand, the left hand side of (3.21) has rank 4 or less as $[\hat{\mathbf{X}}_1 \hat{\mathbf{X}}_2 \ldots \hat{\mathbf{X}}_n]$ is $4 \times n$. This is a contradiction since $n \geq 8$. □

[2]According to (G3) the $n-1$ points $\mathbf{X}_j$ corresponding to nonzero zero vectors $(\hat{\lambda}_{kj}, \hat{\lambda}_{lj})$ and the camera centre of $\mathtt{P}_l$ do not all lie on a twisted cubic. This is a generic property as $n - 1 \geq 6$ (see Sect. 3.2.1). Notice that here the matrices $\mathtt{P}_l$ and $\mathtt{H}\mathtt{P}_k$ respectively act as $\mathtt{Q}$ and $\hat{\mathtt{Q}}$ in Lemma 3.5. The genericity conditions (G1-G3) provide the conditions (C1, C2) in Lemma 3.5.

Lemma 3.12. *If (D1, D2) and (G1, G2) hold, then for at least one view i we have* $\mathcal{R}\text{ank}(\hat{\mathbf{P}}_i) = 3$.

Proof. To get a contradiction, we assume that $\mathcal{R}\text{ank}(\hat{\mathbf{P}}_i) \leq 2$ for all i. According to Lemma 3.8(iii), this implies that any row $\hat{\mathbf{\Lambda}}$ has at most two nonzero element. Consider an arbitrary view l. We know that among $\hat{\lambda}_{l1}, \hat{\lambda}_{l2}, \ldots, \hat{\lambda}_{ln}$ at most two are nonzero. By relabeling the points $\{\mathbf{X}_j\}$ and accordingly $\{\hat{\mathbf{X}}_j\}$ if necessary, we can assume that $\hat{\lambda}_{l3} = \hat{\lambda}_{l4} = \cdots = \hat{\lambda}_{ln} = 0$. Now, by (D2), we know that the third column of $\hat{\mathbf{\Lambda}}$ is not zero and therefore, there must be some view k for which $\hat{\lambda}_{k3} \neq 0$. As the k-th row of $\hat{\mathbf{\Lambda}}$ has at most two nonzero elements, by relabeling the points $\mathbf{X}_4, \ldots, \mathbf{X}_n$ and accordingly $\hat{\mathbf{X}}_4, \ldots, \hat{\mathbf{X}}_n$, we can assume that $\hat{\lambda}_{k5} = \hat{\lambda}_{k6} = \cdots = \hat{\lambda}_{kn} = 0$. Notice that this relabeling retains $\hat{\lambda}_{l3} = \hat{\lambda}_{l4} = \cdots = \hat{\lambda}_{ln} = 0$.

Now, as $n \geq 8$, we consider the points $\hat{\mathbf{X}}_5$, $\hat{\mathbf{X}}_6$ and $\hat{\mathbf{X}}_7$. They cannot be equal up to scale. The reason is that if they are equal up to scale then by Lemma 3.8(ii), for any view i, the depths $\hat{\lambda}_{i5}$, $\hat{\lambda}_{i6}$ and $\hat{\lambda}_{i7}$ are either all zero or all nonzero. It follows by (D2) that there must be a view i for which $\hat{\lambda}_{i5}$, $\hat{\lambda}_{i6}$ and $\hat{\lambda}_{i7}$ are all nonzero. But this means that $\mathcal{R}\text{ank}(\hat{\mathbf{P}}_i) = 3$ by Lemma 3.8(iii), contradicting our assumption $\mathcal{R}\text{ank}(\hat{\mathbf{P}}_i) \leq 2$ for all i.

Because $\hat{\mathbf{X}}_5$, $\hat{\mathbf{X}}_6$ and $\hat{\mathbf{X}}_7$ are not equal up to scale, the dimension of $\text{span}(\hat{\mathbf{X}}_5, \hat{\mathbf{X}}_6, \hat{\mathbf{X}}_7)$ is at least 2. As $\hat{\lambda}_{k3} \neq 0$ and $\hat{\lambda}_{k5} = \hat{\lambda}_{k6} = \hat{\lambda}_{k7} = 0$, by Lemma 3.8(ii) we have $\hat{\mathbf{X}}_3 \notin \mathcal{N}(\hat{\mathbf{P}}_k)$ and $\text{span}(\hat{\mathbf{X}}_5, \hat{\mathbf{X}}_6, \hat{\mathbf{X}}_7) \subseteq \mathcal{N}(\hat{\mathbf{P}}_k)$. This means that $\dim \text{span}(\hat{\mathbf{X}}_3, \hat{\mathbf{X}}_5, \hat{\mathbf{X}}_6, \hat{\mathbf{X}}_7)$ is at least 3. Now, since $\hat{\lambda}_{l3} = \hat{\lambda}_{l5} = \hat{\lambda}_{l6} = \hat{\lambda}_{l7} = 0$, by Lemma 3.8(ii), we can say $\text{span}(\hat{\mathbf{X}}_3, \hat{\mathbf{X}}_5, \hat{\mathbf{X}}_6, \hat{\mathbf{X}}_7) \subseteq \mathcal{N}(\hat{\mathbf{P}}_l)$. Since $\text{span}(\hat{\mathbf{X}}_3, \hat{\mathbf{X}}_5, \hat{\mathbf{X}}_6, \hat{\mathbf{X}}_7)$ is either 3D or 4D, this means that $\mathcal{R}\text{ank}(\hat{\mathbf{P}}_l) \leq 1$. As we chose l to be any arbitrary view, this means that $\mathcal{R}\text{ank}(\hat{\mathbf{P}}_i) \leq 1$ for all i. But according to Lemma 3.11 this cannot happen, and we get a contradiction. □

Lemma 3.13. *Assume that (D1, D2) and (G1, G2) hold, and denote by n_i the number of nonzero elements of the i-th row of $\hat{\mathbf{\Lambda}}$. If for some view r we have $n_r \geq n - 1$ and $n_i = 1$ for all $i \neq r$, then the matrix $\hat{\mathbf{\Lambda}}$ has to be* cross-shaped *(see Definition 3.2).*

Proof. As $m \geq 2$, there exist at least another view k other than r. Assume the (only) nonzero element on the k-th row of $\hat{\mathbf{\Lambda}}$ is $\hat{\lambda}_{kc}$. We will show that for any view l other that r and k (if there is any) the only nonzero element in the l-th row of $\hat{\mathbf{\Lambda}}$ has to be $\hat{\lambda}_{lc}$.

Consider a view l other than r and k. As $n \geq 8$, and there is exactly one nonzero element in the k-th row of $\hat{\mathbf{\Lambda}}$, one nonzero element in the l-th row of $\hat{\mathbf{\Lambda}}$, and at most one zero element in the r-th row of $\hat{\mathbf{\Lambda}}$, one can find three distinct indices j_1, j_2, j_3 such that $\hat{\lambda}_{rj_1} \neq 0, \hat{\lambda}_{rj_2} \neq 0, \hat{\lambda}_{rj_3} \neq 0$, $\hat{\lambda}_{kj_1} = \hat{\lambda}_{kj_2} = \hat{\lambda}_{kj_3} = 0$ and $\hat{\lambda}_{lj_1} = \hat{\lambda}_{lj_2} = \hat{\lambda}_{lj_3} = 0$. We have

$$\begin{aligned} \hat{\mathbf{P}}_r \,\text{span}(\hat{\mathbf{X}}_{j_1}, \hat{\mathbf{X}}_{j_2}, \hat{\mathbf{X}}_{j_3}) &= \text{span}(\hat{\mathbf{P}}_r\hat{\mathbf{X}}_{j_1}, \hat{\mathbf{P}}_r\hat{\mathbf{X}}_{j_2}, \hat{\mathbf{P}}_r\hat{\mathbf{X}}_{j_3}) \\ &= \text{span}(\mathbf{P}_r\mathbf{X}_{j_1}, \mathbf{P}_r\mathbf{X}_{j_2}, \mathbf{P}_r\mathbf{X}_{j_3}). \end{aligned} \tag{3.22}$$

where the product $\hat{\mathbf{P}}_r \,\text{span}(\hat{\mathbf{X}}_{j_1}, \hat{\mathbf{X}}_{j_2}, \hat{\mathbf{X}}_{j_3})$ represents the set created by multiplying $\hat{\mathbf{P}}_r$ by each element of the subspace $\text{span}(\hat{\mathbf{X}}_{j_1}, \hat{\mathbf{X}}_{j_2}, \hat{\mathbf{X}}_{j_3})$. The last equality in (3.22) comes

from (3.17) and the fact that $\hat{\lambda}_{rj_1}$, $\hat{\lambda}_{rj_2}$ and $\hat{\lambda}_{rj_3}$ are nonzero. According to (G2-5), span($\mathrm{P}_r\mathbf{X}_{j_1}, \mathrm{P}_r\mathbf{X}_{j_2}, \mathrm{P}_r\mathbf{X}_{j_3}$) is 3D, and therefore, (3.22) suggests that span($\hat{\mathbf{X}}_{j_1}, \hat{\mathbf{X}}_{j_2}, \hat{\mathbf{X}}_{j_3}$) has to be also 3D. From $\hat{\lambda}_{kj_1} = \hat{\lambda}_{kj_2} = \hat{\lambda}_{kj_3} = 0$ and $\hat{\lambda}_{lj_1} = \hat{\lambda}_{lj_2} = \hat{\lambda}_{lj_3} = 0$ respectively we conclude that span($\hat{\mathbf{X}}_{j_1}, \hat{\mathbf{X}}_{j_2}, \hat{\mathbf{X}}_{j_3}$) $\in \mathcal{N}(\hat{\mathrm{P}}_k)$ and span($\hat{\mathbf{X}}_{j_1}, \hat{\mathbf{X}}_{j_2}, \hat{\mathbf{X}}_{j_3}$) $\in \mathcal{N}(\hat{\mathrm{P}}_l)$ (Lemma 3.8(ii)). As $\hat{\mathrm{P}}_k$ and $\hat{\mathrm{P}}_l$ are both nonzero (Lemma 3.8(i)), and hence, of rank one or more, and their null-spaces include a the 3D subspace span($\hat{\mathbf{X}}_{j_1}, \hat{\mathbf{X}}_{j_2}, \hat{\mathbf{X}}_{j_3}$), it follows that $\mathcal{N}(\hat{\mathrm{P}}_k) = \mathcal{N}(\hat{\mathrm{P}}_l) = \mathrm{span}(\hat{\mathbf{X}}_{j_1}, \hat{\mathbf{X}}_{j_2}, \hat{\mathbf{X}}_{j_3})$. This means that for any j, $\hat{\lambda}_{kj}$ and $\hat{\lambda}_{lj}$ are either both nonzero or both zero. As $\hat{\lambda}_{kc} \neq 0$, we must have $\hat{\lambda}_{lc} \neq 0$. Since this is true for any view l other than k and r, we can say that for all views $i \neq r$, the (only) nonzero element is in the c-th column of $\hat{\lambda}_{ic}$.

By the assumption of the lemma, the r-th row of $\hat{\Lambda}$ can have either no zero element or one zero element. If it does have one zero element, it has to be $\hat{\lambda}_{rc}$, as otherwise, if $\hat{\lambda}_{rc'} = 0$ for some $c' \neq c$, the c'-th column of $\hat{\Lambda}$ would be zero, violating (D2). Now, we have the case where all elements of $\hat{\Lambda}$ are zero except those in the r-th row or the c-th column, and among the elements in the r-th row or the c-th column, all are nonzero except possibly $\hat{\lambda}_{rc}$. This means that $\hat{\Lambda}$ is cross-shaped. □

Lemma 3.14. *Under (D1-D3), (G1-G3) there exist two views i and k such that* $\mathcal{R}\mathrm{ank}(\hat{\mathrm{P}}_i) \geq 2$, $\mathcal{R}\mathrm{ank}(\hat{\mathrm{P}}_k) \geq 2$ *and* $\mathrm{stack}(\hat{\mathrm{P}}_i, \hat{\mathrm{P}}_k)$ *has rank 4.*

Proof. Lemma 3.12 says that under our assumptions, there exists at least one estimated camera matrix $\hat{\mathrm{P}}_i$ of rank 3. With a possible re-indexing of the views, we can assume that $\mathcal{R}\mathrm{ank}(\hat{\mathrm{P}}_1) = 3$. Now we consider two cases. The first case is when among $\hat{\lambda}_{11}, \hat{\lambda}_{12}, \ldots, \hat{\lambda}_{1n}$ there exists at most one zero element. In this case there must be at least another view k with two or more nonzero elements in the k-th row of $\hat{\Lambda}$, as otherwise, according to Lemma 3.13, $\hat{\Lambda}$ would be cross-shaped, violating (D3). By Lemma 3.8(iii) then we have $\mathcal{R}\mathrm{ank}(\hat{\mathrm{P}}_k) \geq 2$. Because at least for $n-1$ point indices j we have $\hat{\lambda}_{1j} \neq 0$, and thus $(\hat{\lambda}_{1j}, \hat{\lambda}_{kj})^T \neq \mathbf{0}$, from Lemma 3.10 we know that the row space of $\hat{\mathrm{P}}_k$ cannot be a subset of the row space of $\hat{\mathrm{P}}_1$. Therefore, as $\mathcal{R}\mathrm{ank}(\hat{\mathrm{P}}_1) = 3$ we have $\mathcal{R}\mathrm{ank}\begin{bmatrix}\hat{\mathrm{P}}_1 \\ \hat{\mathrm{P}}_k\end{bmatrix} = 4$. This along with the fact that $\mathcal{R}\mathrm{ank}(\hat{\mathrm{P}}_1) = 3 \geq 2$ and $\mathcal{R}\mathrm{ank}(\hat{\mathrm{P}}_k) \geq 2$ completes the proof for this case.

The only case left is when there are at least two zero elements among $\hat{\lambda}_{11}, \hat{\lambda}_{12}, \ldots, \hat{\lambda}_{1n}$. By a possible re-indexing we can assume that $\hat{\lambda}_{11} = \hat{\lambda}_{12} = 0$. This means that $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ must be equal up to scale (Lemma 3.8(v)). According to (D2), there must be at least one view k for which $\hat{\lambda}_{k1} \neq 0$. As $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ are nonzero (Lemma 3.8(i)) and equal up to scale, $\hat{\lambda}_{k1} \neq 0$ implies $\hat{\lambda}_{k2} \neq 0$. This means that $\mathcal{R}\mathrm{ank}(\hat{\mathrm{P}}_k) \geq 2$ (Lemma 3.8(iii)). As we have $\mathcal{R}\mathrm{ank}(\hat{\mathrm{P}}_1) = 3$, $\hat{\lambda}_{11} = 0$ and $\hat{\lambda}_{k1} \neq 0$, by Lemma 3.8(iv) we get $\mathcal{R}\mathrm{ank}\begin{bmatrix}\hat{\mathrm{P}}_1 \\ \hat{\mathrm{P}}_k\end{bmatrix} = 4$. This completes the proof as we also have $\mathcal{R}\mathrm{ank}(\hat{\mathrm{P}}_1) \geq 2$ and $\mathcal{R}\mathrm{ank}(\hat{\mathrm{P}}_k) \geq 2$. □

Lemma 3.7 now follows directly from Lemmas 3.14 and 3.9.

3.2.3 Projective Equivalence for Two Views

The main result of this section is the following lemma:

Lemma 3.15. *Under (G1, G2, G4) and (D2), If the fundamental matrix $\mathcal{F}(\hat{\mathtt{P}}_k, \hat{\mathtt{P}}_l)$ is nonzero for two views k and l, then the two configurations $(\mathtt{P}_k, \mathtt{P}_l, \{\mathbf{X}_j\})$ and $(\hat{\mathtt{P}}_k, \hat{\mathtt{P}}_l, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent.*

Proof. For simplicity, we take $k = 1$ and $l = 2$. The other cases follow by relabeling the views. For each j we have $\hat{\mathtt{P}}_1\hat{\mathbf{X}}_j = \hat{\lambda}_{1j}\mathtt{P}_1\mathbf{X}_j$ and $\hat{\mathtt{P}}_2\hat{\mathbf{X}}_j = \hat{\lambda}_{2j}\mathtt{P}_2\mathbf{X}_j$, or equivalently

$$\begin{bmatrix} \hat{\mathtt{P}}_1, & \mathtt{P}_1\mathbf{X}_j & \mathbf{0} \\ \hat{\mathtt{P}}_2, & \mathbf{0} & \mathtt{P}_2\mathbf{X}_j \end{bmatrix} \begin{pmatrix} -\hat{\mathbf{X}}_j \\ \hat{\lambda}_{1j} \\ \hat{\lambda}_{2j} \end{pmatrix} = 0, \qquad j = 1, 2, \ldots, n. \tag{3.23}$$

As, $\hat{\mathbf{X}}_j \neq 0$ (Lemma 3.8(i)) the 6×6 matrix on the left hand side of (3.23) has a nontrivial null space, and hence, a vanishing determinant. Define the function $\mathcal{S}\colon \mathbb{R}^4 \to \mathbb{R}$ as

$$\mathcal{S}(\mathbf{X}) \stackrel{\text{def}}{=} \det \begin{bmatrix} \hat{\mathtt{P}}_1, & \mathtt{P}_1\mathbf{X} & \mathbf{0} \\ \hat{\mathtt{P}}_2, & \mathbf{0} & \mathtt{P}_2\mathbf{X} \end{bmatrix}. \tag{3.24}$$

Using the properties of the determinant and Definition 3.1 of the fundamental matrix, the above can be written as [Hartley and Zisserman, 2004, Sect. 17.1]:

$$\mathcal{S}(\mathbf{X}) = \mathbf{X}^T\mathtt{P}_1^T\hat{\mathtt{F}}_{12}\mathtt{P}_2\mathbf{X} = \mathbf{X}^T\mathtt{S}\,\mathbf{X} \tag{3.25}$$

where $\hat{\mathtt{F}}_{12} \stackrel{\text{def}}{=} \mathcal{F}(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_2)$ is the fundamental matrix of $\hat{\mathtt{P}}_1$ and $\hat{\mathtt{P}}_2$ as defined in Definition 3.1, and $\mathtt{S} \stackrel{\text{def}}{=} \mathtt{P}_1^T\hat{\mathtt{F}}_{12}\mathtt{P}_2$. We shall show that $\mathcal{S}$ has to be identically zero (that is $\mathcal{S}(\mathbf{X}) = 0$ for all $\mathbf{X}$). To see this, assume that $\mathcal{S}$ is not identically zero. Then the equation

$$\mathcal{S}(\mathbf{X}) = \mathbf{X}^T\mathtt{S}\,\mathbf{X} = 0 \tag{3.26}$$

defines a quadric surface. From (3.23) we know $\mathcal{S}(\mathbf{X}_j) = 0$ for all $j = 1, 2, \ldots, n$ and therefore all the points $\{\mathbf{X}_j\}$ lie on this quadric surface. Also, for any pair of nonzero vectors $\mathbf{C}_1 \in \mathcal{N}(\mathtt{P}_1)$ and $\mathbf{C}_2 \in \mathcal{N}(\mathtt{P}_2)$ (camera centres) one can easily check that $\mathcal{S}(\mathbf{C}_1) = \mathcal{S}(\mathbf{C}_2) = 0$ and therefore, $\mathbf{C}_1$ and $\mathbf{C}_2$ also lie on the quadric surface.

As the fundamental matrix $\hat{\mathtt{F}}_{12} \stackrel{\text{def}}{=} \mathcal{F}(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_2)$ is rank deficient [Hartley and Zisserman, 2004], we can have a nonzero vector $\mathbf{v} \in \mathcal{N}(\hat{\mathtt{F}}_{12})$. Since $\mathtt{P}_2$ has full row rank by (G1), we can write $\mathbf{v} = \mathtt{P}_2\mathbf{Y}$ for some $\mathbf{Y} \in \mathbb{R}^4$. Then, by taking a nonzero vector

$\mathbf{C}_2 \in \mathcal{N}(\mathtt{P}_2)$, one can easily check that for any two scalars α and β we have

$$\begin{aligned}
\mathcal{S}(\alpha\mathbf{Y}+\beta\mathbf{C}_2) &= (\alpha\mathbf{Y}+\beta\mathbf{C}_2)^T(\mathtt{P}_1^T\hat{\mathtt{F}}_{12}\mathtt{P}_2)(\alpha\mathbf{Y}+\beta\mathbf{C}_2), && (3.27)\\
&= (\alpha\mathbf{Y}+\beta\mathbf{C}_2)^T\mathtt{P}_1^T(\alpha\hat{\mathtt{F}}_{12}\mathtt{P}_2\mathbf{Y}+\beta\hat{\mathtt{F}}_{12}\mathtt{P}_2\mathbf{C}_2), && (3.28)\\
&= (\alpha\mathbf{Y}+\beta\mathbf{C}_2)^T\mathtt{P}_1^T(\alpha\hat{\mathtt{F}}_{12}\mathbf{v}+\beta\hat{\mathtt{F}}_{12}\cdot\mathbf{0}), && (3.29)\\
&= (\alpha\mathbf{Y}+\beta\mathbf{C}_2)^T\mathtt{P}_1^T(\alpha\cdot\mathbf{0}+\mathbf{0}). && (3.30)\\
&= 0 && (3.31)
\end{aligned}$$

This, plus the fact that $\mathbf{Y}$ and $\mathbf{C}_2$ are linearly independent (as $\mathbf{C}_2 \neq \mathbf{0}$ and $\mathtt{P}_2\mathbf{C}_2 = \mathbf{0} \neq \mathbf{v} = \mathtt{P}_2\mathbf{Y}$), implies that the quadric surface $\mathcal{S}(\mathbf{X}) = 0$ contains a projective line and hence is *ruled*.

Now, we have the case that the nonzero vectors $\mathbf{C}_1 \in \mathcal{N}(\mathtt{P}_1)$ and $\mathbf{C}_2 \in \mathcal{N}(\mathtt{P}_2)$ (camera centres) plus the points $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ all lie on a (proper or degenerate) ruled quadric surface represented by (3.26). This contradicts the genericity condition (G4). This only leaves the possibility that $\mathcal{S}(\mathbf{X})$ is identically zero or equivalently, $\mathtt{S} + \mathtt{S}^T = 0$, that is

$$\mathtt{P}_1^T\hat{\mathtt{F}}_{12}\mathtt{P}_2 + \mathtt{P}_2^T\hat{\mathtt{F}}_{12}^T\mathtt{P}_1 = 0 \tag{3.32}$$

Therefore, according to Lemma 3.3 (whose conditions hold by (G1) and (G2-2)) the matrix $\hat{\mathtt{F}}_{12} = \mathcal{F}(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_2)$ is a multiple of $\mathcal{F}(\mathtt{P}_1, \mathtt{P}_2)$. As we have assumed that $\mathcal{F}(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_2) \neq 0$, and having (G1) and (G2-2), by Lemma 3.2 we know that $(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_2)$ is projectively equivalent to $(\mathtt{P}_1, \mathtt{P}_2)$ that is

$$\hat{\mathtt{P}}_1 = \tau_1\mathtt{P}_1\mathtt{H} \tag{3.33}$$

$$\hat{\mathtt{P}}_2 = \tau_2\mathtt{P}_2\mathtt{H} \tag{3.34}$$

for a non-singular matrix $\mathtt{H}$ and nonzero scalars τ_1 and τ_2. Now, for any point $\mathbf{X}_j$, the relation (3.17), that is $\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j$, gives

$$\tau_1\mathtt{P}_1\mathtt{H}\hat{\mathbf{X}}_j = \hat{\mathtt{P}}_1\hat{\mathbf{X}}_j = \hat{\lambda}_{1j}\mathtt{P}_1\mathbf{X}_j, \tag{3.35}$$

$$\tau_2\mathtt{P}_2\mathtt{H}\hat{\mathbf{X}}_j = \hat{\mathtt{P}}_2\hat{\mathbf{X}}_j = \hat{\lambda}_{2j}\mathtt{P}_2\mathbf{X}_j. \tag{3.36}$$

It follows that

$$\mathtt{P}_1(\mathtt{H}\hat{\mathbf{X}}_j) = \frac{\hat{\lambda}_{1j}}{\tau_1}\mathtt{P}_1\mathbf{X}_j, \tag{3.37}$$

$$\mathtt{P}_2(\mathtt{H}\hat{\mathbf{X}}_j) = \frac{\hat{\lambda}_{2j}}{\tau_2}\mathtt{P}_2\mathbf{X}_j. \tag{3.38}$$

Having the genericity conditions (G1) and (G2-4), one can apply the Triangulation Lemma 3.4 to prove that $\mathtt{H}\hat{\mathbf{X}}_j$ is equal to $\mathbf{X}_j$ up to a nonzero scaling factor, that is

$\mathtt{H}\hat{\mathbf{X}}_j = \nu_j \mathbf{X}_j$ or

$$\hat{\mathbf{X}}_j = \nu_j \mathtt{H}^{-1} \mathbf{X}_j. \tag{3.39}$$

Notice that ν_j cannot be zero as $\hat{\mathbf{X}}_j \neq 0$ (from Lemma 3.8(i)). From (3.33), (3.34) and (3.39) it follows that $(\mathtt{P}_1, \mathtt{P}_2, \{\mathbf{X}_j\})$ and $(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_2, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent. □

3.2.4 Projective Equivalence for All Views

Lemma 3.16. *Under (G1-G4) and (D1, D2), if for two views k and l the two configurations $(\mathtt{P}_k, \mathtt{P}_l, \{\mathbf{X}_j\})$ and $(\hat{\mathtt{P}}_k, \hat{\mathtt{P}}_l, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent, then for the whole camera matrices and points, the configurations $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent.*

Proof. For convenience, take $k = 1$ and $l = 2$ (the other cases follow by relabeling the views). First of all, notice that as $(\mathtt{P}_1, \mathtt{P}_2, \{\mathbf{X}_j\})$ and $(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_2, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent, we have

$$\hat{\mathtt{P}}_1 = \tau_1 \mathtt{P}_1 \mathtt{H}, \quad \hat{\mathtt{P}}_2 = \tau_2 \mathtt{P}_2 \mathtt{H}, \tag{3.40}$$

$$\hat{\mathbf{X}}_j = \nu_j \mathtt{H}^{-1} \mathbf{X}_j, \quad j = 1, 2, \ldots, n, \tag{3.41}$$

for an invertible matrix $\mathtt{H}$ and nonzero scalars τ_1, τ_2 and $\nu_1, \ldots, \nu_n$. From (G2) and (3.41), we can say that for any four distinct point indices $j_1, \ldots, j_4$, the points $\hat{\mathbf{X}}_{j_1}$, $\hat{\mathbf{X}}_{j_2}$, $\hat{\mathbf{X}}_{j_3}$ and $\hat{\mathbf{X}}_{j_4}$ span a 4-dimensional space. Therefore, for each view i at most 3 depth scalars $\hat{\lambda}_{ij}$ can be zero, as otherwise, if we have $\hat{\lambda}_{ij_1} = \hat{\lambda}_{ij_2} = \hat{\lambda}_{ij_3} = \hat{\lambda}_{ij_4} = 0$ it means that $\hat{\mathbf{X}}_{j_1}, \hat{\mathbf{X}}_{j_2}, \hat{\mathbf{X}}_{j_3}, \hat{\mathbf{X}}_{j_4} \in \mathcal{N}(\hat{\mathtt{P}}_i)$ (Lemma 3.8(ii)). This, however, implies $\hat{\mathtt{P}}_i = 0$ contradicting Lemma 3.8(i).

Now, since we know that for each view i we have at most 3 zero depths $\hat{\lambda}_{ij}$, from $n \geq 8$, we know that there are more than 3 nonzero depths $\hat{\lambda}_{ij}$ at each row i. Therefore, according to Lemma 3.8(iii), we can say that $\mathcal{R}\text{ank}(\hat{\mathtt{P}}_i) = 3$ for all i.

Now, notice that as $(\mathtt{P}_1, \mathtt{P}_2, \{\mathbf{X}_j\})$ and $(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_2, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent, from Lemma 2.2 (whose conditions hold by (G1, G2) and their consequences (G2-1) and (G2-3)) we have $\hat{\lambda}_{1j} \neq 0$ and $\hat{\lambda}_{2j} \neq 0$ for all $j = 1, 2, \ldots, n$. Now, for any view $k \geq 3$, consider the pair of matrices $(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_k)$. We have $\mathcal{R}\text{ank}(\hat{\mathtt{P}}_k) = \mathcal{R}\text{ank}(\hat{\mathtt{P}}_1) = 3$ and moreover, the vector $(\hat{\lambda}_{1j}, \hat{\lambda}_{kj})$ is nonzero for all j. Therefore, by Lemma 3.10 we get $\mathcal{R}\text{ank}\,(\text{stack}(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_k)) = 4$. After that, by Lemma 3.14 it follows that the fundamental matrix $\mathcal{F}(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_k)$ is nonzero. Then by Lemma 3.15 we can say that $(\mathtt{P}_1, \mathtt{P}_k, \{\mathbf{X}_j\})$ and $(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_k, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent. Therefore,

$$\hat{\mathtt{P}}_1 = \tau_1' \mathtt{P}_1 \mathtt{G}, \quad \hat{\mathtt{P}}_k = \tau_k' \mathtt{P}_k \mathtt{G}, \tag{3.42}$$

$$\hat{\mathbf{X}}_j = \nu_j' \mathtt{G}^{-1} \mathbf{X}_j, \quad j = 1, 2, \ldots, n, \tag{3.43}$$

for an invertible matrix $\mathtt{G}$ and nonzero scalars τ_1', τ_k' and $\nu_1', \nu_2', \ldots, \nu_n'$. Now, we can apply Lemma 2.1 for equations (3.41) and (3.43). Notice that according to (G2) every four points among $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^4$ are linearly independent. The reader can check that this plus the fact that $n \geq 8$ implies conditions (P1) and (P2) in Lemma

2.1 for $r = 4$. By applying Lemma 2.1 we get $\mathtt{G}^{-1} = \mathtt{H}^{-1}/\alpha$ (or $\mathtt{G} = \alpha\mathtt{H}$) and $v'_j = \alpha v_j$ for some nonzero scalar α. This, plus (3.40) and (3.42) gives $\tau'_1 = \tau_1/\alpha$. By using and $\tau_1 = \alpha\tau'_1$, and defining $\tau_k \stackrel{\text{def}}{=} \alpha\tau'_k$ we have

$$\hat{\mathtt{P}}_1 = \tau_1 \mathtt{P}_1 \mathtt{H}, \quad \hat{\mathtt{P}}_k = \tau_k \mathtt{P}_k \mathtt{H}, \tag{3.44}$$

$$\hat{\mathbf{X}}_j = v_j \mathtt{H}^{-1} \mathbf{X}_j, \quad j = 1, 2, \ldots, n, \tag{3.45}$$

Since the above is true for all $k = 3, \ldots, n$, and also for $k = 2$ by (3.40), we conclude that the two configurations $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent.

□

3.2.5 Minimality of (D1-D3) and Cross-shaped Configurations

From depth assumptions (D1-D3) we see that in order to get the projective reconstruction working we require that none of the rows or columns of the depth matrix $\hat{\Lambda} = [\hat{\lambda}_{ij}]$ are zero and that $\hat{\Lambda}$ is not cross-shaped. One might wonder whether projective reconstruction is possible under a priori weaker conditions on the estimated depth matrix. For example, what happens if we just require that the matrix has no zero rows and no zero columns.

In this section we shall show that, in some specific sense, (D1-D3) is a minimal assumption for projective reconstruction. However, by this we do not mean that it is the weakest possible constraint that guarantees the uniqueness of projective reconstruction up to projectivity. But, it is minimal in the sense that if any of (D1), (D2) or (D3) is relaxed completely, and no extra conditions are added, the resulting constraints cannot rule out false solutions to projective reconstruction. This shows that the false solutions to the factorization problem $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathtt{P}}\hat{\mathtt{X}}$ are not limited to the trivial cases of having depth matrices with some zero rows or columns.

The necessity of (D1) is obvious, as, for example, if we allow the k-th row of $\hat{\Lambda}$ to be zero, then we can set $\hat{\lambda}_{k1} = \hat{\lambda}_{k2} = \cdots = \hat{\lambda}_{kn} = 0$ and $\hat{\mathtt{P}}_k = 0$, as it satisfies $\hat{\mathtt{P}}_k \hat{\mathbf{X}}_j = \hat{\lambda}_{kj} \mathbf{x}_{kj}$ for all j. For the rest of variables we can have $\hat{\mathbf{X}}_j = \mathbf{X}_j$, $\hat{\mathtt{P}}_i = \mathtt{P}_i$ and $\hat{\lambda}_{ij} = \lambda_{ij}$ for all i, j where $i \neq k$. Similarly, if we relax (D2) by allowing the l-th column of $\hat{\Lambda}$ to be nonzero, we can have a configuration in which $\mathbf{X}_l = \mathbf{0}$.

The more difficult job is to show that the relaxation of (D3) can allow a projectively non-equivalent setup. Relaxing this condition means that $\hat{\Lambda}$ is *cross-shaped*. We show that in this case for any configuration of the true camera matrices $\mathtt{P}_i$, points $\mathbf{X}_j$ and depths λ_{ij}, we can find a non-equivalent setup $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\}, \{\hat{\lambda}_{ij}\})$.

Consider m arbitrary 3×4 projection matrices $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m$ and an arbitrary set of points $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^4$ (with m and n arbitrary), giving the image points $\mathbf{x}_{ij}$ through the relation $\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j$. Now, for any arbitrary view r and point index c

we can take

$$\hat{\lambda}_{ic} = \lambda_{ic}, \quad i = 1,2,\ldots,m, \tag{3.46}$$

$$\hat{\lambda}_{rj} = \lambda_{rj}, \quad j = 1,2,\ldots,n, \tag{3.47}$$

$$\hat{\lambda}_{ij} = 0, \quad i \neq r,\ j \neq c. \tag{3.48}$$

$$\hat{\mathtt{P}}_r = \mathtt{P}_r, \tag{3.49}$$

$$\hat{\mathtt{P}}_i = \mathtt{P}_i \mathbf{X}_c \bar{\mathbf{C}}_r^T \quad i \neq r \tag{3.50}$$

$$\hat{\mathbf{X}}_c = (\mathtt{I} - \bar{\mathbf{C}}_r \bar{\mathbf{C}}_r^T)\,\mathbf{X}_c + \bar{\mathbf{C}}_r, \tag{3.51}$$

$$\hat{\mathbf{X}}_j = (\mathtt{I} - \bar{\mathbf{C}}_r \bar{\mathbf{C}}_r^T)\,\mathbf{X}_j \quad j \neq c. \tag{3.52}$$

where $\bar{\mathbf{C}}_r$ is a unit vector in the null-space of $\mathtt{P}_r$. Notice that the matrix $\mathtt{I} - \bar{\mathbf{C}}_r \bar{\mathbf{C}}_r^T$ is the orthogonal projection onto the row space of $\mathtt{P}_r$. Now, it can be easily checked that

$$\hat{\mathtt{P}}_i \hat{\mathbf{X}}_j = \mathtt{P}_i \mathbf{X}_j = \lambda_{ij} \mathbf{x}_{ij} = \hat{\lambda}_{ij} \mathbf{x}_{ij} \quad \text{if } i = r \text{ or } j = c \tag{3.53}$$

$$\hat{\mathtt{P}}_i \hat{\mathbf{X}}_j = \mathbf{0} = 0 \cdot \mathbf{x}_{ij} = \hat{\lambda}_{ij} \mathbf{x}_{ij} \quad \text{if } i \neq r \text{ and } j \neq c \tag{3.54}$$

Notice that to derive (3.53) one has to check three cases separately: first $i = r, j = c$, second $i = r, j \neq c$, and third $i \neq r, j = c$. You can see that with this choice we have $\hat{\mathtt{P}}_i \hat{\mathbf{X}}_j = \hat{\lambda}_{ij} \mathbf{x}_{ij}$ for all i and j. It is obvious that $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ is not generally projectively equivalent to $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$, as, for example, for any $i \neq r$ we have $\mathcal{R}\text{ank}(\hat{\mathtt{P}}_i) = 1$ regardless of the value of $\mathtt{P}_i$. From (3.46-3.48) it follows that

$$\hat{\Lambda} = \begin{bmatrix} 0 & \mathbf{1}_{r-1} & 0 \\ \mathbf{1}_{c-1}^T & 1 & \mathbf{1}_{n-c}^T \\ 0 & \mathbf{1}_{m-r} & 0 \end{bmatrix} \circ \Lambda \tag{3.55}$$

where the zero matrices denoted by 0 are of compatible size and $\circ$ denotes the Hadamard (element-wise) product. This shows that $\hat{\Lambda} = [\hat{\lambda}_{ij}]$ is a nonzero-centred cross-shaped matrix centred at (r,c). An example of such a configuration has been illustrated in Fig. 3.3 for $r = 1, c = 1$.

One can observe that instead of (3.51) we can give any arbitrary value to $\hat{\mathbf{X}}_c$, provided that it is not perpendicular to $\mathbf{C}_r$, and still get a setup with a cross-shaped depth matrix. Especially, we leave it to the reader to check that by taking $\hat{\mathbf{X}}_c$ equal to $\bar{\mathbf{C}}_r$ instead of $(\mathtt{I} - \bar{\mathbf{C}}_r \bar{\mathbf{C}}_r^T)\,\mathbf{X}_c + \bar{\mathbf{C}}_r$ in (3.51), we have a setup in which the depth matrix $\hat{\Lambda}$ is arranged as (3.46-3.48) with the exception that the central element $\hat{\lambda}_{rc}$ is zero, that is

$$\hat{\Lambda} = \begin{bmatrix} 0 & \mathbf{1}_{r-1} & 0 \\ \mathbf{1}_{c-1}^T & 0 & \mathbf{1}_{n-c}^T \\ 0 & \mathbf{1}_{m-r} & 0 \end{bmatrix} \circ \Lambda. \tag{3.56}$$

This means that $\hat{\Lambda}$ is a zero-centred cross-shaped matrix. Obviously for any pair of vectors $\boldsymbol{\tau} \in \mathbb{R}^m$ and $\boldsymbol{\nu} \in \mathbb{R}^n$ with all nonzero entries, we can find a new configuration with $\hat{\Lambda}' = \mathrm{diag}(\boldsymbol{\tau})\,\hat{\Lambda}\,\mathrm{diag}(\boldsymbol{\nu})$, $\hat{\mathtt{P}}_i' = \tau_i \hat{\mathtt{P}}_i$ and $\hat{\mathbf{X}}_j' = \nu_j \hat{\mathbf{X}}_j$, satisfying $\hat{\mathtt{P}}_i' \hat{\mathbf{X}}_j' = \hat{\lambda}_{ij}' \mathbf{x}_{ij}$

	$=\mathrm{R}_1\mathbf{X}_1+\bar{\mathbf{C}}_1$	$\hat{\mathbf{X}}_j=\mathrm{R}_1\mathbf{X}_j$				
	$\hat{\mathbf{X}}_1$	$\hat{\mathbf{X}}_2$	$\hat{\mathbf{X}}_3$	$\hat{\mathbf{X}}_4$	$\hat{\mathbf{X}}_5$	$\hat{\mathbf{X}}_6$
$\mathrm{P}_1=\hat{\mathrm{P}}_1$	λ_{11}	λ_2	λ_3	λ_4	λ_5	λ_6
$\mathrm{P}_2\mathbf{X}_1\bar{\mathbf{C}}_1^T=\hat{\mathrm{P}}_2$	λ_{21}	0	0	0	0	0
$\mathrm{P}_3\mathbf{X}_1\bar{\mathbf{C}}_1^T=\hat{\mathrm{P}}_3$	λ_{31}	0	0	0	0	0
$\mathrm{P}_4\mathbf{X}_1\bar{\mathbf{C}}_1^T=\hat{\mathrm{P}}_4$	λ_{41}	0	0	0	0	0
$\mathrm{P}_5\mathbf{X}_1\bar{\mathbf{C}}_1^T=\hat{\mathrm{P}}_5$	λ_{51}	0	0	0	0	0
	$\hat{\Lambda}$					

Figure 3.3: An example of a cross-shaped configuration where the cross is centred at (1,1), that is $r = 1$ and $c = 1$, with 6 points and 5 camera matrices. In the above, $\bar{\mathbf{C}}_1$ is a unit-length vector in the null space of P_1 and $\mathrm{R}_1 = (\mathrm{I} - \bar{\mathbf{C}}_1\bar{\mathbf{C}}_1^T)$ is the *orthogonal projection* into the row space of P_1. One can check that $\hat{\mathrm{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\lambda}_{ij}(\frac{1}{\lambda_{ij}}\mathrm{P}_i\mathbf{X}_j)$ for all i and j, or equivalently $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathrm{P}}\,\hat{\mathbf{X}}$.

(as $(\tau_i\hat{\mathrm{P}}_i)(\nu_j\hat{\mathbf{X}}_j) = (\tau_i\nu_j\hat{\lambda}_{ij})\,\mathbf{x}_{ij}$). Notice that, according to the above discussion, both configurations (3.55) and (3.56) can be obtained for any configuration of m views and n points, and for any choice of r and c. We also know from Lemma 3.6 that any $m\times n$ cross-shaped matrix is diagonally equivalent to either (3.55) or (3.56) for some choice of r and c. Putting all these together we get the following lemma.

Lemma 3.17. *Consider any configuration of m camera matrices and n points $(\{\mathrm{P}_i\},\{\mathbf{X}_j\})$ giving the image points $\{\mathbf{x}_{ij}\}$ through the relations $\lambda_{ij}\mathbf{x}_{ij} = \mathrm{P}_i\mathbf{X}_j$ with nonzero scalars $\lambda_{ij} \neq 0$. Then for* any *cross-shaped matrix $\hat{\Lambda} = [\hat{\lambda}_{ij}]$, there exists a configuration $(\{\hat{\mathrm{P}}_i\},\{\hat{\mathbf{X}}_j\})$, such that the relation $\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathrm{P}}_i\hat{\mathbf{X}}_j$ holds for all $i = 1,\ldots,m$ and $j = 1,\ldots,n$.*

This lemma is used in the next session as a useful test for the assessment of depth constraints. It says that if a constraint allows *any* cross-shaped structure for the depth matrix, then it allows for a false solution.

3.3 The Constraint Space

In this section we will have a closer look at the depth constraints used in factorization-based projective reconstruction. Consider a set of $m \geq 2$ projection matrices $\mathrm{P}_1,\ldots,\mathrm{P}_m \in \mathbb{R}^{3\times4}$ and a set of $n \geq 8$ points $\mathbf{X}_1,\ldots,\mathbf{X}_n \in \mathbb{R}^4$, generically configured in the sense of (G1-G4) and projecting into a set of image points $\mathbf{x}_{ij} \in \mathbb{R}^3$ according to $\lambda_{ij}\mathbf{x}_{ij} = \mathrm{P}_i\mathbf{X}_j$. Given a constraint space $C \subseteq \mathbb{R}^{m\times n}$ we want to assess the solutions to the problem

$$\text{find}_{\hat{\Lambda},\,\hat{\mathrm{P}}_{3m\times4},\,\hat{\mathbf{X}}_{4\times n}}\ \text{s.t.}\ \ \hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathrm{P}}\,\hat{\mathbf{X}},\ \ \hat{\Lambda} \in C \tag{3.57}$$

in terms of whether $(\{\hat{\mathrm{P}}_i\},\{\hat{\mathbf{X}}_j\})$ is projectively equivalent to $(\{\mathrm{P}_i\},\{\mathbf{X}_j\})$, where $\hat{\mathrm{P}} = \text{stack}(\hat{\mathrm{P}}_1,\hat{\mathrm{P}}_2,\cdots,\hat{\mathrm{P}}_m)$, $\hat{\mathbf{X}} = [\hat{\mathbf{X}}_1\hat{\mathbf{X}}_2\cdots\hat{\mathbf{X}}_n]$ and $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathrm{P}}\,\hat{\mathbf{X}}$ represents all the relations $\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathrm{P}}_i\hat{\mathbf{X}}_j$ in matrix form, as described for (2.12) and (2.13). By $\hat{\mathrm{P}}_{3m\times4}$ and $\hat{\mathbf{X}}_{4\times n}$ we respectively mean $\hat{\mathrm{P}} \in \mathbb{R}^{3m\times4}$ and $\hat{\mathbf{X}} \in \mathbb{R}^{4\times n}$.

Notice that, it is not sufficient that every $\hat{\Lambda}$ in C satisfies depth assumptions (D1-D3). The constraint space must also be *inclusive*, that is, it must make possible the existence of $\{\hat{P}_i\}$ and $\{\hat{X}_j\}$ for which $\hat{\Lambda} \odot [x_{ij}] = \hat{P}\,\hat{X}$ holds for all i and j. In other words, it must guarantee that (3.57) has at least one solution. One can check that for any $\hat{\Lambda}$ diagonally equivalent to the true depth matrix Λ, there exists a setup $(\{\hat{P}_i\}, \{\hat{X}_j\})$, defined by $\hat{P}_i = \tau_i P_i$, $\hat{X}_j = \nu_j X_j$, which is projectively equivalent to $(\{P_i\}, \{X_j\})$ and satisfies the relation $\hat{\Lambda} \odot [x_{ij}] = \hat{P}\,\hat{X}$. Therefore, for (3.57) to have at least one solution, it is sufficient that the constraint space C allows at least one $\hat{\Lambda}$ which is diagonally equivalent to Λ. Actually, this requirement is also necessary, since, according to Lemma 2.2, if there exists a setup $(\{\hat{P}_i\}, \{\hat{X}_j\})$ projectively equivalent to $(\{P_i\}, \{X_j\})$ which satisfies the relations $\hat{\lambda}_{ij} x_{ij} = \hat{P}_i \hat{X}_j$, then $\hat{\Lambda}$ must be diagonally equivalent to Λ. As we do not know the true depths Λ beforehand, we would like the constraint $\hat{\Lambda} \in C$ to work for any initial value of depths Λ. Hence, we need it to allow at least one diagonally equivalent matrix for every depth matrix Λ whose entries are all nonzero. If we have some prior knowledge about the true depth matrix Λ in the form of $\Lambda \in P$ for some set $P \subseteq \mathbb{R}^{m\times n}$, the constraint is only required to allow at least one diagonally equivalent matrix for every depth matrix Λ in P. For example, in many applications it is known a priori that the true depths λ_{ij} are all positive. In such cases P is the set of $m\times n$ matrices with all positive elements. The concept of inclusiveness, therefore, can be defined formally as follows:

Definition 3.3. *Given a set $P \subseteq \mathbb{R}^{m\times n}$ representing our prior knowledge about the possible values of the true depth matrix ($\Lambda \in P$), the constraint space $C \subseteq \mathbb{R}^{m\times n}$ is called* inclusive *if for every $m\times n$ matrix $\Lambda \in P$, there exists at least one matrix $\hat{\Lambda} \in C$ which is diagonally equivalent to Λ.*

Definition 3.4. *The constraint space $C \subseteq \mathbb{R}^{m\times n}$ is called* uniquely inclusive *if for every $m\times n$ matrix $\Lambda \in P$, there exists* exactly *one matrix $\hat{\Lambda} \in C$ which is diagonally equivalent to Λ.*

Here, whenever we use the term inclusive without specifying P, we mean the general case of P being the set of all $m\times n$ matrices with no zero element. We will only consider one other case where P is the set of all $m\times n$ matrices with all positive elements.

In addition to inclusiveness as a necessary property for a constraint, it is desirable for a constraint to *exclude* false solutions. This property can be defined as follows:

Definition 3.5. *For $m\geq 2$ and $n\geq 8$, a constraint space $C \subseteq \mathbb{R}^{m\times n}$ is called* exclusive[3] *if every $\hat{\Lambda} \in C$ satisfies (D1-D3).*

Now, we can present a class of constraints under which solving problem (3.57) leads to projective reconstruction:

[3]In fact, the term exclusive might not be a precise term here, as (D1-D3) holding for all $\hat{\Lambda} \in C$ is just a sufficient condition for a constraint to exclude false solutions. While, according to Lemma 3.17, (D3) holding for all $\hat{\Lambda} \in C$ is necessary for ruling out false solutions, (D1) and (D2) holding for all members of C is not necessary for this purpose. This is because there might exist some $\hat{\Lambda} \in C$ for which (D1) or (D2) do not hold, but it is excluded by $\hat{\Lambda} \odot [x_{ij}] = \hat{P}\hat{X}$. This is why we said in Sect. 3.2.5 that (D1-D3) is minimal *in a specific sense.*

Definition 3.6. *Given integers $m \geq 2$ and $n \geq 8$, and a set $P \subseteq \mathbb{R}^{m\times n}$ representing our prior knowledge about the true depth matrix, we call the constraint space $C \subseteq \mathbb{R}^{m\times n}$ (uniquely)* reconstruction friendly *if it is both* exclusive *and (uniquely)* inclusive *with respect to P.*

We will apply the same terms (inclusive, exclusive, reconstruction friendly) to the constraints themselves (as relations), and what we mean is that the corresponding constraint space has the property. The following proposition follows from the discussion above and Theorem 3.1.

Proposition 3.18. *Consider a setup of $m \geq 2$ camera matrices and $n \geq 8$ points $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ generically configured in the sense of (G1-G4), and projecting into the image points $\{\mathbf{x}_{ij}\}$ according to $\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j$ with nonzero scalars λ_{ij}. If C is a reconstruction friendly constraint space, then problem* (3.57) *has at least one solution and for any solution $(\hat{\Lambda}, \hat{\mathtt{P}}, \hat{\mathtt{X}})$, the configuration $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ is projectively equivalent to $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$, where the matrices $\hat{\mathtt{P}}_i \in \mathbb{R}^{3\times 4}$ and the points $\hat{\mathbf{X}}_j \in \mathbb{R}^4$ come from $\hat{\mathtt{P}} = \mathrm{stack}(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_2, \cdots, \hat{\mathtt{P}}_m)$ and $\hat{\mathtt{X}} = [\hat{\mathbf{X}}_1\hat{\mathbf{X}}_2\cdots\hat{\mathbf{X}}_n]$. If C is uniquely reconstruction friendly, then there is a unique depth matrix $\hat{\Lambda}$ as the solution to* (3.57).

Notice, that the uniqueness is with respect to $\hat{\Lambda}$, however a certain solution $\hat{\Lambda}$ gives a class of camera matrices and points, namely $(\hat{\mathtt{P}}\mathtt{H}, \mathtt{H}^{-1}\hat{\mathtt{X}})$ where $\mathtt{H}$ is an arbitrary invertible matrix.

Being reconstruction friendly is a desirable property for a constraint. However, this does not mean that other constraints are not useful. There can be other ways of avoiding false solutions, including choosing a proper initial solution for iterative factorization algorithms or trying different initial solutions or different forms of a certain class of constraints. What is important for reconstruction *unfriendly* constraints is to be aware of possible false solutions and being able to determine whether the algorithm has fallen into any of them.

Besides giving correct solutions to (3.57), there are other desirable properties one likes the constraint space to possess. We are specifically talking about the properties making the constraint usable with practical algorithms. For example, when dealing with iterative algorithms that converge to the final solution, it is essential that the constraint space C is closed. This is because for a non-closed constraint space, even if the sequence of solutions throughout all iterations satisfy all the constraints, they may converge to something outside C.

In the next subsections, to demonstrate how the theory we developed can be applied to the analysis of depth constraints, we examine some of the depth constraints used in the literature on factorization-based algorithms. It turned out that all of the constraints we could find in the literature either have a *compact* constraint space or are in the form of *linear equalities*. We consider each of these classes in a separate subsection. For each class, in addition to reviewing the constraints in the literature, we introduce a new class of constraints with extra desirable properties. This gives the reader an idea as to how our theory can be exploited for the design of new constraints. In particular, in Sect. 3.3.2.3, we introduce a class of linear equality constraints which are reconstruction friendly.

3.3.1 Compact Constraint Spaces

3.3.1.1 The Transportation Polytope Constraint

We consider the constraint used in [Dai et al., 2010, 2013], which is requiring $\hat{\Lambda}$ to have prescribed row and column sums and to have all nonnegative elements. This can be represented as

$$\hat{\Lambda}\mathbf{1}_n = \mathbf{u}, \quad \hat{\Lambda}^T\mathbf{1}_m = \mathbf{v}, \tag{3.58}$$

$$\hat{\Lambda} \succeq 0, \tag{3.59}$$

where the vectors $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^n$ are such that $u_i > 0$ for all i, $v_j > 0$ for all j and $\sum_{i=1}^m u_i = \sum_{j=1}^n v_j$. The relation $\succeq$ means element-wise greater or equal. Notice that although (3.58) introduces $m+n$ constraints, only $m+n-1$ of them are linearly independent. In [Angst et al., 2011] it has been noted that the corresponding constraint space is known as the Transportation Polytope. Thanks to a generalization of the well-known Sinkhorn's Theorem [Sinkhorn, 1964] for rectangular matrices [Sinkhorn, 1967], one can say that for every $m \times n$ matrix Λ with all positive elements and any two vectors $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^n$ with all positive entries, there exists a matrix $\hat{\Lambda}$ which is diagonally equivalent to Λ and satisfies the row and column sums constraint (3.58). Therefore, (3.58) is inclusive if the true depth matrix Λ is known to have all positive values, that is the set P representing the prior knowledge in Definition 3.6 is equal to the set of all $m \times n$ matrices with all positive elements. It is also obvious that the constraint (3.58) enforces all rows and all columns of $\hat{\Lambda}$ to be nonzero. Hence, every matrix in the constraint space satisfies depth assumptions (D1, D2). Therefore, to see if the constraint is exclusive it only remains to see whether or not constraints (3.58) and (3.59) allow for any cross-shaped depth matrix.

Assume that $\hat{\Lambda}$ is a cross-shaped matrix centred at (r, c), as in Fig. 3.4. Then the elements of $\hat{\Lambda}$ are uniquely determined by (3.58) as follows: $\hat{\lambda}_{ic} = u_i$ for all $i \neq r$, $\hat{\lambda}_{rj} = v_j$ for all $j \neq c$ and $\hat{\lambda}_{rc} = u_r - \sum_{j \neq c} v_j = v_c - \sum_{i \neq r} u_j$ (the latter equality is true due to $\sum_{i=1}^m u_i = \sum_{j=1}^n v_j$). This has been illustrated in Fig. 3.4. It is easy to check at all elements of $\hat{\Lambda}$ are nonnegative except possibly $\hat{\lambda}_{rc}$. Therefore, to satisfy (3.59), we must have $u_r - \sum_{j \neq c} v_j \geq 0$. Therefore, if for any choice of r and c, $u_r - \sum_{j \neq c} v_j \geq 0$ is satisfied, then the constraints (3.58) and (3.59) allow for a cross-shaped structure and hence, according to Lemma 3.17, allow a false solution to (3.57). Otherwise, (3.58) and (3.59) together give a reconstruction friendly constraint space, and hence, do not allow any false solution by Proposition 3.18.

As a major example, if we take $\mathbf{u} = n\mathbf{1}_m$ and $\mathbf{v} = m\mathbf{1}_n$ as chosen in [Dai et al., 2010, 2013], for any choice of r and c we have $u_r - \sum_{j \neq c} v_j = m + n - mn$. This is always negative by our assumption of having two or more views ($m \geq 2$) and 8 or more points ($n \geq 8$). Therefore, with the choice of $\mathbf{u} = n\mathbf{1}_m$ and $\mathbf{v} = m\mathbf{1}_n$, (3.58) and (3.59) give a reconstruction friendly constraint space. The disadvantage of this constraint is that it includes inequalities. This makes it difficult to implement fast and efficient algorithms for large scale problems.

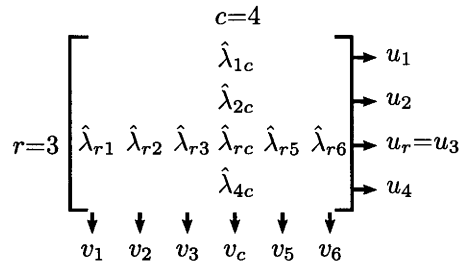


Figure 3.4: A 4×6 cross-shaped depth matrix $\hat{\Lambda}$ centred at (r,c) with $r = 3, c = 4$. The blank parts of the matrix indicate zero elements. The only way for the rows and columns of the matrix to sum up to the marginal values $\{u_i\}$ and $\{v_j\}$ is to have $\hat{\lambda}_{ic} = u_i$ for $i \neq r$, $\hat{\lambda}_{rj} = v_j$ for $j \neq c$, and $\hat{\lambda}_{rc} = u_r - \sum_{j\neq c} v_j = v_c - \sum_{i\neq r} u_j$.

3.3.1.2 Fixing the Norms of Rows and Columns

As suggested by Triggs [1996] and Hartley and Zisserman [2004], after each iteration of a factorization-based algorithm, one can alternatingly scale row and columns of $\hat{\Lambda}$ to have prescribed norms. Here, we analyse this case for the cases where the norms are l^p-norms for some real number $p \geq 1$ (being real implies $p < \infty$). Consider the matrix $\hat{\Gamma} \stackrel{\text{def}}{=} [|\hat{\lambda}_{ij}|^p]$, whose ij-th element is equal to $|\hat{\lambda}_{ij}|^p$. If all $\hat{\lambda}_{ij}$-s are nonzero, all elements of $\hat{\Gamma}$ are positive, and hence, alternatingly scaling row and columns of $\hat{\Lambda}$ to have prescribed l^p-norms is equivalent to alternatingly scaling rows and columns of $\hat{\Gamma}$ to have prescribed sums, that is applying the Sinkhorn's algorithm to $\hat{\Gamma}$ [Sinkhorn, 1964, 1967], making $\hat{\Gamma}$ converge to a matrix with prescribed row and column sums and hence making $\hat{\Lambda}$ converge to a matrix with prescribed row and column l^p-norms. Therefore, applying this iterative procedure after every iteration of a factorization-based algorithms keeps $\hat{\Lambda}$ in the following constraint space

$$\sum_{j=1}^{n} |\hat{\lambda}_{ij}|^p = u_i, \quad i = 1,\ldots,m \tag{3.60}$$

$$\sum_{i=1}^{m} |\hat{\lambda}_{ij}|^p = v_j, \quad j = 1,\ldots,n \tag{3.61}$$

for vectors $\mathbf{u} = [u_1,\ldots,u_m]^T$ and $\mathbf{v} = [v_1,\ldots,v_n]^T$ with all positive elements. Notice that $\mathbf{u}$ and $\mathbf{v}$ must be taken such that $\sum_{i=1}^{m} u_i = \sum_{j=1}^{n} v_j$. The above constrains $\hat{\Gamma} = [|\hat{\lambda}_{ij}|^p]$ as follows:

$$\hat{\Gamma}\mathbf{1}_n = \mathbf{u}, \quad \hat{\Gamma}^T\mathbf{1}_m = \mathbf{v}. \tag{3.62}$$

Moreover, $\hat{\Gamma} \succeq 0$ is automatically satisfied by the definition of $\hat{\Gamma}$. For the true depths λ_{ij}, take $\Gamma \stackrel{\text{def}}{=} [|\lambda_{ij}|^p]$ and notice that it has all positive elements as λ_{ij}-s are all nonzero. Thus, by applying the generalization of the Sinkhorn's theorem to rectangular matrices [Sinkhorn, 1967] we can say that there exists vectors $\boldsymbol{\tau} = [\tau_1,\tau_2,\ldots,\tau_m]^T$,

$\nu = [\nu_1, \nu_2, \ldots, \nu_n]^T$ with all positive entries such that $\hat{\Gamma} = \mathrm{diag}(\tau)\,\Gamma\,\mathrm{diag}(\nu)$ satisfies (3.62). Thus, for $\tau' = [\tau_1^{1/p}, \tau_2^{1/p}, \ldots, \tau_m^{1/p}]^T$, $\nu' = [\nu_1^{1/p}, \nu_2^{1/p}, \ldots, \nu_n^{1/p}]^T$, the matrix $\hat{\Lambda} = \mathrm{diag}(\tau')\,\Lambda\,\mathrm{diag}(\nu')$ satisfies (3.60) and (3.61). Therefore, (3.60) and (3.61) together give an inclusive constraint space. To check for (D1-D3), notice that $\hat{\Gamma}$ and $\hat{\Lambda}$ have a common zero pattern. Therefore, (D1-D3) are satisfied for $\hat{\Lambda}$ if and only if they are satisfied for $\hat{\Gamma}$. By considering (3.62) and $\hat{\Gamma} \succeq 0$, with the same discussion as the previous subsection we can say that (3.60) and (3.61) form a reconstruction friendly constraint if and only if $u_r - \sum_{j \neq c} v_j \geq 0$ for all r and c. Specifically, if one requires rows to have common norms and also columns to have common norms, as suggested by Triggs [1996] and Hartley and Zisserman [2004], then we have $\mathbf{u} = \alpha n \mathbf{1}_m$ and $\mathbf{v} = \alpha m \mathbf{1}_n$ for some nonzero scaling factor α. A similar argument as in the previous subsection shows that with this choice of $\mathbf{u}$ and $\mathbf{v}$, fixing l^p-norms of rows and columns results in a reconstruction friendly constraint space.

The problem with (3.62) as a constraint is that even simple target functions are hard to optimize subject to it. Implementing this constraint as a balancing stage after every iteration of a factorization-based algorithm can prevent us from having a descent move at every iteration.

3.3.1.3 Fixed Row or Column Norms

Heyden et al. [1999] uses the constraint of fixing the l^2-norms of the rows of the depth matrix. This constraint can be written as

$$\sum_{j=1}^{n} |\hat{\lambda}_{ij}|^2 = u_i, \qquad i = 1, \ldots, m \tag{3.63}$$

for fixed positive numbers u_i. Indeed, this constraint is inclusive as for every matrix Λ with all nonzero rows one can scale the rows to obtain a matrix $\hat{\Lambda} = \mathrm{diag}(\tau)\Lambda$ with prescribed row norms. Every matrix $\hat{\Lambda}$ satisfying this constraint cannot have zero rows. However, the constraint allows for zero columns and cross-shaped solutions. A similar situation holds for [Mahamud et al., 2001] where the columns of the depth matrix are required to have a unit (weighted) l^2-norm.

The disadvantage of these constraints is allowing for zero columns (or zero rows in the second case) and cross-shaped structures. The advantage is that they can be efficiently implemented with iterative factorization-based algorithms, by solving a number of eigenvalue problems at every iteration [Mahamud et al., 2001]. The compactness of the constraint space contributes to the proof of special convergence properties for special factorization-based algorithms [Mahamud et al., 2001].

3.3.1.4 Fixing Norms of Tiles

In this subsection we show how the fixed row and fixed column constraints can be somehow combined to make more desirable constraints. This is done by *tiling* the depth matrix $\hat{\Lambda}$ with row and column vectors, and requiring each tile to have a unit norm (or a fixed norm in general). Examples of tiling can be seen in Fig. 3.5.

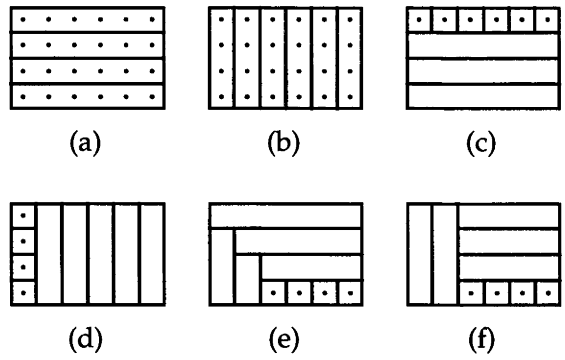


Figure 3.5: Examples of tiling a 4×6 depth matrix with row and column vectors. The associated constraint is to force every tile of the depth matrix to have a unit (or a fixed) norm. This gives a compact constraint space. If the tiling is done according to (a) every row of the constrained depth matrix has unit norm. Similarly, tiling according to (b) requires columns with unit norms. Constraints associated with (a) and (b), respectively, allow zero columns and zero rows in the depth matrix, along with cross-shaped configurations. The associated constraints for (c-f) do not allow any zero rows or zero columns, however, they all allow cross-shaped structures. For each of the cases (a-f), the dots indicate possible locations where the cross-shaped structures allowed by the associated constraint can be centred. Clearly, for (a) and (b) the cross can be centred anywhere, whereas for (c-f) they can only be centred at 1×1 tiles.

The process of tiling is done as follow: It starts by putting a single tile (row vector or column vector) in the matrix. We then keep adding tiles such that the tiled area stays rectangular. At every stage either a horizontal tile (row vector) is vertically concatenated or a vertical tile (column vector) is horizontally concatenated to the already tiled area, with the constraint that the tiled region remains rectangular. The process is continued until the whole $\hat{\Lambda}$ is tiled. This process is illustrated in Fig. 3.6. By tiling the matrix in this way, the corresponding constraint will be inclusive. We do not prove this formally here, instead, we show how the proof is constructed by giving an example in Fig. 3.6.

Fig. 3.5 shows six examples of tiling a 4×6 depth matrix. Looking at Fig. 3.5(a) one can see that for an $m\times n$ matrix, if the tiling begins by placing a $1\times n$ block, all other tiles have to be also $1\times n$ and the constraint is reduced to the case of requiring fixed row norms, a special case of which was discussed in the previous subsection. Similarly, if the first tile is $m\times1$, the constraint amounts to fixing the norms of columns of the depth matrix Fig. 3.5(b). But the case of interest here is when the first tile is a 1×1 block, like Fig. 3.5(c-f). In this case, the constraint rules out having zero rows or zero columns in the depth matrix. It does not rule out cross-shaped structures, but it constrains the central position of the cross to the location of 1×1 tiles (see Fig. 3.5(c-f)).

If the norms used for the constraints are weighted l^2-norms with properly chosen

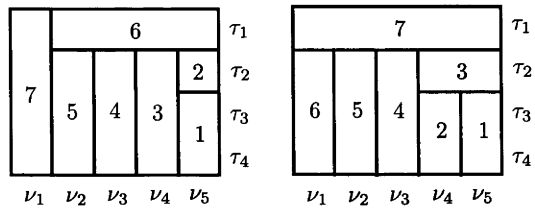


Figure 3.6: Examples of the procedure of tiling a 4×5 depth matrix. The numbers show the order in which the tiles are placed. In these examples, we start by placing a 2×1 tile on the left bottom of the matrix. The tiles are added such that the tiled region at any time remains a rectangle. Having an $m' \times n'$ rectangular area tiled already, we either concatenate an $m' \times 1$ vertical block to its left, or a $1 \times n'$ block to its top. The claim is that with this procedure the constraint of every tile having a unit (or a fixed positive) norm is inclusive. This can be shown as follows: We start by taking $\hat{\Lambda} = \Lambda$, and keep updating $\hat{\Lambda}$ by scaling one of its rows or one of its columns at a time until it satisfies all the constraints, that is all of its tiles have a unit norm. For matrix (a), the updates can be done as follows: choose arbitrary nonzero values for τ_3 and τ_4 and apply them to the matrix (multiply them respectively by the 3rd and 4th row of $\hat{\Lambda}$). Now, choose ν_5 such that tile 1 has a unit norm and apply it. Then choose τ_2 and apply it such that tile 2 has a unit norm. Now, choose and apply ν_4, ν_3 and ν_2 such that tiles 3, 4, 5 have a unit norm, and finally choose and apply τ_1 and then ν_1 to respectively make tiles 6 and 7 have a unit norm. The procedure for (b) is similar, but the order of finding τ_i-s and ν_j-s is as follows: $\tau_3, \tau_4, \nu_5, \nu_4, \tau_2, \nu_3, \nu_2, \nu_1, \tau_1$.

weights, an efficient factorization algorithm can be implemented. For more details see Sect. 6.2. Similar convergence properties as in [Mahamud et al., 2001] can be proved for these constraints given a proper algorithm.

3.3.2 Linear Equality Constraints

3.3.2.1 Fixing Sums of Rows and Columns

In this subsection, we consider constraining $\hat{\Lambda}$ to have prescribed row and column sums, that is

$$\hat{\Lambda}\mathbf{1}_n = \mathbf{u}, \quad \hat{\Lambda}^T\mathbf{1}_m = \mathbf{v}, \tag{3.64}$$

for two m- and n-dimensional vectors $\mathbf{u}$ and $\mathbf{v}$ with all nonzero entries for which $\sum_{i=1}^{m} u_i = \sum_{j=1}^{n} v_j$. This is similar to the transportation polytope constraint introduced in Sect. 3.3.1.1, but it does not require $\hat{\Lambda} \succeq 0$. Thus, it has the advantage of allowing for more efficient algorithms compared to the case where inequality constraints are also present. We can see this in [Dai et al., 2013], where the inequality constraint $\hat{\Lambda} \succeq 0$ has been disregarded when proposing fast and scalable algorithms.

With a similar argument as was made in Sect. 3.3.1.1, one can say that (3.64)

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} & & & 6 & & \\ 4 & 4 & 4 & -14 & 4 & 4 \\ & & & 6 & & \\ & & & 6 & & \end{bmatrix}$$

(a) (b)

Figure 3.7: Examples of 4×6 matrices, both satisfying $\hat{\Lambda}\mathbf{1}_n = n\mathbf{1}_m$ and $\hat{\Lambda}^T\mathbf{1}_m = m\mathbf{1}_n$. (a) is a typical initial state for iterative factorization-based algorithm, (b) is the only cross-shape structure centred at (2,4) allowed by the constraint. If the true depths are all positive, it can be harder for an algorithm to converge from (a) to (b), compared to converging to a correct solution with all positive elements.

gives an inclusive constraint space when the true depth matrix Λ is known to have all positive elements, and $\mathbf{u}$ and $\mathbf{v}$ are chosen to have all positive entries. The constraint also enforces all rows and columns of $\hat{\Lambda}$ to be nonzero.

However, as noted in Sect. 3.3.1.1, a cross-shaped matrix with any arbitrary centre (r, c) whose elements are chosen as $\hat{\lambda}_{ic} = u_i$ for all $i \neq r$, $\hat{\lambda}_{rj} = v_j$ for all $j \neq c$ and $\hat{\lambda}_{rc} = u_r - \sum_{j \neq c} v_j = v_c - \sum_{i \neq r} u_j$, satisfies (3.64). Therefore, by Lemma 3.17 we can say that it always allows for cross-shaped solutions.

The bad thing about this type of constraint is that there is no limitation as to where the cross-shaped structure can be centred. But the good thing is that according to our experiments it can be hard for an iterative algorithm to converge to a cross-shaped solution with the choice of $\mathbf{u} = n\mathbf{1}_m$ and $\mathbf{v} = m\mathbf{1}_n$. This could be explained as follows: As noted in Sect. 3.3.1.1, if any cross-shaped structure occurs, the central element will have to be equal to $m + n - mn$. Under our assumptions ($m \geq 2, n \geq 8$), this is a negative number and its absolute value grows linearly both with respect to m and n. This can make it hard for the algorithm to converge to a cross-shaped structure starting from an initial solution like a matrix of all ones. This has been depicted in Fig. 3.7 for a 4×6 matrix, where the central element of the cross has to be -14. For a fairly small configuration of 20-views and 8-points this value is -132. This suggests that as the dimension of the depth matrix grows, it is made harder for the algorithm to converge to a cross-shaped solution.

3.3.2.2 Fixing Elements of one row and one column

Here, we consider the constraint of having all elements of a specific row and a specific column of the depth matrix equal to one, as used in [Ueshiba and Tomita, 1998]. This means requiring $\hat{\lambda}_{rj} = 1$ for all j, and $\hat{\lambda}_{ic} = 1$ for all i. This can be represented as

$$\mathtt{M} \circ \hat{\Lambda} = \mathtt{M}. \tag{3.65}$$

where $\circ$ represents the Hadamard (element-wise) product and $\mathtt{M}$ is a mask matrix, having all elements of a specific row r and a specific column c equal to 1, and the rest of its elements equal to zero. This means that the mask matrix $\mathtt{M}$ is a cross-shaped

matrix centred at (r,c). We leave it to the reader to check that this is an inclusive constraint, and also every matrix in the constraint space satisfies depth assumptions (D1) and (D2). However, one can easily check that, as M itself is a cross-shaped matrix, the constraint (3.65) allows for cross-shaped depth matrices. Therefore, by using the above constraint problem (3.57) can admit false solutions.

One advantage of this type of constraint is its elementwise nature. This can make the formulation of iterative factorization algorithms much easier compared to other types of constraints. The other advantage is that there is only a single possibility about where the cross in centred, which is the centre of cross in M. Therefore, the occurrence of a cross-shaped solution can be easily verified. In the case where a cross-shaped solution happens, one can try rerunning the algorithm with a different mask M whose cross is centred elsewhere.

3.3.2.3 Step-like Mask Constraint: A Linear Reconstruction Friendly Equality Constraint

This section demonstrates a group of linear equality constraints which are reconstruction friendly, and therefore exclude all possible wrong solutions to the projective factoriation problem. Like the previous subsection, the linear equalities are in the form of fixing elements of the depth matrix at certain sites. Therefore, it enjoys all the benefits of elementwise constraints.

To present the constraint, we first define the concept of a step-like mask. Consider an $m \times n$ matrix M. To make a step-like mask, we have a travel starting from the upper-left corner of the matrix (location $1,1$) and ending at its lower-right corner (location m,n). The travel from $(1,1)$ to (m,n) is done by taking $m+n-2$ moves, such that at each move we either go one step to the right or go one step down. In total, we will make $m-1$ downward moves and $n-1$ moves to the right. Therefore, the travel can be made in $(m+n-2)!/((m-1)!\,(n-1)!)$ ways. After doing a travel, we make the associated step-like mask by setting to 1 all $(m+n-1)$ elements of M corresponding to the locations that we have visited and setting to zero the rest of the elements. Examples of step-like masks are shown in Fig. 3.8 for $m=4$ and $n=6$.

Notice that a step-like mask has $m+n-1$ nonzero elements which are arranged such that the matrix has no zero rows and no zero columns. An exclusive step-like mask is defined to be a step-like mask which is not cross-shaped (see Fig. 3.8). With an $m \times n$ step-like mask we can put linear equality constraints on a depth matrix $\hat{\Lambda}$ as follows

$$\mathtt{M} \circ \hat{\Lambda} = \mathtt{M}. \tag{3.66}$$

where $\circ$ represents the Hadamard (element-wise) product. In other words, it enforces the matrix $\hat{\Lambda}$ to have unit elements at the sites where M has ones.

One can show that with an exclusive step-like mask M, the constraint (3.66) is uniquely reconstruction friendly. As the constraints enforce $\hat{\Lambda}$ to be nonzero at the sites where M has ones, it is easy to see that if $\hat{\Lambda}$ satisfies (3.66), it satisfies (D1-D3) and hence the constraint space is exclusive. Therefore, we just have to show that for each

$$\begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & 1 & & \\ & & & 1 & 1 & \\ & & & & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 & & & \\ & & & 1 & & \\ & & & 1 & & \\ & & & 1 & 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & & & & & \\ 1 & & & & & \\ 1 & & & & & \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(a) (b) (c)

Figure 3.8: Examples of 4×6 step-like mask matrices. Blank parts of the matrices indicate zero values. A step-like matrix contains a chain of ones, starting from its upper left corner and ending at its lower right corner, made by making rightward and downward moves only. An exclusive step-like mask is one which is not cross-shaped. In the above, (a) and (b) are samples of an exclusive step-like mask while (c) is a nonexclusive one. Associated with an $m \times n$ step-like mask $\mathtt{M}$, one can put a constraint on an $m \times n$ depth matrix $\hat{\Lambda}$ in the form of fixing the elements of $\hat{\Lambda}$ to 1 (or some nonzero values) at sites where $\mathtt{M}$ has ones. For an exclusive step-like mask, this type of constraint rules out all the wrong solutions to the factorization-based problems.

matrix Λ with all nonzero elements, there exists exactly one diagonally equivalent matrix $\hat{\Lambda}$ satisfying (3.66). The proof is quite simple, but instead of the formal proof, we explain the idea by giving an example of a special case in Fig. 3.9.

$$\begin{array}{c} \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{array} \begin{array}{c} \begin{array}{cccc} \nu_1 & \nu_2 & \nu_3 & \nu_4 \end{array} \\ \begin{bmatrix} \underline{\lambda_{11}} & \underline{\lambda_{12}} & \lambda_{13} & \lambda_{14} \\ \lambda_{21} & \underline{\lambda_{22}} & \lambda_{23} & \lambda_{24} \\ \lambda_{31} & \underline{\lambda_{32}} & \underline{\lambda_{33}} & \underline{\lambda_{34}} \end{bmatrix} \end{array} \qquad \mathtt{M} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Figure 3.9: An example of a 3×4 depth matrix Λ (left) and an exclusive step-like mask $\mathtt{M} = [m_{ij}]$ (right). The elements λ_{ij} of Λ are underlined at the sites where $m_{ij} = 1$, which is where $\hat{\lambda}_{ij}$-s are constrained to be equal to 1. The aim is to show that there exists a unique $\hat{\Lambda}$ in the form of $\hat{\Lambda} = \mathrm{diag}(\tau)\,\Lambda\,\mathrm{diag}(\nu)$ whose elements are 1 at the sites where $\mathtt{M}$ has ones. Equivalently $\mathtt{M} \circ \hat{\Lambda} = \mathtt{M}$. This can be done as follows: Start by taking $\hat{\Lambda} = \Lambda$, and keep updating $\hat{\Lambda}$ by scaling its rows and columns, one at a time, until it satisfies the constraint $\mathtt{M} \circ \hat{\Lambda} = \mathtt{M}$. For the above matrix, we start by assigning an arbitrary nonzero value to τ_1 and multiplying τ_1 by the first row of $\hat{\Lambda}$. Then we choose ν_1 and ν_2 and multiply them by the corresponding columns of $\hat{\Lambda}$ such that $\hat{\lambda}_{11} = 1$ and $\hat{\lambda}_{12} = 1$. Now, we choose τ_2 and τ_3 and multiply them by the corresponding rows of $\hat{\Lambda}$ such that we have $\hat{\lambda}_{22} = 1$ and $\hat{\lambda}_{32} = 1$. Finally, we choose ν_3 and ν_4 and multiply them by the corresponding columns of $\hat{\Lambda}$ to have $\hat{\lambda}_{33} = 1$ and $\hat{\lambda}_{34} = 1$. Notice that in this process, except τ_1 which is chosen arbitrarily, there is only one choice for each of the entries $\tau_2, \tau_3, \nu_1, \nu_2, \nu_3, \nu_4$ for each choice of τ_1. Because, given any pair of vectors (τ, ν), all pairs of vectors $(\alpha\tau, \alpha^{-1}\nu)$ for all $\alpha \neq 0$ have the same effect, this means that given the matrices Λ and $\mathtt{M}$, the choice of $\hat{\Lambda} = \mathrm{diag}(\tau)\,\Lambda\,\mathrm{diag}(\nu)$ is unique.

$$\begin{bmatrix} 1 & 0 & & & & \\ & 1 & 1 & 0 & & \\ & & & 1 & 0 & \\ & & & & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 0 & & & \\ & & 1 & & & \\ & & 1 & & & \\ & & 0 & 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & & & & & \\ 1 & & & & & \\ 1 & & & & & \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(a) (b) (c)

Figure 3.10: Examples of 4×6 edgeless step-like mask matrices obtained by removing (making zero) some of the stair edges of matrices in Fig. 3.8. The blank parts of the matrices are zero. The elements explicitly shown by 0 are the removed edges (those that are 1 on the original step-like matrix). (a) and (b) are examples of an exclusive edgeless step-like matrix, resulting in a reconstruction friendly constraint.

One can think of many ways to extend the step-like constraints. For example, one can fix the desired elements of $\hat{\Lambda}$ to arbitrary nonzero values instead of ones. The reader can also check that if $\mathtt{M}$ is obtained by applying any row and column permutation to an exclusive step-like mask, then the constraint (3.66) will be still reconstruction friendly. One important extension is to remove some of the constraints by turning to 0 some of the elements of the mask matrix $\mathtt{M}$. Potential elements of a step-like matrix $\mathtt{M}$ for the removal (switching to zero) are the stair edges, which are the elements whose left and lower elements (or right and upper elements) are 1 (see Fig. 3.10). We call the new matrices *edgeless* step-like masks. As switching some elements of $\mathtt{M}$ to zero amounts to removing some linear equations from the set of constraints, an edgeless step-like mask still gives an inclusive constraint. If the edge elements for the removal are chosen carefully from an exclusive step-like mask, the corresponding constraint $\mathtt{M} \circ \hat{\Lambda} = \mathtt{M}$ can still be exclusive, not allowing for the violation of (D1-D3). Fig. 3.10(a,b) illustrates examples of exclusive edgeless step-like masks. The corresponding constraint $\mathtt{M} \circ \hat{\Lambda} = \mathtt{M}$ for such a mask is reconstruction friendly, however it is not *uniquely* reconstruction friendly. Our experiments show that, using the same algorithm, an edgeless mask results in a faster convergence than its corresponding edged mask. One explanation is that, in this case, the removal of each constraint, in addition to increasing the dimension of the search space, increases the dimension of the solution space[4] by one. This can allow an iterative algorithm to find a shorter path from the initial estimate of $\hat{\Lambda}$ to a correct solution.

3.4 Projective Reconstruction via Rank Minimization

Recall from the last section that in the factorization-based projective reconstruction the following problem is sought to be solved

$$\text{find}_{\hat{\Lambda},\, \hat{\mathtt{P}}_{3m\times4},\, \hat{\mathtt{X}}_{4\times n}} \text{ s.t. } \hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathtt{P}}\,\hat{\mathtt{X}}, \quad \hat{\Lambda} \in C \tag{3.67}$$

[4] namely $\{\hat{\Lambda} \mid \hat{\Lambda} = \text{diag}(\tau)\,\Lambda\,\text{diag}(\nu),\ \mathtt{M} \circ \hat{\Lambda} = \mathtt{M}\}$

which is a restatement of (3.57). Rank minimization is one of the approaches to factorization-based projective reconstruction, in which, in lieu of (3.67), the following problem is solved:

$$\min_{\hat{\Lambda}} \mathcal{R}\text{ank}(\hat{\Lambda} \odot [\mathbf{x}_{ij}]) \quad \text{s.t. } \hat{\Lambda} \in C. \tag{3.68}$$

Two other closely related problems are

$$\text{find } \hat{\Lambda} \quad \text{s.t. } \mathcal{R}\text{ank}(\hat{\Lambda} \odot [\mathbf{x}_{ij}]) \leq 4, \ \hat{\Lambda} \in C, \tag{3.69}$$

$$\text{find } \hat{\Lambda} \quad \text{s.t. } \mathcal{R}\text{ank}(\hat{\Lambda} \odot [\mathbf{x}_{ij}]) = 4, \ \hat{\Lambda} \in C. \tag{3.70}$$

If any solution $\hat{\Lambda}$ is found for any of the above problems such that $\mathcal{R}\text{ank}(\hat{\Lambda} \odot [\mathbf{x}_{ij}]) \leq 4$, the camera matrices and points can be estimated from the factorization of $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ as $\hat{\mathbf{P}}\hat{\mathbf{X}}$. We shall show that if C is reconstruction friendly, any solution to any of the above problems leads to projective reconstruction. First, it is easy to see that (3.69) is in fact equivalent to problem (3.67):

Lemma 3.19. *Given any set of 3D points $\mathbf{x}_{ij}$ for $i = 1,2,\ldots,m$ and $j = 1,2,\ldots,n$, the problems* (3.69) *and* (3.67) *are equivalent in terms of finding $\hat{\Lambda}$.*

Here, by being equivalent we mean that any solution $\hat{\Lambda}$ to one problem is a solution to the other. Obviously, this implies that if there exists no solution to one of the problems, then there cannot exist any solution to the other. The proof is quite simple:

Proof. Consider a solution $(\hat{\Lambda}, \hat{\mathbf{P}}, \hat{\mathbf{X}})$ to (3.67). Since $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathbf{P}}\hat{\mathbf{X}}$ for $\hat{\mathbf{X}} \in \mathbb{R}^{4\times n}$, it has rank 4 or less. Therefore, $\hat{\Lambda} \in C$ is also a solution to (3.69).

Now, consider a solution $\hat{\Lambda} \in C$ to (3.69). As $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ has rank $r' \leq 4$, it can be can be factored as $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \mathbf{U}\mathbf{V}^T$ where $\mathbf{U}$ is $3m \times r'$ and $\mathbf{V}$ is $n \times r'$. Let $\hat{\mathbf{P}} = [\mathbf{U}, \mathbf{0}_{3m\times(4-r')}] \in \mathbb{R}^{3m\times 4}$ and $\hat{\mathbf{X}} = [\mathbf{V}, \mathbf{0}_{n\times(4-r')}]^T \in \mathbb{R}^{4\times n}$. Then we have $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \mathbf{U}\mathbf{V}^T = \hat{\mathbf{P}}\hat{\mathbf{X}}$. Thus, $(\hat{\Lambda}, \hat{\mathbf{P}}, \hat{\mathbf{X}})$ is a solution to (3.67). □

Notice that to prove the above lemma we need not make any assumption about C or how the points $\mathbf{x}_{ij}$ are created. The two other problems (3.68) and (3.70) are not in general equivalent to (3.67). However, if C is reconstruction friendly, one can show that all the four problems (3.68), (3.69), (3.70) and (3.67) are equivalent:

Proposition 3.20. *Consider a setup of $m \geq 2$ camera matrices and $n \geq 8$ points $(\{\mathbf{P}_i\}, \{\mathbf{X}_j\})$ generically configured in the sense of (G1-G4), and projecting into the image points $\{\mathbf{x}_{ij}\}$ according to $\lambda_{ij}\mathbf{x}_{ij} = \mathbf{P}_i\mathbf{X}_j$ with nonzero scalars λ_{ij}. If $C \subseteq \mathbb{R}^{m\times n}$ is a reconstruction friendly constraint space, then given the image points $\mathbf{x}_{ij}$, the problems* (3.68), (3.69) *and* (3.70) *are all equivalent to* (3.67) *in terms of finding $\hat{\Lambda}$.*

Proof. As (3.69) and (3.67) are equivalent, the proof will be complete by showing

- (3.70) $\subseteq$ (3.69),
- (3.67) $\subseteq$ (3.70),

- (3.68) $\subseteq$ (3.69),
- (3.70) $\subseteq$ (3.68),

where (P1) $\subseteq$ (P2) means that any solution to (P1) is a solution to (P2). The first part, that is (3.70) $\subseteq$ (3.69), is obvious. To show (3.67) $\subseteq$ (3.70), assume that $(\hat{\Lambda}, \hat{\mathtt{P}}, \hat{\mathtt{X}})$ is a solution to (3.67). By Proposition 3.18 and the definition of projective equivalence we can conclude that $\hat{\mathtt{P}} = \mathrm{diag}(\tau \otimes \mathbf{1}_3)\,\mathtt{PH}$ and $\hat{\mathtt{X}} = \mathtt{H}^{-1}\mathtt{X}\,\mathrm{diag}(\nu)$ for some invertible matrix $\mathtt{H}$ and vectors τ and ν with all nonzero entries, where $\mathtt{P} = \mathrm{stack}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$, $\mathtt{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_n]$ and $\otimes$ denotes the Kronecker product. This gives

$$\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathtt{P}}\hat{\mathtt{X}} = \mathrm{diag}(\tau \otimes \mathbf{1}_3)\,\mathtt{PX}\,\mathrm{diag}(\nu) \tag{3.71}$$

From (G1,G2) it follows that $\mathtt{P}$ and $\mathtt{X}$ respectively have full column and full row rank, and hence, $\mathtt{PX}$ is of rank 4. Given this, plus the fact that τ and ν have all nonzero entries, (3.71) implies that $\mathcal{R}\mathrm{ank}(\hat{\Lambda} \odot [\mathbf{x}_{ij}]) = 4$, meaning $\hat{\Lambda}$ is a solution to (3.70).

To see (3.68) $\subseteq$ (3.69), notice that according to Proposition 3.18, (3.67) has at least one solution. This means that the equivalent problem (3.69) has also one solution and therefore, there exist a $\hat{\Lambda}' \subseteq C$ for which $\mathcal{R}\mathrm{ank}(\hat{\Lambda}' \odot [\mathbf{x}_{ij}]) \leq 4$. Now, for any solution $\hat{\Lambda} \subseteq C$ to (3.68) we have $\mathcal{R}\mathrm{ank}(\hat{\Lambda} \odot [\mathbf{x}_{ij}]) \leq \mathcal{R}\mathrm{ank}(\hat{\Lambda}' \odot [\mathbf{x}_{ij}]) \leq 4$. This means that $\hat{\Lambda}$ is also a solution to (3.69).

Finally, to show (3.70) $\subseteq$ (3.68), notice that since (3.69) and (3.67) are equivalent, from (3.68) $\subseteq$ (3.69) and (3.67) $\subseteq$ (3.70) we conclude that any solution $\hat{\Lambda}$ to (3.68) is also a solution to (3.70). This, plus the fact that (3.68) always attains its minimum[5], means that $\mathcal{R}\mathrm{ank}(\hat{\Lambda} \odot [\mathbf{x}_{ij}]) \geq 4$ for all $\hat{\Lambda} \in C$. Thus, any solution to (3.70) minimizes $\mathcal{R}\mathrm{ank}(\hat{\Lambda} \odot [\mathbf{x}_{ij}])$, and hence, is also a solution to (3.68). □

Moreover, as Proposition 3.18 suggests that (3.67) has at least one solution, we can say that with the conditions of Proposition 3.20, all the problems (3.68), (3.69) and (3.70) have at least one solution.

3.5 Iterative Projective Reconstruction Algorithms

Nearly, all of the projective factorization-based problems are solved iteratively. The output of such algorithms is not in the form of a deterministic final solution, but rather is a sequence $(\{\hat{\mathtt{P}}_i^{(t)}\}, \{\hat{\mathbf{X}}_j^{(t)}\}, \{\hat{\lambda}_{ij}^{(t)}\})$ which one hopes to converge to a sensible solution. There are many questions such as whether this sequence converges, and if it does, whether it converges to a correct solution. Answering such algorithm-specific questions, however, is beyond the scope of this thesis. However, a more basic question that needs answering is that, given a constraint space C, if the sequence $\{\hat{\Lambda}^{(t)}\} \subseteq C$ converges to some $\hat{\Lambda}$ and moreover the sequence $\{\hat{\Lambda}^{(t)} \odot [\mathbf{x}_{ij}] - \hat{\mathtt{P}}^{(t)}\hat{\mathtt{X}}^{(t)}\}$ converges to zero, then whether $\hat{\Lambda}$ is a solution to the factorization problem (3.57), that is $\hat{\Lambda} \in C$ and $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathtt{P}}\,\hat{\mathtt{X}}$ for some $\hat{\mathtt{P}} \in \mathbb{R}^{3m\times 4}$ and $\hat{\mathtt{X}} \in \mathbb{R}^{4\times n}$. It is easy to check that C being *closed* is sufficient for this to happen:

[5]The reason is that $\mathcal{R}\mathrm{ank}(\hat{\Lambda} \odot [\mathbf{x}_{ij}])$ is a member of a finite set.

Proposition 3.21. *Consider a set of image points* $\{\mathbf{x}_{ij}\}$, $i = 1,\ldots,m$ *and* $j = 1,\ldots,n$, *and a* closed *constraint space* $C \subseteq \mathbb{R}^{m\times n}$. *If there exists a sequence of depth matrices* $\{\hat{\Lambda}^{(t)}\} \subseteq C$ *converging to a matrix* $\hat{\Lambda}$, *and for each* $\hat{\Lambda}^{(t)}$ *there exist* $\hat{\mathrm{P}}^{(t)} \in \mathbb{R}^{3m\times 4}$ *and* $\hat{\mathrm{X}}^{(t)} \in \mathbb{R}^{4\times n}$ *such that* $\hat{\Lambda}^{(t)} \odot [\mathbf{x}_{ij}] - \hat{\mathrm{P}}^{(t)}\hat{\mathrm{X}}^{(t)} \to 0$ *as* $t \to \infty$, *then there exist* $\hat{\mathrm{P}} \in \mathbb{R}^{3m\times 4}$ *and* $\hat{\mathrm{X}} \in \mathbb{R}^{4\times n}$ *such that* $(\hat{\Lambda}, \hat{\mathrm{P}}, \hat{\mathrm{X}})$ *is a solution to the factorization problem*

$$\text{find}_{\hat{\Lambda},\, \hat{\mathrm{P}}_{3m\times 4},\, \hat{\mathrm{X}}_{4\times n}} \quad s.t. \;\; \hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathrm{P}}\hat{\mathrm{X}}, \;\; \hat{\Lambda} \in C \tag{3.72}$$

Proof. Let $\mathrm{A}^{(t)} = \hat{\mathrm{P}}^{(t)}\hat{\mathrm{X}}^{(t)}$. As the mapping $\Lambda' \mapsto \Lambda' \odot [\mathbf{x}_{ij}]$ is continuous, $\hat{\Lambda}^{(t)} \odot [\mathbf{x}_{ij}] - \mathrm{A}^{(t)} \to 0$ and $\hat{\Lambda}^{(t)} \to \hat{\Lambda}$ give $\mathrm{A}^{(t)} \to \hat{\Lambda} \odot [\mathbf{x}_{ij}] \stackrel{\text{def}}{=} \mathrm{A}$. Also, $\mathcal{R}\text{ank}(\mathrm{A}) \leq 4$ because $\mathcal{R}\text{ank}(\mathrm{A}^{(t)}) \leq 4$ and the space of $3m\times n$ real matrices with rank 4 or less is closed. Thus, A can be factored as $\mathrm{A} = \hat{\mathrm{P}}\hat{\mathrm{X}}$ for some $\hat{\mathrm{P}} \in \mathbb{R}^{3m\times 4}$ and $\hat{\mathrm{X}} \in \mathbb{R}^{4\times n}$, giving $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \mathrm{A} = \hat{\mathrm{P}}\hat{\mathrm{X}}$. Moreover, as C is closed and $\{\hat{\Lambda}^{(t)}\} \subseteq C$ we have $\hat{\Lambda} \in C$. This completes the proof. □

According to the above, as long as the constraint space C is closed, all the results obtained in the previous section about the solutions to the factorization problem (3.57), can be safely used for iterative algorithms when the sequence of depths $\{\hat{\Lambda}^{(t)}\}$ is convergent and $\hat{\Lambda}^{(t)} \odot [\mathbf{x}_{ij}] - \hat{\mathrm{P}}^{(t)}\hat{\mathrm{X}}^{(t)}$ converges to zero.

3.6 Summary

We presented a generalized theorem of projective reconstruction in which it has not been assumed, a priori, that the estimated projective depths are all nonzero. We also presented examples of the wrong solutions to the projective factorization problem when not all the estimated projective depths are constrained to be nonzero. We used our theory to analyse some of the depth constraints used in the literature for projective factorization problem, and also demonstrated how the theory can be used for the design of new constraints with desirable properties.

Chapter 4

Arbitrary Dimensional Projections

In this chapter we consider the problem of projective reconstruction for arbitrary dimensional projections, where we have multiple projections with the i-th projection being from $\mathbb{P}^{r-1}$ to $\mathbb{P}^{s_i-1}$. We give theories for deducing projective reconstruction from the set of projection equalities

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j \tag{4.1}$$

for $i = 1,\ldots,m$ and $j = 1,\ldots,n$, where $\mathbf{X}_j \in \mathbb{R}^r$ are high-dimensional (HD) points, representing points in $\mathbb{P}^{r-1}$ in homogeneous coordinates, $\mathtt{P}_i \in \mathbb{R}^{s_i \times r}$ are projection matrices, representing projections $\mathbb{P}^{r-1} \to \mathbb{P}^{s_i-1}$ and $\mathbf{x}_{ij} \in \mathbb{R}^{s_i}$ are image points. Each image point $\mathbf{x}_{ij} \in \mathbb{R}^{s_i}$ represents a point in $\mathbb{P}^{s_i-1}$ in homogeneous coordinates. The nonzero scalars λ_{ij}-s are known as *projective depths* (see Sect. 2.1 for more details).

After providing the required background in Sect. 4.1, we give a basic theorem in Sect. 4.2 which proves the uniqueness of projective reconstruction given the image points $\mathbf{x}_{ij}$ from the set of relation 4.1, under some conditions on the estimated projection matrices and HD points. The main step to prove the theorem is proving the uniqueness of the multi-view (Grassmann) tensor given the image points $\mathbf{x}_{ij}$ which is done in Sect. 4.2.1.

In Sect. 4.3 we prove that all configurations of projection matrices and HD points projecting into the same image points $\mathbf{x}_{ij}$ (all satisfying (4.1) with nonzero depths λ_{ij}) are projectively equivalent. Notice that uniqueness of the Grassmann tensor is not sufficient for obtaining this result, as it does not rule out the existence of degenerate solutions $\{\mathtt{P}_i\}$ whose corresponding Grassmann tensor is zero.

Finally, in Sect. 4.4 we classify the degenerate wrong solutions to the projective factorization equation $\Lambda \odot [\mathbf{x}_{ij}] = \mathtt{P}\,\mathtt{X}$ where not all the projective depths are restricted to be nonzero.

4.1 Background

4.1.1 Triangulation

The problem of Triangulation is to find a point $\mathbf{X}$ given its images through a set of known projections $\mathtt{P}_1,\ldots,\mathtt{P}_m$. The next lemma provides conditions for the unique-

ness of triangulation.

Lemma 4.1 (Triangulation). *Consider a set of projection matrices* $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m$ *with* $\mathtt{P}_i \in \mathbb{R}^{s_i \times r}$, *and a point* $\mathbf{X} \in \mathbb{R}^r$, *configured such that*

(T1) there does not exist any linear subspace of dimension less than or equal to 2, passing through $\mathbf{X}$ *and nontrivially intersecting*[1] *all the null spaces* $\mathcal{N}(\mathtt{P}_1), \mathcal{N}(\mathtt{P}_2), \ldots, \mathcal{N}(\mathtt{P}_m)$.

Now, for any nonzero $\mathbf{Y} \neq \mathbf{0}$ *in* $\mathbb{R}^r$ *if the relations*

$$\mathtt{P}_i\mathbf{Y} = \beta_i \mathtt{P}_i\mathbf{X}, \qquad i = 1, 2, \ldots, m \tag{4.2}$$

hold for scalars β_i, *then* $\mathbf{Y} = \beta\mathbf{X}$ *for some scalar* $\beta \neq 0$.

Notice that we have not assumed $\beta_i \neq 0$.

Proof. From $\mathtt{P}_i\mathbf{Y} = \beta_i \mathtt{P}_i\mathbf{X}$ we deduce

$$\mathbf{Y} = \beta_i \mathbf{X} + \mathbf{C}_i \tag{4.3}$$

for some $\mathbf{C}_i \in \mathcal{N}(\mathtt{P}_i)$, which means $\mathbf{C}_i \in \text{span}(\mathbf{X}, \mathbf{Y})$. Now, if all $\mathbf{C}_i$-s are nonzero, then the subspace $\text{span}(\mathbf{X}, \mathbf{Y})$ nontrivially intersects all the subspaces $\mathcal{N}(\mathtt{P}_i)$, $i = 1, \ldots, m$, violating (T1). Hence, for some index k we must have $\mathbf{C}_k = \mathbf{0}$. By (4.3), therefore, we have $\mathbf{Y} = \beta_k \mathbf{X}$, that is $\mathbf{Y}$ is equal to $\mathbf{X}$ up to scale. As $\mathbf{Y}$ is nonzero, β_k cannot be zero. □

Notice that for the classic case of projections $\mathbb{P}^3 \to \mathbb{P}^2$, (T1) simply means that the camera centres $\mathcal{N}(\mathtt{P}_i)$ and the projective point $\text{span}(\mathbf{X}) \in \mathbb{P}^3$ are collinear. For general dimensional projections, however, it is not trivial to show that (T1) is generically true. This is answered in the following proposition.

Proposition 4.2. *Consider a set of projection matrices* $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m$ *with* $\mathtt{P}_i \in \mathbb{R}^{s_i \times r}$ *such that* $\sum_{i=1}^{m}(s_i - 1) \geq r$, *and a* nonzero *point* $\mathbf{X} \neq 0$ *in* $\mathbb{R}^r$. *Now, if the null spaces* $\mathcal{N}(\mathtt{P}_1), \mathcal{N}(\mathtt{P}_2), \ldots, \mathcal{N}(\mathtt{P}_m)$ *as well as* $\text{span}(\mathbf{X})$ *are in general position (with* $\dim(\mathcal{N}(\mathtt{P}_i)) = r - s_i$*), then there is no linear subspace of dimension bigger than or equal to 2 passing through* $\mathbf{X}$ *and nontrivially intersecting* $\mathcal{N}(\mathtt{P}_1), \mathcal{N}(\mathtt{P}_2), \ldots, \mathcal{N}(\mathtt{P}_m)$.

4.1.2 An exchange lemma

The next lemma is similar to (but not the same as) the Steinitz exchange lemma. It plays a key role in our proofs.

Lemma 4.3 (Exchange Lemma). *Consider a set of* m linearly independent *vectors* $A = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m\} \subseteq \mathbb{R}^r$ *and a single vector* $\mathbf{b} \in \mathbb{R}^r$. *Define* A_i *as the set made by replacing* $\mathbf{a}_i$ *in* A *by* $\mathbf{b}$, *that is* $A_i = (A - \{\mathbf{a}_i\}) \cup \{\mathbf{b}\}$. *Now, given* $k \leq m$, *if for all* $i = 1, 2, \ldots, k$, *the vectors in* A_i *are linearly dependent, then* $\mathbf{b}$ *is in the span of* $\mathbf{a}_{k+1}, \ldots, \mathbf{a}_m$. *If* $k = m$ *then* $\mathbf{b} = \mathbf{0}$.

[1] Two linear subspaces nontrivially intersect if their intersection has dimension one or more.

Proof. As the vectors in A are linearly independent so are the vectors in $A - \{\mathbf{a}_i\}$. Therefore, if the vectors in $A_i = (A - \{\mathbf{a}_i\}) \cup \{\mathbf{b}\}$ are not linearly independent it means that $\mathbf{b}$ is in the span of $A - \{\mathbf{a}_i\}$, that is $\mathbf{b} = \sum_{j=1}^{m} c_{ji}\mathbf{a}_j$, where $c_{ii} = 0$. This can be shown as $\mathbf{b} = \mathtt{A}\,\mathbf{c}_i$ where $\mathtt{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m]$ and $\mathbf{c}_i = [c_{1i}, c_{2i}, \ldots c_{mi}]^T$, where the i-th element of each $\mathbf{c}_i$ is zero. According to the assumptions of the lemma we have

$$\mathbf{b}\mathbf{1}^T = \mathtt{A}\,[\mathbf{c}_1\ \mathbf{c}_2\ \cdots\ \mathbf{c}_k] \tag{4.4}$$

where the i-th element of each $\mathbf{c}_i$ is zero. As $\mathtt{A}$ has full column rank, we can write

$$[\mathbf{c}_1\ \mathbf{c}_2\ \cdots\ \mathbf{c}_k] = \mathbf{h}\,\mathbf{1}^T \tag{4.5}$$

where $\mathbf{h} = (\mathtt{A}^T\mathtt{A})^{-1}\mathbf{b}$. It means that all $\mathbf{c}_i$-s are equal. As the i-th element of each $\mathbf{c}_i$ is zero, it follows that the first k elements of all $\mathbf{c}_i$-s are zero. From $\mathbf{b} = \sum_{j=1}^{m} c_{ji}\mathbf{a}_j$ then it follows that $\mathbf{b} = \sum_{j=k+1}^{m} c_{ji}\mathbf{a}_j$, or $\mathbf{b} \in \text{span}(\mathbf{a}_{k+1}, \ldots, \mathbf{a}_m)$, and if $k = m$, it follows that $\mathbf{b} = \mathbf{0}$. □

Corollary 4.4. *Consider a* full-row-rank $p \times q$ *matrix* $\mathtt{Q}$ *partitioned as* $\mathtt{Q} = \begin{pmatrix}\mathtt{A}\\ \mathtt{B}\end{pmatrix}$, *and a horizontal vector* $\mathbf{q}^T$ *whose size is* q. *Now, if replacing any row of* $\mathtt{A}$ *by* $\mathbf{q}^T$ *turns* $\mathtt{Q}$ *into a rank deficient matrix, then* $\mathbf{q}$ *is in the row space of* $\mathtt{B}$. *If* $\mathtt{B}$ *has zero rows, that is* $\mathtt{Q} = \mathtt{A}$, *then* $\mathbf{q}^T$ *is zero.*

4.1.3 Valid profiles and the Grassmann tensor

Consider a set of projection matrices $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m$, with $\mathtt{P}_i \in \mathbb{R}^{s_i \times r}$, such that $\sum_{i=1}^{m}(s_i - 1) \geq r$. We define a *valid profile* [Hartley and Schaffalitzky, 2004] as an m-tuple of nonnegative[2] integers $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ such that $0 \leq \alpha_i \leq s_i - 1$ and $\sum \alpha_i = r$. Clearly, there might exist different valid profiles for a setup $\{\mathtt{P}_i\}$. One can choose $r \times r$ submatrices of $\mathtt{P} = \text{stack}(\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m)$ according to a profile $\boldsymbol{\alpha}$, by choosing α_i rows from each $\mathtt{P}_i$. Notice that due to the property $\alpha_i \leq s_i - 1$, never the whole rows of any $\mathtt{P}_i$ is chosen for building the submatrix.

The set of all $r \times r$ minors (determinant of $r \times r$ submatrices) of $\mathtt{P} = \text{stack}(\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m)$ form the Grassmann coordinates of the column space of $\mathtt{P}$. Here, however, we are only interested in a subset of these coordinates, namely those corresponding to a valid profile. Consider m index sets $I_1, I_2, \ldots, I_m$, such that each I_i contains the indices of α_i rows of $\mathtt{P}_i$. In other words, I_i is a subset of $\{1, 2, \ldots, s_i\}$ with α_i elements. Each way of choosing $I_1, I_2, \ldots, I_m$ gives a square submatrix of $\mathtt{P} = \text{stack}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$ where the rows of each $\mathtt{P}_i$ are chosen in order according to I_i. The determinant of this submatrix is multiplied by a corresponding sign[3] to form

[2]Notice that, the definition of a valid profile here slightly differs from that of [Hartley and Schaffalitzky, 2004] which needs $\alpha_i \geq 1$. We choose this new definition for convenience, as it does not impose the restriction $m \leq r$ on the number of views.

[3]The sign is defined by $\prod_{i=1}^{m} \text{sign}(I_i)$ where $\text{sign}(I_i)$ is $+1$ or -1 depending on whether the sequence $(\text{sort}(I_i)\,\text{sort}(\bar{I}_i))$ is an even or odd permutation for $\bar{I}_i = \{1, \ldots, s_i\} \setminus I_i$ (see [Hartley and Schaffalitzky, 2004]).

an entry of the Grassmann coordinate of $\mathtt{P} = \text{stack}(\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m)$, shown here by $\mathcal{T}_{\boldsymbol{\alpha}}^{I_1, I_2, \ldots, I_m}$. Such entries for different choices of the I_i-s can be arranged in a multidimensional array $\mathcal{T}_{\boldsymbol{\alpha}}$ called the Grassmann tensor corresponding to $\boldsymbol{\alpha}$. The dimension of $\mathcal{T}_{\boldsymbol{\alpha}}$ is equal to the number of nonzero entries of $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$, as $\mathcal{T}_{\boldsymbol{\alpha}}$ does not depend on those matrices $\mathtt{P}_i$ with $\alpha_i = 0$. To show the dependence of the Grassmann tensor on projection matrices $\mathtt{P}_i$, we sometimes use the mapping $\mathcal{G}_{\boldsymbol{\alpha}}$ which takes a set of projection matrices to the corresponding Grassmann tensor, that is $\mathcal{T}_{\boldsymbol{\alpha}} = \mathcal{G}_{\boldsymbol{\alpha}}(\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m)$. Notice that $\mathcal{G}_{\boldsymbol{\alpha}}$ itself is not a tensor. Obviously, $\mathcal{G}_{\boldsymbol{\alpha}}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$ is nonzero if and only if $\mathtt{P}$ has a non-singular submatrix chosen according to $\boldsymbol{\alpha}$.

Hartley and Schaffalitzky [2004] show that the Grassmann tensor encodes a relation between the corresponding image points in a subset of images. This is a multilinear relation between the Grassmann coordinates of subspaces with certain dimensions passing from each image point. To see this, consider a profile $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ for a set of projection matrices $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m$, with the extra condition that $\alpha_i \geq 1$ for all i. This can only be the case when the number of views is not more than r, that is $m \leq r$, as $\sum_{i=1}^{m} \alpha_i = r$ (If $m > r$ we consider a subset of views). For each view i consider an $s_i \times (s_i - \alpha_i)$ matrix $\mathtt{U}_i$ with linearly independent columns. Columns of $\mathtt{U}_i$ span a subspace of codimension α_i. Now, assume that there exists a *nonzero* point $\mathbf{X} \in \mathbb{R}^r$ projected via each $\mathtt{P}_i$ into a point on each of the associated subspaces $\mathtt{U}_i$. In other words, for each $\mathtt{P}_i$ there exists a vector $\mathbf{a}_i$ such that $\mathtt{U}_i \mathbf{a}_i = \mathtt{P}_i \mathbf{X}$. This can be written in the matrix form as

$$\begin{bmatrix} \mathtt{P}_1 & \mathtt{U}_1 & & & \\ \mathtt{P}_2 & & \mathtt{U}_2 & & \\ \vdots & & & \ddots & \\ \mathtt{P}_m & & & & \mathtt{U}_m \end{bmatrix} \begin{pmatrix} \mathbf{X} \\ -\mathbf{a}_1 \\ -\mathbf{a}_2 \\ \vdots \\ -\mathbf{a}_m \end{pmatrix} = \mathbf{0} \tag{4.6}$$

The matrix on the left is square (as its height is $\sum_{i=1}^{m} s_i$ and its width is $r + \sum_{i=1}^{m} (s_i - \alpha_i) = \sum_{i=1}^{m} s_i + r - \sum_{i=1}^{m} \alpha_i = \sum_{i=1}^{m} s_i$) and has non-trivial null space (as $\mathbf{X} \neq \mathbf{0}$) and hence a zero determinant. Consider m index set $I_1, I_2, \ldots, I_m$, where each I_i is a set with α_i members chosen from $\{1, 2, \ldots, s_i\}$. Also define $\bar{I}_i$ the complement of I_i with respect to the set $\{1, \ldots, s_i\}$, that is $\bar{I}_i = \{1, \ldots, s_i\} \setminus I_i$.

To compute the determinant of the matrix on the left hand side of (4.6), notice that for an $k \times k$ square matrix in the form $[\mathtt{A}, \mathtt{B}]$ with blocks $\mathtt{A} \in \mathbb{R}^{k \times s}$ and $\mathtt{B} \in \mathbb{R}^{k \times k - s}$, we have

$$\det([\mathtt{A}, \mathtt{B}]) = \sum_{|I| = s} \text{sign}(I) \det(\mathtt{A}^I) \det(\mathtt{B}^{\bar{I}}), \tag{4.7}$$

where I runs through all subsets of $\{1, \ldots, r\}$ of size s, $\bar{I}$ is $\{1, \ldots, r\} \setminus I$, $\mathtt{A}^I$ is the matrix created by choosing rows of $\mathtt{A}$ in order according to I and $\mathtt{B}^{\bar{I}}$ is defined similarly. The sign coefficient "sign(I)" is equal to $+1$ or -1 depending on whether the sequence sort(I) sort($\bar{I}$) is an even or odd permutation.

The matrix on the left hand side of (4.6), that is

$$\begin{bmatrix} \mathtt{P}_1 & \mathtt{U}_1 & & \\ \mathtt{P}_2 & & \mathtt{U}_2 & \\ \vdots & & & \ddots \\ \mathtt{P}_m & & & & \mathtt{U}_m \end{bmatrix} \tag{4.8}$$

can be written as $[\mathtt{A},\mathtt{B}]$ where $\mathtt{A} = \mathtt{P} = \text{stack}(\mathtt{P}_1,\mathtt{P}_2,\ldots,\mathtt{P}_m)$ and $\mathtt{B} = \text{diag}(\mathtt{U}_1,\mathtt{U}_2,\ldots,\mathtt{U}_m)$, where diag(.) makes a block diagonal matrix. Using (4.7), and the fact that (4.8) has a zero determinant, we obtain the following relation

$$\sum_{I_1,\ldots,I_m} \mathcal{T}_{\boldsymbol{\alpha}}^{I_1,I_2,\ldots,I_m} \det(\mathtt{U}_1^{\bar{I}_1}) \det(\mathtt{U}_2^{\bar{I}_2}) \cdots \det(\mathtt{U}_m^{\bar{I}_m}) = 0, \tag{4.9}$$

where $\mathtt{U}_i^{\bar{I}_i}$ is comprised of rows of $\mathtt{U}_i$ chosen according to $\bar{I}_i$, and

$$\mathcal{T}_{\boldsymbol{\alpha}}^{I_1,I_2,\ldots,I_m} = \left(\prod_{i=1}^{m} \text{sign}(I_i)\right) \det(\mathtt{P}^{I_1,I_2,\ldots,I_m}) \tag{4.10}$$

where $\det(\mathtt{P}^{I_1,I_2,\ldots,I_m})$ shows the minor of $\mathtt{P}$ made by choosing rows α_i rows from each $\mathtt{P}_i$ according to I_i. From 4.10, it is obvious that the coefficients $\mathcal{T}_{\boldsymbol{\alpha}}^{I_1,I_2,\ldots,I_m}$ form the elements of the Grassmann tensor $\mathcal{T}_{\boldsymbol{\alpha}}$ defined at the beginning of this subsection.

Notice that in (4.9), for each i, the quantities $\det(\mathtt{U}_i^{\bar{I}_i})$ for different choices of $\bar{I}_i$ form the Grassmann coordinates of the subspace $U_i = \mathcal{C}(\mathtt{U}_i)$, the column space of $\mathtt{U}_i$. The main theorem of [Hartley and Schaffalitzky, 2004] states that the projection matrices $\mathtt{P}_i$ can be uniquely constructed from the Grassmann tensor, up to projectivity:

Theorem 4.1 ([Hartley and Schaffalitzky, 2004]). *Consider a set of m generic projection matrices $\mathtt{P}_1,\mathtt{P}_2,\ldots,\mathtt{P}_m$, with $\mathtt{P}_i \in \mathbb{R}^{s_i \times r}$, such that $m \leq r \leq \sum_i s_i - m$, and an m-tuple $(\alpha_1,\alpha_2,\ldots,\alpha_m)$ of integers α_i such that $1 \leq \alpha_i \leq m-1$ for all i and $\sum_{i=1}^{m} \alpha_i = r$. Then if at least for one i we have $s_i \geq 3$, the matrices $\mathtt{P}_i$ are determined up to a projective ambiguity from the set of minors of the matrix $\mathtt{P} = \text{stack}(\mathtt{P}_1,\mathtt{P}_2,\ldots,\mathtt{P}_m)$ chosen with α_i rows from each $\mathtt{P}_i$ (that is the elements of the Grassmann tensor). If $s_i = 2$ for all i, there are two equivalence classes of solutions.*

The constructive proof given by Hartley and Schaffalitzky [2004] provides a procedure to construct the projection matrices $\mathtt{P}_i$ from the Grassmann tensor. From each set of image point correspondences $\mathbf{x}_{1j},\mathbf{x}_{2j},\ldots,\mathbf{x}_{mj}$ different sets of subspaces $U_1,U_2,\ldots,U_m$ can be passed such that $\mathbf{x}_{ij} \in U_i$. Each choice of subspaces $U_1,\ldots,U_m$ gives a linear equation (4.9) on the elements of the Grassmann tensor. The Grassmann tensor can be obtained as the null vector of the matrix of coefficients of the resulting set of linear equations[4].

[4] In Sect. 4.2.1 we prove that the Grassmann tensor is unique, meaning that the matrix of coefficients of these linear equations has a 1D null space.

The next lemma will be used in the proof of projective reconstruction for arbitrarily large number of views. It implies that if a nonzero Grassmann tensor is found for a subset of views, then we can find a nonzero Grassmann tensors for other subsets of views, such that the whole set of views finally is spanned by these subsets.

Lemma 4.5. *Consider a set of projection matrices* $\mathtt{P}_1,\ldots,\mathtt{P}_m$ *with* $\mathtt{P}_i \in \mathbb{R}^{s_i\times r}$ *and* $\mathtt{P}_i \neq 0$ *for all* i. *Assume that there exists a valid profile* $\boldsymbol{\alpha} = (\alpha_1,\alpha_2,\ldots,\alpha_m)$ *with* $\alpha_k = 0$ *such that* $\mathcal{G}_{\boldsymbol{\alpha}}(\mathtt{P}_1,\ldots,\mathtt{P}_m)$ *is nonzero. Then there exists a valid profile* $\boldsymbol{\alpha}' = (\alpha'_1,\alpha'_2,\ldots,\alpha'_m)$ *with* $\alpha'_k > 0$ *such that* $\mathcal{G}_{\boldsymbol{\alpha}'}(\mathtt{P}_1,\ldots,\mathtt{P}_m)$ *is nonzero.*

We remind the reader that, for a set of projection matrices $\mathtt{P}_1,\ldots,\mathtt{P}_m$ with $\mathtt{P}_i \in \mathbb{R}^{s_i\times r}$, a profile $\boldsymbol{\alpha} = (\alpha_1,\alpha_2,\ldots,\alpha_m)$ is valid if $\sum_{i=1}^m \alpha_i = r$, and further, for all i we have $\alpha_i \leq s_i - 1$.

Proof. Consider an invertible $r\times r$ submatrix $\mathtt{Q}$ of $\mathtt{P} = \mathrm{stack}(\mathtt{P}_1,\ldots,\mathtt{P}_m)$ chosen according to $\boldsymbol{\alpha}$, with α_i rows chosen from each $\mathtt{P}_i$. As $\alpha_k = 0$, no row of $\mathtt{Q}$ is chosen among rows of $\mathtt{P}_k$. Now, as $\mathtt{P}_k \neq 0$ it has at least one *nonzero* row $\mathbf{p}^T$. Show by $\mathtt{Q}_i$ the matrix $\mathtt{Q}$ whose i-th row has been replaced by $\mathbf{p}^T$. Now, at least for one i the matrix $\mathtt{Q}_i$ must have full rank, because otherwise, according to Corollary 4.4, $\mathbf{p}^T$ would be zero. Assume that the i-th row of $\mathtt{Q}$ has been chosen from $\mathtt{P}_l$. This implies $\alpha_l > 0$. It is easy to check that $\mathtt{Q}_i$ is an $r\times r$ submatrix of $\mathtt{P}$ chosen according to a profile $\boldsymbol{\alpha}' = (\alpha'_1,\alpha'_2,\ldots,\alpha'_m)$ for which $\alpha'_k = 1$, $\alpha'_l = \alpha_l - 1 \geq 0$, and $\alpha'_i = \alpha_i$ for all i other than k and l. This shows that $\boldsymbol{\alpha}'$ is a valid profile. Moreover, the tensor $\mathcal{G}_{\boldsymbol{\alpha}'}(\mathtt{P}_1,\ldots,\mathtt{P}_m)$ is nonzero as it has at least one nonzero element $\det(\mathtt{Q}_i)$. □

4.2 Projective Reconstruction

Here, we state one version of the projective reconstruction theorem, proving the projective equivalence of two configurations $(\{\mathtt{P}_i\},\{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\},\{\hat{\mathbf{X}}_j\})$ projecting into the same image points, given conditions on $(\{\hat{\mathtt{P}}_i\},\{\hat{\mathbf{X}}_j\})$. In the next section, based on this theorem, we present an alternative theorem with conditions on the projective depths $\hat{\lambda}_{ij}$.

Theorem 4.2 (Projective Reconstruction). *Consider a configuration of* m *projection matrices and* n *points* $(\{\mathtt{P}_i\},\{\mathbf{X}_j\})$ *where the matrices* $\mathtt{P}_i \in \mathbb{R}^{s_i\times r}$ *are* generic, $\sum_{i=1}^m (s_i - 1) \geq r$, *and* $s_i \geq 3$ *for all views*[5], *and the points* $\mathbf{X}_j \in \mathbb{R}^r$ *are* sufficiently many *and* in general position. *Given a second configuration* $(\{\hat{\mathtt{P}}_i\},\{\hat{\mathbf{X}}_j\})$ *that satisfies*

$$\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j \tag{4.11}$$

for some scalars $\{\hat{\lambda}_{ij}\}$, *if*

(C1) $\hat{\mathbf{X}}_j \neq \mathbf{0}$ *for all* j, *and*

[5]We could have assumed the milder condition of $s_i \geq 3$ for at least one i. Our assumption, however, avoids unnecessary complications.

(C2) $\hat{\mathtt{P}}_i \neq \mathtt{0}$ *for all* i*, and*

(C3) there exists at least one non-singular $r \times r$ *submatrix* $\hat{\mathtt{Q}}$ *of* $\hat{\mathtt{P}} = \text{stack}(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_2, \ldots, \hat{\mathtt{P}}_m)$ *containing strictly fewer than* s_i *rows from each* $\mathtt{P}_i$*. (equivalently* $\mathcal{G}_{\boldsymbol{\alpha}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m) \neq \mathbf{0}$ *for some valid profile* $\boldsymbol{\alpha}$*),*

then the two configurations $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ *and* $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ *are projectively equivalent.*

It is important to observe the theorem does not assume a priori that the projective depths $\hat{\lambda}_{ij}$ are nonzero. At a first glance, this theorem might seem to be of no use, especially because condition (C3) looks hard to verify for a given setup $\{\hat{\mathtt{P}}_i\}$. But, this theorem is important as it forms the basis of our theory, by giving the minimal required conditions on the setup $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$, from which simpler necessary conditions can be obtained.

Overview of the proof of Theorem 4.2 is as follows. Given the profile $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ from condition (C3),

1. for the special case of $\alpha_i \geq 1$ for all i, we prove that the Grassmann tensors $\mathcal{G}_{\boldsymbol{\alpha}}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$ and $\mathcal{G}_{\boldsymbol{\alpha}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m)$ are equal up to a scaling factor, (Sect. 4.2.1).
2. Using the theory of Hartley and Schaffalitzky [2004], we show that $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent for the special case of $\alpha_i \geq 1$ for all i, (Sect. 4.2.2).
3. We prove the theorem for the general case where some of α_i-s might be zero, and hence the number of views can be arbitrarily large, (Sect. 4.2.3).

4.2.1 The uniqueness of the Grassmann tensor

The main purpose of this subsection is to show that if $\hat{\mathbf{X}}_j \neq \mathbf{0}$ for all j, the relations $\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j$ imply that the Grassmann tensor $\mathcal{G}_{\boldsymbol{\alpha}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m)$ is equal to $\mathcal{G}_{\boldsymbol{\alpha}}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$ up to a scaling factor. This implies that the Grassmann tensor is unique up to scale given a set of image points $\mathbf{x}_{ij}$ obtained from $\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j/\lambda_{ij}$ with $\lambda_{ij} \neq 0$.

Theorem 4.3. *Consider a setup* $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ *of* m generic *projection matrices, and* n *points* in general position and sufficiently many, *and a valid profile* $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$*, meaning* $\sum_{i=1}^{m} \alpha_i = r$ *and* $\alpha_i \leq s_i - 1$*, such that* $\alpha_i \geq 1$ *for all* i*. Now, for any other configuration* $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ *with* $\hat{\mathbf{X}}_j \neq \mathbf{0}$ *for all* j*, the set of relations*

$$\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j \tag{4.12}$$

implies $\mathcal{G}_{\boldsymbol{\alpha}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m) = \beta\, \mathcal{G}_{\boldsymbol{\alpha}}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$ *for some scalar* β*.*

Notice that it has not been assumed that the estimated depths $\hat{\lambda}_{ij}$ are nonzero. In this section we only give the idea of the proof. The formal proof is given in Sect. 4.5.2.

We consider two submatrices $\mathtt{Q}$ and $\mathtt{Q}'$ of $\mathtt{P} = \text{stack}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$ chosen according to the valid profile $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$, such that all rows of $\mathtt{Q}$ and $\mathtt{Q}'$ are equal except

for the l-th rows $\mathbf{q}_l^T$ and $\mathbf{q}_l'^T$, which are chosen from different rows of $\mathtt{P}_k$. We also represent by $\hat{\mathtt{Q}}$ and $\hat{\mathtt{Q}}'$ the corresponding submatrices of $\hat{\mathtt{P}} = \text{stack}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m)$. Then we show that if $\det(\mathtt{Q}) \neq 0$, the equations $\hat{\mathtt{P}}_i \hat{\mathbf{X}}_j = \hat{\lambda}_{ij} \mathtt{P}_i \mathbf{X}_j$ imply

$$\det(\hat{\mathtt{Q}}') = \frac{\det(\mathtt{Q}')}{\det(\mathtt{Q})} \det(\hat{\mathtt{Q}}). \tag{4.13}$$

The rest of the proof is as follows: By starting with a submatrix $\mathtt{Q}$ of $\mathtt{P}$ according to $\boldsymbol{\alpha}$, and iteratively updating $\mathtt{Q}$ by changing one row at a time in the way described above, we can finally traverse all possible submatrices chosen according to $\boldsymbol{\alpha}$. Due to genericity we assume that all submatrices of $\mathtt{P}$ chosen according to $\boldsymbol{\alpha}$ are non-singular[6]. Therefore, (4.89) implies that during the traversal procedure the ratio $\beta = \det(\hat{\mathtt{Q}})/\det(\mathtt{Q})$ stays the same. This means that each element of $\mathcal{G}_{\boldsymbol{\alpha}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m)$ is β times the corresponding element of $\mathcal{G}_{\boldsymbol{\alpha}}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$, implying $\mathcal{G}_{\boldsymbol{\alpha}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m) = \beta\, \mathcal{G}_{\boldsymbol{\alpha}}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$.

The relation (4.13) is obtained in two steps. The first step is to write equations (4.12), that is $\hat{\mathtt{P}}_i \hat{\mathbf{X}}_j = \hat{\lambda}_{ij} \mathtt{P}_i \mathbf{X}_j$, in matrix form as

$$\mathcal{M}(\mathbf{X}_j) \begin{pmatrix} \hat{\boldsymbol{\lambda}}_j \\ \hat{\mathbf{X}}_j \end{pmatrix} = \mathbf{0}, \qquad j = 1, 2, \ldots, n, \tag{4.14}$$

where $\hat{\boldsymbol{\lambda}}_j = [\hat{\lambda}_{1j}, \ldots, \hat{\lambda}_{mj}]^T$, and

$$\mathcal{M}(\mathbf{X}) = \begin{bmatrix} \mathtt{P}_1\mathbf{X} & & & & \hat{\mathtt{P}}_1 \\ & \mathtt{P}_2\mathbf{X} & & & \hat{\mathtt{P}}_2 \\ & & \ddots & & \vdots \\ & & & \mathtt{P}_m\mathbf{X} & \hat{\mathtt{P}}_m \end{bmatrix}. \tag{4.15}$$

The matrix $\mathcal{M}(\mathbf{X})$ is $(\sum_i s_i) \times (m+r)$, and therefore a tall (or square) matrix. Due to the assumption $\hat{\mathbf{X}}_j \neq \mathbf{0}$ in Theorem 4.3, we conclude that $\mathcal{M}(\mathbf{X}_j)$ is rank deficient for all $\mathbf{X}_j$. Then, considering the fact that $\mathcal{M}(\mathbf{X})$ is rank deficient for sufficiently many points $\mathbf{X}_j$ in general position, we show that $\mathcal{M}(\mathbf{X})$ is rank deficient for all $\mathbf{X} \in \mathbb{R}^r$. Therefore, for all $(m+r) \times (m+r)$ submatrices $\mathcal{M}'(\mathbf{X})$ of $\mathcal{M}(\mathbf{X})$ we have $\det(\mathcal{M}'(\mathbf{X})) = 0$.

The second step is to choose a proper value for $\mathbf{X}$ and a proper submatrix $\mathcal{M}'(\mathbf{X})$ of $\mathcal{M}(\mathbf{X})$, such that (4.13) follows from $\det(\mathcal{M}'(\mathbf{X})) = 0$. This proper value for $\mathbf{X}$ is $\mathtt{Q}^{-1}\mathbf{e}_l$, where $\mathbf{e}_l$ is the l-th standard basis and l is the row which is different in $\mathtt{Q}$ and $\mathtt{Q}'$, as defined above. The submatrix $\mathcal{M}'(\mathbf{X})$, is made by choosing the corresponding rows of $\mathtt{P} = \text{stack}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$ contributing to making $\mathtt{Q}$, choosing the corresponding row $\mathbf{q}_l'^T$ of $\mathtt{P}_k$ contributing to making $\mathtt{Q}'$, and choosing one extra row form each $\mathtt{P}_i$ for $i \neq k$. See Sect. 4.5.2 for more details.

[6]Although the proof is possible under a slightly milder assumption.

4.2.2 Proof of reconstruction for the special case of $\alpha_i \geq 1$

Lemma 4.6. *Theorem 4.2 is true for the special case of $\alpha_i \geq 1$ for all i.*

The steps of the proof are: Given the $\boldsymbol{\alpha}$ introduced in condition (C3) of Theorem 4.2, Theorem 4.3 tells $\mathcal{G}_{\boldsymbol{\alpha}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m) = \beta \mathcal{G}_{\boldsymbol{\alpha}}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$. From (C3) it follows that $\beta \neq 0$. Thus, Theorem 4.1 (proved by Hartley and Schaffalitzky [2004]), suggests that $\{\mathtt{P}_i\}$ and $\{\hat{\mathtt{P}}_i\}$ are projectively equivalent. Then, using the Triangulation Lemma 4.1, we prove that $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent. Next comes the formal proof.

Proof. From Theorem 4.3 we know that $\mathcal{G}_{\boldsymbol{\alpha}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m) = \beta \mathcal{G}_{\boldsymbol{\alpha}}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$ for some scalar β. From condition (C3) in Theorem 4.2 we conclude that β is nonzero. Thus, using the main theorem of [Hartley and Schaffalitzky, 2004] (restated here as Theorem 4.1 in Sect. 4.1.3), we can conclude that the two set of projection matrices $\{\mathtt{P}_i\}$ and $\{\hat{\mathtt{P}}_i\}$ are projectively equivalent. Thus, there exists an invertible matrix $\mathtt{H}$ and nonzero scalars $\tau_1, \tau_2, \ldots, \tau_m$ such that

$$\hat{\mathtt{P}}_i = \tau_i \mathtt{P}_i \mathtt{H} \tag{4.16}$$

for $i = 1, \ldots, m$. Now, from $\hat{\mathtt{P}}_i \hat{\mathbf{X}}_j = \hat{\lambda}_{ij} \mathtt{P}_i \mathbf{X}_j$ and (4.16) for each j we have

$$\mathtt{P}_i(\mathtt{H}\hat{\mathbf{X}}_j) = \frac{\hat{\lambda}_{ij}}{\tau_i} \mathtt{P}_i \mathbf{X}_j \tag{4.17}$$

As $\hat{\mathbf{X}}_j \neq \mathbf{0}$, $\mathtt{H}$ is invertible, $\mathtt{P}_i$-s are generic and $\mathbf{X}_j$ is in general position, using the triangulation Lemma 4.1 we have $(\mathtt{H}\hat{\mathbf{X}}_j) = \nu_j \mathbf{X}_j$ for some nonzero scalar $\nu_j \neq 0$, which gives

$$\hat{\mathbf{X}}_j = \nu_j \mathtt{H}^{-1} \mathbf{X}_j. \tag{4.18}$$

The above is true for $j = 1, \ldots, m$. From (4.16) and (4.18) it follows that the two configurations $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent. □

4.2.3 Proof of reconstruction for general case

To prove Theorem 4.2 in the general case, where we might have $\alpha_i = 0$ for some elements of the valid profile $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$, given in condition (C3) of the theorem, we proceed as follows: By (C3) we have $\mathcal{G}_{\boldsymbol{\alpha}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m) \neq \mathbf{0}$, by Lemma 4.5, for each view k, there exists a valid profile $\boldsymbol{\alpha}^{(k)}$ for which $\alpha_k^{(k)} \geq 1$ and the Grassmann tensor $\mathcal{G}_{\boldsymbol{\alpha}^{(k)}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m)$ is nonzero. Define $I_k = \{i \,|\, \alpha_i^{(k)} \geq 1\}$. Lemma (4.6) proves for each I_k that the configurations $(\{\mathtt{P}_i\}_{I_k}, \{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\}_{I_k}, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent. As $\cup_k I_k = \{1, \ldots, m\}$, using Lemma 2.1 we show the projective equivalence holds for the whole set of views, that is $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$. The formal proof is as follows.

Proof. According to (C3), there exists a valid profile $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ such that $\mathcal{G}_{\boldsymbol{\alpha}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m) \neq \mathbf{0}$. Hence, by Lemma 4.5 we can say that for each view k, there exists a valid profile $\boldsymbol{\alpha}^{(k)}$ for which $\alpha_k^{(k)} \geq 1$ and the corresponding Grassmann tensor $\mathcal{G}_{\boldsymbol{\alpha}^{(k)}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m)$ is nonzero. Define $I_k = \{i \,|\, \alpha_i^{(k)} \geq 1\}$. Lemma 4.6 proves that for each k the configurations $(\{\mathtt{P}_i\}_{I_k}, \{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\}_{I_k}, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent. Therefore, for each k we have

$$\hat{\mathtt{P}}_i = \tau_i^k \, \mathtt{P}_i \mathtt{H}_k^{-1}, \qquad i \in I_k \tag{4.19}$$

$$\hat{\mathbf{X}}_j = \nu_j^k \, \mathtt{H}_k \mathbf{X}_j, \qquad j = 1, \ldots, n \tag{4.20}$$

for nonzero scalars $\{\tau_i^k\}_{i \in I_k}$ and $\{\nu_j^k\}$, and the invertible matrix $\mathtt{H}_k$. Now, from relations (4.20) for different values of k, using Lemma 2.1 we can conclude that, by possibly rescaling the matrix $\mathtt{H}_k$ and accordingly the scalars ν_j^k (and also τ_i^k) for each k, we can have the matrix $\mathtt{H}$ and scalars $\nu_1, \nu_2, \ldots, \nu_m$, such that $\mathtt{H}_k = \mathtt{H}$ and $\nu_j^k = \nu_j$ for all k. Therefore, (4.19) and (4.20) become

$$\hat{\mathtt{P}}_i = \tau_i^k \, \mathtt{P}_i \mathtt{H}^{-1}, \qquad i \in I_k \tag{4.21}$$

$$\hat{\mathbf{X}}_j = \nu_j \, \mathtt{H} \mathbf{X}_j, \qquad j = 1, \ldots, n \tag{4.22}$$

Now, as $\mathtt{P}_i\mathtt{H}^{-1} \neq 0$ (since $\mathtt{P}_i \neq 0$ and $\mathtt{H}^{-1}$ is invertible), (4.21) implies that for each i all scalars τ_i^k have a common value τ_i. This gives

$$\hat{\mathtt{P}}_i = \tau_i \, \mathtt{P}_i \mathtt{H}^{-1}, \qquad i \in I_k, \quad k = 1, \ldots, m \tag{4.23}$$

$$\hat{\mathbf{X}}_j = \nu_j \, \mathtt{H} \mathbf{X}_j, \qquad j = 1, \ldots, n \tag{4.24}$$

As $\cup_k I_k = \{1, 2, \ldots, m\}$, the above suggests that $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ and $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ are projectively equivalent. □

4.3 Restricting projective depths

This section provides a second version of Theorem 4.2 in which it is assumed that $\hat{\lambda}_{ij}$-s are all nonzero, instead of putting restrictions on $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$.

Theorem 4.4 (Projective Reconstruction). *Consider a configuration of m projection matrices and n points $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ where the matrices $\mathtt{P}_i \in \mathbb{R}^{s_i \times r}$ are* generic *and as many such that $\sum_{i=1}^m (s_i - 1) \geq r$, and $s_i \geq 3$ for all views, and the points $\mathbf{X}_j \in \mathbb{R}^r$ are* sufficiently many *and* in general position. *Now, for any second configuration $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ satisfying*

$$\hat{\mathtt{P}}_i \hat{\mathbf{X}}_j = \hat{\lambda}_{ij} \mathtt{P}_i \mathbf{X}_j. \tag{4.25}$$

for nonzero *scalars $\hat{\lambda}_{ij} \neq 0$, the configuration $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ is projectively equivalent to $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$.*

The condition $\hat{\lambda}_{ij} \neq 0$ is not tight, and used here to avoid complexity. In Sect. 4.4 we will discuss that the theorem can be proved under milder restrictions. However,

by proving projective equivalence, it eventually follows that all $\hat{\lambda}_{ij}$-s are nonzero. We prove the theorem after giving required lemmas.

Lemma 4.7. *Consider m projection matrices $\hat{\mathrm{P}}_1, \hat{\mathrm{P}}_2, \ldots, \hat{\mathrm{P}}_m$ with $\hat{\mathrm{P}}_i \in \mathbb{R}^{s_i \times r}$, such that $\sum_{i=1}^{n}(s_i - 1) \geq r$, and $\hat{\mathrm{P}} = \mathrm{stack}(\hat{\mathrm{P}}_1, \ldots, \hat{\mathrm{P}}_m)$ has* full column rank r. *If $\hat{\mathrm{P}}$ has no full rank $r \times r$ submatrix chosen by strictly fewer than s_i rows form each $\hat{\mathrm{P}}_i$, then there exists a partition $\{I, J, K\}$ of the set of views $\{1, 2, \ldots, m\}$, with $I \neq \emptyset$ (nonempty) and $\sum_{i \in I} s_i + \sum_{i \in J}(s_i - 1) \leq r$, such that $\hat{\mathrm{P}}^K = \mathrm{stack}(\{\hat{\mathrm{P}}_i\}_{i \in K})$ has rank $r' = r - \sum_{i \in I} s_i - \sum_{i \in J}(s_i - 1)$. Further, the row space of $\hat{\mathrm{P}}^K$ is spanned by the rows of an $r' \times r$ submatrix $\hat{\mathrm{Q}}^K = \mathrm{stack}(\{\hat{\mathrm{Q}}_i\}_{i \in K})$ of $\hat{\mathrm{P}}^K$, where each $\hat{\mathrm{Q}}_i$ is created by choosing strictly less than s_i rows from $\hat{\mathrm{P}}_i$.*

The proof is based on taking a full-rank $r \times r$ submatrix $\hat{\mathrm{Q}}$ of $\hat{\mathrm{P}}$, and trying to replace some of its rows with other rows of $\hat{\mathrm{P}}$, while keeping the resulting submatrix full-rank, so as to reduce the number of matrices $\hat{\mathrm{P}}_i$ whose whole rows are included in $\hat{\mathrm{Q}}$. By this process, we can never have a case where no $\hat{\mathrm{P}}_i$ contributes all of its rows in the resulting full-rank submatrix, as otherwise, we would have a submatrix chosen by less than s_i rows from each $\hat{\mathrm{P}}_i$. Studying consequences of this fact leads to the conclusion of the lemma. The proof is given in Sect. 4.5.3.

Lemma 4.8. *Under the conditions of Theorem 4.4, if the matrix $\hat{\mathrm{P}} = \mathrm{stack}(\hat{\mathrm{P}}_1, \hat{\mathrm{P}}_2, \ldots, \hat{\mathrm{P}}_m)$ has full column rank, it has a non-singular $r \times r$ submatrix chosen with strictly fewer than s_i rows from each $\hat{\mathrm{P}}_i \in \mathbb{R}^{s_i \times r}$.*

Proof. To get a contradiction, assume that $\hat{\mathrm{P}}$ does not have any full-rank $r \times r$ submatrix created with strictly fewer than s_i rows from each $\hat{\mathrm{P}}_i$. Then by Lemma 4.7, there exists a partition $\{I, J, K\}$ of views $\{1, 2, \ldots, m\}$, with $I \neq \emptyset$ and $\sum_{i \in I} s_i + \sum_{i \in J}(s_i - 1) \leq r$, such that $\hat{\mathrm{P}}^K = \mathrm{stack}(\{\hat{\mathrm{P}}_i\}_{i \in K})$ has a row space of dimension

$$r' = r - \sum_{i \in I} s_i - \sum_{i \in J}(s_i - 1),$$

spanned by the rows of an $r' \times r$ matrix $\hat{\mathrm{Q}}^K = \mathrm{stack}(\{\hat{\mathrm{Q}}_i\}_{i \in K})$, where each $\hat{\mathrm{Q}}_i$ consists of strictly less than s_i rows from $\hat{\mathrm{P}}_i$. By rearranging the rows of $\hat{\mathrm{P}}_i$-s if necessary, we can assume that

$$\hat{\mathrm{P}}_i = \begin{bmatrix} \hat{\mathrm{Q}}_i \\ \hat{\mathrm{R}}_i \end{bmatrix} \tag{4.26}$$

for all $i \in K$, where $\hat{\mathrm{R}}_i$ consists of rows of $\hat{\mathrm{P}}_i$ not chosen for the creation of $\hat{\mathrm{Q}}^K$. We do not rule out the possibility that for some $i \in K$ no row of $\hat{\mathrm{P}}_i$ is contained in $\hat{\mathrm{Q}}^K$ (that is $\hat{\mathrm{P}}_i = \hat{\mathrm{R}}_i$). In this case one can think of $\hat{\mathrm{Q}}_i$ as a matrix with zero rows. Notice that, as $\hat{\mathrm{Q}}_i$ consists of strictly fewer than s_i rows of $\hat{\mathrm{P}}_i$, each $\hat{\mathrm{R}}_i$ must have at least one row. By relabeling the views if necessary, we assume that $K = \{1, 2, \ldots, l\}$ (thus

$I \cup J = \{l+1, \ldots, m\}$). Then, we have

$$\hat{\mathtt{P}}^K = \text{stack}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_l),$$
$$\hat{\mathtt{Q}}^K = \text{stack}(\hat{\mathtt{Q}}_1, \ldots, \hat{\mathtt{Q}}_l),$$
$$\hat{\mathtt{R}}^K = \text{stack}(\hat{\mathtt{R}}_1, \ldots, \hat{\mathtt{R}}_l).$$

As rows of $\hat{\mathtt{Q}}^K$ span the row space of $\hat{\mathtt{P}}^K$, and thus, the row space of $\hat{\mathtt{R}}^K$, we have $\hat{\mathtt{R}}^K = \mathtt{A}\,\hat{\mathtt{Q}}^K$ for some matrix $\mathtt{A}$ with r' columns. From (4.25), we have $\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j$ and, as a result

$$\hat{\mathtt{Q}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\,\mathtt{Q}_i\mathbf{X}_j \tag{4.27}$$
$$\hat{\mathtt{R}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\,\mathtt{R}_i\mathbf{X}_j \tag{4.28}$$

where $\mathtt{Q}_i$ (resp. $\mathtt{R}_i$) is the submatrix of $\mathtt{P}_i$ corresponding to $\hat{\mathtt{Q}}_i$ (resp. $\hat{\mathtt{R}}_i$), which means $\text{stack}(\mathtt{Q}_i, \mathtt{R}_i) = \mathtt{P}_i$. This gives

$$\hat{\mathtt{Q}}^K\,\hat{\mathbf{X}}_j = \text{diag}(\mathtt{Q}_1\mathbf{X}_j, \mathtt{Q}_2\mathbf{X}_j, \ldots, \mathtt{Q}_l\mathbf{X}_j)\,\hat{\boldsymbol{\lambda}}_j^K \tag{4.29}$$
$$\hat{\mathtt{R}}^K\,\hat{\mathbf{X}}_j = \text{diag}(\mathtt{R}_1\mathbf{X}_j, \mathtt{R}_2\mathbf{X}_j, \ldots, \mathtt{R}_l\mathbf{X}_j)\,\hat{\boldsymbol{\lambda}}_j^K \tag{4.30}$$

where diag(.) makes a block diagonal matrix out of its arguments, and $\hat{\boldsymbol{\lambda}}_j^K = [\hat{\lambda}_{1j}, \ldots, \hat{\lambda}_{lj}]^T$. From $\hat{\mathtt{R}}^K = \mathtt{A}\hat{\mathtt{Q}}^K$, then we have

$$\mathcal{M}(\mathbf{X}_j)\,\hat{\boldsymbol{\lambda}}_j^K = \mathbf{0}, \tag{4.31}$$

where

$$\mathcal{M}(\mathbf{X}) = \text{diag}(\mathtt{R}_1\mathbf{X}, \mathtt{R}_2\mathbf{X}, \ldots, \mathtt{R}_l\mathbf{X}) - \mathtt{A}\,\text{diag}(\mathtt{Q}_1\mathbf{X}, \mathtt{Q}_2\mathbf{X}, \ldots, \mathtt{Q}_l\mathbf{X}). \tag{4.32}$$

Clearly, $\mathcal{M}(\mathbf{X})$ has l columns, and since each $\mathtt{R}_i$ has at least one row, $\mathcal{M}(\mathbf{X})$ has at least l rows. Hence, it is a tall (or square) matrix. As $\hat{\boldsymbol{\lambda}}_j^K \neq \mathbf{0}$ (since $\hat{\lambda}_{ij} \neq 0$ for all i, j), $\mathcal{M}(\mathbf{X}_j)\hat{\boldsymbol{\lambda}}_j^K = \mathbf{0}$ implies that $\mathcal{M}(\mathbf{X}_j)$ is rank deficient. Since $\mathcal{M}(\mathbf{X})$ is rank-deficient at sufficiently many points $\mathbf{X}_j$ in general position, with the same argument as given in the proof of Lemma 4.10, we conclude that for all $\mathbf{X} \in \mathbb{R}^r$ the matrix $\mathcal{M}(\mathbf{X})$ is rank-deficient[7]. As $\hat{\mathtt{Q}}^K$ is $r' \times r$ with $r' < r$ and the matrices $\mathtt{P}_i = \text{stack}(\mathtt{Q}_i, \mathtt{R}_i)$ are generic, we can take a nonzero vector $\mathbf{Y}$ in the null space of $\hat{\mathtt{Q}}^K = \text{stack}(\hat{\mathtt{Q}}_1, \ldots, \hat{\mathtt{Q}}_l)$ such that no matrix $\hat{\mathtt{R}}_i$ for $i = 1, \ldots, l$ has $\mathbf{Y}$ in its null space[8]. In this case, we have $\mathtt{Q}_i\mathbf{Y} = 0$ for all i, implying $\mathcal{M}(\mathbf{Y}) = \text{diag}(\mathtt{R}_1\mathbf{Y}, \ldots, \mathtt{R}_l\mathbf{Y})$. Now, from $\mathbf{Y} \notin \mathcal{N}(\hat{\mathtt{R}}_i)$, we have $\mathtt{R}_i\mathbf{Y} \neq 0$ for $i = 1, \ldots, l$. This implies that $\mathcal{M}(\mathbf{Y}) = \text{diag}(\mathtt{R}_1\mathbf{Y}, \ldots, \mathtt{R}_l\mathbf{Y})$ has full column rank,

[7] In short, the argument goes as follows: The determinant of every $l \times l$ submatrix of $\mathcal{M}(\mathbf{X}_j)$ is zero for all j. Since the determinant of each submatrix is a polynomial expression on $\mathbf{X}_j$, each polynomial being zero for sufficiently many $\mathbf{X}_j$-s in general position imply that it is identically zero. This means that for every $\mathbf{X}$ all submatrices of $\mathcal{M}(\mathbf{X})$ have a zero determinant, and hence, $\mathcal{M}(\mathbf{X})$ is rank deficient.

[8] $\mathbf{Y}$ must be chosen from $\mathcal{N}(\mathtt{Q}^K) \setminus \cup_{i=1}^{l}\mathcal{N}(\mathtt{R}^K)$ which is nonempty (in fact open and dense in $\mathcal{N}(\mathtt{Q}^K)$) for generic $\mathtt{P}_i$-s.

contradicting the fact that $\mathcal{M}(\mathbf{X})$ is rank deficient for all $\mathbf{X}$. □

Proof of Theorem 4.4. Using Theorem 4.2 we just need to prove that the condition $\hat{\lambda}_{ij} \neq 0$ imply conditions (C1-C3) of Theorem 4.2. Assume that $\hat{\lambda}_{ij} \neq 0$ for some i and j, then from the genericity of $\mathtt{P}_i$ and $\mathbf{X}_j$ we have $\mathtt{P}_i\mathbf{X}_j \neq 0$, and thus $\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j \neq 0$, implying $\hat{\mathtt{P}}_i \neq 0$ and $\hat{\mathbf{X}}_j \neq \mathbf{0}$. This means that $\hat{\lambda}_{ij} \neq 0$ for all i and j imply (C1) and (C2). Now, it is left to show that $\hat{\lambda}_{ij} \neq 0$ imply (C3), that is $\hat{\mathtt{P}}$ has a full-rank $r \times r$ submatrix chosen with strictly fewer than s_i rows from each $\hat{\mathtt{P}}_i$. This is proved in Lemma 4.8 for when $\hat{\mathtt{P}} = \text{stack}(\hat{\mathtt{P}}_1, \hat{\mathtt{P}}_2, \ldots, \hat{\mathtt{P}}_m)$ has full column rank r. We complete the proof by showing that $\hat{\mathtt{P}}$ always has full column rank.

Assume, $\hat{\mathtt{P}}$ is rank deficient. Consider the matrix $\hat{\mathtt{X}} = [\hat{\mathbf{X}}_1, , \ldots, \hat{\mathbf{X}}_m]$. The matrix $\hat{\mathtt{P}}\hat{\mathtt{X}}$ can always be re-factorized as $\hat{\mathtt{P}}\hat{\mathtt{X}} = \hat{\mathtt{P}}'\hat{\mathtt{X}}'$, with $\hat{\mathtt{P}}'$ and $\hat{\mathtt{X}}'$ respectively of the same dimensions as $\hat{\mathtt{P}}$ and $\hat{\mathtt{X}}$, such that $\hat{\mathtt{P}}'$ *has full column rank.* By defining the same block structure as $\hat{\mathtt{P}}$ and $\hat{\mathtt{X}}$ for $\hat{\mathtt{P}}'$ and $\hat{\mathtt{X}}'$, that is $\hat{\mathtt{P}} = \text{stack}(\hat{\mathtt{P}}'_1, \ldots, \hat{\mathtt{P}}'_m)$ and $\hat{\mathtt{X}}' = [\hat{\mathbf{X}}'_1, \ldots, \hat{\mathbf{X}}'_m]$, we observe that $\hat{\mathtt{P}}'_i\hat{\mathbf{X}}'_j = \hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j$. As $\hat{\mathtt{P}}'$ has full column rank, from the discussion of the first half of the proof, we can say that $(\{\hat{\mathtt{P}}'_i\}, \{\hat{\mathbf{X}}'_j\})$ is projectively equivalent to $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$. This implies that $\hat{\mathtt{X}}' = [\hat{\mathbf{X}}'_1, \ldots, \hat{\mathbf{X}}'_m]$ has full row rank. As $\hat{\mathtt{P}}'$ and $\hat{\mathtt{X}}'$ both have maximum rank r, their product $\hat{\mathtt{P}}'\hat{\mathtt{X}}' = \hat{\mathtt{P}}\hat{\mathtt{X}}$ has rank r, requiring $\hat{\mathtt{P}}$ to have full column rank, a contradiction. □

4.4 Wrong solutions to projective factorization

Let us write equations $\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathtt{P}}_i\hat{\mathbf{X}}_j$ in matrix form

$$\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathtt{P}}\,\hat{\mathtt{X}}, \tag{4.33}$$

where $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = [\hat{\lambda}_{ij}\mathbf{x}_{ij}]$, $\hat{\mathtt{P}} = \text{stack}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m)$ and $\hat{\mathtt{X}} = [\hat{\mathbf{X}}_1, \ldots, \hat{\mathbf{X}}_n]$. The factorization-based algorithms seek to find $\hat{\Lambda}$ such that $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ can be factorized as the product of a $(\sum_i s_i) \times r$ matrix $\hat{\mathtt{P}}$ by an $r \times n$ matrix $\hat{\mathtt{X}}$. If $\mathbf{x}_{ij}$-s are obtained from a set of projection matrices $\mathtt{P}_i$ and points $\mathbf{X}_j$, according to $\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j/\lambda_{ij}$, our theory says that any solution $(\hat{\Lambda}, \hat{\mathtt{P}}, \hat{\mathtt{X}})$ to (4.33), is equivalent to the true solution $(\Lambda, \mathtt{P}, \mathtt{X})$, if $(\hat{\Lambda}, \hat{\mathtt{P}}, \hat{\mathtt{X}})$ satisfies some special restrictions, such as conditions (C1-C3) on $\hat{\mathtt{P}}$ and $\hat{\mathtt{X}}$ in Theorem 4.2, or $\hat{\Lambda}$ having no zero element in Theorem 4.4. It is worth to see what degenerate (projectively nonequivalent) forms a solution $(\hat{\Lambda}, \hat{\mathtt{P}}, \hat{\mathtt{X}})$ to (4.33) can take when such restrictions are not completely imposed.

In Chapter 3 we observed that for the special case of 3D to 2D projections, in any wrong solution to (4.33), the depth matrix $\hat{\Lambda}$ has some (entirely) zero rows, some zero columns, or it has a cross-like shape where the matrix is zero everywhere except at a certain row and a certain column. It is nice to see how the form of these *zero patterns* generalizes for arbitrary dimensional projections. This is important in the factorization-based methods, in which sometimes such restrictions as all nonzero depths cannot be efficiently implemented. Knowing the form of the wrong solutions, any reconstruction algorithm needs only to prevent certain zero patterns in the depth matrix, rather than constraining all elements of a depth matrix away from zero.

The reader can check that Theorem 4.4 can be proved under weaker assumptions than $\hat{\lambda}_{ij} \neq 0$ for all i and j, as follows

(D1) The matrix $\hat{\Lambda} = [\hat{\lambda}_{ij}]$ has no zero rows,

(D2) The matrix $\hat{\Lambda} = [\hat{\lambda}_{ij}]$ has no zero columns,

(D3) For every partition $\{I, J, K\}$ of views $\{1, 2, \ldots, m\}$ with $I \neq \emptyset$ and $\sum_{i \in I} s_i + \sum_{j \in J}(s_j - 1) < r$, the matrix $\hat{\Lambda}^K$ has *sufficiently many* nonzero columns, where $\hat{\Lambda}^K$ is the submatrix of $\hat{\Lambda}$ created by selecting rows according to K.

Notice that (D1) and (D2), respectively guarantee (C1) and (C2) in Theorem 4.2. This is due to the relation $\hat{\mathtt{P}}_i \hat{\mathbf{X}}_j = \hat{\lambda}_{ij} \mathtt{P}_i \mathbf{X}_j \neq 0$ for a nonzero $\hat{\lambda}_{ij}$ and by assuming $\mathtt{P}_i \mathbf{X}_j \neq 0$ due to genericity. Condition (D3) implies (C3) in Theorem 4.2, as we will shortly discuss.

By looking at the partition $\{I, J, K\}$ in Lemmas 4.7 and 4.8, we can say that the condition (D3) guarantees that the vector $\hat{\boldsymbol{\lambda}}_j^K$ used in (4.31) in the proof of Lemma 4.8 is nonzero for sufficiently many j-s. This is sufficient for the proof of Lemma 4.8 (compared to requiring $\hat{\Lambda}$ to have all-nonzero elements). Observing that $\hat{\boldsymbol{\lambda}}_j^K$ is the same thing as the j-th column of $\hat{\Lambda}^K$ defined in (D3), it is clear that (D3) is used to guarantee (C3) in Theorem 4.2, that is $\hat{\mathtt{P}}$ has a nonzero minor chosen according to some valid profile[9]. We suggest reading the proof of Lemma 4.8 for further understanding the discussion.

It is trivial to see how violating (D1) and (D2) can lead to a false solution to (4.33). For example set $\hat{\mathtt{X}} = \mathtt{X}$, $\hat{\mathtt{P}}_k$ and the k-th row of $\hat{\Lambda}$ equal to zero, and the rest of $\hat{\mathtt{P}}$ and $\hat{\Lambda}$ equal to $\mathtt{P}$ and Λ. In what comes next, we assume that (D1) and (D2) hold, that is $\hat{\Lambda}$ has no zero rows or zero columns, and look for less trivial false solutions to (4.33). According to our discussion above, for this class of wrong solutions conditions (C3) about $\hat{\mathtt{P}}$ and (D3) about $\hat{\Lambda}$ must be violated. This means that the set of views $\{1, 2, \ldots, m\}$ can be partitioned into I, J, K with I nonempty and $\sum_{i \in I} s_i + \sum_{i \in J}(s_i - 1) < r$, such that the submatrix $\hat{\Lambda}^K$ of $\hat{\Lambda}$ has *few*[10] nonzero columns. Moreover, by Lemma 4.7, the submatrix $\hat{\mathtt{P}}^K$ of $\hat{\mathtt{P}}$ has rank $r' = r - \sum_{i \in I} s_i - \sum_{i \in J}(s_i - 1)$. Here, we show how this can happen by first providing a simple example in which $J = \emptyset$ in Sect. 4.4.1. Next, in Sect. 4.4.2, we will demonstrate the wrong solutions in their general form, and show that degenerate solutions exist for every possible partition $\{I, J, K\}$.

[9]The reader might have noticed by comparing (D3) to Lemma 4.7 that here we have not considered the case of $\sum_{i \in I} s_i + \sum_{j \in J}(s_j - 1) = r$. If this case happens, we have $r' = r - \sum_{i \in I} s_i + \sum_{j \in J}(s_j - 1) = 0$. Therefore, $\hat{\mathtt{P}}^K$ has rank $r' = 0$, meaning that the rest of the projection matrices (whose indices are contained in K) have to be zero. However, as we discussed, zero projection matrices are precluded by (D1). Notice that, in this case, K cannot be empty. This is because we assumed $\sum_{i=1}^{n}(s_i - 1) \geq r$ about the size and the number of the projection matrices. But, if K is empty we have $\sum_{i=1}^{n}(s_i - 1) = \sum_{i \in I}(s_i - 1) + \sum_{j \in J}(s_j - 1) < \sum_{i \in I} s_i + \sum_{j \in J}(s_j - 1) = r$, where the inequality is due to the fact that I is nonempty.

[10]We will shortly discuss about the formal meaning of the term *few* here, and the term *sufficiently many* in (D3).

4.4.1 A simple example of wrong solutions

For a setup $(\{\mathtt{P}_i\},\{\mathbf{X}_j\})$, partition the views into two subsets I and K, such that $\sum_{i\in I} s_i < r$. Split $\mathtt{P}$ into two submatrices $\mathtt{P}^I = \text{stack}(\{\mathtt{P}_i\}_{i\in I})$ and $\mathtt{P}^K = \text{stack}(\{\mathtt{P}_i\}_{i\in K})$, and by possibly relabeling the views, assume that

$$\mathtt{P} = \text{stack}(\mathtt{P}^I, \mathtt{P}^K).$$

Notice that $\mathtt{P}^I$ has $\sum_{i\in I} s_i$ rows and r columns, and therefore, at least an $r' = r - \sum_{i\in I} s_i$ dimensional null space. Consider an $r\times r'$ matrix $\mathtt{N}$ with orthonormal columns all in the null space of $\mathtt{P}^I$. Also, let $\mathtt{R}$ be the orthogonal projection matrix into the row space of $\mathtt{P}^I$. Divide the matrix $\mathtt{X} = [\mathbf{X}_1,\ldots,\mathbf{X}_m]$ into two parts as $\mathtt{X} = [\mathtt{X}_1,\mathtt{X}_2]$ where $\mathtt{X}_1 = [\mathbf{X}_1,\ldots,\mathbf{X}_{r'}]$ and $\mathtt{X}_2 = [\mathbf{X}_{r'+1},\ldots,\mathbf{X}_m]$. Notice that $\mathtt{X}_1$ has r' columns. Define the corresponding submatrices $\hat{\mathtt{P}}^I$ and $\hat{\mathtt{P}}^K$ of $\hat{\mathtt{P}}$, and also, the corresponding submatrices $\hat{\mathtt{X}}_1$ and $\hat{\mathtt{X}}_2$ of $\hat{\mathtt{X}}$ as

$$\hat{\mathtt{P}}^I = \mathtt{P}^I, \qquad \hat{\mathtt{P}}^K = \mathtt{P}^K\mathtt{X}_1\mathtt{N}^T, \tag{4.34}$$

$$\hat{\mathtt{X}}_1 = \mathtt{R}\mathtt{X}_1 + \mathtt{N}, \qquad \hat{\mathtt{X}}_2 = \mathtt{R}\mathtt{X}_2. \tag{4.35}$$

One can easily check that

$$\hat{\mathtt{P}}\hat{\mathtt{X}} = \begin{bmatrix}\hat{\mathtt{P}}^I\\ \hat{\mathtt{P}}^K\end{bmatrix}[\hat{\mathtt{X}}_1,\hat{\mathtt{X}}_2] = \begin{bmatrix}\mathtt{P}^I\mathtt{X}_1 & \mathtt{P}^I\mathtt{X}_2\\ \mathtt{P}^K\mathtt{X}_1 & 0\end{bmatrix} = \hat{\Lambda}\odot(\mathtt{P}\mathtt{X}), \tag{4.36}$$

where $\hat{\Lambda}$ has a block structure of the form

$$\hat{\Lambda} = \begin{bmatrix}\hat{\Lambda}^I\\ \hat{\Lambda}^K\end{bmatrix} = \begin{bmatrix}1 & 1\\ 1 & 0\end{bmatrix}. \tag{4.37}$$

As $\hat{\mathtt{P}}^I \in \mathbb{R}^{(r-r')\times r}$ has at most rank $r-r'$ and $\hat{\mathtt{P}}^K = \mathtt{P}^K\mathtt{X}_1\mathtt{N}^T$ (with $\mathtt{N}\in\mathbb{R}^{r\times r'}$) has at most rank r', if $\hat{\mathtt{P}} = \text{stack}(\hat{\mathtt{P}}^I,\hat{\mathtt{P}}^K)$ has maximal rank r then $\hat{\mathtt{P}}^K$ has to have rank r', as also confirmed by Lemma 4.7. Since $\hat{\mathtt{P}}^K$ has at most rank r' and $\hat{\mathtt{P}}^I$ has $r-r'$ rows, any non-singular $r\times r$ submatrix of $\hat{\mathtt{P}} = \text{stack}(\hat{\mathtt{P}}^I,\hat{\mathtt{P}}^K)$ must contain all rows of $\hat{\mathtt{P}}^I$. Therefore, $\hat{\mathtt{P}} = \text{stack}(\hat{\mathtt{P}}^I,\hat{\mathtt{P}}^K)$ has no full rank $r\times r$ submatrix chosen by less than s_i rows from each $\hat{\mathtt{P}}_i$. Thus, (C3) is violated and the Grassmann tensor of $\{\hat{\mathtt{P}}_i\}$ with any valid profile is zero. Also, observe that in (4.37) the submatrix $\hat{\Lambda}^K$ of $\hat{\Lambda}\in\mathbb{R}^{m\times n}$ has only r' nonzero columns, no matter how large n is. This is how (D3) is violated. Notice that $\hat{\Lambda}$ need not have the exact block structure as above. By permuting the views and HD points, a wrong solution can be obtained in which rows and columns of $\hat{\Lambda}$ in (4.37) are permuted.

Using the above style for finding wrong solutions the matrix $\hat{\Lambda}^K$ can have at most r' nonzero columns. This happens to cover all sorts of wrong solutions in some special cases including the common case of projections $\mathbb{P}^3 \to \mathbb{P}^2$. But, unfortunately, this is not always the case. In other words, *sufficiently many* in the condition (D3) to rule out false solutions does not always mean more than $r' = r - \sum_{i\in I} s_i + \sum_{i\in J}(s_i - 1)$. In some cases, there might exist more general types of degenerate solutions with

more than r' nonzero columns in $\hat{\Lambda}^K$, even when J is empty. We consider this issue in more detail in the next subsection, where the wrong solutions are demonstrated in their general form.

4.4.2 Wrong solutions: The general case

In this subsection, we show that a degenerate solution can be constructed for every valid partition $\{I,J,K\}$. Consider any partition $\{I,J,K\}$ of views $\{1,2,\ldots,m\}$ with $I \neq \emptyset$ and $\sum_{i\in I} s_i + \sum_{j\in J}(s_j-1) < r$. Define

$$\hat{\mathtt{P}}^{IJ} = \hat{\mathtt{P}}^{I\cup J} = \mathrm{stack}(\{\hat{\mathtt{P}}_i\}_{i\in I\cup J}), \tag{4.38}$$

and as before, let $\hat{\mathtt{P}}^K = \mathrm{stack}(\{\hat{\mathtt{P}}_i\}_{i\in K})$. With possibly rearranging the views, we can assume that

$$\hat{\mathtt{P}} = \begin{bmatrix} \hat{\mathtt{P}}^{IJ} \\ \hat{\mathtt{P}}^K \end{bmatrix}. \tag{4.39}$$

Similarly, for the true projections $\mathtt{P} = \mathrm{stack}(\mathtt{P}_1,\ldots,\mathtt{P}_m)$ we can define $\mathtt{P}^{IJ}$ and $\mathtt{P}^K$ in the same way, and assume that $\mathtt{P} = \mathrm{stack}(\mathtt{P}^{IJ},\mathtt{P}^K)$. We construct an example in which $\hat{\mathtt{P}}^K$ has at most rank $r' = r - \sum_{i\in I} s_i - \sum_{j\in J}(s_j-1)$, and $\hat{\mathtt{P}}^{IJ}$ has at most rank

$$r'' = r - r' = \sum_{i\in I} s_i + \sum_{j\in J}(s_j-1).$$

Notice that $r' + r'' = r$.

4.4.2.1 Dealing with the views in I and J

Now, one challenge is how to construct $\hat{\mathtt{P}}^{IJ}$ with rank r'' or less, such that $\hat{\mathtt{P}}^{IJ}\,\hat{\mathtt{X}}$ projects into the same image points as $\mathtt{P}^{IJ}\,\mathtt{X}$, that is $\hat{\mathtt{P}}^{IJ}\,\hat{\mathtt{X}} = \hat{\Lambda}^{IJ} \odot (\mathtt{P}^{IJ}\,\mathtt{X}) = [\hat{\lambda}_{ij}\,\mathtt{P}_i\,\mathbf{X}_j]_{i\in I\cup J}$ for some depth matrix $\hat{\Lambda}^{IJ} = [\hat{\lambda}_{ij}]_{i\in I\cup J}$. When J was empty, this was easy as then $\hat{\mathtt{P}}^{IJ}$ would have exactly r'' rows, and could not have a rank of more than r''. But, in general $\hat{\mathtt{P}}^{IJ}$ has

$$\sum_{i\in I\cup J} s_i = \sum_{i\in I} s_i - \sum_{j\in J}(s_j-1) + |J| = r'' + |J|$$

rows, which is more than r'' when J is nonempty. But if we consider the matrix

$$\hat{\mathtt{Q}}^{IJ} = \mathrm{stack}(\{\hat{\mathtt{Q}}_i\}_{i\in I\cup J}), \tag{4.40}$$

where $\hat{\mathtt{Q}}_i = \hat{\mathtt{P}}_i$ for all $i \in I$ and $\hat{\mathtt{Q}}_i$ consists of the first $s_i - 1$ rows[11] of $\hat{\mathtt{P}}_i$ for all $i \in J$, then $\hat{\mathtt{Q}}^{IJ}$ has r'' rows. Note that we have

$$\hat{\mathtt{P}}_i = \begin{cases} \hat{\mathtt{Q}}_i & i \in I \\ \text{stack}(\hat{\mathtt{Q}}_i, \hat{\mathbf{r}}_i^T) & i \in J \end{cases} \tag{4.41}$$

where for every $i \in J$ the row vector $\hat{\mathbf{r}}_i^T$ is the final row of $\hat{\mathtt{P}}_i$. As $\hat{\mathtt{Q}}^{IJ}$ has r'' rows, it has rank r'' or less. To get a clue on how to construct $\hat{\mathtt{P}}^{IJ}$, first observe that for $\hat{\mathtt{P}}^{IJ}$ to have rank r'' or less, it is sufficient that for all $i \in J$ the rows $\hat{\mathbf{r}}_i^T$ are in the row space of $\hat{\mathtt{Q}}^{IJ}$. In other words,

$$\hat{\mathbf{r}}_i^T = \mathbf{b}_i^T \hat{\mathtt{Q}}^{IJ} \tag{4.42}$$

for some $\mathbf{b}_i \in \mathbb{R}^{r''}$. Now, with a possible permutation of the views, we can assume that

$$\begin{aligned} J &= 1, 2, \ldots, p, \\ I &= p+1, p+2, \ldots, q, \end{aligned}$$

where $|J| = p < q = |I| + |J|$. Now, we can write (4.42) for all i-s as

$$\begin{bmatrix} \hat{\mathbf{r}}_1^T \\ \vdots \\ \hat{\mathbf{r}}_p^T \end{bmatrix} = \mathtt{B}\,\hat{\mathtt{Q}}^{IJ} = \mathtt{B} \begin{bmatrix} \hat{\mathtt{Q}}_1^T \\ \vdots \\ \hat{\mathtt{Q}}_q^T \end{bmatrix},$$

where $\mathtt{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_p]^T \in \mathbb{R}^{p \times r''}$. Multiplying both sides by $\hat{\mathbf{X}}_j$ we get

$$\begin{bmatrix} \hat{\mathbf{r}}_1^T \hat{\mathbf{X}}_j \\ \vdots \\ \hat{\mathbf{r}}_p^T \hat{\mathbf{X}}_j \end{bmatrix} = \mathtt{B} \begin{bmatrix} \hat{\mathtt{Q}}_1^T \hat{\mathbf{X}}_j \\ \vdots \\ \hat{\mathtt{Q}}_q^T \hat{\mathbf{X}}_j \end{bmatrix}. \tag{4.43}$$

Using the projection equations $\hat{\mathtt{P}}_i \hat{\mathbf{X}}_j = \hat{\lambda}_{ij} \mathtt{P}_i \mathbf{X}_j$, the above gives

$$\begin{bmatrix} \hat{\lambda}_{1j}\, \mathbf{r}_1^T \mathbf{X}_j \\ \vdots \\ \hat{\lambda}_{pj}\, \mathbf{r}_p^T \mathbf{X}_j \end{bmatrix} = \mathtt{B} \begin{bmatrix} \hat{\lambda}_{1j}\, \mathtt{Q}_1^T \mathbf{X}_j \\ \vdots \\ \hat{\lambda}_{1j}\, \mathtt{Q}_q^T \mathbf{X}_j \end{bmatrix}. \tag{4.44}$$

[11] Actually, $\hat{\mathtt{Q}}_i$ here can consist of any $s_i - 1$ rows of $\hat{\mathtt{P}}_i$. The first $s_i - 1$ rows are considered for simplicity.

where $\mathbf{r}_i^T$ is the last row of $\mathtt{P}_i$ and $\mathtt{Q}_i$ is the submatrix of $\mathtt{P}_i$ corresponding to $\hat{\mathtt{Q}}_i$. The above can be reformulated as

$$\begin{bmatrix} \mathbf{r}_1^T\mathbf{X}_j & & \\ & \ddots & \\ & & \mathbf{r}_p^T\mathbf{X}_j \end{bmatrix} \begin{pmatrix} \hat{\lambda}_{1j} \\ \vdots \\ \hat{\lambda}_{pj} \end{pmatrix} = \mathtt{B} \begin{bmatrix} \mathtt{Q}_1^T\mathbf{X}_j & & \\ & \ddots & \\ & & \mathtt{Q}_q^T\mathbf{X}_j \end{bmatrix} \begin{pmatrix} \hat{\lambda}_{1j} \\ \vdots \\ \hat{\lambda}_{qj} \end{pmatrix}. \tag{4.45}$$

In other words,

$$\left(\begin{bmatrix} \operatorname{diag}(\mathbf{r}_1^T\mathbf{X}_j, \ldots, \mathbf{r}_p^T\mathbf{X}_j) & 0_{p\times(q-p)} \end{bmatrix} - \mathtt{B} \operatorname{diag}(\mathtt{Q}_1^T\mathbf{X}_j, \ldots, \mathtt{Q}_q^T\mathbf{X}_j) \right) \hat{\boldsymbol{\lambda}}_j^{IJ} = 0. \tag{4.46}$$

where $\operatorname{diag}(\cdot)$ makes block-diagonal matrices, and

$$\hat{\boldsymbol{\lambda}}_j^{IJ} = \begin{pmatrix} \hat{\lambda}_{1j} \\ \vdots \\ \hat{\lambda}_{qj} \end{pmatrix} \tag{4.47}$$

Notice that the matrix on the left hand side of (4.46) has p rows and q columns. As p is strictly larger than q (since I is nonempty), the equation (4.46) is satisfied by setting $\hat{\boldsymbol{\lambda}}_j^{IJ}$ to a nonzero vector in the null space of this $p\times q$ matrix. Therefore, we have now found a $\hat{\boldsymbol{\lambda}}_j^{IJ}$ such that

$$\begin{bmatrix} \operatorname{diag}(\mathbf{r}_1^T\mathbf{X}_j, \ldots, \mathbf{r}_p^T\mathbf{X}_j) & 0_{p\times(q-p)} \end{bmatrix} \hat{\boldsymbol{\lambda}}_j^{IJ} = \mathtt{B}\ \operatorname{diag}(\mathtt{Q}_1^T\mathbf{X}_j, \ldots, \mathtt{Q}_q^T\mathbf{X}_j)\ \hat{\boldsymbol{\lambda}}_j^{IJ}, \tag{4.48}$$

or

$$\operatorname{stack}(\hat{\lambda}_{1j}\ \mathbf{r}_1^T\mathbf{X}_j, \ldots, \hat{\lambda}_{pj}\ \mathbf{r}_p^T\mathbf{X}_j) = \mathtt{B}\ \operatorname{stack}(\hat{\lambda}_{1j}\ \mathtt{Q}_1^T\mathbf{X}_j, \ldots, \hat{\lambda}_{qj}\ \mathtt{Q}_q^T\mathbf{X}_j). \tag{4.49}$$

It follows that

$$\begin{bmatrix} \operatorname{stack}(\hat{\lambda}_{1j}\ \mathtt{Q}_1^T\mathbf{X}_j, \ldots, \hat{\lambda}_{qj}\ \mathtt{Q}_q^T\mathbf{X}_j) \\ \operatorname{stack}(\hat{\lambda}_{1j}\ \mathbf{r}_1^T\mathbf{X}_j, \ldots, \hat{\lambda}_{pj}\ \mathbf{r}_p^T\mathbf{X}_j) \end{bmatrix} = \begin{bmatrix} \mathtt{I} \\ \mathtt{B} \end{bmatrix} \operatorname{stack}(\hat{\lambda}_{1j}\ \mathtt{Q}_1^T\mathbf{X}_j, \ldots, \hat{\lambda}_{1j}\ \mathtt{Q}_q^T\mathbf{X}_j). \tag{4.50}$$

Notice that the matrix on the left hand side is equal to $\operatorname{stack}(\hat{\lambda}_{1j}\mathtt{P}_1^T\mathbf{X}_j, \ldots, \hat{\lambda}_{qj}\mathtt{P}_q^T\mathbf{X}_j)$ up to permutation of rows. Thus, we have

$$\begin{bmatrix} \hat{\lambda}_{1j}\ \mathtt{P}_1^T\mathbf{X}_j \\ \vdots \\ \hat{\lambda}_{qj}\ \mathtt{P}_q^T\mathbf{X}_j \end{bmatrix} = \mathtt{S} \begin{bmatrix} \hat{\lambda}_{1j}\ \mathtt{Q}_1^T\mathbf{X}_j \\ \vdots \\ \hat{\lambda}_{qj}\ \mathtt{Q}_q^T\mathbf{X}_j \end{bmatrix} = \mathtt{S}\,\mathbf{t}_j. \tag{4.51}$$

where $\mathtt{S}$ is obtained by properly permuting the rows of $\mathrm{stack}(\mathtt{I},\mathtt{B})$, and $\mathbf{t}_j = \mathrm{stack}(\hat{\lambda}_{1j}\,\mathtt{Q}_1^T\mathbf{X}_j,\ldots,\hat{\lambda}_{qj}\,\mathtt{Q}_q^T\mathbf{X}_j) \in \mathbb{R}^{r''}$. Thus, we have

$$\hat{\Lambda}^{IJ} \odot (\mathtt{P}^{IJ}\mathtt{X}) = \begin{bmatrix} \hat{\lambda}_{11}\,\mathtt{P}_1^T\mathbf{X}_1 & \cdots & \hat{\lambda}_{1n}\,\mathtt{P}_1^T\mathbf{X}_n \\ \vdots & \ddots & \vdots \\ \hat{\lambda}_{qj}\,\mathtt{P}_q^T\mathbf{X}_j & \cdots & \hat{\lambda}_{qj}\,\mathtt{P}_q^T\mathbf{X}_j \end{bmatrix} = \mathtt{S}\,\left[\,\mathbf{t}_1,\mathbf{t}_2,\ldots,\mathbf{t}_n\,\right]. \tag{4.52}$$

This means that we have found $\hat{\Lambda}^{IJ} = [\hat{\boldsymbol{\lambda}}_1^{IJ},\hat{\boldsymbol{\lambda}}_2^{IJ},\ldots,\hat{\boldsymbol{\lambda}}_n^{IJ}]$ such that $\hat{\Lambda}^{IJ} \odot (\mathtt{P}^{IJ}\mathtt{X})$ has rank r'' or less, and can be factorized as

$$\hat{\Lambda}^{IJ} \odot (\mathtt{P}^{IJ}\mathtt{X}) = \mathtt{S}\,\mathtt{T}^T, \tag{4.53}$$

where $\mathtt{S}$ and $\mathtt{T} = [\mathbf{t}_1,\ldots,\mathbf{t}_n]^T$ both have r'' columns[12]. We leave the above here and turn our attention to the subset of views K.

4.4.2.2 Dealing with the views in K

In the previous subsection we presented a simple example of a degenerate solution in which $\hat{\Lambda}^K$ had r' nonzero columns. Here, we show that the limit on the number of nonzero columns of $\hat{\Lambda}^K$ for having a degenerate solution can be generally more than r'. Recall from Lemma 4.7 that for every degenerate solution[13], if $\hat{\mathtt{P}}$ *is chosen* to have full column rank[14], there exists a corresponding partition $\{I,J,K\}$ for which $\hat{\mathtt{P}}^K$ has rank $r' = r - \sum_{i\in I} s_i - \sum_{i\in J}(s_i-1)$, and further, its row space is spanned by the rows of an $r'\times r$ submatrix $\hat{\mathtt{Q}}^K = \mathrm{stack}(\{\hat{\mathtt{Q}}_i\}_{i\in K})$ of $\hat{\mathtt{P}}^K$, where each $\hat{\mathtt{Q}}_i$ consists of strictly less than s_i rows from $\hat{\mathtt{P}}_i$. In Lemma 4.8 we used this to show that if $\hat{\Lambda}^K$ has sufficiently many nonzero columns $\hat{\boldsymbol{\lambda}}_j^K$ then a wrong solution according to $\{I,J,K\}$ cannot occur. In other words, for having a wrong solution according to a certain partition $\{I,J,K\}$, there is a limit on the number of nonzero columns of $\hat{\Lambda}^K$. If this limit is violated, either $\hat{\mathtt{P}}^K$ has to have rank more than r', or it has no full-row-rank $r'\times r$ submatrix chosen by strictly fewer than s_i rows from each $\hat{\mathtt{P}}_i$ with $i \in K$. In such cases, either the current solution is not degenerate, or it is degenerate, but associated with a partition different from $\{I,J,K\}$. But what is the limit on the number of nonzero columns of $\hat{\Lambda}^K$ allowing for a wrong solution? To answer this, we need to look back at the proof of Lemma 4.8.

In the proof of Lemma 4.8 for each $i \in K$ the matrix $\hat{\mathtt{P}}_i$ was divided, row-wise,

[12]We do not elaborate on such technicalities as how to make sure that $\hat{\Lambda}^{IJ}$ do not have an entirely zero row. Just notice that in the above approach, in (4.46), some control over each column $\hat{\boldsymbol{\lambda}}_j^{IJ}$ of $\hat{\Lambda}^{IJ}$ can be obtained by playing with the matrix $\mathtt{B}$.

[13]As mentioned at the beginning of this section, we only deal with nontrivial degenerate solutions in which $\hat{\Lambda}$ has no zero rows and no zero columns. In other words, we are talking about the degenerate solutions arising from the violation of (D3), or equivalently (C3), while assuming that (D1) and (D2) are satisfied.

[14]Notice that by possibly re-factorizing $\hat{\mathtt{P}}\hat{\mathtt{X}}$ we can always choose $\hat{\mathtt{P}}$ to have full column rank. More precisely, if $\hat{\Lambda} \odot (\mathtt{PX})$ has rank r or less, then there exist matrices $\hat{\mathtt{P}} \in \mathbb{R}^{(\sum_i s_i)\times r}$ and $\hat{\mathtt{X}} \in \mathbb{R}^{r\times n}$ such that $\hat{\mathtt{P}}\hat{\mathtt{X}} = \hat{\Lambda} \odot (\mathtt{PX})$ and $\hat{\mathtt{P}}$ has maximal rank r (see also the proof of Theorem 4.4).

into two submatrices $\hat{\mathtt{Q}}_i$ and $\hat{\mathtt{R}}_i$. Then, it was required that the row space of $\hat{\mathtt{R}}^K = \text{stack}(\{\hat{\mathtt{R}}_i\}_{i\in K})$ is spanned by the rows of the $r' \times r$ matrix $\hat{\mathtt{Q}}^K = \text{stack}(\{\hat{\mathtt{Q}}_i\}_{i\in K})$, that is $\hat{\mathtt{R}}^K = \mathtt{A}\,\hat{\mathtt{Q}}^K$. From there, we obtained

$$\mathcal{M}(\mathbf{X}_j)\,\hat{\boldsymbol{\lambda}}_j^K = \mathbf{0}, \tag{4.54}$$

where $\hat{\boldsymbol{\lambda}}_j^K$ is the j-th column of $\hat{\Lambda}^K$ and

$$\mathcal{M}(\mathbf{X}) = \text{diag}(\{\mathtt{R}_i\mathbf{X}\}_{i\in K}) - \mathtt{A}\,\text{diag}(\{\mathtt{Q}_i\mathbf{X}\}_{i\in K}) \tag{4.55}$$

is a tall or square matrix. If $\hat{\boldsymbol{\lambda}}_j^K$ is nonzero, from (4.54) it follows that $\mathcal{M}(\mathbf{X}_j)$ is rank deficient, and thus, all its $|K|\times|K|$ submatrices have a zero determinant. Let $\mathcal{M}^k(\mathbf{X})$ be the k-th $|K|\times|K|$ submatrix of $\mathcal{M}(\mathbf{X})$. Notice that $\det(\mathcal{M}^k(\mathbf{X}))$ is a polynomial in $\mathbf{X}$. Define the polynomial surface S^k as the kernel of $\det(\mathcal{M}^k(\mathbf{X}))$, that is

$$S^k = \{\mathbf{X} \mid \mathcal{M}^k(\mathbf{X}) = 0\}. \tag{4.56}$$

For every nonzero $\hat{\boldsymbol{\lambda}}_j^K$ equation (4.54) implies that $\det(\mathcal{M}^k(\mathbf{X}_j)) = 0$ for all k, or equivalently

$$\mathbf{X}_j \in S = \cup_k S^k. \tag{4.57}$$

Notice that, for generic projection matrices $\mathtt{P}_i$, no matter how the submatrices $\mathtt{Q}_i$ and the matrix $\mathtt{A}$ are chosen, at least for some choices of k, the polynomial $\det(\mathcal{M}^k(\cdot))$ is not identically zero. Therefore, S is a non-generic (nowhere dense) set. To see this, assume that the l-th submatrix $\det(\mathcal{M}^l(\mathbf{X}_j))$ is created by choosing one row $\mathbf{r}_i^T$ from each $\mathtt{R}_i$, that is

$$\mathcal{M}^l(\mathbf{X}) = \text{diag}(\{\mathbf{r}_i^T\mathbf{X}\}_{i\in K}) - \mathtt{A}^l\,\text{diag}(\{\mathtt{Q}_i\mathbf{X}\}_{i\in K}), \tag{4.58}$$

where $\mathbf{r}_i^T$ is an arbitrary row of $\mathtt{R}_i$ and $\mathtt{A}^l$ is the corresponding $|K|\times r'$ submatrix of $\mathtt{A}$. Notice that this can be done since each $\mathtt{R}_i$ have at least one row. Now, because $\mathtt{P}_i$-s are generic, we can assume that a nonzero $\mathbf{Y}$ in the null space of $\hat{\mathtt{Q}}^K = \text{stack}(\{\hat{\mathtt{Q}}_i\}_{i\in K})$ can be chosen such that $\mathbf{r}_i^T\mathbf{Y} \neq 0$ for all $i \in K$ (see also the proof of Lemma 4.8). In this case we have $\det(\mathcal{M}^l(\mathbf{Y})) \neq 0$, and thus $\det(\mathcal{M}^l(\cdot))$ cannot be identically zero. Therefore, S is a non-generic (nowhere dense) set as the intersection of polynomial surfaces S^k. As a result, we cannot have arbitrarily many points $\mathbf{X}_j$ in general position all lying on S. This restricts the number of nonzero $\hat{\boldsymbol{\lambda}}_j^K$-s allowed in a degenerate solution.

Notice that for a given configuration $\{\mathtt{P}_i\}$, the set S is fully determined by the choice of the submatrices $\mathtt{Q}_i$ and the matrix $\mathtt{A}$. Therefore, the term *sufficiently many* in condition (D3) can be translated as strictly more than the maximum number of points in general position which can lie on S for any choice of $\mathtt{Q}_i$-s and $\mathtt{A}$ (this means that the maximum is also taken over all possible choices of $\mathtt{Q}_i$-s and $\mathtt{A}$).

Now, lets see what happens if this new interpretation of (D3) is violated. In this case the number of nonzero $\hat{\lambda}_j^K$-s is sufficiently small such that for at least one choice of $\mathtt{Q}_i$-s and $\mathtt{A}$ all the points $\mathbf{X}_j$ with a nonzero corresponding $\hat{\lambda}_j^K$ lie on S. In this case, for every j with nonzero $\hat{\lambda}_j^K$, $\det(\mathcal{M}^k(\mathbf{X}_j))$ is zero for all k, and thus all submatrices of $\mathcal{M}(\mathbf{X}_j)$ are singular. Hence, $\mathcal{M}(\mathbf{X}_j)$ is rank-deficient and we can choose a nonzero $\hat{\lambda}_j^K$ such that $\mathcal{M}(\mathbf{X}_j)\hat{\lambda}_j^K = 0$. This gives

$$\operatorname{diag}(\{\mathtt{R}_i\mathbf{X}_j\}_{i\in K})\hat{\lambda}_j^K = \mathtt{A}\operatorname{diag}(\{\mathtt{Q}_i\mathbf{X}_j\}_{i\in K})\hat{\lambda}_j^K \tag{4.59}$$

or

$$\operatorname{stack}(\{\hat{\lambda}_{ij}\mathtt{R}_i\mathbf{X}_j\}_{i\in K}) = \mathtt{A}\operatorname{stack}(\{\hat{\lambda}_{ij}\mathtt{Q}_i\mathbf{X}_j\}_{i\in K}), \tag{4.60}$$

which gives

$$\begin{bmatrix} \operatorname{stack}(\{\hat{\lambda}_{ij}\mathtt{Q}_i\mathbf{X}_j\}_{i\in K}) \\ \operatorname{stack}(\{\hat{\lambda}_{ij}\mathtt{R}_i\mathbf{X}_j\}_{i\in K}) \end{bmatrix} = \begin{bmatrix} \mathtt{I} \\ \mathtt{A} \end{bmatrix} \operatorname{stack}(\{\hat{\lambda}_{ij}\mathtt{Q}_i\mathbf{X}_j\}_{i\in K}). \tag{4.61}$$

By permuting the rows in the above we get

$$\operatorname{stack}(\{\hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j\}_{i\in K}) = \mathtt{U}\operatorname{stack}(\{\hat{\lambda}_{ij}\mathtt{Q}_i\mathbf{X}_j\}_{i\in K}) = \mathtt{U}\,\mathbf{v}_j, \tag{4.62}$$

where $\mathtt{U}$ is obtained by an appropriate permutation of the rows of $\operatorname{stack}(\mathtt{I},\mathtt{A})$, and $\mathbf{v}_j = \operatorname{stack}(\{\hat{\lambda}_{ij}\mathtt{Q}_i\mathbf{X}_j\}_{i\in K})$. Notice that $\mathtt{U}$ has r' columns.

By rearranging the columns of $\hat{\Lambda}^K$ (and accordingly $\mathbf{X}_j$-s) if necessary, we assume that the nonzero columns of $\hat{\Lambda}^K$ are the leftmost columns, that is $\hat{\Lambda}^K = [\,\hat{\Lambda}_1^K \;\; 0\,]$ where $\hat{\Lambda}_1^K$ contains the nonzero columns. Accordingly, we divide $\mathtt{X} = [\mathbf{X}_1,\ldots,\mathbf{X}_n]$ into two parts as $\mathtt{X} = [\mathtt{X}_1,\mathtt{X}_2]$ such that $\mathtt{X}_1$ has the same number of columns as $\hat{\Lambda}_1^K$. Therefore, by horizontally concatenating both sides of (4.62) for all j we get

$$\hat{\Lambda}_1^K \odot (\mathtt{P}^K\mathtt{X}_1) = \mathtt{U}\,\mathtt{V}^T \tag{4.63}$$

where $\mathtt{V} = [\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_{n'}]^T$ with n' being the number of columns of $\hat{\Lambda}_1^K$. We shall shortly show that a degenerate solution can be constructed using (4.63) and (4.53). But before that, lets discuss a few points.

Another indication of a wrong solution is obtained by looking at the rank of $\hat{\Lambda}^K \odot (\mathtt{P}^K\mathtt{X})$. Notice that (4.63) implies that for having a degenerate solution the rank of $\hat{\Lambda}_1^K \odot (\mathtt{P}^K\mathtt{X}_1)$ can be at most r'. Since $\hat{\Lambda}^K = [\,\hat{\Lambda}_1^K \;\; 0\,]$, it means that also the rank of $\hat{\Lambda}^K \odot (\mathtt{P}^K\mathtt{X})$ can be at most r'. This is confirmed by the relation $\hat{\mathtt{P}}^K\hat{\mathtt{X}} = \hat{\Lambda}^K \odot (\mathtt{P}^K\mathtt{X})$ and the fact that for a degenerate solution $\hat{\mathtt{P}}^K$ has rank r' (if $\hat{\mathtt{P}}$ is chosen to have full column rank) according to Lemma 4.7. Therefore, $\hat{\Lambda}^K \odot (\mathtt{P}^K\mathtt{X})$ having a rank of at most r' is necessary for having a degenerate solution corresponding to $\{I,J,K\}$. One can show that, under generic conditions, this is also sufficient for having a wrong solution. Notice that given a generic configuration $(\mathtt{P},\mathtt{X})$ of projection matrices and

HD points, for any projectively equivalent solution $(\hat{\mathtt{P}}, \hat{\mathtt{X}})$ the matrix $\hat{\Lambda}^K \odot (\mathtt{P}^K\mathtt{X}) = \hat{\mathtt{P}}^K\hat{\mathtt{X}}$ has rank $\sum_{i\in K} s_i$ which is strictly bigger than $r' = r - \sum_{i\in I} s_i - \sum_{i\in J}(s_i-1)$ due to the condition $\sum_{i=1}^{m}(s_i-1) \geq r$. Therefore, if $\hat{\Lambda}^K \odot (\mathtt{P}^K\mathtt{X})$ has rank r' or less, one can make sure that a degenerate solution has occurred. However, one should bear in mind that $\hat{\Lambda}^K \odot (\mathtt{P}^K\mathtt{X})$ having rank r' or less for a partition $\{I, J, K\}$ is only sufficient for the existence a wrong solution. The wrong solution, however, may correspond to a partition different from $\{I, J, K\}$.

A relevant question is whether $\hat{\Lambda}^K \odot (\mathtt{P}^K\mathtt{X})$ having rank r' or less for some partition $\{I, J, K\}$ always puts an upper bound on the number of nonzero columns of $\hat{\Lambda}^K$. The answer is it does put an upper bound if $\{I, J, K\}$ is the true partition corresponding to the wrong solution. In other words, in addition to $\hat{\mathtt{P}}^K$ having rank r', there should be a full-row-rank $r' \times r$ submatrix $\hat{\mathtt{Q}}^K = \text{stack}(\{\hat{\mathtt{Q}}_i\}_{i\in K})$ of $\hat{\mathtt{P}}^K$, where each $\hat{\mathtt{Q}}_i$ is created by choosing strictly less than s_i rows from $\hat{\mathtt{P}}_i$ (and thus, the rows of $\hat{\mathtt{Q}}^K$ naturally span the row space of $\hat{\mathtt{P}}^K$). If this does not happen, $\hat{\Lambda}^K$ can have sufficiently many nonzero columns without the rank of $\hat{\Lambda}^K \odot (\mathtt{P}^K\mathtt{X})$ exceeding r'. In this case, $\hat{\Lambda}$ can still be a degenerate solution corresponding to a partition different than $\{I, J, K\}$.

The maximum number of nonzero $\hat{\lambda}_j^K$-s allowed for a degenerate solution cannot be smaller than $r' = r - \sum_{i\in I} s_i - \sum_{i\in J}(s_i-1)$. If $\hat{\Lambda}^K$ has r' nonzero columns, then $\hat{\Lambda}_1^K \odot (\mathtt{P}^K\mathtt{X}_1)$ has r' columns ($\hat{\Lambda}_1^K$ and $\mathtt{X}_1$ were defined above). Thus, it can naturally be factorized as $\mathtt{U}\mathtt{V}^T$ as required by (4.62) using which a degenerate solution is constructed in the next subsection. With a little effort one can show for any r' points $\mathbf{X}_j$ in general position, one could choose $\mathtt{Q}_i$-s and $\mathtt{A}$ such that all $\mathbf{X}_j$ lie on the corresponding set S. While in some special cases, including the case of 3D to 2D projections, the maximum number of nonzero columns in $\hat{\Lambda}^K$ for a degenerate solution is exactly equal to r', this number can be bigger than r' in the general case.

4.4.2.3 Constructing the degenerate solution

Now, all the required means are available to construct a degenerate solution. Looking at (4.53) and (4.63) it is clear that we have found a $\hat{\Lambda}$ with the block structure

$$\hat{\Lambda} = \begin{bmatrix} \hat{\Lambda}^{IJ} \\ \hat{\Lambda}^K \end{bmatrix} = \begin{bmatrix} \hat{\Lambda}_1^{IJ} & \hat{\Lambda}_2^{IJ} \\ \hat{\Lambda}_1^K & 0, \end{bmatrix} \tag{4.64}$$

for which $\hat{\Lambda} \odot (\mathtt{P}\mathtt{X})$ has the following form

$$\hat{\Lambda} \odot (\mathtt{P}\mathtt{X}) = \begin{bmatrix} \hat{\Lambda}^{IJ} \odot (\mathtt{P}^{IJ}\mathtt{X}) \\ \hat{\Lambda}^K \odot (\mathtt{P}^K\mathtt{X}) \end{bmatrix} = \begin{bmatrix} \mathtt{S}\,\mathtt{T}^T & \\ \mathtt{U}\mathtt{V}^T & 0 \end{bmatrix} = \begin{bmatrix} \mathtt{S}\mathtt{T}_1^T & \mathtt{S}\mathtt{T}_2^T \\ \mathtt{U}\mathtt{V}^T & 0 \end{bmatrix}, \tag{4.65}$$

where $\mathtt{S}$ and $\mathtt{T}$ both have r'' columns, $\mathtt{U}$ and $\mathtt{V}$ both have r' columns, and $\mathtt{T}_1$ and $\mathtt{T}_2$ are submatrices of $\mathtt{T}$ such that $\text{stack}(\mathtt{T}_1, \mathtt{T}_2) = \mathtt{T}$ and $\mathtt{T}_1$ has the same number of rows

as V. Now, it remains to find $\hat{\mathtt{P}} \in \mathbb{R}^{(\sum_i s_i)\times r}$ and $\hat{\mathtt{X}} \in \mathbb{R}^{r\times n}$ such that $\hat{\mathtt{P}}\hat{\mathtt{X}} = \hat{\Lambda} \odot (\mathtt{PX})$. Let

$$\hat{\mathtt{P}} = \begin{bmatrix} \hat{\mathtt{P}}^{IJ} \\ \hat{\mathtt{P}}^{K} \end{bmatrix}, \qquad \hat{\mathtt{X}} = \begin{bmatrix} \hat{\mathtt{X}}_1 & \hat{\mathtt{X}}_2 \end{bmatrix} \tag{4.66}$$

where

$$\hat{\mathtt{P}}^{IJ} = \begin{bmatrix} \mathtt{S} & \mathtt{0}_{(\cdot)\times r'} \end{bmatrix}, \qquad \hat{\mathtt{X}}_1 = \begin{bmatrix} \mathtt{T}_1^T \\ \mathtt{V}^T \end{bmatrix}, \tag{4.67}$$

$$\hat{\mathtt{P}}^{K} = \begin{bmatrix} \mathtt{0}_{(\cdot)\times r''} & \mathtt{U} \end{bmatrix} \qquad \hat{\mathtt{X}}_2 = \begin{bmatrix} \mathtt{T}_2^T \\ \mathtt{0}_{r'\times(\cdot)} \end{bmatrix}. \tag{4.68}$$

Here, $(\cdot)$ means *"of compatible size"*. Notice that $\hat{\mathtt{P}}^{IJ}$ and $\hat{\mathtt{P}}^{K}$ both have r columns, and $\hat{\mathtt{X}}_1$ and $\hat{\mathtt{X}}_2$ both have r rows, as $r' + r'' = r$. One can easily check that

$$\hat{\mathtt{P}}\hat{\mathtt{X}} = \begin{bmatrix} \hat{\mathtt{P}}^{IJ} \\ \hat{\mathtt{P}}^{K} \end{bmatrix} \begin{bmatrix} \hat{\mathtt{X}}_1 & \hat{\mathtt{X}}_2 \end{bmatrix} = \begin{bmatrix} \mathtt{S}\mathtt{T}_1^T & \mathtt{S}\mathtt{T}_2^T \\ \mathtt{U}\mathtt{V}^T & \mathtt{0} \end{bmatrix} = \hat{\Lambda} \odot (\mathtt{PX}), \tag{4.69}$$

which is clearly a degenerate solution.

4.4.3 The special case of $\mathbb{P}^3 \to \mathbb{P}^2$

For the classic case of $\mathbb{P}^3 \to \mathbb{P}^2$ projections, the only possible partition $\{I, J, K\}$ is when I is a singleton and J is empty. This is due to the condition

$$r' = 3\,|I| + 2\,|J| = \sum_{i\in I} s_i + \sum_{j\in J}(s_j - 1) < r = 4 \tag{4.70}$$

and the restriction that I is nonempty. In this case $\hat{\Lambda}^{IJ} = \hat{\Lambda}^{I}$ consists of only one row. Further, we have

$$r' = r - \sum_{i\in I} s_i - \sum_{j\in J}(s_j - 1) = 4 - 3 - 0 = 1. \tag{4.71}$$

The reader can check that the condition $\mathcal{R}\mathrm{ank}(\hat{\Lambda}^{K} \odot (\mathtt{P}^{K}\mathtt{X})) \leq r' = 1$ requires that, for generic projection matrices and HD points, only one column of $\hat{\Lambda}^{K}$ can be nonzero[15], causing $\hat{\Lambda}$ to have a cross-shaped structure. Therefore, the theory given in Chapter 3 follows as a special case.

[15]To obtain this result, one should notice that $\hat{\Lambda}^{K}$ cannot have any zero rows, as we have restricted $\hat{\Lambda}$ to have no zero rows and no zero columns.

4.5 Proofs

4.5.1 Proof of Proposition 4.2

Proposition 4.2 (restatement). *Consider a set of projection matrices* $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m$ *with* $\mathtt{P}_i \in \mathbb{R}^{s_i \times r}$ *such that* $\sum_{i=1}^{m}(s_i - 1) \geq r$, *and a* nonzero *point* $\mathbf{X} \neq 0$ *in* $\mathbb{R}^r$. *Now, if the null spaces* $\mathcal{N}(\mathtt{P}_1), \mathcal{N}(\mathtt{P}_2), \ldots, \mathcal{N}(\mathtt{P}_m)$ *as well as* $\mathrm{span}(\mathbf{X})$ *are in general position (with* $\dim(\mathcal{N}(\mathtt{P}_i)) = r - s_i$*), then there is no linear subspace of dimension bigger than or equal to 2 passing through* $\mathbf{X}$ *and nontrivially intersecting* $\mathcal{N}(\mathtt{P}_1), \mathcal{N}(\mathtt{P}_2), \ldots, \mathcal{N}(\mathtt{P}_m)$.

Proof. For brevity of notation let $N_i = \mathcal{N}(\mathtt{P}_i)$. Define the linear subspaces $T_1, T_2, \ldots, T_m$ as follows

$$T_1 = N_1, \tag{4.72}$$

$$T_i = (\mathrm{span}(\mathbf{X}) + T_{i-1}) \cap N_i, \tag{4.73}$$

where the summation of two linear subspaces U and V is defined as $U + V = \{\mathbf{u} + \mathbf{v} \mid \mathbf{u} \in U, \mathbf{v} \in V\}$. As $(\mathrm{span}(\mathbf{X}) + T_{i-1})$ does not depend on N_i, and N_i is in general position, we can assume

$$\dim(T_i) = \dim((\mathrm{span}(\mathbf{X}) + T_{i-1}) \cap N_i) = \max(\dim(\mathrm{span}(\mathbf{X}) + T_{i-1}) + \dim(N_i) - r, 0) \tag{4.74}$$

Since $\dim(\mathrm{span}(\mathbf{X}) + T_{i-1}) \leq \dim(T_{i-1}) + 1$, the above gives

$$\dim(T_i) \leq \max(\dim(T_{i-1}) + \dim(N_i) + 1 - r, 0) \tag{4.75}$$

Now, to get a contradiction we assume that there exist a subspace S, with $\dim(S) \leq 2$ and $\mathbf{X} \in S$, which nontrivially intersects N_i for all i. For each i, let $\mathbf{Y}_i \neq \mathbf{0}$ be a nonzero point in $S \cap N_i$. As $\mathrm{span}(\mathbf{X})$ and N_i are in general location and $\dim(N_i) = r - s_i < r$ we have

$$\mathrm{span}(\mathbf{X}) \cap N_i = \{\mathbf{0}\}. \tag{4.76}$$

As $\mathbf{Y}_i \in N_i$, the above gives $\dim(\mathrm{span}(\mathbf{X}, \mathbf{Y}_i)) = 2$. This, plus the facts $\mathbf{X}, \mathbf{Y}_i \in S$ and $\dim(S) \leq 2$ gives

$$S = \mathrm{span}(\mathbf{X}, \mathbf{Y}_i) \tag{4.77}$$

We show that

$$\mathbf{Y}_i \in T_i, \tag{4.78}$$

This is done by induction. For $i = 1$ this is trivial as $\mathbf{Y}_1 \in N_1 = T_1$. Now, suppose that $\mathbf{Y}_{i-1} \in T_{i-1}$. Thus, from $S = \mathrm{span}(\mathbf{X}, \mathbf{Y}_{i-1})$ we can conclude that $S \subseteq \mathrm{span}(\mathbf{X}) + T_{i-1}$.

Now, by definition of $\mathbf{Y}_i$ we have $\mathbf{Y}_i \in N_i$ and $\mathbf{Y}_i \in S \subseteq \text{span}(\mathbf{X}) + T_{i-1}$. Thus

$$\mathbf{Y}_i \in N_i \cap (\text{span}(\mathbf{X}) + T_{i-1}) = T_i.$$

As $\mathbf{Y}_i$ is nonzero, (4.78) implies that $\dim(T_i) \geq 1$. Therefore, (4.75) gives

$$\dim(T_i) \leq \dim(T_{i-1}) + \dim(N_i) + 1 - r. \tag{4.79}$$

By induction, the above gives

$$\dim(T_m) \leq \sum_{i=1}^{m} \dim(N_i) - (m-1)(r-1). \tag{4.80}$$

By replacing $\dim(N_i) = r - s_i$, we get

$$\dim(T_m) \leq m + r - 1 - \sum_{i=1}^{m} s_i. \tag{4.81}$$

Due to our assumption $\sum_{i=1}^{m}(s_i - 1) \geq r$, we have $\sum_{i=1}^{n} s_i \geq r + m$. This together with (4.81) gives $\dim(T_i) \leq -1$, a contradiction.

□

4.5.2 Proof of Theorem 4.3 (Uniqueness of the Grassmann Tensor)

Theorem 4.3 (restatement). *Consider a setup* $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ *of* m generic *projection matrices, and* n *points* in general position and sufficiently many, *and a valid profile* $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$, *meaning* $\sum_{i=1}^{m} \alpha_i = r$ *and* $\alpha_i \leq s_i - 1$, *such that* $\alpha_i \geq 1$ *for all* i. *Now, for any other configuration* $(\{\hat{\mathtt{P}}_i\}, \{\hat{\mathbf{X}}_j\})$ *with* $\hat{\mathbf{X}}_j \neq \mathbf{0}$ *for all* j, *the set of relations*

$$\hat{\mathtt{P}}_i \hat{\mathbf{X}}_j = \hat{\lambda}_{ij} \mathtt{P}_i \mathbf{X}_j \tag{4.82}$$

implies $\mathcal{G}_\alpha(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m) = \beta\, \mathcal{G}_\alpha(\mathtt{P}_1, \ldots, \mathtt{P}_m)$ *for some scalar* β.

In Sect. 4.2.1 we described the idea of the proof. Here, we state each of the building blocks of the proof as a lemma, and finally prove the theorem.

Lemma 4.9. *Consider an* $r \times r$ *matrix* $\mathtt{Q} = [\mathbf{q}_1\, \mathbf{q}_2 \cdots \mathbf{q}_r]^T$, *with* $\mathbf{q}_i^T$ *denoting its* i*-th row. For a vector* $\mathbf{p} \in \mathbb{R}^r$ *define the matrix* $\mathtt{Q}_{i,\mathbf{p}} = [\mathbf{q}_1, \ldots, \mathbf{q}_{i-1}, \mathbf{p}, \mathbf{q}_{i+1}, \ldots, \mathbf{q}_r]^T$, *that is the matrix* $\mathtt{Q}$ *whose* i*-th row is replaced by* $\mathbf{p}^T$. *Then*

$$\det(\mathtt{Q}_{i,\mathbf{p}}) = (\mathbf{p}^T \mathtt{Q}^{-1} \mathbf{e}_i) \det(\mathtt{Q}) \tag{4.83}$$

where $\mathbf{e}_i$ *is the* i*-th standard basis vector.*

Proof.

$$\begin{aligned}\det(\mathtt{Q}_{i,\mathbf{p}}) &= \det([\mathbf{q}_1,\ldots,\mathbf{q}_{i-1},\mathbf{p},\mathbf{q}_{i+1},\ldots,\mathbf{q}_r]^T)\\ &= \det([\mathbf{e}_1,\ldots,\mathbf{e}_{i-1},\mathtt{Q}^{-T}\mathbf{p},\mathbf{e}_{i+1},\ldots,\mathbf{e}_r]^T\,\mathtt{Q})\\ &= \det([\mathbf{e}_1,\ldots,\mathbf{e}_{i-1},\mathtt{Q}^{-T}\mathbf{p},\mathbf{e}_{i+1},\ldots,\mathbf{e}_r])\,\det(\mathtt{Q})\\ &= (\mathbf{e}_i^T\mathtt{Q}^{-T}\mathbf{p})\,\det(\mathtt{Q}) = (\mathbf{p}^T\mathtt{Q}^{-1}\mathbf{e}_i)\,\det(\mathtt{Q}).\end{aligned}$$

□

Lemma 4.10. *Given the assumptions of Theorem 4.3, the matrix*

$$\mathcal{M}(\mathbf{X}) = \begin{bmatrix} \mathtt{P}_1\mathbf{X} & & & \hat{\mathtt{P}}_1 \\ & \mathtt{P}_2\mathbf{X} & & \hat{\mathtt{P}}_2 \\ & & \ddots & \vdots \\ & & \mathtt{P}_m\mathbf{X} & \hat{\mathtt{P}}_m \end{bmatrix} \tag{4.84}$$

is rank deficient for all $\mathbf{X} \in \mathbb{R}^r$.

Notice that the blank sites of the matrix (4.84) represent zero elements.

Proof. By combining the relations (4.82), that is $\hat{\mathtt{P}}_i\hat{\mathbf{X}}_j = \hat{\lambda}_{ij}\mathtt{P}_i\mathbf{X}_j$, for all i we get

$$\begin{bmatrix} \mathtt{P}_1\mathbf{X}_j & & & \hat{\mathtt{P}}_1 \\ & \mathtt{P}_2\mathbf{X}_j & & \hat{\mathtt{P}}_2 \\ & & \ddots & \vdots \\ & & \mathtt{P}_m\mathbf{X}_j & \hat{\mathtt{P}}_m \end{bmatrix} \begin{pmatrix} \hat{\lambda}_{1j} \\ \hat{\lambda}_{2j} \\ \vdots \\ \hat{\lambda}_{mj} \\ \hat{\mathbf{X}}_j \end{pmatrix} = \mathbf{0}, \qquad j = 1,2,\ldots,n, \tag{4.85}$$

that is

$$\mathcal{M}(\mathbf{X}_j)\begin{pmatrix} \hat{\boldsymbol{\lambda}}_j \\ \hat{\mathbf{X}}_j \end{pmatrix} = \mathbf{0}, \qquad j = 1,2,\ldots,n, \tag{4.86}$$

where the mapping $\mathcal{M}$ was defined in (4.84) and $\hat{\boldsymbol{\lambda}}_j = [\hat{\lambda}_{1j},\ldots,\hat{\lambda}_{mj}]^T$. The matrix $\mathcal{M}(\mathbf{X}_j)$ is $(\sum_{i=1}^m s_i)\times(m+r)$, it is a tall matrix[16] from $\sum_{i=1}^m s_i \geq \sum_{i=1}^m(\alpha_i+1) = r+m$. Since $\hat{\mathbf{X}}_j \neq \mathbf{0}$, (4.86) implies that $\mathcal{M}(\mathbf{X}_j)$ is rank-deficient for $j = 1,\ldots,n$. Let $\mathcal{M}'(\mathbf{X})$ be an arbitrary $(m+r)\times(m+r)$ submatrix of $\mathcal{M}(\mathbf{X})$, made by selecting certain rows of $\mathcal{M}(\mathbf{X})$ (for all $\mathbf{X}$ the same rows are chosen). As, $\mathcal{M}(\mathbf{X}_j)$ is rank deficient, we have $\det(\mathcal{M}'(\mathbf{X}_j)) = 0$. Notice that $\det(\mathcal{M}'(\mathbf{X}))$ is a projective polynomial expression in $\mathbf{X}$ (of degree m and with r variables). If the polynomial defined by $\mathbf{X} \mapsto \det(\mathcal{M}'(\mathbf{X}))$ is not identically zero, the relation $\det(\mathcal{M}'(\mathbf{X})) = 0$ defines a polynomial surface, on which all the points $\mathbf{X}_j$ lie. However, since there are sufficiently many points $\mathbf{X}_j$ in general position, they cannot all lie on a polynomial surface. Therefore, the

[16]Here, a matrix $\mathtt{M} \in \mathbb{R}^{m\times n}$ is called tall if $m \geq n$. Thus, square matrices are also tall.

polynomial $\mathbf{X} \mapsto \det(\mathcal{M}'(\mathbf{X}))$ is identically zero, that is

$$\det(\mathcal{M}'(\mathbf{X})) = 0 \tag{4.87}$$

for all $\mathbf{X} \in \mathbb{R}^r$. This is true for any $(m+r)\times(m+r)$ submatrix $\mathcal{M}'(\mathbf{X})$ of $\mathcal{M}(\mathbf{X})$. Thus, for any $\mathbf{X}$, all $(m+r)\times(m+r)$ submatrices of $\mathcal{M}(\mathbf{X})$ are singular. Therefore, $\mathcal{M}(\mathbf{X})$ is rank-deficient for all $\mathbf{X}$. □

In the proof of the next Lemma we calculate the determinant of $\mathcal{M}(\mathbf{X})$ for a special choice of $\mathbf{X}$. It has been discussed in [Hartley and Schaffalitzky, 2004] that for a square matrix of the form $[\mathtt{A},\mathtt{B}]$, the determinant is given by

$$\det([\mathtt{A},\mathtt{B}]) = \sum_I \operatorname{sign}(I)\det(\mathtt{A}^I)\det(\mathtt{B}^{\bar{I}}), \tag{4.88}$$

where the summation is over all index sets I of size equal to the number of columns of $\mathtt{A}$, the set $\bar{I}$ is the complement of I, $\mathtt{A}^I$ is the submatrix of $\mathtt{A}$ created by choosing rows in order according to I and similarly is defined $\mathtt{B}^{\bar{I}}$. Depending on whether the sequence $(\operatorname{sort}(I), \operatorname{sort}(\bar{I}))$ represents an even or odd permutation, $\operatorname{sign}(I)$ is equal to $+1$ or -1.

Lemma 4.11. *Assume the assumptions of Theorem 4.3, and consider two submatrices* $\mathtt{Q}$ *and* $\mathtt{Q}'$ *of* $\mathtt{P} = \operatorname{stack}(\mathtt{P}_1,\ldots,\mathtt{P}_m)$ *chosen according to* $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_m)$*, such that all rows of* $\mathtt{Q}$ *and* $\mathtt{Q}'$ *are equal except for the l-th rows* $\mathbf{q}_l$ *and* $\mathbf{q}'_l$*, which are chosen from different rows of* $\mathtt{P}_k$ *for some k. Similarly, consider submatrices* $\hat{\mathtt{Q}}$ *and* $\hat{\mathtt{Q}}'$ *of* $\hat{\mathtt{P}}$ *made by choosing the corresponding rows from* $\hat{\mathtt{P}} = \operatorname{stack}(\hat{\mathtt{P}}_1,\ldots,\hat{\mathtt{P}}_m)$*. If* $\det(\mathtt{Q}) \neq 0$ *we have*

$$\det(\hat{\mathtt{Q}}') = \frac{\det(\mathtt{Q}')}{\det(\mathtt{Q})}\det(\hat{\mathtt{Q}}) \tag{4.89}$$

Proof. For convenience, we assume that $\mathtt{Q}$ (similarly $\hat{\mathtt{Q}}$) is made by choosing first α_i rows from each $\mathtt{P}_i$ ($\hat{\mathtt{P}}_i$), and $\mathtt{Q}'$ (similarly $\hat{\mathtt{Q}}'$) are made by choosing the same rows as for $\mathtt{Q}$ ($\hat{\mathtt{Q}}$), except instead of choosing the α_1-th row of $\mathtt{P}_1$ ($\hat{\mathtt{P}}_1$) we choose the (α_1+1)-th row. The proof for other cases are similar. Therefore, if we denote the i-th row of $\mathtt{Q}$ by $\mathbf{q}_i^T$ and the (α_1+1)-th row of $\mathtt{P}_1$ by $\mathbf{p}_1^T$, then we have

$$\mathtt{Q}' = [\mathbf{q}_1,\ldots,\mathbf{q}_{\alpha_i-1},\mathbf{p}_1,\mathbf{q}_{\alpha_i+1},\ldots,\mathbf{q}_r]^T \tag{4.90}$$

For ease of notation, let $\beta_i = \alpha_i + 1$, and let $\mathtt{P}_i^{1..\beta_i}$ represent the matrix made by choosing first β_i rows from $\mathtt{P}_i$. Consider the matrix $\mathcal{M}(\mathbf{X})$ defined in (4.84) and define the $(m+r)\times(m+r)$ submatrix $\mathcal{M}'(\mathbf{X})$ of $\mathcal{M}(\mathbf{X})$ as

$$\mathcal{M}'(\mathbf{X}) = \begin{bmatrix} \mathtt{P}_1^{1..\beta_1}\mathbf{X} & & & & \hat{\mathtt{P}}_1^{1..\beta_1} \\ & \mathtt{P}_2^{1..\beta_2}\mathbf{X} & & & \hat{\mathtt{P}}_2^{1..\beta_2} \\ & & \ddots & & \vdots \\ & & & \mathtt{P}_m^{1..\beta_m}\mathbf{X} & \hat{\mathtt{P}}_m^{1..\beta_m} \end{bmatrix} \tag{4.91}$$

From Lemma 4.10, we have $\det(\mathcal{M}'(\mathbf{X})) = 0$ for all $\mathbf{X}$. Set $\mathbf{X} = \mathtt{Q}^{-1}\mathbf{e}_{\alpha_i}$, where $\mathbf{e}_{\alpha_i} \in \mathbb{R}^r$ is the α_i-th standard basis vector. Remember that $\mathtt{Q}$ is the submatrix of $\mathtt{P} = \text{stack}(\mathtt{P}_1, \ldots, \mathtt{P}_n)$ created by choosing first α_i rows from each $\mathtt{P}_i$. Choosing the same rows (as the ones created $\mathtt{Q}$) from the vector $\mathtt{P}\mathtt{Q}^{-1}\mathbf{e}_{\alpha_i}$ results in the vector $\mathtt{Q}\mathtt{Q}^{-1}\mathbf{e}_{\alpha_i} = \mathbf{e}_{\alpha_i}$. Thus, for $\mathbf{X} = \mathtt{Q}^{-1}\mathbf{e}_{\alpha_i}$ we have[17]

$$\mathtt{P}_1^{1..\beta_1}\mathbf{X} = \begin{pmatrix} \mathbf{0}_{\alpha_1 - 1} \\ 1 \\ \gamma_1 \end{pmatrix} = \mathbf{e}_{\alpha_1} + \gamma_i \mathbf{e}_{\beta_1} \tag{4.92}$$

$$\mathtt{P}_i^{1..\beta_i}\mathbf{X} = \begin{pmatrix} \mathbf{0}_{\alpha_i} \\ \gamma_i \end{pmatrix} = \gamma_i \mathbf{e}_{\beta_i} \qquad i = 2, \ldots, m \tag{4.93}$$

where the scalars γ_i are defined as

$$\gamma_i = \mathbf{p}_i^T \mathtt{Q}^{-1} \mathbf{e}_{\alpha_i} \tag{4.94}$$

with $\mathbf{p}_i^T$ representing the β_i-th (that is (α_i+1)-th) row of $\mathtt{P}_i$. Note that

1. By genericity of $\mathtt{P}_i$-s we can assume that γ_i-s are all nonzero (as $\mathbf{p}_i$-s and $\mathtt{Q}$ come from rows of $\mathtt{P}_i$-s.).

2. From (4.90), Lemma 4.9 gives

$$\det(\mathtt{Q}') = (\mathbf{p}_1^T \mathtt{Q}^{-1} \mathbf{e}_{\alpha_i}) \det(\mathtt{Q}) = \gamma_1 \det(\mathtt{Q}) \tag{4.95}$$

By replacing $\mathtt{P}_i^{1..\beta_i}\mathbf{X}$ given by (4.92) and (4.93) in (4.91) we have

$$\mathcal{M}'(\mathbf{X}) = \begin{bmatrix} \begin{pmatrix} \mathbf{0} \\ 1 \\ \gamma_1 \end{pmatrix} & & & \hat{\mathtt{P}}_1^{1..\beta_1} \\ & \begin{pmatrix} \mathbf{0} \\ \gamma_2 \end{pmatrix} & & \hat{\mathtt{P}}_2^{1..\beta_2} \\ & & \ddots & \vdots \\ & & \begin{pmatrix} \mathbf{0} \\ \gamma_m \end{pmatrix} & \hat{\mathtt{P}}_m^{1..\beta_m} \end{bmatrix} \tag{4.96}$$

By using the formula (4.88), one can obtain that for $\mathbf{X} = \mathtt{Q}^{-1}\mathbf{e}_{\alpha_i}$ we have

$$\det(\mathcal{M}'(\mathbf{X})) = \pm(\prod_{i=2}^{m} \gamma_i)(\gamma_1 \cdot \det(\hat{\mathtt{Q}}) - 1 \cdot \det(\hat{\mathtt{Q}}')). \tag{4.97}$$

Where $\hat{\mathtt{Q}}$ and $\hat{\mathtt{Q}}'$ were defined in the lemma. From Lemma 4.10, we have $\det(\mathcal{M}'(\mathbf{X})) = 0$. As, we assumed $\gamma_i \neq 0$ for all i, setting (4.97) equal to zero

[17]Notice that the standard basis vector $\mathbf{e}_i$ in each equation is of compatible size. For example, $\mathbf{e}_{\alpha_1}$ is of size β_1 in (4.92), while it is of size r in the expression $\mathtt{Q}^{-1}\mathbf{e}_{\alpha_i}$ or in (4.94).

gives

$$\det(\hat{\mathtt{Q}}') = \gamma_1 \det(\hat{\mathtt{Q}}) \tag{4.98}$$

Since we have assumed that $\det(\mathtt{Q}) \neq 0$, (4.95) and (4.98) together give $\det(\hat{\mathtt{Q}}') = \frac{\det(\mathtt{Q}')}{\det(\mathtt{Q})} \det(\hat{\mathtt{Q}})$. □

Proof of Theorem 4.3. As $\mathtt{P}_i$-s are generic we assume that all minors of $\mathtt{P} = \text{stack}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$, chosen according to the profile $\boldsymbol{\alpha}$ are nonzero[18]. By starting with a submatrix $\mathtt{Q}$ of $\mathtt{P}$ according to $\boldsymbol{\alpha}$, and updating $\mathtt{Q}$ by changing one of its rows at a time in the way described in Lemma 4.11, we can finally traverse all possible submatrices chosen according to $\boldsymbol{\alpha}$. As we assume that $\det(\mathtt{Q}) \neq 0$ for all those submatrices, according to Lemma 4.11 during this procedure the ratio $\beta = \det(\hat{\mathtt{Q}})/\det(\mathtt{Q})$ stays the same. This means that each element of $\mathcal{G}_{\boldsymbol{\alpha}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m)$ is β times the corresponding element of $\mathcal{G}_{\boldsymbol{\alpha}}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$, implying $\mathcal{G}_{\boldsymbol{\alpha}}(\hat{\mathtt{P}}_1, \ldots, \hat{\mathtt{P}}_m) = \beta\, \mathcal{G}_{\boldsymbol{\alpha}}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$. □

4.5.3 Proof of Lemma 4.7

Lemma 4.7 (restatement). *Consider m projection matrices $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m$ with $\mathtt{P}_i \in \mathbb{R}^{s_i \times r}$, such that $\sum_{i=1}^{n}(s_i - 1) \geq r$, and $\mathtt{P} = \text{stack}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$ has* full column rank r. *If $\mathtt{P}$ has no full rank $r \times r$ submatrix chosen by strictly fewer than s_i rows form each $\mathtt{P}_i$, then there exists a partition $\{I, J, K\}$ of the set of views $\{1, 2, \ldots, m\}$, with $I \neq \emptyset$ (nonempty) and $\sum_{i \in I} s_i + \sum_{i \in J}(s_i - 1) \leq r$, such that $\mathtt{P}^K = \text{stack}(\{\mathtt{P}_i\}_{i \in K})$ has rank $r' = r - \sum_{i \in I} s_i - \sum_{i \in J}(s_i - 1)$. Further, the row space of $\mathtt{P}^K$ is spanned by the rows of an $r' \times r$ submatrix $\mathtt{Q}^K = \text{stack}(\{\mathtt{Q}_i\}_{i \in K})$ of $\mathtt{P}^K$, where each $\mathtt{Q}_i$ is created by choosing strictly less than s_i rows from $\mathtt{P}_i$.*

The above is a restatement of Lemma 4.7. However, for simplicity, instead of the *hatted* quantities like $\hat{\mathtt{P}}_i$ we have used the *unhatted* ones like $\mathtt{P}_i$.

The proof of this lemma can be somehow confusing. Thus, before giving the full proof, we give the reader some ideas about our approach. Notice that, as $\mathtt{P} = \text{stack}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$ has full column rank, it has an $r \times r$ non-singular submatrix $\mathtt{Q}$. This submatrix has chosen according to a (not necessarily valid) profile $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ by choosing α_i rows from each $\mathtt{P}_i$. In fact, $\boldsymbol{\alpha}$ cannot be valid due to the assumption of the lemma that $\mathtt{P}$ has no full-rank $r \times r$ submatrix chosen by strictly fewer than s_i rows form each $\mathtt{P}_i$. Therefore, every non-singular submatrix $\mathtt{Q}$ has s_i rows from $\mathtt{P}_i$ for at least one view i. In other words $\alpha_i = s_i$ for one or more indices i. Partition the set of views $\{1, 2, \ldots, m\}$ into three subsets I, J and L such that

$$I = \{i \,|\, \alpha_i = s_i\} \tag{4.99}$$

$$J = \{i \,|\, \alpha_i = s_i - 1\} \tag{4.100}$$

$$L = \{i \,|\, \alpha_i \leq s_i - 2\}. \tag{4.101}$$

In other words, I contains the indices of the projection matrices $\mathtt{P}_i$ whose all rows

[18] Though the proof is possible under a milder assumption.

contribute into making $\mathtt{Q}$, J contains the indices those $\mathtt{P}_i$-s whose all but one rows contribute into making $\mathtt{Q}$, and L contains the rest of views. The matrix $\mathtt{P}$ might have more than one non-singular submatrix. From all those possible cases, we choose a submatrix with the least number of indices i for which $\alpha_i = s_i$. In this case, corresponding subset I has minimal size among the possible choices of non-singular submatrices of $\mathtt{P}$. We say that $\mathtt{Q}$ is a submatrix with minimal I. Notice that I cannot be empty, otherwise $\boldsymbol{\alpha}$ would be a valid profile, that is $\mathtt{P} = \mathrm{stack}(\mathtt{P}_1, \ldots, \mathtt{P}_m)$ has a non-singular $r \times r$ submatrix, namely $\mathtt{Q}$, chosen by strictly fewer than s_i rows form each $\mathtt{P}_i$.

For any index set $K \subseteq \{1, 2, \ldots, m\}$ we denote by $\mathtt{P}^K$ the stack of projection matrices whose indices are contained in K, that is

$$\mathtt{P}^K = \mathrm{stack}(\{\mathtt{P}_i\}_{i \in K}).$$

This way we can divide the matrix $\mathtt{P}$ into $\mathtt{P}^I$, $\mathtt{P}^J$ and $\mathtt{P}^L$. We split each projection matrix $\mathtt{P}_i$ into two submatrices $\mathtt{Q}_i$ and $\mathtt{R}_i$, correspondingly consisting of rows of $\mathtt{P}_i$ which are or are not included in the submatrix $\mathtt{Q}$. Therefore, $\mathtt{Q}_i$ and $\mathtt{R}_i$ respectively have α_i and $s_i - \alpha_i$ rows, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ is the (invalid) profile according to which $\mathtt{Q}$ is chosen. Notice that

$$\mathtt{Q} = \mathrm{stack}(\mathtt{Q}_1, \mathtt{Q}_2, \ldots, \mathtt{Q}_m). \tag{4.102}$$

Notice that for $i \in I$ we have $\mathtt{P}_i = \mathtt{Q}_i$. Therefore, $\mathtt{R}_i$ cannot be defined for $i \in I$ as it would have zero rows. Any $\mathtt{R}_j$ with $j \in J$ has exactly one row. If for some view i no row of $\mathtt{P}_i$ is chosen to form $\mathtt{Q}$, that is $\alpha_i = 0$, then we have $\mathtt{P}_i = \mathtt{R}_i$. In this case $\mathtt{Q}_i$ does not exist, however, one could think of $\mathtt{Q}_i$ as a matrix with *zero* rows so that (4.102) can be used consistently. Similarly to $\mathtt{P}^K$, we for any subset K of views we define

$$\mathtt{Q}^K = \mathrm{stack}(\{\mathtt{Q}_i\}_{i \in K}),$$
$$\mathtt{R}^K = \mathrm{stack}(\{\mathtt{R}_i\}_{i \in K}).$$

Notice that $\mathtt{R}^I$ does not exist. The general strategy of the proof is to take a row $\mathbf{r}^T$ from some $\mathtt{R}_i$ and make it replace a row in $\mathtt{Q}$ to have a new $r \times r$ submatrix $\mathtt{Q}'$, such that $\mathtt{Q}'$ is also non-singular. This action can be done repeatedly. For each new submatrix $\mathtt{Q}'$ we can define a corresponding partition $\{I', J', L'\}$ in the same way $\{I, J, L\}$ was defined for $\mathtt{Q}$. The key fact used in the proof is that we can never have a situation in which size of I' is smaller than the size of I. This is because I is assumed to be minimal.

To be succinct, given a row vector $\mathbf{r}^T$ and the $r \times r$ submatrix $\mathtt{Q} = \mathrm{stack}(\mathtt{Q}_1, \mathtt{Q}_2, \ldots, \mathtt{Q}_m)$, we use the term

"$\mathbf{r}^T$ *can replace a row of* $\mathtt{Q}$" or "$\mathbf{r}^T$ *can replace a row of* $\mathtt{Q}_i$ *in* $\mathtt{Q}$",

and by that we mean that the replacement can be done such that the resulting submatrix $\mathtt{Q}'$ is still non-singular.

To better understand the idea behind the proof, we first consider a special case

in which the subset J is empty (for a submatrix $\mathtt{Q}$ with minimal I). In this case the proof of Lemma 4.7 is simple. By possibly relabeling the views, we can assume that $\mathtt{P} = \text{stack}(\mathtt{P}^I, \mathtt{P}^L)$, $\mathtt{Q} = \text{stack}(\mathtt{Q}^I, \mathtt{Q}^L)$ and $\mathtt{R} = \text{stack}(\mathtt{R}^I, \mathtt{R}^L)$. Consider an arbitrary row $\mathbf{r}^T$ of $\mathtt{R}_l$ for some $l \in L$. Assume $\mathbf{r}^T$ can replace[19] a matrix $\mathtt{Q}_i$ in $\mathtt{Q}$ for some $i \in I$, resulting in a new submatrix $\mathtt{Q}'$. This submatrix is chosen according to a profile $\boldsymbol{\alpha}' = (\alpha'_1, \ldots, \alpha'_m)$ defined by

$$\alpha'_i = \alpha_i - 1, \tag{4.103}$$

$$\alpha'_l = \alpha_l + 1, \tag{4.104}$$

$$\alpha'_t = \alpha_t \quad \text{for all } l \notin \{i, l\} \tag{4.105}$$

The above is due to the fact that one row of $\mathtt{R}_l$ has replaced a row of $\mathtt{Q}_i$ in $\mathtt{Q}$. As $i \in I$ and $l \in L$, we have $\alpha_i = s_i$ and $\alpha_l \leq s_l - 2$, and thus, $\alpha'_i = s_i - 1$ and $\alpha'_l \leq s_l - 1$. Now, if we define the partition $\{I', J', L'\}$ for the new submatrix $\mathtt{Q}'$ (in the same way I, J, L was defined for $\mathtt{Q}$) we know that the index $i \in I$ is no longer in I' (as $\alpha'_i = s_i - 1$). The index $l \in L$ either remains in L' or moves to J' depending on whether $\alpha'_l < s_i - 1$ or $\alpha'_l = s_l - 1$. It can never move to I' as $\alpha'_l < s_l$. Therefore, we have $I' = I \setminus \{i\}$, which gives

$$|I'| = |I| - 1, \tag{4.106}$$

where $|\cdot|$ denotes the size of a set. This, however, is a contradiction since we have assumed that I has minimal size. Therefore, now row of $\mathtt{Q}_i$ in $\mathtt{Q}$ can be replaced by $\mathbf{r}^T$. As i was chosen arbitrarily from I, we conclude that $\mathbf{r}^T$ cannot replace any row of $\mathtt{Q}^I$. Therefore, as $\mathtt{Q} = \text{stack}(\mathtt{Q}^I, \mathtt{Q}^L)$, according to Corollary 4.4, $\mathbf{r}^T$ must belong to the row space of $\mathtt{Q}^L$. Since $\mathbf{r}^T$ can be chosen to be any arbitrary row of $\mathtt{R}_l$ for any $l \in L$, it means that all rows of $\mathtt{R}^L = \text{stack}(\{\mathtt{R}_l\}_{l \in L})$ are in the row space of $\mathtt{Q}^L$. Notice that $\mathtt{P}^L$ is equal to $\text{stack}(\mathtt{Q}^L, \mathtt{R}^L)$ up to permutation of rows. Therefore, all rows of $\mathtt{P}^L$ are in the row space of $\mathtt{Q}^L$. Since $\mathtt{Q} = \text{stack}(\mathtt{Q}^I, \mathtt{Q}^L)$ is non-singular, the row space of $\mathtt{Q}^L$ has dimension $r' = r - \sum_{i \in I} s_i$. Further, $\mathtt{Q}^L$ is equal to $\text{stack}(\{\mathtt{Q}_l\}_{l \in L})$, and for all $l \in L$ the matrix $\mathtt{Q}_l$ is made by choosing strictly less than s_i rows (in fact less than $s_i - 1$ rows) from each $\mathtt{P}_l$. This completes the proof of Lemma 4.7 for the special case of $J = \emptyset$.

The proof, however, is more difficult when J is nonempty. In this case, by choosing $\mathbf{r}^T$ from the rows of $\mathtt{R}^L$ and using the same argument as above, we can prove that all rows of $\mathtt{P}^L$ is in the row space of $\text{stack}(\mathtt{Q}^J, \mathtt{Q}^L)$, rather than the row space of $\text{stack}(\mathtt{Q}^L)$. If we choose $\mathbf{r}^T$ from the rows of $\mathtt{R}^J$, the above argument does not apply, because if a row $\mathbf{r}^T$ of $\mathtt{R}^J$ replaces a row of $\mathtt{Q}^I$ in $\mathtt{Q}$, for the corresponding partition $\{I', J', L'\}$ of the new submatrix $\mathtt{Q}'$ we have $|I'| = |I|$. The reason is as follows: Consider two indices $j \in J$ and $i \in I$ and assume that a row $\mathbf{r}^T$ of $\mathtt{R}_j$ has replaced a row of $\mathtt{Q}_i$ in $\mathtt{Q}$ resulting in a new non-singular $r \times r$ submatrix $\mathtt{Q}'$. Notice that in this case $\mathtt{R}_j$ has only one row as $j \in J$. Let $\boldsymbol{\alpha}' = (\alpha'_1, \ldots, \alpha'_m)$ be the profile according to which

[19] Remember that by replacing we mean replacing such that the resulting $r \times r$ submatrix remains non-singular.

$\mathtt{Q}'$ is chosen. Then, as a row of $\mathtt{R}_j$ has replaced a row of $\mathtt{Q}_i$ in $\mathtt{Q}$, we have

$$\alpha'_i = \alpha_i - 1, \tag{4.107}$$
$$\alpha'_j = \alpha_j + 1, \tag{4.108}$$
$$\alpha'_t = \alpha_t \quad \text{for all } l \notin \{i,j\} \tag{4.109}$$

Notice that $\alpha_i = s_i$ and $\alpha_j = s_j - 1$ (as $i \in I, j \in J$). Thus, $\alpha'_i = s_i - 1$ and $\alpha'_j = s_j$. Therefore, the number of indices l for which we have $\alpha'_t = s_t$ remains the same as the number of cases for which $\alpha_t = s_t$. In other words, by defining I', J', L' for $\mathtt{Q}'$ as I, J, L where defined for $\mathtt{Q}$, we have $I' = (I \setminus \{i\}) \cup \{j\}$, and hence, $|I'| = |I|$. Therefore, the same argument as in the case of $J = \emptyset$ cannot be applied.

To prove Lemma 4.7 for the general case, we will show that there exists an index set K with $L \subseteq K \subseteq (L \cup J)$ such that the rows of $\mathtt{P}^K$ are all in the row space of $\mathtt{Q}^K$. The rest of the proof is straightforward. The views can be partitions into subsets I, $J \setminus K$ and K. We argued before that I cannot be empty. Since $\mathtt{Q} \in \mathbb{R}^{r \times r}$ has full rank, the rank of $\mathtt{Q}^K$ is equal to its number of rows, that is $r' = r - \sum_{i \in I} s_i - \sum_{i \in J \setminus K}(s_i - 1)$, which is also equal to the rank of $\mathtt{Q}^K$. Therefore, $\mathtt{P}^K$ has also rank r' since its row space is spanned by the rows of $\mathtt{Q}^K$. Further, $\mathtt{Q}^K$ is in the form of stack$(\{\mathtt{Q}_k\}_{k \in K})$, and since $K \subseteq L \cup J$, for every $k \in K$ the matrix $\mathtt{Q}_k$ is created by choosing strictly less than s_k rows from $\mathtt{P}_k$.

Now, it is left to prove the following:

Lemma 4.12. *There exists a subset K, with $L \subseteq K \subseteq (L \cup J)$, such that the row space of $\mathtt{P}^K$ is spanned by the rows of $\mathtt{Q}^K$.*

Before starting the proof, we introduce the following notation. For two matrices $\mathtt{A}$ and $\mathtt{B}$ of the same size the relation

$$\mathtt{A} \equiv \mathtt{B} \tag{4.110}$$

means that $\mathtt{A}$ equals $\mathtt{B}$ up to permutation of rows. For example, we can say $\mathtt{Q} \equiv$ stack$(\mathtt{Q}^I, \mathtt{Q}^J, \mathtt{Q}^L)$ and $\mathtt{Q}^{L \cup J} \equiv$ stack$(\mathtt{Q}^J, \mathtt{Q}^L)$.

Proof. For an index set $\Gamma \subseteq \{1, 2, \ldots, m\}$ define

$$\mathcal{S}(\Gamma) = \{l \mid \text{some row } r^T \text{ of } \mathtt{R}^\Gamma \text{ can replace a row of } \mathtt{Q}_l \text{ in } \mathtt{Q}\}. \tag{4.111}$$

We remind the reader that $\mathtt{P}^\Gamma = \text{stack}(\{\mathtt{P}_i\}_{i \in \Gamma})$, and by a row $\mathbf{r}^T$ *being able* to replace some $\mathtt{Q}_l$ in $\mathtt{Q}$ we mean replacing such that the resulting $r \times r$ submatrix $\mathtt{Q}'$ is non-singular. Notice that $\mathcal{S}(\Gamma) \subseteq \{1, 2, \ldots, m\}$. Now, define the sequence of sets L^t as follows

$$L^0 = L \tag{4.112}$$
$$L^t = L^{t-1} \cup \mathcal{S}(L^{t-1}) \tag{4.113}$$

Let $\bar{L}^t = \{1,2,\ldots,m\} \setminus L^t$ be the complement of L^t. From (4.113) it follows that $\mathcal{S}(L^{t-1}) \cap \bar{L}^t = \emptyset$. Therefore, given any row $\mathbf{r}^T$ of $\mathtt{R}^{L^{t-1}}$, using the definition of $\mathcal{S}(L^{t-1})$, we can say that $\mathbf{r}^T$ cannot replace any row of $\mathtt{Q}^{\bar{L}^t}$ in $\mathtt{Q}$. As $\mathtt{Q} \equiv \text{stack}(\mathtt{Q}^{\bar{L}^t}, \mathtt{Q}^{L^t})$ (that is $\mathtt{Q}$ is equal to $\text{stack}(\mathtt{Q}^{\bar{L}^t}, \mathtt{Q}^{L^t})$ up to permutation of rows), by Corollary 4.4 we conclude that $\mathbf{r}^T$ is in the row space of $\mathtt{Q}^{\bar{L}^t}$. Since this is true for any row $\mathbf{r}^T$ of $\mathtt{R}^{L^{t-1}}$, it follows that

$$\mathcal{R}(\mathtt{R}^{L^{t-1}}) \subseteq \mathcal{R}(\mathtt{Q}^{L^t}), \tag{4.114}$$

where $\mathcal{R}$ gives the row space of a matrix. From (4.113) we have $L^{t-1} \subseteq L^t$, and thus

$$\mathcal{R}(\mathtt{Q}^{L^{t-1}}) \subseteq \mathcal{R}(\mathtt{Q}^{L^t}). \tag{4.115}$$

As $\mathtt{P}^{L^{t-1}} \equiv \text{stack}(\mathtt{Q}^{L^{t-1}}, \mathtt{R}^{L^{t-1}})$, the relations (4.114) and (4.115) imply that

$$\mathcal{R}(\mathtt{P}^{L^{t-1}}) \subseteq \mathcal{R}(\mathtt{Q}^{L^t}) \tag{4.116}$$

From (4.113) it follows that $L^0 \subseteq L^1 \subseteq L^2 \subseteq \cdots$, and also that L^t is always a subset of the *finite* set of views $\{1,2,\ldots,m\}$. Therefore, we must have $L^{t^*} = L^{t^*+1}$ for some t^*. Since (4.113) is in the form of $L^t = \mathcal{F}(L^{t-1})$ for some mapping $\mathcal{F}$, the equality $L^{t^*} = L^{t^*+1}$ implies

$$L^t = L^{t^*} \qquad \text{for all } t \geq t^*. \tag{4.117}$$

We choose the set K as

$$K = L^{t^*} \tag{4.118}$$

and will show that K has the properties mentioned in the lemma. First, notice that, by induction, from $L^{t-1} \subseteq L^t$, we get $L = L^0 \subseteq L^{t^*}$, therefore

$$L \subseteq K. \tag{4.119}$$

Also, from $L^{t^*-1} = L^{t^*}$, the relation (4.116) gives $\mathcal{R}(\mathtt{P}^{L^{t^*}}) \subseteq \mathcal{R}(\mathtt{Q}^{L^{t^*}})$, that is

$$\mathcal{R}(\mathtt{P}^K) \subseteq \mathcal{R}(\mathtt{Q}^K). \tag{4.120}$$

As $\mathtt{Q}^K$ is a submatrix of $\mathtt{P}^K$, it follows that

$$\mathcal{R}(\mathtt{P}^K) = \mathcal{R}(\mathtt{Q}^K). \tag{4.121}$$

This means that rows of $\mathtt{Q}^K$ span the row space of $\mathtt{P}^K$.

Now, it is only left to prove that $K \subseteq (L \cup J)$. This is indeed the hardest part of the proof. Notice that as $\{I, J, K\}$ is a partition of views $\{1,2,\ldots,m\}$, this is equivalent to proving $K \cap I = \emptyset$.

Define the sequence C^t as

$$C^0 = L^0 = L, \tag{4.122}$$

$$C^t = L^t \setminus L^{t-1}. \tag{4.123}$$

Notice that the sets $C^0, C^1, \ldots, C^{t^*}$ partition $L^{t^*} = K$. Obviously, $C^t = \emptyset$ for $t > t^*$. Therefore, every pair of sets C^t and $C^{t'}$ with $t \neq t'$ are non-intersecting.

To get a contradiction, assume that $K \cap I \neq \emptyset$. Then there is an index $k \in K \cap I$. As $I \cap L = \emptyset$, we have $k \in K \setminus L$. We will show in Lemma 4.13 that in this case there exists a chain of distinct indices

$$j_0, j_1, \ldots, j_p$$

with $j_t \in C^t$ and $j_p = k$, such that for every $t < p$, there exists a row of $\mathtt{R}_{j_t}$ which can replace some row of $\mathtt{Q}_{j_{t+1}}$ in $\mathtt{Q}$ (giving a non-singular matrix). For each t we represent such a row of $\mathtt{R}_{j_t}$ by $\mathbf{r}_{j_t}^T$ and such row of $\mathtt{Q}_{j_{t+1}}$ by $\mathbf{q}_{j_{t+1}}^T$:

> $\mathbf{r}_{j_t}^T$ and $\mathbf{q}_{j_{t+1}}^T$ are respectively rows of $\mathtt{R}_{j_t}$ and $\mathtt{Q}_{j_{t+1}}$, chosen such that by removing $\mathbf{q}_{j_{t+1}}^T$ from $\mathtt{Q}$ and putting $\mathbf{r}_{j_t}^T$ in its place, the resulting submatrix is non-singular.

Remember that, as $j_t \in C^t \subseteq L^t$, from (4.114) we have

$$\mathbf{r}_{j_t}^T \in \mathcal{R}(\mathtt{Q}^{L^{t+1}}), \tag{4.124}$$

where $\mathcal{R}(\cdot)$ represents the row space of a matrix.

Now, we define the sequence

$$\mathtt{Q}_{(0)}, \mathtt{Q}_{(1)}, \ldots, \mathtt{Q}_{(p)} \tag{4.125}$$

of $r \times r$ submatrices of $\mathtt{P}$ as follows[20]. Let $\mathtt{Q}_{(0)} = \mathtt{Q}$. Now, according to our discussion above, we know that there exists a row of $\mathtt{R}_{j_0}$, namely $\mathbf{r}_{j_0}^T$, which can replace a row of $\mathtt{Q}_{j_0}$ in $\mathtt{Q} \in \mathbb{R}^{r \times r}$ such that the resuting matrix $\mathtt{Q}' \in \mathbb{R}^{r \times r}$ is non-singular. We define $\mathtt{Q}_{(1)} = \mathtt{Q}'$. Similarly, we can define $\mathtt{R}_{(1)}$ as the submatrix of $\mathtt{P}$ created by the rows of $\mathtt{P}$ which are not chosen for $\mathtt{Q}_{(1)}$. Now we can observe that the rows of the matrix $\mathtt{R}_{j_1}$ in $\mathtt{R} = \mathtt{R}_{(0)}$ are still contained in $\mathtt{R}_{(1)}$, and also the rows of $\mathtt{Q}_{j_2}$ in $\mathtt{Q} = \mathtt{Q}_{(0)}$ are still contained in $\mathtt{Q}_{(1)}$. We make the row $\mathbf{r}_{j_1}^T$ of $\mathtt{R}_{j_1}$ replace the row $\mathbf{q}_{j_2}^T$ of $\mathtt{Q}_{j_2}$ in $\mathtt{Q}_{(1)}$ to get a new $r \times r$ matrix $\mathtt{Q}_{(2)}$. Notice that we have not yet made any claim about the non-singularity of $\mathtt{Q}_{(2)}$. In general, starting by $\mathtt{Q}_{(0)} = \mathtt{Q}$ and $\mathtt{R}_{(0)} = \mathtt{R}$, the sequences $\mathtt{Q}_{(t)}$ and $\mathtt{R}_{(t)}$ are defined recursively as follows:

> The matrices $\mathtt{Q}_{(t+1)} \in \mathbb{R}^{r \times r}$ and $\mathtt{R}_{(t+1)}$ are created by picking $\mathbf{r}_{j_t}^T$ from $\mathtt{R}_{(t)}$ and $\mathbf{q}_{j_{t+1}}^T$ from $\mathtt{Q}_{(t)}$ and swapping their places. In other words, $\mathbf{r}_{j_t}^T$ replaces $\mathbf{q}_{j_{t+1}}^T$ in $\mathtt{Q}_{(t)}$ to create $\mathtt{Q}_{(t+1)}$, and $\mathbf{q}_{j_{t+1}}^T$ replaces $\mathbf{r}_{j_t}^T$ in $\mathtt{R}_{(t)}$ to create $\mathtt{R}_{(t+1)}$.

Clearly, we first need to show that the above definition is well-defined by showing that $\mathbf{r}_{j_t}^T$ and $\mathbf{q}_{j_{t+1}}^T$ are respectively among the rows of $\mathtt{R}_{(t)}$ and $\mathtt{Q}_{(t)}$. In Lemma 4.14 we

[20] One should distinguish between $\mathtt{Q}_{(t)} \in \mathbb{R}^{r \times r}$ and $\mathtt{Q}_i \in \mathbb{R}^{\alpha_i \times r}$.

will prove this by showing that $\mathtt{R}_{j_t}$ and $\mathtt{Q}_{j_{t+1}}$ are respectively contained in $\mathtt{R}_{(t)}$ and $\mathtt{Q}_{(t)}$. Notice that we have not yet stated any claim as to whether or not $\mathtt{Q}_{(t)}$ is non-singular.

For each submatrix $\mathtt{Q}_{(t)} \in \mathbb{R}^{r\times r}$ of $\mathtt{P}$ one can associate a corresponding profile $\boldsymbol{\alpha}^{(t)} = (\alpha_1^{(t)}, \ldots, \alpha_m^{(t)})$. This means that $\mathtt{Q}_{(t)}$ is created by choosing $\alpha_i^{(t)}$ rows from each $\mathtt{P}_i$. Using the recursive definition of $\mathtt{Q}_{(t)}$ we have

$$\alpha_{j_t}^{(t+1)} = \alpha_{j_t}^{(t)} + 1 \tag{4.126}$$

$$\alpha_{j_{t+1}}^{(t+1)} = \alpha_{j_{t+1}}^{(t)} - 1 \tag{4.127}$$

$$\alpha_l^{(t+1)} = \alpha_l^{(t)} \qquad l \notin \{j_t, j_{t+1}\} \tag{4.128}$$

for $t = 0, 1, \ldots, p-1$. Using the above, for each i, we can start from $\alpha_i^{(0)} = \alpha_i$ and calculate $\alpha_i^{(1)}, \alpha_i^{(2)}, \ldots, \alpha_i^{(p)}$. As the indices $j_0, j_1, \ldots, j_p$ are distinct, the above gives

$$\alpha_i^{(p)} = \alpha_i \qquad i \notin \{j_0, j_1, \ldots, j_p\} \tag{4.129}$$

$$\alpha_{j_0}^{(p)} = \alpha_{j_0} + 1 \tag{4.130}$$

$$\alpha_{j_l}^{(p)} = \alpha_{j_l} \qquad l = 1, 2, \ldots, p-1 \tag{4.131}$$

$$\alpha_{j_p}^{(p)} = \alpha_p - 1 \tag{4.132}$$

Thus, the only cases where $\alpha_i^{(p)}$ is different from α_i are $i = j_0$ and $i = j_p$. As $j_0 \in L$ and $j_p = k \in I$, we have $\alpha_{j_0} \leq s_i - 2$ and $\alpha_{j_p} = s_{j_p}$, and therefore, $\alpha_{j_0}^{(p)} \leq s_i - 1$ and $\alpha_{j_p}^{(p)} = s_i - 1$. This means that the number of indices i for which $\alpha_i^{(p)} = s_i$ is one less than the number of indices i for which $\alpha_i = s_i$. Notice that $\boldsymbol{\alpha}^{(p)} = (\alpha_1^{(p)}, \ldots, \alpha_m^{(p)})$ is the profile according to which $\mathtt{Q}^{(p)}$ is chosen. As we assumed that I has minimal size, that is among all the profiles whose corresponding submatrix of $\mathtt{P}$ is non-singular, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ is the one with minimum number of indices i for which $\alpha_i = s_i$, the matrix $\mathtt{Q}^{(p)}$ must be singular. We demonstrate a contradiction by proving in Lemma 4.15 that all matrices $\mathtt{Q}^{(0)}, \mathtt{Q}^{(1)}, \ldots, \mathtt{Q}^{(p)}$ are non-singular.

□

Lemma 4.13. *For every $k \in K \setminus L$, there exists a sequence of distinct indices*

$$j_0, j_1, \ldots, j_p$$

with $j_p = k$, such that $j_t \in C^t$, and for every $t < p$, there exists a row $\mathbf{r}^T$ of $\mathtt{R}_{j_t}$ which can replace a row of $\mathtt{Q}_{j_{t+1}}$ in $\mathtt{Q} \in \mathbb{R}^{r\times r}$, such that the resulting submatrix $\mathtt{Q}' \in \mathbb{R}^{r\times r}$ is non-singular.

Proof. As $k \in K = L^{t*}$ and $k \notin L = L^0$, there must exist a $p \geq 1$ such that $k \in L^p$ and $k \notin L^{p-1}$. Therefore,

$$k \in L^p \setminus L^{p-1} = C^p. \tag{4.133}$$

From (4.113) we have $L^p = L^{p-1} \cup \mathcal{S}(L^{p-1})$, and as $k \notin L^{p-1}$, we conclude that $k \in \mathcal{S}(L^{p-1})$. Considering the definition of $\mathcal{S}(L^{p-1})$, it follows that there exists an index $k' \in L^{p-1}$ such that a row $\mathbf{r}^T$ of $\mathtt{R}_{k'}$ can replace some row of $\mathtt{Q}_k$ in $\mathtt{Q} \in \mathbb{R}^{r\times r}$ (such that the resulting submatrix $\mathtt{Q}' \in \mathbb{R}^{r\times r}$ is non-singular).

Now, two situations might happen. The first case is when we have $k' \in L = L^0$. In this case, from $k' \in L^0$ and the fact that some row of $\mathtt{R}_{k'}$ can replace some row of $\mathtt{Q}_k$ in $\mathtt{Q}$ (resulting in a non-singular matrix) we get $k \in \mathcal{S}(L^0) \subseteq L^1$. Thus, $k \in L^1$. Adding the fact that, by the Lemma's assumption, we have $k \notin L = L^0$, it follows that $p = 1$. The required sequence would be $j_0, j_1 = k', k$. This sequence has all the properties required in the lemma. Notice that $j_0 = k' \in L^0 = C^0$.

If $k' \notin L$, then notice that $k' \in L^{p-1} \subseteq K$, and therefore, $k' \in K \setminus L$. Thus, the same argument as for k can be applied to k'. By recursively applying this argument (by induction) we can prove the existence of the sequence $j_0, j_1, \ldots, j_p$ with $j_p = k$ and $j_{p-1} = k'$, which possesses the properties required in the lemma. Notice that j_t-s are distinct as $j_t \in C^t$. □

Lemma 4.14. *The matrices* $\mathtt{R}_{(t)}$ *and* $\mathtt{Q}_{(t)}$ *are well-defined, and* $\mathtt{R}_{j_t}$ *and* $\mathtt{Q}_{j_{t+1}}$ *are respectively contained in* $\mathtt{R}_{(t)}$ *and* $\mathtt{Q}_{(t)}$ *for* $t = 0, 1, \ldots, p-1$.

By $\mathtt{Q}_{j_{t+1}}$ being contained in $\mathtt{Q}_{(t)}$ we mean that all rows of $\mathtt{Q}_{j_{t+1}}$ are among the rows of $\mathtt{Q}_{(t)}$.

Proof. We prove a more general statement from which the claim of the lemma follows as a consequence:

(S1) The two matrices $\mathtt{R}_{(t)}$ and $\mathtt{Q}_{(t)}$ are well-defined, and further, $\mathtt{R}_{j_t}, \mathtt{R}_{j_{t+1}}, \ldots, \mathtt{R}_{j_{p-1}}$ are all contained in $\mathtt{R}_{(t)}$ and $\mathtt{Q}_{j_{t+1}}, \mathtt{Q}_{j_{t+2}}, \ldots, \mathtt{Q}_{j_p}$ are all contained in $\mathtt{Q}_{(t)}$.

The proof is done by induction. For $t = 0$ we know that $\mathtt{R}_{j_0}, \mathtt{R}_{j_1}, \ldots, \mathtt{R}_{j_{p-1}}$ are all contained in $\mathtt{R}_{(0)} = \mathtt{R}$, and $\mathtt{Q}_{j_1}, \mathtt{Q}_{j_2}, \ldots, \mathtt{Q}_{j_p}$ are all contained in $\mathtt{Q}_{(0)} = \mathtt{Q}$. This is due to the fact that for all i the matrices $\mathtt{R}_i$ and $\mathtt{Q}_i$ are respectively contained in $\mathtt{R}$ and $\mathtt{Q}$.

Now, assume that (S1) is true for $t < p-1$. We show that it is true for $t+1$. Remember that $\mathtt{R}_{(t+1)}$ and $\mathtt{Q}_{(t+1)}$ were made by taking the row $\mathbf{r}_{j_t}^T$ from $\mathtt{R}_{j_t}$ in $\mathtt{R}_{(t)}$ and the row $\mathbf{q}_{j_{t+1}}^T$ from $\mathtt{Q}_{j_{t+1}}$ in $\mathtt{Q}_{(t)}$, and swapping their places. According to (S1), $\mathtt{R}_{j_t}$ is contained in $\mathtt{R}_{(t)}$ and $\mathtt{Q}_{j_{t+1}}$ is contained in $\mathtt{Q}_{(t)}$, and therefore, this *swapping* is possible. Hence, $\mathtt{R}_{(t+1)}$ and $\mathtt{Q}_{(t+1)}$ are both well-defined.

As, by (S1), the matrices $\mathtt{R}_{j_t}, \mathtt{R}_{j_{t+1}}, \ldots, \mathtt{R}_{j_{p-1}}$ are all contained in $\mathtt{R}_{(t)}$, and the only change in the transition between $\mathtt{R}_{(t)}$ and $\mathtt{R}_{(t+1)}$ is that a row of $\mathtt{R}_{j_t}$ in $\mathtt{R}_{(t)}$ has been replaced, all the matrices $\mathtt{R}_{j_{t+1}}, \ldots, \mathtt{R}_{j_{p-1}}$ are still contained in $\mathtt{R}_{(t+1)}$. Similarly, as $\mathtt{Q}_{j_{t+1}}, \mathtt{Q}_{j_{t+2}}, \ldots, \mathtt{Q}_{j_p}$ are contained in $\mathtt{Q}_{(t)}$ and the matrix $\mathtt{Q}_{(t+1)}$ is obtained by only replacing a certain row of $\mathtt{Q}_{j_{t+1}}$ in $\mathtt{Q}_{(t)}$, the matrices $\mathtt{Q}_{j_{t+2}}, \ldots, \mathtt{Q}_{j_p}$ are still contained in $\mathtt{Q}_{(t+1)}$. □

Lemma 4.15. $\mathtt{Q}^{(t)}$ *is non-singular for all* $t = 0, 1, \ldots, p$.

Proof. First we prove the following

$$\mathcal{R}(\mathtt{Q}_{(t)}^{L^t}) = \mathcal{R}(\mathtt{Q}^{L^t}). \tag{4.134}$$

where $\mathcal{R}$ gives the row space of a matrix, and for any subset Γ of views we have $\mathtt{Q}_{(t)}^{\Gamma} = \text{stack}(\{\mathtt{Q}_{(t),i}\}_{i\in\Gamma})$ with $\mathtt{Q}_{(t),i}$ is the submatrix of $\mathtt{P}_i$ created by the rows of $\mathtt{P}_i$ chosen for making $\mathtt{Q}_{(t)}$. We prove the above by induction.

First notice that as $\mathtt{Q}_{(0)} = \mathtt{Q}$, we have $\mathcal{R}(\mathtt{Q}_{(0)}^{L^0}) = \mathcal{R}(\mathtt{Q}^{L^0})$. Now, assume that (4.134) holds for some t, we will show that it is true for $t+1$. We prove this by looking at the intermediate matrix $\mathtt{Q}_{(t)}^{L^{t+1}}$, first showing $\mathcal{R}(\mathtt{Q}_{(t)}^{L^{t+1}}) = \mathcal{R}(\mathtt{Q}^{L^{t+1}})$, and then showing $\mathcal{R}(\mathtt{Q}_{(t+1)}^{L^{t+1}}) = \mathcal{R}(\mathtt{Q}_{(t)}^{L^{t+1}})$. Observe that, as $\{L^t, C^{t+1}\}$ is a partition of L^{t+1}, we have $\mathtt{Q}_{(t)}^{L^{t+1}} \equiv \text{stack}(\mathtt{Q}_{(t)}^{L^t}, \mathtt{Q}_{(t)}^{C^{t+1}})$. Therefore,

$$\mathcal{R}(\mathtt{Q}_{(t)}^{L^{t+1}}) = \mathcal{R}(\mathtt{Q}_{(t)}^{L^t}) + \mathcal{R}(\mathtt{Q}_{(t)}^{C^{t+1}}). \tag{4.135}$$

As we have assumed (4.134) is true for t, we get

$$\mathcal{R}(\mathtt{Q}_{(t)}^{L^{t+1}}) = \mathcal{R}(\mathtt{Q}^{L^t}) + \mathcal{R}(\mathtt{Q}_{(t)}^{C^{t+1}}). \tag{4.136}$$

Now, from Lemma 4.16 we have $\mathtt{Q}_{(t)}^{C^{t+1}} = \mathtt{Q}^{C^{t+1}}$

$$\begin{aligned}\mathcal{R}(\mathtt{Q}_{(t)}^{L^{t+1}}) &= \mathcal{R}(\mathtt{Q}^{L^t}) + \mathcal{R}(\mathtt{Q}^{C^{t+1}}) \\ &= \mathcal{R}(\text{stack}(\mathtt{Q}^{L^t}, \mathtt{Q}^{C^{t+1}})) \\ &= \mathcal{R}(\mathtt{Q}^{L^{t+1}}).\end{aligned} \tag{4.137}$$

Now, we are done if prove $\mathcal{R}(\mathtt{Q}_{(t+1)}^{L^{t+1}}) = \mathcal{R}(\mathtt{Q}_{(t)}^{L^{t+1}})$. First, notice that $\mathtt{Q}_{(t+1)}$ is made by taking $\mathtt{Q}_{(t)}$ and replacing the row $\mathbf{q}_{j_{t+1}}^T$ of $\mathtt{Q}_{j_{t+1}}$ in $\mathtt{Q}_{(t)}$ with $\mathbf{r}_{j_t}^T$. From (4.124) we have $\mathbf{r}_{j_t}^T \in \mathcal{R}(\mathtt{Q}^{L^{t+1}})$, which using (4.137) gives

$$\mathbf{r}_{j_t}^T \in \mathcal{R}(\mathtt{Q}_{(t)}^{L^{t+1}}). \tag{4.138}$$

Notice that, as $j_{t+1} \in C^{t+1} \subseteq L^{t+1}$, the matrix $\mathtt{Q}_{j_{t+1}}$ in contained in $\mathtt{Q}_{(t)}^{L^{t+1}}$. Therefore, $\mathtt{Q}_{(t+1)}^{L^{t+1}}$ is made by replacing some row of $\mathtt{Q}_{(t)}^{L^{t+1}}$ with $\mathbf{r}_{j_t}^T$. This together with (4.138) gives

$$\mathcal{R}(\mathtt{Q}_{(t+1)}^{L^{t+1}}) \subseteq \mathcal{R}(\mathtt{Q}_{(t)}^{L^{t+1}}). \tag{4.139}$$

Now, observe that, as $j_{t+1} \in C^{t+1}$, the matrix $\mathtt{Q}_{j_{t+1}}$ (and therefore its row $\mathbf{q}_{j_{t+1}}^T$) in contained is $\mathtt{Q}_{(t)}^{C^{t+1}}$. From Lemma 4.16 we have $\mathtt{Q}_{(t)}^{C^{t+1}} = \mathtt{Q}^{C^{t+1}}$. Therefore, $\mathtt{Q}_{(t+1)}^{L^{t+1}}$ is made by taking $\mathtt{Q}_{(t)}^{L^{t+1}} \equiv \text{stack}(\mathtt{Q}_{(t)}^{L^t}, \mathtt{Q}_{(t)}^{C^{t+1}}) = \text{stack}(\mathtt{Q}_{(t)}^{L^t}, \mathtt{Q}^{C^{t+1}})$ and replacing the row $\mathbf{q}_{j_{t+1}}^T$ in $\mathtt{Q}^{C^{t+1}}$ with $\mathbf{r}_{j_t}^T$. Let $\mathtt{M}$ be the matrix obtained by replacing $\mathbf{r}_{j_t}^T$ with $\mathbf{q}_{j_{t+1}}^T$ in $\mathtt{Q}^{C^{t+1}}$.

Therefore, we have

$$\mathtt{Q}_{(t)}^{L^{t+1}} \equiv \text{stack}(\mathtt{Q}_{(t)}^{L^t}, \mathtt{Q}^{C^{t+1}}) \tag{4.140}$$

$$\mathtt{Q}_{(t+1)}^{L^{t+1}} \equiv \text{stack}(\mathtt{Q}_{(t)}^{L^t}, \mathtt{M}) \tag{4.141}$$

Thus, we can say

$$\mathcal{R}(\mathtt{Q}_{(t+1)}^{L^{t+1}}) = \mathcal{R}(\mathtt{Q}_{(t)}^{L^t}) + \mathcal{R}(\mathtt{M}) \tag{4.142}$$

Using the induction hypothesis (4.134), the above gives

$$\begin{aligned}\mathcal{R}(\mathtt{Q}_{(t+1)}^{L^{t+1}}) &= \mathcal{R}(\mathtt{Q}^{L^t}) + \mathcal{R}(\mathtt{M}) \\ &= \mathcal{R}(\text{stack}(\mathtt{Q}^{L^t}, \mathtt{M}))\end{aligned} \tag{4.143}$$

The matrix $\text{stack}(\mathtt{Q}^{L^t}, \mathtt{M})$ is created by taking $\text{stack}(\mathtt{Q}^{L^t}, \mathtt{Q}^{C^{t+1}}) \equiv \mathtt{Q}^{L^{t+1}}$, and replacing the row $\mathbf{q}_{j_{t+1}}^T$ in $\mathtt{Q}^{C^{t+1}}$ by $\mathbf{r}_{j_t}^T$. By, the definition of $\mathbf{q}_{j_{t+1}}^T$ and $\mathbf{r}_{j_t}^T$, replacing $\mathbf{q}_{j_{t+1}}^T$ with $\mathbf{r}_{j_t}^T$ in $\mathtt{Q} \equiv \text{stack}(\mathtt{Q}^{L^t}, \mathtt{Q}^{C^{t+1}}, \mathtt{Q}^{\bar{L}^t})$ results in a non-singular matrix $\mathtt{Q}' \equiv \text{stack}(\mathtt{Q}^{L^t}, \mathtt{M}, \mathtt{Q}^{\bar{L}^t})$. This suggests that $\text{stack}(\mathtt{Q}^{L^t}, \mathtt{M})$ has full row rank. Using (4.143), it follows that $\mathtt{Q}_{(t+1)}^{L^{t+1}}$ has also full row rank. This together with (4.139) imply

$$\mathcal{R}(\mathtt{Q}_{(t+1)}^{L^{t+1}}) = \mathcal{R}(\mathtt{Q}_{(t)}^{L^{t+1}}). \tag{4.144}$$

Using (4.137) we conclude

$$\mathcal{R}(\mathtt{Q}_{(t+1)}^{L^{t+1}}) = \mathcal{R}(\mathtt{Q}^{L^{t+1}}). \tag{4.145}$$

This completes our inductive proof of (4.134), that is $\mathcal{R}(\mathtt{Q}_{(t)}^{L^t}) = \mathcal{R}(\mathtt{Q}^{L^t})$ for all t. The rest of the proof is simple. Notice that $\mathtt{Q}_{(t)} \equiv \text{stack}(\mathtt{Q}_{(t)}^{L^t}, \mathtt{Q}_{(t)}^{\bar{L}^t})$, and also, by Lemma 4.16, $\mathtt{Q}_{(t)}^{\bar{L}^t} = \mathtt{Q}^{\bar{L}^t}$. Therefore, we have

$$\begin{aligned}\mathcal{R}(\mathtt{Q}_{(t)}) &= \mathcal{R}(\text{stack}(\mathtt{Q}_{(t)}^{L^t}, \mathtt{Q}_{(t)}^{\bar{L}^t})) \\ &= \mathcal{R}(\text{stack}(\mathtt{Q}_{(t)}^{L^t}, \mathtt{Q}^{\bar{L}^t})) \\ &= \mathcal{R}(\mathtt{Q}_{(t)}^{L^t}) + \mathcal{R}(\mathtt{Q}^{\bar{L}^t}) \\ &= \mathcal{R}(\mathtt{Q}^{L^t}) + \mathcal{R}(\mathtt{Q}^{\bar{L}^t}) \\ &= \mathcal{R}(\text{stack}(\mathtt{Q}^{L^t}, \mathtt{Q}^{\bar{L}^t})) \\ &= \mathcal{R}(\mathtt{Q}).\end{aligned} \tag{4.146}$$

As $\mathtt{Q}$ is non-singular, it follows that $\mathtt{Q}_{(t)}$ has full rank for all $t = 0, 1, \ldots, p$. □

Lemma 4.16. *The following hold*

$$\mathtt{Q}_{(t)}^{\bar{L}^t} = \mathtt{Q}^{\bar{L}^t}, \qquad \text{for } t = 0,1,\ldots,p, \tag{4.147}$$

$$\mathtt{Q}_{(t)}^{C^{t+1}} = \mathtt{Q}^{C^{t+1}}, \qquad \text{for } t = 0,1,\ldots,p-1, \tag{4.148}$$

where $\bar{L}^t = \{1,2,\ldots,m\} \setminus L^t$ *is the complement of* L^t.

Proof. During the transition $\mathtt{Q} = \mathtt{Q}_{(0)} \to \mathtt{Q}_{(1)} \to \ldots \to \mathtt{Q}_{(t)}$, only the matrices $\mathtt{R}_{j_0}, \mathtt{R}_{j_1}, \ldots, \mathtt{R}_{j_{t-1}}$ and $\mathtt{Q}_{j_1}, \mathtt{Q}_{j_2}, \ldots, \mathtt{Q}_{j_t}$ are involved in terms of exchanging rows. Therefore, for an index $i \notin \{j_0, j_1, \ldots, j_t\}$, if $\mathtt{Q}_i$ is contained in $\mathtt{Q}_{(0)} = \mathtt{Q}$, then $\mathtt{Q}_i$ will be still present in $\mathtt{Q}_{(t)}$ and also $\mathtt{R}_i$ is contained in $\mathtt{R}_{(t)}$, which means that no row of $\mathtt{R}_i$ is contained in $\mathtt{Q}_{(t)}$. In other words, $\mathtt{Q}_{(t),i} = \mathtt{Q}_i$ where $\mathtt{Q}_{(t),i}$ is the submatrix of $\mathtt{P}_i$ whose rows are present in $\mathtt{Q}_{(t)}$. As for all $t' \leq t$ we have $j_{t'} \in C^{t'} \subseteq L^{t'} \subseteq L^t$, it follows that $j_{t'} \notin \bar{L}^t$ for all $t' = 0,1,\ldots,t$. This means that

$$\mathtt{Q}_{(t)}^{\bar{L}^t} = \operatorname{stack}(\{\mathtt{Q}_{(t),i}\}_{i \in \bar{L}^t}) = \operatorname{stack}(\{\mathtt{Q}_i\}_{i \in \bar{L}^t}) = \mathtt{Q}^{\bar{L}^t} \tag{4.149}$$

Finally, (4.148) immediately follows as $C^{t+1} = L^{t+1} \setminus L^t \subseteq \bar{L}^t$. □

4.6 Summary

We developed the theory of projective reconstruction for projections from an arbitrary dimensional space. Theorems were presented which derived projective reconstruction from the projection equations. We also classified the wrong solutions to the projective factorization problem where not all the estimated projective depths are constrained to be nonzero.

Chapter 5

Applications

In this chapter we present examples showing how the reconstruction of certain types of dynamic scenes can be modeled as projections from higher dimensional spaces.

5.1 Motion Segmentation

Assume that we have a number of rigid objects in the scene that move with respect to each other. In a very simple scenario one could consider a rigid object moving with respect to a static background. We take 2D images of the scene at different times. The problem of motion segmentation is to find the rigid bodies and classify them according to their motion.

The input to the motion segmentation problem is complete or partial tracks of 2D image points for different views. The task of motion segmentation is to segment the point tracks according to their associated rigid body and find the camera matrix (or matrices), the motions, and the location of the 3D points. We start our analysis with the simpler case of affine cameras and show how the motion segmentation in this case is related to the problem of subspace segmentation. We then turn to the more complex case of projective cameras.

5.1.1 Affine Cameras

In affine camera model the projected 2D points are related to the 3D points through an affine transformation. This can be shown by

$$\mathbf{x} = \mathtt{P}\mathbf{X}, \tag{5.1}$$

where $\mathbf{X} = [X_1, X_2, X_3, 1]^T \in \mathbb{R}^4$ represent a 3D scene point in homogeneous coordinates, $\mathbf{x} = [x_1, x_2]^T \in \mathbb{R}^2$ represent the 2D image point and $\mathtt{P} \in \mathbb{R}^{2\times 4}$ is the affine camera matrix. Affine cameras are usually used as an approximation of perspective camera when the scene objects are relatively far away from the camera.

Now, assume that there are n points $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ in the scene, all moving according to a global rigid motion. We have 2D images of the points in n different frames. Let $\mathtt{Q}_i$ be the rigid motion matrix representing the motion of the points in the

i-th frame. This matrix has the form

$$\mathtt{Q}_i = \begin{bmatrix} \mathtt{R}_i & \mathbf{t}_i \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{5.2}$$

where $\mathtt{R}_i$ and $\mathbf{t}_i$ respectively represent the rotation and translation of the points in the i-th frame. The location of the j-th 3D point in the i-th frame can be represented as

$$\mathbf{X}_{ij} = \mathtt{Q}_i \mathbf{X}_j, \tag{5.3}$$

that is the motion matrix $\mathtt{Q}_i$ applied to the scene point $\mathbf{X}_j$. Now, assume that the scene points at every frame i is seen by an affine camera with the camera matrix $\mathtt{P}_i$. Then we have

$$\mathbf{x}_{ij} = \mathtt{P}_i \mathbf{X}_{ij} = \mathtt{P}_i \mathtt{Q}_i \mathbf{X}_j. \tag{5.4}$$

Notice that if all the images are captured with the same camera whose parameters are fixed among different frames, then we can drop the index i from $\mathtt{P}_i$. But, for now, we consider the general case. If the 2D image points $\mathbf{x}_{ij}$ are arranged in a $2m \times n$ matrix $[\mathbf{x}_{ij}]$, then from (5.4) we have

$$[\mathbf{x}_{ij}] = \begin{bmatrix} \mathbf{x}_{11} & \mathbf{x}_{12} & \cdots & \mathbf{x}_{1n} \\ \mathbf{x}_{21} & \mathbf{x}_{22} & \cdots & \mathbf{x}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{m1} & \mathbf{x}_{m2} & \cdots & \mathbf{x}_{mn} \end{bmatrix} = \begin{bmatrix} \mathtt{P}_1\mathtt{Q}_1 \\ \mathtt{P}_2\mathtt{Q}_2 \\ \vdots \\ \mathtt{P}_m\mathtt{Q}_m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix} = \mathtt{M}\mathtt{X} \tag{5.5}$$

where $\mathtt{M} = \text{stack}(\mathtt{P}_1\mathtt{Q}_1, \mathtt{P}_2\mathtt{Q}_2, \ldots, \mathtt{P}_m\mathtt{Q}_m) \in \mathbb{R}^{2m \times 4}$ and $\mathtt{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n] \in \mathbb{R}^{4 \times n}$. The above says that $[\mathbf{x}_{ij}]$ can be factorized as the multiplication of a $2m \times 4$ by a $4 \times n$ matrix. This means that the columns of $[\mathbf{x}_{ij}]$ (the point tracks) lie on a linear subspace of dimension 4 or less. As (5.5) suggests, this subspace is generally equal to the column space of $\mathtt{M}$. For general motions, the column space of $\mathtt{M}$ is four-dimensional. However, the dimension can be lower for special cases (see [Vidal et al., 2008] for a brief discussion).

Now, consider the case where the points $\{\mathbf{X}_j\}$ belong to p different rigid bodies, each undergoing a potentially different rigid motion. The motions are represented by

$$\mathtt{Q}_i^k = \begin{bmatrix} \mathtt{R}_i^k & \mathbf{t}_i^k \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{5.6}$$

where $\mathtt{Q}_i^k$ represents the motion of the k-th body in the i-th frame. Let $c_j \in \{1, 2, \ldots, p\}$ be the class of the j-th scene point, that is the rigid body to which $\mathbf{X}_j$ belongs. Thus, the location of the j-th scene point at frame i can be represented by $\mathbf{X}_{ij} = \mathtt{Q}_i^{c_j} \mathbf{X}_j$. Let, $\mathbf{X}^k = [\cdots \mathbf{X}_j \cdots]_{c_j = k} \in \mathbb{R}^{4 \times n_k}$ be the horizontal concatenation of the scene points belonging to the k-th rigid body, and $[\mathbf{x}_{ij}]_{c_j=k} \in \mathbb{R}^{2m \times n_k}$ be the arrange-

ment of the image points belonging to the k-th rigid body in a $2m \times n_k$ matrix, where n_k is the number of the points of the k-th body. As each body moves rigidly, from (5.5) for the k-th rigid body one can write

$$[\mathbf{x}_{ij}]_{c_j=k} = \begin{pmatrix} \mathtt{P}_1\mathtt{Q}_1^k \\ \mathtt{P}_2\mathtt{Q}_2^k \\ \vdots \\ \mathtt{P}_m\mathtt{Q}_m^k \end{pmatrix} \left[\cdots \mathbf{X}_j \cdots\right]_{c_j=k} = \mathtt{M}^k\mathtt{X}^k, \tag{5.7}$$

where $\mathtt{M}^k = \text{stack}(\mathtt{P}_1\mathtt{Q}_1^k, \mathtt{P}_2\mathtt{Q}_2^k, \cdots, \mathtt{P}_m\mathtt{Q}_m^k)$. Therefore, the image point tracks of the k-th rigid body (the columns of $[\mathbf{x}_{ij}]_{c_j=k}$) belong to a four (or less) dimensional linear subspace, which is generally spanned by the columns of $\mathtt{M}^k$. Now, consider the whole set of image points $[\mathbf{x}_{ij}]$. From the above discussion we can say that the j-th column of $[\mathbf{x}_{ij}]$ lies on the column space of $\mathtt{M}^{c_j}$. Therefore, the columns of $[\mathbf{x}_{ij}]$ lie on a union of p subspaces. Each subspace is of dimension four or less, and corresponds to one of the rigid bodies. By clustering the points according to their corresponding subspaces we can find out which point belongs to which rigid body. Hence, we require methods that, given a bunch of points lying on a mixture of subspaces, can segment them according to their associated subspaces. These methods are knows as *subspaces clustering* or *subspaces segmentation* techniques. In the next section, we describe this problem, and review some of the subspace clustering techniques.

After segmenting the point tracks, the points belonging to each rigid body can be dealt with separately as a rigid scene reconstruction problem with affine cameras. We then use the fact that the camera matrix is the same in each frame for all rigid bodies to obtain consistency between the reconstruction of the scene points (and motions) belonging to different rigid bodies. One can further reduce the ambiguities, for example when the camera matrix is known to be fixed among all frames.

5.1.2 Subspace Clustering

Subspace clustering is an important problem in data analysis with applications in many different areas in computer vision including motion segmentation [Vidal et al., 2008; Kanatani, 2001; Costeira and Kanade, 1998; Zelnik-Manor and Irani, 2003], video shot segmentation [Lu and Vidal, 2006], illumination invariant clustering [Ho et al., 2003], image segmentation [Yang et al., 2008] and image representation and compression [Hong et al., 2005].

Subspace clustering deals with the case where the set of data points $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n \in \mathbb{R}^d$ lie on a union of different subspaces. The task is to label the points according to their corresponding subspace and give a basis for each subspace. In some cases the number or dimensions of subspaces is unknown and the algorithm is supposed to find them as well. For most applications the dimension of each subspace is much smaller than the dimension of the ambient space $\mathbb{R}^d$.

Many different methods have been proposed to cluster the data into multiple subspaces. Here, we briefly describe some of the major subspace clustering algorithms.

For a thorough survey on this topic we refer the reader to [Vidal, 2011]. The reader may safely skip the rest of this subsection and move forward to Sect. 5.1.3.

Matrix Factorization Consider a set of points $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ belonging to a mixture of subspaces. Matrix factorization approaches try to find the subspaces from some factorization of the data matrix $\mathtt{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$. A well-known example is the work of Costeira and Kanade [1998] where the segmentation is obtained from the SVD of the data matrix. Particularly, if the subspaces are *independent*, for $\mathtt{A} = \mathtt{U}\Sigma\mathtt{V}^T$ being the skinny SVD of the matrix $\mathtt{A}$, the matrix $\mathtt{Q} = \mathtt{V}\mathtt{V}^T$ is such that $\mathtt{Q}_{ij} = 0$ if $\mathbf{a}_i$ and $\mathbf{a}_j$ belong to different subspaces [Vidal et al., 2008; Kanatani, 2001].

Generalized PCA (GPCA) In GPCA [Vidal et al., 2005] each linear (resp. affine) subspace is modeled as the null space of a linear (resp. affine) transformation. Here, for simplicity we consider the case where all subspaces are hyperplanes, that is to say, heir dimension is the dimension of the ambient space less 1. The i-th subspace can be represented as the set of points satisfying $\mathbf{v}_i^T\mathbf{a} - t_i = 0$. Therefore, a point lying on the mixture of these subspaces will satisfy the polynomial equation:

$$\mathcal{P}(\mathbf{a}) = \prod_{i=1}^{l}(\mathbf{v}_i^T\mathbf{a} - t_i) = 0 \tag{5.8}$$

where l is the number of subspaces. If l is known, we can find the polynomial parameters by fitting a degree l polynomial to the data. Now, if a point $\mathbf{a}$ belongs to the k-th subspace, then it is easy to check that the gradient of $\mathcal{P}$ at $\mathbf{a}$ is equal to $\mathbf{v}_i$ up to scale, that is the normal vector to the k-th subspace. This gives a way to cluster the data points $\mathbf{a}_i$ to different subspaces.

In practical applications where data is noisy, for two points on one subspaces the derivatives of p are not exactly equal. Thus, a follow-up clustering should be performed after calculating the derivatives. A common approach is to form a similarity matrix for each pair of derivatives and segment the data using spectral clustering. GPCA can be extended to deal with subspaces of arbitrary dimension. For more details see Vidal et al. [2005].

K-subspaces The basic idea behind such methods is to iterate between point segmentation and subspace estimation [Bradley and Mangasarian, 2000; Tseng, 2000; Agarwal and Mustafa, 2004]. Assuming the labels of the points are known each subspace can be easily estimated using simple methods like PCA. On the other hand, if the subspaces are known labels can be estimated according to their distance to the subspaces. The algorithms simply iterate between these two stages. This is similar to the *k-means* algorithm adapted for clustering subspaces. These approaches are usually used as a post processing stage, as they require a good initial solution.

Mixture of Probabilistic PCA (MPPCA) The MPPCA method [Tipping and Bishop, 1999] can be thought of as a probabilistic version of K-subspaces. Data is assumed to

be normally distributed in each subspace and is also contaminated with a Gaussian noise. These leads to a mixture of Gaussians model which is usually solved using the Expectation Maximization (EM) approach or its variants.

Agglomerative Lossy Compression (ALC) The ALC Ma et al. [2007] takes an information theoretic approach. It defines a measure of the information (number of bits) required to optimally code the data belonging to a mixture of subspaces allowing a distortion of ϵ (to account for noise). This measure is actually a trade-off between the number of bits required to encode the data in each subspace and the number of bits needed to represent the membership of each point in its corresponding subspace. An approximate incremental method is applied to minimize the target function.

Random Sample Consensus (RANSAC) The RANSAC [Fischler and Bolles, 1981] is originally designed for fitting a model to a collection of data where a rather small proportion of the data are outliers. At each iteration it selects k points at random where k is usually the minimum number of data for fitting the model. Using these k points it estimates a model. Then it classifies all the other points as inliers/outliers based on their proximity to the model. The algorithm stops when a good number of inliers are obtained. For subspace clustering RANSAC can be used to extract one subspace at a time. In this case, one hopes that RANSAC chooses k points from a common subspace at some stage and obtains the points belonging to that subspace as inliers. However, using the basic RANSAC for subspace clustering can be impractical in many cases.

Sparse Subspace Clustering Sparse Subspace Clustering (SSC) proposed by Elhamifar and Vidal [2009] is one of the state-of-the-art methods of subspace segmentation with major advantages over the previous methods (see Vidal [2011]). In SSC the subspace clustering is done based on the neighbourhood graph obtained by the l^1-norm sparse representation of each point by the other points. The basic SSC method works as follows:

Consider a set of points $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ in $\mathbb{R}^D$, sampled from a mixture of different subspaces such that no point lies on the origin. Each $\mathbf{a}_i$ can be obtained as a linear combination of the others:

$$\mathbf{a}_i = \sum_j c_j \mathbf{a}_j = \mathtt{A}\mathbf{c}, \quad \text{where } c_i = 0, \tag{5.9}$$

where $\mathtt{A}$ is the matrix $[\mathbf{a}_1 \mathbf{a}_2 \cdots \mathbf{a}_n]$ and $\mathbf{c} = [c_1 c_2 \cdots c_n]^T$.

Of course, this combination (if it exists) is not unique in general. In SSC we are interested in a combination with smallest l^1-norm of the corresponding combination coefficient $\mathbf{c}$. This means that for each $\mathbf{a}_i$ the following is solved:

$$\min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{s.t.} \quad \mathbf{a}_i = \mathtt{A}\,\mathbf{c}, \ c_i = 0. \tag{5.10}$$

Usually, the optimal $\mathbf{c}$ has many zero entries. The corresponding points of the nonzero elements of the optimal $\mathbf{c}$ are set to be the neighbours of $\mathbf{a}_i$. Doing the same thing for every point forms a directed neighbourhood graph on the set of points.

In Elhamifar and Vidal [2009] it has been proved that if the subspaces are independent, then the neighbours of each point would be in the same subspace. This means that there is no link between the graphs of two different subspaces. Based on this fact, a subspace segmentation method is proposed by finding the connected components of the neighbourhood graph. However, in practice, where the noise is present, this is done by spectral clustering.

Errors and outliers To deal with errors the above optimization problem is slightly changed:

$$\min_{\mathbf{c},\mathbf{e}} \|\mathbf{c}\|_1 + \frac{\alpha}{2}\|\mathbf{e}\|_2 \quad \text{s.t.} \quad \mathbf{a}_i = \mathtt{A}\,\mathbf{c} + \mathbf{e}, \;\; c_i = 0. \tag{5.11}$$

As you can see, each $\mathbf{a}_i$ is represented as a combination of the other points plus some error. This model is not optimal as all elements of $\mathbf{e}$ are equally weighted. This is while the error vector $\mathbf{e}$ here is dependent on the combination vector $\mathbf{c}$.

To deal with outliers as well, the following optimization problem has been proposed:

$$\min_{\mathbf{c},\mathbf{e}} \|\mathbf{c}\|_1 + \lambda\|\mathbf{g}\|_1 + \frac{\alpha}{2}\|\mathbf{e}\|_2 \quad \text{s.t.} \quad \mathbf{a}_i = \mathtt{A}\,\mathbf{c} + \mathbf{g} + \mathbf{e}, \;\; c_i = 0. \tag{5.12}$$

The above assumes that the vector of outliers $\mathbf{g}$ is sparse for each $\mathbf{a}_i$.

Low-Rank Subspace Clustering Before describing this method, let us rewrite (5.10) in matrix form:

$$\min_{\mathtt{C}} \|\mathtt{C}\|_1 \quad \text{s.t.} \quad \mathtt{A} = \mathtt{A}\,\mathtt{C}, \;\; \operatorname{diag}(\mathtt{C}) = \mathbf{0}. \tag{5.13}$$

In the above $\mathtt{C} \in \mathbb{R}^{n\times n}$ is the matrix of combination coefficients, $\|.\|_1$ is the (entrywise) l^1 matrix norm and $\operatorname{diag}(\mathtt{C})$ gives the vector of diagonals of a matrix.

In low-rank subspace clustering [Liu et al., 2010b], instead of seeking sparsity, one tries to minimize the rank of the combination matrix $\mathtt{C}$. To make the problem tractable the trace norm is minimized instead of rank:

$$\min_{\mathtt{C}} \|\mathtt{C}\|_* \quad \text{s.t.} \quad \mathtt{A} = \mathtt{A}\,\mathtt{C}, \tag{5.14}$$

where $\|.\|_*$ represents the trace norm, that is the sum of the singular values of the matrix. Liu et al. [2010b] prove that if subspaces are independent, then for the optimal coefficient matrix $\mathtt{C}$ all the elements c_{ij} would be zero where $\mathbf{a}_i$ and $\mathbf{a}_j$ belong to different subspaces. Therefore, similar to SCC, the clustering can be done by finding the connected components of the corresponding graph of $\mathtt{C}$ (by spectral clustering in practice).

In a later paper Liu et al. [2010a], the authors proved that the above problem has the unique optimal solution of:

$$\mathtt{C}^* = \mathtt{V}_r\mathtt{V}_r^T \tag{5.15}$$

where the n by r matrix $\mathtt{V}_r$ is the matrix of right singular vectors of $\mathtt{A}$, that is $\mathtt{X} = \mathtt{U}_r\Sigma_r\mathtt{V}_r^T$ is the skinny rank r singular value decomposition of $\mathtt{A}$.

Actually solving the noiseless problem (5.14) is of little practical value. Similar to the SSC method, here the following model has been proposed to deal with noise:

$$\text{argmin}_{\mathtt{C},\mathtt{E}} \|\mathtt{C}\|_* + \alpha \|\mathtt{E}\|_{2,1} \quad \text{s.t.} \quad \mathtt{A} = \mathtt{A}\,\mathtt{C} + \mathtt{E}, \tag{5.16}$$

where $\|\mathtt{E}\|_{2,1} = \sum_{i=1}^{n} \|\mathbf{e}_i\|_2$, is the $l^{2,1}$ norm of the matrix $\mathtt{E}$, with $\mathbf{e}_i$ being the i-th column of $\mathtt{E}$. This is actually the l^1 norm of the vector of the l^2 norms of columns of $\mathtt{E}$. It is used to deal with outliers, that is where a small portion of data is contaminated by noise, however, the perturbation is not sparse for each $\mathbf{e}_i$.

A closed form solution Favaro et al. [2011] proposed a method for subspace clustering with noise which has a closed form solution. Here, the data $\mathtt{D}$ is written as $\mathtt{A} + \mathtt{E}$, where $\mathtt{E}$ is the noise and $\mathtt{A}$ is the clean data. In other words, columns of $\mathtt{A}$ are exactly on the union of subspaces. The following optimization problem is solved:

$$\min_{\mathtt{C},\mathtt{A},\mathtt{E}} \|\mathtt{C}\|_* + \alpha \|\mathtt{E}\|_{\mathtt{F}} \quad \text{s.t.} \quad \mathtt{A} = \mathtt{A}\,\mathtt{C}, \quad \mathtt{D} = \mathtt{A} + \mathtt{E} \tag{5.17}$$

or equivalently:

$$\min_{\mathtt{C},\mathtt{A}} \|\mathtt{C}\|_* + \alpha \|\mathtt{D} - \mathtt{A}\|_{\mathtt{F}} \quad \text{s.t.} \quad \mathtt{A} = \mathtt{A}\,\mathtt{C}, \tag{5.18}$$

It turns out that the closed-form solution can be obtained in a much simpler way than what is given in [Favaro et al., 2011]. As mentioned in section 5.1.2, given $\mathtt{A}$, the optimal $\mathtt{C}$ can be achieved as $\mathtt{C}^* = \mathtt{V}_r\mathtt{V}_r^T$ where $\mathtt{V}_r$ is obtained from $\mathtt{A} = \mathtt{U}_r\Sigma_r\mathtt{V}_r^T$, the skinny SVD of $\mathtt{A}$. Let r be the rank of $\mathtt{A}$. For the optimal $\mathtt{C}$ we have $\|\mathtt{C}^*\|_* = \|\mathtt{V}\,\mathtt{V}^T\|_* = r$. The problem, thus, turns to:

$$\min_{r} \min_{\mathtt{A}} r + \alpha \|\mathtt{D} - \mathtt{A}\|_{\mathtt{F}} \quad \text{s.t.} \quad \text{rank}(\mathtt{A}) = r \tag{5.19}$$

It is well known that with a fixed r, the optimal solution for $\mathtt{A}$ is a matrix with the same singular vectors and the same first (biggest) r singular values as $\mathtt{D}$ and the rest of the singular values zero. This means that the matrices Σ_r, $\mathtt{U}_r$ and $\mathtt{V}_r$ introduce above are respectively the matrix of first r singular values, first r left singular vectors and first r right singular vectors of $\mathtt{D}$. Therefore, for each choice of r, the optimal $\mathtt{A}$ can be obtained as $\mathtt{U}_r\Sigma_r\mathtt{V}_r^T$, which is the rank-$r$ SVD thresholding of $\mathtt{D}$. For this choice of $\mathtt{A}$, we have $\|\mathtt{D} - \mathtt{A}\|_{\mathtt{F}} = \sum_{k=r+1}^{n} \sigma_k^2$, where σ_k is the k-th singular value of $\mathtt{D}$. We can do this for all possible values of r and choose the one with the smallest target value. Hence, the optimization problem is

$$\min_{r} r + \alpha \sum_{k=r+1}^{n} \sigma_k^2 = \min_{r} \sum_{k=1}^{r} (1 - \alpha\sigma_k^2), \tag{5.20}$$

where, by convention, $\sum_{k=1}^{0}(.)$ is assumed to be zero. This shows that the optimal r is achieved by thresholding the singular values of $\mathtt{D}$ at $1/\sqrt{\alpha}$.

5.1.3 Projective Cameras

Now, we turn to the more complex case motion segmentation with projective cameras and show that how different cases of the problem can be modeled as projections from higher dimensions.

Again, we consider p rigidly moving bodies. Recall from Sect. 5.1.1 that the rigid motion was represented with the matrix

$$\mathtt{Q}_i^k = \begin{bmatrix} \mathtt{R}_i^k & \mathtt{t}_i^k \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{5.21}$$

where $\mathtt{Q}_i^k$ is the rigid motion matrix corresponding to the k-th body in the i-th frame. We also defined $c_j \in \{1,2,\ldots,p\}$ to be the class of the j-th scene point, meaning that $\mathbf{X}_j$ belongs to the c_j-th rigid body. The location of the j-th scene point at frame i is

$$\mathbf{X}_{ij} = \mathtt{Q}_i^{c_j}\mathbf{X}_j \tag{5.22}$$

Therefore, having projective cameras, the image points are created as follows:

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_{ij} = \mathtt{P}_i\mathtt{Q}_i^{c_j}\mathbf{X}_j, \tag{5.23}$$

where $\mathbf{x}_{ij} \in \mathbb{R}^3$ represents an image point in homogeneous coordinates, $\mathtt{P}_i \in \mathbb{R}^{3\times 4}$ is the camera matrix of the i-th frame and λ_{ij} is the projective depth. In a similar way to the case of affine cameras, for the points $\mathbf{X}_j$ belonging to the k-th rigid body ($c_j = k$) we can write

$$[\lambda_{ij}\mathbf{x}_{ij}]_{c_j=k} = \begin{pmatrix} \mathtt{P}_1\mathtt{Q}_1^k \\ \mathtt{P}_2\mathtt{Q}_2^k \\ \vdots \\ \mathtt{P}_m\mathtt{Q}_m^k \end{pmatrix} \begin{bmatrix} \cdots & \mathbf{X}_j & \cdots \end{bmatrix}_{c_j=k} = \mathtt{M}^k\mathtt{X}^k, \tag{5.24}$$

Therefore, the columns of the matrix $[\lambda_{ij}\mathbf{x}_{ij}]_{c_j=k}$, created by arranging into a matrix the weighted image points $\lambda_{ij}\mathbf{x}_{ij}$ of a single rigid body, lie on a 4 (or less) dimensional subspace. Thus, the columns of the complete matrix of weighted image points $[\lambda_{ij}\mathbf{x}_{ij}]$ lie on a mixture of subspaces. This means that, if we somehow manage to find the projective depths λ_{ij}, motion segmentation can be performed by applying a subspace clustering algorithm on the weighted data matrix $[\lambda_{ij}\mathbf{x}_{ij}]$.

In the next three subsections, we will show that how different forms of relative motions can be modeled as projections from higher dimensional projective spaces. Using such models, the projective depths λ_{ij} can be obtained using projective reconstruction in higher dimensions.

5.1.3.1 The pure relative translations case

This case was studied in [Wolf and Shashua, 2002]. We have a setup of p rigid bodies that all share the same rotation, and move with repsect to each other only by

(relative) translations. In this case the rigid motion matrix of the k-th rigid body in the i-th frame can be written as

$$\mathtt{Q}_i^k = \begin{bmatrix} \mathtt{R}_i & \mathbf{t}_i^k \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{5.25}$$

Comparing to (5.21), we can see that in the above the rotation matrix at every frame $\mathtt{R}_i$ does not depend on the rigid body k. Recall from (5.23) that

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathtt{Q}_i^{c_j}\mathbf{X}_j \tag{5.26}$$

By representing $\mathtt{Q}_i^k$ as in (5.25) and $\mathbf{X}_j$ as $[X_j, Y_j, Z_j, 1]^T$ the above gives

$$\begin{aligned} \lambda_{ij}\mathbf{x}_{ij} &= \mathtt{P}_i \begin{bmatrix} \mathtt{R}_i & \mathbf{t}_i^{c_j} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{pmatrix} X_j \\ Y_j \\ Z_j \\ 1 \end{pmatrix} \\ &= \mathtt{P}_i \begin{bmatrix} \mathtt{R}_i & \mathbf{t}_i^1 & \mathbf{t}_i^2 & \cdots & \mathbf{t}_i^p \\ \mathbf{0}^T & 1 & 1 & \cdots & 1 \end{bmatrix} \begin{pmatrix} X_j \\ Y_j \\ Z_j \\ \mathbf{e}_{c_j} \end{pmatrix}. \end{aligned} \tag{5.27}$$

where $\mathbf{e}_{c_j} \in \mathbb{R}^p$ is the c_j-th standard basis of $\mathbb{R}^p$. By taking

$$\mathtt{M}_i = \mathtt{P}_i \begin{bmatrix} \mathtt{R}_i & \mathbf{t}_i^1 & \mathbf{t}_i^2 & \cdots & \mathbf{t}_i^p \\ \mathbf{0}^T & 1 & 1 & \cdots & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{Y}_j = \begin{pmatrix} X_j \\ Y_j \\ Z_j \\ \mathbf{e}_{c_j} \end{pmatrix} \tag{5.28}$$

we can write

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{M}_i\mathbf{Y}_j, \tag{5.29}$$

where $\mathtt{M}_i \in \mathbb{R}^{3\times(p+3)}$ and $\mathbf{Y}_j \in \mathbb{R}^{3\times(p+3)}$. It shows that with p rigid bodies, the problem of motion segmentation with pure translation can be modeled as projections from $\mathbb{P}^{p+2}$ to $\mathbb{P}^2$. Since $\mathbf{x}_{ij}$-s are given, by performing a high-dimensional projective reconstruction, one can obtain the projective depths λ_{ij} up to a diagonal ambiguity. Then, as mentioned before, motions can be segmented by applying subspace clustering to the columns of the weighted data matrix $[\lambda_{ij}\mathbf{x}_{ij}]$. Notice that the fact the matrix of depths $\Lambda = [\lambda_{ij}]$ is obtained up to a diagonal ambiguity does not alter this property that columns of $[\lambda_{ij}\mathbf{x}_{ij}]$ lie on a mixture of linear subspaces.

5.1.3.2 The coplanar motions case

Assume that all the rigid objects have a coplanar rotation, that is, all rotate around a common axis $\mathbf{u}$, which is the unit normal vector to the plane of rotation. Each

object has an arbitrary translation which is not necessarily in the plane of rotation. Consider the unit vectors $\mathbf{v}$ and $\mathbf{w}$ foring the orthogonal complement of $\mathbf{u}$ such that the matrix

$$\mathtt{U} = [\mathbf{w}, \mathbf{v}, \mathbf{u}] \tag{5.30}$$

is a rotation matrix. Therefore, $\mathbf{v}$ and $\mathbf{w}$ form a basis for the plane of rotation. In this case, the rotation matrix of rigid body k at frame i has the form of

$$\mathtt{R}_i^k = \mathtt{U} \begin{bmatrix} \mathtt{C}_i^k & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathtt{U}^T. \tag{5.31}$$

where $\mathtt{C}_i^k$ is a 2D rotation matrix, that is

$$\mathtt{C}_i^k = \begin{bmatrix} \cos(\theta_i^k) & -\sin(\theta_i^k) \\ \sin(\theta_i^k) & \cos(\theta_i^k) \end{bmatrix}. \tag{5.32}$$

with θ_i^k being the angle of rotation. From (5.31) and (5.30), we can write $\mathtt{R}_i^k$ as

$$\mathtt{R}_i^k = \begin{bmatrix} [\mathbf{w}, \mathbf{v}]\, \mathtt{C}_i^k & \mathbf{u} \end{bmatrix} \mathtt{U}^T = \begin{bmatrix} \mathtt{B}_i^k & \mathbf{u} \end{bmatrix} \mathtt{U}^T. \tag{5.33}$$

where $\mathtt{B}_i^k = [\mathbf{w}, \mathbf{v}]\, \mathtt{C}_i^k$. Now, the projection equation can be written as

$$\begin{aligned} \lambda_{ij}\mathbf{x}_{ij} &= \mathtt{P}_i\, \mathtt{Q}_i^{c_j}\, \mathbf{X}_j \\ &= \mathtt{P}_i \begin{bmatrix} \mathtt{R}_i^{c_j} & \mathbf{t}_i^{c_j} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{pmatrix} X_j \\ Y_j \\ Z_j \\ 1 \end{pmatrix} \\ &= \mathtt{P}_i \begin{bmatrix} [\mathtt{B}_i^{c_j}, \mathbf{u}]\mathtt{U}^T & \mathbf{t}_i^{c_j} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{pmatrix} X_j \\ Y_j \\ Z_j \\ 1 \end{pmatrix}. \end{aligned} \tag{5.34}$$

define X_j', Y_j' and Z_j' as

$$\begin{pmatrix} X_j' \\ Y_j' \\ Z_j' \end{pmatrix} = \mathtt{U}^T \begin{pmatrix} X_j \\ Y_j \\ Z_j \end{pmatrix}, \tag{5.35}$$

Now, the derivation (5.34) can be continued as

$$\begin{aligned}
\lambda_{ij}\mathbf{x}_{ij} &= \mathtt{P}_i \begin{bmatrix} [\mathtt{B}_i^{c_j}, \mathbf{u}] & \mathbf{t}_i^{c_j} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{pmatrix} X'_j \\ Y'_j \\ Z'_j \\ 1 \end{pmatrix} \\
&= \mathtt{P}_i \begin{bmatrix} \mathtt{B}_i^{c_j} & \mathbf{u} & \mathbf{t}_i^{c_j} \\ \mathbf{0}^T & 0 & 1 \end{bmatrix} \begin{pmatrix} X'_j \\ Y'_j \\ Z'_j \\ 1 \end{pmatrix} \\
&= \mathtt{P}_i \begin{bmatrix} \mathtt{B}_i^1 & \mathtt{B}_i^2 & \cdots & \mathtt{B}_i^p & \mathbf{u} & \mathbf{t}_i^1 & \mathbf{t}_i^2 & \cdots & \mathbf{t}_i^p \\ \mathbf{0}^T & \mathbf{0}^T & \cdots & \mathbf{0}^T & 0 & 1 & 1 & \cdots & 1 \end{bmatrix} \begin{pmatrix} \mathbf{e}_{c_j} \otimes \begin{pmatrix} X'_j \\ Y'_j \end{pmatrix} \\ Z'_j \\ \mathbf{e}_{c_j} \end{pmatrix}.
\end{aligned} \tag{5.36}$$

where $\otimes$ is the Kronecker product and $\mathbf{e}_{c_j} \in \mathbb{R}^p$ is the c_j-th standard basis. Notice that $\mathbf{Y}_j \mathbf{e}_{c_j} \otimes [X'_j, Y'_j]^T \in \mathbb{R}^{2p}$. Now, if we take

$$\mathtt{M}_i = \mathtt{P}_i \begin{bmatrix} \mathtt{B}_i^1 & \mathtt{B}_i^2 & \cdots & \mathtt{B}_i^p & \mathbf{u} & \mathbf{t}_i^1 & \mathbf{t}_i^2 & \cdots & \mathbf{t}_i^p \\ \mathbf{0}^T & \mathbf{0}^T & \cdots & \mathbf{0}^T & 0 & 1 & 1 & \cdots & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{Y}_j = \begin{pmatrix} \mathbf{e}_{c_j} \otimes \begin{pmatrix} X'_j \\ Y'_j \end{pmatrix} \\ Z'_j \\ \mathbf{e}_{c_j} \end{pmatrix}, \tag{5.37}$$

we can write

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{M}_i \mathbf{Y}_j, \tag{5.38}$$

The matrix $\mathtt{M}_i$ is 3 by $(3p+1)$, and $\mathbf{Y}_j \in \mathbb{R}^{3p+1}$. It shows that the problem of motion segmentation with p rigid bodies undergoing a coplanar rotation can be modeled as projections $\mathbb{P}^{3p} \to \mathbb{P}^2$. The projective depths λ_{ij} can be obtained up to a diagonal equivalence through high-dimensional projective reconstruction, and the motions can be segmented via subspace clustering, as discussed before.

5.1.3.3 General rigid motions

We consider the case of general rigid motions. Remember the projection relation for the multi-body case

$$\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i \mathtt{Q}_i^{c_j} \mathbf{X}_j, \tag{5.39}$$

where $\mathtt{Q}_i^k \in \mathbb{R}^{4\times 4}$ shows the rigid motion matrix of the k-th rigid body at frame i, and $c_j \in \{1,2,\ldots,p\}$ is the rigid body to which the $\mathbf{X}_j$ belong. We can write the above as

$$\begin{aligned}\lambda_{ij}\mathbf{x}_{ij} &= \mathtt{P}_i[\mathtt{Q}_i^1, \mathtt{Q}_i^2, \ldots, \mathtt{Q}_i^p]\,(\mathbf{e}_{c_j} \otimes \mathbf{X}_j),\\ &= \mathtt{M}_i\mathbf{Y}_j,\end{aligned} \tag{5.40}$$

where, $\mathtt{M}_i = \mathtt{P}_i[\mathtt{Q}_i^1, \mathtt{Q}_i^2, \ldots, \mathtt{Q}_i^p] \in \mathbb{R}^{3\times 4p}$ and $\mathbf{Y}_j = (\mathbf{e}_{c_j} \otimes \mathbf{X}_j) \in \mathbb{R}^{4p}$. Notice that the Kronecker product $\mathbf{e}_k \otimes \mathbf{X}_j$ is in the form of

$$\mathbf{Y}_j = (\mathbf{e}_{c_j} \otimes \mathbf{X}_j) = \begin{pmatrix} \mathbf{0}_{4k-4} \\ \mathbf{X}_j \\ \mathbf{0}_{4p-4k} \end{pmatrix}.$$

This means that if $\mathbf{X}_j$ belongs to the k-th rigid body ($c_j = k$), the high-dimensional point $\mathbf{Y}_j\mathbb{R}^{4p}$ is the stack of p blocks of vectors of size 4, such that the k-th block is equal to $\mathbf{X}_j$ and the rest of them are zero.

Actully, the application of projective reconstruction in this case needs further investigation as the reconstruction is not unique up to projectivity. This means that the points $\mathbf{Y}_j$ have some special nongeneric structure such that they cannot be uniquely reconstructed given the image points $\mathbf{x}_{ij}$. Notice that, by dividing each $\mathtt{M}_i \in \mathbb{R}^{3\times 4p}$ into 3×4 blocks as

$$\mathtt{M}_i = [\mathtt{M}_i^1, \mathtt{M}_i^2, \cdots, \mathtt{M}_i^p] \tag{5.41}$$

Then, considering the form of $\mathbf{Y}_j = (\mathbf{e}_{c_j} \otimes \mathbf{X}_j)$, for the points $\mathbf{X}_j$ belonging to the k-th rigid body we have

$$\hat{\lambda}_{ij}\mathbf{x}_{ij} = \mathtt{M}_i^k\mathbf{X}_j \qquad \text{for all } j \text{ such that } c_j = k \tag{5.42}$$

Therefore, each set of points belonging to a certain rigid body corresponds to a projective reconstruction problem which is independent of the reconstruction problem associated with other rigid bodies. Each projection matrix $\mathtt{M}_i^k$, thus, can be recovered up to a projective ambiguity, that is a valid reconstruction $\hat{\mathtt{M}}_i^k$ is in the form of

$$\hat{\mathtt{M}}_i^k = \tau_i^k\mathtt{M}_i^k\mathtt{H}^k \tag{5.43}$$

Therefore, the ambiguity of the higher-dimensional projective matrix $\mathtt{M}_i = [\mathtt{M}_i^1, \mathtt{M}_i^2, \cdots, \mathtt{M}_i^p]$ is in the form of

$$\hat{\mathtt{M}}_i = \begin{bmatrix} \tau_i^1\mathtt{M}_i^1 & \tau_i^2\mathtt{M}_i^2 & \cdots & \tau_i^p\mathtt{M}_i^p \end{bmatrix} \begin{bmatrix} \mathtt{H}^1 & & & \\ & \mathtt{H}^2 & & \\ & & \ddots & \\ & & & \mathtt{H}^p \end{bmatrix} \tag{5.44}$$

Any solution of the above is a valid reconstruction projecting into the same image

points $\mathbf{x}_{ij}$ (given appropriate HD points $\hat{Y}_j$). This is while a projective ambiguity for the projection matrix $\mathtt{M}_i$ is in the form of

$$\hat{\mathtt{M}}_i = \tau_i' \, \mathtt{M} \, \mathtt{H}' = \tau_i' \, [\mathtt{M}_i^1, \mathtt{M}_i^2, \cdots, \mathtt{M}_i^p] \, \mathtt{H}' \tag{5.45}$$

Therefore, in this case, by solving the projection equations we might obtain solutions which are not projective equivalent to the true solution. An open question is whether our knowledge the special form of the projection matrices $\mathtt{M}_i^k$, namely $\mathtt{M}_i^k = \mathtt{P}_i \mathtt{Q}_i^k$, can help to deal with this further ambiguity. Another question is whether handling this ambiguity is necessary at all.

5.2 Nonrigid Shape Recovery

One way to model nonrigid deformations in a scene is assuming that the shape at each time is a linear combination of a set of shape bases. This has been frist proposed by Bregler et al. [2000] under the assuption of orthographic projections. The idea can be adapted for perspective cameras [Xiao and Kanade, 2005; Vidal and Abretske, 2006; Hartley and Vidal, 2008] as follows.

Consider n scene points indexed by j and m frames (time steps) indexed by i. We represent the 3D location of the j-th point at time i by $\mathbf{X}_{ij}' \in \mathbb{R}^3$, and the collection of points at time i by the shape matrix $\mathtt{X}_i' = [\mathbf{X}_{i1}' \mathbf{X}_{i2}' \cdots \mathbf{X}_{in}'] \in \mathbb{R}^{3\times n}$. Here, we use the "prime" symbol to distinguish $\mathbf{X}_{ij}' \in \mathbb{R}^3$ from the homogeneous coordinate representation $\mathbf{X}_{ij} \in \mathbb{R}^4$ of the 3D points. Now, we assume that the collection of point $\mathtt{X}_i'$ at each view can be written as a linear combination of a set of p rigid bases $\mathtt{B}_1, \mathtt{B}_2, \ldots, \mathtt{B}_p$. In other words, the location of points at the i-th frame is given by

$$\mathtt{X}_i' = \sum_{k=1}^{p} c_{ik} \mathtt{B}_k \tag{5.46}$$

If $\mathbf{b}_{kj}$ represents the j-th column of $\mathtt{B}_k$, the above gives

$$\mathbf{X}_{ij}' = \sum_{k=1}^{p} c_{ik} \mathbf{b}_{kj}. \tag{5.47}$$

Now, assume that we have 2D images $\mathbf{x}_{ij} \in \mathbb{R}^3$ (in homogeneous coordinates) of the 3D points at each frame taken by a projective camera, where the camera matrix for the i-th frame is $\mathtt{P}_i$ ($\mathtt{P}_i$-s can be potentially the same). If we divide the camera matrices as $\mathtt{P}_i = [\mathtt{Q}_i \mathbf{t}_i]$ with $\mathtt{Q}_i \in \mathbb{R}^{3\times 3}$ and $\mathbf{t}_i \in \mathbb{R}^3$, then the projection equation can be written

as

$$\begin{aligned}\lambda_{ij}\mathbf{x}_{ij} &= \mathtt{Q}_i\mathbf{X}'_{ij} + \mathbf{t}_i \\ &= \mathtt{Q}_i(\sum_{k=1}^{p} c_{ik}\mathbf{b}_{kj}) + \mathbf{t}_i \\ &= [c_{i1}\mathtt{Q}_i,\, c_{i2}\mathtt{Q}_i,\, \ldots, c_{ip}\,\mathtt{Q}_i,\, \mathbf{t}_i] \begin{pmatrix} \mathbf{b}_{1j} \\ \mathbf{b}_{2j} \\ \vdots \\ \mathbf{b}_{pj} \\ 1 \end{pmatrix} \\ &= \mathtt{M}_i\mathbf{Y}_j, \end{aligned} \tag{5.48}$$

where $\mathtt{M}_i \in \mathbb{R}^{3\times(3p+1)}$ and $\mathbf{Y}_j \in \mathbb{R}^{3p+1}$. This is obviously a projection from $\mathbb{P}^{3p}$ to $\mathbb{P}^2$. We refer the reader to [Hartley and Vidal, 2008] for more details.

The problem of nonrigid motion recovery is to recover the basis matrices $\mathtt{B}_k$, the camera matrices $\mathtt{P}_i = [\mathtt{Q}_i\mathbf{t}_i]$ and the coefficients c_{ik}, given the image points $\mathbf{x}_{ij}$. The first step in solving this problem is to recover the high-dimensional projection matrices $\mathtt{M}_i$ and the points $\mathbf{Y}_j$, up to projectivity, via some high-dimensional projective reconstruction algorithm. After this step, the camera matrices $\mathtt{P}_i$, the shape matrices $\mathtt{B}_k$ and the coefficients c_{ik} can be recovered (up to an ambiguity) by imposing the special block-wise structure of the matrices $\mathtt{M}_i$ given in (5.48) using the degrees of freedom from the projective ambiguity in recovering $\mathtt{M}_i$-s and $\mathbf{Y}_j$-s.

This problem has been looked into in [Hartley and Vidal, 2008], where the projective reconstruction is conducted using the tensor-based technique proposed by Hartley and Schaffalitzky [2004]. After the projective reconstruction an algebraic approach is proposed for the recovery of $\mathtt{P}_i$-s, $\mathtt{B}_k$-s and c_{ik}-s.

5.3 Correspondence Free Structure from Motion

Angst and Pollefeys [2013] consider the case of a rigid rig of multiple *affine* cameras observing a scene with a global rigid motion. The input to the problem is tracks of points captured by each camera. However, point correspondences between the cameras are not required. The cameras may observe non-overlapping parts of the scenes. The central idea come from the fact that "all cameras are observing a common motion". They show that, if the scene has a general motion, the problem involves a rank 13 factorization. In the case of *planar motions* it involves a rank 5 factorization.

Here, we describe the idea in the context of projective cameras. Consider a set of m projective cameras with camera matrices $\mathtt{P}_1, \mathtt{P}_2, \ldots, \mathtt{P}_m \in \mathbb{R}^{3\times4}$. Each camera observes *a subset* of the scene points during p frames (time steps). We represent the points observed by the i-th camera by $\mathbf{X}_{i1}, \mathbf{X}_{i2}, \ldots, \mathbf{X}_{in_i}$. Each point $\mathbf{X}_{ij}$ is visible in all frames, which means that incomplete tracks are disregarded. Notice that, as a scene point can be observed by several cameras, we might have the case where for the two

cameras i and i', the two vectors $\mathbf{X}_{ij}$ and $\mathbf{X}_{i',j'}$ are identical. In this method, however, $\mathbf{X}_{ij}$ and $\mathbf{X}_{i',j'}$ are treated as different points. Therefore, the method does not need information about point correspondences between different cameras.

Considering a projective camera model, the image of the j-th point observed by the i-th camera at the k-th frame is created by

$$\lambda_{ij}^f \mathbf{x}_{ij}^f = \mathtt{P}_i \mathtt{Q}^f \mathbf{X}_{ij} \tag{5.49}$$

where $\mathtt{Q}^f \in \mathbb{R}^{4\times4}$ represents the rigid motion matrix of the f-th frame, $\mathbf{x}_{ij}^f \in \mathbb{R}^3$ is the image point and λ_{ij}^f is the projective depth. Remember from Sect. 5.1 that the rigid motion matrix has the form of

$$\mathtt{Q}^f = \begin{bmatrix} \mathtt{R}^f & \mathbf{t}^f \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{5.50}$$

where $\mathtt{R}^f$ and $\mathbf{t}^f$ are respectively the rotation matrix and the translation vector of the f-th frame. Notice that, as all the scene points undergo a common rigid motion, the motion matrix only depends on the frame f. By considering $\mathtt{R}^f = [\mathbf{r}_1^f, \mathbf{r}_2^f, \mathbf{r}_3^f]$, $\mathbf{X}_{ij} = [X_{ij}, Y_{ij}, Z_{ij}, 1]^T$ and $\mathtt{P}_i = [\mathtt{A}_i, \mathbf{b}_i]$ with $\mathtt{A}_i \in \mathbb{R}^{3\times3}$ and $\mathbf{b}_i \in \mathbb{R}^3$, we have

$$\begin{aligned}
\lambda_{ij}^f \mathbf{x}_{ij}^f &= \begin{bmatrix} \mathtt{A}_i & \mathbf{b}_i \end{bmatrix} \begin{bmatrix} \mathtt{R}^f & \mathbf{t}^f \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}_{ij} \\
&= \begin{bmatrix} \mathtt{A}_i\mathtt{R}^f & \mathtt{A}_i\mathbf{t}^f + \mathbf{b}_i \end{bmatrix} \mathbf{X}_{ij} \\
&= \begin{bmatrix} \mathtt{A}_i\mathbf{r}_1^f & \mathtt{A}_i\mathbf{r}_2^f & \mathtt{A}_i\mathbf{r}_3^f & \mathtt{A}_i\mathbf{t}^f + \mathbf{b}_i \end{bmatrix} \begin{pmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \\ 1 \end{pmatrix} \\
&= X_{ij}\,\mathtt{A}_i\,\mathbf{r}_1^f + Y_{ij}\,\mathtt{A}_i\,\mathbf{r}_2^f + Z_{ij}\,\mathtt{A}_i\,\mathbf{r}_3^f + \mathtt{A}_i\mathbf{t}^f + \mathbf{b}_i \\
&= \begin{bmatrix} X_{ij}\mathtt{A}_i & Y_{ij}\mathtt{A}_i & Z_{ij}\mathtt{A}_i & \mathtt{A}_i & \mathbf{b}_i \end{bmatrix} \begin{pmatrix} \mathbf{r}_1^f \\ \mathbf{r}_2^f \\ \mathbf{r}_3^f \\ \mathbf{t}^f \\ 1 \end{pmatrix}. \\
&= \mathtt{M}_{ij}\mathbf{Y}_f
\end{aligned} \tag{5.51}$$

where

$$\mathtt{M}_{ij} = \begin{bmatrix} X_{ij}\mathtt{A}_i & Y_{ij}\mathtt{A}_i & Z_{ij}\mathtt{A}_i & \mathtt{A}_i & \mathbf{b}_i \end{bmatrix} \in \mathbb{R}^{3\times13} \tag{5.52}$$

$$\mathbf{Y}_f = \text{stack}(\mathbf{r}_1^f, \mathbf{r}_2^f, \mathbf{r}_3^f, \mathbf{t}^f, 1) \in \mathbb{R}^{13} \tag{5.53}$$

This can be seen as a projection from $\mathbb{P}^{12}$ to $\mathbb{P}^2$. Notice that projection matrices $\mathtt{M}_{ij}$ are indexed by a pair (i,j). This means that corresponding to every point $\mathbf{X}_{ij}$ observed

in camera i there exists a distinct high-dimensional projection matrix $\mathtt{M}_{ij}$.

By solving a projective reconstruction problem one can obtain $\mathtt{M}_{ij}$-s and $\mathbf{Y}_f$-s up to a projective ambiguity. One should set the free parameters of this ambiguity such that the projection matrices $\mathtt{M}_{ij}$ and points $\mathbf{Y}_f$ conform with the required structures shown in (5.52) and (5.53). This has been done by Angst and Pollefeys [2013] for an affine ambiguity. However, solving the problem for the projective camera model is still an open question.

5.4 Summary

We considered different scene analysis problems and demonstrated how they can be modeled as projections from higher-dimensional projective spaces to $\mathbb{P}^2$.

Chapter 6

Experimental Results

The results provided in this thesis are not bound to any particular algorithm and our research was not concerned with convergence properties or how to find global minima. The aim of this chapter is, therefore, the verification of our theory by implementing a basic iterative factorization procedure and showing the algorithm's behaviour for different choices of the depth constraints, in terms of finding the correct solutions. Especially, we present cases in which the degenerate false solutions discussed in the previous chapters happen in the factorization-based algorithms, and demonstrate how the use of proper constraints can help to avoid them.

6.1 Constraints and Algorithms

Given the image data matrix $[\mathbf{x}_{ij}]$ and a constraint space C, we estimate the depths by solving the following optimization problem:

$$\min_{\hat{\Lambda},\hat{\mathbf{P}},\hat{\mathbf{X}}} \quad \left\|\hat{\Lambda} \odot [\mathbf{x}_{ij}] - \hat{\mathbf{P}}\hat{\mathbf{X}}\right\|_F \quad \text{subject to} \quad \hat{\Lambda} \in C, \tag{6.1}$$

where $\hat{\Lambda} \in \mathbb{R}^{m\times n}$, $\hat{\mathbf{X}} \in \mathbb{R}^{m\times r}$ and $\hat{\mathbf{P}} \in \mathbb{R}^{r\times n}$ for a configuration of m views and n points. Thus, for 3D to 2D projections we have $\hat{\mathbf{X}} \in \mathbb{R}^{m\times 4}$ and $\hat{\mathbf{P}} \in \mathbb{R}^{4\times n}$. Clearly, when the data is noise-free (that is $\mathbf{x}_{ij}$ exactly equals $\mathbf{P}_i\mathbf{X}_j/\lambda_{ij}$ for all i,j), and the constraint space C is inclusive (allows at least one correct solution), the above problem has global minima with zero target value, including the correct solutions. For 3D to 2D projections, we can say that if the constraint space is also exclusive (excludes all the false solutions), and therefore is *reconstruction friendly*, the global minima contain only the correct solutions for which $(\{\hat{\mathbf{P}}_i\},\{\hat{\mathbf{X}}_j\})$ are projectively equivalent to the true configuration $(\{\mathbf{P}_i\},\{\mathbf{X}_j\})$.

Here, we try to solve (6.1) by alternatingly minimizing over different sets of variables.

To make a clear comparison, among many different possible choices for depth constraints, we choose only four, each representing one class of constraints discussed before. A schema of these four constraints is depicted in Fig. 6.1. The first two constraints are linear equality ones and the next two are examples of compact constraint spaces. The first constraint, abbreviated as ES-MASK is a masked constraint which

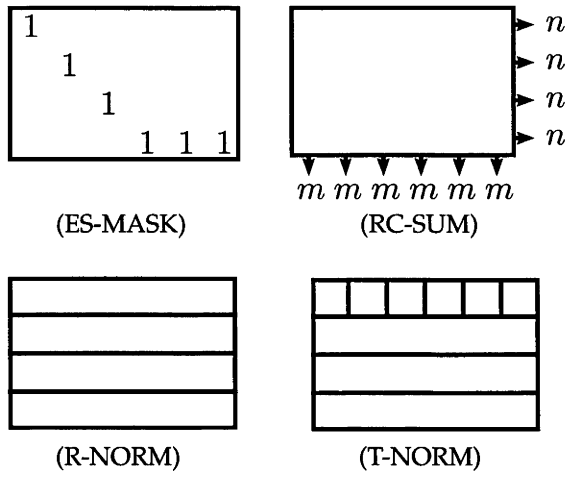


Figure 6.1: Four constraints implemented for the experiments. ES-MASK is a masked constraint with an edgeless step-like mask M. The constraint fixes some elements of $\hat{\Lambda}$ according to $\mathtt{M} \circ \hat{\Lambda} = \mathtt{M}$. RC-SUM fixes row and column sums according to $\hat{\Lambda}\mathbf{1}_n = n\mathbf{1}_m, \hat{\Lambda}^T\mathbf{1}_m = m\mathbf{1}_n$. R-NORM fixes a weighted l^2-norm of each rows of $\hat{\Lambda}$, and T-NORM fixes a weighted l^2-norm of tiles of $\hat{\Lambda}$.

fixes some elements of $\hat{\Lambda}$ according to $\mathtt{M} \circ \hat{\Lambda} = \mathtt{M}$ for a mask M. ES-MASK uses a specific exclusive edgeless step-like mask. In the case of a fat depth matrix ($n \geq m$), this mask is the horizontal concatenation of an $m \times m$ identity matrix and an $m \times (n-m)$ matrix whose last row consists of ones and its rest of elements are zero (see Fig. 6.1). A similar choice can be made for tall matrices. We choose the edgeless step-like mask as our experiments show that it converges more quickly than the edged version (see Sect. 3.3.2.3 for a discussion). We showed in Sect. 3.3.2.3 that this constraint rules out all false solutions for 3D to 2D projections. The second-constraint, RC-SUM, makes the rows of $\hat{\Lambda}$ sum up to n and its columns sum up to m, that is $\hat{\Lambda}\mathbf{1}_n = n\mathbf{1}_m, \hat{\Lambda}^T\mathbf{1}_m = m\mathbf{1}_n$ (Sect. 3.3.2.1). The third constraint, R-NORM, requires rows of the depth matrix to have a unit norm (Sect. 3.3.1.3). The final constraint, T-norm, is requiring tiles of the depth matrix to have a unit norm (Sect. 3.3.1.4), where the tiling is done according to Fig. 6.1. The last two constraints constraints can be considered as examples of tiled constraints (see Sect. 3.3.1.4). The norm use in these two constraints are weighted l^2-norms with special weights, as follows. For an $m' \times n'$ tile ($m' = 1$ or $n' = 1$) in the depth matrix, the constraint is that the corresponding $3m' \times n'$ block in $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ has a unit Frobenius norm, which amounts to a unit weighted l^2-norm for the corresponding $m' \times n'$ block of $\hat{\Lambda}$. For example, consider a horizontal tile in the form of $[\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{in'}]$. The corresponding constraint used here is that the $3 \times n'$ matrix $[\lambda_{i1}\mathbf{x}_{i1}, \lambda_{i2}\mathbf{x}_{i2}, \ldots, \lambda_{in'}\mathbf{x}_{in'}]$ has a unit Frobenius norm. This is equivalent to a weighted l^2-norm of the vector $[\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{in'}]$ where the weight corresponding to the j-th entry is equal to $\|\mathbf{x}_{ij}\|_2$.

With linear equality constraints, we consider two algorithms for the minimization

of (6.1). The first algorithm is to iterate between minimizing with respect to $\hat{\Lambda}$ (subject to the depth constraint $\hat{\Lambda} \in C$) and minimizing with respect to $(\hat{\mathtt{X}}, \hat{\mathtt{P}})$. The former step is minimizing a positive definite quadratic form with respect to a linear constraint, which has a closed-form solution, and the latter can be done by a rank-4 SVD thresholding of $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ and factorizing the rank-4 matrix as $\hat{\mathtt{P}}\hat{\mathtt{X}}$. The second approach is to alternate between minimizing with respect to $(\hat{\Lambda}, \hat{\mathtt{P}})$ and $(\hat{\Lambda}, \hat{\mathtt{X}})$. Similar to the first step of the first algorithm, each step of this algorithm has a closed-form solution. While the second method is generally harder to implement, our experiments show that it results in faster convergence. Here, we use the second method for optimizing with respect to ES-MASK. For optimizing with respect to RC-SUM we use the first method to get a less complex optimization formula at each step.

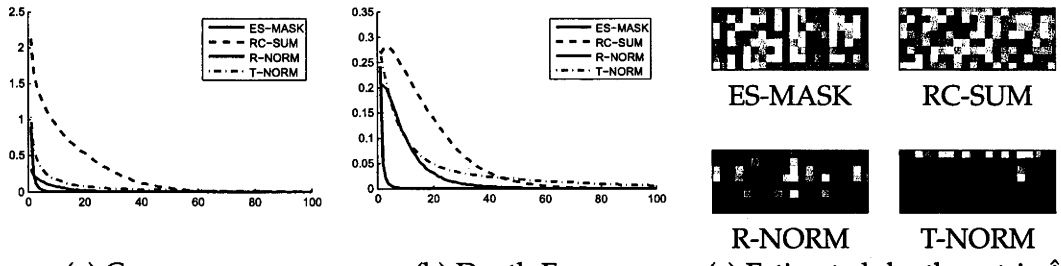
The last two constraints are both examples of tiling constraints. Our method for optimizing (6.1) is to alternatingly minimize with respect to $\hat{\Lambda}$ and then with respect to $(\hat{\mathtt{X}}, \hat{\mathtt{P}})$. The latter is done by a rank-4 SVD thresholding of $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ and factorization. For the former step, we fix $\hat{\mathtt{P}}\hat{\mathtt{X}}$ and minimize $\left\|\hat{\Lambda} \odot [\mathbf{x}_{ij}] - \hat{\mathtt{P}}\hat{\mathtt{X}}\right\|_F$ subject to the constraint that for each $m' \times n'$ tile of $\hat{\Lambda}$, the corresponding $3m' \times n'$ block of $\hat{\Lambda} \odot [\mathbf{x}_{ij}]$ has unit Frobenius norm. This means that, each tile of $\hat{\Lambda}$ can be optimized separately. Showing by $\hat{\boldsymbol{\lambda}}$, the vector of elements of $\hat{\Lambda}$ belonging to a special tile, the corresponding optimization problem for this tile is in the form of $\min_{\hat{\boldsymbol{\lambda}}} \|\mathtt{A}\hat{\boldsymbol{\lambda}} - \mathbf{b}\|_2$ with respect to $\|\mathtt{W}\hat{\boldsymbol{\lambda}}\|_2 = 1$ for some matrix $\mathtt{W}$ and some vector $\mathbf{b}$. This problem has a closed-form solution. For 1×1 tiles we fix the value of the corresponding $\hat{\lambda}_{ij}$ to 1.

6.2 3D to 2D projections

6.2.1 Synthetic Data

We take a configuration of 8 views and 20 points. The elements of the matrices $\mathtt{P}_i$ and points $\mathbf{X}_j$ are sampled according to a standard normal distribution. The depths are taken to be $\lambda_{ij} = 3 + \eta_{ij}$, where the η_{ij}-s are sampled from a standard normal distribution. This way we can get a fairly wide range of depths. Negative depths are not allowed, and if they happen, we repeat the sampling. This is mainly because of the fact that for the RC-SUM constraint, the inclusiveness is only proved for positive depths. The image data is calculated according to $\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j/\lambda_{ij}$, with no added error. Notice that here, unlike in the case of real data in the next subsection, we do not require the last element of the $\mathbf{X}_j$-s and the $\mathbf{x}_{ij}$-s to be 1, and consider the projective factorization problem in its general algebraic form.

In each case, we plot the convergence graph, which is the value of the target function $\left\|\hat{\Lambda} \odot [\mathbf{x}_{ij}] - \hat{\mathtt{P}}\hat{\mathtt{X}}\right\|_F$ throughout iterations, followed by a graph of depth error. To deal with diagonal ambiguity of the depth matrix, the depth error is calculated as $\|\Lambda - \mathrm{diag}(\boldsymbol{\tau})\,\hat{\Lambda}\,\mathrm{diag}(\boldsymbol{\nu})\|$, where $\boldsymbol{\tau}$ and $\boldsymbol{\nu}$ are set such that $\mathrm{diag}(\boldsymbol{\tau})\,\hat{\Lambda}\,\mathrm{diag}(\boldsymbol{\nu})$ has the same row norms and column norms as the true depth matrix $\Lambda = [\lambda_{ij}]$. This can be done using Sinkhorn's algorithm as described in Sect. 3.3.1.2. Finally, for each constraint we depict the estimated depth matrix $\hat{\Lambda}$ as a grayscale image whose intensity values show the absolute values of the elements of $\hat{\Lambda}$.



(a) Convergence (b) Depth Error (c) Estimated depth matrix $\hat{\Lambda}$

Figure 6.2: An example where all algorithms converge to a correct solution. (a) shows all the four cases have converged to a global minimum, (b) shows that all the four cases have obtained the true depths up to diagonal equivalence, and (c) confirms this by showing that the depth matrix $\hat{\Lambda}$ satisfies (D1-D3). In (c) the gray-level of the image at different locations represents the absolute value of the corresponding element in $\hat{\Lambda}$.

In the first test, we set the initial value of $\hat{\Lambda}$ to $1_{m\times n}$ which is a matrix of all ones. The results for one run of the algorithm are shown in Fig. 6.2. It is clear from Fig. 6.2(a) that the algorithm has converged to a global minimum for all four constraints. Fig. 6.2(b) shows that in all four cases the algorithm has converged to a correct solution. Fig. 6.2(c) confirms this by showing that in no case the algorithm has converged to a cross-shaped solution or a solution with zero rows or zero columns.

In the second test, we set the initial value of $\hat{\Lambda}$ to be 1 at the first row and 10th column, and 0.02 elsewhere. This makes the initial $\hat{\Lambda}$ close to a cross-shaped matrix. The result is shown in Fig. 6.3. According to Fig. 6.3(a), in all cases the target error has converged to zero, meaning that a solution is found for the factorization problem $\hat{\Lambda} \odot [x_{ij}] = \hat{P}\hat{X}$. Fig. 6.3(b), shows that for the constraint ES-MASK and RC-SUM, the algorithm gives a correct solution, however, for R-NORM and T-NORM, it has converged to a wrong solution. Fig. 6.3(c) supports this by showing that the algorithm has converged to a cross-shaped solution for R-NORM and T-NORM. Although, the constraint RC-SUM allows for cross-shaped configurations, according to our discussion in Sect. 3.3.2.1, it is unlikely for the algorithm to converge to a cross if the initial solution has all positive numbers (see Fig. 3.7). However, according to our experiments, if we start from a configuration close to the cross-shaped solution of the constraint RC-SUM (with a negative element at the centre of the cross), the algorithm will converge to a cross-shaped configuration.

6.2.2 Real Data

We use the Model House data set provided by the Visual Geometry Group at Oxford University[1]. As our theory does not deal with the case of missing data, from the data matrix we choose a block of 8 views and 19 points for which there is no missing data. Here, the true depths are not available. Thus, to see if the algorithm has converged

[1]http://www.robots.ox.ac.uk/~vgg/data/data-mview.html

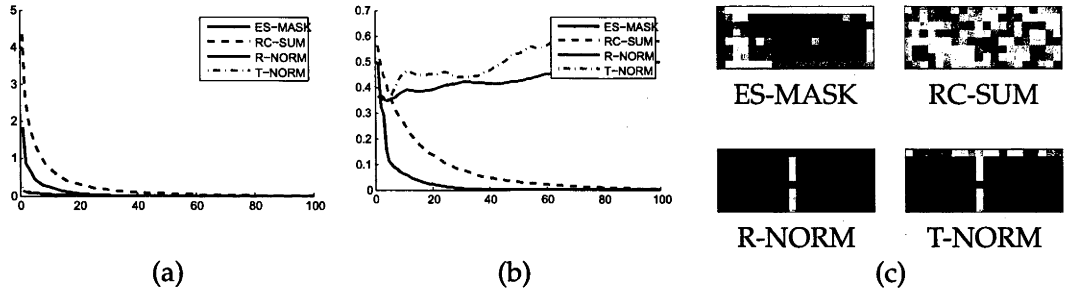


Figure 6.3: (a) the target error in all cases has converged to zero, (b) the depth error has converged to zero only for ES-MASK and RC-SUM, meaning that only ES-MASK and RC-SUM have converged to a correct solution, (c) confirms this by showing that R-NORM and T-NORM have converged to cross-shaped solutions.

to a correct solution, we use a variant of the reprojection error. The basic reprojection error is $\sum_{ij}\|\mathbf{x}_{ij} - \alpha_{ij}\hat{\mathbf{P}}_i\hat{\mathbf{X}}_j\|$ where for each i and j, α_{ij} is chosen such that the third entry of the vector $\alpha_{ij}\hat{\mathbf{P}}_i\hat{\mathbf{X}}_j$ is equal to the third entry of $\mathbf{x}_{ij}$, which is 1 in this case. However, as this can cause fluctuations in the convergence graph at iterations where the last element of $\hat{\mathbf{P}}_i\hat{\mathbf{X}}_j$ gets close to zero, we instead choose each α_{ij} such that it minimizes $\|\mathbf{x}_{ij} - \alpha_{ij}\hat{\mathbf{P}}_i\hat{\mathbf{X}}_j\|$.

Fig. 6.4 shows one run of the algorithm for each of the four constraints starting from $\hat{\Lambda} = \mathbf{1}_{m\times n}$. It can be seen that for all the constraints the algorithm has converged to a solution with a very small error. Fig. 6.4(b) shows that all of them have converged to something close to a correct solution. This is affirmed by Fig. 6.4(c), showing that no solution is close to a configuration with zero rows, zero columns or cross-shaped structure in the depth matrix. Comparing Fig. 6.4(c) with Fig. 6.2(c) one can see that the matrices in 6.4(c) are more uniform. One reason is that the true depths in the case of real data are relatively close together compared to the case of synthetic data. Except, T-NORM, all the other constraints tend to somewhat preserve this uniformity, especially when the initial solution is a uniform choice like $\mathbf{1}_{m\times n}$. T-NORM does to preserve the uniformity as it requires that each of the 1×1 tiles in the first row of the depth matrix to have a unit weighted l^2-norm, while for the rest parts of the matrix, each row is required to have a unit weighted l^2-norm. This is why other parts of the depth matrix in T-NORM look considerably darker than the first row.

In the second test we start from an initial $\hat{\Lambda}$ which is close to a cross-shaped matrix, as chosen in the second test for the synthetic data. The result is shown in Fig. 6.5. Fig. 6.5(a) shows that the RC-SUM has not converged to a solution with a small target error, but the other 3 constraints have[2]. Therefore, we cannot say anything about RC-SUM. Fig. 6.5(b) shows that R-NORM and T-NORM did not converge to a correct solution. Fig. 6.5(c) confirms this by showing that R-NORM and T-NORM have converged to something close to a cross-shaped solution.

[2]Notice that the scale of the vertical axis in Fig. 6.5(a) is different from that of Fig. 6.4(a)

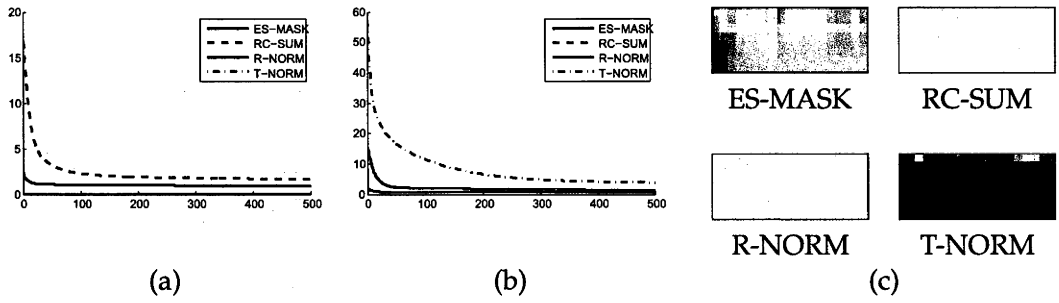


Figure 6.4: An example where all algorithms converge to a solution with a very small target value which is also close to a correct solution. In (c), one can observe a bright strip on the top of the corresponding image of T-NORM. The reason is that T-NORM forces each elements of the top row of $\hat{\Lambda}$ to have a unit (weighted l^2) norm, while for the other rows, the whole row is required to have a unit norm. See Fig. 6.1(T-NORM).

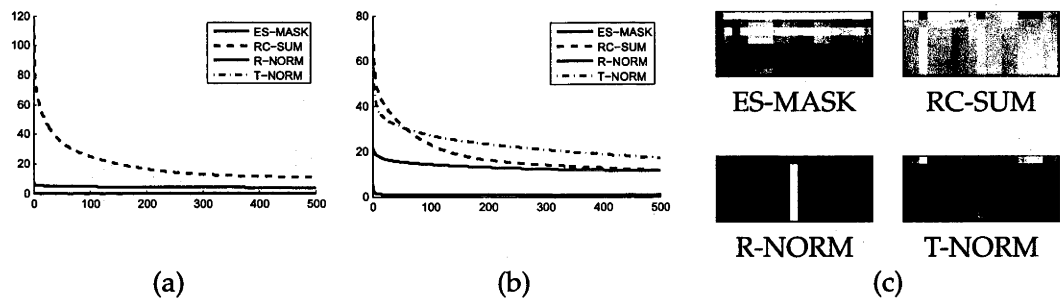


Figure 6.5: An example where the algorithms are started from an initial solution which is close to a cross-shaped matrix. (a) shows that RC-SUM has not converged to a solution with a small target error. R-NORM and T-NORM have converged to something with a small target value, but did not get close to a correct solution, as it is obvious from (b). This is confirmed by (c), which shows that R-NORM and T-NORM have converged to a something close to a cross-shaped solution.

6.3 Higher-dimensional projections

In this section we run numerical experiments to study projections from $\mathbb{P}^{r-1} \to \mathbb{P}^2$ for $r-1 > 3$. Like our experiments in Sect. 6.2.1 for synthetic data, here we consider the projective factorization problem in the general algebraic sense. We choose the elements of the projection matrices $\mathtt{P}_i \in \mathbb{R}^{3\times r}$ and HD points $\mathbf{X}_j \in \mathbb{R}^r$ as samples of a standard normal distribution. The depths are taken to be $\lambda_{ij} = 3 + \eta_{ij}$, where the η_{ij}-s are samples of a standard normal distribution, and negative depths are avoided in the similar way as in Sect. 6.2.1. The image points are created according to $\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j/\lambda_{ij}$. Notice that we do not restrict $\mathbf{X}_j$-s and the $\mathbf{x}_{ij}$-s to have a unit final element.

The experiments are conducted similarly to the previous section, with the same four constraints introduced in Fig. 6.1. The reader must keep in mind that we only have analysed the constraint for the special case of 3D to 2D projections. Therefore, it is possible that some of the so-called *reconstruction friendly* constraints defined in the context of 3D to 2D projections are unable to prevent all wrong solutions for some cases of higher dimensional projections. The effectiveness of each constraint must be studied for each class of higher dimensional projections separately.

From our results in Sect. 4.4 we can conclude that, under generic conditions, for the special case of projections $\mathbb{P}^{r-1} \to \mathbb{P}^2$ a solution $(\hat{\Lambda}, \hat{\mathtt{P}}, \hat{\mathtt{X}})$ to the projective factorization equation $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathtt{P}}\hat{\mathtt{X}}$ is projectively equivalent to the true configuration $(\Lambda, \mathtt{P}, \mathtt{X})$, if the following holds

(D1) The matrix $\hat{\Lambda} = [\hat{\lambda}_{ij}]$ has no zero rows,

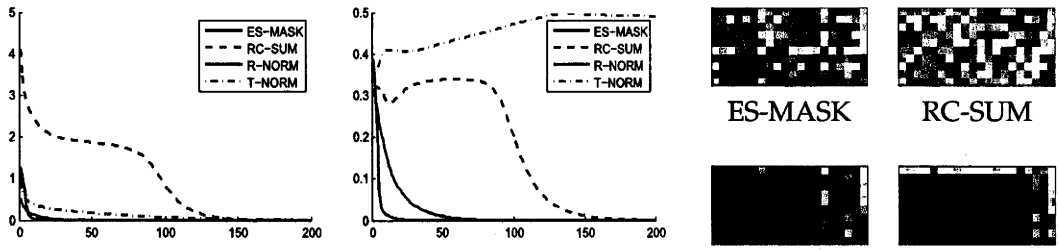
(D2) The matrix $\hat{\Lambda} = [\hat{\lambda}_{ij}]$ has no zero columns,

(D3) For every partition $\{I, J, K\}$ of views $\{1, 2, \ldots, m\}$ with $I \neq \emptyset$ and $3\,|I| + 2\,|J| < r$, the matrix $\hat{\Lambda}^K$ has *sufficiently many* nonzero columns, where $\hat{\Lambda}^K$ is the submatrix of $\hat{\Lambda}$ created by selecting rows in K.

Notice that in Sect. 4.4, the inequality condition in (D3) was stated in its general form as $\sum_{j\in I} s_i + \sum_{j\in J}(s_j - 1) < r$, instead of $3|I| + 2|J|$. Since here we only consider projections $\mathbb{P}^{r-1} \to \mathbb{P}^2$, and thus $s_i = 3$ for all i, the value of $\sum_{j\in I} s_i + \sum_{j\in J}(s_j - 1)$ is equal to $3|I| + 2|J|$, where $|\cdot|$ gives the size of a set. We study the application of the factorization-based algorithms and the wrong solution for higher-dimensional projections by running simulations for the two cases of projections $\mathbb{P}^4 \to \mathbb{P}^2$ and $\mathbb{P}^9 \to \mathbb{P}^2$.

6.3.1 Projections $\mathbb{P}^4 \to \mathbb{P}^2$

We start with the simple case of projections $\mathbb{P}^4 \to \mathbb{P}^2$. In this case we have $r = 5$. To find possible wrong solutions created by violating (D3), we need to look for partitions $\{I, J, K\}$ where I is nonempty and $3\,|I| + 2\,|J| < r = 5$. This can only happen when $|I| = 1$ and $|J| = 0$, that is I is a singleton and J is empty. It follows that $|K| = m - 1$. Therefore, in this case, wrong solutions violating (D3) happen when a submatrix $\hat{\Lambda}^K$ of $\hat{\Lambda}$, created by choosing all but one row of $\hat{\Lambda}$, has a *limited number of* nonzero columns



(a) Convergence (b) Depth Error (c) Estimated depth matrix $\hat{\Lambda}$

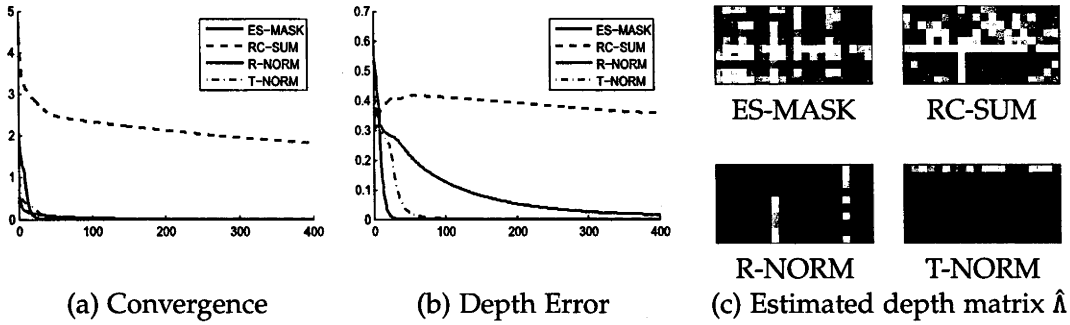
Figure 6.6: Applying the projective factorization algorithm for 4D to 2D projections. For all four cases, the cost function has converged to zero as it is obvious from (a). All cases have converged to a correct solution except for T-NORM which has converged to a wrong solution, as shown in (b). The estimated depth matrix $\hat{\Lambda}$ given by each algorithm confirms the results, as only T-NORM has given a degenerate $\hat{\Lambda}$ corresponding to a wrong solution.

(a lot of zero columns). For projections $\mathbb{P}^4 \rightarrow \mathbb{P}^2$, one can prove that, generically, this *limited number* means *at most two*. Therefore, for wrong solutions the submatrix $\hat{\Lambda}^K$ has either 1 or 2 nonzero columns.

We conduct the experiments in the same way as in Sect. 6.2. We take a configuration of 10 projection matrices and 20 points and run the iterative factorization algorithm with the four different constraints introduced in Fig. 6.1. We initiate the algorithm by a depth matrix $\hat{\Lambda}$ of all ones. The results are depicted in Fig. 6.6. Looking at the convergence graph in Fig. 6.6(a), we can expect[3] that for all four constraints the algorithm has found a solution to $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathrm{P}}\hat{\mathrm{X}}$. From the depth estimation error graph in Fig. 6.6(b), we realize that the algorithm has found a correct solution for all constraints except for T-NORM. Therefore, we can expect that the depth matrix obtained by T-NORM is degenerate, with zero patterns as described in the previous paragraph. This can be seen in Fig. 6.6(c). As expected, for the depth matrix of T-NORM, the submatrix created by choosing rows $2,3,\ldots,m$, has only 2 nonzero columns, which is the maximum possible nonzero columns for a wrong solution violating (D3). For this case of wrong solution we have $I = \{1\}$, $J = \emptyset$ and $K = \{2,3,\ldots,m\}$. According to Lemma 4.7 we must expect that the submatrix $\hat{\mathrm{P}}^K = \mathrm{stack}(\hat{\mathrm{P}}_2, \hat{\mathrm{P}}_3, \ldots, \hat{\mathrm{P}}_m)$ has rank $r' = r - (3|I| + 2|J|) = 5 - (3+0) = 2$. This can be confirmed by looking at the singular values of the matrix $\hat{\mathrm{P}}^K$ obtained from our experiments which are 1.6, 1.3, 0.005, 0.0023 and 0.0009. Note that in this example, even without starting from a close-to-degenerate initial solution, the algorithm converged to a degenerate solution for one of the constraints.

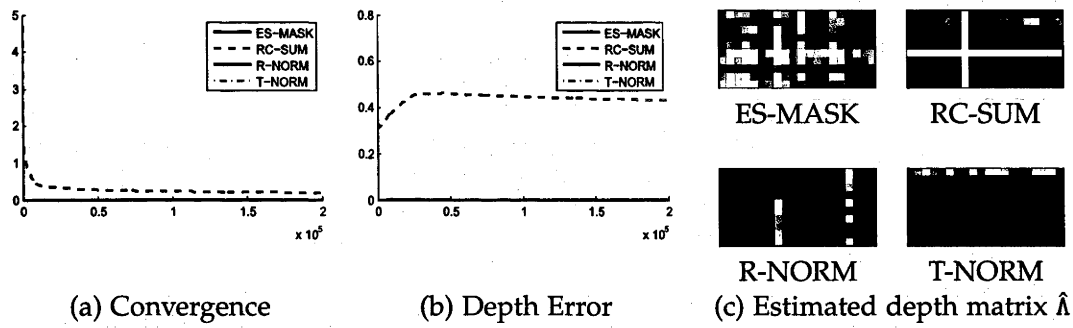
The second experiment is run exactly in the same way, with different projection matrices, HD points and projective depths, which are sampled according to the same

[3]Actually, for non-compact constraints ES-MASK and RC-SUM, there is a possibility that the cost function (and therefore $\hat{\Lambda} \odot [\mathbf{x}_{ij}] - \hat{\mathrm{P}}\hat{\mathrm{X}}$) converges to zero, but the algorithm does not converge in terms of $\hat{\Lambda}$.



(a) Convergence (b) Depth Error (c) Estimated depth matrix $\hat{\Lambda}$

Figure 6.7: Another run of the experiment with a different configuration of points, projection matrices and projective depths. The algorithm has not converged in 400 iterations for RC-SUM. For the rest of the cases, a correct solution has been found.



(a) Convergence (b) Depth Error (c) Estimated depth matrix $\hat{\Lambda}$

Figure 6.8: The result of continuing the experiment of Fig. 6.7 for 200,000 iterations. One can say that with the constraint RC-SUM, either the algorithm do not converge, or it is converging very slowly to a wrong solution. Either ways, RC-SUM has not found a correct solution.

distribution. The results are shown in Fig. 6.7. From Fig. 6.7(a) it is clear that with all constraint the cost function has converged to zero except for RC-SUM, for which the algorithm has not converged in 400 iterations. For all other three cases the algorithm has converged to a correct solution, as shown in Fig. 6.7(b) and confirmed by Fig. 6.7(c). Since the algorithm has not converged for RC-SUM, we continue the same experiment for 200,000 iterations. The result is shown in Fig. 6.8. Looking at Fig. 6.8(b), it is obvious that the algorithm for RC-SUM has not (yet) converged to a correct solution. Two scenarios are possible. The first is that the algorithm has not converged at all, in term of $\hat{\Lambda}$. This can be plausible as the constraint space of RC-SUM is compact. The second scenario is that it is converging, though extremely slowly, to a wrong solution. Fig. 6.8(c) somehow supports this hypothesis as the estimated $\hat{\Lambda}$ for RC-SUM is close to a degenerate solution[4].

[4]There is a third possibility that for RC-SUM the algorithm is converging to a local minimum. However, it is less likely as the cost seems to be (slowly) converging to zero.

6.3.2 Projections $\mathbb{P}^9 \to \mathbb{P}^2$

For projections $\mathbb{P}^9 \to \mathbb{P}^2$ we have $r = 10$. To find all possible wrong solutions violating (D3) one needs to find partitions $\{I, J, K\}$ such that $3|I| + 2|J| < r = 10$ and I is nonempty. There are 7 possibilities which can be categorized as follows:

- $|I| = 1, |J| = 0, 1, 2, 3,$
- $|I| = 2, |J| = 0, 1,$
- $|I| = 3, |J| = 0.$

Here, we conduct the experiments similarly to the previous subsection, but this time with 20 views and 40 points. In the first experiment we start with a depth matrix of all ones as the initial solution. The results are illustrated in Fig. 6.9. In this experiments the cost has converged to zero for all constraints except RC-SUM, as shown in Fig. 6.9(a). Therefore, RC-SUM has not solved the projective factorization equation $\hat{\Lambda} \odot [\mathbf{x}_{ij}] - \hat{\mathrm{P}}\hat{\mathrm{X}}$, and we cannot say anything more about it. By looking at Fig. 6.9(b), we can see that ES-MASK and R-NORM has converged to a correct solution, while T-NORM has led to a wrong solution. Thus, we must expect that T-NORM has converged to a degenerate $\hat{\Lambda}$. This is confirmed by 6.9(c), showing that in estimated depth matrix $\hat{\Lambda}$ for T-NORM only the first row has all-nonzero elements, and the matrix comprised of the rest of the rows of $\hat{\Lambda}$ have few (namely 7) nonzero columns. For this case, the corresponding partition $\{I, J, K\}$ is as follows:

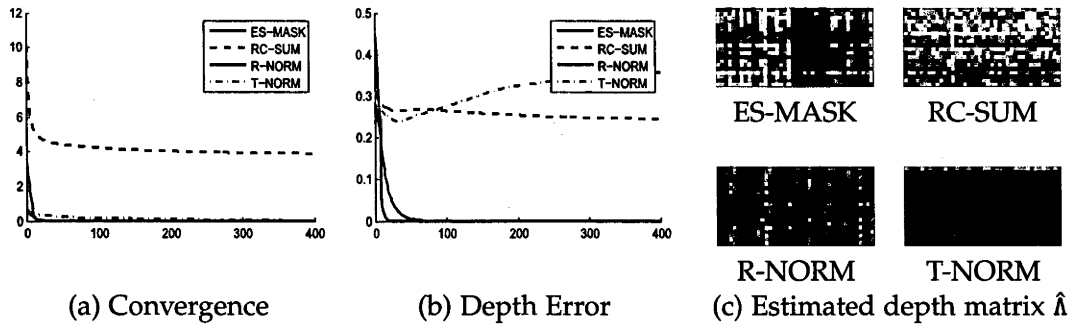
$$I = \{1\},\ J = \emptyset,\ K = \{2, 3, \ldots, 20\}$$

By Lemma 4.7 one must expect that the matrix $\hat{\mathrm{P}}^K = \mathrm{stack}(\hat{\mathrm{P}}_2, \hat{\mathrm{P}}_3, \ldots, \hat{\mathrm{P}}_m) \in \mathbb{R}^{57\times 10}$ has rank $r' = r - (3|I| + 2|J|) = 10 - (3 + 0) = 7$. This can be verified by looking at the singular values of the estimated $\hat{\mathrm{P}}^K$:

$$1.6, 1.5, 1.3, 1.2, 1.1, 0.5, 0.4, 0.000008, 0.000004, 0.000002.$$

In the next experiment, we try to produce other types of degenerate solutions. Therefore, for the initial $\hat{\Lambda}$ we set all elements of the first 3 rows and also the 10th column equal to 1. The rest of the elements are set to 0.05. The results are shown in 6.10. From Fig. 6.10(a) we can see that for the three cases RC-SUM, R-NORM and P-NORM the cost has converged to zero. For ES-MASK it seems like the cost is converging, though slowly, to zero and running the algorithm for more iterations supports this. Fig. 6.10(b) say that only ES-SUM has converged to a correct solution. From Fig. 6.10(c) we can see that T-NORM and R-NORM have converged to the degenerate solutions of the expected type (both violating depth condition (D3)).

The case of ES-MASK seems unusual. From 6.10(a) it seems that the cost is converging to zero, and from 6.10(b) it is obvious that it has not converged to a correct solution. However, the estimated depth matrix $\hat{\Lambda}$ shown in Fig. 6.10(c) for ES-MASK does not violate any of the conditions (D1-D3), even though it is somehow degenerate as the estimated $\hat{\Lambda}$ seems to have a lot of zero elements. Looking at Fig.



(a) Convergence (b) Depth Error (c) Estimated depth matrix $\hat{\Lambda}$

Figure 6.9: The results of one run of our experiments for projections $\mathbb{P}^9 \to \mathbb{P}^2$. (a) shows that the cost has converged to zero for all constraints except RC-SUM. (b) shows that only ES-MASK and RC-NORM has given a correct solution. (c) show that T-NORM has converged to a degenerate wrong solution violating (D3).

6.10(c) for ES-MASK, it is clear that the first three row, plus the last row of are in $I \cup J$, that is $I \cup J = \{1,2,3,20\}$. Thus $K = \{4,5,\ldots,19\}$. From $3|I|+2|J| < r = 10$ the only possible case is $|I| = 1, |J| = 3$. This we have $r' = r - (3|I|+2|J|) = 10-(3+6) = 1$. It can be proved that for $r' = 1$, the matrix $\hat{\Lambda}^K$ (that is the submatrix of $\hat{\Lambda}$ created by choosing rows in K) can have at most one nonzero column. However, by looking at Fig. 6.10(c), it is clear that with the chosen K, the matrix $\hat{\Lambda}^K$ has 16 nonzero columns (columns 4 to 19). The reason why this has happened is that the algorithm actually has not converged for the constraint ES-MASK, even though the cost is converging. In fact, our tests show that the norm of $\hat{\Lambda}$ is getting unboundedly large. This is possible because the constraint space of ES-MASK is non-compact.

For both T-NORM and R-NORM the $\hat{\Lambda}$ estimated by the algorithm is among the expected wrong solutions, both violating (D3). Looking at Fig. 6.10(c), it is obvious that for R-NORM we have

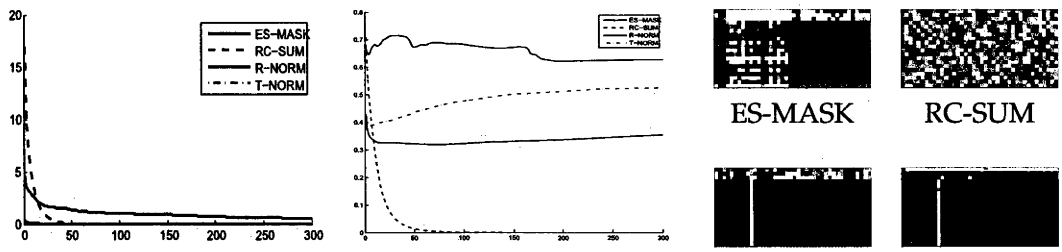
$$I \cup J = \{1,2,3,14\},$$

and thus, $K = \{4,\ldots,13\} \cup \{15,\ldots,20\}$. From the condition $3|I|+2|J| < r = 10$ it is only possible to have $|I| = 1, |J| = 3$. Thus, By Lemma 4.7 we must have the situation where $\hat{\mathrm{P}}^K = \mathrm{stack}(\{\hat{\mathrm{P}}_i\}_{i\in K}) \in \mathbb{R}^{48\times 10}$ has rank $r' = r - (3|I|+2|J|) = 10-(3+6) = 1$. The singular values of $\hat{\Lambda}$ obtained after 2000 iterations confirms this:

$$2.0, 0.0002, 0.00013, 0.00013, 0.00009, 0.00008, 0.00007, 0.00006, 0.00005, 0.00004$$

By looking at the rows of $\hat{\Lambda}$ shown in Fig. 6.10(c) for T-NORM we can conclude that for this case $I \cup J = \{1,2,3\}$. From the condition $3|I|+2|J| < r = 10$, three cases are possible, which are listed below along with the corresponding $r' = r - (3|I| + 2|J|)$:

1. $|I| = 3, |J| = 0, r' = 1$,



(a) Convergence (b) Depth Error (c) Estimated depth matrix $\hat{\Lambda}$

Figure 6.10: One run of our experiments for projections $\mathbb{P}^9 \rightarrow \mathbb{P}^2$. (a) shows that for all cases the costs are converging to zero. (b) shows that only RC-SUM has converged to a correct solution. (c) shows that R-NORM and T-NORM have converged to two different types of the wrong solutions violating (D3). Our tests show that for the constraint ES-MASK the algorithm does not converge (in terms of finding $\hat{\Lambda}$), even though the cost is converging to zero.

2. $|I| = 2, |J| = 1, r' = 2$,
3. $|I| = 1, |J| = 2, r' = 3$.

To see which case have happened, we can use Lemma 4.7, suggesting $\hat{\mathrm{P}}^K = \mathrm{stack}(\hat{\mathrm{P}}_4, \ldots, \hat{\mathrm{P}}_{20})$ has rank $r' = r - (3|I| + 2|J|)$ when $\hat{\mathrm{P}}$ has full column rank (which is the case here according to our test). Now, the singular values of $\hat{\mathrm{P}}^K$ after 2000 iterations are

$$2.0, 1^{\times 10^{-8}}, 1^{\times 10^{-8}}, 1^{\times 10^{-8}}, 4^{\times 10^{-9}}, 4^{\times 10^{-10}}, 3^{\times 10^{-10}}, 1^{\times 10^{-10}}, 5^{\times 10^{-11}}, 5^{\times 10^{-11}}.$$

This clearly suggests that $r' = \mathcal{R}\mathrm{ank}(\hat{\mathrm{P}}^K) = 1$. Therefore, from the three cases listed above, the first one holds here, that is $|I| = 3, |J| = 0$.

6.4 Summary

We ran experiments separately for 3D to 2D projections and higher-dimensional projections. For 3D to 2D, by conducting a projective factorization algorithm for both synthetic and real data, we demonstrated how the degenerate cross-shaped solutions can happen, and how the use of proper constraints can prevent them from happening. For higher-dimensional projections we ran numerical simulations testing the algorithm for two cases of projections $\mathbb{P}^4 \rightarrow \mathbb{P}^2$ and $\mathbb{P}^9 \rightarrow \mathbb{P}^2$. In each case, we showed how different types of degenerate solutions classified by our theory can happen.

Conclusion

7.1 Summary and Major Results

We extended the theory of projective reconstruction for the case of 3D to 2D projections as well as arbitrary dimensional projections. The purpose was to provide tools for the analysis of projective reconstruction algorithms, such as projective factorization and bundle adjustment, which seek to directly solve the projection equations for projection matrices and high-dimensional points.

In the case of 3D to 2D projections, we proved a more general version of the projective reconstruction theorem, which is well suited to the choice and analysis of depth constraints for factorization-based projective reconstruction algorithms. The main result was that the false solutions to the factorization problem $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathrm{P}}\,\hat{\mathrm{X}}$, are restricted to the cases where $\hat{\Lambda}$ has zero rows or zero columns, and also, when it has a cross-shaped structure. Any solution which does not fall in any of these classes is a correct solution, equal to the true setup of camera matrices and scene points up to projectivity.

We demonstrated how our theoretical results can be used for the analysis of existing depth constraints used for the factorization-based algorithms and also for the design of new types of depth constraints. Amongst other results, we presented a new class of linear equality constraints which are able to rule out all the degenerate false solutions. Our experiments also showed that choosing a good initial solution can result in finding the correct depths, even with some of the constraints that do not completely rule out all the false solutions.

Next, we investigated the more general problem of projective reconstruction for multiple projections from an arbitrary dimensional space $\mathbb{P}^{r-1}$ to lower dimensional spaces $\mathbb{P}^{s_i-1}$. We obtained the following results for a generic setup with sufficient number of projection matrices and high-dimensional points:

- The multi-view (Grassmann) tensor obtained from the image points $\mathbf{x}_{ij}$ is unique up to a scaling factor.
- Any solution to the set of equations $\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathrm{P}}_i\hat{\mathbf{X}}_j$ is projectively equivalent to the true setup, if the $\hat{\mathrm{P}}_i$-s and $\hat{\mathbf{X}}_j$-s are nonzero and $\hat{\mathrm{P}} = \mathrm{stack}(\hat{\mathrm{P}}_1, \ldots, \hat{\mathrm{P}}_m)$ has a non-singular $r \times r$ submatrix created by choosing strictly fewer than s_i rows from each $\hat{\mathrm{P}}_i \in \mathbb{R}^{s_i \times r}$.

- Any solution to the set of equations $\hat{\lambda}_{ij}\mathbf{x}_{ij} = \hat{\mathbf{P}}_i\hat{\mathbf{X}}_j$ is projectively equivalent to the true setup if $\hat{\lambda}_{ij} \neq 0$ for all i, j.
- False solutions to the projective factorization problem $\hat{\Lambda} \odot [\mathbf{x}_{ij}] = \hat{\mathbf{P}}\hat{\mathbf{X}}$, where elements of $\hat{\Lambda} = [\hat{\lambda}_{ij}]$ are allowed to be zero, can be much more complex than in the case of projections $\mathbb{P}^3 \rightarrow \mathbb{P}^2$, as demonstrated theoretically in Sect. 4.4 and experimentally in Sect. 6.3.

7.2 Future Work

The current work can be extended in many ways. For example, here it has been assumed that all points are visible in all views. A very important extension is therefore considering the case of incomplete image data. Notice that dealing with this problem is harder than the case of zero estimated projective depths $\hat{\lambda}_{ij}$, because knowing $\hat{\lambda}_{ij} = 0$ implies that the estimated scene point $\hat{\mathbf{X}}_j$ is in the null space of the estimated camera matrix $\hat{\mathbf{P}}_i$. This is while a missing image point $\mathbf{x}_{ij}$ provides no information at all. Another assumption here was that the image data is not contaminated with noise. Giving theoretically guaranteed results for the case of noisy data is another major issue which needs to be addressed in future work.

Another follow-up of this work is the study of the convergence of specific factorization-based algorithms for each of the constraints, and the design of constraints with desirable convergence properties. For example, we know that certain convergence properties can be proved for certain algorithms when the sequence of iterative solutions lie in a compact set. However, guaranteed convergence to a global minimum is still an unsolved problem. Another interesting problem is to find compact constraints which can be efficiently implemented with the factorization based algorithms, give a descent move at every iteration, and are able to rule out all the false solutions, at least for 3D to 2D projections. A partial solution to this problem has been given in Sect. 3.3.1.4, where we introduced a compact constraint with all these desired properties, except that it only rules out most cases of wrong solutions. Finding such constraints which can exclude all possible wrong solutions is still an unanswered problem.

For the case of arbitrary dimensional projections we obtained our results assuming a generic configuration of projection matrices and high-dimensional points, without specifying the corresponding generic set clearly in geometric terms. Therefore, it would be useful to compile a simplified list of all the required generic properties needed for the proof of projective reconstruction. This is because, in almost all applications (motion segmentation, nonrigid shape recovery, etc.) the projection matrices and points have a special structure, meaning they are members of a nongeneric set. It is now a nontrivial question whether the restriction of the genericity conditions to this nongeneric set is relatively generic.

Bibliography

AGARWAL, P. K. AND MUSTAFA, N. H., 2004. k-means projective clustering. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '04 (Paris, France, 2004), 155–165. ACM, New York, NY, USA. doi:http://doi.acm.org/10.1145/1055558.1055581. http://doi.acm.org/10.1145/1055558.1055581. (cited on page 106)

AGARWAL, S.; SNAVELY, N.; SEITZ, S. M.; AND SZELISKI, R., 2010. Bundle adjustment in the large. In *Proceedings of the 11th European Conference on Computer Vision: Part II*, ECCV'10 (Heraklion, Crete, Greece, 2010), 29–42. Springer-Verlag, Berlin, Heidelberg. http://dl.acm.org/citation.cfm?id=1888028.1888032. (cited on page 13)

ANGST, R. AND POLLEFEYS, M., 2013. Multilinear factorizations for multi-camera rigid structure from motion problems. *International Journal of Computer Vision*, 103, 2 (2013), 240–266. (cited on pages 24, 116, and 118)

ANGST, R.; ZACH, C.; AND POLLEFEYS, M., 2011. The generalized trace-norm and its application to structure-from-motion problems. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2502 –2509. doi:10.1109/ICCV.2011.6126536. (cited on pages 17 and 51)

BRADLEY, P. S. AND MANGASARIAN, O. L., 2000. k-plane clustering. *J. of Global Optimization*, 16 (January 2000), 23–32. doi:10.1023/A:1008324625522. http://portal.acm.org/citation.cfm?id=596077.596262. (cited on page 106)

BREGLER, C.; HERTZMANN, A.; AND BIERMANN, H., 2000. Recovering non-rigid 3d shape from image streams. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, 690–696 vol.2. doi:10.1109/CVPR.2000.854941. (cited on page 115)

BUCHANAN, T., 1988. The twisted cubic and camera calibration. *Computer Vision, Graphics, and Image Processing*, 42, 1 (1988), 130–132. (cited on page 32)

COSTEIRA, J. P. AND KANADE, T., 1998. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29 (1998), 159–179. http://dx.doi.org/10.1023/A:1008000628999. 10.1023/A:1008000628999. (cited on pages 105 and 106)

DAI, Y.; LI, H.; AND HE, M., 2010. Element-wise factorization for n-view projective reconstruction. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10 (Heraklion, Crete, Greece, 2010), 396–409. Springer-Verlag, Berlin,

Heidelberg. http://dl.acm.org/citation.cfm?id=1888089.1888119. (cited on pages 16, 17, 21, and 51)

DAI, Y.; LI, H.; AND HE, M., 2013. Projective multi-view structure and motion from element-wise factorization. *PAMI*, PP, 99 (2013), 1–1. doi:10.1109/TPAMI.2013.20. (cited on pages 2, 16, 17, 21, 51, and 55)

ELHAMIFAR, E. AND VIDAL, R., 2009. Sparse subspace clustering. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (Miami, FL, June 2009), 2790–2797. IEEE. doi:10.1109/CVPRW.2009.5206547. http://dx.doi.org/10.1109/CVPRW.2009.5206547. (cited on pages 107 and 108)

FAUGERAS, O. D., 1992. What can be seen in three dimensions with an uncalibrated stereo rig. In *Proceedings of the Second European Conference on Computer Vision*, ECCV '92, 563–578. Springer-Verlag, London, UK, UK. http://dl.acm.org/citation.cfm?id=645305.648717. (cited on page 11)

FAVARO, P.; VIDAL, R.; AND RAVICHANDRAN, A., 2011. A closed form solution to robust subspace estimation and clustering. In *2011 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. (cited on page 109)

FISCHLER, M. A. AND BOLLES, R. C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24, 6 (1981), 381–395. (cited on page 107)

HARTLEY, R.; GUPTA, R.; AND CHANG, T., 1992. Stereo from uncalibrated cameras. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, 761–764. doi:10.1109/CVPR.1992.223179. (cited on page 11)

HARTLEY, R. AND KAHL, F., 2007. Critical configurations for projective reconstruction from multiple views. *Int. J. Comput. Vision*, 71, 1 (Jan. 2007), 5–47. doi:10.1007/s11263-005-4796-1. http://dx.doi.org/10.1007/s11263-005-4796-1. (cited on pages 36 and 37)

HARTLEY, R. AND VIDAL, R., 2008. Perspective nonrigid shape and motion recovery. 276–289. doi:doi:10.1007/978-3-540-88682-2_22. http://dx.doi.org/10.1007/978-3-540-88682-2_22. (cited on pages 5, 22, 23, 115, and 116)

HARTLEY, R. I. AND SCHAFFALITZKY, F., 2004. Reconstruction from projections using Grassmann tensors. In *European Conference on Computer Vision*. (cited on pages 2, 5, 6, 12, 17, 19, 22, 65, 66, 67, 69, 71, 89, and 116)

HARTLEY, R. I. AND ZISSERMAN, A., 2004. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edn. (cited on pages 2, 11, 13, 15, 16, 17, 18, 20, 24, 30, 31, 32, 36, 37, 39, 43, 52, and 53)

HEINRICH, S. B. AND SNYDER, W. E., 2011. Internal constraints of the trifocal tensor. *CoRR*, abs/1103.6052 (2011). (cited on page 18)

HEYDEN, A.; BERTHILSSON, R.; AND SPARR, G., 1999. An iterative factorization method for projective structure and motion from image sequences. *Image Vision Comput.*, 17, 13 (1999), 981–991. (cited on pages 2, 5, 15, 21, and 53)

HO, J.; YANG, M.-H.; LIM, J.; LEE, K.-C.; AND KRIEGMAN, D., 2003. Clustering appearances of objects under varying illumination conditions. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1, I–11 – I–18 vol.1. doi:10.1109/CVPR.2003.1211332. (cited on page 105)

HONG, W.; WRIGHT, J.; HUANG, K.; AND MA, Y., 2005. A multiscale hybrid linear model for lossy image representation. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, 764 – 771 Vol. 1. doi:10.1109/ICCV.2005.12. (cited on page 105)

KANATANI, K., 2001. Motion segmentation by subspace separation and model selection. In *Proc. 8th Int. Conf. Comput. Vision*, 586–591. (cited on pages 105 and 106)

LIN, Z.; CHEN, M.; AND WU, L., 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Analysis*, math.OC, Technical Report UILU-ENG-09-2215 (2010), –09–2215. http://arxiv.org/abs/1009.5055. (cited on pages 17 and 21)

LIU, G.; LIN, Z.; YAN, S.; SUN, J.; YU, Y.; AND MA, Y., 2010a. Robust recovery of subspace structures by low-rank representation. *CoRR*, abs/1010.2955 (2010). (cited on page 108)

LIU, G.; LIN, Z.; AND YU, Y., 2010b. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, 663–670. (cited on page 108)

LU, L. AND VIDAL, R., 2006. Combined central and subspace clustering for computer vision applications. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06 (Pittsburgh, Pennsylvania, 2006), 593–600. ACM, New York, NY, USA. doi:http://doi.acm.org/10.1145/1143844.1143919. http://doi.acm.org/10.1145/1143844.1143919. (cited on page 105)

LUENBERGER, D. G., 1984. *Linear and Nonlinear Programming*. Addison-Wesley Publishing Company, 2nd ed. edn. (cited on page 5)

MA, Y.; DERKSEN, H.; HONG, W.; AND WRIGHT, J., 2007. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29, 9 (2007), 1546–1562. (cited on page 107)

MAHAMUD, S.; HEBERT, M.; OMORI, Y.; AND PONCE, J., 2001. Provably-convergent iterative methods for projective structure from motion. In *CVPR*, 1018–1025. (cited on pages 2, 5, 15, 16, 18, 21, 53, and 55)

OLIENSIS, J. AND HARTLEY, R., 2007. Iterative extensions of the Sturm/Triggs algorithm: convergence and nonconvergence. *PAMI*, 29, 12 (2007), 2217 – 2233. doi:doi:10.1109/TPAMI.2007.1132. http://dx.doi.org/10.1109/TPAMI.2007.1132. (cited on pages 2, 3, 13, 15, 16, and 18)

SEMPLE, J. AND KNEEBONE, G., 1952. *Algebraic Projective Geometry*. Oxford Classic Texts in the Physical Sciences Series. Clarendon Press. ISBN 9780198503637. http://books.google.com.au/books?id=qIFzkgBikEUC. (cited on pages 32 and 37)

SINKHORN, R., 1964. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35, 2 (1964), pp. 876–879. (cited on pages 15, 51, and 52)

SINKHORN, R., 1967. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74, 4 (1967), pp. 402–405. (cited on pages 15, 51, and 52)

STURM, P. F. AND TRIGGS, B., 1996. A factorization based algorithm for multi-image projective structure and motion. In *ECCV*, 709–720. http://dl.acm.org/citation.cfm?id=645310.649025. (cited on pages 2, 14, and 18)

TIPPING, M. E. AND BISHOP, C. M., 1999. Mixtures of probabilistic principal component analyzers. *Neural Comput.*, 11 (February 1999), 443–482. doi:10.1162/089976699300016728. http://portal.acm.org/citation.cfm?id=309394.309427. (cited on page 106)

TRIGGS, B., 1996. Factorization methods for projective structure and motion. In *CVPR*, 845–. http://dl.acm.org/citation.cfm?id=794190.794634. (cited on pages 2, 15, 18, 52, and 53)

TRIGGS, B.; MCLAUCHLAN, P. F.; HARTLEY, R. I.; AND FITZGIBBON, A. W., 2000. Bundle adjustment - a modern synthesis. In *ICCV Proceedings of the International Workshop on Vision Algorithms*, 298–372. (cited on pages 2 and 18)

TSENG, P., 2000. Nearest q-flat to mpoints. *J. Optim. Theory Appl.*, 105 (April 2000), 249–252. doi:10.1023/A:1004678431677. http://portal.acm.org/citation.cfm?id=345260.345322. (cited on page 106)

UESHIBA, T. AND TOMITA, F., 1998. A factorization method for projective and euclidean reconstruction from multiple perspective views via iterative depth estimation. *Computer*, I (1998), 296–310. http://www.springerlink.com/index/vcxuej3m7d300f4d.pdf. (cited on pages 2, 15, 16, 21, and 56)

VIDAL, R., 2011. Subspace clustering. *Signal Processing Magazine, IEEE*, 28, 2 (march 2011), 52–68. doi:10.1109/MSP.2010.939739. (cited on pages 106 and 107)

VIDAL, R. AND ABRETSKE, D., 2006. Nonrigid shape and motion from multiple perspective views. In *ECCV*, vol. 3952 of *Lecture Notes in Computer Science*, 205–218. Springer. (cited on pages 5, 21, and 115)

VIDAL, R.; MA, Y.; AND SASTRY, S., 2005. Generalized principal component analysis (gpca). *IEEE Trans. Pattern Anal. Mach. Intell.*, 27, 12 (2005), 1945–1959. (cited on page 106)

VIDAL, R.; TRON, R.; AND HARTLEY, R., 2008. Multiframe motion segmentation with missing data using powerfactorization and gpca. *Int. J. Comput. Vision*, 79 (August 2008), 85–105. doi:10.1007/s11263-007-0099-z. http://portal.acm.org/citation.cfm?id=1363334.1363356. (cited on pages 104, 105, and 106)

WOLF, L. AND SHASHUA, A., 2002. On projection matrices $\mathbb{P}^k \to \mathbb{P}^2$, $k = 3, \ldots, 6$, and their applications in computer vision. *IJCV*, 48, 1 (2002), 53–67. (cited on pages 5, 21, 23, and 110)

XIAO, J. AND KANADE, T., 2005. Uncalibrated perspective reconstruction of deformable structures. In *Tenth IEEE International Conference on Computer Vision (ICCV '05)*, vol. 2, 1075 - 1082. (cited on pages 5, 6, 21, and 115)

YANG, A. Y.; WRIGHT, J.; MA, Y.; AND SASTRY, S. S., 2008. Unsupervised segmentation of natural images via lossy data compression. *Comput. Vis. Image Underst.*, 110 (May 2008), 212–225. doi:10.1016/j.cviu.2007.07.005. http://portal.acm.org/citation.cfm?id=1363359.1363381. (cited on page 105)

YANG, J. AND YUAN, X., 2013. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Math. Comp.*, 82, 281 (2013), 301–329. doi:10.1090/S0025-5718-2012-02598-1. http://dx.doi.org/10.1090/S0025-5718-2012-02598-1. (cited on pages 17 and 21)

ZANGWILL, W., 1969. *Nonlinear programming: a unified approach*. Prentice-Hall international series in management. Prentice-Hall. http://books.google.com.au/books?id=TWhxLcApH9sC. (cited on page 5)

ZELNIK-MANOR, L. AND IRANI, M., 2003. Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, II - 287–93 vol.2. doi:10.1109/CVPR.2003.1211482. (cited on page 105)