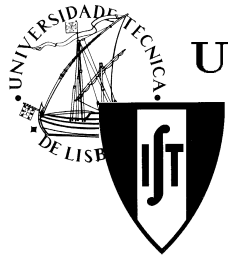


UNIVERSIDADE TÉCNICA DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

A MULTI-BODY FACTORIZATION METHOD FOR MOTION ANALYSIS

JOÃO PAULO SALGADO ARRISCADO COSTEIRA

(Mestre)

Tese para obtenção do grau de doutor em
Engenharia Electrotécnica e de Computadores

Lisboa, Maio de 1995

Tese realizada sob a orientação de

João José dos Santos Sentieiro

Professor Catedrático do
Departamento de Engenharia Electrotécnica e de Computadores
INSTITUTO SUPERIOR TÉCNICO

À Ana Paula

ACKNOWLEDGEMENTS

I had the great and demanding experience of having two advisors. I owe Professors Takeo Kanade and João Sentieiro a great deal of patience, dedication and ultimately friendship. From Prof. Kanade I learned the rare ability of searching for what is fundamental behind what looks like a good idea; and to pursuit with patience. From Prof. Sentieiro I got the meaning and reward of a lifetime commitment to an ideal and to people. My only possible tribute to them is to pass on their lessons to others.

To Martial Hebert for the fruitful discussions and suggestions. I wish we had deepened further the cooperation. My fault!

To José Moura for listening and criticizing my ideas. His thoughtfulness and rigor have been always a reference. His friendship will be missed.

To Maria João Rendas. I didn't use clustering techniques but the orthogonal projectors were precious.

To José Bioucas the prompt answers and clever suggestions we exchanged over the 'cyberspace'.

To Carnegie Mellon University and the Robotics Institute, for having me as a visiting student for more than 3 years, and for making me feel at home. In Carnegie Mellon I found the joy for knowledge, the pleasure of discovery and the work ethics which will follow me for as long as I live.

To JNICT who partially funded my stay in Pittsburgh.

To IST to which I have a strong and emotional attachment for also partially funding my stay.

Whatever I accomplished, I owe it to my friend João José. I owe him ten years of friendship and dedication. He was also the one who 'pushed' me to CMU (I didn't want to leave Portugal!). I'll never be grateful enough for that.

Because a PhD is not only about science, and because life outside the office is as important as inside, my deep gratitude to all the friends whose company was precious. Among an endless count I wish to thank: Carlos Bispo, my undergraduate classmate, my 'chief cook' in London, colleague in Lisbon, all-time-friend and babysitter in Pittsburgh. I guess this makes us inseparable. Margarida Jácome, my delightful and close friend. As friend and as a peer she showed me the meaning of the words "high standards". My dears Lynn, Mona, Maged and Robert with whom I shared my last tears in Pittsburgh, Maria Joao and Richard in whose house Beatriz had her first 'dinner out', and Raj Reddy, Ana Barroso, João Silva, Lino Santos and Manuela Veloso with whom I spent most of the time. Also, Radú and Takumi Jasinschi who became friends-at-first-sight. Jim and Dorothy Rehg, whom I regret not having spent more time with. My officemates Juan Leon, Amy Zaremsky and Aarti Gupta for their daily friendship. My last four months in Pittsburgh were spent without my family. Instead of a predictably painful period, it is unforgetable; I shared with Lynn the joy, the anger, the noise and the silence that only a warm and great friend can give. Likewise, António Alonso was my companion of far too many hours of intellectual intimacy and care. They all made my years in Pittsburgh one of the best and most intense periods of my life.

To Arabica I thank for making some of these friendships possible, and for the long nights spent with coffee and the music of Madredeus, Zeca Afonso, Amália Rodrigues, Fausto and others. Also, by turning the café into a non-smoking place they gave me the opportunity to enjoy the beautiful Pittsburgh weather!

To my inlaws Olímpia and Teotónio who absorbed most of the shock of my return. Writing a thesis and handling an N-degree of freedom articulated one year old with a 100dB built-in loudspeaker are not compatible tasks. They helped to make it possible.

To Ana Paula whom I don't even dare to thank.

AGRADECIMENTOS

Passei pela gratificante e exigente experiência de ter tido dois orientadores. Aos Professores Takeo Kanade e João Sentieiro expresso aqui a minha gratidão pela paciência e dedicação de que fui alvo, reveladoras de uma grande amizade. Em ciência, frequentemente, o importante é a pergunta. O Prof. Kanade ensinou-me o difícil processo da descoberta daquilo que é fundamental numa pergunta e da procura paciente da resposta. O Prof. Sentieiro mostrou-me que os sonhos têm nomes de pessoas e duram uma vida.

Ao Martial Hebert pela disponibilidade e inúmeras sugestões. Lamento que a colaboração não tivesse sido, no entanto, mais profunda. *Mea culpa.*

Ao José Moura por ter tido a paciência e constante disponibilidade de me ouvir e criticar. O seu talento e rigor são, para mim, uma referência. Espero um dia poder, novamente, disfrutar da sua amizade no dia-a-dia. Até lá resta-me recordá-la.

À Maria João Rendas. Afinal o problema não era de "clustering" mas os operadores de projecção ortogonal eram a solução.

Ao José Bioucas pela interação, inteligência e prontidão das respostas. O cyber-diálogo que mantivemos durante a minha estadia foi-me gratificante.

À Carnegie Mellon University, por me aceitar como estudante visitante por 3 anos e pela sua hospitalidade. Ali aprendi o prazer do conhecimento, a alegria da descoberta, e acima de tudo a ética de trabalho que, espero, me guie pela vida fora.

À JNICT pelo financimento parcial da minha estadia em Pittsburgh.

Ao Instituto Superior Técnico, instituição que representa para mim muito mais do que um local de trabalho e com a qual tenho uma forte ligacção emocional, agradeco também o financiamento parcial da minha estadia.

Quaisquer que tenham sido os meus conseguimentos, devo-os ao meu amigo João José, companheiro de mais de dez anos de amizade e dedicação. Além disso, a ele devo o "empurrão" que me levou à CMU. Também por isso, ficar-lhe-ei eternamente grato.

Um doutoramento é uma experiência que vai muito para além da ciência e das quatro paredes de um gabinete. O ónus de gratidão para com os inúmeros amigos que me acompanharam neste capítulo importante da minha vida é enorme. Em particular, gostaria de agradecer ao Carlos Bispo, meu colega de licenciatura, "professor de culinária" em Londres, colega assistente em Lisboa e all-time-friend e babysitter em Pittsburgh. Esta diáspora tornou-nos inseparáveis. À Margarida Jácome, o prazer da profunda e íntima amizade que a distância apenas reforça. Ao meus queridos amigos

Lynn, Mona, Maged e Robert, com quem partilhei as minhas últimas lágrimas em Pittsburgh, à Maria João e Richard em cuja casa a Beatriz teve o seu primeiro 'jantar fora', juntamente com o Raj Reddy, a Ana Barroso, o João Silva, o Lino Santos, e a Manuela Veloso, com os quais passei a maior parte do tempo. Ao Radú e Takumi Jasinschi que foram amigos à primeira vista. Ao Jim e Dorothy Rehg de quem lamento nao ter disfrutado mais da sua companhia. Aos meus companheiros de gabinete Juan Leon, Amy Zaremsky e Aarti Gupta pela amizade diária. Os meus últimos quatro meses em Pittsburgh foram passados sem a família. Em vez da predizível tortura foi um período inesquecível; Com a Lynn partilhei os momentos de alegria e tristeza, a algazarra e o silêncio que apenas uma forte e carinhosa amizade consegue dar. Do mesmo modo, o António Alonso foi o meu companheiro de longas horas de intimidade intelectual e carinho. Todos eles fizeram dos anos passados em Pittsburgh um dos mais ricos e melhores períodos da minha vida.

Ao café Arabica por ter feito algumas destas amizades possíveis, e pela disponibilidade do seu tocador de CD's que nos deleitou em longas noites com a música dos Madredeus, Zeca Afonso, Amália Rodrigues, Fausto e outros. Além disso, a sua fúria antí-fumadora proporcionou-me o prazer de disfrutar do maravilhoso e ameno clima de Pittsburgh.

Aos meus sogros Olímpia e Teotónio, que absoveram grande parte do choque da volta a casa. Acabar de escrever uma tese, e ao mesmo tempo lidar com um 'robozinho' altamente articulado e com um altifalante de 100dB são tarefas normalmente incompatíveis, mas que eles ajudaram a tornar possível.

À Ana Paula a quem não me atrevo sequer agradecer.

Resumo

No campo da Visão por Computador, o problema da determinação da estrutura tridimensional do mundo em cenas dinâmicas tem sido um dos mais extensivamente estudados. Não obstante, a vasta maioria do trabalho publicado assume que a cena é composta por apenas um só objecto. O caso mais complexo e realista, de um número arbitrário de objectos presentes na cena tem sido alvo de pouca atenção, nomeadamente, no que refere aos seus aspectos teóricos. Nesta tese propõe-se um novo método que separa e determina a forma tridimensional dos objectos, partindo de uma sequência de imagens captadas em cenas onde múltiplos objectos se movem independentemente. O método não requer conhecimento prévio do número de objectos, e tão pouco depende de nenhum processo local de agrupamento de pontos característicos ao nível da imagem. Neste âmbito, introduz-se uma entidade matemática, denominada *matriz de interacção de forma*, que exibe interessantes propriedades de invariância, nomeadamente relativamente ao movimento dos objectos, à característica da sua matriz de forma e à selcção do sistema de eixos escolhidos para representar a mesma. Esta estrutura invariante é calculável a partir, apenas, da trajectória na imagem dos pontos característicos dos objectos. A análise de cenas dinâmicas com multíplos objectos em movimento assenta na representação decorrente da decomposição em valores/vectores singulares da matriz de observações, definida pelas coordenadas das trajectórias dos referidos pontos característicos. Na tese mostra-se, ainda, que em cenas com um único objecto, estas estruturas algébricas contêm informação geométrica sobre a forma e o movimento do referido objecto. O potencial do método é também ilustrado numa sequência de experiências com diferentes graus de complexidade e nível de ruído. Numa experiência com dados sintetizados mostra-se que a matriz de interacção de forma contém toda a informação necessária à segmentação de objectos transparentes e com forma degenerada (rectas e planos). A robustez ao ruído é tambem tratada e ilustrada em experiências realizadas com imagens reais. Finalmente, o problema detecção da característica da matriz de observações é tratado através da integração das medidas de incerteza do processo de seguimento dos pontos característicos.

Palavras Chave: Forma a partir do movimento, Invariantes,

Abstract

The structure-from-motion problem has been extensively studied in the field of computer vision. Yet, the bulk of the existing work assumes that the scene contains only a single moving object. The more realistic case where an unknown number of objects move in the scene has received little attention, especially regarding the theoretical treatment. In the thesis we present a new method for separating and recovering the motion and shape of multiple independently moving objects in a sequence of images. The method does not require prior knowledge of the number of objects, nor is dependent on any grouping of features into an object at the image level. For this purpose, we introduce a mathematical construct of object shapes, called the shape interaction matrix, which is invariant to both the object motions, shape rank, and selection of coordinate systems. This invariant structure is computable solely from the observed trajectories of image features without grouping them into individual objects. Once the structure is computed, it allows for segmenting features into objects by the process of transforming it into a canonical form, as well as recovering the shape and motion of each object.

The multiple motion analysis presented in this thesis is supported on the singular value/vector representation of the feature trajectories. We show that these linear algebraic structures convey important geometric information about the shape and motion of the objects in single body scenes. The potential of the method is illustrated in a set of three experiments of different degrees of scene complexity and level of noise. In a synthetic experiment we show that the shape interaction matrix contains all the information necessary to solve scenes with transparent objects, objects with degenerate shape (planes and lines). The robustness of the method, in noisy conditions, is also shown in two experiments using real images. Finally, the rank detection problem is addressed by integrating the feature tracking uncertainty into the rank detection procedure.

Keywords: Shape from Motion, Motion Analysis, Invariants

Contents

List of Figures

Chapter 1

Introduction

Since the early days of computers, the scientific community has been working towards providing robots with sensing capabilities. Vision is the most promising and yet the most challenging of the artificial senses. From images, biological systems collect most of the information they need to manipulate, navigate and interact with the environment, suggesting that visual sensors are fundamental to providing artificial systems with perception capabilities.

Recovering the 3D structure of a dynamic scene from an image sequence is one of the most extensively studied problems in Computer Vision. This is because not only do objects and robots themselves move, but also motion provides important cues about 3D geometry. While a large amount of literature exists about this structure-from-motion problem, most previous theoretical work is based on the assumption that only a single motion is present in the image sequence; either the environment is static and the observer moves, or the observer is static and only one object in the scene is moving. More difficult, less studied, and of more practical interest is the general case of an unknown number of objects moving independently in the scene. If a set of features has been identified and tracked in an image sequence, but the correspondence between each feature and the respective object is not known, the question is whether the motion and shape of the multiple objects contained in the image sequence can be segmented and recovered.

Previous approaches to the structure-from-motion problem for multiple objects can

be grouped into two classes: image motion-based (2D) and three-dimensional (3D) modeling. The image-motion based approach relies mostly on the spatio-temporal properties of an image sequence. Regions corresponding to different velocity fields are extracted using Fourier domain analysis [WA83, AB85] or scale-space and space-time filters [BBHP90, BHK91, IBP94, JRS92]. These image-based methods have limited applicability either because object motions are restricted to a certain type, such as translation only, or because image-level properties, such as locality, need to be used for segmentation without assuring consistent segmentation into 3D objects.

To overcome these limitations, models of the motion and the scene can be introduced that provide more constraints. Representative constraints include rigidity of an object [Ull83] and smoothness (or similarity) of motion [Sin93, NZ92, Adi85, BB91]. Then the problem is reduced to segmenting image events, such as feature trajectories, into objects so that the recovered motion and shape satisfy these constraints; that is, it becomes a clustering problem with constraints derived from a physical model. Though sound in theory, the practical difficulty lies in the cyclic dilemma: to check the constraints it is necessary to segment features, and to perform this segmentation requires the computation of the constraints. So, current methods tend to be of a "generate-and-test" nature, or require prior knowledge of the number of objects (clusters). In [Ull83] Ullman describes a computational scheme to recursively recover shape from the tracks of image features. A model of the object's shape is matched with the current position of the features, and a new model that maximizes rigidity is computed, that is, the updated shape is the rigid object that best fits the scene data. He suggests that this scheme could be used to segment multibody scenes by local application of the rigidity principle. Since a single rigid body model does not fit the data, collections of points that could be explained by a rigid transformation would be searched and grouped into an object. Although in a different context, this view of the problem is followed in [BB91] and [Gea94] under the framework of the factorization method [TK90b], where the role of rigidity is replaced by linear dependence between feature tracks. Since the

factorization produces a matrix that is related to shape, segmentation is obtained by recursively clustering columns of feature trajectories into linearly dependent groups.

This thesis presents a new method for segmenting and recovering the motion and shape of multiple independently moving objects, from a set of feature trajectories tracked in a sequence of images. The method, at the image level, does not require any grouping of features into an object or prior knowledge of the number of objects. It directly computes shape information and allows segmentation into objects. This has been made possible by introducing a liner-algebraic construct of object shapes, called the *shape interaction matrix*. The entries in this matrix are invariant to individual object motions and yet are computable only from tracked feature trajectories without segmentation. Once the matrix is computed, transforming it into the canonical form results in segmenting features as well as recovering the shape and motion of each object. We have developed our theory by using the factorization method by Tomasi and Kanade [TK90b] as a way to relate feature measurements to motion and shape with an orthographic camera. It is, however, easily seen that the theory, and thus the method, work under a broader projection model including scaled orthography and paraperspective [PK93] up to an affine camera [KvD91].

1.1 Multiple Motion Segmentation: An Overview

Most contributions to the area of structure from motion rely on the assumption that the essence of multiple motion analysis comes down to a classification procedure by which image patterns, or image features, are grouped by velocity in "classes" of distinct motions. The approaches used by researchers that have addressed this problem can be grouped into two major classes: image motion-based (2D) approaches and three dimensional (3D) modeling approaches.

1.1.1 Image-based Analysis

The first class of methods addresses the problem of multiple motion and transparency, relying mostly on spatiotemporal representations of image sequences, and uses Fourier domain analysis [WA83, AB85]. Here, the processing is done at the image level and the strategy consists of grouping image regions into "classes" of image velocity similarity (also known as optical flow).

In general, the spatiotemporal representation of an image sequence is a three-variable function $I(x, y, t)$. This representation has too many degrees of freedom to be of any use in finding a unique solution for the optical flow. To overcome this difficulty, some constraints are usually imposed on the allowable motions [HS81, Hil84, Hee88a]. Most of the proposed methodologies consider that the image motion is generated by translating objects. This assumption constrains $I(x, y, t)$ to be a function of the original texture, by defining the image at instant t as

$$I(x, y, t) = I_0(x_0 - v_x t, y_0 - v_y t), \tag{1.1}$$

where I_0 is the first image of the sequence and (v_x, v_y) is the velocity of the pixel at position (x_0, y_0). Computing the Fourier transform of both sides of (1.1) yields the relation (in the frequency domain):

$$\tilde{I}(\omega_x, \omega_y, \omega_t) = \tilde{I}_0(\omega_x, \omega_y)\ \delta(\omega_t - v_x\omega_x - v_y\omega_y) \tag{1.2}$$

where $\tilde{I}$ represents the transform of I, $\delta()$ the Dirac's impulse function and $\omega_x, \omega_y, \omega_t$ the spatial and temporal frequencies respectively. Equation (1.2) reveals that image sequences have all their energy distributed along the plane $\omega_t = v_x\omega_x + v_y\omega_y$ in the space-time frequency domain. The plane's slope corresponds to the image velocity and the spectrum in this plane is that of the first image.

In the case of more than one motion in the image, the total spectrum will be the sum of as many planes as thre are different motions. To illustrate the procedure better, assume that the image has one spatial dimension only. Then, instead of a plane, the Fourier transform of the sequence will be constrained to a line in the spatiotemporal frequency:

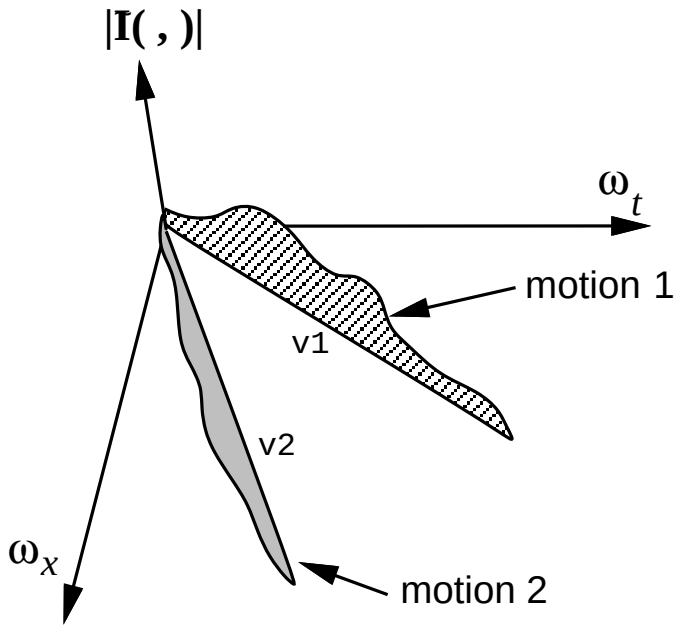


Figure 1.1: Absolute value of the Fourier transform of an image sequence with two objects moving. The spectrum lies along two lines correspondent to each of the motions

The Fourier transform consists of the sum of two functions taking values only along the two lines $\omega_t = v_{1(2)}\omega_x$. Since the spectrum of the moving pattern spreads over different frequency regions, these can be separated by a selective filter. The selective filtering is usually recursive. The fast motion is identified and eliminated from the sequence, and then the procedure is repeated to the next fast motion until all pixels in

the image have been classified.

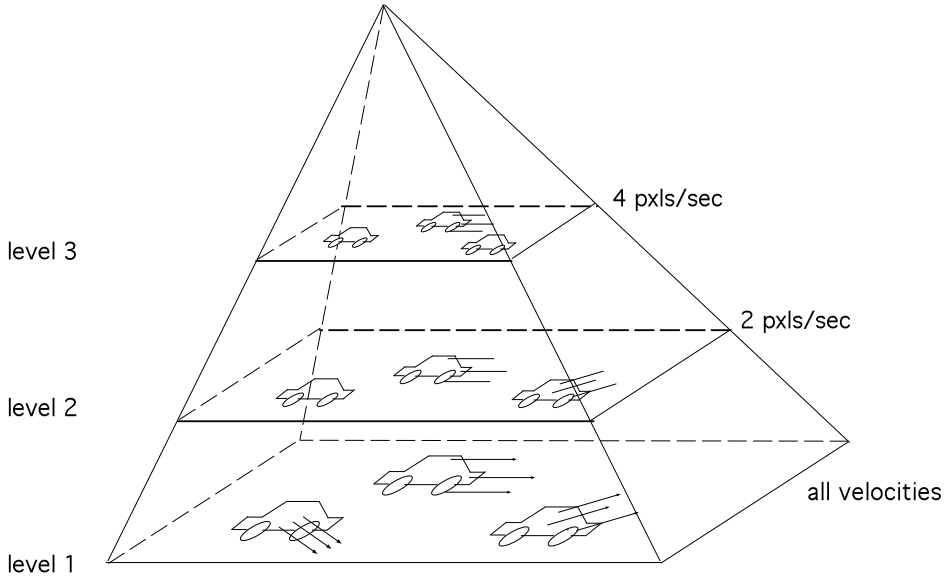


Figure 1.2: Pyramid representation of one image. Going up in the pyramid eliminates image motions.

This general approach is applied by [BBHP90] to segment two motions in the image using three frames. The authors use to a multi-resolution pyramid [Ros84] to represent the image signal and to compute image motion at each level. Forming a (Gaussian) pyramid involves, at each resolution level, a spatial lowpass filtering of the image and a subsampling.

Consider Figure 1.2 where a pyramid is shown with several image patterns moving with different velocities. If two consecutive frames are subsampled, choosing for the next resolution level every other pixel of the current resolution level, only displacements larger than one pixel will be detected. This can be understood as a spatial lowpass and temporal high-pass filtering that discards slow moving patterns. In terms of Fourier representation, the pyramid is selecting the sequence spectrum in a zone equivalent to the shaded area of Figure 1.3, which has high time (ω_t) and low space (ω_x) frequency components. As we can see, there is only one signal with non-zero energy in this region. The hierarchy can go as high as necessary to eliminate all the motions but one. Therefore single motion techniques can be used. Once the pixels corresponding to the

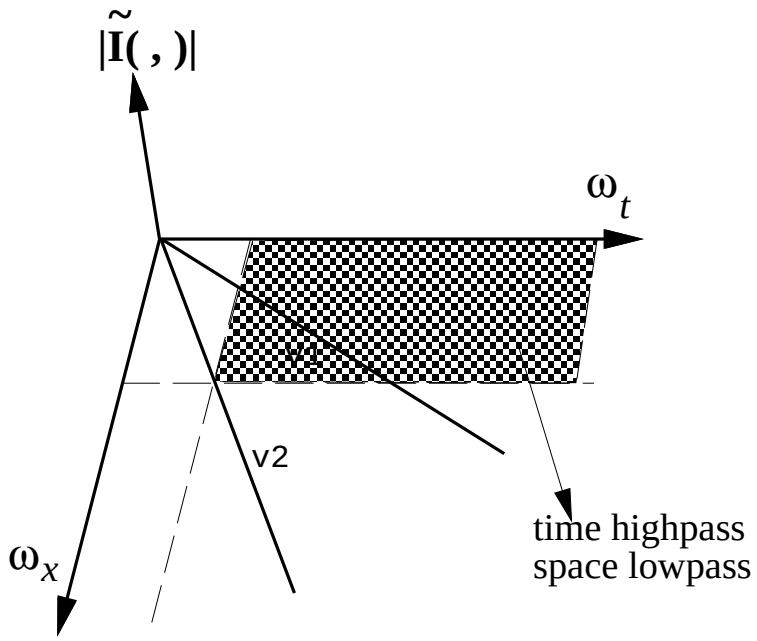


Figure 1.3: Selection of one motion in the Fourier domain. The fast motion has energy in higher regions of the spectrum. With a linear filter, represented by the shaded area, one motion can be selected

fast motion are detected this same procedure can be applied recursively to detect all motions. Variants of this technique have been used in [BHK91] and [IBP94] with some degree of success.

The importance of geometric image information was pointed out in [JRS92], where it is shown that the geometric structure of the actual image texture (*e. g.* image contours) plays an important role in the perception and classification of multiple motions. Taking into account the curvature of contours it is possible to recover multiple motions from just the normal component of the optical flow.

These methods have had some degree of success in the case of transparent scenes, since they essentially measure the energy produced by the motion, however the drawbacks are too severe. These are a consequence of both the assumption imposed on motion, which is very restrictive, and the lack of criteria for separating between the various resolution levels which actually define the motion "classes". In real cases the spectrum of an image sequence does not lie on a plane, even in trivial cases. Imagining

a textured sphere rotating, we realize that the pyramid will split velocity, components of the same physical object along all resolution levels. Points in the occluding contour have zero velocity whereas points in the mid region move fast. Using a classification of the optical flow leads to a split of regions belonging to the same object, with completely wrong results. Furthermore, if uncertainty is present in the measurements distinguishing between the filtered motions can be very hard. In [Jas92] it is shown that in the presence of error there are certain requirements for the spatial and temporal sampling rates which may (in terms of uncertainty) exclude some levels of the pyramid.

One important contribution of space-time image based methods is to show the possibility of recovering multiple motions with relatively simple mechanisms in low level vision, when motion is essentially translational.

1.1.2 3D Modeling Approaches

To overcome the limitations of the image-based methods referred to above, the relationship between the three-dimensional geometry of the scene and the image data must be modeled. In other words, the image data (e.g. feature tracks) must be related to, or parametrized by, the three-dimensional shape and/or motion coordinates.

Ever so, in scenes with multiple objects the number of possible combinations of image points into physical objects is enormous. Just to emphasize this point, note that one trivial solution can be found by considering each point of the image to be itself a single moving object. Structure from motion literature traditionally uses the rigidity of the scene as a geometric constraint to limit the possibilities, together with clustering/classification techniques to group image points into objects. Considering rigid objects, the 3D coordinates of one point in the object are motion invariant relative to other points of that same object. Using this method, we can identify approaches that compute the local shape of the object and then verify rigidity compliance, and approaches that use the linear dependence of image feature tracks.

In order to emphasize the differences between the approach presented in this thesis and thouse previouly proposed, let us consider a particular case. Figure 1.4 illustrates

a dynamic scene where two bodies move. A set of points is selected in the image and the task consists of grouping them in such a way that each group contains points of a single object. The data is the image position of these points over time. These features

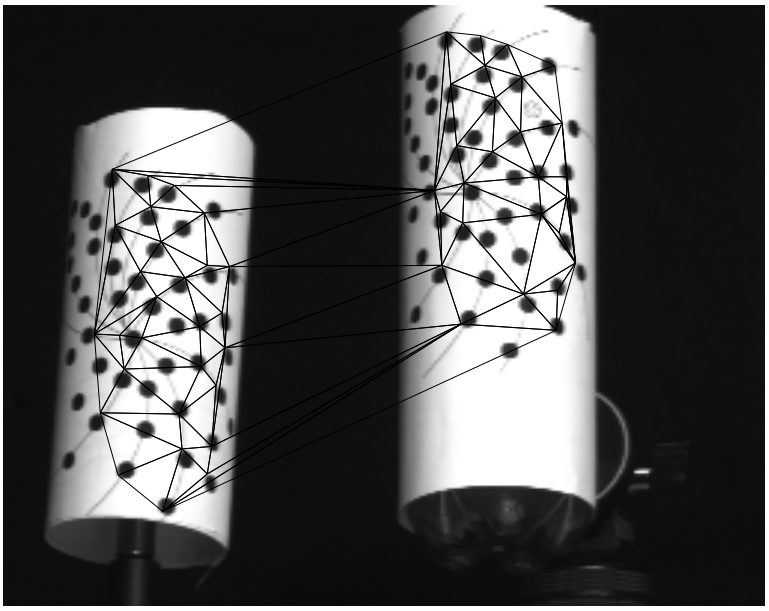


Figure 1.4: Multibody scene.

can be connected by a locality heuristic, that is, close features are more likely to be part of the same object and therefore a close link between them is established. This locality heuristic can be represented by a Delaunay triangulation, shown as a superimposed graphic in Figure 1.4[1]. The features are located at the vertices of each triangle. Since the triangulation links all the features, segmenting the scene is equivalent to grouping features into clusters representing each one of the objects.

One of the most representative examples of the first kind of approaches is that of Ullman [Ull83] who, in an early study, proposed a computational scheme for three-dimensional shape recovery. Using feature tracks as input, his algorithm computes shape by incremental updating of a 3D shape model defined *a priori*. Using the previous example, Ullman's idea comes down to a procedure that first computes the shape of every triangle, assuming it contains only features of the same body. Repeating this

[1]The triangulation is used as an example. Any other equivalent criterion is also valid.

procedure for all triangles gives a local estimate of the 3D shape. Then, if features belong to the same object, the 3D length of the triangle edges does not change over time. In other words, the object being rigid, the length of the triangle edges is invariant to rotations and translations. Then, recursively comparing neighboring triangles, the same rigidity evaluation is performed, and, if possible, the correspondent features are merged into a single cluster. The segmentation is obtained by recursively applying the procedure to the newly formed clusters until no more merging is possible. This methodology is an example of the "conviction" that multiple motion segmentation is a clustering problem, thus solutions must be worked around locality criteria.

Besides the need to set up thresholds in advance, the fundamental drawback of this approach is that in order to verify rigidity compliance, the shape must be computed, while this shape is valid only if segmentation is performed beforehand. This cyclic dilemma produces high complexity, and does not guarantee the uniqueness of the solution.

Following the same concept, other approaches indirectly use the rigidity property, by clustering the trajectories of feature points over a sequence of images. In this context, rigidity is translated into linear dependence of the feature trajectories of the same object. This is the general approach used by Boult and Brown [BB91], and Gear [Gea94]. The authors address the segmentation problem using the factorization method developed by Tomasi and Kanade [TK90b]. Unlike Ullman's rigidity maximization algorithm, the factorization method translates the idea of rigid interpretations into a global constraint of the matrix formed by the tracks of moving image features. This constraint is the rank 3 property of the measurements matrix, which, if decomposed into its singular vectors, generates all possible subspaces where feature coordinates lie.

What Bould and Brown did was to use the singular value decomposition of the measurements matrix to generate the whole space and then recursively cluster points in a rank 3 subspace, assuming that the number of objects was known. Under the same way, there are other algorithms which parametrize motion and carry out the

clustering in the parameter space[Sin93]. Different motions produce clusters in the parameter space which can be recovered using methods similar to the generalized Hough transform.

In [NZ92] and [Adi85] the more complex perspective projection model is used but the underlying methodology hinges on the same principle, that multiple body segmentation is a *classification* problem, thus utilizing the same framework as those cited above.

The drawbacks of these approaches are mainly threefold:

- First, to form clusters and to group points a distance between clusters has to be defined. Most often this distance is based on heuristics, with no consideration of the physics of the problem.

- Second, looking for clusters usually requires prior knowledge about the number of objects. Looking for clusters without knowing how many they are could be overwhelming, since combinations of points belonging to different objects generate cluster elements whose distance from a subspace is not defined. Furthermore, degenerate cases pose severe problems under this framework since the clustering has to be performed in subspaces with different dimensions.

- Finally there are the computational issues. Clustering means that decisions have to be made based on few and local data. We need to measure distances between sets of 3 points and decide whether they are "close" or not. With noisy data the distance could, in many cases, be totally meaningless.

1.2 Structure of the Thesis

This thesis addresses the problem of shape-from-motion recovery for scenes with multiple moving objects. In Chapter 1 the problem is introduced and an overview of related work in the area of multiple motion segmentation is presented. The original contributions of this thesis are emphasized. Chapter 2 revisits the single object problem, where a reformulation of the factorization method is presented. This method allows for the shape and motion computation of a single moving object from an image stream. This formulation is carried out in homogeneous coordinates, explicitly introducing the object's translational component in the motion equations.

Chapter 3 provides a geometrical interpretation of the factorization method. We will deliver physical meaning to all matrices involved in the singular vector/value representation of the shape from motion process.

The multibody problem is formally introduced in Chapter 4, and a mathematical construct called *shape interaction matrix* is defined. The first three sections of this chapter explain the main difficulties of the problem, the limitations of previous approaches, and the shape and motion invariance properties of the shape interaction matrix. Under noise-free conditions this matrix delivers the solution without any extra computation. In other words, each entry of the construct, by itself, indicates whether two image points belong to the same object. The real situation with noisy measurements is dealt with in the last section of the chapter, where a new iterative algorithm is proposed.

In Chapter 5 a set of three experiments is described. These experiments were planned in order to reveal various properties of the solution. In the first experiment we consider a synthetic scene, and show the potential of our result when segmenting a scene with three transparent objects. Experimental conditions are ideal in the sense that an orthographic camera is modeled, even though noise is included. The second experiment uses real images in a laboratory environment. The tracking is reliable and the results show robustness in real conditions. The third experiment uses images from

an outdoor scene with high levels of noise and unreliable tracking. The scene was correctly recovered despite these severe conditions.

In summary, Chapters 1 to 5 contain the main theoretical body and experimental results of this thesis, which were constructed based on some assumptions and pre-processed data. Chapter 6 explains how the assumptions can be verified in practice and also how the pre-processing is done. Specifically, we show how to compute rank properties associated with our measurements and how to track features over time. In the previous chapters these problems were assumed to be solved. In this chapter we complete a 3D computational theory of shape and motion recovery (under orthography) for multiple object scenes, using a sequence of raw images as input. Finally, in the last chapter we draw some conclusions and propose possible future developments of this work.

1.3 Original Contributions

Shape recovery from dynamic scenes is a fundamental process in reconstructing and representing the 3D geometry of the world. The more general and complex case, where several objects move, has allready been addressed by others though with some degree of restrictions and prior knowledge.

We show that segmentation of multiple motions can be achieved in a global fashion without any prior knowledge of the number of objects or their shapes. A mathematical construct of shapes called the *shape interaction matrix* is introduced. This mathematical construct reveals the global constraints of the problem and condenses all the information necessary to segment the scene, without using any local property. It is obtained directly from the image measurements, and it should be noted that among its properties is *motion and shape coordinate system invariance*. We also develop an iterative algorithm for noisy conditions, exploring the properties of the shape interaction matrix.

For the single body shape recovery problem we provide a geometric meaning to

algebraic entities which are related to the symmetry properties of objects and motion.

Chapter 2

Shape From Motion in Single Body Scenes: The Factorization Method Revisited

The factorization method was originally introduced by Tomasi and Kanade [TK90b] for the case of single object motion. The core of the method is a procedure based on singular value decomposition that separates a matrix of measurements into the product of two matrices which represent, respectively, the shape and motion of an object. The method does not need any prior assumptions about either structure or motion.

2.1 A New Formulation Including Translation

The original Tomasi-Kanade formulation addressed the case of a moving camera observing a static scene. In this chapter we will reformulate the method in order to consider the problem of a static camera observing a scene with a moving object. Also, whereas the translation component of motion is eliminated in the original method, we will retain that component in our formulation. Though equivalent for the single object case, this new formulation helps to simplify some of the representations. Furthermore, in multiple object scenes the translation component of motions cannot be computed directly from image data, so the original representation used in the factorization method is not applicable.

2.2 World and Observation Models

Let us assume for the moment that a static camera observes a single moving object. To represent the situation we need two coordinate systems: a moving system **O** attached to the object, and a static system **C** attached to the camera, as shown in Figure 2.1.

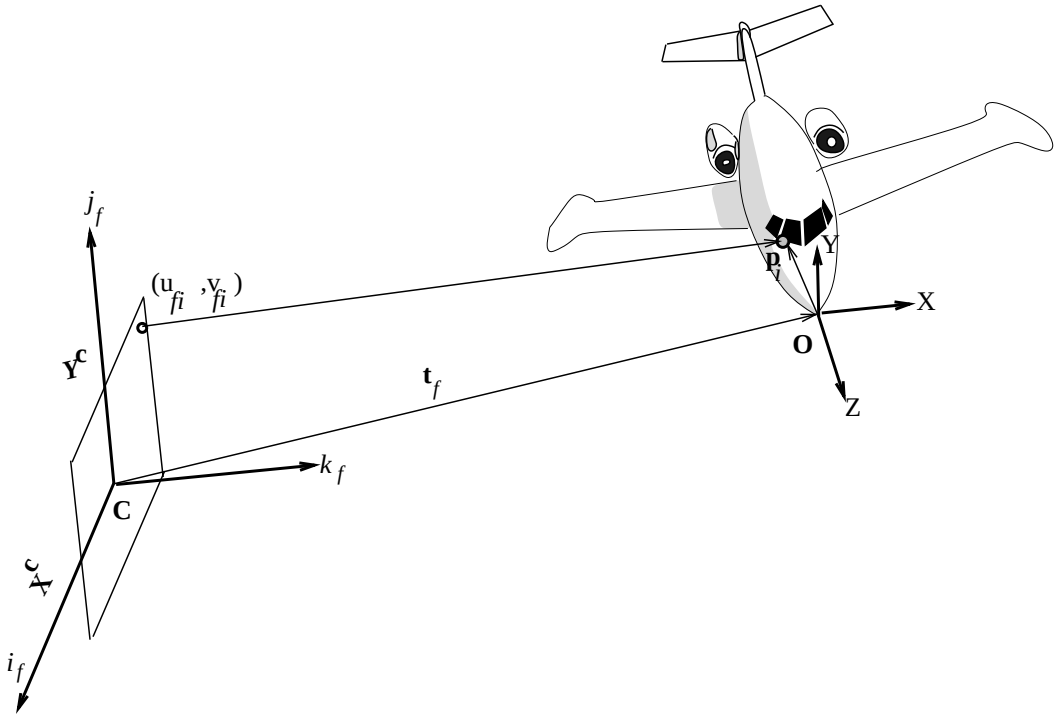


Figure 2.1: The camera and the object with its coordinatee system. The (unit) vectors i,j define the image plane and k its normal.

Consider a point $\mathbf{p}_i$ on the object. Let $\mathbf{p}^c_{fi}$ denote the position of $\mathbf{p}_i$, at instant f, represented in the camera coordinate system:

$$\mathbf{p}^c_{fi} = \mathbf{R}_f \; \mathbf{p}_i + \mathbf{t}_f. \tag{2.1}$$

Here

$$\mathbf{R}_f = \begin{bmatrix} \mathbf{i}_f^T \\ \mathbf{j}_f^T \\ \mathbf{k}_f^T \end{bmatrix} \tag{2.2}$$

is the rotation matrix whose rows $\mathbf{i}_f^T = \begin{bmatrix} i_{x_f} \; i_{y_f} \; i_{z_f} \end{bmatrix}$, $\mathbf{j}_f^T = \begin{bmatrix} j_{x_f} \; j_{y_f} \; j_z \end{bmatrix}$ and $\mathbf{k}_f^T = \begin{bmatrix} k_{x_f} \; k_{y_f} \; k_{z_f} \end{bmatrix}$ are the axes of the camera coordinate frame **C** expressed in the object's frame. The

vector

$$\mathbf{t}_f = \begin{bmatrix} t_{x_f} \\ t_{y_f} \\ t_{z_f} \end{bmatrix} \tag{2.3}$$

represents the position of the object's coordinate frame, at instant f, in the camera frame. The representation (2.1) can be simplified if we use homogeneous coordinates,

$$\mathbf{s} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.4}$$

for the object's point. In the homogeneous coordinates, equation (2.1) can be expressed as

$$\mathbf{s}^c_{fi} = \begin{bmatrix} \mathbf{p}^c_{fi} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_f & \mathbf{t}_f \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} \tag{2.5}$$

$$= \begin{bmatrix} \mathbf{R}_f & \mathbf{t}_f \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \mathbf{s}_i \tag{2.6}$$

The camera is modeled here as an orthographic projection. It produces the image by projecting a world point, parallel to the optical axis, onto the image plane. The image of point $\mathbf{p}_i$, at time f, is then given by the first two elements of $\mathbf{p}^c_{fi}$:

$$\begin{bmatrix} u_{fi} \\ v_{fi} \end{bmatrix} = \left[\begin{array}{ccc|c} i_{x_f} & i_{y_f} & i_{z_f} & t_{x_f} \\ j_{x_f} & j_{y_f} & j_{z_f} & t_{y_f} \end{array}\right] \mathbf{s}_i. \tag{2.7}$$

The object moves relative to the camera which acquires the images. In the sequence we track feature points from frame to frame. Suppose that we track N feature points over F frames, and that we collect all these measurements into a single matrix

$$\mathbf{W} = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1N} \\ & \vdots & \vdots & \\ u_{F1} & u_{F2} & \dots & u_{FN} \\ v_{11} & v_{12} & \dots & v_{1N} \\ & \vdots & \vdots & \\ v_{F1} & v_{F2} & \dots & v_{FN} \end{bmatrix}. \tag{2.8}$$

Each row of $\mathbf{W}$ lists the image coordinates u or v of all the feature points in each frame, and each column represents the image trajectory of one feature over the whole image sequence.

Using (2.7) we can represent $\mathbf{W}$ as the matrix product,

$$\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1N} \\ & \vdots & \vdots & \\ u_{F1} & u_{F2} & \dots & u_{FN} \\ v_{11} & v_{12} & \dots & v_{1N} \\ & \vdots & \vdots & \\ v_{F1} & v_{F2} & \dots & v_{FN} \end{bmatrix} = \left[\begin{array}{ccc|c} i_{x_1} & i_{y_1} & i_{z_1} & t_{x_1} \\ & \vdots & & \vdots \\ i_{x_F} & i_{y_F} & i_{z_F} & t_{x_F} \\ j_{x_1} & j_{y_1} & j_{z_1} & t_{y_1} \\ & \vdots & & \vdots \\ j_{x_F} & j_{y_F} & j_{z_F} & t_{y_F} \end{array}\right] \begin{bmatrix} s_{x_1} & & s_{x_N} \\ s_{y_1} & \dots & s_{y_N} \\ s_{z_1} & & s_{z_N} \\ 1 & & 1 \end{bmatrix}. \quad (2.9)$$

If we denote

$$\mathbf{M} = \left[\begin{array}{ccc|c} i_{x_1} & i_{y_1} & i_{z_1} & t_{x_1} \\ & \vdots & & \vdots \\ i_{x_F} & i_{y_F} & i_{z_F} & t_{x_F} \\ j_{x_1} & j_{y_1} & j_{z_1} & t_{y_1} \\ & \vdots & & \vdots \\ j_{x_F} & j_{y_F} & j_{z_F} & t_{y_F} \end{array}\right] \quad (2.10)$$

$$\mathbf{S} = \begin{bmatrix} s_{x_1} & & s_{x_N} \\ s_{y_1} & \dots & s_{y_N} \\ s_{z_1} & & s_{z_N} \\ 1 & & 1 \end{bmatrix} \quad (2.11)$$

as the motion and shape matrices respectively, we have the compact representation

$$\mathbf{W} = \mathbf{MS}. \quad (2.12)$$

Compared with the original formulation in [TK90b], note that the motion matrix contains translational components t_{x_f} and t_{y_f}. In the original formulation they were eliminated from the bilinear equation corresponding to (2.9) by subtracting beforehand the mean of the measurements. That procedure reflected the fact that under orthography the translation components do not provide any information about shape. In the case of a scene with multiple independent moving objects, the situation is not the same: on one hand the translation component of each object cannot be computed without segmentation, and on the other hand the translation component does provide information for segmenting the multibody scene. For these reasons, the translation component is kept in our formulation of the factorization method.

2.3 Solution for Shape and Motion by Factorization

We have derived the bilinear relationship (2.9) by modeling the imaging process. The problem of recovering shape and motion is in obtaining a motion matrix $\mathbf{M}$ and a shape matrix $\mathbf{S}$, given the measurements matrix $\mathbf{W}$. By simple inspection of (2.9) we can see that since $\mathbf{M}$ and $\mathbf{S}$ can be at most rank 4, $\mathbf{W}$ will be at most rank 4. However, in real situations $\mathbf{W}$ is constructed from noisy measurements, so the rank of $\mathbf{W}$ can be higher due to noise in the feature tracking. To approximate $\mathbf{W}$ by a rank-4 matrix, singular value decomposition (SVD) is the most robust and the best rank-revealing of all possible matrix decompositions [Ste92b]. With SVD, $\mathbf{W}$ is decomposed and approximated as:

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{2.13}$$

where $\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ is a diagonal matrix made of the four largest singular values, which reveal the most important components in the data, $\mathbf{U} \in R^{2F\times 4}$, and $\mathbf{V} \in R^{N\times 4}$ are respectively the left and right singular matrices, such that $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_{4\times 4}$.

By defining

$$\hat{\mathbf{M}} = \mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}} \tag{2.14}$$

$$\hat{\mathbf{S}} = \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{V}^T \tag{2.15}$$

we have the two matrices whose product can represent the bilinear system $\mathbf{W}$. However, this factorization is not unique, since for any invertible 4×4 matrix $\mathbf{A}$, $\mathbf{M} = \hat{\mathbf{M}}\mathbf{A}$ and $\mathbf{S} = \mathbf{A}^{-1}\hat{\mathbf{S}}$ are also a possible solution because

$$\mathbf{M}\mathbf{S} = \left(\hat{\mathbf{M}}\mathbf{A}\right)\left(\mathbf{A}^{-1}\hat{\mathbf{S}}\right) = \hat{\mathbf{M}}\hat{\mathbf{S}} = \mathbf{W}. \tag{2.16}$$

In other words, the singular value decomposition (2.13) provides a solution both for shape and motion up to an affine transformation.

The exact solution can be computed, using the fact that $\mathbf{M}$ must have certain properties. Let us denote the 4×4 matrix $\mathbf{A}$ as the concatenation of two blocks,

$$\mathbf{A} = [\mathbf{A}_R | \mathbf{a}_t] \,. \tag{2.17}$$

The first block $\mathbf{A}_R$ is a 4×3 submatrix related to the rotational component and $\mathbf{a}_t$ is a 4×1 vector related to translation. Now, since

$$\mathbf{M} \;=\; \hat{\mathbf{M}}\mathbf{A} = \left[\hat{\mathbf{M}}\mathbf{A}_R | \hat{\mathbf{M}}\mathbf{a}_t\right], \tag{2.18}$$

we can impose motion constraints, one on rotation and the other on translation, in order to solve for $\mathbf{A}$.

2.3.1 Rotation Constraints

Block $\mathbf{A}_R$ of $\mathbf{A}$, which is related to rotational motion, is constrained by the orthonormality of axis vectors $\mathbf{i}_f^T$ and $\mathbf{j}_f^T$; each of the $2F$ row entries of matrix $\hat{\mathbf{M}}\mathbf{A}_R$ is a unit norm vector and the first and second set of F rows are pairwise orthogonal. This yields a set of constraints

$$\hat{\mathbf{m}}_i \mathbf{A}_R \mathbf{A}_R^T \hat{\mathbf{m}}_i^T \;=\; 1 \tag{2.19}$$

$$\hat{\mathbf{m}}_j \mathbf{A}_R \mathbf{A}_R^T \hat{\mathbf{m}}_j^T \;=\; 1 \tag{2.20}$$

$$\hat{\mathbf{m}}_i \mathbf{A}_R \mathbf{A}_R^T \hat{\mathbf{m}}_j^T \;=\; 0 \tag{2.21}$$

for $i = 1 \ldots F, j = F+1 \ldots 2F$, where $\hat{\mathbf{m}}_i, \hat{\mathbf{m}}_j$ are rows i and j of matrix $\hat{\mathbf{M}}$. This is an overconstrained system which can be solved for the entries of $\mathbf{A}_R\mathbf{A}_R^T$ by using least squares techniques, and subsequently solving for $\mathbf{A}_R$. See [TK90a] for a detailed solution procedure.

2.3.2 Translation Constraints

In orthography, the projection of the 3D centroid of the features of an object into the image plane is the centroid of the feature points. The X and Y position of the centroid of the feature points is the average of each row of $\mathbf{W}$:

$$\overline{\mathbf{w}} = \begin{bmatrix} \frac{1}{N}\sum u_{1,i} \\ \vdots \\ \frac{1}{N}\sum v_{F,i} \end{bmatrix} \tag{2.22}$$

$$= \mathbf{M}\overline{\mathbf{s}} = \left[\hat{\mathbf{M}}\mathbf{A}_R | \hat{\mathbf{M}}\mathbf{a}_t\right] \begin{bmatrix} \overline{\mathbf{p}} \\ 1 \end{bmatrix}, \tag{2.23}$$

where

$$\overline{\mathbf{p}} = \frac{1}{N}\sum \mathbf{p}_i \tag{2.24}$$

is the centroid of the object.

The origin of the object's coordinate system is arbitrary, so we make it coincide with the centroid of the object, yelding $\overline{\mathbf{p}} = 0$. Then it follows immediately from (2.23) that

$$\overline{\mathbf{w}} = \hat{\mathbf{M}}\mathbf{a}_t. \tag{2.25}$$

This expression is also an overconstrained system of equations, which can be solved for the entries of $\mathbf{a}_t$ in the least square sense. The best estimate will be given by

$$\mathbf{a}_t = (\hat{\mathbf{M}}^T\hat{\mathbf{M}})^{-1}\hat{\mathbf{M}}^T\overline{\mathbf{w}} \tag{2.26}$$

$$= \mathbf{\Sigma}^{-1/2}\mathbf{U}^T\overline{\mathbf{w}}, \tag{2.27}$$

which completes the computation of all the elements of matrix $\mathbf{A}$.

2.4 Summary of Algorithm

Shape and motion using the factorization method can then be computed by implementing the following steps:

1. Build matrix $\mathbf{W}$ from the tracks of N features.
2. Using SVD, decompose the measurements matrix into $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$.
3. Compute $\mathbf{A}$ by imposing the rotational and translational constraints.
4. Define shape and motion as $\mathbf{S} = \mathbf{A}^{-1}\mathbf{\Sigma}^{1/2}\mathbf{V}^T$ and $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}^{1/2}\mathbf{A}$.

5. Align the camera and object reference systems such that $\mathbf{i}_1^T = [1\ 0\ 0]$ and $\mathbf{j}_1^T = [0\ 1\ 0]$

2.5 Experiments

In this section we present a single experiment to highlight the performance and robustness of the algorithm. The observed scene is a moving golf ball, where 120 features were selected and tracked over 72 frames.

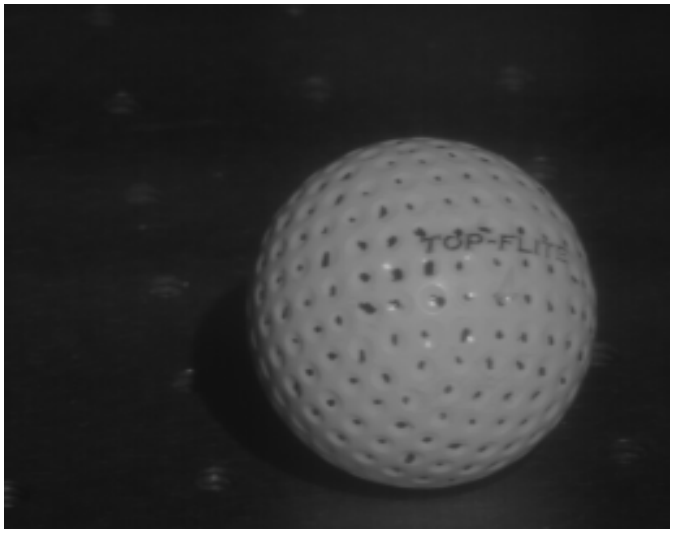


Figure 2.2: The first image of the sequence

Figure 2.2 shows the first image of the sequence, and figure 2.3 shows the last image with the trajectories of the selected features superimposed. Feature selection is done by using statistical parameters of all image points, and choosing the most "trackable". In Chapter 6 we will describe the tracking process and so details on how the features are tracked are omitted. Figure 2.4 shows a 3D plot of the recovered shape of the ball. Notice the correct spherical curvature of the plots. To convey a better perception of the recovered shape we show in Figure 2.5 a surface computed by bilinear interpolation of the 3D shape points, and with the original image texture mapped on it. Again, note the correctness of the outline contour of the surface.

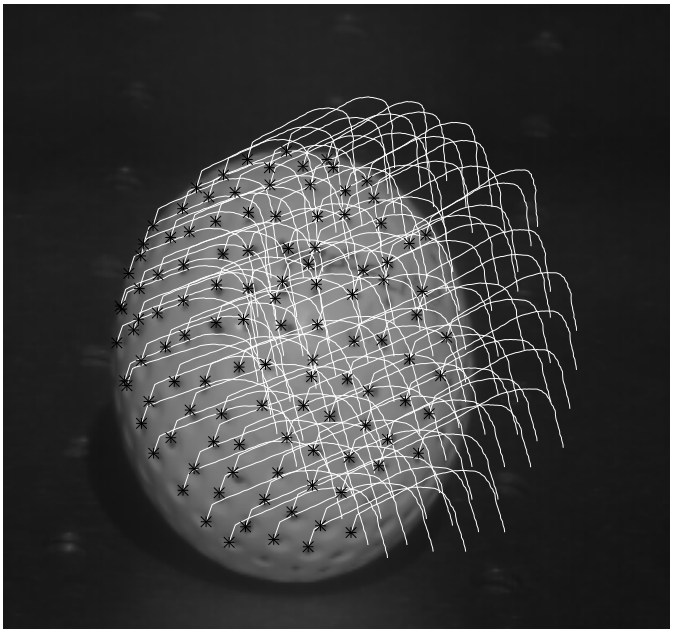


Figure 2.3: Tracked features over the whole sequence

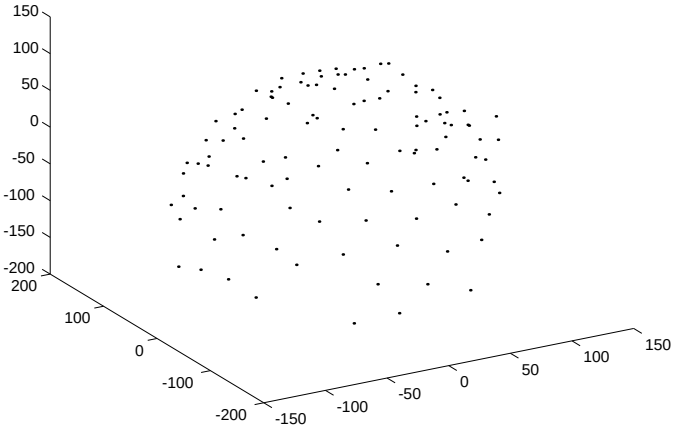


Figure 2.4: Recovered shape of the golf ball

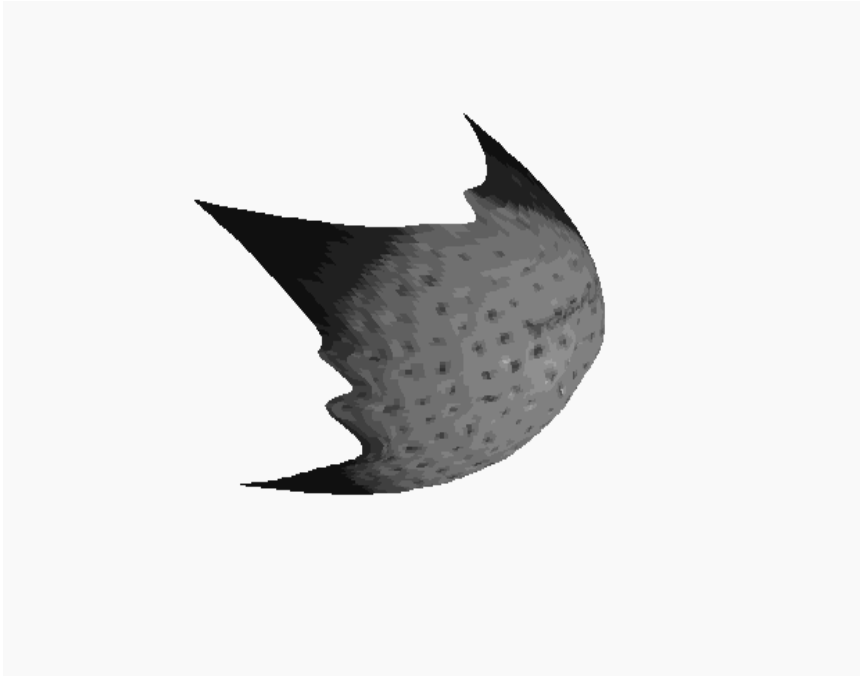


Figure 2.5: The shape with the texture of the original image mapped in the interpolated 3D recovered shape

Chapter 3

Geometrical Interpretation of Shape and Motion Decomposition

The factorization procedure developed in the previous section can be summarized as follows. Given the measurements matrix $\mathbf{W}$, compute its singular value decomposition (2.13)

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{3.1}$$

This decomposition allows the recovery of the shape and motion, $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$, up to an affine transform. Then, by using the constraints (2.19)-(2.21) and (2.27), we compute $\mathbf{A}$ and obtain a unique solution for motion and shape, being:

$$\mathbf{W} = \mathbf{M}\mathbf{S} \tag{3.2}$$

$$\mathbf{S} = \mathbf{A}^{-1}\hat{\mathbf{S}} = \mathbf{A}^{-1}\mathbf{\Sigma}^{\frac{1}{2}}\mathbf{V}^T \tag{3.3}$$

$$\mathbf{M} = \hat{\mathbf{M}}\mathbf{A} = \mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}}\mathbf{A}. \tag{3.4}$$

So far, all the matrix operations involved in the factorization have been considered from a purely numerical and algebraic point of view. It is useful to give a geometric interpretation to these matrices. Let us first consider the right singular matrix $\mathbf{V}^T$. From equation (3.3) we see that

$$\mathbf{V}^T = \mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{A}\mathbf{S}. \tag{3.5}$$

This equation reveals that $\mathbf{V}^T$ is a linear transformation of the shape. This transformation, produced by $\mathbf{A}$ and $\mathbf{\Sigma}$, is done in such a way that the resultant $\mathbf{V}$ is orthonormal.

To understand how $\mathbf{A}$ and $\mathbf{\Sigma}$ are related to shape, first we need to introduce a few geometric concepts. Let $\overline{\mathbf{s}}$ be, as before, the centroid of the object;

$$\overline{\mathbf{s}} = \frac{1}{N}\begin{bmatrix} \sum X_n \\ \sum Y_n \\ \sum Z_n \\ N \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{p}} \\ 1 \end{bmatrix}. \tag{3.6}$$

The centroid is the first-order moment of a set of points. The second order moments of a set of points are given in homogeneous coordinates by

$$\mathbf{\Lambda} = \mathbf{S}\mathbf{S}^T \tag{3.7}$$

$$= \begin{bmatrix} \sum X_n^2 & \sum X_n Y_n & \sum X_n Z_n & \sum X_n \\ \sum X_n Y_n & \sum Y_n^2 & \sum Y_n Z_n & \sum Y_n \\ \sum X_n Z_n & \sum Y_n Z_n & \sum Z_n^2 & \sum Z_n \\ \sum X_n & \sum Y_n & \sum Z_n & N \end{bmatrix} \tag{3.8}$$

$$= N\begin{bmatrix} \mathbf{\Lambda_0} & \overline{\mathbf{p}} \\ \overline{\mathbf{p}}^T & 1 \end{bmatrix}. \tag{3.9}$$

Generally speaking, the matrix $\mathbf{\Lambda}$ represents the orientation of the point distribution. Submatrix $\mathbf{\Lambda_0}$ is the matrix of the moments of inertia of the object. Its eigenvectors represent the directions of the three axes of symmetry of the ellipsoid of inertia. Using equation (3.3) for the shape representation, the matrix $\mathbf{\Lambda}$ can be written as

$$\mathbf{\Lambda} = \left(\mathbf{A}^{-1}\Sigma^{\frac{1}{2}}\mathbf{V}^T\right)\left(\mathbf{V}\mathbf{\Sigma}^{\frac{1}{2}}\mathbf{A}^{-T}\right) \tag{3.10}$$

$$= \mathbf{A}^{-1}\mathbf{\Sigma}\mathbf{A}^{-T}. \tag{3.11}$$

Similarly, we can also introduce the moments of motion. Vectors $\mathbf{i}_f$ and $\mathbf{j}_f$ of the motion matrix represent the X and Y axes of the camera in object coordinates. As the object moves, these vectors describe trajectories in the unit sphere. The second order moments of motion vectors can be defined as:

$$\mathbf{\Omega} = \mathbf{M}^T\mathbf{M} \tag{3.12}$$

$$= \begin{bmatrix} \sum(i_{x_f}^2 + j_{x_f}^2) & \sum(i_{x_f}i_{y_f} + j_{x_f}j_{y_f}) & \sum(i_{x_f}i_{z_f} + j_{x_f}j_{z_f}) & \sum(i_{x_f}t_{x_f} + j_{x_f}t_{y_f}) \\ \sum(i_{x_f}i_{y_f} + j_{x_f}j_{y_f}) & \sum(i_{y_f}^2 + j_{y_f}^2) & \sum(i_{y_f}i_{z_f} + j_{y_f}j_{z_f}) & \sum(i_{y_f}t_{x_f} + j_{y_f}t_{y_f}) \\ \sum(i_{x_f}i_{z_f} + j_{x_f}j_{z_f}) & \sum(i_{y_f}i_{z_f} + j_{y_f}j_{z_f}) & \sum(i_{z_f}^2 + i_{z_f}^2) & \sum(i_{z_f}t_{x_f} + j_{z_f}t_{y_f}) \\ \sum(i_{x_f}t_{x_f} + j_{x_f}t_{y_f}) & \sum(i_{y_f}t_{x_f} + j_{y_f}t_{y_f}) & \sum(i_{z_f}t_{x_f} + j_{z_f}t_{y_f}) & \sum(t_{x_f}^2 + t_{y_f}^2) \end{bmatrix} \tag{3.13}$$

$$= 2F\begin{bmatrix} \mathbf{\Omega_0} & \hat{\mathbf{t}} \\ \hat{\mathbf{t}}^T & \| \mathbf{t} \|_{\mathbf{\Sigma}}^2 \end{bmatrix}, \tag{3.14}$$

where $\mathbf{\Omega_0}$ is the matrix of the "moments of inertia" of the image plane motion axes, $\hat{\mathbf{t}}$ is the average position of the camera origin in the object's coordinate system and $\| \mathbf{t} \|^2_{\mathbf{\Sigma}}$ the average of the translation vector norm.

Using (3.4) we can write (3.12) as:

$$\mathbf{\Omega} = \mathbf{A}^T \mathbf{\Sigma} \mathbf{A}. \tag{3.15}$$

The bilinearity of the observations is also reflected in the second-order motion moments. By multiplying the motion moment (3.15) by the shape moment (3.11), we have

$$\mathbf{\Omega\Lambda} = \mathbf{A}^T \mathbf{\Sigma}^2 \mathbf{A}^{-T}. \tag{3.16}$$

or

$$\mathbf{\Omega\Lambda}\, \mathbf{A}^T = \mathbf{A}^T\, \mathbf{\Sigma}^2. \tag{3.17}$$

This is a standard form of a 4×4 eigensystem, where $\mathbf{\Sigma}^2$ is the matrix of the eigenvalues and $\mathbf{A}^T$ the matrix of the eigenvectors. The squares of the singular values σ_i^2 of the measurements matrix $\mathbf{W}$ are the eigenvalues of the product of the motion and shape moment matrices, and their eigenvectors form the rows of the transformation matrix $\mathbf{A}$. Geometrically, the eigenvectors represent space orientation, resulting from projecting the symmetry axes of motion into the symmetry axes of shape. The eigenvalues (thus singular values of W) represent object's "lengths" multiplied by motion moments.

In order to illustrate the meaning of these matrices, let us consider one simple case where we choose a particular set of coordinate axis and a particular type of motion. Since the object coordinate system is arbitrary, we choose the object's centroid and axis of inertia to set up the origin and orientation of its coordinate system. Then, in object coordinates $\overline{\mathbf{p}} = 0$, and the translational component $\mathbf{t} = \overline{\mathbf{w}}$ can be computed and eliminated from $\mathbf{W}$. On the other hand, since the object's reference frame coincides with its axis of symmetry, the matrix Λ_0 is diagonal. Then, the eigensystem (3.17) is

reduced to the 3×3 eigensystem

$$\mathbf{A}^T \mathbf{\Sigma}^2 = \mathbf{\Omega_0} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \mathbf{A}^T, \tag{3.18}$$

where

$$\lambda_1 = \sum X_i^2 \tag{3.19}$$

$$\lambda_2 = \sum Y_i^2 \tag{3.20}$$

$$\lambda_3 = \sum Z_i^2 \tag{3.21}$$

represent the moments of inertia of the object sorted in decreasing order.

Regarding motion, we specify that the object undergoes a symmetrical motion as shown in Figure 3.1. It rotates up and down along the $Y'Z'$ plane (of the camera), followed by an orthogonal motion to the left and right along the $X'Z'$ plane. Furthermore,

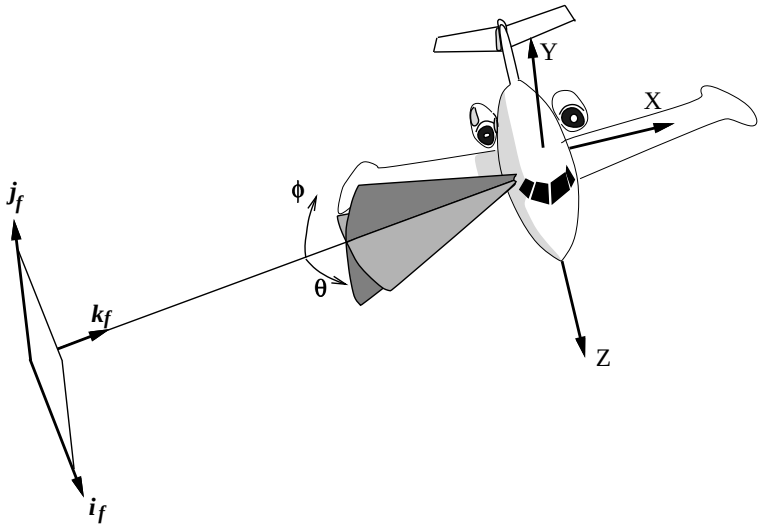


Figure 3.1: Symmetrical motion

the motion is such that the amplitude of the upward (leftward) and downward (rightward) motion is the same, that is, the maximum and minimum angles of the trajectory are equal. Under these restrictions the matrix of the moments of inertia of the motion

is also diagonal:

$$\mathbf{\Omega_0} = \begin{bmatrix} \omega_1 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{bmatrix} \tag{3.22}$$

where

$$\omega_1 = F_1 + \sum_{i=1}^{F_2} \cos^2(\theta_i) \tag{3.23}$$

$$\omega_2 = F_2 + \sum_{i=1}^{F_1} \cos^2(\phi_i) \tag{3.24}$$

$$\omega_3 = \sum_{i=1}^{F_1} \sin^2(\phi_i) + \sum_{i=1}^{F_2} \sin^2(\theta_i) \tag{3.25}$$

are the moments of inertia of the motion and F_1 and F_2 are the number of frames captured in each section of the motion.

The eigensystem (3.18) will be given by

$$\mathbf{A}^T\mathbf{\Sigma}^2 = \begin{bmatrix} \omega_1\lambda_1 & 0 & 0 \\ 0 & \omega_2\lambda_2 & 0 \\ 0 & 0 & \omega_2\lambda_2 \end{bmatrix} \mathbf{A}^T. \tag{3.26}$$

Equation (3.26) shows more explicitly the previously deduced properties of structure from motion recovery with an orthographic camera. Under "symmetric" motion $\mathbf{A}^T = \mathbf{I}_{3\times3}$ and we will have:

- The columns of the singular matrix $\mathbf{V}$ represent the 3D shape of the object expressed in coordinates of its axis of symmetry and scaled so that its moments are unitary. As long as motion remains "symmetric", matrix $\mathbf{V}$ does not depend on the actual motion.

- The squares of the singular values of the track matrix are the moments of inertia of the object multiplied by a scale factor. If the amplitude of the motion is the same along both axes, there is one motion for which $\omega_1 = \omega_2 = \omega_3$ and then the singular values of $\mathbf{W}$ will be the moments of inertia of the object, up to a single scale factor.

Chapter 4

The Multi-body Factorization Method

Until now we have assumed that the scene contains a single moving object. If there is more than one moving object, the measurements matrix $\mathbf{W}$ will contain features (columns) which are produced by different motions. One may think that solving the problem requires first sorting the columns of the measurements matrix $\mathbf{W}$ into submatrices, each of which contains features from one object only, so that the factorization technique of the previous sections can be applied individually. In fact this is exactly the approach taken by [BB91] and [Gea94]. We will show in this section that the multibody problem can be solved without prior segmentation. For the sake of simplicity we will present the theory and the method for the two body case, but it will be clear that the method is applicable to the general case of an arbitrary unknown number of bodies.

4.1 The Multi-body Motion Recovery Problem: Its Difficulty

Suppose we have a scene in which two objects are moving and we take an image sequence of F frames. In this case, the relevant coordinate systems are depicted in figure 4.1. Suppose also that the set of features that we have observed and tracked in the image sequence actually consists of N_1 feature points from object 1 and N_2 from object 2. For the moment, imagine that somehow we knew the classification of features

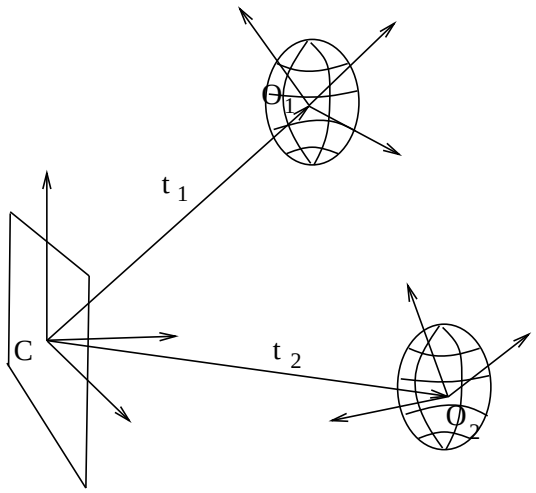


Figure 4.1: Two bodies: The coordinate systems

and thus could permute the columns of $\mathbf{W}$ in such a way that the first N_1 columns belong to object 1 and the following N_2 columns to object 2. Matrix $\mathbf{W}$ would have the canonical form:

$$\mathbf{W}^* = [\,\mathbf{W}_1 \,|\, \mathbf{W}_2\,]. \tag{4.1}$$

Each measurements submatrix can be factorized as

$$\mathbf{W}_l = \mathbf{U}_l \mathbf{\Sigma}_l \mathbf{V}_l^T \tag{4.2}$$

$$= \mathbf{M}_l \mathbf{S}_l = (\hat{\mathbf{M}}_l \mathbf{A}_l)(\mathbf{A}_l^{-1} \hat{\mathbf{S}}_l) \tag{4.3}$$

with $l = 1$ and 2 for object 1 and 2 respectively. Equation (4.1) has now the canonical factorization:

$$\mathbf{W}^* = [\mathbf{M}_1|\mathbf{M}_2] \begin{bmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2 \end{bmatrix} \tag{4.4}$$

$$= \left[\hat{\mathbf{M}}_1|\hat{\mathbf{M}}_2\right] \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{A}_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2^{-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{S}}_1 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{S}}_2 \end{bmatrix}. \tag{4.5}$$

By denoting

$$\hat{\mathbf{M}}^* = \left[\hat{\mathbf{M}}_1|\hat{\mathbf{M}}_2\right] \tag{4.6}$$

$$\hat{\mathbf{S}}^* = \begin{bmatrix} \hat{\mathbf{S}}_1 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{S}}_2 \end{bmatrix} \tag{4.7}$$

$$\mathbf{A}^* = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{bmatrix} \tag{4.8}$$

$$\mathbf{U}^* = [\mathbf{U}_1 | \mathbf{U}_2] \tag{4.9}$$

$$\mathbf{\Sigma}^* = \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{bmatrix} \tag{4.10}$$

$$\mathbf{V}^{*T} = \begin{bmatrix} \mathbf{V}_1^T & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2^T \end{bmatrix}, \tag{4.11}$$

we obtain a factorization similar to a single object case, where the canonical measurements matrix relates to shape and motion according to:

$$\mathbf{W}^* = \mathbf{M}^* \mathbf{S}^* \tag{4.12}$$

$$\mathbf{S}^* = \mathbf{A}^{*-1} \hat{\mathbf{S}}^* = \mathbf{A}^{*-1} \mathbf{\Sigma}^{*\frac{1}{2}} \mathbf{V}^{*T} \tag{4.13}$$

$$\mathbf{M}^* = \hat{\mathbf{M}}^* \mathbf{A}^* = \mathbf{U}^* \mathbf{\Sigma}^{*\frac{1}{2}} \mathbf{A}^* \tag{4.14}$$

From equation (4.4), we see that $\mathbf{W}^*$ (and therefore $\mathbf{W}$) will have at most rank 8; $\mathbf{W}_1$ and $\mathbf{W}_2$ are at most rank 4. For the remainder of this section let us consider non-degenerate cases where the rank of $\mathbf{W}$ is in fact equal to 8; that is, the object shape is actually full three-dimensional (excluding planes and lines) and the motion vectors span a three dimensional space for both objects. Degenerate cases will be discussed later on.

In reality, we do not know which features belong to which object, and thus the order of columns of the given measurements matrix $\mathbf{W}$ is a mixture of features from objects 1 and 2. We can still apply singular value decomposition (SVD) to the measurements matrix, and obtain

$$\mathbf{W} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T. \tag{4.15}$$

Then it appears that the remaining task is to find the linear transformation $\mathbf{A}$ such that shape and motion will have the block structure of equations (4.13) and (4.14).

There is, however, a fundamental difficulty in doing this. The metric (rotation and translation) constraints (eq.(2.19)-(2.20) and (2.25)-(2.27)) were obtained in section 2.3 by considering the motion matrix for one object, that is, by assuming that the measurements matrix consists of features from a single object. These constraints

are therefore only applicable when the segmentation is known. This is exactly the mathematical evidence of the cyclic dilemma mentioned earlier.

Faced with this difficulty, the usual approach would be to group features bit by bit so that we segment $\mathbf{W}$ into two rank-4 matrices and obtain the factorization as in equation (4.4). For example, a simplistic procedure would be as follows: pick the first four columns of $\mathbf{W}$ and span a rank-4 subspace. If the fifth column belongs to the subspace (ie. is linear dependent on the first four, or almost linear dependent in the case of noisy measurements), then classify it as belonging to the same object as the first four columns and update the subspace representation. Otherwise, it belongs to a new object. Apply this procedure recursively to all the remaining columns. This approach is in fact essentially that used by [BB91] and [Gea94] to split matrix $\mathbf{W}$, and similar to that suggested by Ullman [Ull83], where the criterion for merging was local rigidity.

However, this cluster-and-test approach presents several disadvantages. First, there is no guarantee that the first four columns, which always form a rank-4 subspace, are generated by the same object. Second, if we use a sequential procedure like that above or a variation on it, the final result is dependent on where we start the procedure, and alternatively, the search for the globally optimal segmentation will most likely be computationally very expensive. Finally, prior knowledge of the number of objects becomes very critical, since depending on the decision criterion of subspace inclusion, the final number of objects may vary arbitrarily.[1]

4.2 A Mathematical Construct of Shapes Invariant to Motions

In the multi-body structure-from-motion problem, the main difficulty, revealed just above, is due to the fact that shape and motion interact. Mathematically, as shown

[1] While this is beyond the scope of the assumption in this section, this cluster-and-test approach also requires the prior knowledge of the ranks of objects, since for example a rank-8 measurements matrix might have been generated by two line (rank-2) objects and one full 3D (rank 4) object instead of two full 3D objects, and hence attempting to find two rank-4 subspaces could be wrong.

in (4.4), the rank-8 measurement space is originally generated by the two subspaces of rank 4, each represented by the block-diagonal shape matrix $\mathbf{S}^*$. However, the recovered shape space $\mathbf{V}^T$, obtained by the singular value decomposition, is in general a linear combination of the two subspaces and does not exhibit a block-diagonal structure.

There is however a mathematical construct that preserves the original subspace structure. Let us define $\mathbf{Q}$ as the $(N_1 + N_2) \times (N_1 + N_2)$ square matrix

$$\mathbf{Q} = \mathbf{V}\mathbf{V}^T. \tag{4.16}$$

We will call this matrix the *shape interaction matrix*. Mathematically, it is the orthogonal operator that projects $N = (N_1 + N_2)$ dimensional vectors to the subspace spanned by the columns of $\mathbf{V}$. This matrix $\mathbf{Q}$ has several interesting and useful properties. First, by definition it is uniquely computable only from the measurements $\mathbf{W}$, since $\mathbf{V}$ is uniquely obtained by the singular value decomposition of $\mathbf{W}$.

Secondly, each element of $\mathbf{Q}$ provides important information about whether a pair of features belong to the same object. Since $\mathbf{W}^*$ is formed applying a set of column permutations to $\mathbf{W}$, $\mathbf{V}^{*T}$ will also result by permuting the same set of columns of $\mathbf{V}^T$[2]. Thus, the canonical $\mathbf{Q}^*$ will result by permuting columns and rows of $\mathbf{Q}$ (the order of each operation is irrelevant) so that both matrices have the same entry values but in different locations. Then, let us compute $\mathbf{Q}^*$ for the canonical form of $\mathbf{W}^*$. By inserting (4.13) into the canonical version of (4.16) we can obtain the following derivation:

$$\begin{aligned}
\mathbf{Q}^* &= \mathbf{V}^*\mathbf{V}^{*T} && (4.17)\\
&= \mathbf{S}^{*T}\mathbf{A}^{*T}\mathbf{\Sigma}^*\,\mathbf{A}^*\,\mathbf{S}^* && (4.18)\\
&= \mathbf{S}^{*T}(\mathbf{A}^{*-1}\mathbf{\Sigma}^{*-1}\mathbf{A}^{*-T})^{-1}\mathbf{S}^* && (4.19)\\
&= \mathbf{S}^{*T}\left[(\mathbf{A}^{*-1}\mathbf{\Sigma}^{*-1/2}\mathbf{V}^{*T})(\mathbf{V}^*\mathbf{\Sigma}^{*-1/2}\mathbf{A}^{*-T})\right]^{-1}\mathbf{S}^* = \mathbf{S}^{*T}(\mathbf{S}^*\mathbf{S}^{*T})^{-1}\mathbf{S}^* && (4.20)
\end{aligned}$$

[2] $\mathbf{V}$ and $\mathbf{V}^*$ may still differ up to an orthonormal transformation, but this is irrelevant to our derivations.

$$= \begin{bmatrix} \mathbf{S_1}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{S_2}^T \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda_1}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda_2}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{S_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S_2} \end{bmatrix} \tag{4.21}$$

$$= \begin{bmatrix} \mathbf{S_1}^T\mathbf{\Lambda_1}^{-1}\mathbf{S_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S_2}^T\mathbf{\Lambda_2}^{-1}\mathbf{S_2} \end{bmatrix}. \tag{4.22}$$

,

where $\mathbf{\Lambda_1}$ and $\mathbf{\Lambda_2}$ are the 4×4 matrices of the moments of inertia of each object. This means that the canonical $\mathbf{Q}^*$ matrix for the sorted $\mathbf{W}^*$ has a well-defined block-diagonal structure. Also, each entry has the value

$$Q^*_{ij} = \begin{cases} \mathbf{s}^T_{1_i}\mathbf{\Lambda_1}^{-1}\mathbf{s}_{1_j} & \text{if feature trajectory } i \text{ and } j \text{ belong to object 1} \\ \mathbf{s}^T_{2_i}\mathbf{\Lambda_2}^{-1}\mathbf{s}_{2_j} & \text{if feature trajectory } i \text{ and } j \text{ belong to object 2} \\ 0 & \text{if feature trajectory } i \text{ and } j \text{ belong to different objects.} \end{cases} \tag{4.23}$$

Properties of $\mathbf{Q}^*$

Invariant to the Number of Objects

Even though expression (4.22) was derived for the case of two objects, it is now clear that its structure is the same for any number of objects. In fact, if the scene has M moving objects $\mathbf{Q}^*$ would still have the block diagonal form:

$$\mathbf{Q}^* = \begin{bmatrix} \mathbf{S_1}^T\mathbf{\Lambda_1}^{-1}\mathbf{S_1} & 0 & 0 & 0 & 0 \\ & \ddots & & & \\ 0 & 0 & \mathbf{S_k}^T\mathbf{\Lambda_k}^{-1}\mathbf{S_k} & 0 & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & 0 & \mathbf{S_M}^T\mathbf{\Lambda_M}^{-1}\mathbf{S_M} \end{bmatrix}. \tag{4.24}$$

If any features i and j belong to different objects their entry Q^*_{ij} will be zero. This property also holds in $\mathbf{Q}$, that is, regardless the way features are sorted in $\mathbf{W}$, the shape interaction matrix contains all the necessary information about the multibody scene.

Invariant to Object Motion

Most importantly, entries Q^*_{ij} are invariant to motion. This is true since equations (4.23) include only shape variable $\mathbf{S}_k$, and not $\mathbf{M}$. In other words, no matter how the objects move, they will produce the same set of entries in matrix $\mathbf{Q}^*$.

Invariant to Coordinate System

The shape interaction matrix is invariant to the coordinate system in which we represent the shape. Suppose we transform the shape, $\mathbf{S}$, of object k by the general transformation $\mathbf{T} \in R^{4\times4}$:

$$\mathbf{S}' = \mathbf{TS}. \tag{4.25}$$

The corresponding block-diagonal element matrix will be

$$\mathbf{S}'^T(\mathbf{S}'\mathbf{S}'^T)^{-1}\mathbf{S}' \;=\; (\mathbf{TS})^T\left[(\mathbf{TS})(\mathbf{TS})^T\right]^{-1}(\mathbf{TS}) = \mathbf{S}^T(\mathbf{SS}^T)^{-1}\mathbf{S} \tag{4.26}$$

which remains the same.

Invariant to Shape Rank

Finally, the shape interaction matrix is also invariant to the type of object. The rank of the shape matrix $\mathbf{S}$ can be 2 for a line, 3 for a plane and 4 for a full 3D object. However, the entries of $\mathbf{Q}^*$ will have the same general expression. For degenerate shapes (lines and planes), the difference will be the number of rows and columns of matrices $\mathbf{S}$ and $\mathbf{\Lambda}$. Since $\mathbf{Q}$ is invariant to the coordinate system, if object k is a line, $\mathbf{S}_k$ can be represented as a $2 \times N_k$ matrix ($3 \times N_k$ for a plane), therefore $\mathbf{\Lambda}_k$ will be 2×2 (3×3 for a plane). In both cases, the total rank of $\mathbf{Q}^*$ changes but not its structure nor its entries.

4.3 Sorting Matrix Q into Canonical Form

In the previous section we have shown that we can compute matrix $\mathbf{Q}$ without knowing the segmentation of the features. Each element Q_{ij} can be interpreted as a

measure of the interaction between features i and j: if its value is non-zero, then the features belong to the same object, otherwise they belong to different objects if the value is zero. Also, if the features are sorted correctly into the canonical form of the measurement matrix $\mathbf{W}^*$, then the corresponding canonical shape interaction matrix $\mathbf{Q}^*$ must be block diagonal.

Now, the problem of segmenting and recovering motion of multiple objects has been reduced to that of sorting the entries of matrix $\mathbf{Q}$, by swapping pairs of rows and columns until it becomes block diagonal. Once this is achieved, applying the corresponding permutations to the columns of $\mathbf{W}$ will transform it to the canonical form where features from one object are grouped into adjacent columns. This equivalence between sorting $\mathbf{Q}$ and permuting $\mathbf{W}$ is illustrated in figure 4.2.

With noisy measurements, a pair of features from different objects may exhibit a small non-zero entry. We can regard Q_{ij}^2 as representing the energy of the shape interaction between features i and j. Then, the block diagonalization of $\mathbf{Q}$ can be achieved by minimizing the total energy of all possible off-diagonal blocks over all sets of permutations of rows and columns of $\mathbf{Q}$. This is a computationally overwhelming task since the number of possibilities is factorial with the number of features.

Alternatively, since matrix $\{Q_{ij}^2\}$ is symmetric and all elements are positive, it defines the incidence matrix of a graph of $N_1 + N_2$ nodes, where the Q_{ij}^2 indicates the weight of the link (i, j). Several graph-theoretical algorithms [CLR86], such as the minimum spanning tree (MST), can be used to achieve block diagonalization much more efficiently than energy minimization.

The importance of these methods lie in the interesting interpretation of the shape interaction matrix (or the square of its elements). In noise-free environments $\mathbf{Q}$ is in fact a forest: a graph made of several non-connected subgraphs, and segmentation reduces to looking for the connected components. In the presence of noise $\mathbf{Q}$ is interpreted as a single fully connected graph from which the noisy links have to be removed. We can use the MST to achieve a minimum representation of $\mathbf{Q}$ where the noisy links can be easily

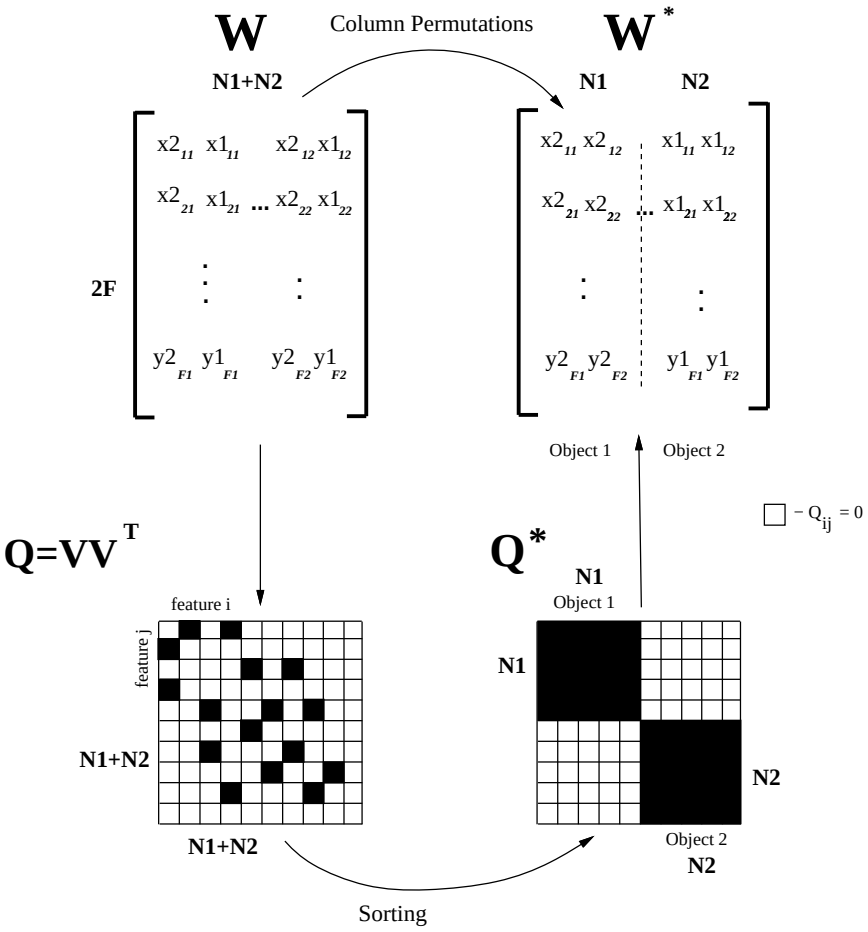


Figure 4.2: Segmentation process

removed. However, a single spike of noise can be understood by the sorting algorithm as a link, jeopardizing the entire segmentation. Because of this, and also because of the difficulty of coding prior knowledge in the MST algorithm we have devised another algorithm that explores the global constraints on $\mathbf{Q}$, allowing a much more efficient and robust sorting.

4.4 Segmentation Algorithm

The algorithm we propose here segments a multibody scene in two steps: In the first step rows and columns of $\mathbf{Q}$ are iteratively permuted in such a way that features of the same object are arranged adjacently into blocks, transforming $\mathbf{Q}$ into the canonical shape interaction matrix $\mathbf{Q}^*$. Though sorted, $\mathbf{Q}^*$ alone does not provide information about the size and location of each block, therefore a second step is required to compute the rows/columns that limit the blocks corresponding to features of a single object.

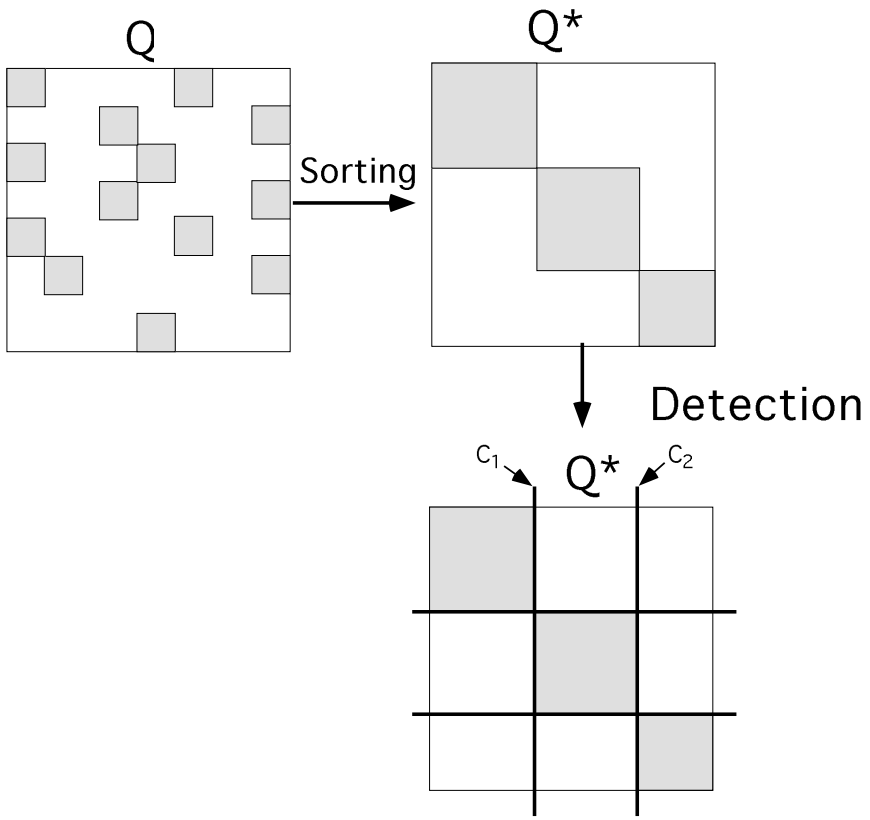


Figure 4.3: The segmentation algorithm: Sorting matrix $\mathbf{Q}$ and detecting the blocks

Consider an example where three objects move independently. Figure 4.3 depicts the two steps of the algorithm graphically. First, the sorting process transforms the original $\mathbf{Q}$ into $\mathbf{Q}^*$. Then the detection process determines columns c_1 and c_2 which isolate the blocks corresponding to each object. This detection process is quite relevant mainly for two reasons: on one hand the off-diagonal elements are non-zero and we have no prior knowledge about either the signal or the noise models, hence we are unable to detect the limiting columns based on local information alone. In other words, we cannot compute optimal thresholds to classify the elements of $\mathbf{Q}$ as either noise or signal [VT68]. Also, the detection process must take into consideration shape degeneracy issues, that is, cases where objects have less than 3 independent dimensions (lines and planes). Fortunately, using the properties of $\mathbf{Q}$, the block diagonal structure is invariant with shape rank therefore we have been able to develop a detection algorithm

that robustly handles any shape degeneracy.

4.4.1 Sorting

As already stated, sorting $\mathbf{Q}$ is equivalent to minimizing the energy of the off-diagonal blocks, over the set of all permutations of rows and columns. A straightforward inspection shows that this type of optimization leads to a combinatorial explosion. Instead, we can considerably reduce the search effort by using suboptimal strategies without jeopardizing performance.

Our algorithm implements a hill-climbing tree search. By hill-climbing we mean a search procedure that, at each search level (or iteration), chooses the best path without taking into account past decisions.

At each iteration, say k, the current state is represented by a $k \times k$ submatrix $\mathbf{Q}^{*k}$ which contains the features sorted so far. A set of operations expands the current state, producing candidates (features) to be included in the current sorted $\mathbf{Q}^{*k}$. Figure 4.4

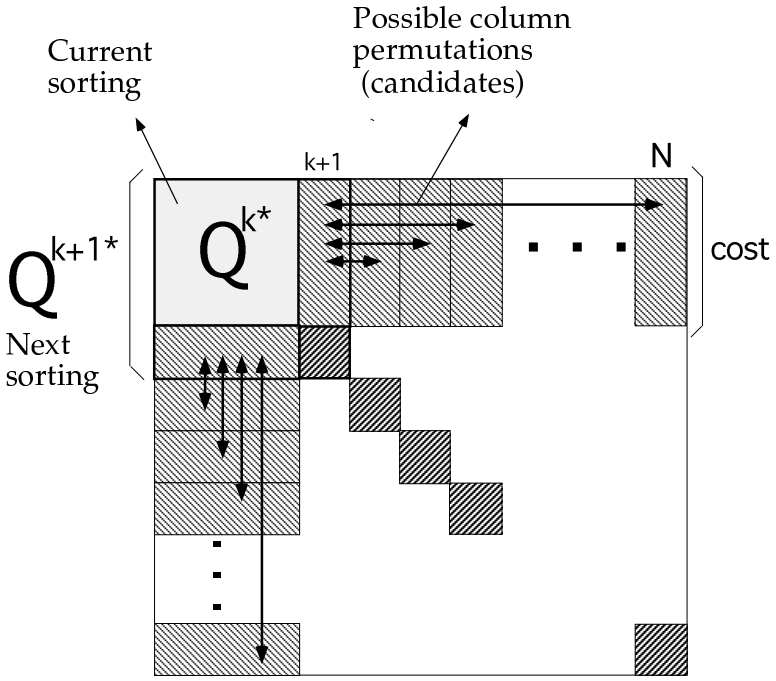


Figure 4.4: Sorting Algorithm: At iteration k, columns $k+1$ to N are permuted and the column with the highest norm is selected to form $\mathbf{Q}^{*k+1}$

shows iteration k where feature $k+1$ is to be selected from among the $N-k$ candidates. The candidates are features $k+1$ to N whose columns and rows are not included in the current segmentation. The cost, C_j^k of each candidate is given by the energy of the first k elements

$$C_j^k = \sum_{i=1}^{k} Q_{i,j}^2 \quad \text{for } (j = k+1, \cdots, N), \tag{4.27}$$

which represents the total energy of interaction between each of the candidate features and the set of already sorted features[3]. By maximizing the cost function C_j^k, our search strategy selects the feature whose global energy of interaction with the current segmentation, is the largest.

The updated state Q^{*k+1} is obtained by augmenting Q^{*k} with the column and the row of the best feature. The column corresponding to this feature is first permuted with column $k+1$, followed by a permutation of rows with the same indices. Matrix $(\mathbf{Q}*^{k+1})$ is then formed with the first $(k+1) \times (k+1)$ elements of the permuted shape interaction matrix. As a result of this maximization strategy, submatrix Q^{*k+1} has maximal energy among all possible $(k+1) \times (k+1)$ submatrices of $\mathbf{Q}$. Unless the noise energy is similar to that of the signal, for all features in a set, this sorting procedure groups features by the strength of their coupling. Even though this procedure may look like a blind search, in the next section we will show that this maximization relates the energy maximization to rank properties of operators $\mathbf{Q}^{*k}$, thus taking into account the structure of the problem.

4.4.2 Block Detection

Having sorted matrix $\mathbf{Q}$ into canonical form, the matrix $\mathbf{Q}^*$ for an arbitrary number of objects M has the block form:

[3]Note that the diagonal elements are not included in the cost since they do not measure any interaction among features

$$\mathbf{Q}^* = \begin{bmatrix} \mathbf{S_1}^T\mathbf{\Lambda_1}^{-1}\mathbf{S_1} & 0 & 0 & 0 & 0 \\ & \ddots & & & \\ 0 & 0 & \mathbf{S_K}^T\mathbf{\Lambda_K}^{-1}\mathbf{S_K} & 0 & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & 0 & \mathbf{S_M}^T\mathbf{\Lambda_M}^{-1}\mathbf{S_M} \end{bmatrix} \tag{4.28}$$

$$= \begin{bmatrix} \mathbf{Q}_1 & 0 & 0 & 0 & 0 \\ & \ddots & & & \\ 0 & 0 & \mathbf{Q}_K & 0 & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & 0 & \mathbf{Q}_M \end{bmatrix}. \tag{4.29}$$

Since noise induces a small non-zero value in the off-diagonal elements, instead of detecting zeros we must detect the transition between blocks (signal) and the off-diagonal elements (noise). Even assuming correct sorting, this transition is quite hard to detect, based on local values alone, due to the lack of knowledge about noise characteristics. In other words, it is not possible to set up a threshold below which an entry could be considered zero. The threshold is determined by an optimality criterion involving the noise probability distribution function [VT68].

However, there are global constraints that can be applied to $\mathbf{Q}^*$. First the rank of each block is constrained to be 2 (a line), 3 (a plane) or 4 (a full 3D object). Second, we can relate the rank of a block to its energy: in fact, equation (4.28) shows that the rank of each $\mathbf{Q}_K$ is the same as the rank of the shape matrix of object K. Also, the square of the Frobenius norm (F-norm) of matrix $\mathbf{Q}_K$ relates to the block energy and to its singular values σ_{K_i} according to:

$$\|\mathbf{Q}_K\|_F^2 = \sum_{i=1}^{N_K}\sum_{j=1}^{N_K} {Q_{K_{ij}}}^2 = \sigma_{K_1}^2 + \cdots + \sigma_{K_R}^2, \tag{4.30}$$

where N_K is the number of features of each block and R its rank. The number of non-zero singular values is $R = 2, 3, 4$ depending whether the object is a line, a plane or a 3D object respectively. Since $\mathbf{Q}_K$ is orthonormal, all singular values, σ_{K_i}, are equal

to 1 and hence for each type of object, the sum (4.30) adds to 2 (line), 3 (plane) or 4 (3D object). Then, we can relate the energy of each block with its rank by

$$\|\mathbf{Q}_K\|_F^2 = \sum_{i=1}^{N_K}\sum_{j=1}^{N_K} {Q_{ij}}^2 \tag{4.31}$$

$$= \sigma_{K_1}^2 + \cdots + \sigma_{K_R}^2 = rank(\mathbf{Q}_K). \tag{4.32}$$

Instead of considering an individual block, let us compute the sum (4.30) for the first m columns/rows of $\mathbf{Q}^*$, defined by the function $\varepsilon(\cdot)$:

$$\varepsilon(m) = \sum_{i=1}^{m}\sum_{j=1}^{m} {\mathbf{Q}^*_{ij}}^2, \tag{4.33}$$

for $m = 1...N$. Then, columns for which the integer part of ε increases one unit are potential block limiting columns, provided the block rank constraints are satisfied. Consider Figure 4.5 which illustrates one possible shape of the function ε for the case of 2 objects with rank 4 shape. The vertical dashed lines indicate rank jumps, that is, columns where ε is a whole number or its integer part increases by one (under noisy conditions, except for $m = N$, the function ε may never be a whole number). Given the indices of the columns of integer crossing by ε, segmentation consists in finding the blocks that match these rank jumps and satisfy the constraints. The solution is not unique, as our examples illustrate in Figure 4.6. For a rank 8 matrix $\mathbf{Q}^*$, we have 8 possible segmentations, obtained by considering all possible combinations of rank 2, 3 or 4 blocks whose sum is 8.

In Figure 4.6 we show only four possible solutions for this example. The numbers in Figure 4.6(a) represent the columns and rows that limit submatrices of $\mathbf{Q}^*$ for which the rank jumps by one. In the same figure, the shaded rectangles represent the correct feature segmentation. The superimposed white blocks in Figures 4.6(b), (c) and (d) show other possible segmentations that also match the rank jumps and satisfy the block rank constraint. Among the 8 possible configurations, figure 4.6 considers the following scene segmentations for the rank 8 $\mathbf{Q}^*$:

4.6(a) -Two rank 4 objects: assuming the scene is formed by two, rank 4, objects there is only one block configuration. The first four rank

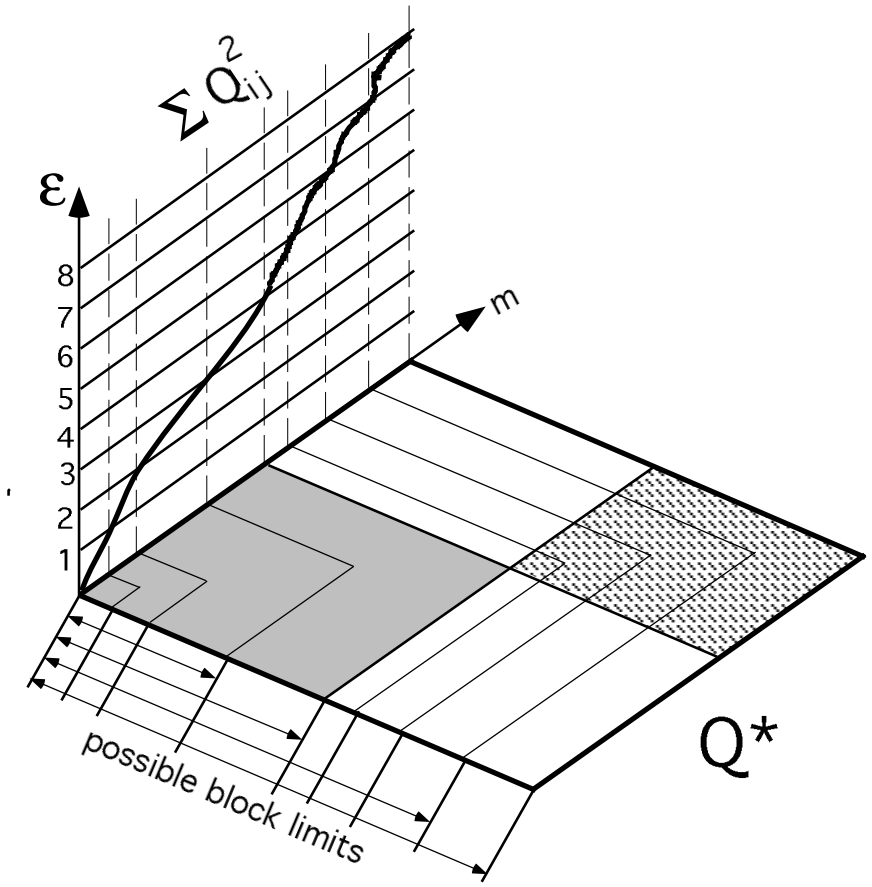


Figure 4.5: Evolution of the norm of $\mathbf{Q}*$

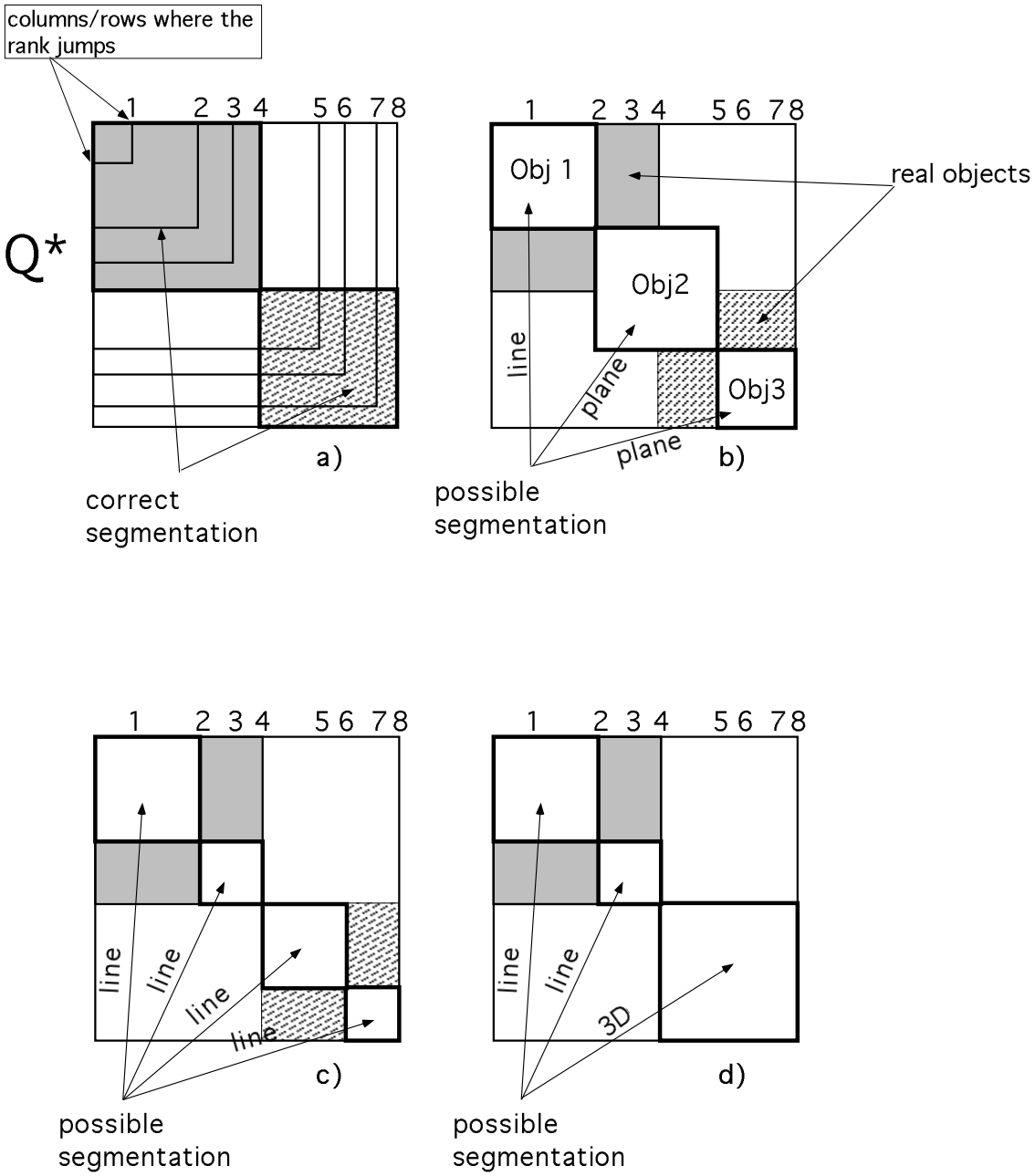


Figure 4.6: Possible $\mathbf{Q}_K^*$'s for a rank 8 $\mathbf{Q}^*$: (a) One line and two planes. (b) Four lines. (c) Two 3D objects. (d) One full 3D object and two lines.

jumps form object one and the remaining four the second object. This is also the correct solution.

4.6(b) -One object with rank 2 and two objects with rank 3. In other words, the scene is made of one moving line and two planes. These objects also form a shape interaction matrix with rank 8. In the figure we show one possible configuration, where the first block has rank 2 (a line) and the other two blocks have rank 3 (planes).

4.6(c) -Four lines. Each of the blocks has rank 2 and represents one line. In a rank 8 $\mathbf{Q}$ we can have four of these blocks.

4.6(d) -One 3D object and two lines. In this configuration the first block has rank 3, the second has rank 2 and the third rank 4. With these three blocks two more combinations are possible.

Considering all possible combinations, the correct solution is easily determined through the energy maximization of the blocks. Since the total energy of $\mathbf{Q}^*$ is equal to the constant

$$\varepsilon(N) \quad = \quad \|\mathbf{Q}^*\|_F^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} Q_{ij}^* = rank(\mathbf{Q}), \tag{4.34}$$

we can divide it into the energy of the blocks and the energy of the off-diagonal. The best solution is then the one which concentrates most energy in the blocks. In summary, the detection process is a constrained optimization process which maximizes the energy of the blocks subject to the constraint that each block represents a physical object (line, plane or full 3D).

4.4.3 Interpretation of the Cost Function

The algorithm described in the previous sections has an interesting interpretation. This interpretation will support the decision reduce the search effort by using a hill-climbing strategy. We will show that the hill climbing strategy finds a solution that represents the whole class of all possible solutions that make $\mathbf{Q}$ block diagonal.

Assuming that we have a correctly sorted $\mathbf{Q}$, let us recall the definition of function $\varepsilon(\cdot)$ as in (4.30):

$$\varepsilon(m) = \sum_{i=1}^{m} \sum_{j=1}^{m} {Q_{ij}^*}^2. \tag{4.35}$$

Now let us compute a family of functions $\varepsilon^O(\cdot)$ where O represents the set of all "correct" shape interaction matrices. By "correct" we mean all possible block diagonal matrices that result from permutations of rows and columns of $\mathbf{Q}^*$. For segmentation purposes these matrices are in fact indistinguishable. In short, these permutations switch columns and rows of features belonging to the same object. Figure 4.7 illustrates

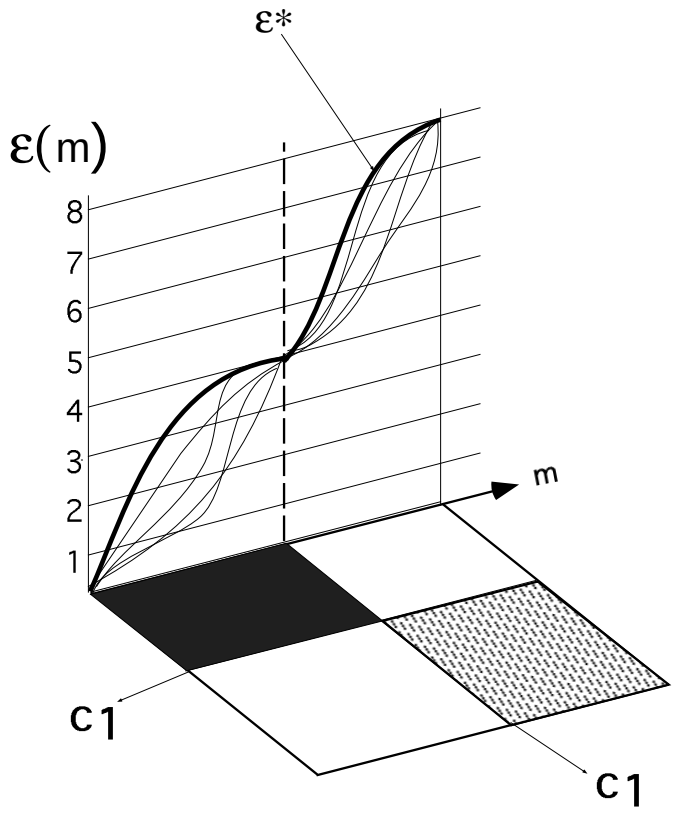


Figure 4.7: Several possibilities for energy functions. Each curve represents the energy function for a different set of permutations of the columns of $\mathbf{Q}$. Function ε^* is an upper bound of the all set of ε^i

how the set ε^O might look for the example considered throughout this section. In a noise-free environment, if $\mathbf{Q}^*$ contains M blocks, each of the functions ε^i, $(i \in O)$, has

the same value for columns, say $C = \{c_1, \cdots, c_K, \cdots, c_M\}$, that limits each of the blocks (see Figures 4.3 and 4.7). At each of these columns the function $\varepsilon^i(c_K)$ represents the total energy of the first K blocks:

$$\varepsilon^i(c_K) \;=\; \sum_{i=1}^{c_K}\sum_{j=1}^{c_K} {Q^*_{ij}}^2 = \sum_{n=1}^{K} rank(\mathbf{Q}_n). \tag{4.36}$$

Values $\varepsilon^i(c_K)$ are invariant to "correct" permutations of $\mathbf{Q}^*$ due to its block diagonal structure. In fact, they are the only important points for detecting the limits of the blocks. Among the whole family of functions, we denoted ε^* as,

$$\varepsilon^* \;=\; \max_{\forall_{i \in O}}(\varepsilon^i), \tag{4.37}$$

which bounds the whole set, and is represented by the thick curve in Figure 4.7. Then, function ε^* maximizes the energy of any submatrix of $\mathbf{Q}^*$ formed by its first m columns and rows. Since values $\varepsilon^i(K)$ contain all the information needed for block detection, and are invariant to permutations that block diagonalize $\mathbf{Q}$, all functions $\varepsilon^i()$ can be represented by ε^* without any loss of information. As we showed in section 4.4.1, function $\varepsilon^*()$ can be computed by a hill-climbing search, thus reducing the search space to polynomial complexity. Due to noise, the energy of the block will not be a whole number at the block's border. As Figure 4.8 shows, at the block limiting columns, the value of the energy will exhibit a small difference δ from the integer crossing, which is the energy of the off-diagonal elements. Since we do not have a statistical description of noise, we cannot compute an estimate of δ and use it to modify the threshold of the possible block limiting columns. However, this limitation can be easily overcome. The energy of noise induces an uncertainty α in the position of the block limiting column (see figure 4.8). In other words, we do not know whether feature c_K, for which $\varepsilon^*(c_K)$ is integer, belongs to the previous block or the following one. By testing the linear dependence between feature c_K and the neighbouring ones, we can determine to which of the blocks it is closer.

Finally, recall that one of the reasons we did not use graph theoretical algorithms was because they rely on local information. Hence, the segmentation is done based on

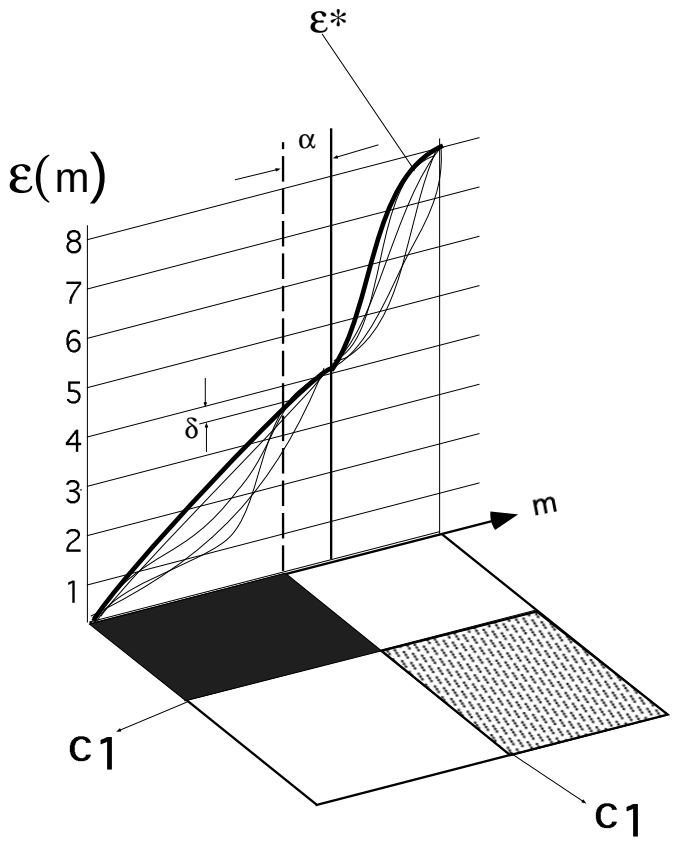


Figure 4.8: Noisy $\mathbf{Q}^*$. Funtion ε^* deviates due to the energy spread in the off-diagonal

relationships between individual features, making the sorting quite sensitive to noisy features. As a consequence, due to a single strong noise spike between two features belonging to different objects, the MST algorithm joins two different objects through that link.

Instead, with our algorithm, the effect of a single noisy feature is smoothed. Assume

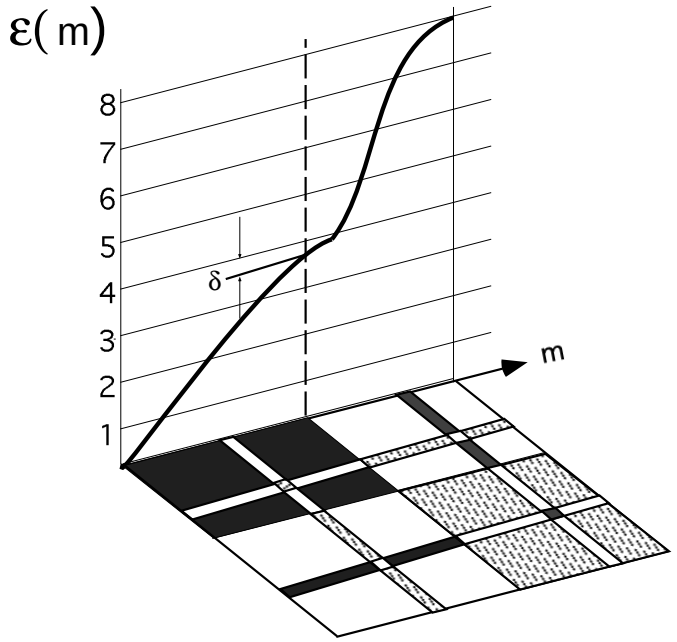


Figure 4.9: Noisy **Q** with misclassification of two features.

that there is one feature whose energy of interaction with features of a different object is high. Then, since we are sorting features by the strength of their coupling, the noisy elements will be understood (and sorted) as signal. This is illustrated in Figure 4.9, where the correct column and row of a feature has been swapped with the column and row of a feature belonging to another object. If the block has $N_k \times N_k$ elements, the number of elements that have been wrongly swapped is $2N_k - 1$ (one row and one column), that is the ratio of noisy elements over the total size of the block is $(2N_k - 1)/N_k^2$. If $N_k \gg 1$, the influence of noise in the function $\varepsilon^*(\cdot)$ is of the order of $1/N_k$ of the noise to signal ratio. In summary, the fact that we use global constraints to sort the matrix **Q** by maximization of the energy of all its submatrices, produces

a smoothing effect on noise, making the process more reliable against individual noise spikes.

4.5 Summary of Algorithm

Now the of our algorithm can be summarized as the sequence of the following steps:

1. Run the tracking process and create matrix $\mathbf{W}$
2. Compute $r = rank(\mathbf{W})$
3. Decompose the matrix $\mathbf{W}$ using SVD, and yielding $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$
4. Compute the shape interaction matrix $\mathbf{Q}$ using the first r rows of $\mathbf{V}^T$
5. Block-diagonalize $\mathbf{Q}$
6. Permute matrix $\mathbf{V}^T$ into submatrices corresponding to a single object each
7. Compute $\mathbf{A}_i$ for each object, and obtain the corresponding shape and motion.

It should be clear by now that the segmentation algorithm presented above is independent of the number of objects, that is, the block diagonal structure of $\mathbf{Q}^*$ is valid for an arbitrary number of moving objects. Furthermore, this property holds also when the shape matrix of the objects has rank less than 4 (planes and lines) so that the total rank of $\mathbf{W}$ is the only required prior knowledge. Finally note that instead of permuting columns of $\mathbf{W}$ in step 6 we permute columns of $\mathbf{V}^T$, which is equivalent.

Chapter 5

Experiments

In order to test the algorithm's performance under increasingly demanding conditions, three sets of experiments were planned. In the first, the feature trajectories are synthetically generated. In the second they are extracted from real images taken in the laboratory under controlled imaging conditions, and in the third the real images are acquired in a highly noisy outdoor scene, where tracking is unreliable.

5.1 Experiment 1: Synthetic Data

Figure 5.1 shows the 3D synthetic scene. It contains three transparent objects in front of each other moving independently. A static camera takes 100 images during the motion. The closest object to the camera is planar (rank 3) and the other two are full 3D objects (rank 4). So this is in fact a shape-degenerate case. Each object

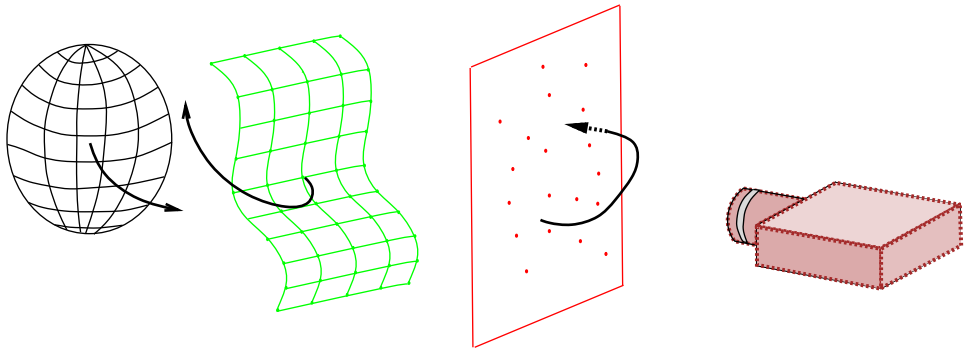


Figure 5.1: Synthetic scene. Three objects move transparently with arbitrary motion

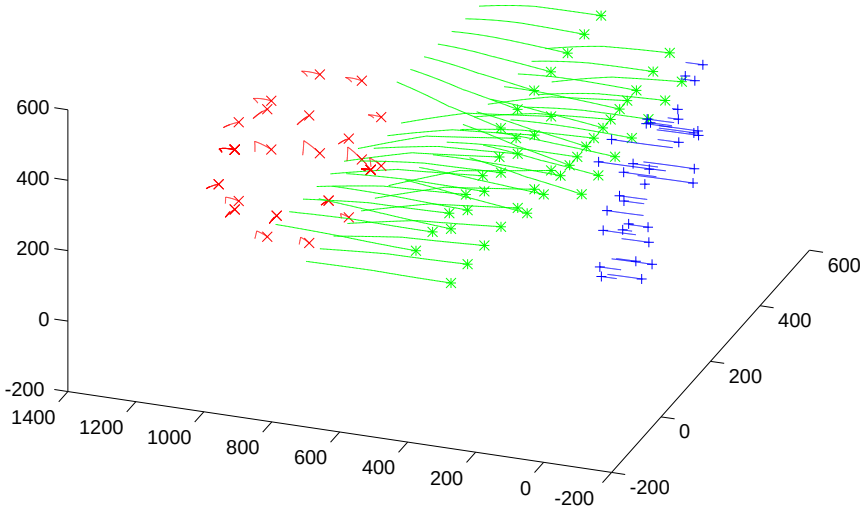


Figure 5.2: 3D trajectories of the points

translates slightly and rotates over its own centroid in such a way that the features of all objects are completely intermingled in the image plane. This complexity is intentionally introduced in order to demonstrate the fact that our motion segmentation and recovery method does not use any local information in the images. A total of one hundred and eighteen (118) points are chosen: 33 features from the first object, 49 from the second, and 36 from the third. Figure 5.2 illustrates the 3D motions of those 118 points.

The projections of the 118 scene points onto the image plane during the motion, that is, the simulated trajectories of tracked image features, are displayed in figure 5.2 with a different color for each object. Independently distributed Gaussian noise with one pixel of variance was added to the image feature positions to simulate errors in feature tracking (Figure 5.3). Of course, the identities of the features are assumed unknown, so the measurement matrix created by randomly ordering the features was given to the algorithm.

Figure 5.4 (a) shows the shape interaction matrix $\mathbf{Q}$: the height is the square of the entry value. The result of sorting the matrix into a block diagonal form is shown in Figure 5.4(b). We can observe the three blocks corresponding to objects 3, 2 and 1: the

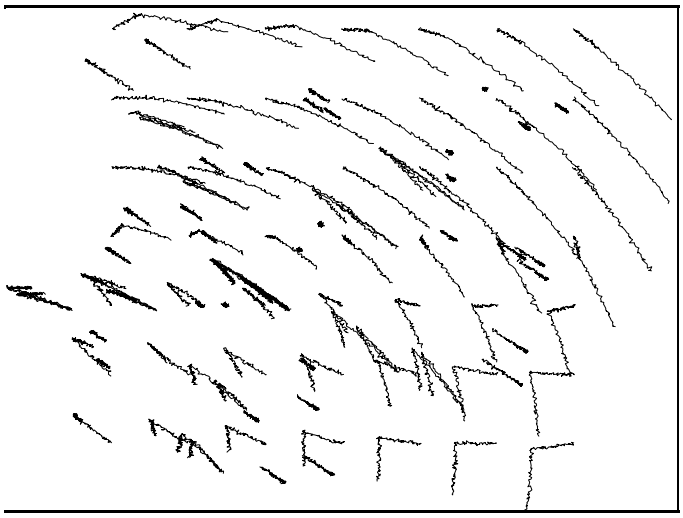
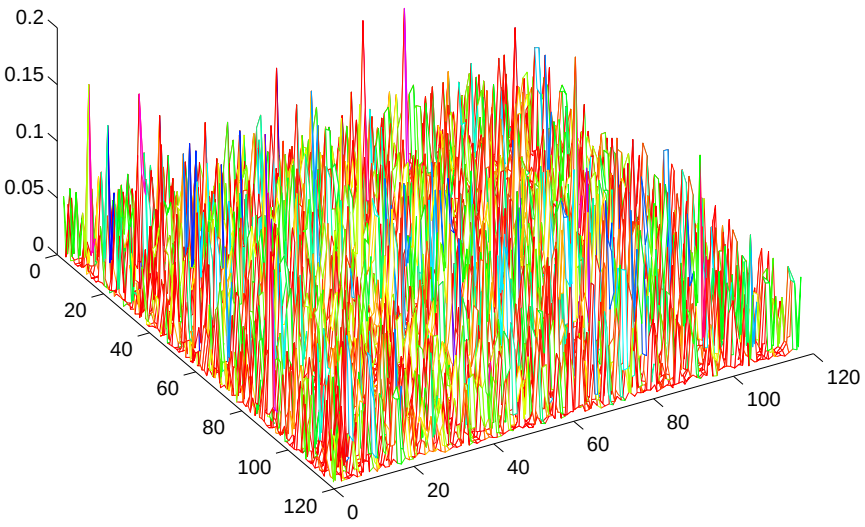
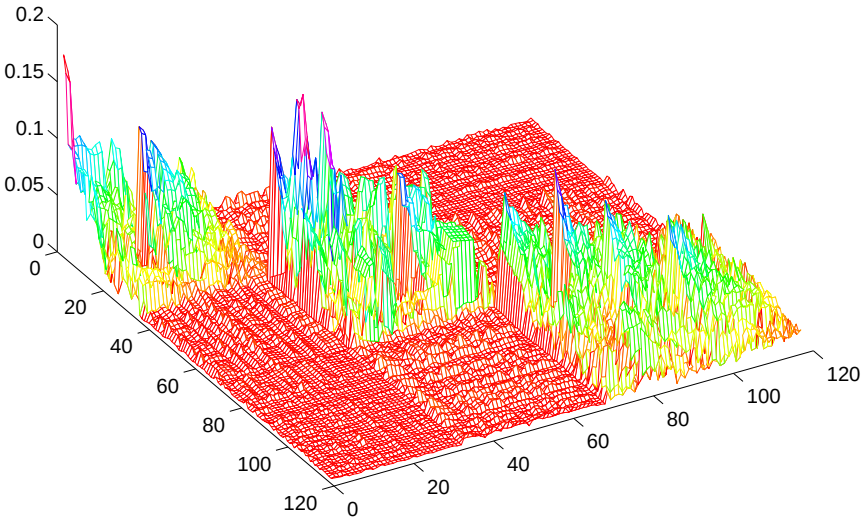


Figure 5.3: Noisy image tracks

118 features are correctly sorted. Regarding the block detection algorithm, figure 5.5 shows the profile of the energy function ε^*. The vertical lines indicate the rank jumps. Object 1 is the plane and the graph shows that the energy of the submatrix made up by the first 36 rows and columns of $\mathbf{Q}^*$ has norm 3, which is in fact the rank of object 1. Likewise, the first 85 rows and columns form a matrix with energy 7 which is the result of adding a rank 4 object (the wavy object). Finally the features of the spherical object are the last to be sorted, again producing an increase of 4 in the energy. Figures 5.6, 5.7 and 5.8 show one view of each of the recovered shapes of the three objects in the same order as in figure 5.2. Figure 5.8, showing the planar object viewed edge-on, indicates a correct recovery.



(a)



(b)

Figure 5.4: (a)Unsorted and (b) sorted shape interaction matrix

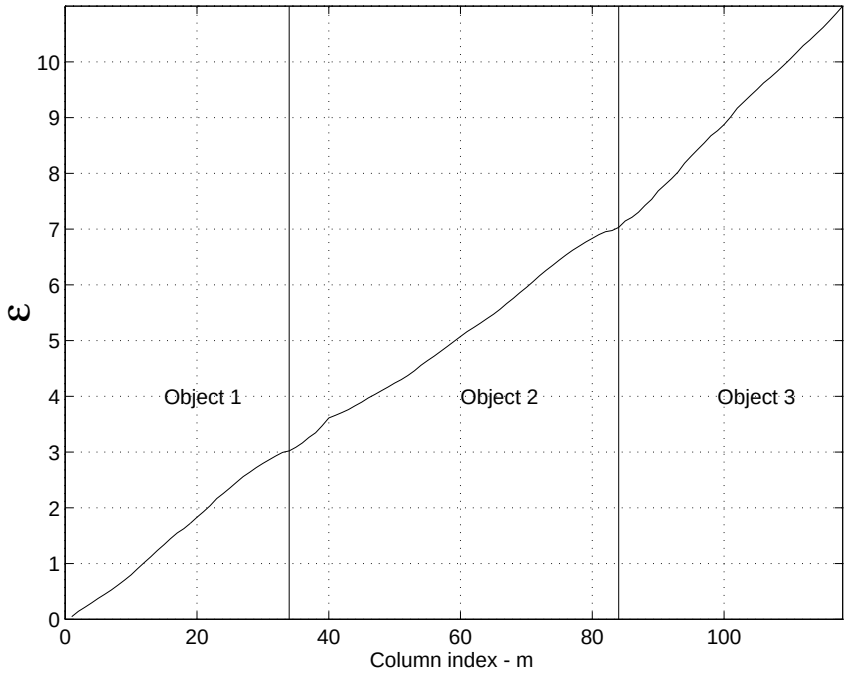


Figure 5.5: The energy function ε^*

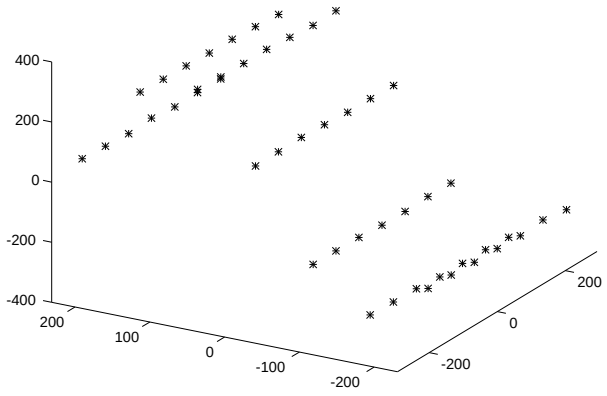


Figure 5.6: Recovered shape of the wavy object

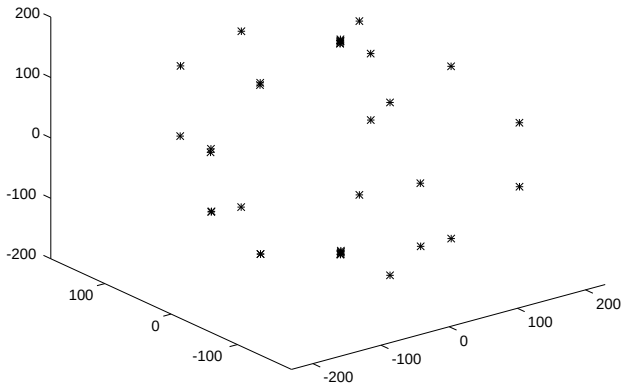


Figure 5.7: Recovered shape of the spherical object

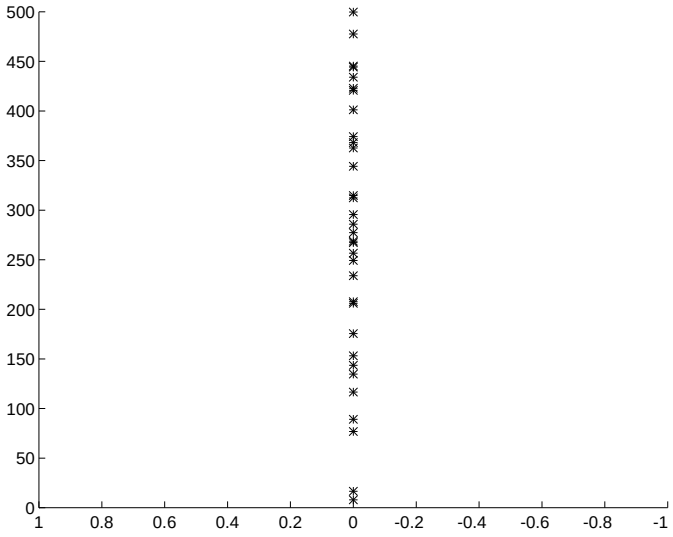


Figure 5.8: Recovered shape of the planar object

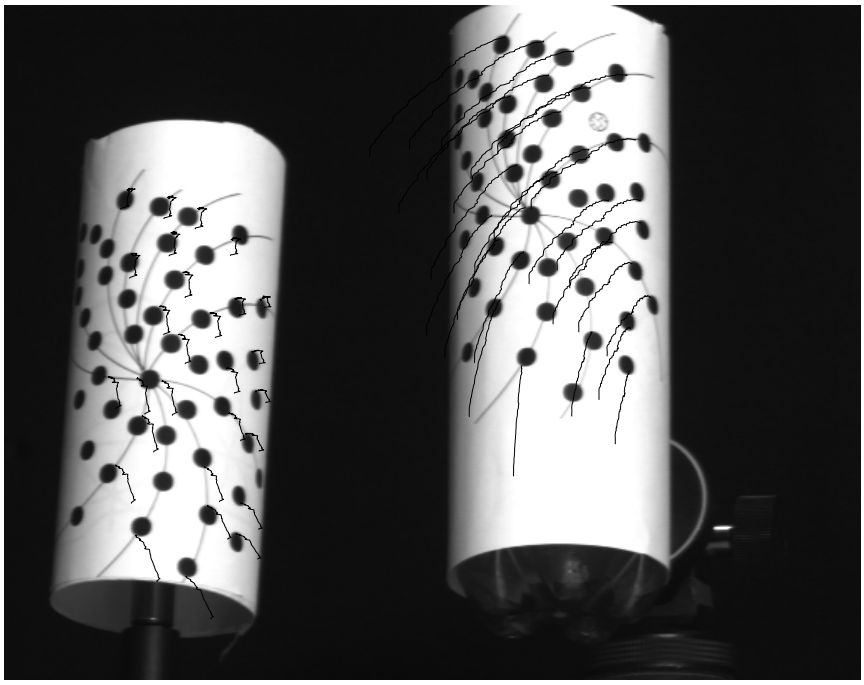
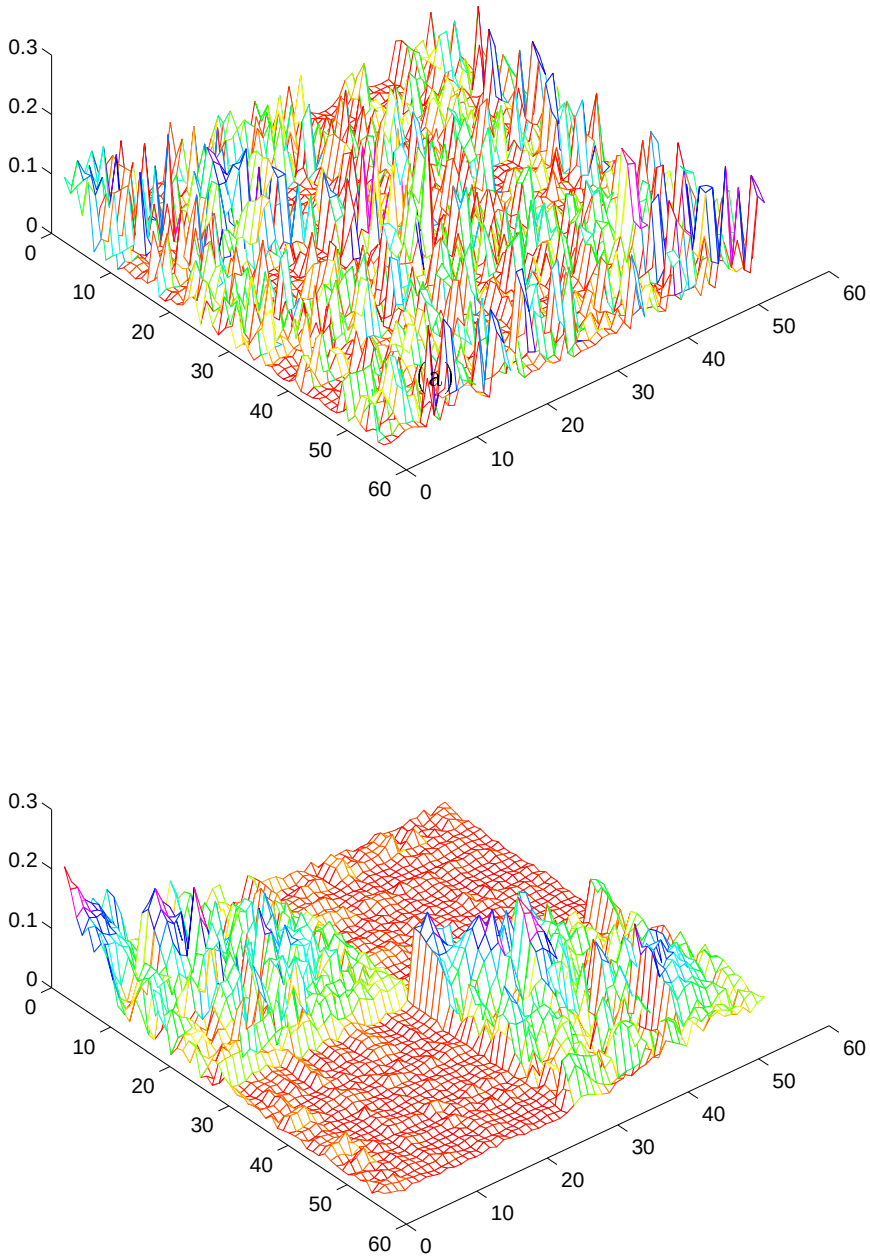


Figure 5.9: Image of the objects and feature tracks

5.2 Experiment 2: Laboratory Data

The laboratory scene consists of two roughly cylindrical shapes made by rolling cardboard sheets and drawing dots on the surface to create reliable features. The cylinder on the right tilts and rotates in the plane parallel to the image plane while the cylinder on the lefthand side rotates around its axis. The 85 images were taken by a camera equipped with a telephoto lens to approximate orthographic projections, and lighting was controlled to provide the best image quality. A total of 55 features are detected and tracked throughout the sequence: 27 belonging to the left cylinder and 28 to the other. Of course the algorithm was not given this information. Figure 5.9 shows the first 85-frame sequence, and the tracks of the selected features are shown as superimposed graphics. The scene is well approximated by orthography and the tracking was very reliable due to the high quality of the images.

Figure 5.10 (a) shows the shape interaction matrix $\mathbf{Q}$ for the unsorted input features. The sorted block diagonal matrix $\mathbf{Q}^*$ is shown in Figure 5.10 (b), and the features were correctly grouped for individual shape recovery. The resultant three-dimensional



(b)

Figure 5.10: The shape interaction matrix for Experiment 2: (a) - Unsorted $\mathbf{Q}$.(b) - The block diagonal $\mathbf{Q}^*$

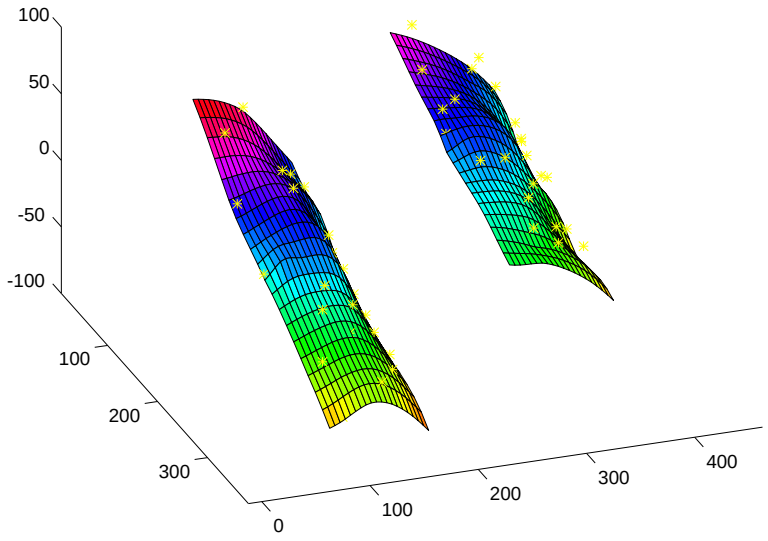
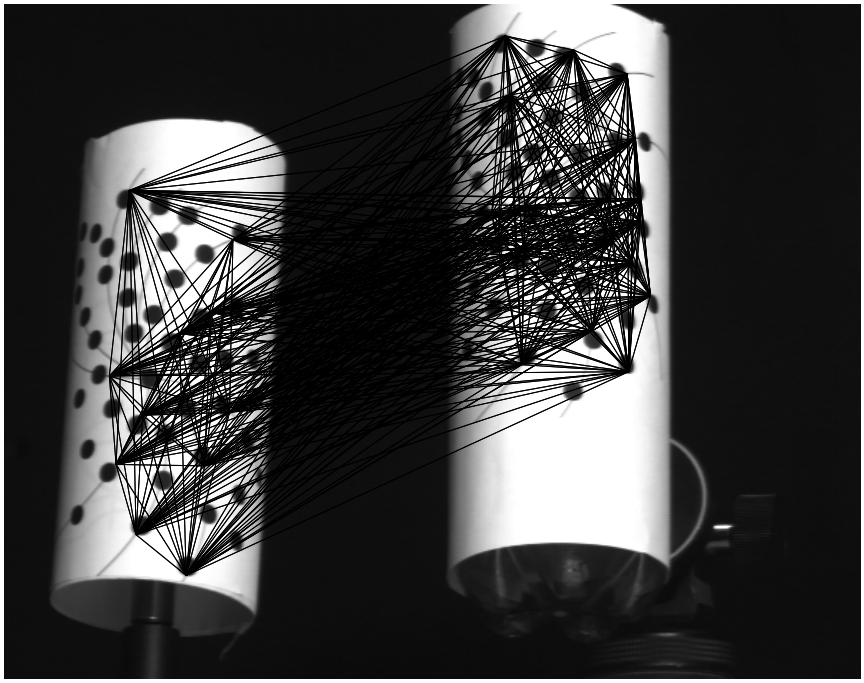


Figure 5.11: The recovered shape of the two cylinders

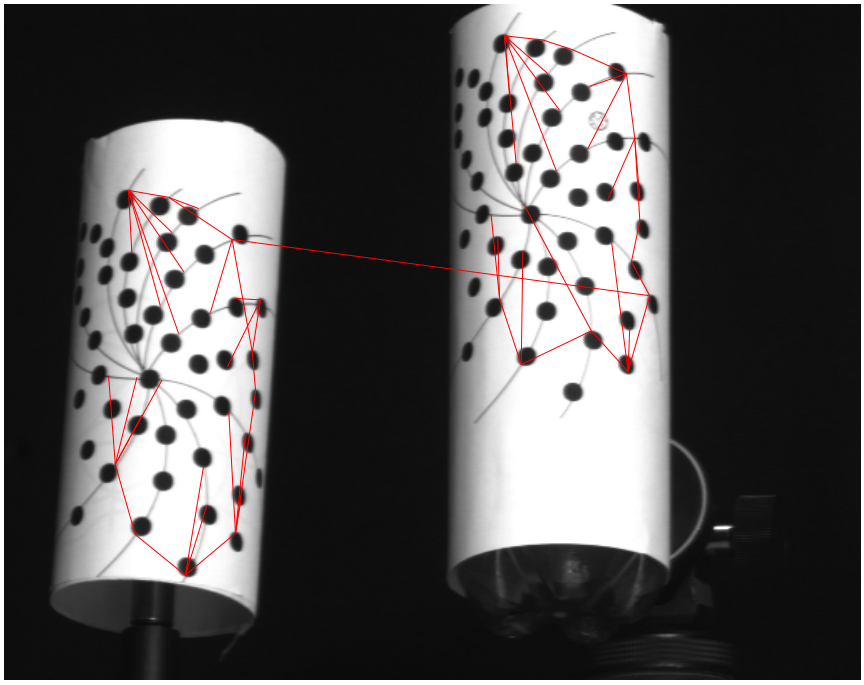
points were linearly interpolated to produce the surface of figure 5.11 in order to convey a better perception of their shape. The actual 3D feature position is shown by the asterisks (*) over the surfaces.

To show the theoretical and practical validity of the graph interpretation of matrix $\mathbf{Q}$, we also implemented the Maximum Spanning Tree algorithm. Figure 5.12(a) shows the links between all features. The weight of an edge, linking features i and j, is given by Q_{ij}^2. Computing the MST we obtain the tree of figure 5.12(b).

The MST is formed of two subtrees with features of one object only and linked by one noisy edge. This edge is the one with the least weight (minimum energy) in the whole tree. Then, segmentation is obtained by removing a number of edges (in ascending order starting from the minimum energy edge) which depends on the number of objects. The general solution is computationally intricate. Above all, the reliability of the MST against noise spikes is very low: if two features belonging to each one of the cylinders were linked by a strong noisy link, the MST algorithm would select it instead of a correct link. The final tree would have two noisy edges linking both cylinders, instead of one. Since for this scene only one edge should be removed, the two subtrees



(a)



(b)

Figure 5.12: (a) - The graph representing **Q**. (b) - The Maximum Spanning Tree

Figure 5.13: First image with tracks

would have never be obtained.

5.3 Experiment 3: Noisy Outdoor Scene

In this section we show some tests done with images taken in an outdoor scene. The main difficulty in the analysis of this type of scenes is the lack of tracking reliability. Feature tracking is particularly noisy in this scene for two main reasons. First, the camera we used has low signal to noise ratios. Second, unlike laboratory scenes, we cannot control the texture and illumination of the scene so the selected features seldom verify the assumptions made beforehand. Also, the objects in this scene move in a particular fashion highligting the shape and motion degeneracy problems.

Figure 5.13 shows the first of the 72 images of the sequence. The scene is formed by a moving face in the foreground and a still background. The unreliability of the

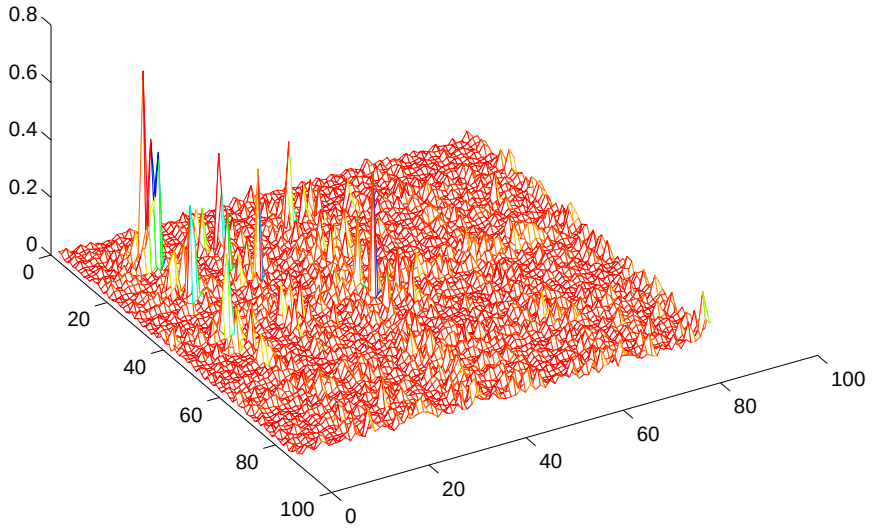


Figure 5.14: The unsorted shape interaction matrix for the outdoor scene

tracking can be observed on the building in the background. Since they are still, the tracks should look like a dot or a small cloud. However, we observe that some tracks have a small linear motion, considerably greater than the one pixel variance which is the maximum that can be expected in laboratory environments. In this case the rank of $\mathbf{W}$ was 5. The shape interaction matrix for this scene can be observed unsorted in figure 5.14 and sorted in figure 5.15.

Notice the few peaks in the upper left corner of the matrix in contrast with the generally flat pattern. This is due to the fact that the total energy of the matrix is constant and the number of features is unbalanced between the two objects. Recall from previous discussions that, for features i and j belonging to the same object, Q_{ij} is given (again for a particular configuration of the object's coordinate system) by:

$$Q_{ij} = S_i \mathbf{\Lambda}^{-1} S_j \tag{5.1}$$

$$\mathbf{\Lambda} = \mathbf{S}\mathbf{S}^T \tag{5.2}$$

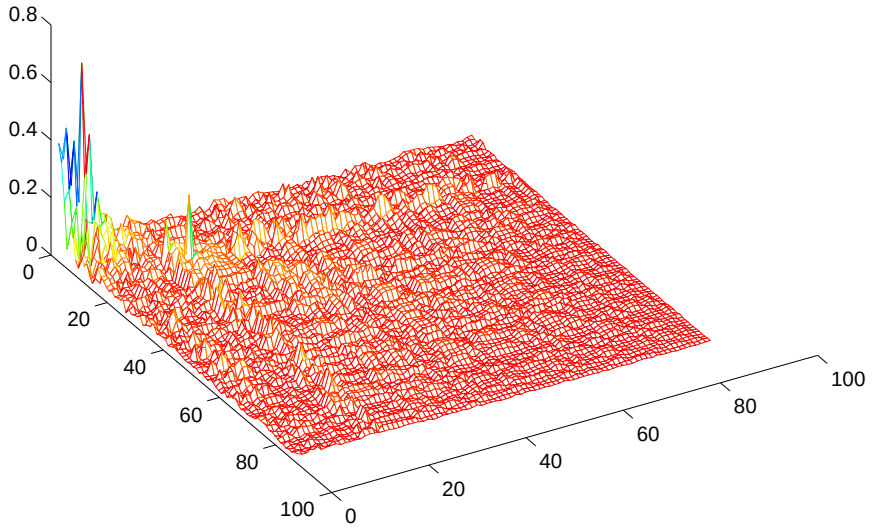


Figure 5.15: The sorted shape interaction matrix for the outdoor scene.

$$= \begin{bmatrix} \sum X_n^2 & 0 & 0 & 0 \\ 0 & \sum Y_n^2 & 0 & 0 \\ 0 & 0 & \sum Z_n^2 & 0 \\ 0 & 0 & 0 & N \end{bmatrix} \tag{5.3}$$

$$\tag{5.4}$$

The norm of matrix $\mathbf{\Lambda}$ increases, in general, with the number of points. The value of Q_{ij}, which depends on the inverse of $\mathbf{\Lambda}$, decreases with the same number. This is one of the drawbacks of the methodology for large numbers of features. In Chapter 6 we will see that noise energy grows with the size of the measurements matrix, and since the total energy of $\mathbf{Q}$ is constant, for larger numbers of features the noise influence becomes more important.

Nevertheless, the segmentation algorithm has performed without error. In Figure 5.16 we show the energy function ε^* for the sorted $\mathbf{Q}^*$. As we can see, feature number 5, for which ε^* crosses the level 2, bounds the foreground block, providing the segmentation shown in Figure 5.17. The small squares indicate the feature position and the respective tracking window, and the numbers list the sequence by which features were

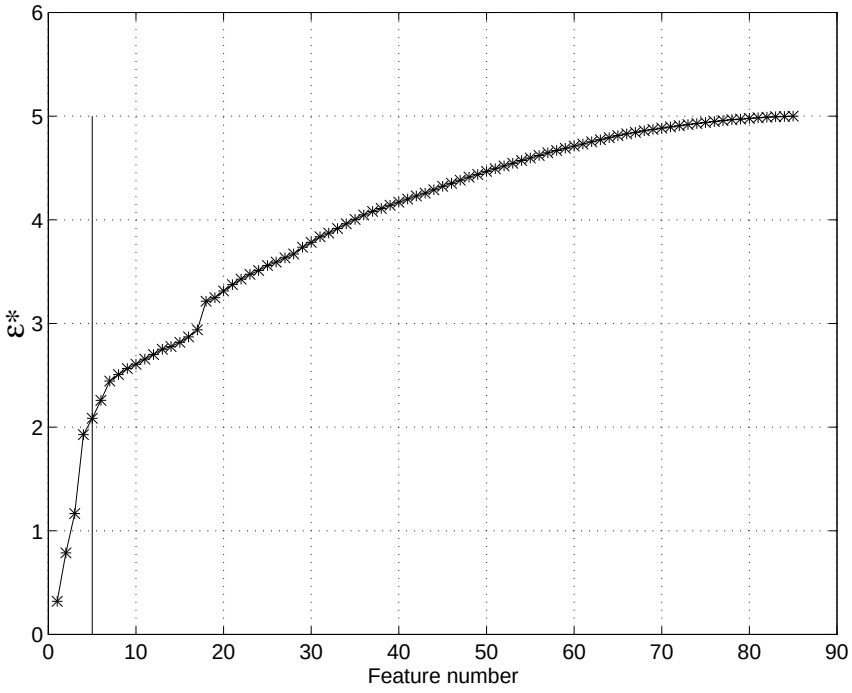


Figure 5.16: The profile of $\varepsilon^*()$.

sorted.

From the profile of ε^* we conclude that the detection algorithm segmented the scene into a rank-2 (foreground) and a rank-3 (background) object.

Recall that the background does not move. Even though the shape matrix of the background has rank 4 (3D object), its motion matrix has rank 2. In fact, with the appropriate set of coordinate frames and noting that translation is zero, the motion matrix of the background can be written as

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & t_{x_1} \\ & \vdots & & \\ 1 & 0 & 0 & t_{x_1} \\ 0 & 1 & 0 & t_{y_1} \\ & \vdots & & \\ 0 & 1 & 0 & t_{y_1} \end{bmatrix}, \tag{5.5}$$

where t_{x_1} and t_{y_1} are the coordinates of the background centroid. From (5.5) we conclude that only the first two columns are linearly independent, therefore instead of a 3D object the measurements generate a rank 2 matrix (equivalent to a line). However,

Figure 5.17: List of the sorted features. Note the correct sequence of the features. The vertical line of Figure 5.16 shows that the first four features make a rank-2 block

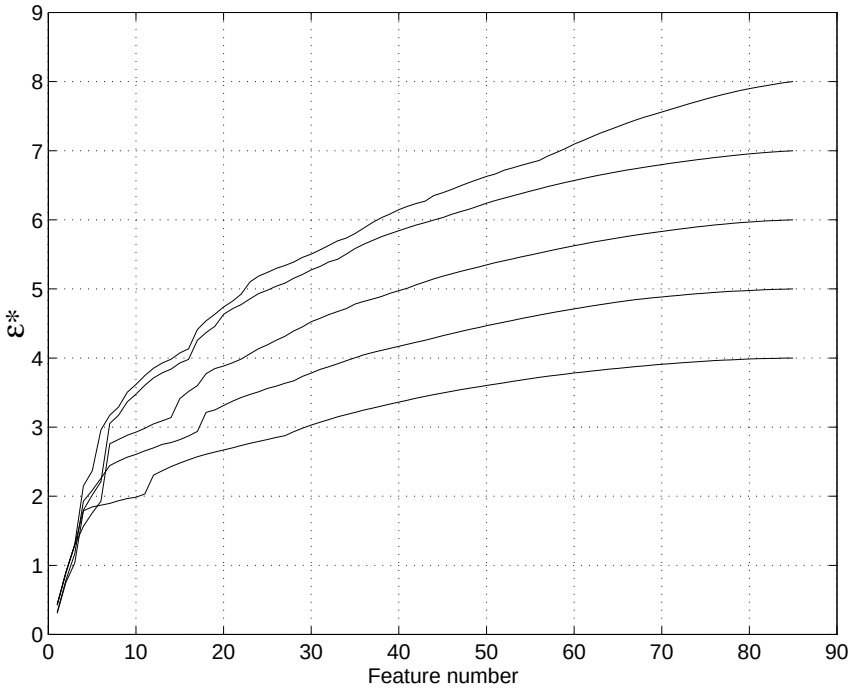


Figure 5.18: Energy function for five different assumed ranks for matrix $\mathbf{Q}^*$

since the centroid is not still, there is a noisy translational component, measurable in the singular value decomposition of $\mathbf{W}$ that may increase the detectable rank to 3. The same situation applyes to the foreground, where the motion is, essentially, a rotation around the axis perpendicular to the image plane.

To verify the sensitivity of the algorithm, in this particularly harsh case, we run the sorting and detection algorithms for different values of the total rank of $\mathbf{W}$. Figure 5.18 shows five curves, representing $\varepsilon^*()$ for matrices $\mathbf{Q}^*$ with ranks 4, 5, 6, 7 and 8 (all possibilities with two objects). In all cases the sorting was correct, that is, features were sorted in the same sequence shown in figure 5.17: First come the foreground features and in the same order, and then the background features, however the segmentation outcome differed. Note the common and steep profile of ε^* of the first 4 features (belonging to the foreground) which contain all the energy of the subspace of the foreground shape. Then, profiles differ almost of a constant bias.

Chapter 6

Computing the Rank of Matrix W

In the previous theoretical developments we assumed that the rank of the measurements matrix, $\mathbf{W}$, was known. In order to build the shape interaction matrix, $\mathbf{Q}$, this knowledge is essential since the rank specifies the number of singular vectors $\mathbf{V}$ from which $\mathbf{Q}$ is created. Due to camera noise and other tracking errors, the rank of matrix $\mathbf{W}$ will, in general, differ from the correct one. Therefore, we need a rank determination procedure which selects the significant singular values and vectors.

Several approaches have been developed regarding the subject of signal/noise separation. One of the most efficient of all is the MUSIC algorithm [BK79, Sch80]. This algorithm decomposes a measurement matrix into two matrices whose columns are in orthogonal subspaces, one containing the signal and another the noise, provided we know the covariance of the noise. This algorithm is quite robust but the coordinates used to represent the signal are inconvenient for our purposes and hide the block diagonal properties (of $\mathbf{Q}$) we are looking for. Since we use SVD to build our constructs, and given its rank revealing properties and numerical stability, we closely follow the approach of [Ste92a], formalizing the rank determination problem under the SVD framework and including uncertainty models of the feature tracking process.

The rank of the observations matrix, $\mathbf{W}$, is determined by an approximation criterion for which the noise model is required. We model the noisy imaging process as the result of additive noise to the projection equations (2.7)[Wil94]. Then, the ith feature

position, in frame f, is given by

$$\begin{bmatrix} \tilde{u}_{f,i} \\ \tilde{v}_{f,i} \end{bmatrix} = \begin{bmatrix} u_{f,i} \\ v_{f,i} \end{bmatrix} + \begin{bmatrix} \varepsilon_{u_{f,i}} \\ \varepsilon_{v_{fi}} \end{bmatrix}, \tag{6.1}$$

where $\varepsilon_{u_{fi}}$ and $\varepsilon_{v_{fi}}$ are the additive noise components in the X and Y directions of the image plane. From equation (6.1) we foresee the need for feature tracking modeling, namely the characterization of the feature position uncertainty. To make the procedure more clear we will develop the general rank determination approach and we will then introduce the tracking specifics.

The $2F \times N$ noisy measurement matrix, $\tilde{\mathbf{W}}$, in this case, will be given by the matrix sum

$$\tilde{\mathbf{W}} = \mathbf{MS} + \begin{bmatrix} \mathcal{E}_u \\ \mathcal{E}_v \end{bmatrix} \tag{6.2}$$

$$\tilde{\mathbf{W}} = \mathbf{W} + \mathcal{E}, \tag{6.3}$$

where $\mathcal{E}_u$ and $\mathcal{E}_v$ are two $F \times N$ matrices and $\mathbf{W}$ is the noise-free measurement matrix, whose rank we want to estimate. It is now clear from (6.3) that the rank of $\tilde{\mathbf{W}}$ can be at most N. Then, the singular value decomposition will be given by:

$$\tilde{\mathbf{W}} = \mathbf{U}\mathbf{\Sigma}V^T \tag{6.4}$$

$$\mathbf{\Sigma} = diag(\sigma_1, \cdots, \sigma_N) \tag{6.5}$$

$$\mathbf{U} \in R^{2F \times N} \tag{6.6}$$

$$\mathbf{V} \in R^{N \times N}. \tag{6.7}$$

In a noise free environment the rank of $\mathbf{W}$ is determined by the number of non zero singular values σ_i, whereas in the noisy case we will have to compute a threshold that separates the noisy singular values from the significant ones. From the SVD decomposition of $\tilde{\mathbf{W}}$ we have to estimate the number of columns of $\mathbf{U}$,$\mathbf{V}$ and the singular values σ_i that represent an estimate $\hat{\mathbf{W}}$ of $\mathbf{W}$.

6.1 Matrix Approximation

The singular value decomposition of $\tilde{\mathbf{W}}$ "spreads" the noise components over all the elements of $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{\Sigma}$. In other words, it is not possible to isolate the noise-free components of these matrices or even directly estimate noise influence. Then, we will seek an estimate $\hat{\mathbf{W}}$, with the same rank of $\mathbf{W}$, that approximates $\tilde{\mathbf{W}}$ in the least squares sense [Ste92a, Dem87, GHS87]. In other words, we must solve two problems here: we have to determine the "real" rank of $\tilde{\mathbf{W}}$ and also obtain an approximation of the "real" observations matrix $\mathbf{W}$. Then, following the minimum error criterion, if r is the rank of the noise-free measurement matrix $\mathbf{W}$, for all possible $2F \times N$ matrices $\mathbf{Y}$ with $rank(\mathbf{Y}) \leq r$, we seek an estimate $\hat{\mathbf{W}}$ that minimizes the error:

$$\|\tilde{\mathbf{W}} - \hat{\mathbf{W}}\|_F^2 = \min_{rank(\mathbf{Y}) \leq r} \|\tilde{\mathbf{W}} - \mathbf{Y}\|_F^2. \tag{6.8}$$

Fischer's theorem [Ste92a] states that a solution $\hat{\mathbf{W}}$ exists, and the error (6.8) is explicitly given by:

$$\|\tilde{\mathbf{W}} - \hat{\mathbf{W}}\|_F^2 = \sigma_{r+1}^2 + \cdots + \sigma_N^2. \tag{6.9}$$

Expression (6.9) is equivalent to saying that the approximation given by

$$\hat{\mathbf{W}} = \mathbf{U}_{2F \times r} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \mathbf{V}_{N \times r}^T \tag{6.10}$$

is the minimum error approximation of $\tilde{\mathbf{W}}$ by a rank r matrix. Comparing with the "correct" equations, we have the following correspondence:

$$\tilde{\mathbf{W}} = \mathbf{W} + \mathcal{E} \tag{6.11}$$

$$\tilde{\mathbf{W}} = \hat{\mathbf{W}} + \hat{\mathcal{E}} \tag{6.12}$$

$$\tilde{\mathbf{W}} = \mathbf{U}_{2F,1:r} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \mathbf{V}_{N,1:r}^T + \mathbf{U}_{2F,r+1:N} \begin{bmatrix} \sigma_{r+1} & & \\ & \ddots & \\ & & \sigma_N \end{bmatrix} \mathbf{V}_{N,r+1:N}^T \tag{6.13}$$

where the notation $_{i:j}$ denotes column or row range. Since $\hat{\mathbf{W}}$ is the closest matrix to $\tilde{\mathbf{W}}$ with rank r, its error is minimum; in particular, it is smaller than the error in the real measurements $\mathbf{W}$, that is:

$$\|\tilde{\mathbf{W}} - \hat{\mathbf{W}}\|_F^2 \leq \|\tilde{\mathbf{W}} - \mathbf{W}\|_F^2 = \|\mathcal{E}\|_F^2. \tag{6.14}$$

Using (6.9) in (6.14) yields:

$$\sigma_{r+1}^2 + \cdots + \sigma_N^2 \leq \|\mathcal{E}\|_F^2 = \sum_{i=1}^{2F}\sum_{j=1}^{N} \varepsilon_{ij}^2. \tag{6.15}$$

Using this relation and knowing the magnitude (norm) of the noise matrix, we can define the rank of $\mathbf{W}$ as the smallest integer, r, such that inequality (6.15) holds, or equivalently, the smallest integer r, for which the sum of the last $N - r$ singular values of the noisy matrix is less than or equal to the norm of the noise matrix.

However, there is one problem with this strategy: the entries of matrix $\mathcal{E}$ are stochastic variables, therefore we do not know their absolute values (realizations). If the feature tracking provided a statistical description of the feature positions we could have a decision based on mean values of the noise component. Then, if we compute the mean of equation (6.15) we obtain the relation:

$$E\left[\|\tilde{\mathbf{W}} - \hat{\mathbf{W}}\|_F^2\right] = E\left[\sigma_{k+1}^2\right] + \cdots + E\left[\sigma_N^2\right] \leq E\left[\|\mathcal{E}\|_F^2\right] \tag{6.16}$$

$$\leq \sum_{i=1}^{N}\sum_{j=1}^{2F} E\left[\varepsilon_{ij}^2\right]. \tag{6.17}$$

The terms ε_{ij} in (6.17) are the statistical second moment (variance) of the feature noise. Using the notation as in (6.1), the covariance of the noise is given by:

$$\sum_{i=1}^{N}\sum_{j=1}^{2F} E\left[\varepsilon_{ij}^2\right] = \sum_{i=1}^{N}\sum_{f=1}^{F}(E[\varepsilon_{u_{f,i}}^2] + E[\varepsilon_{v_{f,i}}^2]). \tag{6.18}$$

In general, for each sampling time, the uncertainty of each feature point is characterized by a 2×2 covariance matrix

$$\mathbf{\Pi}_{f,i} = E\left(\begin{bmatrix} \varepsilon_{u_{fi}} \\ \varepsilon_{v_{fi}} \end{bmatrix} \begin{bmatrix} \varepsilon_{u_{fi}} & \varepsilon_{v_{fi}} \end{bmatrix}\right) \tag{6.19}$$

$$= \begin{bmatrix} E[\varepsilon_{u_{fi}}^2] & E[\varepsilon_{u_{fi}}\varepsilon_{v_{fi}}] \\ E[\varepsilon_{u_{fi}}\varepsilon_{v_{fi}}] & E[\varepsilon_{v_{fi}}] \end{bmatrix} \tag{6.20}$$

$$= \begin{bmatrix} \pi_{u_{fi}}^2 & \pi_{uv_{fi}} \\ \pi_{uv_{fi}} & \pi_{v_{fi}}^2 \end{bmatrix}. \tag{6.21}$$

The error (6.17) is finally expressed as:

$$E\left[\|\tilde{\mathbf{W}} - \hat{\mathbf{W}}\|_F^2\right] = E\left[\sigma_{k+1}^2\right] + \cdots + E\left[\sigma_N^2\right] \leq \sum_{i=1}^{N}\sum_{f=1}^{F}(\pi_{u_{f,i}}^2 + \pi_{v_{fi}}^2). \tag{6.22}$$

From (6.22) we can finally describe the procedure to detect the rank of $\mathbf{W}$:

1. Decompose the real measurements matrix $\tilde{\mathbf{W}}$ using SVD.
2. Compute the sum of the last $N - r$ singular values $(\sigma_r + \cdots + \sigma_N)$.
3. Find the rank as the value of r for which $\sigma_r + \cdots + \sigma_N \leq \mathcal{T} \sum_{p=1}^{N}\sum_{f=1}^{F}(\pi_{u_{f,p}}^2 + \pi_{v_{fp}}^2)$, where $\mathcal{T}$ is an experimental constant used to adjust for unaccounted deviations and for the difference between the actual value and the expected value of σ_i.

It is now clear that we need to compute the uncertainty of the feature position, $\mathbf{\Pi}_{fi}$, in order to compute the rank and the approximation of $\mathbf{W}$. This is done in the following section.

6.2 Tracking and Uncertainty Computation

Here, we use a tracking scheme similar to that developed by [LK81] with multiresolution capabilities as in [Mad93]. To compute the rank of the matrix $\mathbf{W}$ we need to have uncertainty estimates of the feature positions, therefore we reformulated the tracking problem in order to include statistical information. The tracking is introduced considering three major issues: First, given the position of image points and their brightness values, we describe how to compute the point displacement between two consecutive images. Secondly we compute the uncertainty affecting the estimates of the displacement and use it to define the feature selection criterion. Finally, we show how to track

and compute the uncertainty along the whole image stream, that is, how to integrate each of these displacements into an image feature trajectory.

6.2.1 Tracking Between Two Frames

Given a set of image points (features), tracking consists in computing the displacement of the image points from the original image to the next. In previous chapters we related these displacements with the 3D parameters like the object's shape or motion. Now we want to compute the displacement itself from the raw brightness data.

Generally speaking, finding the image position of one point corresponding to the same physical point in another image is a search problem known as the correspondence problem in computer vision. Given an arbitrary motion of the object, the image sequence thus generated may change in such a complex way that this task may become impossible. To build a feasible feature tracker, some assumptions have to be made in order to simplify the problem. Without loss of generality, experience has shown that the following assumptions are quite reasonable, and widely used [Hee88b][HS81][WA83]:

1 The motion of the features between two consecutive images is essentially translational. This is identical to assuming smooth motion of the object.

2 The temporal sampling is such that the magnitude of the displacement between two frames is small.

3 The displacement field is smooth. Neighbouring pixels usually belong to the same feature and thus have similar motion.

To avoid unnecessary notation "clutter", we most of the time dropped the indices i and f from the variables. All the development in this section refers to a single feature, therefore indices are used only when strictly required for better understanding.

With the above assumptions in mind, let $I_f(u, v)$ define the image at the f sampling interval, (u, v) the image point coordinates (pixels), and $\mathbf{d} = [d_u \; d_v]^T$ a vector

representing the feature displacement from frame f to frame $f+1$. Using assumption **1**, the $(f+1)^{th}$ image can be expressed as a translated version of the previous image:

$$I_{f+1}(u,v) = I_f(u - d_u, v - d_v). \tag{6.23}$$

Though nonlinear, we can obtain a linearized expression for equation (6.23). Consider the Taylor expansion:

$$I_f(u - d_u, v - d_v) = I_f(u,v) - \mathbf{g} \cdot \mathbf{d} + \mathcal{O}, \tag{6.24}$$

where

$$\mathbf{g} = \begin{bmatrix} g_u \\ g_v \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial u} \\ \frac{\partial I}{\partial v} \end{bmatrix} \tag{6.25}$$

is the image gradient and $\mathcal{O}$ the higher order terms. From assumption **2**, we know that the magnitude of the interframe displacement is small ($\|\mathbf{d}\| \ll 1$): higher order terms of (6.24) are thus negligible, and discarding we obtain

$$I_f(u - d_u, v - d_v) = I_f(u,v) - \mathbf{g} \cdot \mathbf{d}. \tag{6.26}$$

Inserting (6.26) in (6.23), and rearranging the terms, we reach the following relation:

$$h \equiv I_f(u,v) - I_{f+1}(u,v) = \begin{bmatrix} g_u & g_v \end{bmatrix} \begin{bmatrix} d_u \\ d_v \end{bmatrix} \tag{6.27}$$

$$= \mathbf{g} \cdot \mathbf{d}. \tag{6.28}$$

Equation (6.28), known as the optical flow constraint equation [HS81], relates the displacement to the image time variation and the image gradient. To make equation (6.28) a more accurate description of real images we first have to represent pixels, that is, (6.28) has to be discretized. Also, real images are noisy, so we add an additional term reflecting camera perturbations. The final equation describing the effect of motion on the brightness value of each image pixel is:

$$I_f(u_i, v_i) - I_{f+1}(u_i, v_i) = [g_{u_i}\ g_{v_i}] \begin{bmatrix} d_u \\ d_v \end{bmatrix} + \eta_i \tag{6.29}$$

$$h_i = \mathbf{g}_i\, \mathbf{d} + \eta_i \tag{6.30}$$

where h_i, g_{u_i}, g_{v_i} represent pixel variables and η_i the noise term.

Both terms h_i and (g_{u_i}, g_{v_i}) are computable from two consecutive images, however (6.29) does not uniquely solve for the displacement. We need extra constraints in order to estimate **d**. Recall from assumption **3**, that the displacement field is smooth: neighbouring pixels usually belong to the same object and hence have similar motion. We can make **d** unambiguous by combining measurements from the $n \times n$ window

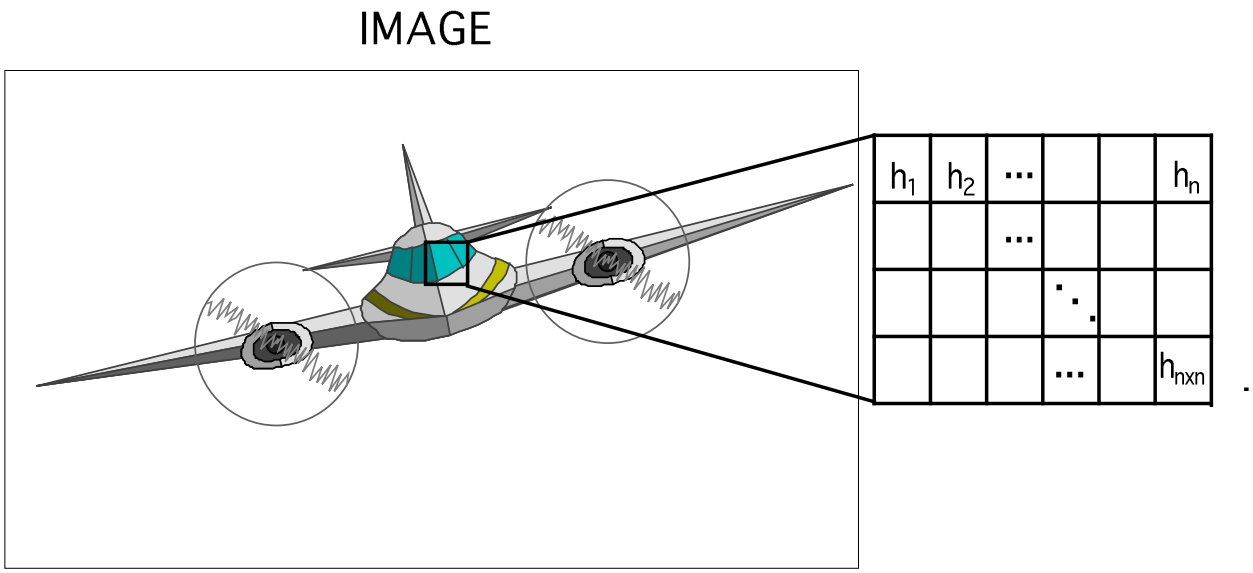


Figure 6.1: Correspondence between pixels and measurement vector

around the feature point. Figure 6.1 shows the relation between the measurements h_i and their image location. The window measurements are representable in a linear matrix equation, forming the complete tracking model:

$$\begin{bmatrix} h_1 \\ \vdots \\ h_{n\times n} \end{bmatrix} = \begin{bmatrix} g_{u_1} & g_{v_1} \\ \vdots & \vdots \\ g_{u_{n\times n}} & g_{v_{n\times n}} \end{bmatrix} \begin{bmatrix} d_u \\ d_v \end{bmatrix} + \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_n \end{bmatrix} \tag{6.31}$$

$$\mathbf{h} = \mathbf{A}\,\mathbf{d}, \tag{6.32}$$

where

$$\mathbf{A} = \begin{bmatrix} g_{u_1} & g_{v_1} \\ \vdots & \vdots \\ g_{u_{n\times n}} & g_{v_{n\times n}} \end{bmatrix} \tag{6.33}$$

is the design matrix. Given the overconstrained system of (6.31), the least square error estimate $\hat{\mathbf{d}}$, of $\mathbf{d}$, is computed from

$$\hat{\mathbf{d}} = (\mathbf{A^T A})^{-1}\mathbf{A^T}\ \mathbf{h}, \tag{6.34}$$

Equation (6.34) is a linear relation between the estimate and the measurements. Through error propagation we can compute the uncertainty of the estimate, given by the covariance $\mathbf{\Pi}_{\hat{\mathbf{d}}}$ of vector $\hat{\mathbf{d}}$:

$$\mathbf{\Pi}_{\hat{d}} = Cov\left[\hat{\mathbf{d}}\right] = E\left[\hat{\mathbf{d}}\hat{\mathbf{d}}^T\right] \tag{6.35}$$

$$= E\left[(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{h}\mathbf{h}^T\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-T}\right] \tag{6.36}$$

$$= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T E\left[\mathbf{h}\mathbf{h}^T\right]\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-T}. \tag{6.37}$$

It is reasonable to assume that the noise is spatially and temporally uncorrelated [Fau94], in other words, one pixel's brightness value is independent of the values of other pixels and also independent of their value in other time instants. Since η_i is gaussian white noise with zero mean and variance σ_I^2, and since $h_i = I_{(f+1)}(u_i, v_i) - I_f(u_i, v_i)$, the covariance of $\mathbf{h}$ is computed by

$$Cov\left[\mathbf{h}\right] = E[\mathbf{h}\mathbf{h}^T] \tag{6.38}$$

$$= \begin{bmatrix} E[h_1^2] & \cdots & E[h_1 h_n] \\ & \ddots & \\ E[h_n h_1] & \cdots & E[h_n^2] \end{bmatrix}. \tag{6.39}$$

Since the noise is spatially and temporally uncorrelated, we have

$$E[h_i h_j] = 0 \ \ (i \neq j), \tag{6.40}$$

so

$$E[h_i^2] = E\left[I_{f+1}(u_i, v_i) - I_f(u_i, v_i)\right]^2 \tag{6.41}$$

$$= E\left[I_{f+1}^2(u_i, v_i) + I_f(u_i, v_i)^2 - 2I_{f+1}(u_i, v_i)I_f(u_i, v_i)\right] \quad (6.42)$$

$$= 2\sigma_I^2. \quad (6.43)$$

Replacing (6.43) back in (6.39), the covariance of $\mathbf{h}$ assumes the simple form:

$$Cov\left[\mathbf{h}\right] = 2\sigma_I^2 \mathbf{I}_{n\times n}, \quad (6.44)$$

where $\mathbf{I}_{n\times n}$ is the $n \times n$ identity matrix. Replacing (6.44) in (6.37) and simplifying some terms, the uncertainty of the estimate $\hat{\mathbf{d}}$ is expressed as:

$$\mathbf{\Pi}_{\hat{\mathbf{d}}} = 2\sigma_I^2(\mathbf{A}^T\mathbf{A})^{-1}. \quad (6.45)$$

Noting that the variance of the gradient $E\left[gg^T\right]$ is estimated by

$$\Lambda_g = \frac{1}{n^2}\sum_i g_i g_i^T \quad (6.46)$$

$$= \frac{1}{n^2}\begin{bmatrix} \sum g_{u_i}^2 & \sum g_{u_i} g_{v_i} \\ \sum g_{u_i} g_{v_{ij}} & \sum g_{v_i}^2 \end{bmatrix} \quad (6.47)$$

$$= \frac{1}{n^2}(\mathbf{A}^T\mathbf{A}), \quad (6.48)$$

the uncertainty of the displacement estimate is given by the expression:

$$\mathbf{\Pi}_{\hat{\mathbf{d}}} = \begin{bmatrix} \pi_u & \pi_{uv} \\ \pi_{uv} & \pi_v \end{bmatrix} \quad (6.49)$$

$$= \frac{2\sigma_I^2}{n^2}\Lambda_g^{-1}. \quad (6.50)$$

6.2.2 Selecting Trackable Features

Equation (6.50) shows that tracking uncertainty depends upon two adjustable factors: the size n of the window and the inverse of the second order moment of the gradient. In terms of window size, the higher the value of n the lower the uncertainty. However, the size cannot increase arbitrarily without violating the assumption of smooth motion:

for high enough n, the window may contain points with different motions, making the least square estimate invalid. Experience has shown that windows of 15×15 are a good trade-off between trackability and constraint satisfaction. The most important factor is in fact the gradient of the window. Notwithstanding high n's, matrix Λ_g may be non-invertible or badly conditioned, thus the uncertainty can be very high. This phenomenon is known as the aperture problem [HS81, Hil84], and is illustrated in Figure 6.2. Assume there is an object with oriented edges and we selected the

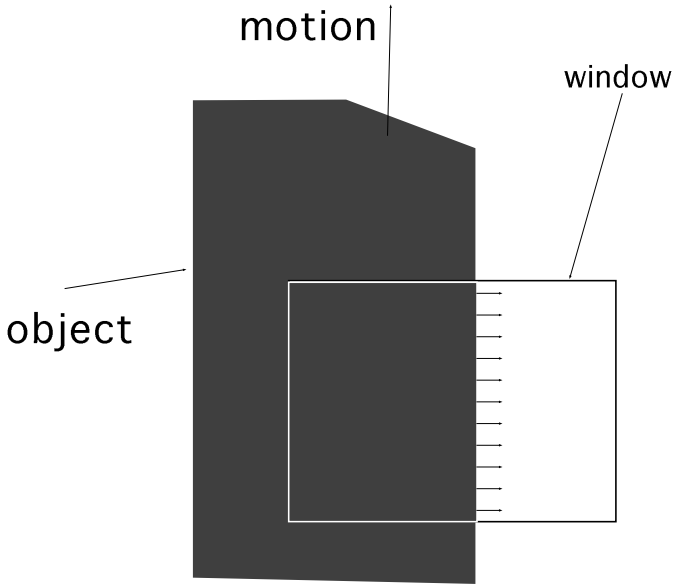


Figure 6.2: Aperture problem in feature tracking

window indicated in figure 6.2. The set of arrows inside the window represent the image gradient in that region. Observe that if the object moves in the perpendicular direction of the gradient, the pattern inside the window remains unchanged, and no motion is detected. This phenomenon is revealed by the zero determinant of Λ_g. Since the v component of the gradient is zero for all points, Λ_g is singular and thus the uncertainty is considered infinity. This fact can be used to select the trackable features. From expression (6.50) we determine $\mathbf{\Pi_{\dot{d}}}$ for all image points and select the best ones from the condition number of matrix Λ_g. The condition number is given by the ratio between the largest and the smallest eigenvalues of matrix Λ_g, and for a well conditioned matrix

this ratio should be close to 1. Clearly, the computational burden is great, since we have to compute the eigenvalues of the moments of the gradient, for all possible $n \times n$ windows. Fortunately, this process runs only once to select the trackable features in the first image. Performance tests presented in [Mad93] show that rates of 3 frames per second are obtainable with ordinary hardware.

6.2.3 Tracking Over the Whole Sequence

Recall that the measurements matrix $\mathbf{W}$ is created with the image position of the selected features. In the previous section we developed a method to compute the feature displacement between two frames. The integration of these displacements is done by registration of the first image and then computing the displacement to the current one, as opposed to adding up all the displacements computed from image 1 to the current one. Figures 6.3 6.4 and 6.5 show graphically how these operations are done.

After selecting the features in the first image I_0, we acquire image I_1 and compute the first displacement $\hat{\mathbf{d}}_1$. Assuming a highly noisy image I_1, the displacement error will be high. In other words the deviation between the correct coordinates (u_1, v_1) and the estimated coordinates $(\hat{u}_1, \hat{v}_1)$ is high. Figure 6.3 shows this error by the shaded squares representing both the estimated and correct position of the feature. The registration of image I_0 consists in shifting the original feature by $\hat{\mathbf{d}}_1$ and forming an "estimated" image $\hat{I}_1$ as shown in Figure 6.4. Then, when a new image I_2 is obtained, we compute the displacement between the registered original window and the current image. If image I_2 has low noise the estimate is accurate (dashed arrow) and the coordinate error will be small, as seen in Figure 6.5. Then, the estimated position in I_2 will be the same as if it was estimated from the correct position (solid arrow). In summary, the error made in the previous sampling interval will not affect the estimate $(\hat{u}_2, \hat{v}_2)$. The basic advantage behind this procedure is that even though the cumulative uncertainty of $(\hat{\mathbf{d}}_0 + \hat{\mathbf{d}}_1 + \cdots + \hat{\mathbf{d}}_f)$ grows, the uncertainty of the feature coordinates at any instant is constant and is given by (6.52). In other words, if the tracking was particularly noisy

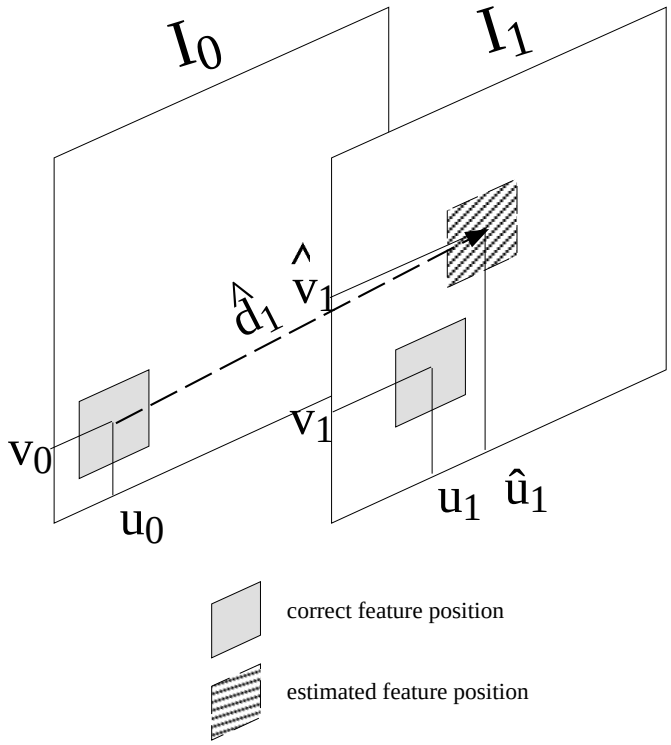


Figure 6.3: Tracking between frame 1 and 2

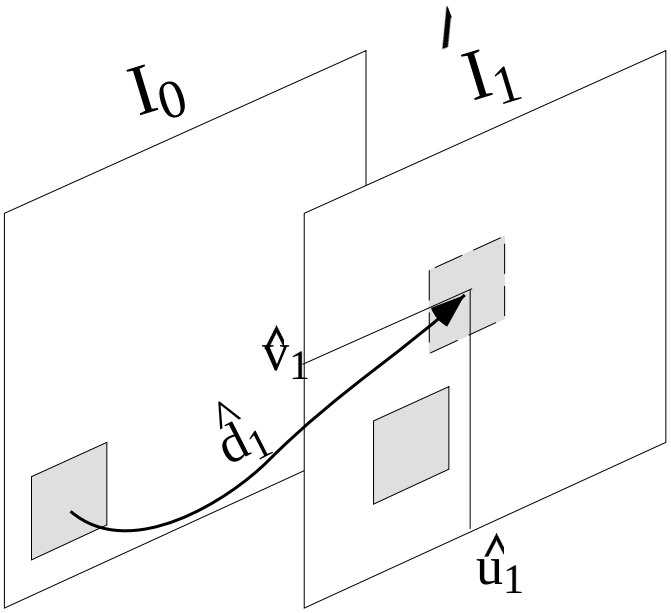


Figure 6.4: Registering first image

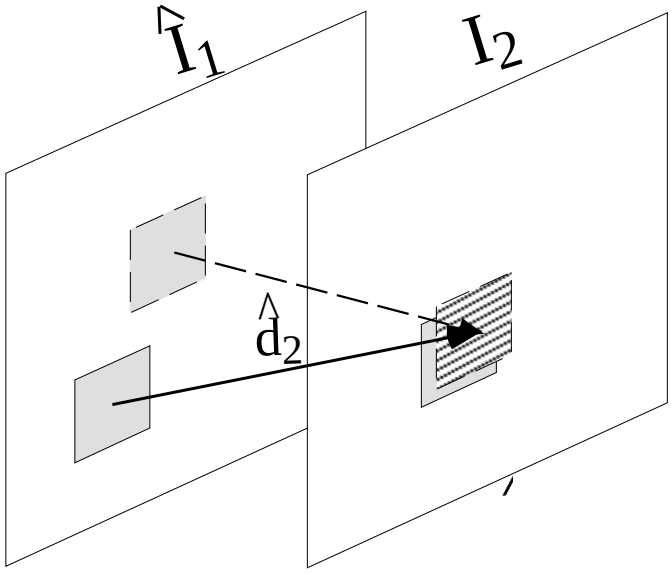


Figure 6.5: Tracking between frames 2 and 3

between any two frames, the error will not propagate to future tracks and we have a constant expression, given by (6.52) for the uncertainty.

In conclusion, having selected N features in the image, the displacement and position uncertainty of each feature i, at each sampling interval f, is given by:

$$\hat{\mathbf{d}}_{if} = (A^T A)^{-1} A^T \, h \tag{6.51}$$

$$\mathbf{\Pi}_{if} = 2\sigma_I^2 \begin{bmatrix} \sum g_{u_{ij}}^2 & \sum g_{u_{ij}} g_{v_{ij}} \\ \sum g_{u_{ij}} g_{v_{ij}} & \sum g_{v_{ij}}^2 \end{bmatrix}^{-1} \tag{6.52}$$

$$= \begin{bmatrix} \pi_u^2 & \pi_{uv} \\ \pi_{uv} & \pi_v^2 \end{bmatrix} \tag{6.53}$$

As a final comment we should emphasize that equations (6.51) and (6.52) must be used under the assumptions referred to before. In other words, we must guarantee that real images obtained in practical situations do not violate the assumption upon which the above equations were obtained. The dominance of translational motion is guaranteed by the temporal image sampling which we talke as high compared with the velocity of the imaged objects. This fact also helps the feature displacement between two frames to be small, so that the Taylor expansion of the image is valid. However, the most effective way to handle this constraint is through multiscale image representations

[BAHH92, Ros84]. In fact, in [Mad93] an image pyramid is built and the feature displacement is then computed from coarse to fine levels of resolution. At each level of resolution the images are registered using the displacement computed at the higher level such that $\|d\| \ll 1$. The details of the implementation are documented in [Mad93].

6.3 Algorithm for Rank Determination of W

With the developments of the previous section we can now easily follow the rank determination algorithm outlined in section 6.1. The noisy measurements matrix $\tilde{\mathbf{W}}$ is given by:

$$\tilde{\mathbf{W}} = \mathbf{W} + \mathcal{E} \tag{6.54}$$

$$= \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1N} \\ & \vdots & \vdots & \\ u_{F1} & u_{F2} & \dots & u_{FN} \\ v_{11} & v_{12} & \dots & v_{1N} \\ & \vdots & \vdots & \\ v_{F1} & v_{F2} & \dots & v_{FN} \end{bmatrix} + \begin{bmatrix} \varepsilon_{u_1} & \varepsilon_{u_2} & \dots & \varepsilon_{u_N} \\ & \vdots & \vdots & \\ \varepsilon_{u_1} & \varepsilon_{u_2} & \dots & \varepsilon_{u_N} \\ \varepsilon_{v_1} & \varepsilon_{v_2} & \dots & \varepsilon_{v_N} \\ & \vdots & \vdots & \\ \varepsilon_{v_1} & \varepsilon_{v_2} & \dots & \varepsilon_{v_N} \end{bmatrix}, \tag{6.55}$$

where each of the noise terms $(\varepsilon_{u_i}, \varepsilon_{v_i})$ has covariance matrices given by (6.52).

Recall from section 6.1 that the approximated matrix $\hat{\mathbf{W}}$ is given by the first r singular values and singular vectors. Constant r is the estimated rank, given by the smallest integer for which the sum of the last r singular values satisfies the inequality:

$$\sigma_{r+1}^2 + \cdots + \sigma_N^2 \leq \mathcal{T} \sum_{p=1}^{N} \sum_{f=1}^{F} (\pi_{u_{f,p}}^2 + \pi_{v_{fp}}^2), \tag{6.56}$$

where $\mathcal{T}$ is an adjusting factor (experimentally obtained).

Chapter 7

Discussion and Conclusion

In this thesis we have addressed the problem of motion and shape recovery in scenes with multiple moving objects. The problem is approached using the representation of the Tomasi-Kanade Factorization Method.

We have presented a new derivation, where the translational motion component is included. The representation used here allows an easy expression of motion and shape in multi-body scenes, where translation cannot be eliminated. We have also provided a geometrical interpretation to the matrices involved in the Factorization Method.

More importantly, we have proposed a new mathematical construct called the *shape interaction matrix* that leads to a solution for the multibody structure-from-motion problem.

The striking fact is that the method allows for segmenting or grouping image features into separate objects based on the shape properties *without* explicitly computing the shapes themselves. Also, no prior knowledge of the number of moving objects in the scene is assumed.

This is due to the interesting and useful invariant properties of the shape-interaction matrix $\mathbf{Q}$. We have shown that $\mathbf{Q}$ is motion invariant. Even when the matrix is computed from a different set of image-level measurements $\mathbf{W}$, generated by a different set of motions of objects, its entries will remain invariant. Each entry has the same unique value independently of the trajectories of the objects. The motion invariance property of $\mathbf{Q}$ also means that the degree of complexity of the solution is dependent

on the scene complexity, but not on the motion complexity.

The shape interaction matrix $\mathbf{Q}$ is also invariant to the selection of the object coordinate frame. Thus, the origin of the object coordinate system can be placed anywhere without changing the entries of $\mathbf{Q}$.

Another interesting fact is that the shape interaction matrix can handle many degenerate cases as well, where one or more objects may not be a full 3-D object, but a linear or planar object. The synthetic example shown in the experiments chapter was in fact a degenerate case where a planar object was included.

Using these properties of the shape interaction matrix we have developped a sorting and detection algorithm that determines the block-diagonalization through an off-diagonal energy minimization.

Underlying the computation of matrix $\mathbf{Q}$, the determination of the rank of $\mathbf{W}$ was handled by estimating the uncertainty of the feature tracking and feeding it back in a minimum error estimate of the significant singular values of $\mathbf{W}$.

Finally, we have shown a set of three experiments under different scene conditions, which have shown the potential of the theoretical derivations.

7.1 Future Developments

This thesis has addressed the problem of multiple motion segmentation when the objects move with independent motions. That is to say, the columns of the motion matrix $\mathbf{M}$ which belong different objects are linearly independent. There are quite interesting aspects of rank degeneracies of the motion and shape matrices. We have shown that the shape interaction matrix is invariant to shape degeneracy. In other words, the zero/non-zero relationship between elements of the shape interaction matrix is not affected by the rank of the shape matrix of each individual object. Furthermore, as illustrated by our outdoor experiment, the rank degeneracy of the individual motion matrix does not affect the segmentation either. Motion rank degeneracies can be understood as limitations of the orthographic camera model, since orthography does not

model depth: for example, a full 3D object rotating around the camera's focal axis is indistinguishable from a planar object rotating the same way.

The problem becomes more complex if motions of two different objects become degenerate between themselves, that is, if some columns of one motion matrix are linearly dependent on the columns of the other motion matrix. This is the case in articulated objects, where the motion of each joint is dependent on the other joints. The challenge here is that the block-diagonality property of the shape interaction matrix is lost. The current method cannot handle this case, although it can easily detect it. We can verify the validity of the segmentation by analyzing the rigidity of the solution using the motion and shape matrices produced by the factorization method.

This motion degeneracy is revealed by the intersection of the shape subspaces (spanned by $\mathbf{V}$) which are no longer orthogonal. One suggestion is to develop an incremental version of this method, which has in fact been done by [MK94] and, at each step, to segment and test the rigidity of the solution. Of course this leads to problems similar to those pointed out earlier : the cyclic dilemma between checking if the solution is rigid and assuming rigidity to compute the solution. Assuming prior knowledge about the scene, the case of articulated objects can be handled using parametrized models [Reg95] which can be adapted to our method. However, the general case of linearly dependent motions requires careful attention and more research in order to fully understand the behavior of the feature trajectories in the shape subspace.

An interesting future development of our method is the extension to more complex projection models like perspective projection. In this case the useful properties of shape subspace orthogonality are no longer valid. Even though we do not see any clear evidence that such an extension is possible, we find the study of feature trajectories in projective space promissing. One reason is that multiple motion segmentation is a problem with weaker constraints than shape or depth computation: we may search for relationships between the subspaces spanned by feature trajectories in projective space without the need to reconstruct the scene.

Bibliography

[AB85] Edward. Adelson and James Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America*, 2(2):284–299, 1985.

[Adi85] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401, July 1985.

[BAHH92] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. of the 2nd. European Conference on Computer Vision*, santa Margherita, Italy, 1992.

[BB91] Terrance Boult and Lisa Brown. Factorization-based segmentation of motions. In *Proceedings of the IEEE Workshop on Visual Motion*, October 1991.

[BBHP90] J. Bergen, P. Burt, R. Hingorani, and S. Peleg. Computing two motions from three frames. In *Proceedings of the IEEE International Conference on Computer Vision*, December 1990.

[BHK91] Peter Burt, Rajesh Hingorani, and Raymond Kolczynski. Mechanisms for isolating component patterns in the sequential analysis of multiple motion. In *IEEE Workshop on visual motion*, pages 187–193, October 1991.

[BK79] G. Bienvenu and L. Kopp. Principe de la noniometre passive adaptive. In *Proc. 7'eme Colloque GRETSI*, pages 106/1–106/10, Nice, France, 1979.

[CLR86] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms.* The MIT Press, 1986.

[Dem87] James Demmel. The smallest perturbation of a submatrix which lowers the rank and constrained total least squares problems. *SIAM Journal of Numverical Analysis*, 24(1), 1987.

[Fau94] Olivier Faugeras. *Three Dimensional Computer Vision.* MIT Press, Cambridge, Mass., 1994.

[Gea94] C. W. Gear. Feature groupin in moving objects. In *Proceedings of the workshop on motion of non-rigid and articulated objects*, Austin, Texas, 1994.

[GHS87] G. Golub, A. HOffman, and G. Stewart. A generalization of the eckart-young-mirsky approximation theorem. *Linear Algebra Applications*, 1987.

[Hee88a] D. Heeger. A model for the extraction of the image flow. *J. Optical Society of America*, A(4), 1988.

[Hee88b] D.J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1(4):279–302, 1988.

[Hil84] Ellen Hildreth. *The Measurement of Visual Motion.* MIT press, Cambrdige, Massachusets, 1984.

[HS81] B.K.P. Horn and B. Schunk. Determining optical flow. *Artificial Intelingence*, 18:185–203, 1981.

[IBP94] Michal Irani, Rousso Benny, and Shmuel Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, February 1994.

[Jas92] Radu S. Jasinschi. Intrinsic constraints in space-time filtering: A new approach to representing uncertainty in low-level vision. *IEEE transactions of pattern analysis and machine intelligence*, 14(3):353–366, March 1992.

[JRS92] Radu S. Jasinschi, A. Rosenfeld, and K. Sumi. Perceptual motion transparency: the role of geometrical information. *Journal of the Optical Society of America*, 9(11):1–15, November 1992.

[KvD91] Jan Koenderink and Andrea van Doorn. Affine strucutre from motion. *Journal of the Optical Society of America*, 8(2):377–385, 1991.

[LK81] Bruce Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981.

[Mad93] Richard Madison. A fast feature tracker for image sequence analysis. Master's thesis, School of Computer Science, Carnegie Mellon University, December 1993.

[MK94] Toshihiko Morita and Takeo Kanade. A sequential factorization method for recovering shape and motion from image streams. Technical Report CMU-CS-94-158, School of Computer Science, Carnegie Mellon University, May 1994.

[NZ92] Nassir Navab and Zhengyou Zhang. From multiple objects motion analysis to behaviour-based object recognition. In *Proc. 10th European Conference on Artificial Intelligence, EC AI 92*, 1992.

[PK93] C Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. Technical Report CS-93-219, School of Computer Science, Carnegie Mellon University, December 1993.

[Reg95] James M. Regh. PhD thesis, ECE, Carnegie Mellon University, 1995. In preparation.

[Ros84] Azriel ed. Rosenfeld. *Multiresolution image analysis*. Springer-Verlag, New York, 1984.

[Sch80] R. Schmidt. *A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation*. PhD thesis, Stanford University, CA, 1980.

[Sin93] D Sinclair. Motion segmentation and local structure. In *Proceedings of the 4^{th} International Conference on Computer Vision*, 1993.

[Ste92a] G.W. Stewart. Determining rank in the presence of error. In *Proceedings of the NATO Workshop on Large Scale Linear ALgebra*, Leuven,Belgium, 1992. Also University of Maryland Tech. Report.

[Ste92b] G.W. Stewart. On te early history of the singular value decomposition. Technical Report TR-92-31, University of Maryland,Institute for Advanced Computer Studies, March 1992.

[TK90a] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams: a factorization method. Technical Report CS-90-166, School of Computer Science, Carnegie Mellon University, September 1990.

[TK90b] Carlo Tomasi and Takeo Kanade. Shape and motion without depth. In *International Conference on Computer Vision*, pages 91–95. IEEE Computer Society, 1990.

[Ull83] Shimon Ullman. Maximizing rigidity: The incremental recovery of 3-d structure from rigid and rubbery motion. Technical Report A.I. Memo No. 721, MIT, June 1983.

[VT68] Harry Van Trees. *Detection, Estimation, and Modulation Theory*, volume 1. Wiley, New York, 1968.

[WA83] A.B. Watson and A.J. Ahumada. A look at motion in the frequency domain. Technical Report TM-84352, NASA, 1983.

[Wil94] Reg Willson. *Modeling and Calibration of Automated Zoom Lenses.* PhD thesis, ECE, Carnegie Mellon University, 1994.