KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ESAT
AFDELING PSI-VISICS
Kasteelpark Arenberg 10 — 3001 Leuven-Heverlee, Belgium

3D RECONSTRUCTION OF DYNAMIC SCENES

Promotor:
Prof. Luc Van GOOL

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de Toegepaste Wetenschappen
door

Kemal Egemen ÖZDEN

November 2007

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ESAT
AFDELING PSI-VISICS
Kasteelpark Arenberg 10 — 3001 Leuven-Heverlee, Belgium

3D RECONSTRUCTION OF DYNAMIC SCENES

Jury:
Voorzitter: Prof. A. Haegemans
Prof. L. Van Gool, promotor
Prof. L. Van Eycken, assessor
Prof. P. Dutre, assessor
Prof. H. Sahli, VUB, Belgium
Prof. M. Pollefeys, ETH, Switzerland

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de Toegepaste Wetenschappen

door

Kemal Egemen ÖZDEN

U.D.C 681.3*I4

November 2007

Wettelijk depot D/2007/7515/102

ISBN 978-90-5682-868-4

Acknowledgments

Doing a PhD study is not much different from exploring unknown lands. You have to go to the depth of seas where nobody has ever gone before hoping to find something new for the mankind. Being in the company of talented people is a must for such a task and luckily I had it.

First of all, I would like to thank Prof. Luc Van Gool for giving me the chance to follow this study in his distinguished group, for his expert guidance and for creating a comfortable environment where I have easily concentrated on my research. In this regard, I would also like to thank our technical team, Paul, Bert, Patricia and Annitta for creating an efficient working environment for all of us.

At times when I could not find a solution to a problem, the fruitful discussions with Kurt were an invaluable source of creativity. Our collaboration was both stimulating and enriching. I am also grateful to Maarten Vergauwen for his technical support. I have asked myself many times "how does he know that ?". I would also like to thank other brilliant researchers that I met in Visics, for not only helping me in critical times but also being a good company.

The last chapter of this dissertation is a joint work with Konrad Schindler and I am grateful for his diligent cooperation. I also would like to thank Philip Dutre, Luc Van Eycken, Hichem Sahli, Marc Pollefeys and Ann Haegemans for being my jury and for their helpful comments.

A prosperous professional life usually needs to be supported by a merry social life and I would like to thank all my friends and my family for their constant support. Living abroad has made me realize how important they are. Last, but not least, I would like to thank my personal sunshine Maya, who is always a great relief in cloudy Belgian days.

Abstract

3D reconstruction of dynamic scenes from monocular images poses various challenges. The unknown relative scale between the background and the foreground object is a subtle one, however it needs to be resolved properly for a realistic reconstruction. Our solutions are based on the fact that the foreground trajectory has components from the camera trajectory at wrong relative scales. This phenomenon is exploited in two ways: statistical approach computes the most independent object motion from the camera's and the geometric approach detects various regularities in the object motion which would hint the correct relative scale. Initially it was inherently assumed that the background object is labeled apriori. However, later it is shown that not only the correct relative scales result in the simplest motions but also the correct background labeling. Hence aforementioned relative scale resolution techniques are also applicable for that problem. However, it is also possible that the moving objects in a scene do not follow any motion simplicity constraint at all. One way to overcome this problem is by using another independently moving camera. In this setting, the relative scales for the foreground objects in both cameras are selected in a way which results in the identical foreground motion for both of the camera reference points. However, in order to achieve that not only the relative scales of the foreground objects but also the similarity transform between two reconstructions from both cameras and plus the time shift parameter need to be computed. The final result turns out to be a space-time-scale registration technique for video streams. Another basic expectation from a multi-body 3D reconstruction framework is proper segmentation of the moving objects. In contrast to many techniques which does feature tracking as a pre-processing step before segmentation, a new framework is presented where segmentation, tracking and reconstruction are done simultaneously. This requires online SfM, hence inevitably to be able to handle object appearance and merge-split operations. With this technique, 3D reconstruction of long and realistic sequences is achieved.

Notations

This section gives a synopsis of the symbols that will be introduced throughout this dissertation. In general, upper case bold leters are matrices, lower case bold letters are vectors and normal letters are scalars.

SfM	structure from motion
SVD	singular value decomposition
KLT	Kanade-Lucas-Tomasi feature tracker
AIC	Akaike information criterion
$RANSAC$	random sampling consensus
$\mathbf{P}$	3×4 projection matrix
$\mathbf{R}$	3×3 rotation matrix
$\mathbf{t}$	3D translation vector
$\mathbf{F}$	3×3 fundamental matrix
$\mathbf{T}$	4×4 non-singular transformation matrix
s	unknown scale value
$\mathcal{L}$	log-likelyhood
$\mathbf{p}$	a 3D point
$\mathbf{v}$	a 3D velocity vector

Contents

Abstract **iii**

Notations **v**

Introduction **1**

1 Structure from Motion **5**
1.1 Previous Work on Static Scenes 6
1.2 Previous Work on Dynamic Scenes 7
1.3 A Typical Sequential SfM Algorithm for Static Scenes 11
1.3.1 Multiple View Geometry 13
1.3.2 Feature tracking and geometry initialization 15
1.3.3 Self-Calibration 17
1.3.4 Bundle Adjustment 17
1.3.5 Dense Reconstruction 18

2 The Relative Scale Ambiguity **21**
2.1 The Relative Scale Problem 22
2.2 The Independence Criterion 26
2.2.1 Measuring independence 26
2.2.2 Experimental results 30
2.3 The non-accidentalness principle 31
2.3.1 The planarity constraint 38
2.3.2 Experimental results 43
2.3.3 The heading constraint 46
2.4 A discussion on choosing the correct criterion. 51
2.5 Concluding remarks 52

3 Background Identification in Dynamic Scenes **55**
3.1 Introduction 55
3.2 Background Detection with Motion Constraints 57
3.2.1 The Independence Constraint 58
3.2.2 The Heading Constraint 59
3.2.3 The Planarity Constraint 61

3.3 Conclusion 69

4 Space-Time-Scale Registration of Dynamic Scenes 71
4.1 Introduction 71
4.2 Problem 73
4.3 Notation and Basic Formulation 73
4.4 Solution 75
4.4.1 Spatial Solution 75
4.4.2 Spatio-Temporal Solution 79
4.5 Degeneracies 79
4.6 Experiments 80
4.7 Conclusion and Discussion 81

5 Simultaneous Segmentation and Reconstruction of Dynamic Scenes 93
5.1 Introduction 93
5.2 A Review of Motion Segmentation 94
5.3 Model Selection Review 96
5.4 A General 3D Reconstruction Framework for Dynamic Scenes . 101
5.4.1 Requirements 101
5.4.2 Splitting and merging motions 102
5.4.3 Splitting versus Merging 102
5.4.4 Relative Scale Resolution 103
5.4.5 Practical Considerations 104
5.5 An example implementation 105
5.5.1 Overview 105
5.5.2 System Details 105
5.6 Details on the critical Sub-routines 106
5.6.1 3-View Motion Segmentation 106
5.6.2 Splitting 108
5.6.3 Merging 109
5.7 Experiments 109
5.8 Conclusion 117

6 Conclusion 121

Introduction

As the first electronic computers started to emerge during the 1940's, the classical philosophical debate "whether intelligent machines can be built or not", received a new momentum and this resulted in the birth of a new, ambitious science called Artificial Intelligence (AI). Since then, scientists are in the search of new paradigms to make computers mimic different aspects of the human mind and a branch of this endeavor that is called Computer Vision (CV), which studies how to make computers "see", proved to be one of the hardest. Today, CV entertains a broad range of application domains such as object recognition/categorization, medical imaging, motion capture, surveillance and security systems, automation, entertainment and forensics just to mention a few and the field is getting richer by adoption of new techniques from various fields, e.g. geometry, statistics, physics, information theory, topology, optimization etc.

Discovering how to generate 3D models out of multiple images has been a longstanding passion of CV scientists. The most prominent and the oldest setup is a stereo rig which consists of two slightly displaced cameras whose positions and internal parameters are precisely calculated with a calibration object. In such a scenario, as the relative camera displacements are known accurately, it is sufficient to find the corresponding pixels or feature points in two images to compute their depths. This biologically inspired technique which is based on the ancient triangulation methods enjoys a broad popularity due to its simplicity and maturity.

However by the year 1981, Longuet-Higgins described a new method in his seminal paper [LH81] to compute the 3D transformation between two cameras without employing a calibration object. This method, which utilizes just the point correspondences between the images, became the seed of a whole new prolific CV branch called Structure from Motion (SfM) and more generally Multiple View Geometry. As the name suggests, SfM algorithms try to compute both the 3D structure of a scene and the 3D geometric relationships between the cameras simultaneously. Later, scientists came up with different techniques such as 3-view, n-view, sequential, factorization, differential and probabilistic methods for different camera models such as orthographic, perspective, paraperspective cameras etc.

Although SfM's relatively old age, it is still a hot research topic as can be

witnessed in academic conferences and journals. The main reason for such an unexpectedly long enthusiasm is not only due to the increasing number of application areas, e.g. conservation of cultural heritage, architecture, navigation, computer graphics, augmented reality but also due to the inherent difficulty of the problem. This adversity was initially alleviated by making strong assumptions on the input data, such as assumption of a static scene where nothing is moving except the camera, existence of enough feature points on the images and the assumption of a small baseline between the camera positions. The elegant mathematical analysis can be achieved for such cases [FL01, HZ00]. However, in order to increase the employment of such techniques, it is necessary to lift or at least ease those assumptions, and the work that is detailed in this dissertation contributes to the disposal of the first one, namely the static scene assumption.

Indeed, the bulk of the relevant literature depends on the assumption that there is nothing moving in the scene except the camera, or the dual case where the camera is static and the scene itself is the only moving object. In the following chapter one can find a short overview of such classical techniques. However those approaches become impractical in the case of real life scenes which typically consist of dynamic elements such as cars, bikes, people etc. moving independently. In this work we endeavored to extend the classical SfM techniques to such dynamic scenes. We embarked on such a research avenue not only due to academic curiosity but also due to increased demand from the industry. 3D TV sets are likely to take their place among consumer electronics in a few years and one of the most obvious challenges is to convert the substantial 2D legacy content to 3D. Considering that interesting 2D contents almost always contain dynamic elements, static scene SfM algorithms alone have little use here. Dynamic scene analysis would also find a prolific niche in the sports arena. It is not uncommon for various sport events to be recorded by several cameras that are either static or moving and 3D reasoning in such cases can be quite helpful both in terms of viewing pleasure (e.g. generating novel views, augmenting original views) or analysis (e.g. player positions, various statistics). Another industrial domain would be 3D city modeling which is getting to be more fashionable due to touristic opportunities, increased usage of GPS maps, up-trending concern on disaster management and other possible urban architecture applications. It is popular to model urban areas from video or an image sequence and the related dynamic scene information, e.g. cars, trams, people etc. are usually discarded. Their incorporation would give more realistic and lively city models. Augmented Reality, which is a sister branch of CV, would also benefit significantly from such techniques. As the name suggests, Augmented Reality is about augmenting an original video footage with artificial objects and giving the viewer an impression that the newly incorporated object is actually a part of the scene. Generating depth information on the scene is crucial as it determines whether the new object is occluded or not. Consequently, estimating the depth of the dynamic elements of the scene would significantly improve the realism of such augmented footage.

Rather than trying to come up with a complete solution for any kind of dynamic scene, which could be a Hercule's task, we shot for a middle milestone where the dynamic elements in the scene are locally rigid. Such scenes are very common thanks to the moving vehicles which are in general quite rigid or at least consist of rigid parts. It is also not uncommon for non-rigid dynamic objects, e.g. people, to show some local rigidity for all the sequence or complete rigidity for a short period of time. In the same line of thinking, some other researchers working to bring dynamic scene information to SfM also made similar assumptions, e.g [VSMS02b, CK95, WS01b, SS06, GQZ05, Tor98]. However the core of the work that is presented here is complementary to their approaches rather than extending or competing with them.

Probably the most important contribution of this dissertation is the first detailed study of the subtle problem called relative scale ambiguity which is an inevitable issue that is raised when there are multiple independently moving objects in a scene. It is a direct result from the fact that every structure and translation parameters computed from a monocular image sequence is defined only up to a scale. When an object is viewed from different viewpoints, although the angles and the length ratios on the structure can be deduced, we can never deduce the actual lengths. This is not a disturbing problem for the purpose of generating novel views of a static scene, since everything is probably scaled for visualization anyhow. However when there are multiple moving objects, the problem of setting their scales relative to each other kicks in.

The first task that is achieved in this Ph.D. work is a detailed analysis of that problem and the proposal of practical solutions based on the existence of realistic motion constraints. Although the techniques we introduce are quite practical, they are based on some apriori information, such as the knowledge of which object is the background, the existence of the motion constraints and the precomputed motion segmentation of the image sequence. In the rest of the work, we explored different ways to lift those limitations.

The first limitation we attacked is the user supplied information regarding the ID of the background object. Such an assumption is undesirable as we would like our processes as autonomous as possible. Typically simple heuristics are used to guess that information, such as the size of the object on the images or the occluding contours. However, many times such heuristics are not helpful, e.g. when the foreground object is close to the camera, size based heuristics definitely fail. In the course of the presented work, we explored a different kind of heuristic based on the 3D motion of the objects. A novel idea that is suggested phrases that the aforementioned motion constraints are not only useful for the detection of the right relative scale, but also beneficial in identifying the background object since the correct background identification would result in the simplest 3D relative motions for the other objects.

The next question we asked is whether the need for motion constraints can also be lifted. However, the missing information due to the lack of motion constraints must be compensated by different means. This vacancy is filled by introducing a second independently moving camera. Although the initial

analysis was first aimed at the resolution of scale ambiguities, the resultant technique turned out to be a more general method which not only solves the scale ambiguities but also registers two or more image sequences in 3D space and time.

In the aforementioned techniques, we presumed that motion segmentation is already achieved which is not unrealistic considering the large volume of successful research reports related to motion segmentation. However, applying a classical SfM computation after motion segmentation is far from optimal. Yet, techniques such as [VSMS02a, CK95] which estimate SfM and motion segmentation simultaneously, assume complete feature tracks are available which is a quite restrictive assumption, especially for long sequences. During the course of tracking, an early segmentation would result in early geometry computation which results in better and more feature tracks. This would later yield more efficient segmentation. Such an approach can run robustly for quite long sequences as incomplete tracks can easily be incorporated and objects popping up and disappearing are handled naturally. To develop such a system, we adopted techniques that are based on a model selection framework. Also, considering the general outline of the dissertation, such a sub-module frees us from third party segmentation schemes and its integration to the work described in the other chapters results in a complete SfM pipeline for dynamic scenes with rigid objects.

The rest of the dissertation is organized as follows. In the first chapter a brief overview of the eminent structure from motion research is presented both for static and dynamic scenes. This discussion is followed by a short description of the typical pipeline we use. It is a rather terse introduction to such a bulky field and assumes the reader is familiar with CV concepts. We detail the novel work starting from the second chapter where the relative scale ambiguity concept is introduced. The problem is analyzed and solutions are proposed based on motion constraints. In the next chapter we describe how such constraints can also be useful in determining the background object correctly. In the fourth chapter, the assumption of motion constraints is lifted, at the expense of introducing a second, independently moving camera and resulting in a space-time-scale registration technique. In the fifth chapter, we introduce a simultaneous tracking-segmentation-reconstruction framework which can run for quite long sequences. In the last chapter, we conclude the dissertation with a discussion of the results and intended future work.

Chapter 1

Structure from Motion

It can be confidently said that structure from motion is one of the mostly studied domains in Computer Vision. During a quest of almost 25 years , a myriad papers have been written about it and yet, the appetite of researchers for the topic seems unlikely to be quenched in the near future. There are several causes that underlie such enthusiasm. The most prominent is the fact that there are numerous number of paths that can be followed depending on the camera projection model, the type of the correspondences between the images (lines, corners, wide baseline features etc.) and how they are matched (windowed search, tracking, optical flow), the unknowns in the motion model (calibrated, semi-calibrated, uncalibrated internal parameters, restricted motions, wide or short base-line etc.), the unknowns in the structure (completely free, planar, piecewise planar scenes etc.), the type of the optimization algorithms (linear, non-linear), timing constraints (batch, on-line, real-time), and the strategy to handle robustness and degeneracy issues (M-estimators, RANSAC, planar degeneracies, probabilistic tracking etc.). In addition to all of the above, a final not-so-easy 3D modeling phase must be implemented properly where photo-consistency, smoothness factors, occlusions, and specularities must be simultaneously taken into account in the ideal case.

All of the aforementioned concepts were first studied for static scenes. Not surprisingly, such issues are also relevant for dynamic scenes and each major SfM branch seems to spawn its peculiar way of handling dynamic scenes. In the following paragraphs, we will first give quite a rough map of the conventional SfM landscape by citing the major seminal works. Later, important developments for dynamic scene analysis in the SfM context will be presented. In the remaining part of the chapter, a typical sequential SfM routine for static scenes is outlined. The described technique is quite similar to the one we will frequently use. The basic motivation in introducing such a section is to familiarize the reader with multiple-view geometry machinery and the syntax throughout the dissertation. However we must emphasize that most of the methods presented in this dissertation require just the precomputed projection

matrices and 3D point clouds as input, and consequently they are independent of how those inputs are generated.

1.1 Previous Work on Static Scenes

As mentioned previously, the SfM field is a rich and diverse world. One crucial factor resulting in such a diversity is the availability of different camera projection models. Indeed, a camera is just a device which projects a 3D scene to a 2D image plane and the choice of a proper mathematical function to model such a transformation highly depends on the type of application. The camera models can be broadly categorized as linear and non-linear. In the linear case, the simplest camera model is the orthographic one, where the projection of a 3D point is independent of its depth, i.e. perspectivity does not exist. Although its applicability is limited to scenes where perspective effects are negligible, the existence of powerful and relatively simple factorization algorithms, such as the pioneering work of Tomasi and Kanade [TK92], makes it quite attractive. Kanade's work is based on the observation that when the feature tracks over a sequence are listed in a matrix, the rank of that matrix is at most three assuming orthographic projection. Consequently SVD-like factorization methods can be used to extract motion and structure parameters from that matrix. Later this method has been extended to more realistic linear camera models, e.g. para-perspective [PK97] and also to perspective cameras by Sturm and Triggs [ST96]. However initialization of the feature point depths is required as a preprocessing step in the latter work.

When the linear models are not adequate due to strong perspectivity in the images, perspective camera models must be used, where the pinhole camera is the most popular one. Typically structure-and-motion is initialized by the help of two- or three-view geometric relationships, aka fundamental matrix and tri-focal tensor, between the first few images. Later, new projection matrices and 3D points are added progressively as the algorithm goes through the images sequentially. Such algorithms are reported to be quite successful [BTZ96, PVV*04, Cor04]. The underlying multi-view geometric concepts are discussed thoroughly in the books by Hartley and Zisserman [HZ00] and by Faugeras and Luong [FL01].

A successful pursuit taken by CV researchers was to compute the internal parameters of a camera from just the images themselves which is a necessary step to come up with realistic 3D models. A category of this type of techniques exploits constraints in the scene structure, such as parallel and orthogonal lines. To give an example, Caprile and Torre [CT90] compute the camera intrinsics from a single image of a cube by using the vanishing points defined by the lines on each cube side. A second category does not make a specific assumption about the 3D scene, but assumes that some or all of the internal parameters of the camera do not change throughout the sequence, which is quite a reasonable assumption in many cases. A practical self-calibration routine for purely

rotating cameras is reported by Hartley [Har94]. The transformation between the images of a rotating camera can be described by 2D homographies and as Hartley's paper demonstrates, it is possible to extract the internal calibration from those homographies. However the technique is inapplicable in the case of a general camera motion. The work of Triggs [Tri97] does not assume any a-priori knowledge on the scene or a specific type of camera motion. However he assumes fixed internal parameters. Triggs introduces an algebraic entity called absolute quadric which is a simpler way of representing the traditional absolute conic notion for self-calibration. The Kruppa equations for self-calibration is a different representation of the problem. The equations result from the epipolar transfer of the tangents of the absolute conic through the epipoles and they are solved by Faugeras *et al.* [FLM92] using the continuation method. Two camera motions yield two epipolar transformations and four constraints on the image of the absolute conic, which depends on five parameters, leaving a one-dimensional family of solutions. Therefore, three camera displacements yield six constraints, defining a unique solution.

Probabilistic tracking frameworks are also applied in SfM context. Broida *et al.* [BCC90], and Azarbayejani and Pentland [AP95] use extended Kalman filtering (EKF) techniques to estimate both the camera state space and the 3D structure parameters on-line. Such methods help to propagate the previous estimates of the parameters to new frames. Qian and Chellappa [QC01] replace the Kalman filter with a particle filter to be able to cope with multi-modalities in the state space.

A practical technique which is proved to be very effective is to compute the internal camera parameters in a separate step before processing the target images. Such a preliminary computation decreases the number of parameters to be estimated during the SfM procedure and consequently results in a better conditioned problem. This can be accomplished with a typical calibration scheme such as Tsai [Tsa87] and Zhang [Zha00] or with the aforementioned uncalibrated SfM techniques. A-priori knowledge on the internal matrices significantly improves the system's performance in the case of scene degeneracies such as dominantly planar regions or when the viewed object is relatively small compared to the image size. Nister's work [Nis03, DNB04] is one successful example. He reported a solution for the 5-point pose estimation problem and a real-time system exploiting it which could run for quite long sequences without drifting (a common curse in SfM algorithms).

1.2 Previous Work on Dynamic Scenes

Given the practical importance of dealing with independent motions, there has recently been an increased interest in the detection and analysis of such cases. We must note that in general the analysis of dynamic scenes is of course not new in Computer Vision. The scenes which are subject to typical CV applications, such as tracking, background modeling, motion segmentation etc.,

contain dynamic elements by definition. However, their analysis in a SfM context is a relatively new issue. We must also note that the motion segmentation concept, which is about grouping pixels or features according to their motion similarity or consistency, is a natural companion to any SfM application for dynamic scenes and most of the times we see papers related to both. Here we ignore the bulk of the motion segmentation literature and consider only the prominent ones which are related to SfM.

Factorization and Subspace Methods: With their multi-body factorization method, Costeria and Kanade [CK95] have extended the static scene factorization method of Tomasi and Kanade [TK92] to the scenes with multiple, independently moving rigid objects. In this work, a new algebraic entity called *the shape interaction matrix* is introduced. This matrix is independent of the types of the motions present in the sequence and transforming it into canonical form results in a natural segmentation of the feature tracks. After segmentation, applying a static scene factorization to each of the independently moving object suffices for 3D structure and motion generation. A major drawback of this method is the assumption of the not-so-general orthographic imaging conditions. However, the method is generic enough to extend it to more realistic linear camera models. A contemporary method in the same vein is from Boult and Brown [BB91] where the SVD of the feature track matrix is used both to estimate the number of motions and the segmentation. Gear [Gea94] also exploited the low-rank property of the feature track matrix but used Gauss-Jordan elimination rather than SVD. Another work on factorization is from Debrunner and Ahuja [DA98], but a limitation is the assumption of simple motion models. Ichimura [Ich99] exploited the aforementioned shape interaction matrix, where a candidate feature track is selected first by the help of a discriminant criterion and later similar feature tracks in terms of the same discriminant criterion are extracted out. This operation is repeated recursively. Factorization schemes typically use a specific form of feature track matrix, where each column represents a single feature track and contains both the horizontal and vertical image coordinates. In an interesting work from Machline *et al.* [MMI02], it is reported that a special rearrangement of the feature track matrix where each row corresponds to exactly one image frame, would result in grouping according to temporal similarities rather than rigidities.

One limitation with the Costeria and Kanade's [CK95] approach is the assumption that motion subspaces are independent which cause those subspaces to be orthogonal. However in many real world cases, the distinct motions are not independent, thus the related motion subspaces for the trajectories are not orthogonal. An interesting remedy to this problem comes from Vidal *et. al.* [VH04, VMP04, VMS03] under the name of GPCA (Generalized Principal Component Analysis), where each subspace is modeled with a linear polynomial and the mixture of n subspaces is modeled as the product of n linear polynomials. Given enough sample points, the coefficients of this higher degree polynomial can be estimated and later the component subspaces can be computed

by differentiating the resultant polynomial. Yan and Pollefeys [YP06b, YP06a] approaches the problem differently. After reducing the dimension of the data and projecting it on a unit sphere by SVD, a subspace is estimated for each feature track using its local neighbourhood, an affinity matrix is generated by computing the distance between those subspaces and spectral clustering is applied later which results in motion segmentation. In contrast to GPCA, this approach requires significantly less sample points at the cost of assuming spatial proximity of the feature tracks that belong to the same subspace. Another interesting theme that Yan and Pollefeys [YP05, YP06a] investigated is the articulated motion, where the moving components are attached to each other with joints. The relationship between the rank of the resultant trajectory matrix and the type of the joints are clarified and an automatic articulated chain building algorithm is presented. A recent benchmark of aforementioned algorithms is given by Tron and Vidal [TV07]. Another recent work, which extends such multibody SfM algorithms from affine projection to perspective projection is given by [LKSV07], where the iterative factorization technique of Sturm and Triggs [ST96], is extended for multibody case.

Algeabric Methods: The analysis of dynamical scenes in the SfM context is not only limited to linear camera models. Interesting works based on more realistic perspective models have also been reported. One group of researchers attacked the problem in an algebraic way. Wolf and Shashua [WS01b] came up with an entity called *segmentation matrix* which is computed from the feature matches between two images of two moving rigid bodies. This matrix is later used to recover the original fundamental matrices related to each moving object. This technique is elaborated further by Vidal *et al.* [VSMS02b] where the technique is extended to the multi-body case by introducing the more general *multi-body fundamental matrix*. Later, Vidal and Ma [VM04] demonstrated that many types of motion models, 2D,3D affine, projective etc. can be handled in a similar way. The underlying idea is the fact that two-view image measurements of any kind of transformation can be fit to a polynomial and the motion parameters can be derived from the derivatives of that polynomial.

Methods with Constrained Motion: Another thread of research, including a significant portion of this dissertation, assumes there are different constraints on the object motion. Avidan and Shashua [AS00] investigated the case where a point is moving on a line or a conic and the camera positions are known beforehand. The bundle of the rays defined by the image projection of the moving 3D point and the camera poses is used as the space of possible solutions. Application of the motion constraint narrows down the solution space further to one possible trajectory in general. Sturm [Stu02] analyzed the case of a mobile stereo camera observing points that are moving on a pencil of planes. He came up with a two-view tensor which partially contains the scene structure and the stereo-rig motion. Shashua and Wolf also considered planar

motion [SW00] where a tensor formulation is derived for 3-views of dynamic points which move on planes. Han and Kanade [HK00, HK03] considered the case where the points are moving with constant velocities which constrained the motion of points to a line and came up with factorization based solutions both for affine and perspective cameras. Another interesting technique has been reported by Wolf and Shashua [WS01a] where a dynamic scene is mapped to a high dimensional static scene. They also employed first or second order derivative constancy of the object motion.

Methods with Model Selection: Information theory and statistics perspective also contributed significantly to the multi-body SfM literature. A prominent work is reported by Torr [Tor98] where a new model selection criterion called GRIC (Geometric Robust Information Criterion) is introduced. Model selection is a framework where not only the model that fits the data best is selected but also the least complex one, in the tradition of Occam's Razor. GRIC endows Kanatani's [Kan96] GIC model selection criterion with a robustness term. Kanatani later applied GIC to the subspace separation problem [Kan01]. Recent work of Schindler and Suter [SS06] applied GRIC to two-view multi-body SfM with a Monte Carlo sampling approach. A notable feature of this system is its ability to cope with feature matches that can belong to two motions at the same time. Later Schindler *et al.* [KSW06] extended this technique to multiple images in an MDL (Minimum Descriptive Length) fashion which is a more general way to handle the trade-off between the model complexity and the fitting error.

Probabilistic Methods: Another thread of research extends the recursive probabilistic tracking based SfM framework for static scenes to dynamic scenes. Darrel *et al.* [TDP94] proposed an EKF based technique based on the work of Azarbayejani and Pentland [AP95]. After hypothesizing possible groupings for each feature track based on their proximity and computing SfM for each of those groups, an MDL based criterion is applied to select the most parsimonious representation. Another multi body SfM algorithm proposed by Soatto and Perona [SP94] is also an EKF based technique. They exclude the structure parameters in the filter which enables the technique to handle feature points that change the motion group to which they belong. Later, Qian *et al.* [GQZ05] introduced a Multi-body SfM technique using particle filters which replace the limited parametric probability representation in EKF.

Calibration: In most of the above work, the existence of dynamic elements is just seen as an extra complexity to be handled, and it does not contribute positively to the overall SfM estimation. Actually, it affects the performance adversely in general because as the number of independently moving elements in a scene increases, the area they individually cover on the image plane tends

to decrease which results in poor parameter estimates for each object. The work of Fitzgibbon and Zisserman [FZ00] diverges in this respect where they showed that the existence of independently moving objects can be exploited to extract better internal calibration out of the image sequence as all the moving objects share the same internal camera calibration. The different motion trajectories from different objects would result in more reliable self-calibration.

Non-Rigid Motion: Constraining the algorithms to scenes with only rigidly moving objects is a common assumption for most of the aforementioned work, including this dissertation. However there are also significant research reports for the case where the objects are non-rigid. In a prominent work, Bregler *et al* [BHB00] presented a factorization method where the non-rigid deformation of an object can be modeled as a linear combination of basis shapes. Later Brand [Bra01] further improved this technique by introducing plausible heuristics for the computation of the corrective transform which is a typical step in every factorization scheme and a final non-linear minimization step. Xiao and Kanade [XCK04] employed constraints on basis shapes to further constraint the corrective transform. Subsequently, they extended their technique for perspective cameras with variable focal lengths [Kan05].

We must note that the discussion above covers only the SfM field whereas the novel techniques that are presented in this dissertation spans a broader range of topics in CV. The corresponding literature review will be presented in the relevant chapters, specifically : in chapter 3 the figure-ground problem, in chapter 4 space-time registration methods, and in chapter 5 segmentation and model selection related literature summaries are presented.

1.3 A Typical Sequential SfM Algorithm for Static Scenes

Here we roughly present a conventional sequential SfM technique which is described in literature by various authors such as Beardsley *et al.* [BTZ96], Pollefeys *et al.* [PVV*04] and Cornelis [Cor04]. The discussion will introduce the reader to the algebraic concepts and the core techniques we use frequently. This is especially essential to comprehend chapter 5 where the static scene based sequential SfM is extended to a simultaneous segmentation and reconstruction scheme for dynamic scenes. However the reader who is only interested in chapters 2,3 and 4 may only scan this section as those chapters employ just the output of the classical SfM routines, independently of how they are computed. Figure 1.1 gives an overview of the pipeline.

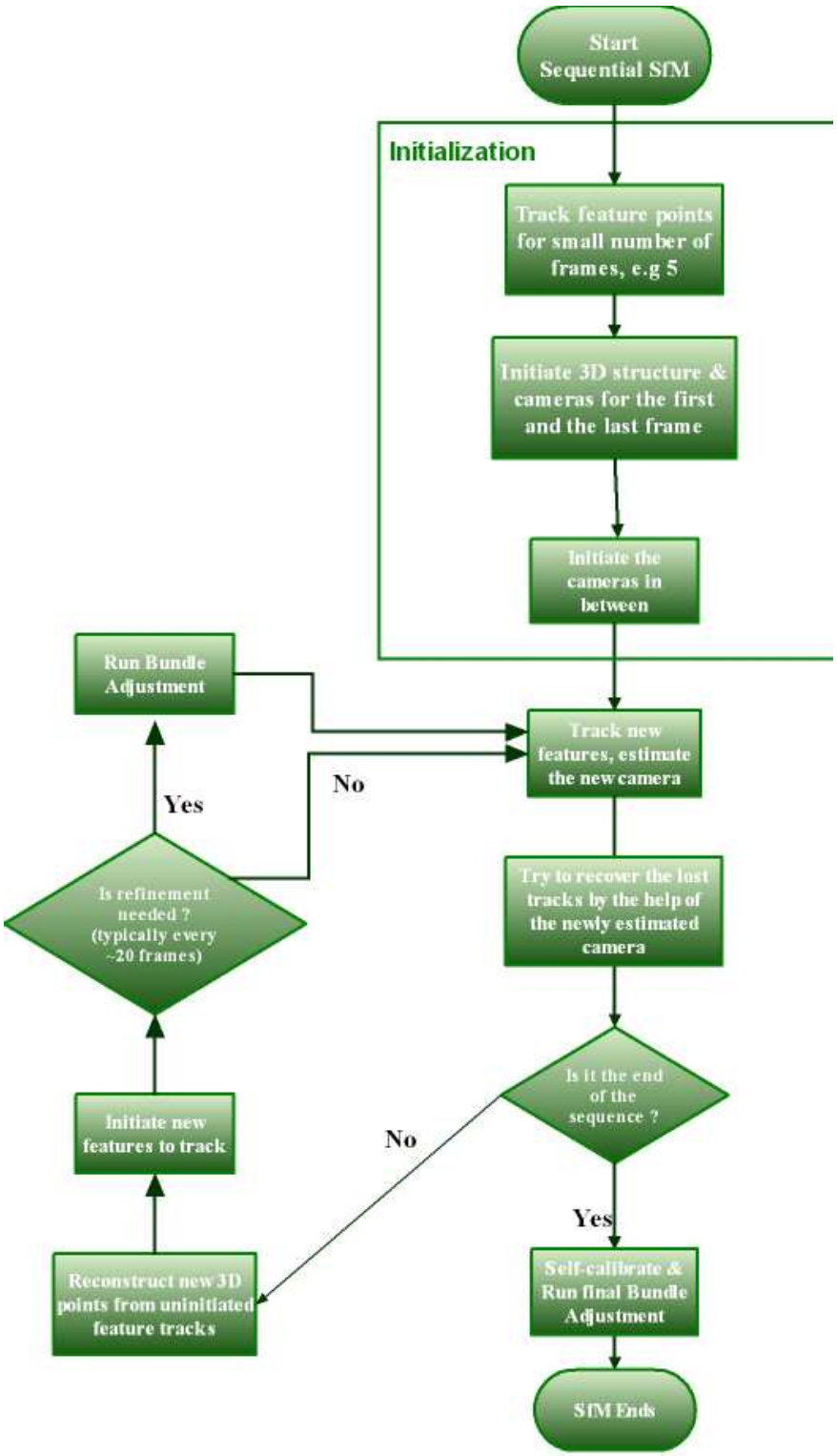


Figure 1.1: A typical sequential SfM algorithm.

1.3.1 Multiple View Geometry

As stated before, a basic step in any Computer Vision application is to decide on the correct mathematical model for the camera projection process which is usually determined by the type of the application. Throughout this dissertation we will assume a simple pin-hole perspective camera model which is a very good compromise between simplicity and generality. The typical projection formula is written as follows :

$$\begin{aligned} m &\sim \mathbf{P}M \\ \mathbf{P} &= \mathbf{K}[\mathbf{R}^T | -\mathbf{R}^T t] \\ \mathbf{K} &= \begin{bmatrix} f & s & u \\ 0 & rf & v \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{1.1}$$

$M = (X, Y, Z, W)^T$ and $m = (x, y, w)^T$ stand for a 3D point and its 2D projection. The points are represented by homogeneous coordinates, i.e. an extra parameter is padded to the end, which helps to hide the perspective non-linearities in the projection equation. The 3×4 matrix $\mathbf{P}$ is called the *projection matrix* and is determined by both the pose and the internal affine transformation. The *internal calibration matrix* $\mathbf{K}$ holds the camera's focal length f, skew s, aspect ratio r and principal point (u, v). It accounts for the internal transformation from retinal to image coordinates and most of its parameters are typically assumed to be fixed in an image sequence. Sometimes the focal length parameter is allowed to vary to account for the typical zooming operation. The camera orientation and the position is parameterized by the 3×3 orthonormal matrix $\mathbf{R}$ and 3-vector t respectively. '$\sim$' denotes that Eq. (1.1) is valid up to a scale factor.

To estimate a proper $\mathbf{P}$ matrix given known or precomputed 3D points (M's) and their corresponding 2D projections (m's) is called *camera resectioning* and it is a typical sub-problem in many CV applications. Looking at Eq. (1.1), it is seen that when the scale factor is removed, each 3D-2D correspondence would give exactly two equations constraining the underlying $\mathbf{P}$. So it can be concluded that 6 correspondences are enough to solve for a general $\mathbf{P}$ matrix which has 11 degrees of freedom. However if the internal calibration of the camera is known beforehand, which is a quite typical assumption, three points are enough to compute the positional parameters and a 3-point pose estimation technique like Grunert's [HCON94] would be proper. In the presence of outliers and redundant number of matches, which are again quite typical cases, robust least squares minimization techniques must be applied [HZ00].

There are also very well understood geometric relationships between the multiple-views of a 3D scene. The most popular one is the epipolar geometry in the case of two images and the relatively less popular tri-focal geometry for 3-view case. Here only the epipolar geometry will be presented. 3-view and N-view geometries are discussed in the book of Hartley and Zisserman [HZ00] in depth.

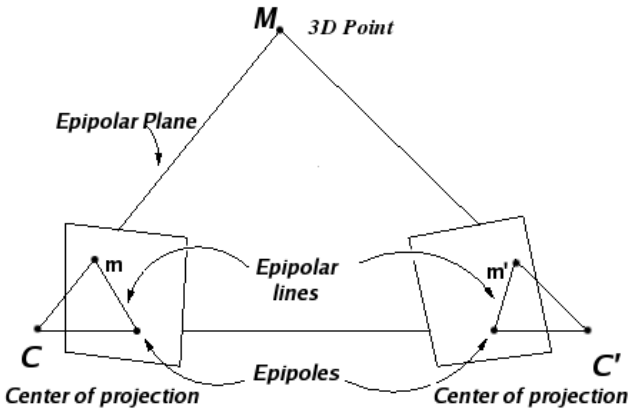


Figure 1.2: The elements of epipolar geometry.

As stated before, a camera is a device which transforms a 3D point in a scene to a 2D point on the image plane. However the depth information is lost during the projection and consequently given only the 2D point and the camera matrix, the possible solutions for the 3D point lie on the back-projection of that 2D point which is a ray in 3D space. The projection of this ray onto a second arbitrary camera image is a line. So it can be concluded that, a 2D point on the first view has its corresponding point on that line in the second view. Those lines are called *epipolar lines*, the bilateral 2D projections of the camera centers are called *epipoles* and the 3D plane which supports the camera centers, epipoles and the corresponding points is called *epipolar plane*. Figure 1.2 demonstrates those elements. In the most general case, where the cameras are not internally calibrated, the epipolar geometry can be written algebraically as:

$$m^T \mathbf{F} m \quad = \quad 0 \tag{1.2}$$

where m and m' are the 2D projections of the same 3D point on two different views, i.e. they are the matching points on the images. The *Fundamental matrix* $\mathbf{F}$ is a 3×3 rank-2 matrix and all the entities are again in homogeneous form. A 3×3 matrix has nine degrees of freedom, but due to the scale invariance and rank deficiency, the matrix loses two degrees of freedom so in total $\mathbf{F}$ has seven degrees of freedom. Each 2D point match applied to Eq. (1.2) gives one constraint on the $\mathbf{F}$ matrix so seven points are enough to estimate the epipolar

geometry. Similar to the arguments corresponding to the aforementioned camera resectioning problem, in case of redundant number of points and outliers, a robust least squares estimation scheme must be deployed [HZ00]. For the case of already known internal camera parameters, the fundamental matrix can be reduced to the 5 degrees of freedom *essential matrix* which is denoted by **E**. An interesting issue is the fact that the essential matrix is purely defined by the parameters of the relative camera poses. Given an essential matrix, the relative 3D rotation and translation parameters of the cameras can be extracted upto a scale factor in translation.

1.3.2 Feature tracking and geometry initialization

Establishing feature correspondences between images is a crucial step in any SfM algorithm. Different types of features have been used in the CV community depending on the application's nature. For narrow baselines, i.e. where the distance between the viewpoints of two consecutive images is small, lines or corners can be used. However for wide-baseline scenarios, where the viewpoints differ significantly, features which are more robust to geometric transformations must be used such as the ones reported by Tuytelaars *et al.* [TV99], Lowe [Low04] and Bay *et al.* [HBG06]. The work in this dissertation is aimed at video sequences, where the viewpoints of the consecutive images do not differ substantially, so it is practical to assume a narrow-baseline setup. However most of our results are independent of that assumption.

We preferred to use KLT features [TK91], as it is successfully applied to many such problems before. A KLT tracker consists of two basic steps, the first one is to generate suitable distinctive features on an image and the second one is to track them. Brightness constancy is assumed between the images, i.e. the corresponding features in consecutive images have the same intensity values. Given a feature point in the first image, a displacement $\triangle d$ is sought for in the second image which minimizes the intensity difference between the two locations supported by a small window. When this expression is differentiated for minimization and linearized using a Taylor expansion, the following expression is obtained:

$$\begin{aligned} \mathbf{Z}\triangle d &= e \\ \text{with } \mathbf{Z} &= \sum_{\mathbf{W}} \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} \mathbf{w}(m) \\ \text{and } e &= \sum_{\mathbf{W}} [\mathbf{I}(m) - \mathbf{I}^{'}(m)] \begin{bmatrix} g_x \\ g_y \end{bmatrix} \end{aligned} \tag{1.3}$$

where $\mathbf{I}$ and $\mathbf{I}^{'}$ are consecutive images, $\mathbf{W}$ is the 2D feature window, $\mathbf{w}(m)$ is a windowed weighting function, typically a Gaussian and g_x and g_y stand for the horizontal and vertical image gradients. This is the basic tracking equation and theoretically it can be applied to any point on an image. However it is a linear

equation and it can only be solved reliably if the matrix $\mathbf{Z}$ is well-conditioned. Intuitively speaking, a good feature must be distinctive in order to be tracked reliably. Here, that intuition is algebraically justified. For a corner feature or a highly textured region, $\mathbf{Z}$ will be rank-2 which makes the above equation solvable. However for a line or a smooth region the rank will be either one or zero respectively so the equation is not well-conditioned, ie. the point is not trackable. With such an insight, a threshold on eigen-values of $\mathbf{Z}$ is used as a feature selection criterion.

It must also be noted that, Eq. (1.3) is only a linear approximation so the solution has to be iterated several times in order to converge to a real solution. In addition, in order to increase the robustness of the technique for large displacements, a Gaussian pyramid is typically built and the solution is refined in a course-to-fine manner.

In the canonical KLT framework, each feature is tracked independently from each other, without considering any possible geometric relationship between their displacements. This is both a blessing and a curse. It is a blessing because the technique becomes quite general and can be applied to many types of scenes such as dynamic scenes, but it is also a curse since a very good source of information is discarded. For example, if the 3D location related to a feature point is known, the corresponding feature in the next image can be precisely determined if the motion parameters of the next frame are available, which is a quite convenient way to recover the lost tracks. Consequently it is desirable to initiate the 3D geometry as early as possible in a SfM system. However 3D reasoning on a sequence is only possible if sufficient parallax is built-up (which is a result of translation) on the feature tracks and the amount of parallax generally increases as the number of processed images increase. As a compromise, we generally use 6-10 frames to initiate the structure.

The type of initialization algorithm depends on whether the internal camera calibration is known or not. If it is not available, a projective reconstruction is carried out after decomposing the computed fundamental matrix between the first and the last image into two projection matrices. Later, the cameras in between are computed from the reconstructed 3D points and the corresponding feature tracks. The celebrated Random Sample Consensus [FB81] (RANSAC) is applied to robustly estimate the $\mathbf{F}$ and $\mathbf{P}$ matrices. RANSAC is a simple recover-and-select approach where the minimum number (7 for $\mathbf{F}$, 6 for $\mathbf{P}$) of correspondences are used to generate many (typically several hundred depending on the estimated percentage of outliers) hypotheses and choose the one which explains the overall matches the best. Later a non-linear minimization scheme is applied to all inliers for a more suitable solution. In the case of calibrated cameras, although the general structure of the technique remains the same, the subroutines change significantly. Rather than applying the 7-point algorithm, the 5-point algorithm of Nister [Nis03] is used to compute the epipolar geometry. The cameras in between are computed by Grunert's [HCON94] 3-point pose estimation algorithm rather than the 6-point algorithm and the recovered structure is euclidean, not projective.

After the initialization is achieved, reconstruction and tracking of the new points and estimation of the new camera poses go hand in hand. Newly computed cameras pave the way to reconstruct new 3D points and with such points new camera poses can be estimated.

1.3.3 Self-Calibration

When there is no information available about the camera intrinsics, the reconstruction we generate has an inherently unsolvable projective ambiguity. This ambiguity is more clear if Eq. (1.1) is written in the form:

$$m \sim \mathbf{PTT}^{-1}M \tag{1.4}$$

where $\mathbf{T}$ is any 4×4 non-singular projective transformation matrix. It implies that there are an infinite number of solutions parameterized by the 15 degrees of freedom of the projective transformation and each member of this family explains the image data equally well. However, in most of the cases projectively distorted reconstructions are far from visually convincing since such transformations retain only the most primitive relationships in a 3D model, such as the point on a line or a plane remains on that line or plane after the transformation, incidence relationships do not change etc. However, the angles and the length ratios between the line segments may change significantly which result in unrealistic reconstructions. This problem can be solved in different ways by incorporating apriori information either on the scene, such as orthogonality of specific lines, or on camera motion, such as the camera goes through pure translation, or on the intrinsic parameters of the camera. Considering that the scene structure and the camera motion for hand-held camera sequences can be quite irregular, the most suitable and popular choice is to use the information on the camera intrinsics which can be assumed fixed or have only one parameter varying (typically the focal length) for most type of sequences.

Throughout this dissertation, we exploited two types of such apriori information on intrinsics to come up with a realistic reconstruction. Either the intrinsic values of the camera are computed beforehand or they are assumed to be unknown but fixed throughout the sequence. For the latter case, the reconstruction is upgraded to a scaled Euclidean form (similarity) by the help of Trigg's [Tri97] algorithm, for the former case which is known as calibrated SfM, the initial reconstruction already comes up in the similarity form so no self-calibration step is necessary. An advantage of the calibrated SfM is the fact that as there are less unknowns to be estimated, the robustness of the system increases, which is quite visible in nearly degenerate cases such as more or less planar scenes or small foreground objects.

1.3.4 Bundle Adjustment

If there are n_1 images and n_2 3D points, $11 * n_1 + 3 * n_2$ parameters need to be handled simultaneously in order to minimize a global error function, such as

the overall 2D reprojection error. This is a paramount number of parameters even for a quite modest number of images. Also the classical reprojection error is a non-linear function which renders the typical linear solutions inapplicable. Bundle adjustment is a specific routine [HZ00] which has been designed to alleviate those problems. It handles the non-linearity of the problem by iterative linearization of the error function in the vein of Newton-Raphson method. The real eccentric part of the algorithm comes from the observation that the 3D points and the camera parameters affect the error functional locally, which results in a sparse Jacobian matrix. Typically bundle adjustment is applied after all the images are processed. However, as the performance of good tracking and 3D geometry computation are interdependent, we polish the 3D points and the camera parameters with the bundle routine at certain intervals.

1.3.5 Dense Reconstruction

The system so far is only capable of computing 3D point clouds corresponding to some distinct feature points on the images and the camera parameters for each input frame. Although that limited output maybe sufficient for different types of applications, such as robot navigation, it is desirable to generate 3D depths for each pixel in the input images for realistic 3D graphics rendering.

The standard method to achieve such dense reconstruction is to employ stereo-matching algorithms, where two images are matched pixel by pixel by typically exploiting intensity similarities, ordering constraints by the help of dynamic programming techniques and smoothness constraints. A preprocessing step where both of the images are rectified with a projective transform is usually performed in order to come up with proper horizontal scanlines for both images. The stereo reconstruction problem is a relatively old and a well studied field but interestingly it is still an active research area where the interested reader can find a good review conducted by Szeliski [SS02]. One may also employ optical flow based techniques which compute the displacement of each pixel between two images however such techniques usually make no assumptions on the existence of global geometric constraints, such as the epipolar geometry, and consequently the search region becomes 2D not 1D which makes the problem more ill-conditioned.

There are also relatively new methods to come up with dense reconstructions. Faugeras and Keriven [FK98] describe a variational method where a surface is evolved with levelset based PDE's to come up with the best 3D representation. Kolmogorov and Zabih [KZ02] proposed a method where the 3D scene is discretized by parallel planes which enables to represent the solution space and associated error as a graph. Typical graph-cut algorithms can later be applied on this graph to find a good solution with minimum error. Another type of method which also discretizes the 3D space is called space or voxel carving, which typically assumes a hypothetical 3D grid in the bounding box of the viewed object and conducts a 3D search for the best possible surface representation. A survey of such volumetric methods has been made

by Slabaugh *et al.* [SCM*03]. Another interesting thread of research is from Strecha *et al.* [CS04, CS06] where probabilistic methods are deployed to generate 3D depth from multiple wide-baseline views. Imaging, occlusions, and outliers are modeled with generative models and the most likely model is inferred in an Expectation-Maximization framework.

Throughout the system that is described in this dissertation, we used a stereo-matching based method to compute the depth-maps of the images whenever needed. However the main contributions of this work are independent of how the dense reconstruction is achieved.

Chapter 2

The Relative Scale Ambiguity

The 3D reconstruction of scenes containing independently moving objects from uncalibrated monocular sequences poses serious challenges. A quite subtle and critical problem is the resolution of the unknown relative scale values between the reconstructed objects in a dynamic scene. Indeed, an issue that did not receive much attention so far is that 3D reconstructions of multiple, independently moving parts – be it background or moving objects – can only be determined up to an unknown, relative scale. Even though the uncertainty about their individual absolute scales is usually of little importance (their visualization on screens will typically introduce a scaling anyway), their undefined relative scales come to haunt us as soon as we want to reconstruct the dynamic scene as a whole. As will be demonstrated, lack of information on the relative scales leads to one-parameter families of possible trajectories of the objects with respect to the static background. Consider the example of a video of a moving car. Without incorporating further knowledge about the world, a computer cannot distinguish between a small toy car hovering in front of the camera or a real car at a larger distance on the road. Apart from some rather loose constraints coming from depth-of-field considerations, there is no other solution to this problem than to introduce either cognitive information, or to introduce more generic types of criteria on the expected scales and trajectories. In this dissertation, we follow the latter approach and propose two such criteria.

Our approach for the analysis of dynamic scenes is rather generic and it is based on motion constraints that exist in a scene. The emphasis is not on detecting independent motions, nor on segmenting the moving objects. These are the subjects of the chapter 5. Assuming that segmentation has been done as a preprocessing step (several techniques [CK95, MMI02, VM04, Tor98, SM98, VL97] have already been proposed to achieve this) and that the moving objects are rigid, we want to reconstruct the trajectories of the different dynamic parts

of the scene with respect to each other. We require the segmentation to be sufficiently precise in order to enable an uncalibrated SfM algorithm to extract robust projection matrices.

Unlike prior SfM contributions for dynamic scenes, we propose assumptions that are of a more probabilistic than a strict geometric nature (as in papers that presume specific object motions). In this sense, the proposed approach tends to be more generic. The price we pay is that there are no hard guarantees that the assumptions we make actually hold, but for typical footage there is a high probability that they do. The long term goal of computer vision is to make complete and detailed reconstructions of scenes with multiple, independently moving objects and we focus in this chapter on the determination of the scale of relative trajectories as a first problem to solve. The findings of this study have been published in the papers [OCVV04, OCE04].

This chapter is organized as follows. Section 2.1 discusses the relative scale problem in more detail. In that section we derive the basic equation that underlies the two proposed scale selection criteria: the independence criterion and the non-accidentalness criterion. Section 2.2 discusses the trajectory independence criterion more extensively, and its practical usefulness is corroborated with experimental results. Section 2.3 discusses the non-accidentalness criterion, again demonstrating its use on the basis of experimental results. In section 2.4, the issue of combining multiple motion constraints is discussed. Section 2.5 summarizes the main ideas of the chapter and suggests future work.

2.1 The Relative Scale Problem

It is known that from an uncalibrated monocular image sequence, we can only come up with a reconstruction up to an unknown overall scale. To give an intuitive example, consider a video that is showing a single rotating 3D cube. Although the euclidean structure of the cube and its rotation parameters can be deduced from the video, there is a scale-depth ambiguity, i.e. a big cube that is far away would generate exactly the same images as a small cube that is close to the camera. The distance to the camera is basically a translation parameter, hence in case of a general object motion there is an overall scale ambiguity both on the structure of the object and its translation parameters.

Consequently, when the scene contains different rigid parts, moving independently of each other, there is a problem in deciding on the relative scale of the translation, but not on the rotation. The relative rotations are fixed at each time instant and not affected by different scale factors unlike the relative translations. For each different relative scale factor between the background and the independently moving object, a different trajectory for the object relative to the background will result. Consider an image sequence of a scene which is static except for one rigid, independently moving object. The restriction to one moving object is in fact not essential, but is introduced to simplify the discussion. Additional moving objects can be dealt with similarly. Suppose we

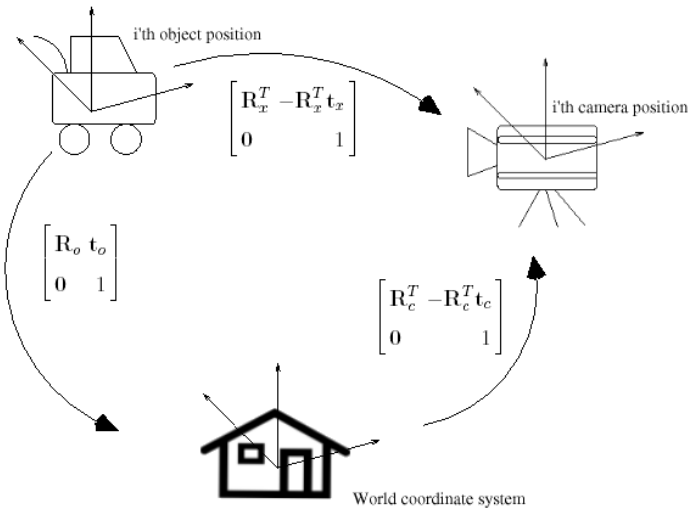


Figure 2.1: *Transformations between the static and the dynamic part of the scene.*

can compute the camera's orientation and position relative to the static part of the scene –'the background' which also yields the world coordinate system – and with respect to the segmented moving object for every frame i of the sequence. The 3×3 rotation matrices $\mathbf{R}_c^i$ and $\mathbf{R}_x^i$ represent these two orientations respectively, and similarly, the 3×1 translation vectors $\mathbf{t}_c^i$ and $\mathbf{t}_x^i$ represent these positions. What we would like to find is the rotation $\mathbf{R}_o^i$ and the translation $\mathbf{t}_o^i$ which represent the motion of the object with respect to the background for every frame i.

These transformations and their corresponding notation are illustrated in Fig. 2.1. The relation among them can be written as:

$$\begin{bmatrix} \mathbf{R}_x^T & -\mathbf{R}_x^T\mathbf{t}_x \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_c^T & -\mathbf{R}_c^T\mathbf{t}_c \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_o & \mathbf{t}_o \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.1}$$

in which we dropped the frame indices for the sake of compact notation. ^{T} stands for the matrix transpose and $\mathbf{0}$ is a 1×3 vector of zeroes. The right hand side of this expression basically says that, to transform a point in the object local coordinate system to camera coordinates, first transform the coordinates in the object frame to the world frame with rotation $\mathbf{R}_o$ and displacement $\mathbf{t}_o$, and then transform them to the camera coordinate system using $\mathbf{R}_c^T$ and

$-\mathbf{R}_c^T\mathbf{t}_c$. The left hand side represents the relative transformation between the object and the camera directly.

The rotation and translation parts of equality (2.1) yield

$$\mathbf{R}_x^T = \mathbf{R}_c^T\mathbf{R}_o \tag{2.2}$$
$$-\mathbf{R}_x^T\mathbf{t}_x = \mathbf{R}_c^T\mathbf{t}_o - \mathbf{R}_c^T\mathbf{t}_c \tag{2.3}$$

If we know $\mathbf{R}_x$ and $\mathbf{R}_c$ in addition to the exact $\mathbf{t}_x$ and $\mathbf{t}_c$, we can extract $\mathbf{R}_o$ from Eq. (2.2) and $\mathbf{t}_o$ from Eq. (2.3). Unfortunately as mentioned before, uncalibrated SfM cannot extract the camera motion with respect to the background and the object at an absolute scale due to a scale ambiguity in the translation components. We are free to fix the scale for one, say the background, but there still remains the relative scale to deal with. Equation (2.3) only holds at the correct relative scale of the translational motion components, $\mathbf{t}_c$ and $\mathbf{t}_x$. As we do not know which scale to apply, each incorrect scale $s \neq 1$ applied to $\mathbf{t}_x$ will yield a different object trajectory $\mathbf{t}_{os}$:

$$s(-\mathbf{R}_x^T\mathbf{t}_x) = \mathbf{R}_c^T\mathbf{t}_{os} - \mathbf{R}_c^T\mathbf{t}_c \tag{2.4}$$

Merging Eqs. (2.3) and (2.4) yields the following relation between the actual trajectory of the object $\mathbf{t}_o$ and the computed trajectory $\mathbf{t}_{os}$ of the object when an incorrect relative scale factor $s \neq 1$ is used:

$$s(\mathbf{R}_c^T\mathbf{t}_o - \mathbf{R}_c^T\mathbf{t}_c) = \mathbf{R}_c^T\mathbf{t}_{os} - \mathbf{R}_c^T\mathbf{t}_c \tag{2.5}$$

Multiplying both sides with $\mathbf{R}_c$ leads to

$$\mathbf{t}_{os} = s\mathbf{t}_o + (1-s)\mathbf{t}_c \tag{2.6}$$

Hence, the object translation $\mathbf{t}_{os}$ found for the relative scale s is a linear combination of the true object translation $\mathbf{t}_o$ and the camera translation $\mathbf{t}_c$. When $s = 1$, i.e. at the correct scale, $\mathbf{t}_{os}$ equals $\mathbf{t}_o$. For values of s other than 1, $\mathbf{t}_{os}$ will always be contaminated with the camera translation. When s gets closer to zero, $\mathbf{t}_{os}$ evolves towards the camera path. Consider the aforementioned car example where we wish to distinguish between a toy car and a real car. The toy car has to move along with the camera in order to generate the same images as the real car did. As the toy car gets smaller, i.e. s gets closer to zero, its path should lock on more and more to the camera path, following its every jerky move.

Equation (2.6) is the key observation of the study in this chapter and the point of departure for two criteria that we propose for the analysis of dynamic scenes. Before introducing these criteria, we elaborate a bit further on this central equation. The translational components $\mathbf{t}_o$ represent the overall motion or 'trajectory' of the origin of the object coordinate system relative to that of the background. Other object points will have differently shaped paths due to the action of rotation. Nevertheless, similar considerations about the coupling with the camera trajectory hold for all object points. Assume $\mathbf{p}^0$ is the position

of a point on the object in the first frame. Its position $\mathbf{p}$ in frame i can be written as:

$$\mathbf{p} = \mathbf{R}_o \mathbf{p}^0 + \mathbf{t}_o \tag{2.7}$$

in which the frame index i is dropped again to simplify the notation. Similarly, its scaled version $\mathbf{p}_s$ moves according to the following transformation:

$$\mathbf{p}_s = \mathbf{R}_o \mathbf{p}_s^0 + \mathbf{t}_{os} \tag{2.8}$$

Without loss of generality, we can assume the fixed world coordinate system to be attached to the initial camera pose (this is also assumed for the rest of the chapter). Then, $\mathbf{p}_s^0$ is equal to $s\mathbf{p}^0$. Introducing this fact and Eq. (2.6) into the above equation yields:

$$\mathbf{p}_s = s(\mathbf{R}_o \mathbf{p}^0 + \mathbf{t}_o) + (1-s)\mathbf{t}_c \tag{2.9}$$

and by incorporating Eq. (2.7), it can be written as

$$\mathbf{p}_s = s\mathbf{p} + (1-s)\mathbf{t}_c \tag{2.10}$$

which is quite similar to Eq. (2.6). This reformulation highlights that, at the wrong scale, the position of each object point demonstrates a linear coupling with the camera position (as the world frame corresponds to the initial camera frame, the position of the camera - i.e.of its optical center - simplifies to $\mathbf{t}_c$).

Following Eq. (2.6) (and with similar conclusions from Eq. (2.10) when observing a specific object point), the systematic coupling with camera motion in case the wrong relative scale s has been chosen, will show up especially in situations where the actual translation $\mathbf{t}_o$ is statistically independent of that of the camera, which cannot be guaranteed of course. As a matter of fact, in cases where the camera tracks the object quite precisely, like a camera traveling alongside a car in a chase scene of a movie, there is a high dependence between object and camera motion. However, in the absence of relative camera-object motion 3D reconstruction of the object would not be possible anyway. On the other hand, approaches that search for the relative scale that maximizes the statistical independence of the camera and object motion seem to yield a promising avenue for many scenarios in which 3D reconstruction is possible. Such approaches can be expected to be successful when objects are not rigorously tracked by the camera, or when the camera does not move very smoothly. These conditions hold particularly well in cases where the images are taken with a hand-held camera. Equation (2.6) will let any jerkiness of the camera motion trickle through into the reconstructed object motion, except at the true scale $s = 1$. To summarize, following Eq. (2.6), there are many situations where the *assumption of maximal linear independence between camera and object motion in a statistical sense* will return a realistic solution. This is the first criterion that we propose for the determination of the relative scale.

Another, second criterion that can be brought to bear given Eq. (2.6) is so-called *non-accidentalness*. This principle has been introduced and successfully

exploited first in the area of visual grouping and object recognition [Low87]. For our problem, if for a particular scale a special object motion results, e.g. the trajectory would be planar, then there is a high probability that such a solution does not exist by sheer accident and that it reflects the true motion. Indeed, it is highly unlikely that such a planar solution is found among the one-parameter family of possible trajectories in the general case. Other examples of non-accidental properties would be trajectories demonstrating straight segments or parts of them having identical shapes, e.g. in a periodic motion. The addition of a motion dependent on that of the camera following Eq. (2.6) would normally destroy such special properties unless the camera motion exhibits the same type of regularity in sync with the object motion. Again, this is unlikely to happen and when it does, the property will typically be shared among all trajectories. In the latter case, the criterion fortunately yields no solution, rather than an incorrect one.

These two assumptions – *independence* and *non-accidentalness* – will now be further explored in terms of their practical use. The next sections give a more detailed description of our implementation of these assumptions and their usefulness is corroborated through experiments.

2.2 The Independence Criterion

2.2.1 Measuring independence

Eqs. (2.6) and (2.10) express that at scales other than the correct one the object translations as well as the positions of object points are coupled to those of the camera. If the true object and camera motion are not linearly dependent (in the statistical sense), such linear dependence will appear as soon as s starts to deviate from its correct value $s = 1$. Hence, in this section we will search for the correct scale as the one yielding the object trajectory that is least correlated with the camera motion.

As a matter of fact, taking derivatives with respect to time for both the left and the right hand sides of Eq.(2.10) yields a similar observation for velocities or accelerations of all object points or any point rigidly connected to the object for that matter. There is good reason to consider whether the exploitation of this velocity or acceleration version would actually not be more appropriate. It is quite usual for the camera to follow the objects of interest to some extent. In such cases the positions will typically be quite dependent, reducing the power of Eq. (2.10). On the other hand, the instantaneous motions of the camera(man) and the object will typically not fluctuate in a similar way, making their velocities less dependent. Therefore, we have used the velocity version of Eq. (2.10). By the same argument, accelerations could even be more effective, but the noise introduced by taking the additional derivative is a factor to be reckoned with. In our experiments we have found velocities to give better results than positions, and accelerations to perform slightly worse than velocities. Compared to the positions, the velocities also tend to be more stationary, i.e.their statistical

properties change less over time and they are better distributed about their mean, therefore making them more suitable for statistical calculations.

It is also important to decide on which point represents best the object path in the first place. For several reasons, we take the centroid of the object point cloud (at least in this section), called $\mathbf{p}_g$. First of all, the centroid is the most natural approximation of the reconstructed point cloud as suggested in Vidal *et al.* [VSMS02a]. Secondly, in practice the relative projection matrices computed by SfM for the object have their highest validity near the reconstructed points. The centroid nicely lies in their midst.

Putting $\mathbf{p}_g$ and $\mathbf{p}_{gs}$ in Eq. (2.10) and isolating $\mathbf{p}_g$ on one side, the equation becomes

$$\mathbf{p}_g = m(\mathbf{p}_{gs}) + (1-m)\mathbf{t}_c \tag{2.11}$$

where we introduced $m = 1/s$. When Eq. (2.11) is compared to Eq. (2.10), we see that both describe a one-parameter family of solutions. Yet, Eq. (2.10) describes the family starting from the true object motion $\mathbf{p}_g$ (hence, $s = 1$ is the correct scale), whereas in our experiments Eq. (2.11) will be used to describe the same family but from a solution generated by an SfM algorithm ($\mathbf{p}_{gs}$) which is randomly scaled. Therefore the m which corresponds to the true object motion will no longer be found at ($m = 1$) but at the inverse of the random scale coming out of SfM.

Differentiating Eq. (2.11) with respect to time yields the preferred velocity version:

$$\mathbf{v}_g = m(\mathbf{v}_{gs}) + (1-m)\mathbf{v}_c \tag{2.12}$$

The question we want to answer is, given the *scaled* object velocities $\mathbf{v}_{gs}$ and the camera velocities $\mathbf{v}_c$ (both output of SfM), what is the value of m which makes the computed object velocities as independent as possible of the camera velocities. These velocities are approximated as follows:

$$\mathbf{v}_c^i = \mathbf{t}_c^{i+1} - \mathbf{t}_c^i \tag{2.13}$$

$$\mathbf{v}_{gs}^i = \mathbf{p}_{gs}^{i+1} - \mathbf{p}_{gs}^i \tag{2.14}$$

Note that the camera velocity is fixed as soon as we fix the scale of the background. As to the assumption of minimal dependence between object and camera motion, a final issue is to decide on how to quantify this 'independence'. There are various methods that can be applied. One set of related techniques have been investigated under the name of Independent Component Analysis (ICA), see [HKO01]. In a canonical ICA problem, the input is various signals which are linear mixtures of original source signals. The aim is to compute back both the original source signals and the mixing matrix by just using those scrambled signals. The solutions typically assume the statistical independence of source signals and they try to find an inverse mixing matrix that would result in source signals that are maximally independent.

In addition to statistical independence, a critical assumption in ICA applications is the assumption of non-gaussianity of the source signals. The central

limit theorem states that as different random variables are added to each other, the resulting distribution looks more like a gaussian distribution. Consequently, linear mixtures of independent source signals will always be more gaussian than the original signals as long as they are not gaussian already. Hence, the search for the most non-gaussian source signal among possible solutions is a key idea behind ICA.

There are prominent similarities between such type of problem and ours. The object trajectory and the camera trajectory can be considered as two source signals and then the resultant ambiguous object trajectory is a linear mixing of those two source signals with parameter s. The aim is to find both the original source signal for the object motion and the mixing factor s. However in our case, in contrast to the original ICA problem, one of the sources (the camera motion) is known. Nevertheless the basic tools used in ICA are still applicable.

One set of techniques directly measures the non-gaussianity of the computed source signals. The classical measure of non-gaussianity is kurtosis or the fourth-order cumulant. It basically measures the spikiness of the distribution and can be written as:

$$K = \frac{m_4}{m_2^2} - 3 \tag{2.15}$$

where m_4 is the forth moment around the mean and m_2 is the variance. Another such measure of nongaussianity is given by negentropy. Negentropy is based on the information-theoretic quantity of entropy [Sha48]. The entropy of a random variable can be interpreted as the degree of information that the observation of the variable gives. As the unpredictability of the event increases, the entropy increases. One basic result of information theory is that gaussian random variables have the largest entropy among all random distributions with the same variance. Negentropy measures the difference between the entropy of a random process and the entropy of a gaussian random process with the same variance. As this difference gets bigger, the process is less gaussian.

As mentioned, our setting is better conditioned than typical ICA as we know one source signal already, namely the camera trajectory. Therefore, it is better to directly measure the statistical dependence between the computed object motion and the camera motion. One such measure is mutual information [Gui77]. It measures how much information the knowledge of one random variable would give on the value of another variable. If it does not give any information, it means that those two variables are independent and mutual information is 0. However as those two variables are more strongly coupled, the mutual information increases. Given two discrete scalar random variables a and b, the mutual information between them is defined to be:

$$MI(a,b) = (\sum_{i=0}^{bins}\sum_{j=0}^{bins} h_{ij}^{ab} log(\frac{h_{ij}^{ab}}{h_i^a h_j^b})) \tag{2.16}$$

where h^{ab} stands for the joints pdf of a and b, h^a and h^b stands for the pdf of a

and b respectively . They are approximated by normalized histograms. If a and b are independent, h_{ij}^{ab} should be equal to $h_i^a h_j^b$, so the above expression becomes zero. The formulation of mutual information is highly related to the Kullback-Leibler divergence. It measures the difference between two distributions. In this case the divergence is computed between the joint probability of h^{ab} and the product of the marginals h^a and h^b. Consequently, it can be said that Mutual Information is the divergence of the observed distribution h^{ab} from a hypothetical distribution where the two process are assumed independent.The effectiveness of the expression (2.16) depends on two critical factors, choosing a good bin size for the histograms and having large number of samples. A wrong bin size or not having enough samples for a certain binsize deteriorates the performance substantially. In order to mitigate the serious problem of needing many samples, we resort to a simplification that seems to work well. The mutual information is calculated for the x, y, and z components of the motions separately, and are then added.

Although mutual information is a very natural measure of independence, considering above issues we opted for classical correlation as more appropriate criterion here. Indeed, this measure specifically probes for the linear dependence between object and camera motions which we have to void here. Classical correlation is an effective criterion then and it does not require the selection of a proper bin size, can be initiated with a polynomial solution and is also not as sensitive as mutual information to the number of samples.

However a subtle issue is how to define this correlation. First of all, we prefer to use the (normalized) correlation coefficient rather than just (unnormalized) correlation. Although this choice is more complex, it eliminates sensitivity to the variances and sizes of the motion vectors. Unnormalized correlation would tend to prefer small motion vectors over big ones. As actual motion vectors could be quite big in magnitude, they must not be penalized for that. From Eq.(2.11), it is seen that only corresponding vector components are interacting with each other. Sticking to the useful concept of correlation coefficients and taking all components equally into account, we suggest the following correlation criterion:

$$E = \sqrt{\sum_{k=1}^{3} (W_k)^2} \tag{2.17}$$

$$\text{with } W_k = \frac{\sum_{i=1}^{n-1} \overline{\mathbf{v}}_g^i(k) \overline{\mathbf{v}}_c^i(k)}{\sqrt{\sum_{i=1}^{n-1} (\overline{\mathbf{v}}_g^i(k))^2} \sqrt{\sum_{i=1}^{n-1} (\overline{\mathbf{v}}_c^i(k))^2}} \tag{2.18}$$

where n is the number of frames, k = 1,2,3 corresponds to the three velocity components along the x,y and z axes, respectively, and the over-lined vectors are mean-shifted. Eq. (2.18) is the correlation coefficient between the k^{th} component of the object velocity and the k^{th} component of the camera velocity. We expect this correlation criterion to be minimal when $\mathbf{v}_c$ and $\mathbf{v}_g$ are as linearly independent as possible. There are three terms in Eq. (2.17) and each

of them turns out to be the ratio of two second degree polynomials in m. To minimize E analytically, one can solve for the roots of its derivative, however this is equivalent to solving a 11^{th} degree polynomial in m. To avoid such a high degree in the polynomial, one can neglect the normalizing denominator in Eq. (2.18) at first to come up with an initial solution (by minimizing the corresponding quadratic equation in m) and make an iterative search starting from this value for the expression including the denominator. This is the strategy we have followed in our experiments. It should also be noted that since Eq. (2.17) is a statistical criterion, we need a sufficient number of frames to have a statistically meaningful estimate. This usually is no problem as the input generally is a video consisting of hundreds of frames.

2.2.2 Experimental results

We report here on two experiments to test the validity of the correlation criterion. The first one is a controlled experiment. Figure 2.2 shows six images of a video of 400 frames where a ball is attached to a robot arm and the robot arm is moving against a static background. While recording the video, the hand-held camera was moving as well. The trajectory of the ball with respect to the background consisted of three straight segments, each one parallel to one of the world coordinate axes. The robot end effector's motion was programmed in this way to serve as ground truth. Our uncalibrated SfM method (described in [PVV*04]) was used to reconstruct the 3D shape of the background and the ball, in addition to the relative camera motion with respect to both. The moving part of the robot could not be reconstructed since it has very little number of features on it.As previously explained, the two reconstructions were obtained with unrelated scales. Using the correlation criterion, we searched for the scaling factor to be applied to the ball in order to get to the correct relative scale. A view on some representatives of the one-parameter family of possible ball trajectories with respect to the background for different values of m is given in Fig. 2.3. As can be seen, most trajectories have lost much of the simplicity of the true motion. As soon as one deviates from the true relative scale, the influence of the camera motion kicks in, rendering the reconstructed motion more complicated.

Although mutual information is not the independence criterion we adopted throughout this dissertation, it is informative to check its performance (For the rest of dissertation we used classical *normalized* correlation measure Eq.(2.17) as the independence criterion). Looking at the evolution of the mutual information measure Eq.(2.16) over different scales (see Fig. 2.4), one observes the existence of a global minimum that is quite outspoken. Having said this, there are also some local minima, which preclude simple gradient descent schemes from being used. A numerical optimization algorithm, like Simulated Annealing, can be used here.The trajectory corresponding to the global minimum of the mutual information measure is shown in Fig. 2.5 (crosses) overlaid on the true trajectory (circles) which is known from the robot motion. As can be seen,

the solution comes quite close to the ground truth.

Looking at the evolution of the combined correlation measure that is defined in Eq.(2.17) over different scales m (see Fig. 2.6), one observes the existence of a global minimum that is quite outspoken. There is a maximum at $m = 0$, which is expected since $\mathbf{v}_o$ degenerates to $\mathbf{v}_c$ for $m = 0$ according to Eq. (2.12). This extreme case results in a maximum correlation between the object and the camera trajectory (i.e.the correlation coefficient W_k is one for each of the three spatial components, so the maximum value of E in Eq. 2.17 is $\sqrt{3} \simeq 1.73$). The trajectory corresponding to the global minimum of the correlation measure is shown in Fig. 2.7, overlaid on the true trajectory which is known from the robot motion. The points are subsampled for better visualisation. The robot has a precision of 0.15 mm, which is highly accurate compared to that of our SfM calculations. Hence, we may take the robot's input as the ground truth. The total robot trajectory was 210 cm in length and the average deviation between the ground truth and the reconstructed trajectory was 1.9 cm. As can be seen, the solution comes quite close to the ground truth.

For the second experiment we chose a more realistic scene where a person walks in front of a building. Three images from 350 input frames can be seen in Fig. 2.8. We were only able to reconstruct the upper part of the body since that is the only reasonably rigid part. The first row in Fig. 2.9 shows the reconstruction of the scene including the rescaled reconstruction of the walking person at the correct scale. All the images in that row of Fig. 2.9 correspond to the same time instant as the middle frame in Fig. 2.8, but from different (virtual) camera positions. The second row shows the same type of results, but now at a wrong relative scale. As can be seen in the middle image of this second row, the projected scene for a virtual camera looking from the real camera's point of view for that frame, is the same as that for the correct scale. Indeed, seen from the original cameras the entire sequence of projected reconstructions will be the same, irrespective of the chosen scale. It is only when seen from other directions (first and third columns) that the scale inconsistencies of the second row become apparent.

2.3 The non-accidentalness principle

As Eq. (2.6) suggests, it would be unlikely that other object trajectories in the one-parameter family of possibilities share the same regularities or 'non-accidental' properties that the true one may have. This would call for the camera path to exhibit similar, synchronized regularities. Taking the true object trajectory in the experiment of Fig. 2.2 as an example, for a trajectory at a wrong scale to exhibit an equally simple shape of three linear segments, the camera would also have to move linearly, changing course simultaneously with the object and showing velocity patterns directly related to it. As Fig. 2.3 demonstrates, in the case of a different type of camera motion, other members of the family of trajectories loose this property quickly as one moves away from

Figure 2.2: *Six frames from the robot sequence. The robot moves the ball through the static scene along a trajectory consisting of three linear segments. The hand-held camera moves around while taking the images.*

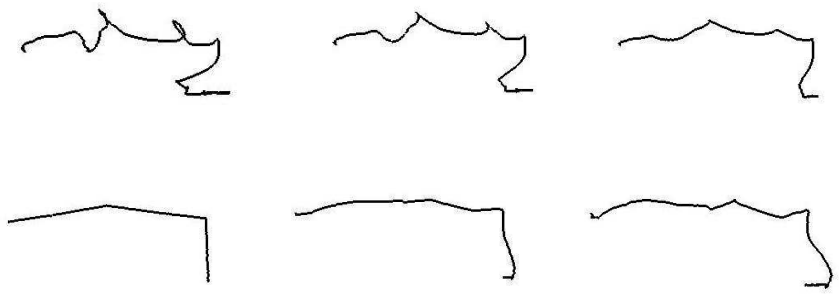


Figure 2.3: *Object path for different scales. The relative scale factors (m) applied to the SfM extracted ball trajectories are* $0.20, 0.25, 0.30, 0.36, 0.40, 0.45$ *respectively.* 0.36 *is the correct one (bottom left).*

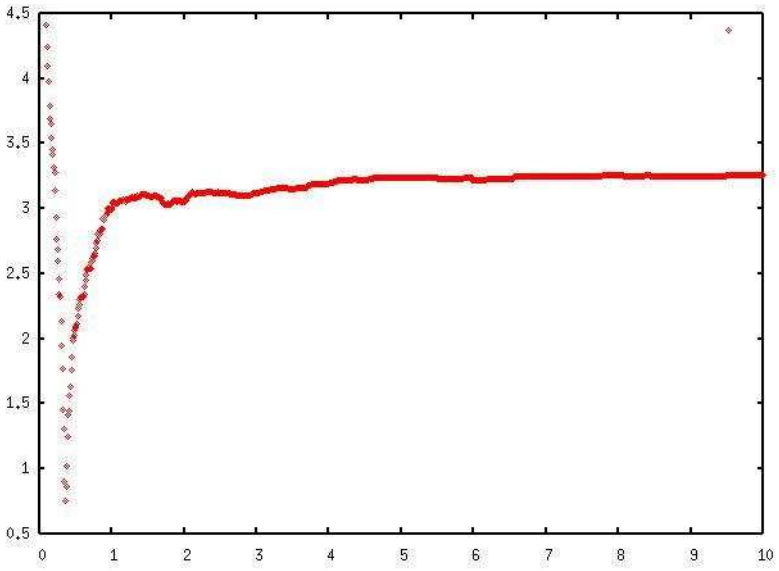


Figure 2.4: Mutual information graph for different rescale factors. 0.36 is the minimum.

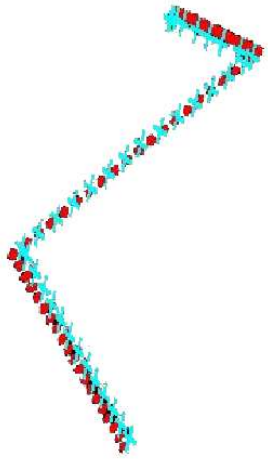


Figure 2.5: Alignment of the original object motion with the trajectory coming from mutual information criterion. Circles denote original motion, crosses denote rescaled trajectory.

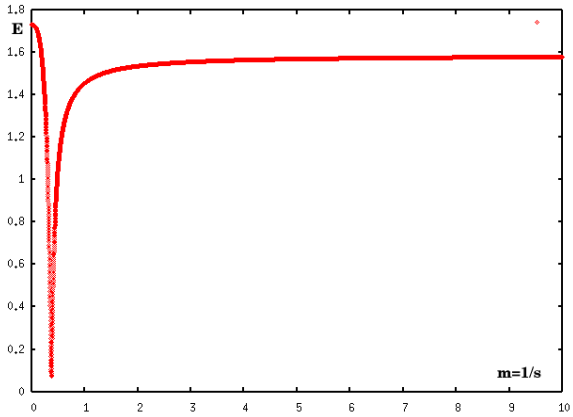


Figure 2.6: *Correlation graph (Eq. (2.17)) for different relative scale factors m; 0.37 is the minimum which is quite close to the actual value, 0.36. The theoretical maximum value for* $\boldsymbol{E}$ *is* $\sqrt{3} \simeq 1.73$

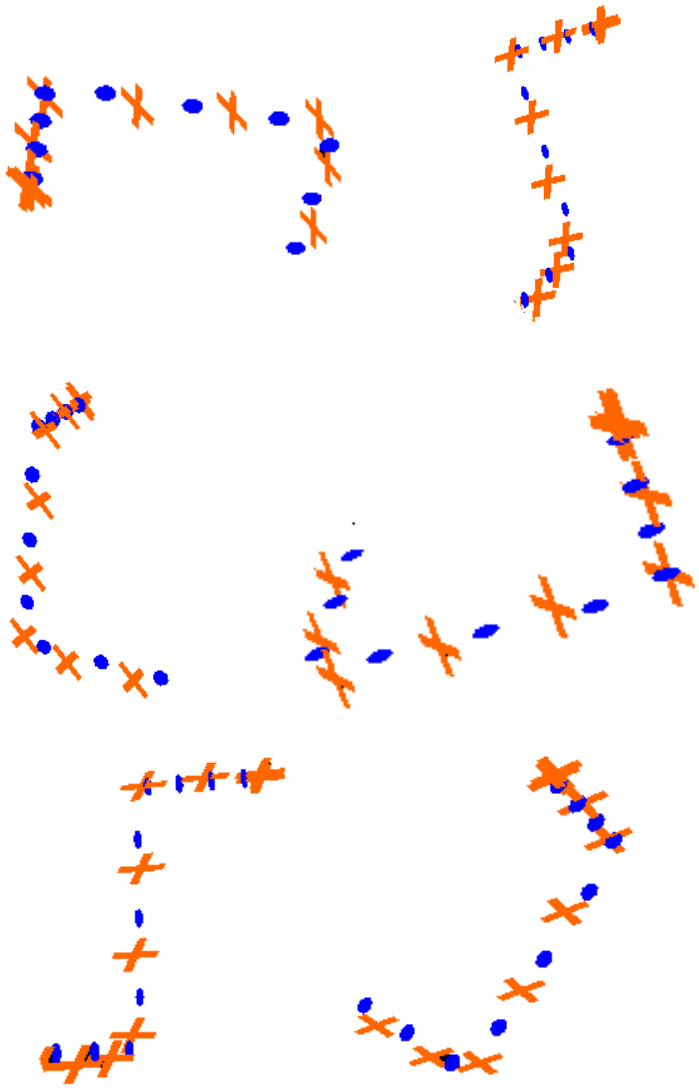


Figure 2.7: *Different views of the trajectory with minimal correlation aligned with the original object motion. Circles denote the ground truth motion, crosses denote the rescaled trajectory reconstruction.*

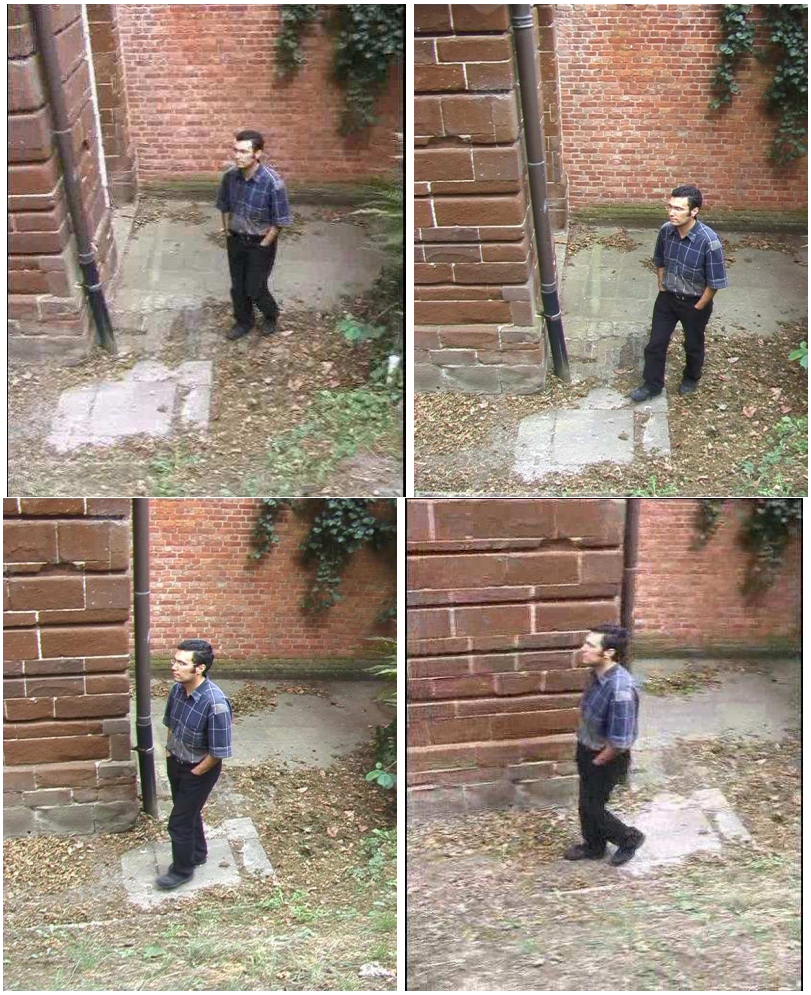


Figure 2.8: *Four images from the* 350 *frames long human walking sequence.*

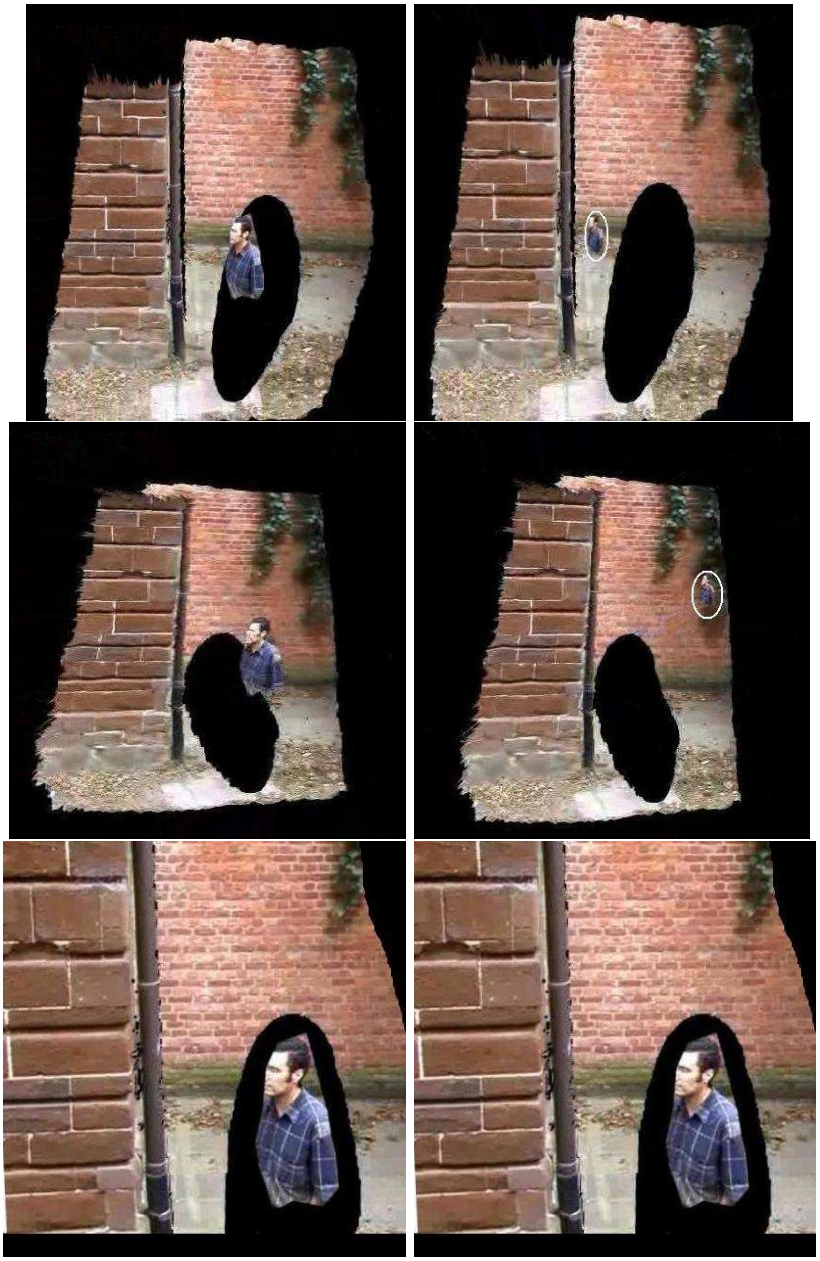


Figure 2.9: *Reconstruction of the human walking sequence. The first column shows three views on the reconstruction at the correct scale. The second column does the same, but at an incorrect scale. The reconstructed body is demarcated with an ellipse when necessary. All these reconstructions correspond to the same time but from different camera positions. It is only when the virtual camera deviates from the original video camera path (first and second row) that the inconsistencies at wrong scales become apparent.*

the correct scale since they are affected by the camera motion.

Piecewise linear motion is only one example of a long list of possible, non-accidental trajectory properties. As already mentioned, alternatives could be periodicity, planarity, but also properties shared by different object trajectories like being the same, albeit with a possible, temporal delay (e.g. one car following another on the same lane). It is clear that the detection of such solutions among the many possibilities is a research program in its own right. Therefore, the discussion here is limited to demonstrating the usefulness of two simple but very effective non-accidentalness properties: planarity and the heading constraint, explained respectively. In the next subsection, we first give examples where the existence of a planar object trajectory is taken to be an indication that this is the correct one. Planar motions are particularly important in practice. Such cases are often found for objects moving on a ground plane. Then, we discuss the second non-accidentallnes criterion which we experimented with: the "heading constraint", which exploits the non-holonomic motion of several object types.

2.3.1 The planarity constraint

Before discussing our implementation and examples of the planarity criterion, we want to highlight a caveat. The usefulness of the planarity criterion depends on the coupled nature of the object and camera trajectories. Indeed, the criterion will only supply us with a solution if there is only one planar trajectory in the one-parameter family. As we show next, there are degenerate situations where this is not the case.

First, let us write the points along the planar object trajectory as

$$\mathbf{p}^i = \mathbf{p}^0 + \alpha^i \mathbf{q}_1 + \beta^i \mathbf{q}_2 \tag{2.19}$$

where $\mathbf{p}^0$ and $\mathbf{p}^i$ are the point positions at frame 0 and i respectively, α^i, β^i are real numbers with $\alpha^0 = \beta^0 = 0$ and $\mathbf{q}_1$ and $\mathbf{q}_2$ are orthonormal 3-vectors spanning the plane through $\mathbf{p}^0$. These vectors are chosen as the first two of three basis vectors necessary to span the 3D space. We also introduce a third basis vector, $\mathbf{q}_3$ that is perpendicular to the first two. Then, the points along the camera trajectory can be written as

$$\mathbf{t}_c^i = \gamma_1^i \mathbf{q}_1 + \gamma_2^i \mathbf{q}_2 + \gamma_3^i \mathbf{q}_3 \tag{2.20}$$

$$\text{with } \gamma_j^0 = 0 \text{ , for } \; j = 1, 2, 3 \tag{2.21}$$

where the γ_j^i's are real numbers. The initial camera position is taken as the origin of our coordinate system, without loss of generality. From Eq. (2.10), for different values of s, the reconstructed trajectory will be of the form:

$$\mathbf{p}_s^i = s\mathbf{p}^o + s\alpha^i \mathbf{q}_1 + s\beta^i \mathbf{q}_2 + (1-s)(\gamma_1^i \mathbf{q}_1 + \gamma_2^i \mathbf{q}_2 + \gamma_3^i \mathbf{q}_3) \tag{2.22}$$

If we stack the observed points along the trajectory in a matrix, we arrive at the following formulation:

$$\begin{bmatrix} \mathbf{p}_s^{0T} \\ \mathbf{...} \\ \mathbf{p}_s^{iT} \\ \mathbf{...} \\ \mathbf{p}_s^{(n-1)T} \end{bmatrix} = s \begin{bmatrix} 1 \\ 1 \\ \mathbf{...} \\ \mathbf{...} \\ 1 \end{bmatrix} \mathbf{p}^{oT} + \mathbf{M} \begin{bmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \mathbf{q}_3^T \end{bmatrix} \tag{2.23}$$

$$\text{with } \mathbf{M} = \begin{bmatrix} s\,\boldsymbol{\alpha} + (1-s)\,\boldsymbol{\gamma}_1 & s\,\boldsymbol{\beta} + (1-s)\,\boldsymbol{\gamma}_2 & (1-s)\,\boldsymbol{\gamma}_3 \end{bmatrix} \tag{2.24}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}_j$ are the n-vectors whose elements are α^i, β^i and γ_j^i. Since the $\mathbf{q}_j$'s are orthonormal basis vectors, the rank of $\mathbf{M}$ determines the dimensionality of the trajectory. If the rank is zero it is the single point $\mathbf{p}_0$, if the rank is one the trajectory is a line, if the rank is two the trajectory is planar and if the rank is three the trajectory is a space curve. For $s = 1$, i.e.for the true object trajectory, $\mathbf{M}$'s rank is two as expected. For $s = 0$, the components from the object motion vanish and the rank is determined solely by the camera motion. For other values of s both the object and the camera trajectories affect the rank. For our criterion to work, this rank should be three for the wrong scales and two for the correct one. Otherwise we have a 'degenerate case'.

A closer inspection of $\mathbf{M}$ yields several basic degenerate cases. First of all, if *all* $\boldsymbol{\gamma}_3^j$ are zero i.e.the camera is moving parallel to the object plane, the rank is definitely two. Secondly, if *all* of the $\boldsymbol{\gamma}_j$'s are linear combinations of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ the rank still remains two. Such a degenerate case occurs when there exists a 3×3 transformation matrix $\mathbf{K}$ which operates as:

$$\begin{bmatrix} \boldsymbol{\gamma}_0 & \boldsymbol{\gamma}_1 & \boldsymbol{\gamma}_2 \end{bmatrix}^T = \mathbf{K} \begin{bmatrix} \boldsymbol{\alpha}^T \\ \boldsymbol{\beta}^T \\ \mathbf{0}^T \end{bmatrix} \tag{2.25}$$

This means that there exists a 3×3 affine transformation matrix which maps all the points on the object trajectory (translated by $-\mathbf{p}^0$) to the concurrent points on the camera trajectory. A third degenerate case is more subtle and it appears when the sub-matrix of $\mathbf{M}$ that is defined by $\mathbf{M}$'s last two columns have rank 1 rather than 2 which would be the general case (The rank deficiency on the first and the last column also causes this type of degeneracy but the analysis is the same so we only consider the last two columns case here). Such a condition happens when $\boldsymbol{\gamma}_2$, $\boldsymbol{\gamma}_3$ and $\boldsymbol{\beta}$ vectors have the same direction, i.e.they are scaled versions of each other. Although it is obvious that such a condition will result in a rank 2 matrix in general, the geometric interpretation is a bit harder to visualize than the previous ones. Consider a case where the camera is also moving on an arbitrary plane and without loss of generality, $\mathbf{q}_1$ points in the direction of the line that is the intersection of the two motion planes. Now consider the one parameter family of planes with a fixed normal direction where this normal is orthogonal to $\mathbf{q}_1$. This corresponds to the case

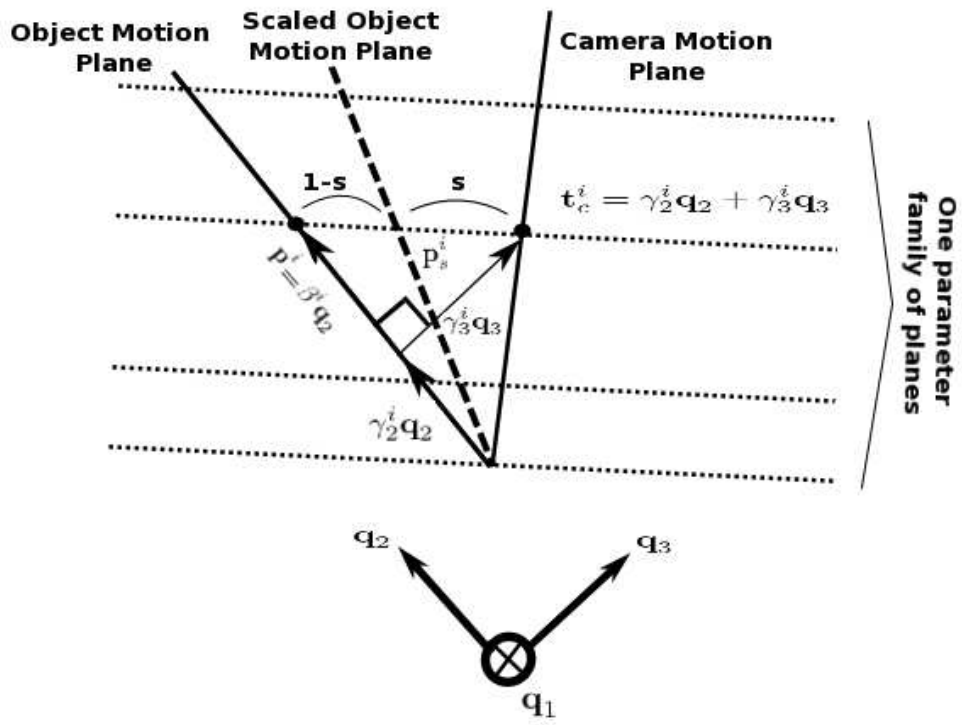


Figure 2.10: *The description of the third type of degenerate case for planar motion (see the text). All the planes are viewed in the direction of* $\boldsymbol{q}_1$.

where each member of this family of planes intersect the two original motion planes (the motion planes of the camera and object) in two lines parallel to $\mathbf{q}_1$. Differently stated, when those two planes and a member of the one-parameter family project to a 2D view in the direction of $\mathbf{q}_1$ as 3 distinct 2D lines (see Fig.2.10). If at each time instant, the plane which is parallel to $\mathbf{q}_1$ and contains the camera position and object position, is a member of that one-parameter family, then we have that degenerate case. Fig.2.10 is a depiction (with an abuse of notation where $\mathbf{p}^o$ is dropped and the 2D quantities have the same name as their 3D counterparts) when all the planes are projected orthogonally to a view in the direction of $\mathbf{q}_1$. Due to the rule of similar triangles, when the camera position $\mathbf{t}_c^i$ and the object point $\mathbf{p}^i$ are on the same member of the one parameter family of the planes, the scaled point $\mathbf{p}_s^i$ always remain on the same plane and $\boldsymbol{\gamma}_2^i$, $\boldsymbol{\gamma}_3^i$ and $\boldsymbol{\beta}^i$ always have a fixed ratio which results in a rank-2 $\mathbf{M}$ matrix. Note that any free motion in the $\mathbf{q}_1$ direction has no effect on this result. The degenerate case mentioned last can be considered to lie

the mid-way between the first and the second one. In the first case the camera trajectory is restricted to a plane parallel to that of the object motion but there is no further restriction on its shape, i.e.it is free to move freely on that plane. All trajectories in the one-parameter family lie in parallel planes. In the second case, two motions could be on arbitrary planes, but they are highly coupled as there is a fixed affine transform between them. However, in the last case, the two motions are free only in one direction ($\mathbf{q}_1$), but the other parameters must have a fixed scale relationship similar to an affine transform.

One trivial example where the conditions for a degenerate case are avoided is when the camera's path is 3D. However this is not a necessary condition. The constraint even works when the camera is moving on a line but this line should not be parallel to the plane of the object trajectory, and the camera positions should not be linearly coupled with the object trajectory as discussed before.

As a conclusion from this succinct and as yet incomplete discussion, if the true object trajectory is planar, it is not guaranteed to be the only one with this property in the one-parameter family of solutions. One should check for the occurrence of degeneracies.

In case the planarity property distinguishes the true solution from the rest, we still have to identify it. One way of finding a planar object trajectory among the one-parameter family is by performing a Principal Component Analysis (PCA) on the positions $\mathbf{p}^i$ (after shifting them to align their average position with the origin as is usual in PCA). From equations (2.2), (2.3) and (2.7) we find the trajectory of a point on the object as:

$$\mathbf{p}^i = \mathbf{t}_c^i + \mathbf{R}_o^i m \mathbf{p}_s^0 - \mathbf{R}_o^i m \mathbf{t}_{xs}^i \tag{2.26}$$

where $m = 1/s$, $\mathbf{p}_s^0 = s\mathbf{p}^0$ and $\mathbf{t}_{xs} = s\mathbf{t}_x$. Such substitutions are necessary since SfM returns only a scaled version of these vectors. The search for the planar trajectory in the one-parameter family proceeds in two steps. We start with an initialization that is based on a suboptimal but simple criterion. In a second step the solution is refined.

For the initialization of the relative scale, we take the one that minimizes the determinant of the corresponding 'covariance matrix'. The determinant of this matrix indicates the volume of space spanned by the principal components. If all the 3D points along a trajectory lie close to a plane, this volume is close to zero. First, we translate both sides of Eq. (2.26) to its mean. We get an expression of the form $m\mathbf{A}^i + \mathbf{B}^i$ in which $\mathbf{A}^i$ represents the mean-shifted $\mathbf{R}_o^i \mathbf{p}_s^0 - \mathbf{R}_o^i \mathbf{t}_{xs}^i$ and $\mathbf{B}^i$ represents the mean-shifted $\mathbf{t}_c^i$. The expression we want to minimize is:

$$V = det\Big(\sum_{i=0}^{\#points} (m\mathbf{A}^i + \mathbf{B}^i)(m\mathbf{A}^i + \mathbf{B}^i)^T\Big) \tag{2.27}$$

This is a polynomial of 6^{th} degree and it would *ideally* be zero at the correct scale m. In reality, it is the non-zero minimum that is of interest, due to noise. To minimize the determinant, we first take its derivative with respect to m, find the derivative's roots and apply them in the original polynomial (2.27) to find the global minimum. Since the determinant of a covariance matrix is always positive, this 6^{th} degree polynomial should converge to $+\infty$ for m approaching $\pm\infty$. This guarantees the existence of a global minimum. As to solving such polynomials, we use the eigen-value decomposition of the companion matrix [EM95].

This approach would work for any point on the object if the object would only translate in a plane and rotate about an axis orthogonal to it. In that case the motion of every individual object point is planar. We are however, interested in the more general case where there is some point that rigidly moves with the object and of which the motion is planar. As an example, take a ball that is rolling on a planar surface bouncing around upon contact with some obstacle at times. None of its observable points is performing a purely planar motion, although the global trajectory of the ball is planar. It is only the centroid that has this special property. Since we are generally not able to observe the complete object, the centroid of the reconstructed point cloud will not correspond to the actual centroid of the ball. Therefore, this particular, planarly moving point has to be found as part of the solution. We take the observable point cloud centroid as our initial guess for the planarly moving point and the PCA solution for m as the initial guess for the relative scale.

Starting from these data, a consecutive refinement step tries to simultaneously come up with an enhanced relative scale and a point with maximally planar motion for that scale. Hence, we have to solve for four unknowns, namely the values for the three coordinates of $\mathbf{p}_s^0$ and m. Since this step is a form of gradient descent, it allows to try different planarity criteria as we do not need to come up with a closed form solution. We used Levenberg-Marquardt (LM) to minimize the ratio of the third to the second eigenvalue of the covariance matrix in order to find the point that yields the maximal planarity. However other criteria are also available, such as the determinant of the covariance matrix as used before, where the volume spanned by the principal components is minimized, or the ratio of the third eigen-value to the product of the first two which can be considered as the ratio of a point's distance to a planar surface and the area of that surface.

The experimental results with the chosen method, described in the next paragraph, are satisfactory. Moreover, the potential of the approach exceeds the cases of motions that are obviously planar (for a human) like that of the rolling ball. The object motion might be constrained in more intricate ways. Consider a scenario where an object is rigidly attached to one end of a stick and the stick's other end is attached to a fixed point. If the stick rotates around that fixed point in a general manner, the object points would sweep spherical paths. Now consider the case where that other end also moves on a plane with the stick + object still rotating. The object moves in quite a complex way in

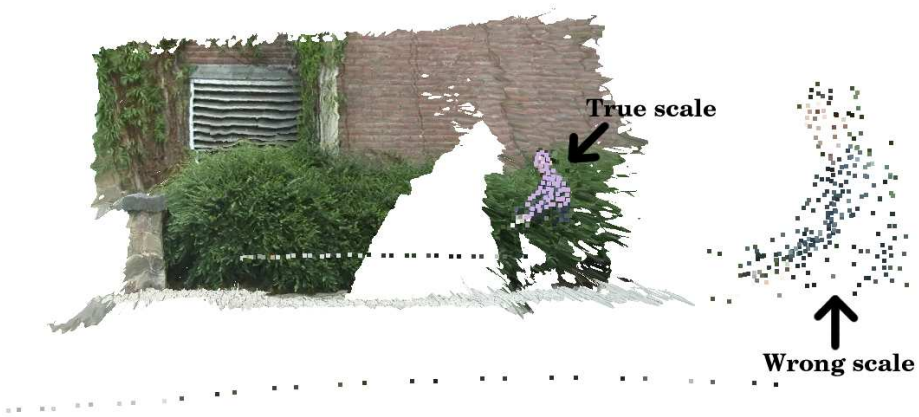
the world coordinate system, but the above type of approach would still come up with the solution as being the one with a non-accidental property. The practical exploration of such cases is out of the scope of this work.

2.3.2 Experimental results

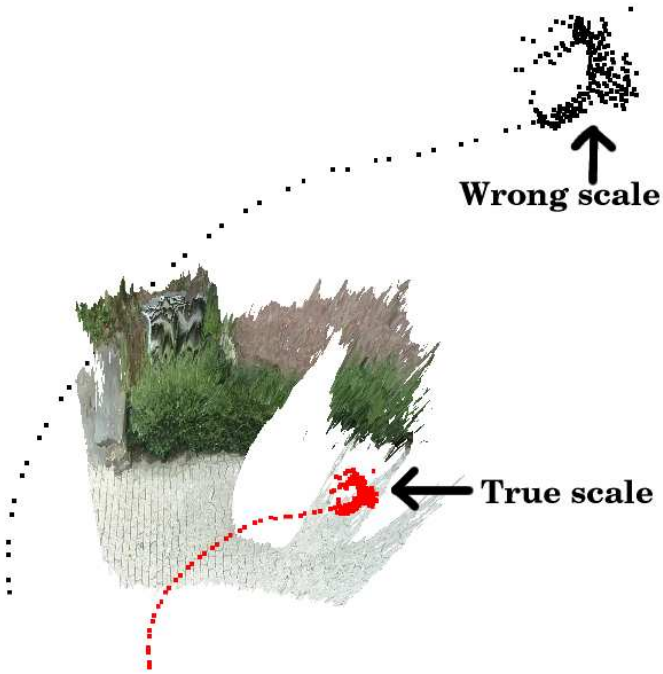
This algorithm was tested on several real image sequences. Our first experiment demonstrates a case where the majority of reconstructed points on the object undergoes planar motion. Consider the biker in Fig. 2.11. He is riding his bike on a plane along a curved path. We segmented the sequence by hand and ran our SfM algorithm separately on the biker and the background. This resulted in trajectories and 3D point clouds for the biker and the background, with a relative scale ambiguity. Assuming that the planarity of the biker motion would indicate the correct reconstruction, our algorithm gave a very realistic scale for the biker. In Fig. 2.12 the reconstruction of the scene is rendered from two different viewpoints. Two trajectories of the one-parameter family are shown, with a reconstructed point cloud of the biker at the starting position illustrating their respective scales. Figure 2.12 (a) corresponds to a viewing direction that is more or less parallel to the ground plane. It shows that the correct trajectory - which is the shortest of the two - is quite planar indeed. Figure 2.12 (b) is a top view. It shows that the longest trajectory is absolutely unrealistic, as it would catapult the biker behind the walls and bushes in Fig. 2.11. The biker himself also appears much too large, turning him into a giant. Fig. 2.13 demonstrates the non-planarity measure of the trajectory of the foreground object's centroid with respect to different scale values. The measure used here is the ratio of the third eigen value to the second. The minimum is out-spoken.

In our second experiment, we chose a harder problem where points extracted on the object surface are not following a planar trajectory themselves. As can be seen in Fig. 2.14 a ball is rolling on a ground plane. The overall motion of the ball may be planar but that of its individual points is not. Only the center of the ball is moving planarly. As previously mentioned, we take the centroid of the reconstructed ball points as an initial estimate of this center but it is not perfect since one side of the ball is not reconstructed. The solution found that way is then refined both in terms of the centroid position and the scale. The camera is moving in 3D with a general motion. Figure 2.15 shows the results in a similar way as Fig. 2.12. Two possible trajectories of the ball are shown. As can be seen from Fig. 2.15(b), the trajectory coming from our solution is close to planar and parallel to the ground plane. This is good evidence that our solution converged to the true centroid of the ball. The other trajectory is not planar at all and the ball is found hovering in the air. The algorithm converges to a reasonable solution even if we take any point in the vicinity of the ball as the initial 3D point for the LM iteration, but we suggest to use the centroid of the observable points for that purpose. Fig. 2.16 demonstrates the non-planarity measure of the trajectory of the foreground object's centroid with respect to different scale values. The measure used is again the ratio of

Figure 2.11: *Four frames from the 38 frame biker sequence.*



(a)



((b)

Figure 2.12: *Two reconstructed bike trajectories from two different angles(a,b). The long track shows the bike's trajectory at a wrong scale and the short track shows the bike's trajectory after we have solved for the relative scale based on planarity. The point clouds at the beginning of the trajectories represent the 3D reconstruction of some points on the biker.*

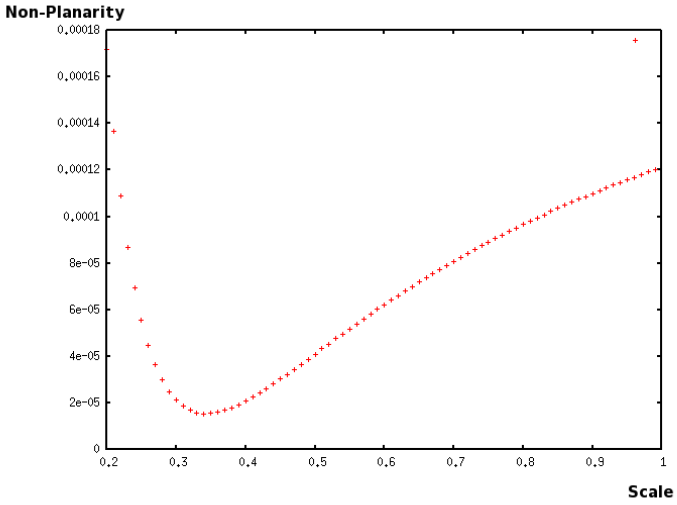


Figure 2.13: *The non-planarity measure (see the text) for the biker sequence at different scales. The minimum is out-spoken.*

the third eigen value to the second. The minimum is distinguishable, although the local neighbourhood is a bit flat.

2.3.3 The heading constraint

Many types of moving objects, such as humans, cars, bikes etc. have a natural frontal side and therefore a natural heading direction. Hence, these heading directions or vectors are usually parallel to the tangent of the object trajectory. If not, the objects would undergo strange motions like cars going into a skid. This does not usually happen. The mathematical equation describing this 'heading constraint' is:

$$l^{ij}\mathbf{R}_o^{ij}\mathbf{v}_o^i = \mathbf{v}_o^j \tag{2.28}$$

where $\mathbf{R}_o^{ij}$ is the rotation of the *object* from frame i to frame j. l^{ij} is a scale factor due to acceleration. $\mathbf{v}_o^i$ is the tangent to the object's trajectory at frame i which can be approximated by:

$$\mathbf{v}_o^i = \mathbf{g}_o^{i+1} - \mathbf{g}_o^{i-1} \tag{2.29}$$

where $\mathbf{g}_o^i$ is the position of the centroid of the object at i^{th} frame. This is a valid approximation since we generally use video sequences with relatively high

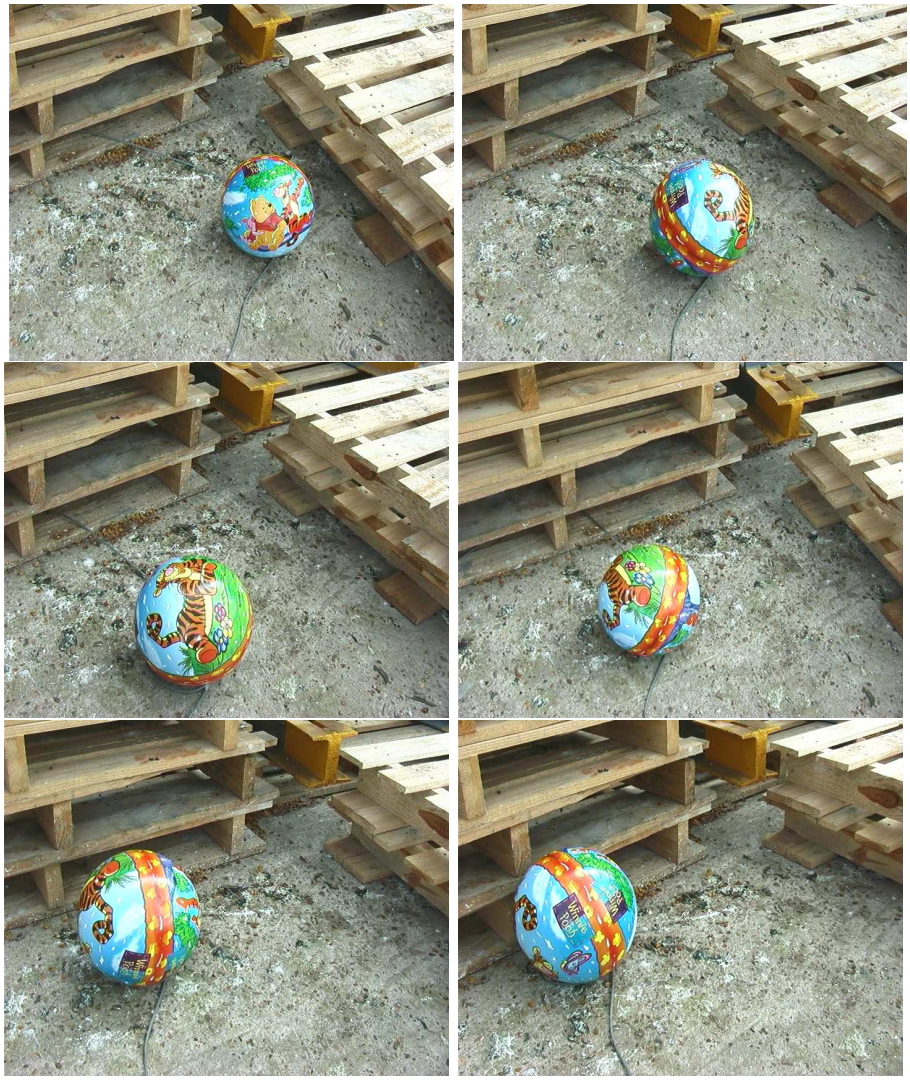
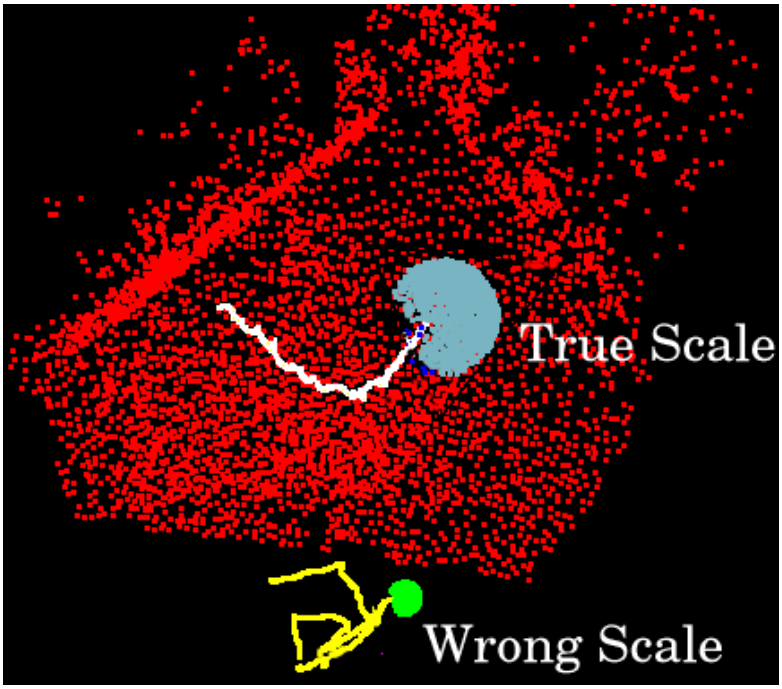
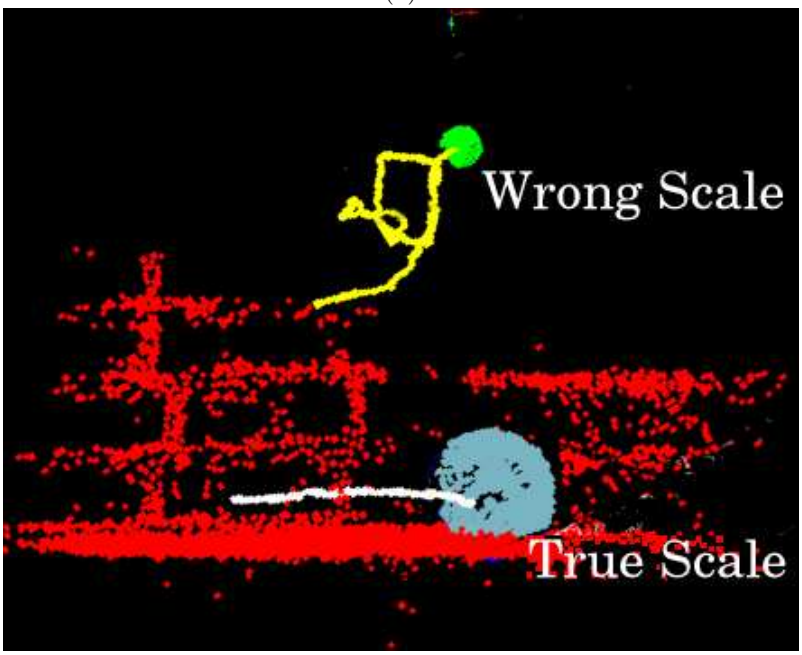


Figure 2.14: *6 frames from the 400 frame ball sequence.*



(a)



(b)

Figure 2.15: *The reconstructed scene of the ball sequence where both the erroneously scaled ball reconstruction and the correctly scaled one are shown with their trajectories.*

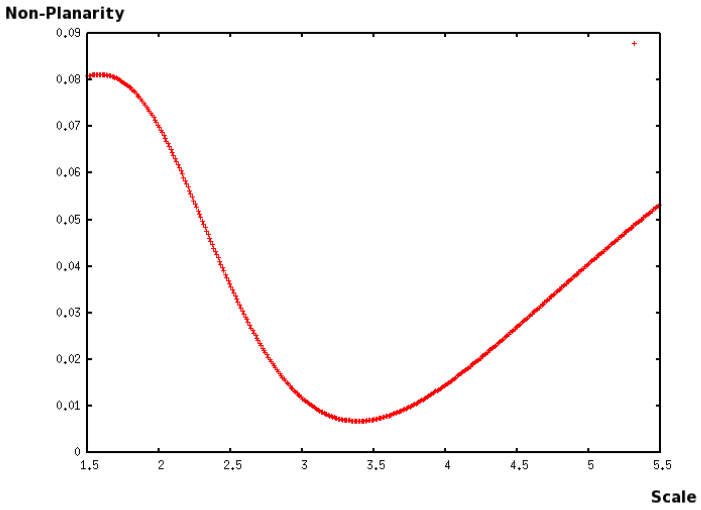


Figure 2.16: *The non-planarity measure (see the text) for the ball sequence at different scales. The minimum is distinguishable, though in a bit flat neighbourhood.*

frame rates. A similar expression is also used for the approximation of the camera velocities.

Eq. (2.6) states that the object trajectory will contain components from the camera translation for the wrong relative scales. Just as they may destroy planarity, they would also tend to lead to the violation of the heading constraint. However, also here, there are degenerate cases where this constraint can not help us. In such cases, Eq. (2.28) will hold for every relative scale. To analyze such cases, let us apply equations (2.29) and (2.28) together with a relative scale s. By changing all subscripts $_o$ to $_{os}$, the following equation results if the heading constraint is to hold for other, incorrect scales s:

$$k\mathbf{R}_o^{ij}(\mathbf{g}_{os}^{i+1} - \mathbf{g}_{os}^{i-1}) = \mathbf{g}_{os}^{j+1} - \mathbf{g}_{os}^{j-1} \tag{2.30}$$

where k is a scale factor. Then we introduce Eq. (2.10) to come up with the following equation:

$$k\mathbf{p} = \mathbf{q} \tag{2.31}$$

where

$$\begin{aligned} \mathbf{p} &= \mathbf{R}_o^{ij}(s(\mathbf{g}_o^{i+1} - \mathbf{g}_o^{i-1}) + (1-s)(\mathbf{t}_c^{i+1} - \mathbf{t}_c^{i-1})) \\ \mathbf{q} &= s(\mathbf{g}_o^{j+1} - \mathbf{g}_o^{j-1}) + (1-s)(\mathbf{t}_c^{j+1} - \mathbf{t}_c^{j-1}) \end{aligned}$$

By introducing $\mathbf{v}_o^i$, $\mathbf{v}_c^i$, $\mathbf{v}_o^j$, $\mathbf{v}_c^j$, Eq. (2.31) turns into :

$$k(s\mathbf{R}_o^{ij}\mathbf{v}_o^i + (1-s)\mathbf{R}_o^{ij}\mathbf{v}_c^i) = s\mathbf{v}_o^j + (1-s)\mathbf{v}_c^j \tag{2.32}$$

If the above equation holds for values of s other than 1, one has a degenerate case. If we insert Eq. (2.28) into the above equation and leave the term $\mathbf{R}_o^{ij}\mathbf{v}_o^i$ on the left side, we come up with

$$\frac{s}{1-s}(k-l^{ij})\mathbf{R}_o^{ij}\mathbf{v}_o^i = \mathbf{v}_c^j - k\mathbf{R}_o^{ij}\mathbf{v}_c^i \tag{2.33}$$

For a degenerate case to occur, the above equation must be solvable for k at every scale s and every frame pair. For different values of s and k, the left hand side spans a line passing through the origin with the direction $\mathbf{R}_o^{ij}\mathbf{v}_o^i$. The right hand side is an equation of a general 2D line for different values of k ($\mathbf{v}_c^j$ is a point on the line and $\mathbf{R}_o^{ij}\mathbf{v}_c^i$ is the direction of the line). Those two lines must be the same in order to solve for k for every s which results in a constraint that $\mathbf{v}_c^j$ is a constant multiple of $\mathbf{R}_o^{ij}\mathbf{v}_o^i$. Using our basic Equation (2.28), we infer that $\mathbf{v}_c^j$ must be in the direction of $\mathbf{v}_o^j$ for a degenerate case to occur. Fortunately, this is really hard to find in real life except for some simple motion cases. One such example is the case where both the camera and the object move on a line.

Let us return to the actual use of the constraint. Given two frames, finding the relative scale amounts to solving a polynomial equation which is formulated next. Merging Eq. (4.2) and Eq. (2.29) yields:

$$\mathbf{v}_o^i = m\mathbf{v}_{os}^i + (1-m)\mathbf{v}_c^i \tag{2.34}$$

The expression we want to maximize is coming from the heading constraint in Eq. (2.28). For the corresponding parallelism of velocity vectors to hold, we can maximize the cosine between the vectors:

$$\mathbf{a}^t\mathbf{b} = \cos(\mathbf{R}_o^{ij}\mathbf{v}_o^i, \mathbf{v}_o^j) \tag{2.35}$$

where

$$\mathbf{a} = \frac{\mathbf{R}_o^{ij}(m\mathbf{v}_{os}^i + (1-m)\mathbf{v}_c^i)}{\sqrt{(m\mathbf{v}_{os}^i + (1-m)\mathbf{v}_c^i)^T(m\mathbf{v}_{os}^i + (1-m)\mathbf{v}_c^i)}} \tag{2.36}$$

$$\mathbf{b} = \frac{m\mathbf{v}_{os}^j + (1-m\mathbf{v}_c^j)}{\sqrt{(m\mathbf{v}_{os}^j + (1-m)\mathbf{v}_c^j)^T(m\mathbf{v}_{os}^j + (1-m)\mathbf{v}_c^j)}} \tag{2.37}$$

This is the scalar product of two *normalized* vectors and it has the form of a rational polynomial. One can maximize the square of the cosine expression in Eq.(2.35) in case of an image sequence where the object suddenly decides to go backwards somewhere in the sequence. We discarded such rare cases to simplify the solution.

Solving for m with different frames i and j results in different m's. One reason is the fact that an object may not always follow its heading perfectly.

For example a person may twist his torso for a few frames. Such cases should be treated as outliers and we can use Eq. (2.35) in a RANSAC [HZ00] scheme for a robust estimation of m. Therefore, several random choices of i and j were made, and the m with maximal support was chosen i.e.depending on how many other i, j pairs have a super-threshold value for that m according to Eq. (2.35) .

Another problem we saw in our experiments is that 'objects' like humans obey the heading constraint globally but not instantaneously, e.g. during a single step. The center of gravity of the torso oscillates between left and right at this level of granularity. To avoid that, while calculating velocities, especially for human gait, we suggest to use the formula

$$\mathbf{v}^i = \mathbf{t}^{i+n} - \mathbf{t}^{i-n} \tag{2.38}$$

where n depends on the speed of the person and the sampling rate of video. We can estimate a good value for n during RANSAC random sampling, as an additional parameter to be estimated.

At the end, an additional refinement step is included on the m value supplied by RANSAC. The selection of the optimal m is based on the values for which the inliers minimize an error functional of the form

$$\sum^{\#inliers} angle^2(\mathbf{R}_o^{ij}\mathbf{v}_o^i, \mathbf{v}_o^j) \tag{2.39}$$

around the initial value of m. The reason for using angles at this stage rather than cosines as in Eq. (2.35) is the fact that angles are geometrically more meaningful.

To show the usefulness of our algorithm, we applied the technique to the video of the walking person which was previously shown in Fig. 2.8 while discussing the independence criterion. This is a sequence where all three constraints we have discussed so far apply, including the heading constraint. The computed relative scale is quite close to the one that results from the independence and the planarity constraint so it is not useful to show the reconstruction results as they are similar to the ones shown in Fig. 2.9. In that sequence the parameter n proved to be quite useful, supporting our aforementioned observation on human motion.

2.4 A discussion on choosing the correct criterion.

As stated earlier, one can come up with many types of motion simplicity constraints so a natural question that arises is how to select the correct criterion in an experiment. This has been an open issue so far but few suggestions can still be given.

Sometimes it will be possible to give that information as high-level input to the algorithm, since the data-set generally comes from a certain context where

the existing motion constraints can be deduced. A typical traffic sequence exhibits both planar motion and heading constraints. For an action movie on the other hand, the objects can move quite freely and the independence criterion would be appropriate.

In cases where such high level information is not available or when there are different objects in the scene that follow quite different motion constraints, reasoning about the type of constraint is still possible. A theoretically sound way is to follow a Bayesian approach and choose the motion constraint which is the most probable one. One simple method would be as follows. For each motion constraint hypothesis, a probability distribution with a single random variable s can be created by assuming the prior probabilities on the possible motion constraints and different values of s are equal. s can be discretized in a plausible range (this range can be deduced from generic but not always accurate independence constraint) and a probability value for each s can be computed from the data error terms. The pdf creation is completed after normalizing the values to make the sums equal to 1. The hypothesis which gives the highest peak can be selected. Another interesting approach would be to compute a weighted sum for the s values if more than one motion constraint gives high peaks. This is possible as the moving objects can follow multiple motion models simultaneously.

However the concept of choosing the most probable motion constraint can become quite involved when there are multiple objects moving in the scene. It is quite typical for objects to follow similar motion constraints, but the opposite is also not rare. Then a joint probability function needs to be derived as the probabilities are not independent any more. A further complication is that, the objects may not only exhibit the same motion constraint, but also the same instantiation of that constraint, i.e. the objects generally move on a certain plane not on any plane, or car heading directions coincide with each other as that direction is highly related to the road's shape. Those are the questions that still need to be investigated.

2.5 Concluding remarks

Reconstructing scenes containing independently moving objects remains a big challenge. In this chapter we proposed two rather generic criteria that can be used to solve the relative scale problem that exists between the independently moving parts of the scene. Both criteria follow from the fact that the true object trajectory tends to be less coupled to the camera motion than a trajectory at a false scale. On the one hand, the *independence constraint* directly goes for the statistical minimization of the correlation between object and camera motion. On the other hand, the *non-accidentalness constraint* assumes that solutions with certain regularities stand a higher chance to correspond to the real trajectory, as the influence of the camera motion tends to destroy these regularities at false scales. In the latter approach, it can happen that only

a single point in space exhibits these particular regularities. Therefore, it is useful that the optimization process simultaneously looks for such point and the true scale of its trajectory. We introduced two such motion constraints which proved to be effective. One of them exploits the fact that many common objects move planarly. A relative scale value which results in the most planar trajectory is selected. We also introduced the heading constraint, which selects scales on the basis of relation between object and trajectory orientation, rather than trajectory shape.

However there still are some open points to be investigated in-depth. For example, the degenerate cases have to be mapped out in more detail and there are more non-accidental properties to explore. Yet, another example is when the object trajectory fits the shape of the background at a particular scale, like having the car move over the terrain (road) rather than in plain air at that very scale. These also are properties that are destroyed under the wrong scaling. Techniques need to be investigated which take into account motion constraints in a multi-object setting where the motion constraints are inter-dependent.

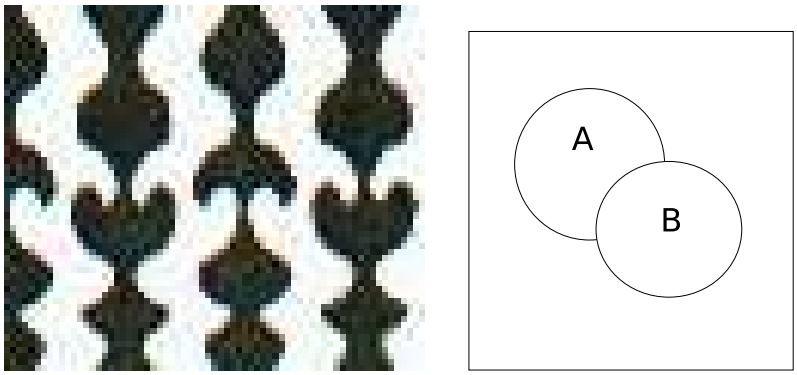
Chapter 3

Background Identification in Dynamic Scenes

3.1 Introduction

For many applications which process real life video data, background-foreground identification is a first vital step. Scene reconstruction, augmented reality, ego motion estimation, etc. are cases in point. Nevertheless given a video, principled ways to identify that part of the scene which is background have been few and far between. Here we make an attempt towards such solutions. In particular, given images of a scene taken with a moving camera and where independently moving parts have already been segmented (as stated in the previous chapter, several motion segmentation algorithms exist [VSMS02b, CK95, WS01b, SS06, GQZ05, Tor98], including the one that is presented later in this dissertation), we propose techniques to identify the background among these segments.

For the type of data we consider, the background is also moving strongly in the video images. In such cases, the background is often identified on the basis of 2D image related features. Examples are relative size (the background corresponds to the largest segment), spread of texture (the background has the highest 2D variance [VSMS02a] or has the highest number of feature points), visibility (the background feature points get swept away as the camera keeps track of the foreground or covers the biggest percentage of the image borders), convexity or symmetry (the foreground looks convex or symmetric) etc. Such approaches are analyzed in Psychology, e.g. [Rub21, RNS96] and Computer Vision, e.g. [SL95, PGR99] under the name of *Figure-Ground problem*. Fig. 3.1 illustrates two such typical clues, symmetry and T-junctions. However, all such clues may easily break down. To give examples, the moving objects can almost fill the screen, can move behind the static scene (like being partially hidden by a low wall) or can cover an entire image border. It must be noted



(a) (b)

Figure 3.1: *Two traditional figure-ground segmentation cues. (a) Symmetry: The black segment is perceived as foreground as it is symmetric (b) T-Junctions: Foreground objects occlude the edges of background object which create T-Junctions.*

that the figure/ground problem involves a depth ordering of the objects in the image. However, in our case the objects are not segmented according to their depth-ordering but according to their 3D rigid motion which is a problem for traditional cues. As we will demonstrate, if 3D analysis of these video shots is possible, it can offer more powerful solutions. The solutions that are proposed in this chapter are based on the motion constraints approach that was presented in chapter 2. In that chapter, it was noted that there is a relative scale ambiguity between the reconstructions of independently moving components of a dynamic scene. It was shown that for relative scale values other than the actual one, the object trajectories lose some of the properties that are quite common in real-life objects.

However in the methods described so far, the background had to be identified beforehand. If not, this adds an additional challenge. This chapter is an attempt to lift that limitation and the proposed techniques are based on the fact that, the aforementioned motion characteristics of an object are lost not only in the case of wrong relative scales but also in the case of a wrong background identification. Just to give a basic intuition of the concept, consider Kopernik's revolutionary discovery on the solar system. He noticed that if the sun is taken as the center of the Solar system, the motions of the planets are explained in a much simpler and more coherent way. Indeed until his time, the earth was considered to be the center of the universe and the motion of every other celestial object was computed relative to the earth. This resulted in complicated trajectories and equally complicated explanations, like the planets following various levels of epicyles (see Fig 3.2) Our approach works in the same spirit. Given the structure and motion information of each segmented

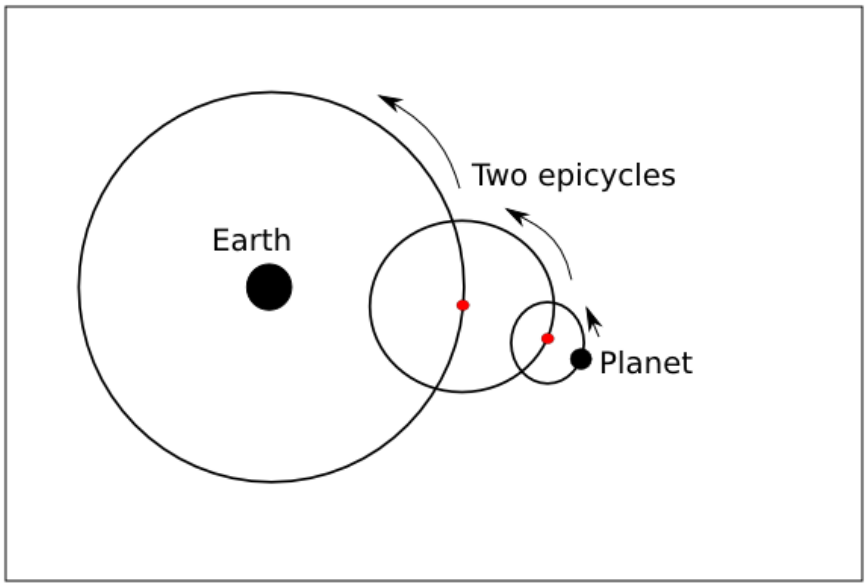


Figure 3.2: *A depiction of earth-centered universe model. Two epicycles are necessary to explain the complex relative motion of the planets.*

object, we look for the one (the background) which explains the overall scene dynamics in the simplest way when taken as the absolute reference. We show that the principles that we have introduced earlier to determine the relative scales can now also be used to identify the background. The findings of this study have also been published [O05].

In the remaining part of the chapter, first the motion constraints for solving the relative scale ambiguity are put in a new light of background identification. The viability of these ideas is corroborated through experiments with real image sequences. The chapter is concluded with the summary of the main ideas that are presented and a discussion of the open issues.

3.2 Background Detection with Motion Constraints

The basic observation of the previous chapter is Eq.(2.6) which states that the reconstructed object trajectory $\mathbf{t}_{os}$ is a mixture of the original object trajectory $\mathbf{t}_o$ and the camera trajectory $\mathbf{t}_c$. As to the independence criterion, we try to find the relative scale $m = 1/s$ which makes the resulting object trajectory statistically the most independent of the camera's trajectory. As to the

non-accidentalness criterion, we exploit the fact that the additive components from the camera trajectory at the wrong relative scales would cause the object motion to lose special properties which many typical moving objects in real life possess. As examples. we have proposed heading constraint and the planarity constraint.

In this section, after a brief reminder of the above constraints, we show that such properties are not only lost when a wrong relative scale is chosen but also when a wrong scene element is used as the 'background'.

3.2.1 The Independence Constraint

If we assume that the true object and camera motion are not linearly dependent (in the statistical sense), a linear dependence will only appear for the wrong relative scales. This is evident from Eq. (2.6). In addition, it will also show up if we identify the background object erroneously. To give an intuitive feeling, consider a scenario where a camera is moving slowly on a linear path and an object is moving randomly in front of the camera. The camera path and the object path would look quite dissimilar. However, if we consider the moving object as the static background, the actual background would look as if it moves randomly and the camera path would also have this motion in addition to its own linear path. Hence, a linear dependence pops up between the camera path and the background path (relative to the actual moving object). To state it more formally, let us write the camera motion and the background motion matrices relative to the moving object. The relative motion of the background is the inverse of the object motion:

$$\mathbf{T}_{bo} = \begin{bmatrix} \mathbf{R}_o^T & -\mathbf{R}_o^T\mathbf{t}_o \\ \mathbf{0} & 1 \end{bmatrix} \tag{3.1}$$

and the camera motion relative to the moving object can be derived as:

$$\mathbf{T}_{co} = \begin{bmatrix} \mathbf{R}_x & \mathbf{t}_x \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_o^T\mathbf{R}_c & \mathbf{R}_o^T\mathbf{t}_c - \mathbf{R}_o^T\mathbf{t}_o \\ \mathbf{0} & 1 \end{bmatrix} \tag{3.2}$$

If we look at the translation components of $\mathbf{T}_{bo}$ and $\mathbf{T}_{co}$, it is seen that they become linearly dependent due to the additive components $-\mathbf{R}_o^T\mathbf{t}_o$ even if $\mathbf{t}_c$ and $\mathbf{t}_o$ are independent. There are many ways to exploit this dependence and one easy technique that we also used in the previous chapter is to measure the classical correlation between the translation components. However, as mentioned before, we should also consider the relative scale problem between the translational components due to the nature of SfM. The solution we propose is, for all possible hypotheses for the background object, solve the relative scale between that proposed background and other objects with the minimum correlation technique and choose the one as the background which gives the lowest overall correlation.

Fig. 3.3 and Fig. 3.4 show two of the four input sequences on which we tested our algorithm. The other two input sequences, robot (Fig. 2.2) and

Figure 3.3: *Image samples from the market sequence which contains one moving object except from the background*

the human-walking (Fig. 2.8) were already shown in the previous chapter. In Fig. 3.3 a person is pushing a shopping trolley. During the course of the video clip, the person moves rigidly with the trolley so both are reconstructed as a single object (foreground). The camera's motion with respect to the static background is mostly backwards although with arbitrary movements. This motion enables us to reconstruct the market itself. In Fig. 3.4, a person is walking while holding a box rigidly. The upper torso, the head and the box are reconstructed as single object (foreground). The legs are not included since they do not move rigidly. As mentioned before, the images are segmented beforehand with a semi-manual technique and an iterative perspective SfM algorithm is run over those individual segments. Table 3.1 shows the minimum correlation computed for both correct and incorrect background selections. Noticeably, correct background selection always results in minimal correlation.

3.2.2 The Heading Constraint

As stated in the previous chapter, many types of moving objects, such as humans, cars, bikes etc. have a natural frontal side and therefore heading

Figure 3.4: *Image samples from the box sequence which contains one moving object except from the background*

Table 3.1: *Minimum correlation values for different test sets and different background selections.*

Test Set	Actual background	Wrong Background
Market	0.430	0.634
Human-box	0.195	0.564
Robot-ball	0.068	0.507
Human-walking	0.361	1.312

direction. Hence, these heading directions or vectors are usually parallel to the tangent of the object trajectory, see Eq. (2.28). This equation prescribes that the trajectory tangent vector remains tangent when rigidly attached to the object. It describes a coupling between the object translation and the rotation. We expect such a coupling to vanish in the case of a wrong relative scale due to added camera components. Unlike for the independence criterion, theoretically two frames can be enough to solve for the relative scale and we proposed a RANSAC [FB81] scheme to estimate it robustly.

One interesting and subtle phenomenon related to the heading constraint is the fact that it is not symmetrically defined. To be more precise, if an object is moving according to the heading constraint, it does not necessarily mean that the background's relative motion with respect to the object also complies with the heading constraint. On the contrary, it is very likely that it will not comply as the simple example in Fig. 3.5 demonstrates. Assume that a car is heading north along a wall. For an observer in the car, the wall is seen as if it is heading south. Then assume that the car turns right and heads east. Then the observer in the car would see the wall heading west. Although the car's heading direction stayed the same for an observer standing on the ground, the wall's heading direction changed suddenly with 90 degrees for the observer in the car.

We use the asymmetrical nature of the heading constraint for the detection of the background. However, as in the case of the independence criterion, the uncertainty about the relative scale between the different reconstructions of the objects in the scene should be taken into account. During the random sampling phase of our heading constraint based technique (see the previous chapter), typically several hundreds of candidate relative scale values are computed and put in a histogram. The peak value in the histogram is taken as an initial hypothesis which is later fed to an optimization function. When the peak of the histogram gets higher and the variance gets lower, we can infer that the calculated relative scale is supported by many frames so the object is complying with the heading constraint very well. If the peak is low and the variance is high, we can conclude that the object is complying poorly. Since the heading constraint is asymmetric, we expect the histogram for the wrong selection of the background to have a much lower peak and a higher variance.

Fig. 3.6 depicts the normalized histograms related to the correct and the incorrect selection of the background for the market sequence shown in Fig. 3.3. The first histogram (peak=10, sample variance=0.126) corresponds to the correct background selection and the second one (peak=7, sample variance=0.186) corresponds to the incorrect background selection. It is visible that the peak values in the first histogram are higher compared to the second one and the second histogram has higher variance. These observations tally with our predictions.

Fig. 3.7 depicts the histograms in the same way but now for the the human walking sequence which is also shown in Fig. 2.8. The first histogram (peak=35, sample variance=0.015) corresponds to the correct background selection and the second one (peak=20, sample variance=0.074) corresponds to the incorrect background selection. The higher peak and lower variance for the actual background is also observed here.

3.2.3 The Planarity Constraint

Referring to the previous chapter, when we observe the moving objects in our daily life, thanks to the ground plane and gravity, the dominance of planar

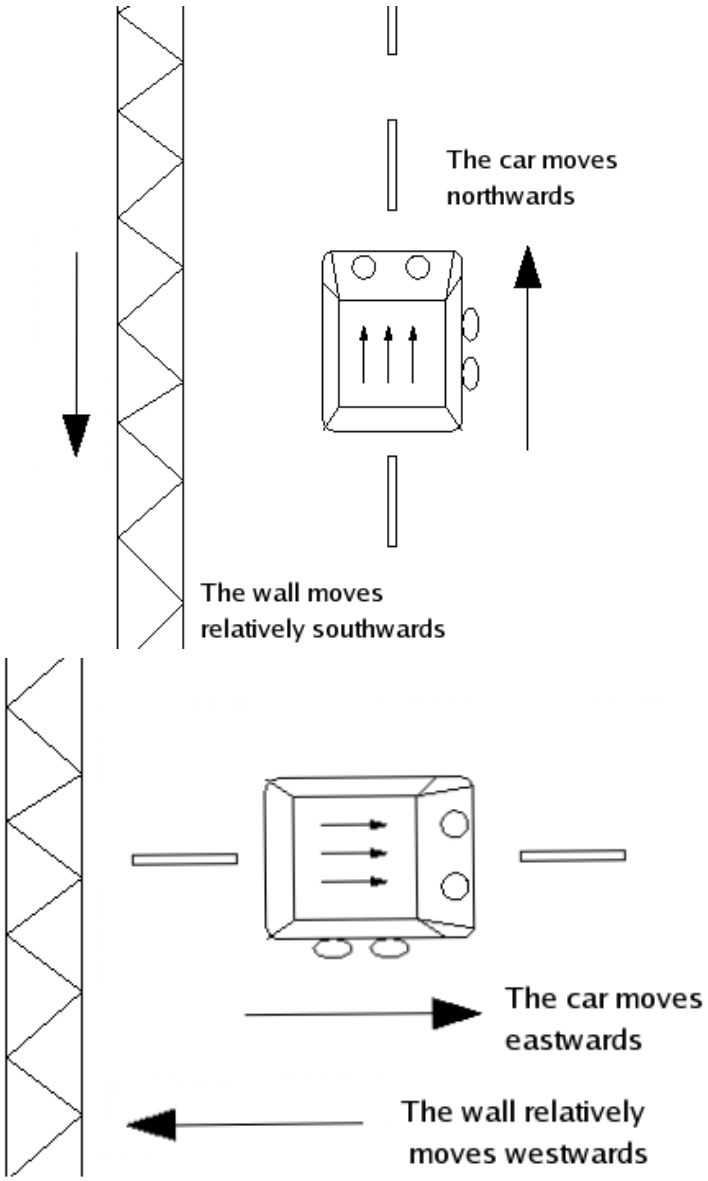


Figure 3.5: *An illustration of asymmetry of the heading constraint. In the top picture, a car is moving northwards along a wall. So for an observer in the car the wall moves southwards.In the lower picture, the car turns right and moves eastwards. For an observer in the car, the wall moves westwards. This means the wall has changed its heading direction with 90 degrees relative to the observer in the car.*

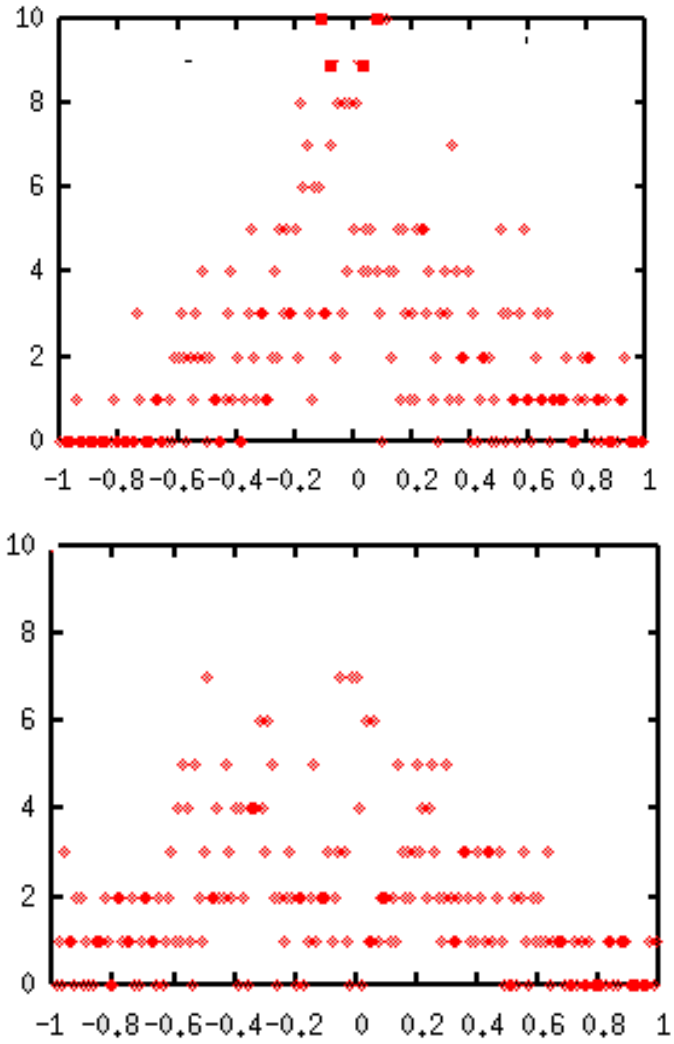


Figure 3.6: *The normalized heading constraint histograms for the market sequence. The first picture (peak=10, sample variance=0.126) corresponds to the correct background selection and the second one (peak=7, sample variance=0.186) corresponds to the wrong selection. Notice the high peak low variance nature of the correct selection.*

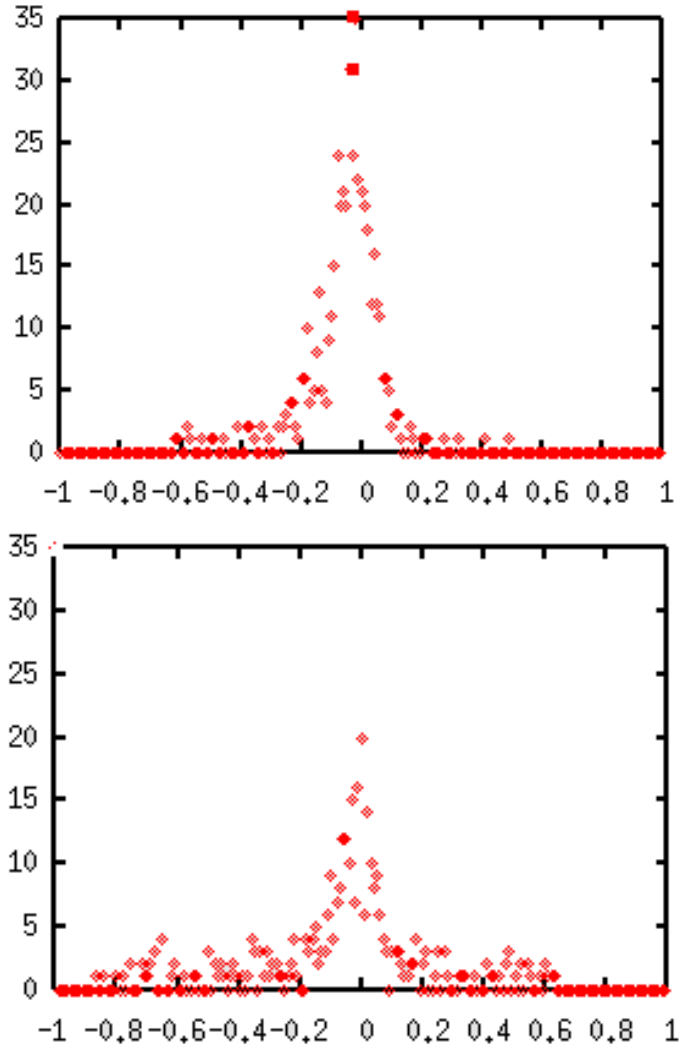


Figure 3.7: *The normalized heading constraint histograms for the human-walking sequence. The first one (peak=35, sample variance=0.015) corresponds to correct background selection and the second one (peak=20, sample variance=0.074) is for the wrong selection as in the case of Fig. 3.6. Notice the high peak low variance nature of the correct selection similar to Fig. 3.6.*

Figure 3.8: *Four images from an image set where two people are moving on non-parallel planes and the camera is moving arbitrarily.*

motion is unquestionable. This makes it a very valuable source of information for resolving the relative scale ambiguity. Since the camera translation components are added to the object trajectory for the wrong relative scales, planarity of the object trajectory is lost in general so the detection of a planar motion among different relative scales is taken as an indication for the true relative scale. As a technique to exploit that constraint, we suggested a PCA based initialization followed by non-linear iterative maximization of a planarity criterion in the previous chapter.

Although this constraint is very useful in finding the relative scales of real life objects, it has two serious deficiencies which limit its applicability for the detection of the background. First of all, if an object is moving planarly according to the background, the relative motion of the background relative to the moving object is also planar, so it is symmetrical unlike the heading constraint. This renders it impossible to detect the background if there is only one moving object. However if there are at least two planarly moving objects, it is still possible to detect the background because in general the moving objects do not 'see' each other move planarly, even if they actually move on planes. This can be proven with a degenerate case analysis similar to the one given

in the previous chapter. Consider a case where there are two independently moving objects, A and B, on two different planes. For simplicity, assume that the objects only translate, i.e.the objects move in the planes without changing their orientation relative to the world coordinate system. The trajectory of a point on the object A relative to the static background can be written as:

$$\mathbf{p}_A^i = \mathbf{p}_A^0 + \alpha_A^i \mathbf{q}_1 + \beta_A^i \mathbf{q}_2 \tag{3.3}$$

similar to notation in Eq. (2.19). $\mathbf{p}_A^0$ and $\mathbf{p}_A^i$ are the point positions at frame 0 and i respectively, α_A^i, β_A^i are real numbers with $\alpha_A^0 = \beta_A^0 = 0$ and $\mathbf{q}_1$ and $\mathbf{q}_2$ are orthonormal 3-vectors spanning the plane. The motion of a point on the object B relative to the static background can be written as:

$$\mathbf{p}_B^i = \mathbf{p}_B^0 + \alpha_B^i \mathbf{q}_1 + \beta_B^i \mathbf{q}_3 \tag{3.4}$$

with similar notations as for A, where $\mathbf{q}_2$ is replaced with $\mathbf{q}_3$. Without losing generality, $\mathbf{q}_1$ is assumed to be in the direction of the intersection of the two planes (if two planes are not intersecting, any direction parallel to the planes is satisfactory) in the formulations. $\mathbf{q}_2$ is orthogonal to $\mathbf{q}_1$ and it is parallel to the first plane, and $\mathbf{q}_3$ is also orthogonal to $\mathbf{q}_1$ but it is parallel to the second plane.

The trajectory of point A relative to point B is just the difference of them which is:

$$\mathbf{p}_r^i = \mathbf{p}_A^0 - \mathbf{p}_B^0 + (\alpha_A^i - \alpha_B^i)\mathbf{q}_1 + \beta_A^i \mathbf{q}_2 - \beta_B^i \mathbf{q}_3 \tag{3.5}$$

or in matrix form

$$\mathbf{P}_r = \begin{bmatrix} 1 \\ 1 \\ \dots \\ \dots \\ 1 \end{bmatrix} (\mathbf{p}_A^0 - \mathbf{p}_B^0)^T + \mathbf{N} * \mathbf{Q} \tag{3.6}$$

$$\text{with } \mathbf{N} = \begin{bmatrix} \boldsymbol{\alpha}_A - \boldsymbol{\alpha}_B & \boldsymbol{\beta}_A & \boldsymbol{\beta}_B \end{bmatrix} \tag{3.7}$$

$$\text{and } \mathbf{Q} = \begin{bmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \mathbf{q}_3^T \end{bmatrix} \tag{3.8}$$

where n×3 matrix $\mathbf{P}_r$ holds the relative point positions, $\boldsymbol{\alpha}_{A,B}$, $\boldsymbol{\beta}_{A,B}$ are n-vectors whose elements are $\alpha_{A,B}^i$, $\beta_{A,B}^i$. The structure of $\mathbf{N}^*\mathbf{Q}$ determines the dimensionality of the resultant trajectory and in general this is a 3D dimensional path as it is linear combination of 3 linearly independent vectors with different multipliers. However, similar to the degenerate case analysis of planar motion in the previous chapter, various degenerate cases can be figured out by inspecting the rank of the matrix $\mathbf{N}^*\mathbf{Q}$.

The first easy to see degeneracy is the case where the planes are parallel to each other. In that situation $\mathbf{q}_3$ is parallel to $\mathbf{q}_2$ so the rank of $\mathbf{N}^*\mathbf{Q}$ is

2. Indeed, if the objects move on parallel planes, they will still see each other moving in such planes. This is a bit discouraging since many objects in real life move parallel to the ground plane. However it should be noted that even if there is only one object which is moving planarly or linearly but not parallel to the motion planes of all other objects, it would help us to identify the background since its planarity would only be supported by the actual background.

The second type of degeneracy is caused by rank deficiency of $\mathbf{N}$. Consider the case where $\boldsymbol{\alpha}_B$ and $\boldsymbol{\beta}_B$ are linear combinations of $\boldsymbol{\alpha}_A$ and $\boldsymbol{\beta}_A$. In such a case the rank of $\mathbf{N}$ is definitely 2 which results in a planar path for the resultant relative motion. This case corresponds to the situation where the trajectory of A is an affine transform of the trajectory B. Indeed, aforementioned linear dependency can be written as:

$$\begin{bmatrix} \boldsymbol{\alpha}_B^T \\ \boldsymbol{\beta}_B^T \end{bmatrix} = \boldsymbol{F} \begin{bmatrix} \boldsymbol{\alpha}_A^T \\ \boldsymbol{\beta}_A^T \end{bmatrix} \tag{3.9}$$

where $\boldsymbol{F}$ is 2×2 affine transform matrix. By introducing the base vectors $\mathbf{q}_j$ to the above equation, a similar relation can also be written for the actual planar paths in 3D.

The third type of degeneracy also stems from the rank deficiency of $\boldsymbol{N}$. Consider the case where $\boldsymbol{\beta}_B$ is a constant multiple of $\boldsymbol{\beta}_A$ then the rank of $\mathbf{N}$ is 2 and the trajectory is planar again. This is a limited form of the affine transformation relationship. Although the trajectories can be completely unrelated in the direction of $\boldsymbol{q}_1$ the other degrees of freedom are a constant multiple of each other, which can be considered as the affine transformation of only one coordinate.

However other than the above degenerate cases, we expect $\mathbf{N} * \mathbf{Q}$ to be rank-3 in general which means that the relative paths are non-planar. An hypothetical example illustrating this phenomenon is given in Fig. 3.9. The first picture shows a scene where two objects and the camera perform random planar motion (can be assumed pure translational motion for simplicity) on different planes. The paths are depicted relative to the static background. However, the second picture depicts the motion trajectories relative to the object A. In this frame of reference, the background is moving planarly with respect to A since it is just symmetric with respect to the original motion of A. However, the other object motions lose their simplicity and their apparent motion now spans a 3D volume.

In terms of implementation of those ideas, we use the ratio of the third eigen-value to the second eigen-value of the scatter matrix of the trajectory positions as non-planarity measure. Then, given any hypothesis for the background, we compute the scale of the other objects relative to it by minimizing this non-planarity criterion over the relative scales and take the maximum of these planarity deviations as an indicator of the deficiency of the proposed background. After repeating the same procedure for all possible backgrounds, we choose the object which gives the minimal maximum deviation.

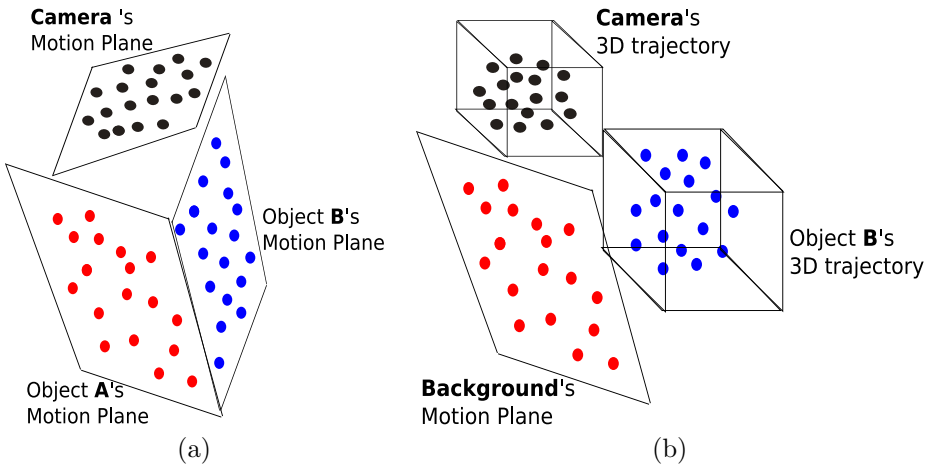


Figure 3.9: *Depiction of the use of the planarity constraint for background detection purpose. The camera and two other object move arbitrarily on three different planes. (a) The trajectories relative to the static background (b) The trajectories relative to the object A. Object B and the camera trajectories lose their planarity.*

To test the proposed algorithm, we used the image sequence in Fig. 3.8 where there are two people moving planarly. Note that the path that is followed by person A on the stairs is approximately a plane which is not parallel to the ground plane. The other person moves arbitrarily on the ground plane. Table 3.2 shows non-planarity values of each object (column-wise) relative to a background hypothesis (row-wise). It should also be noted that the path of the centroid of the associated SfM 3D point cloud of each segment is taken as the representative of that object's path.

As is clear from the table, it is the actual background (the last row) which describes the other object motions in the most planar way, in keeping with our predictions. The path of person B (person walking on the ground plane) gives a very low non-planarity value. But as could be expected, the non-planarity score for person A is higher, as his motion is slightly non-planar. This is quite understandable since a walk on stairs is not a perfectly planar motion. The last column illustrates the planarity of the actual background trajectory relative to the moving objects. In a hypothetical scenario where the objects are complying perfectly with planar motion, we would expect those values to be zero. Yet, the values significantly differ from zero. This somewhat unexpected result is easily explained by the effect of object rotations and increased noise in SfM computation for small foreground objects. As for person A, his axis of rotation is definitely not perpendicular to the plane of the stairs. This strongly violates the planar motion assumption. As a consequence, whereas the person's centroid

Table 3.2: *Non-planarity values of object paths for different background hypotheses. Each row corresponds to a single background hypothesis.*

	Person A	Person B	Background
Person A	–	0.022	0.05
Person B	0.04	–	0.031
Background	0.021	0.000543	–

may move more or less planarly relative to a static background, the background would not at all move planarly according to person A. This is responsible for the high non-planarity value in the first row and the last column. Interestingly, this phenomenon does more good than harm since it decreases the planarity support for the wrong background selections. The relatively low, but still substantial value in the second row and the last column is due to noise (here the rotations are compatible with the planarity assumption) on what ideally would have been planes of motion. It is known that SfM computation is valid only around the structure that is reconstructed. Hence, the effect is accentuated as we get further away from that reference.

3.3 Conclusion

For applications related to dynamic scene analysis, identification of the background among all parts of the scene (assuming motion segmentation is done) can be a vital phase, including our dynamic scene reconstruction system, and this phase is generally implemented with a simple heuristic on 2D image features. Unfortunately, the background is not always the occluded, the biggest, ... object. Such traditional figure-ground clues may not be applicable depending on the complexity of the scene and the existing motions. However, as we have shown in this chapter, 3D analysis of a scene would give a lot of information on the identity of the background where these simple approaches fail. The background is identified as the object which gives the simplest interpretation for the overall scene motion.

We proposed three techniques based on the independence criterion and the non-accidentalness principle, namely the correlation approach, the heading constraint and the planarity constraints. We demonstrated the applicability of those techniques with real life experiments. Each of those criteria has different weaknesses and strengths. The independence criterion is applicable to any scene, however requires many frames to be statistically valid and needs variation in the motion parameters. The heading constraint is rather practical, since it requires a small number of frames and many real world objects follow non-holonomic motion. However it has certain degenerate cases, such as when all the objects follow linear paths in the same direction. The planarity criterion is attractive since the motions of many objects are constrained by planes.

However the approach requires at least two dynamic objects other than the background. Another downside is many objects often move on a single dominant plane which is a degenerate case.

One strong point in the overall approach is the fact that only a small subset of the moving objects is required to follow a motion constraint, rather than all of the objects. For example, consider the case where there are 10 moving objects. Let's say only one of them is moving with heading constraint and the rest demonstrates arbitrary motions. Choosing the right background will result in ten percent of the moving objects follow the heading constraint but choosing the wrong one result in zero percent. Hence, even only one object follows the criterion, it helps to identify the correct background. However, in practice more than one object would be necessary for a healthy estimation.

Although we conducted successful experiments, we are aware that there are still some unexplored phenomena. For example, considering the independence criterion, rotation is also a valuable source of independence information since not only the linear velocities but also the angular velocities are coupled for the wrong background selection. This work may also pave the way towards a wider rank constraint. The background tends to be the object which results in the smallest overall rank of the object motions in the scene. For example, linear paths would be observed as 3D paths if the background is chosen incorrectly. An optimal method which combines all the proposed methods should be investigated further. An interesting study would be whether human visual system is using such kind of motion simplicity assumptions to detect an object as the background.

Chapter 4

Space-Time-Scale Registration of Dynamic Scenes

4.1 Introduction

In the search for methods which will bring typical Structure from Motion techniques to bear on real world sequences with their many dynamic elements, we have introduced various solutions so far. Their common theme was the assumption that an object's motion follows a certain type of motion constraint, which can either be a statistical constraint (e.g. the foreground motion is independent of the camera motion) or more of a geometric nature (e.g. object follows a planar motion). Such constraints have been used for finding the relative scales and the detection of the correct background object so far. Although such an approach is quite practical, it is not always the case that the foreground object obeys such a motion constraint. Its motion can be quite irregular or dependent on the camera motion. Here we propose an alternative solution for such cases. It requires multiple cameras but it works with generic object motions and without any corresponding features between the video streams. Not only does it allow to determine the relative scales, also synchronizes the video and the 3D reconstruction extracted from each stream is spatially registered.

Obviously, this is far from the first work using multiple cameras in a SfM context. Here we only mention the most related work. In [DC99], the relative displacement between the cameras of a stereo rig (the views do not overlap) is computed using several motions of the rig. In terms of approach, that work shows some resemblance to the method presented here. In a similar vein, it uses pure motion information for two cameras to compute the relative displacement between them. However the method is developed for a stereo-rig where the cameras are rigidly attached to each other, the scene is basically static and the

motion parameters for each two time instants are computed separately which increases the number of unknown scale values drastically.

In [WZ02b], a self-calibration method for a moving rig is presented which also does not need any feature matches between the camera views. The rig itself does not need to be rigid, however some constraints on the camera orientations are still required, such as the cameras only rotate around a certain axis. Those constraints are not only used for self-calibration but also for time synchronization. In [WZ02a], a non-rigid scene is reconstructed with static orthographic stereo cameras. A restricted form of class-based approaches [BHB00] was used, where each 3D point in one camera can be written as a linear combination of the 3D points visible in the other camera. All of the above work has the common advantage of being correspondence-free, i.e. there are no stereo correspondences between the cameras.

Here, we use two (hand-held) cameras moving completely independently with respect to each other, still not assuming knowledge of any correspondences. The price to pay for this freedom is that at least one moving and rigid object ought to be observed by both cameras as the information from the background itself is not enough to solve the problem. The fact that the object should move the same way with respect to the background in both sequences is the trivial but key observation exploited by the algorithm. It can thereby fix the scales of the object and the background, bring their partial 3D reconstructions into registration, and even synchronize - i.e. temporally align - the two videos. The work that is presented here is originally published in [OCG06].

Video synchronization in combination with (partial) camera calibration has also been studied by several other researchers, and the exploitation of moving objects in particular as well. For example, Caspi et al. [CI02] use point trajectories to find a suitable transformation to spatio-temporally align image sequences. It is based on the fact that the feature trajectories are much stronger cues than single feature points to register the viewpoints. A transformation for registration both in space-time is searched. The approach is applicable to any kind of dynamic scenes but requires that the relative transformation between the cameras is static and there are common scene points. Sinha and Pollefeys [SP04, SPM04] also combine camera calibration with synchronization. They compute the camera calibration from image silhouettes to account for the time-shift between the video sequences. These papers still require the visibility of the same points to different cameras at the same time.

Caspi et al. [CI01] could lift this restriction by using moving but rigidly attached cameras with either the same optical center or observing a distant scene. The views were then aligned in space (through a homography) and in time.

4.2 Problem

In this chapter we consider two hand-held cameras which move independently with respect to each other. Furthermore, we consider a single object moving independently against a static background (but the method could also work with multiple moving objects with some adaptation). The cameras may view the moving object from totally different directions, so it is well possible that there are no common feature points between the video sequences both for the background and the foreground. However it is required that the cameras see the same rigidly moving object, though possibly different parts thereof.

Similar to the previous chapters, in order to reconstruct such a scene the first step is to segment the foreground object from the background for which several solutions are available including the system that we developed in the framework of this dissertation. This then allows a typical uncalibrated SfM algorithm [HZ00] to be applied to the object and background segments in each of the videos (however it is also possible to perform segmentation and reconstruction together, which has many advantages, as will be explained in the next chapter). This results in four 3D point clouds and four sets of camera matrices (trajectories relative to the capturing camera). These cannot be readily integrated however, not even for the object and background data derived from the same camera as pointed out in previous chapters. Several parameters need to be determined first.

Three of those parameters again come from the fact that uncalibrated SfM is defined only up to a scale factor. Here it is shown that the use of two cameras allows the objects to move arbitrarily. We will have to determine the 3D similarity transformation between the reconstructed backgrounds in the two videos, as well as the relative scales of the foreground in each video with respect to its background. This will require the synchronization of the two videos. So in total we have to solve for nine parameters. The wrong choice for these parameters will result in a different object motion for each video stream which actually must be identical. Hence, our goal is to search for those parameters which will make the object motions for both sequences identical. Stated differently, we look for the parameters that make the overall object motion the most rigid: if the object motions as seen from both cameras are identical, the combined foreground point clouds must move rigidly.

In our analysis, we will first assumed that the cameras are synchronized. Then, in a second step, we will lift this restriction and solve for full spatio-temporal alignment.

4.3 Notation and Basic Formulation

The two cameras are arbitrarily labeled as the first and the second camera. Similar to the notation in chapter 2, applying SfM to the first sequence yields the following object transformation matrices with respect to the static back-

ground:

$$\mathbf{M}^i = \mathbf{T}^i\mathbf{M} = \begin{bmatrix} \mathbf{R}_o^i & \mathbf{t}_o^i \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{M} \tag{4.1}$$

which describe the motion of a 3D homogeneous point $\mathbf{M}$, which is a fixed point in the object coordinate system. $\mathbf{M}^i$ is the position of point $\mathbf{M}$ at frame index i in the world coordinate system. Typically the pose of the first frame is chosen as the world coordinate system, which is also the case here. To remind the reader of the notation, $\mathbf{R}_o^i$ is a 3×3 rotation matrix and $\mathbf{t}_o^i$ is a 3×1 translation vector.

$\mathbf{T}^i$ is computed by multiplying the inverse of the related background motion matrix with the relative motion matrix of the foreground, both of which are direct outputs of the SfM algorithm. However, due to unknown relative scales, there exists a one-parameter family of solutions for these object transformation matrices. Restating Eq. (2.6) by variable substitution $s = 1/m$ as before results in:

$$\mathbf{t}_o^i = m\left(\mathbf{t}_{of}^i - \mathbf{t}_c^i\right) + \mathbf{t}_c^i \tag{4.2}$$

where $\mathbf{t}_{of}^i$ is a particular solution for the object translation and $\mathbf{t}_c^i$ is the position of the camera optical center in the world coordinate system, which are both returned by SfM. The one-parameter family is described by scale factor m. To give an intuitive explanation we can interpret Eq (4.2) as a set of 3D lines which pass through the optical center at each frame index i. Consequently, every point on these lines project to the same location in the same image given any value m.

The world coordinate system will be different for both image sequences since the camera poses for the first frame will differ. However, a similarity transformation exists which aligns the world coordinate systems of both sequences:

$$\mathbf{M}^i = \mathbf{X}\mathbf{M}'^i \text{ with } \mathbf{X} = \begin{bmatrix} k\mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{4.3}$$

where $\mathbf{M}'^i$ is a point in the second image sequence which corresponds to point $\mathbf{M}^i$ in the first sequence. Here, it is important to stress that $\mathbf{X}$ is a transformation between the 3D reconstruction reference frames, and not between the moving cameras. Therefore, $\mathbf{X}$ is constant throughout the sequence but the transformation between the camera local coordinate systems is allowed to change freely.

The aforementioned transformations are all illustrated in Fig. 4.1 in which the superscript $'$ accompanies the symbols related to the second sequence.

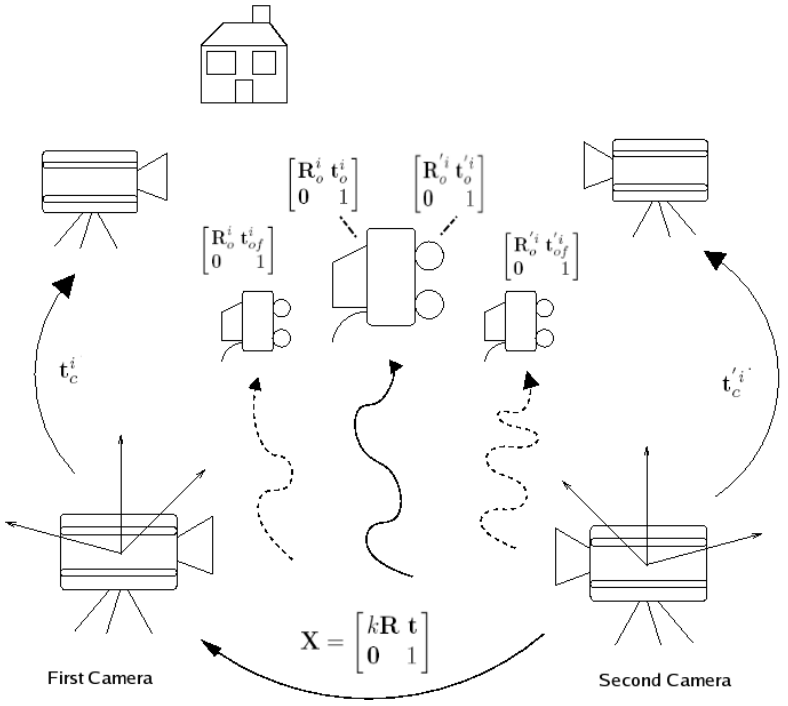


Figure 4.1: A depiction of the transformations. Due to the relative scale ambiguity, different cameras see arbitrarily scaled objects and ambiguous object translations.

4.4 Solution

4.4.1 Spatial Solution

Combining Eq. (4.1) and Eq. (4.3) for both sequences, we arrive at:

$$\begin{bmatrix} \mathbf{R}_o^i & \mathbf{t}_o^i \\ \mathbf{0} & 1 \end{bmatrix} = \mathbf{X} \begin{bmatrix} \mathbf{R}_o^{'i} & \mathbf{t}_o^{'i} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{X}^{-1} \tag{4.4}$$

which is a different form of hand-eye calibration problem. This is a typical problem in robotics and the aim is to compute the unknown transformation between an actuator and a sensor that is rigidly attached to it. Fig. 4.2 illustrates the problem. The robot arm moves with a known transformation $\mathbf{A}$ and the camera external transformation $\mathbf{C}$ can be computed from the calibration pattern. In order to compute the transformation between the final pose of the camera and the initial position of the actuator, we can follow two paths, either first C then T or first T then A which must give the same result. This can be written as $\mathbf{TC} = \mathbf{AT}$ or $\mathbf{C} = \mathbf{T}^{-1}\mathbf{AT}$. Various techniques have been proposed to solve such type of equations, such as the pioneering work

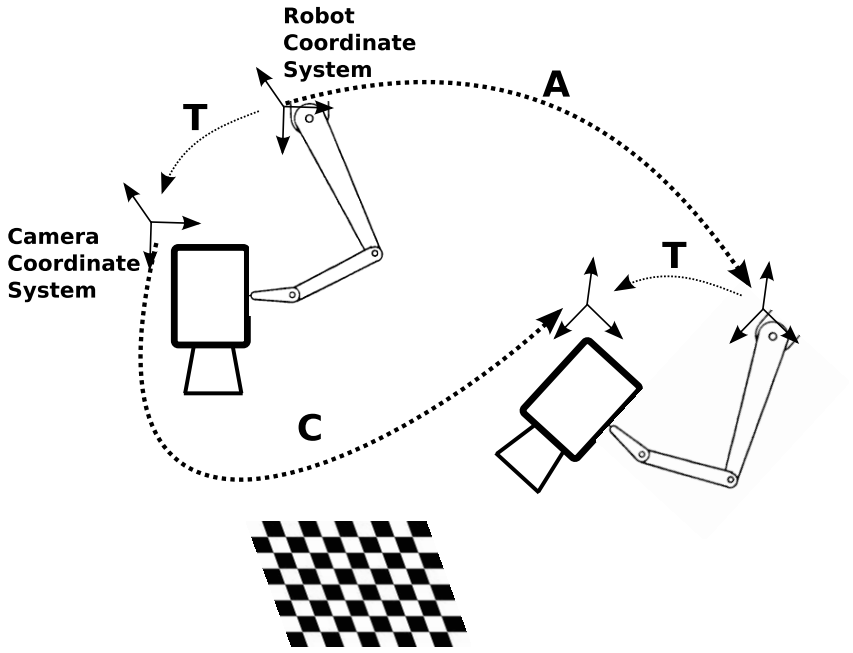


Figure 4.2: An illustration of the hand-eye problem. See the text for details.

of Shiu and Ahmad[SA89], Tsai and Lenz[Tsa87], the simultaneous non-linear solution by Horaud and Dornaika[HD95] and the dual-quaternion approach by Daniilidis[Dan99].

A common technique, such as [SA89, Tsa87] in hand-eye calibration is to solve for the rotation part first:

$$\mathbf{R}_o = \mathbf{R}\mathbf{R}_o^{'}\mathbf{R}^T \tag{4.5}$$

and subsequently solve for the translation part:

$$\mathbf{t}_o = -\mathbf{R}_o\mathbf{t} + k\mathbf{R}\mathbf{t}_o^{'} + \mathbf{t} \tag{4.6}$$

where the frame indices have been dropped for ease of notation. We will follow the same path which is outlined in [HD95]. Eq. (4.5) can be transformed to:

$$\mathbf{R}_o\mathbf{R} = \mathbf{R}\mathbf{R}_o^{'} \tag{4.7}$$

As to the solution of the above equation, it is known that every rotation matrix has an axis which remains unaffected under that particular rotation. This is one of the eigen-vectors of the rotation matrix and has eigen-value one. Let $\mathbf{n}^{'}$ and $\mathbf{n}$ be those specific eigen-vectors of $\mathbf{R}_o^{'}$ and $\mathbf{R}_o$ respectively. If we multiply

both sides of Eq. (4.7) with $\mathbf{n}^{'}$ we have the equation:

$$\begin{aligned} \mathbf{R}_o\mathbf{R}\mathbf{n}^{'} &= \mathbf{R}\mathbf{R}_o^{'}\mathbf{n}^{'} && (4.8)\\ &= \mathbf{R}\mathbf{n}^{'} && (4.9)\end{aligned}$$

which means that $\mathbf{R}\mathbf{n}^{'}$ is the relevant eigen-vector of $\mathbf{R}_o$ which can be written as:

$$\mathbf{n} = \mathbf{R}\mathbf{n}^{'} \tag{4.10}$$

Also as an intuitive derivation, it is expected that after alignment of the world coordinate systems of both sequences the axes of the related rotations must be identical.

Many solutions have been proposed to solve the above equation, e.g. [FH86, SA89, Tsa87]. We chose the unit quaternion approach by Faugeras and Hebert [FH86] which is also detailed in Horaud and Dornaika [HD95]. Unit quaternions are 4-parameter imaginary representations of rotations in 3D. Since their length is one, their degree of freedom is three as expected from a rotation representation. The operation of a rotation matrix on other rotation matrices and 3D vectors can be easily represented as quaternion multiplications. The rotation of a 3D vector $\mathbf{n}$ with the rotation matrix $\mathbf{R}$ can be written as:

$$\mathbf{q} * \mathbf{n}_q^{'} * \overline{\mathbf{q}} = \mathbf{R}\mathbf{n}^{'} \tag{4.11}$$

where $\mathbf{q}$ is the quaternion representation of $\mathbf{R}$, $\overline{\mathbf{q}}$ is the conjugate of $\mathbf{q}$, $\mathbf{n}_q^{'}$ is the quaternion representation of the vector $\mathbf{n}^{'}$ and $*$ is quaternion multiplication.

As we have many frames and noisy data, it is practically impossible to find a perfect solution to Eq. (4.10) so we should minimize an error criterion. The one used here is the total 3D squared Euclidean distance between the corresponding rotation axes after the application of rotation $\mathbf{R}$ which can be written as:

$$E_1 = \sum_{i=1}^{\#frames} \left| \mathbf{n}_q^i - \mathbf{q} * \mathbf{n}_q^{'i} * \overline{\mathbf{q}} \right|^2 \tag{4.12}$$

Since quaternions are of unit length, the statement inside the summation can be written as :

$$\begin{aligned} \left| \mathbf{n}_q^i - \mathbf{q} * \mathbf{n}_q^{'i} * \overline{\mathbf{q}} \right|^2 &= \left| \mathbf{n}_q^i - \mathbf{q} * \mathbf{n}_q^{'i} * \overline{\mathbf{q}} \right|^2 |\mathbf{q}|^2 && (4.13)\\ &= \left| \mathbf{n}_q^i * \mathbf{q} - \mathbf{q} * \mathbf{n}_q^{'i} \right|^2 && (4.14)\\ &= \mathbf{q}^T\mathbf{N}^i\mathbf{q} && (4.15)\\ & && (4.16)\end{aligned}$$

where $\mathbf{N}^i$ is a 4×4 matrix whose elements are computed from $\mathbf{n}^i$ and $\mathbf{n}^{'i}$ [FH86, HD95]. In the end we have a minimization of the form :

$$E_1 = \mathbf{q}^T\mathbf{N}\mathbf{q} \tag{4.17}$$

where $\mathbf{N} = \sum_{i=1}^{\#frames} \mathbf{N}^i$. When we try to minimize Eq. (4.17) with the constraint that quaternions are of unit length, the quaternion turns out to be the eigen-vector of $\mathbf{N}$ corresponding to the minimum eigen-value.

Now that the rotation parameters are computed, we can proceed to solve Eq. (4.6) for the translation and the scale parameters. Inserting Eq.(4.2) for both sequences into Eq.(4.6) results in:

$$\mathbf{t}_c = (\mathbf{t}_c - \mathbf{t}_{of})\, m + (\mathbf{I} - \mathbf{R}_o)\, \mathbf{t} + \left(\mathbf{R}\mathbf{t}_c^{'}\right) k + \left(\mathbf{R}\mathbf{t}_{of}^{'} - \mathbf{R}\mathbf{t}_c^{'}\right) km^{'} \tag{4.18}$$

which is a linear equation in terms of $\mathbf{t}$, k,m and $km^{'}$. In a typical scenario, we would have redundant equations so a simple linear least squares scheme is applicable here.

Since the rotation is estimated separately from other parameters, it is desirable to minimize an error criterion which handles all parameters simultaneously. We must also note that our final aim is to come up with a solution where the foreground objects as reconstructed from both sequences move as rigidly as possible with respect to each other. However, a minimization in transformation space does not necessarily result in the best rigid motion for the foreground objects since it minimizes an algebraic error rather than a geometric one. A good way to express rigidity is by stating that distances between points remain the same. Therefore, the solution so far is used as an initialization of a non-linear iterative refinement technique like Levenberg-Marquardt with the following error criterion:

$$E_2 = \sum_{k=1}^{6} F\left(\mathbf{p}_k\right) \tag{4.19}$$

$$F\left(\mathbf{p}\right) = \sum_{i=1}^{\#frames} \left|\mathbf{T}^i\mathbf{p} - \mathbf{X}\mathbf{T}^{'i}\mathbf{X}^{-1}\mathbf{p}\right|^2 \tag{4.20}$$

where $\mathbf{T}^i$ and $\mathbf{T}^{'i}$ are euclidean transformation matrices describing the object motion in the i^{th} frame for the first and the second camera. F is an error measure between the paths of a 3D point when the motion matrices computed for the first and the second image sequence are applied separately and $\mathbf{p}_k$ is a specific point in the object coordinate system of the first camera. As to the choice for $\mathbf{p}_k$, we followed some guidelines. First of all, a 3D Euclidean transformation is defined by the motion of at least 3 non-collinear points, so the number of points must be more or equal to three and they must be non-collinear. Secondly as the SfM measurements are valid only around the reconstructed object, the points may not be far away from the 3D point cloud of the object but also should not be very close to each other in order not to degenerate to a single point. So in order to satisfy all these criteria, we decided to take the PCA transform of the point cloud and choose the end points of the computed axes (given by the singular values) which result in six points in total.

4.4.2 Spatio-Temporal Solution

So far we implicitly assumed that both video streams are synchronized in time. However, with hand-held cameras this is usually not the case. To overcome this difficulty, researchers proposed different techniques, e.g. [CI02, SP04, WZ02a], and the problem of time synchronization becomes more and more popular.

In our case, Eq.(4.5) and Eq.(4.6) give a geometric relationship between two frames and we would expect that these equations do not hold when two frames do not correspond to each other in time, just like any other geometric relationship like the fundamental matrix etc. So the technique we propose for time synchronization is to shift the video sequences with respect to each other within a reasonable range and compute the residual of the solution to Eq. (4.19). We expect that the correct time shift corresponds to the lowest residual. After a rough discrete shift value is found, the residual graph can be interpolated to search for the solution at sub-frame accuracy. To achieve this, a sub-frame time shift parameter λ is incorporated into Eq. (4.20) which results in:

$$F_{sub}\left(\mathbf{p}\right) = \sum_{i=1}^{\#frames} \left| \lambda \mathbf{T}^{i+shift}\mathbf{p} + \left(1-\lambda\right)\mathbf{T}^{i+1+shift}\mathbf{p} - \mathbf{X}\mathbf{T}'^{i}\mathbf{X}^{-1}\mathbf{p} \right|^2 \quad (4.21)$$

where λ is restricted to be between 0 and 1 and $shift$ is the rough discrete time-shift value. This equation basically introduces linear interpolation to the paths defined by the principal points.

4.5 Degeneracies

The proposed technique suggests to use pure motion information of the foreground objects to spatio-temporally align image sequences. Consequently, the algorithm is expected fail in the cases where the foreground motion is very simple, i.e.does not have enough distinctive properties. For example, as to the solution of the Eq. (4.5), it is known that the existence of at least two distinct rotation axes is necessary and as the number of available axes increase the solution becomes more stable. Consequently, a scene which consists of a single dynamic element that is moving planarly, would cause the initial rotation calculation fail. However the translation information, which is deliberately ignored during the initialization of the rotation parameters (for the sake of dealing with simple equations), could be useful here at the cost of developing a more complicated simultaneous solution for the rotation and translation equations.

We are also aware of the fact that, different types of simple ($\simeq$ degenerate) object motions would cause different type of ambiguities on the final registration parameters. For example, a single arbitrarily translating foreground object (= random motion, no rotation), would enable to solve for the rotation and scale parameters of the final registration but not the translation. As another example, a foreground object which undergoes a pure rotation around a

single point would cause the inter-camera scale resolution fail.

An indepth analysis of the degenerate motions and the resultant ambiguities in the estimated parameters is still an open-issue. However, we expect that the existence of multiple moving objects would significantly decrease such problems.

4.6 Experiments

We conducted two different experiments to demonstrate the effectiveness of the proposed technique. In the first experiment, a person is pushing a dolly on which a pile of boxes are placed. The person and the background are recorded by two freely moving hand-held cameras whose viewing angles are quite different so it is hard to find common features between the two image sequences. Some example frames from the first and the second camera can be seen in Fig. 4.3. The careful reader might notice that although the set-up is very wide-baseline, there are still some common feature points. However those points will only be used for *verification* of the computed registration parameters. Although our algorithm does not require their existence, it helps us to demonstrate that the algorithm works well.

The sequence is 180 frames long (image size is 720×576) and the dolly passes through different poses. Both sequences are segmented beforehand as foreground and background sequences, are reconstructed separately using SfM and subsequently fed to our algorithm. The time-shift between the sequences is approximately known to be 5 frames which is close to 5.13, the value computed by the algorithm.

Fig. 4.4 shows the background reconstructions from two different cameras which are registered together by the proposed method. It can be clearly seen that the corresponding ground planes and walls are aligned quite well. To give a different view of the result we manually chose three common features from the first sequence, computed their 3D positions and projected them in the second sequence using the registration parameters we computed. In Fig. 4.5, the black circles denote the actual position of the feature points and the white squares nearby depict the the reprojection of the corresponding 3D points of the second sequence after transfer to the second sequence. The average pixel error is 6 pixels. If we have a good registration, we also expect the foreground motions to be the same. So in order to test the latter, we chose a 3D point from the foreground object of the first sequence and computed its 3D path according to the motion parameters from the first sequence and also according to the *registered* motion parameters computed from the second sequence. Fig. 4.6 demonstrates such a registration for an arbitrary 3D point. The circles and the triangles correspond to point paths computed with object motions from the two different video streams. The error measure, which is the average distance between the corresponding point positions divided by the path length, is 0.8% which is quite low as expected. Fig. 4.7 shows the registration of the foreground

dolly and the boxes on it for a specific time instant from different view-points. The boxes and the dolly reconstructions are registered pretty well, although registration is not perfect.

As for the second experiment, we recorded a 330 frames-long (image size is 720×576) sequence where a person carries boxes on a staircase and is moving arbitrarily but rigidly. Fig. 4.8 show some example frames. The cameras are also moving freely and view the scene from quite different angles. We computed the reconstructions and registration parameters in the same way as in the previous experiment. Fig. 4.9 shows the registered background reconstructions. As can be seen, the ground plane, the stairs, the walls and the pillars are very well registered. Fig. 4.10 demonstrates the reprojection of some common feature points having an average pixel error of 15. Fig. 4.11 demonstrates the 3D point paths computed from the object motions from the two different image sequences. The error measure, which is the average distance between the corresponding point positions divided by the path length, is 0.4% which is quite low. Fig. 4.12 shows the registration of the foreground person and the box he carries for a specific time instant from different view points. The person and the box reconstructions are registered pretty well in terms of rotation and scale but the translation is a bit off in the direction of gravity (check the side view). However such a result is expected since only the relatively noisy foreground motion parameters are used to register both sequences. It is known that, SfM quality is proportional to the number of features and their spread in the images which lack in small foreground objects. However, the system counter balances this adversity by use of many frames. A more optimal solution would be to develop a bundle adjusment routine tailored to take into account the common foreground motion, thus resulting in an overall simultaneous optimization for every parameter.

4.7 Conclusion and Discussion

In this chapter, we presented a novel technique which finds the space-time-scale parameters between two reconstructions of a scene coming from two independently moving hand-held cameras. Rather than matching features like points, lines etc., it tries to find a consistent transformation which results in the most similar motion for the independently moving foreground object. As a consequence, the cameras are free to observe the scene from totally different angles with the restriction that at least one rigidly moving foreground object visible to both is required.

Although we presented our initial results here, there are still open questions and possible improvements. As an initial improvement, the basic approach can easily be extended to scenarios which contain more than two cameras and multiple rigidly moving foreground objects. Although we have not used common feature points we can find such features much more easily after an initial registration and use them as well in a global optimization. As an interesting fact,

Figure 4.3: Samples from the original image sequence. Each colun belongs to a separate camera, each row is related to a different time instant.

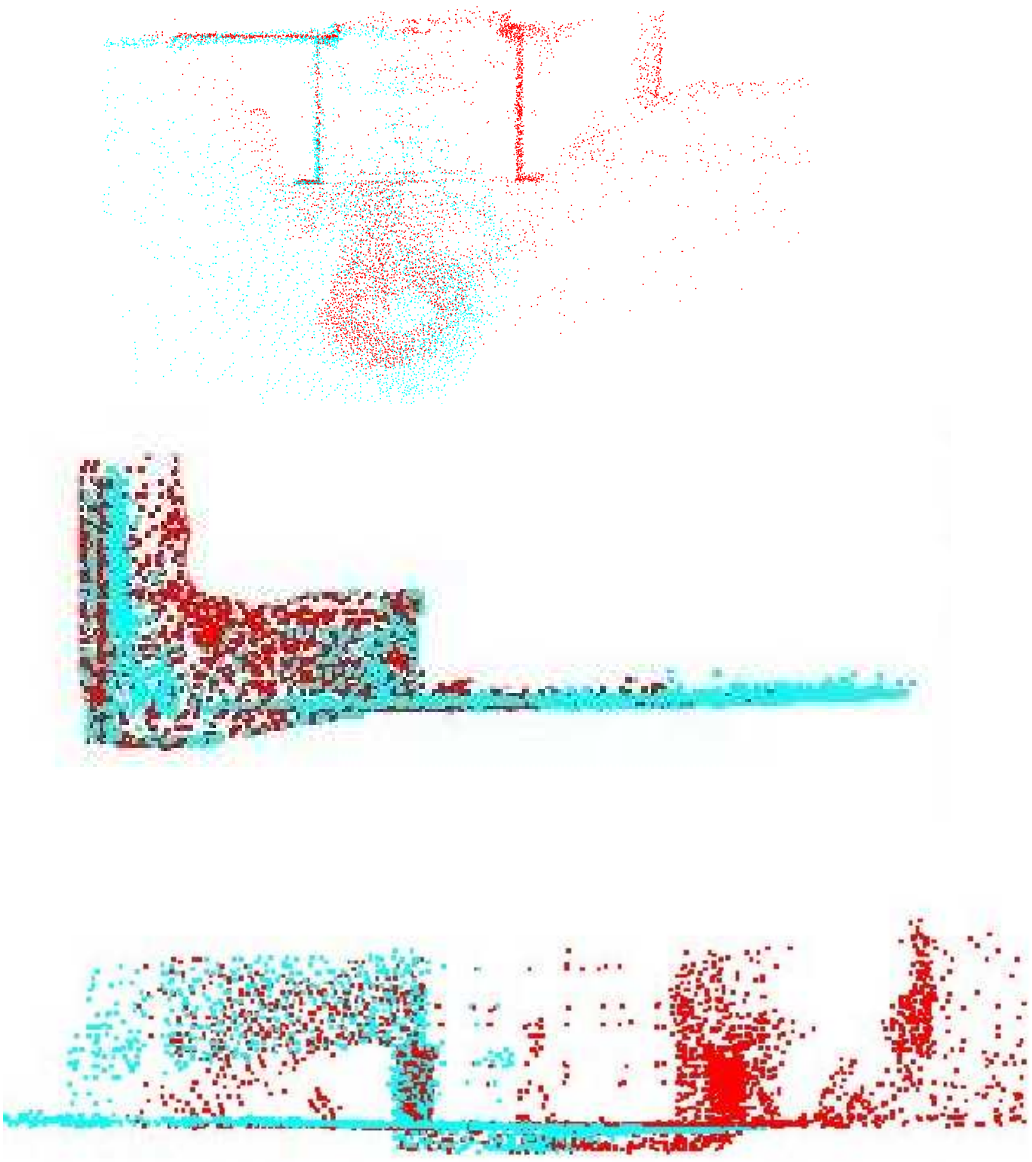


Figure 4.4: A top, a side and a front view of the background reconstruction. Notice how well the walls and the ground planes are registered.

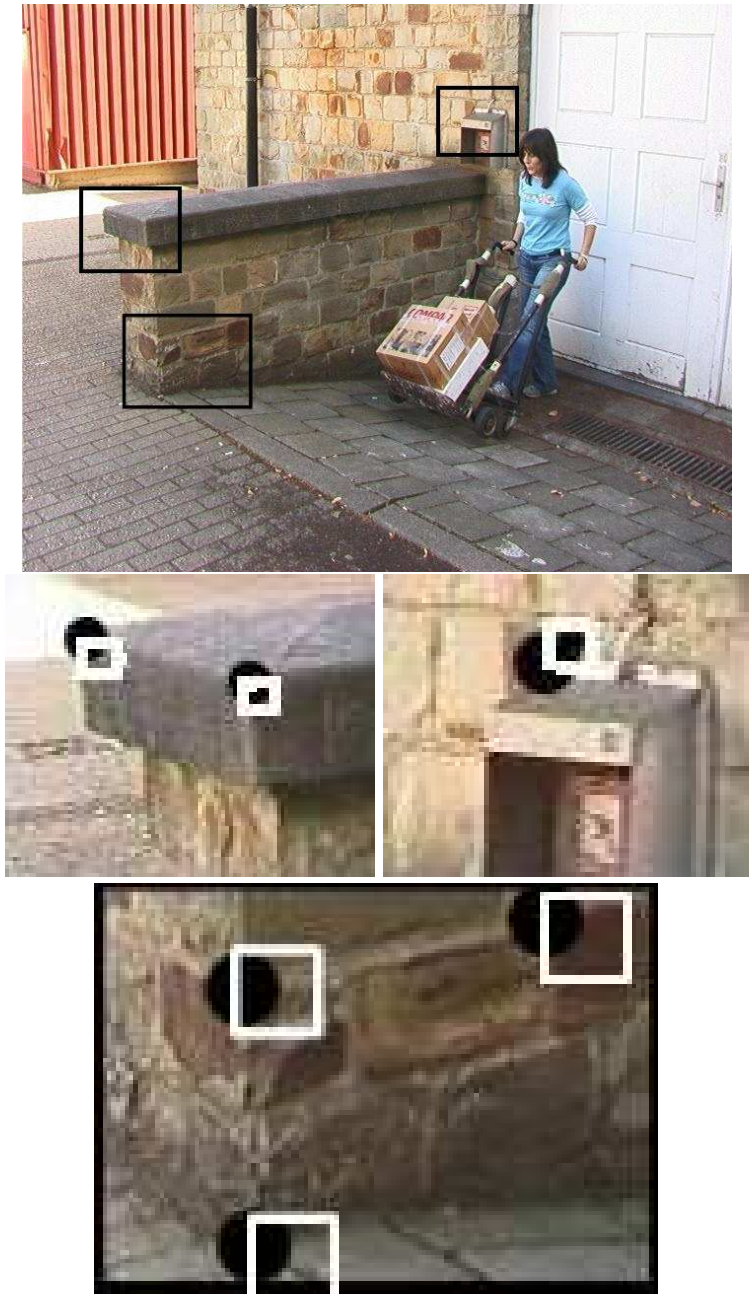


Figure 4.5: Manually tracked features from one image sequence are projected into the other image sequence. In the region of interest, the original features are depicted by black circles, whereas their reprojections are depicted by white squares.

Figure 4.6: Different views on the resulting path of the centroid of the foreground reconstruction in the first image sequence when displaced by object transformations coming from the first sequence (circles) and the second sequence (triangles) after registration. The resultant paths for other points on the object are quite similar, hence they are not shown.

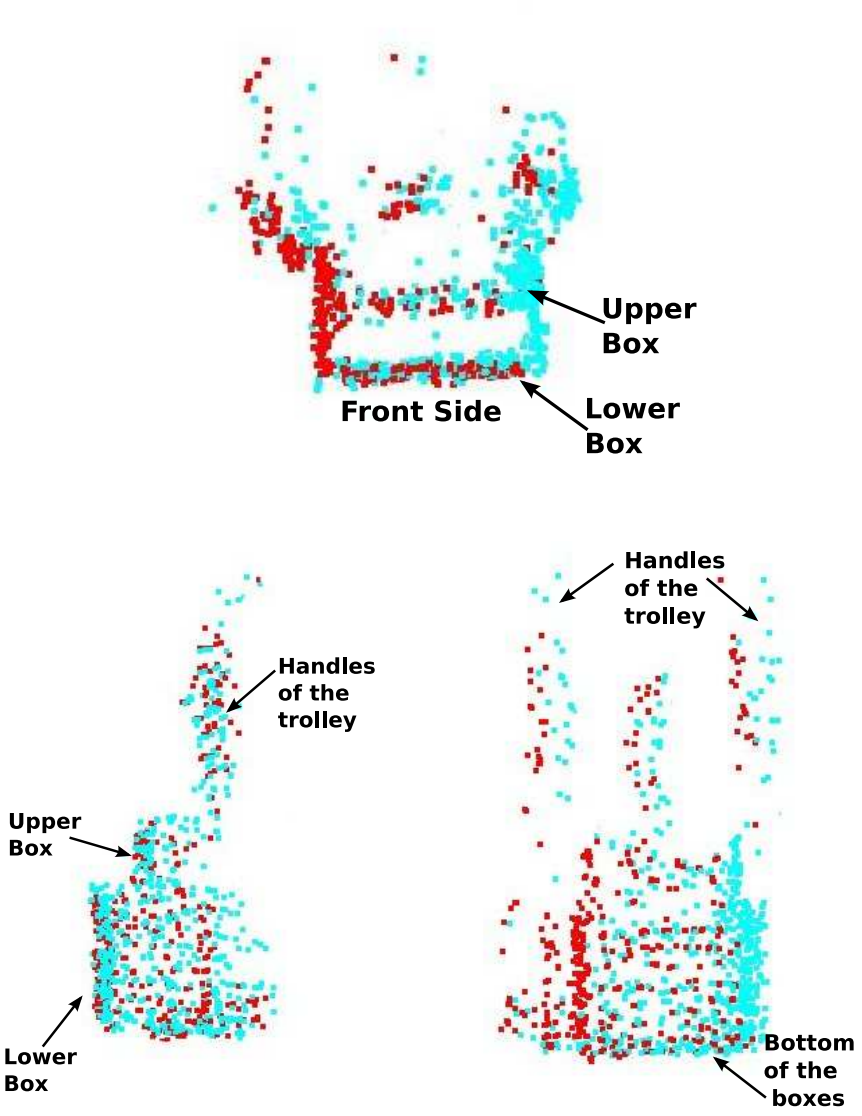


Figure 4.7: A more or less top, side and front view of the reconstructed dolly and the boxes on it. The sides of the boxes and the handles are registered well though not perfectly.

Figure 4.8: Samples from another image sequence. Each column belongs to a separate camera, each row is related to a different time instant.In the region of interest, the original features are depicted by black circles, whereas their reprojections are depicted by white squares.

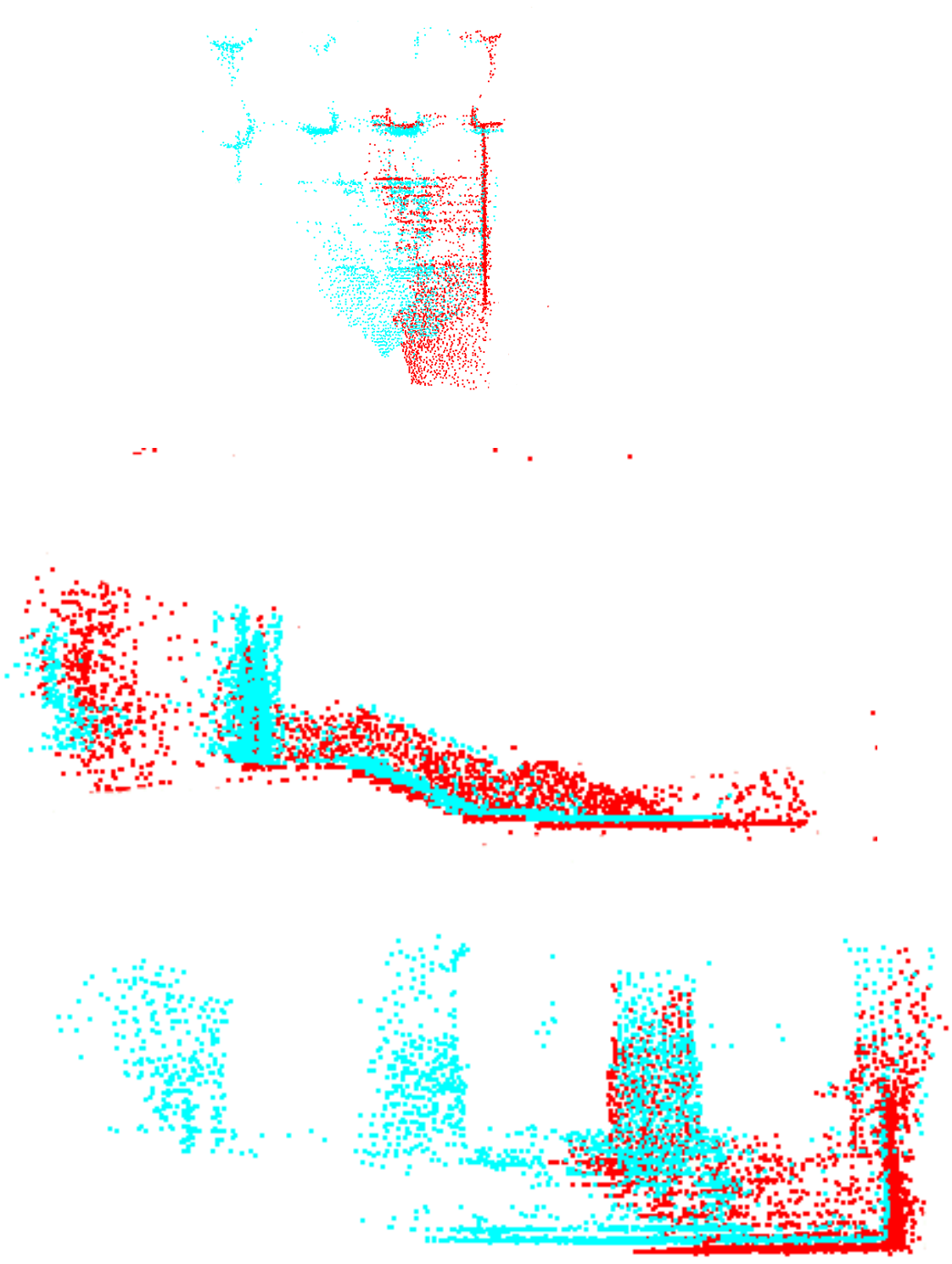


Figure 4.9: A top, side and front view of the registered reconstructions. Notice the good registration of the stairs, the ground plane, the right wall and the pillars.

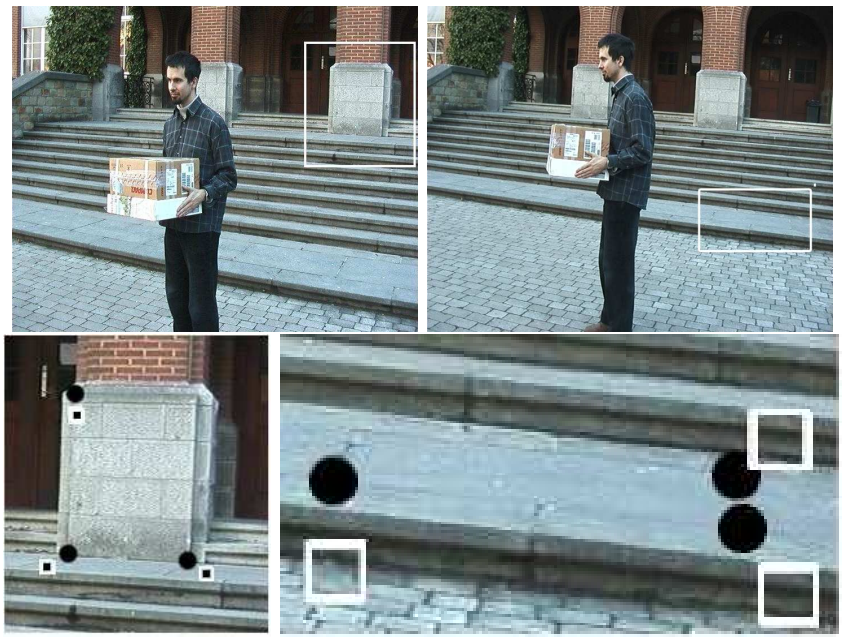


Figure 4.10: Just like the previous experiment, manually selected features from one image sequence are projected into the other image sequence.

such features need not be simultaneously visible (maybe visible from different cameras in different time instants) in both cameras which is a necessity in many multicamera systems. Another interesting remark would be how to determine which part of the segmented scene corresponds to the background and which to the foreground. Upto now, we assumed this to be known a priori. This, however, can be achieved automatically in several ways, e.g. with a typical assumption that the biggest object is the background, or with a more elaborate technique that is described in previous chapter, if the foreground motion complies to a certain constraint. In this regard, our framework itself is capable of identifying the corresponding segmentation parts between the two sequences, since a wrong choice would result in a higher error value after the final minimization. Although such an approach will not explicitely label the moving objects explicitely as the background or the foreground, it will significantly decrease search space.

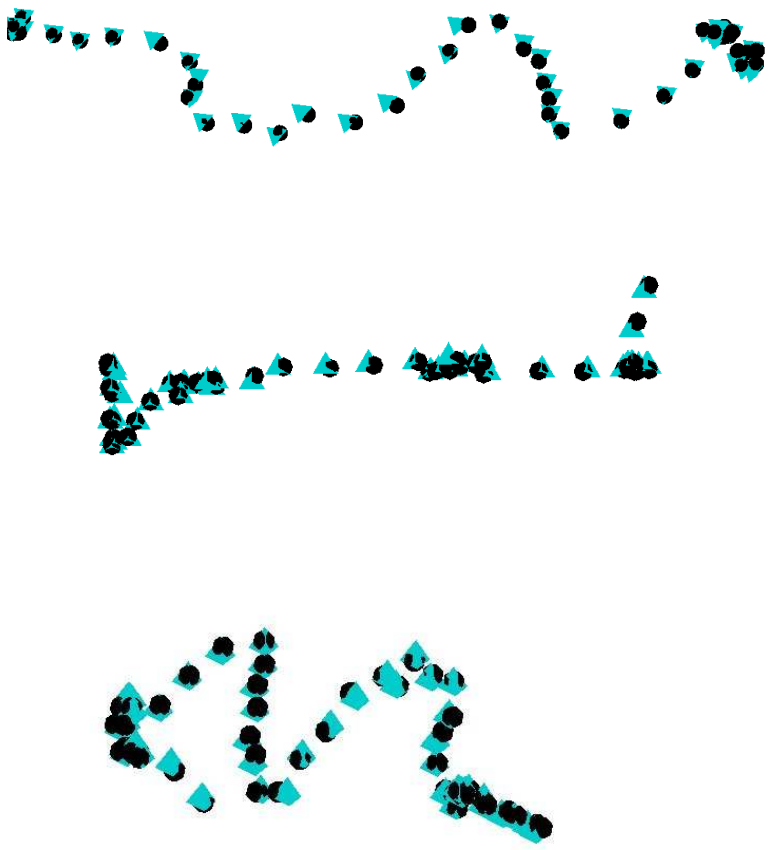


Figure 4.11: Different views on the resulting path of the centroid of the foreground reconstruction in the first image sequence when displaced by object transformations coming from the first sequence (circles) and the second sequence (triangles) after registration. The resultant paths for other points are quite similar, hence they are not shown.

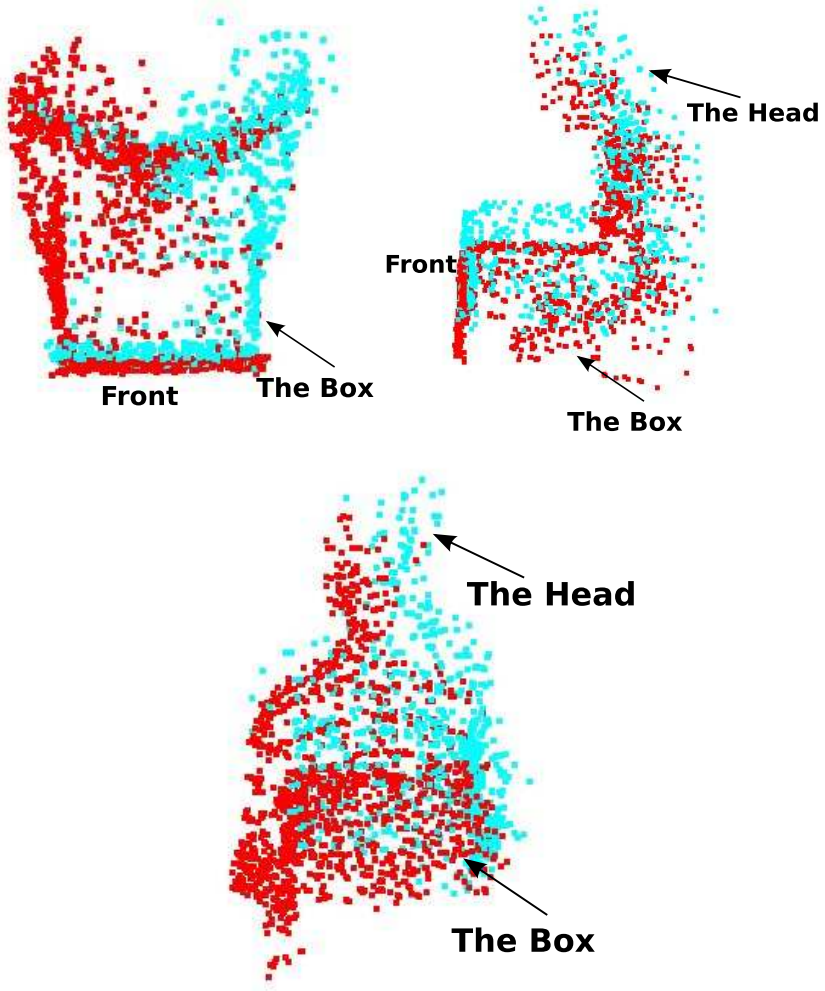


Figure 4.12: A more or less top, side and front view of the reconstructed person and the box he carries. The rotation and the scale looks realistic but the translation is still a bit off-shoot in the direction of gravity.

Chapter 5

Simultaneous Segmentation and Reconstruction of Dynamic Scenes

5.1 Introduction

In the course of this dissertation, we mostly discussed the resolution of various geometric ambiguities such as unknown relative scales and space-time transformations or the correct identification of the background object. However an underlying, restrictive assumption was the availability of the already segmented and reconstructed rigidly moving objects. This assumption is not unrealistic as motion segmentation is one of the basic problems in computer vision and consequently there have been many techniques that have been reported on that issue as we will describe shortly. However, doing SfM estimation and segmentation simultaneously rather than as two separate processes has significant advantages. This is the main topic of this chapter. A short form of the work that is presented here is also published in [OSG07].

We propose a general and practical algorithm, which can handle long and realistic sequences. The proposed framework simultaneously tracks features, groups them into rigidly moving segments, and reconstructs all segments in 3D. Such an online approach, as opposed to batch processing techniques, which first track features, and then perform segmentation and reconstruction, is vital in order to handle small foreground objects. The necessary modules of such a system are identified, both unexplored theoretical issues and practical challenges are highlighted. Theoretical issues include the proper handling of different situations, in which the number of independent motions changes. Objects can enter the scene, objects previously moving together can split and follow independent trajectories, or independently moving objects can merge into one common motion. We derive various model scoring criteria to handle

these changes in the number of segments. Practical issues include robust 3D reconstruction of freely moving foreground objects, which often have few and short feature tracks.

5.2 A Review of Motion Segmentation

To divide a scene into its moving components is one of the first tasks of many CV applications such as video compression where each moving object and their motion is encoded disjointly thus gaining significant bandwith, navigation where moving obstacles are detected and thus a collision is avoided, video retrieval where moving objects are detected and indexed for database searches, surveillance applications where the foreground objects and motions are detected for unusual events, human motion modeling for generating realistic motion models, etc. One way to categorize the related work of that rich field is based on the dimension of the motion model which is either 2D or 3D.

The 2D approaches are many and diverse. Here only a rough summary is going to be given. One set of popular techniques can be bundled under the name layered approaches, where the works of Darrell and Pentland [PD91], and Wang and Adelson [AW94] are often cited. Each object motion is fitted to a different type of 2D affine motion model for consecutive images and the pixels in the images are assigned to the most proper motion model, consequently to a layer. There are two main approaches to estimate the motion parameters for each layer and the pixels associated with it. First one can assume a dominant motion for most of the pixels (e.g. [IAB*96, ASB94, CB99]), which is typically the background motion. The pixels which do not correspond to this motion model are labeled as the foreground objects. Further recursive processing on those foreground pixels would result in further layering and consequently more detailed description of the existing motions in the scene. Another approach is simultaneous detection of all the moving objects and their motion parameters. Typically a large number of motion models is generated (e.g. for each small patch of an image). Then similar motions are grouped together (e.g. [AW94, SA96]) by different statistical techniques such as k-means[Mac67] or Expectation-Maximization[DLR77]. One problem with such dense (pixel level) segmentation approach is the assignment of pixels to the correct motion layer in areas which have homogeneous color intensities since a pixel can fit to more than one motion model in that case. A common solution to such problems is to use a smoothing technique. A popular such technique is the application of Markov Random Fields[Li01] which forces the neighbouring pixels to have the same assignment (e.g.[CB99, WA96]).

So far the discussion on 2D motion segmentation was purely motion specific. However such an approach is not optimal as it ignores the wealth of information that is present in the image intensities. One general approach which combines both motion and intensity cues is from Shi and Malik[SM98] where normalized cuts are applied to a graph structure which is generated from such available

cues. Other examples include the work of Black[Bla92] where different energy terms of intensity, boundary and motion are used in a Markov Random Field. Altunbasak *et. al.*[AEM98] presented a system where the images are first segmented individually according to their color values and later those segments are clustered together if they follow the same motion. Smith [Smi01] reported an approach which exploits the edges and their motion in a video since edges are a very good cue for segmentation.

Although 2D segmentation is popular due to its simplicity, a more general technique must take the 3D nature of an object into account. Indeed some types of object motions, such as a strong rotation around the center of mass, cannot be modeled with 2D transformations. Therefore, the alternative is to compute both the 3D structure and the motion of each object. This created a new hybrid sub-field where SfM and segmentation are studied together.

As stated before, various approaches for two perspective views [VSMS02b, WS01b, SS06], multiple affine views [CK95, VH04], linearly moving objects in multiple affine[HK00] and perspective views [HK03, WS01b] and finally multiple perspective views [KSW06] have been reported so far. In such settings, achieving correct motion segmentation is an inevitable step, and as there is a trade-off between the number of segments that are computed and the fitting error (increasing the number of segments always decreases the error), choosing the correct number of models and their types is a fundamental problem. Consequently, model selection became a popular concept in this arena [Tor98, SS06, KSW06, Kan01]. Recursive filters investigated with static scene SfM are also applied to dynamic scenes in either parametric [TDP94, SP94] or non-parametric [GQZ05] form. Those techniques are succinctly described in sub-section 1.2.

There are also interesting research reports utilizing 3D segmentation and reconstruction for the purpose of video retrieval. Sivic *et. al.* [SFZ06] presented a system where tracked patches throughout a video are grouped together with consistent affine subspaces, but an explicit 3D model is not extracted. Later those objects can be used to index the video which is based on their previous frame based method [SZ03] which is coined as *Video-Google*. Rothanger *et. al.* [RLSP07] presented a paper (an extension of their previous static scene approach [RLSP06]) in the same vein where affine patches are tracked and grouped into related 3D moving objects with the significant difference of explicit 3D modeling which can be utilized for a stronger matching procedure. A common trait among those two works is the application of the hierarchical segmentation routine described by Torr and Murray [TM93].

Most of the research so far has focused on basic theoretical and mathematical aspects of the problem, restricting the experimentation to short sequences and rather simple scenes. Here, we build on this research, and work toward a solution of the problem for real-world sequences. This brings up various challenges in both the theory and in practice.

The practical issues mostly arise due to the free motion of foreground objects and their small size, which destabilizes structure from motion computation.

Doing SfM computation online is one significant way to alleviate that problem. In online SfM, 3D reconstruction is carried out in parallel with tracking in order to aid feature tracking in obtaining the longest possible feature tracks, while robust reconstruction algorithms are executed which can utilize short tracks.

Theoretical issues arise due to the complex interactions of objects in the scene, particularly when objects merge or split, i.e.start to move as an independent rigid entity after having been observed as one rigid unit. Such operations occur quite regularly in a real-world scene, mostly because moving objects stop their motion relative to the background, or start to move. Closer inspection reveals that the two operations, where one may at first glance appear the other in reverse, actually require separate, and quite different treatment.

The rest of the chapter is structured as follows: first we will give a brief overview of the model selection concept and a summary of basic techniques we built upon. Then we will identify the essential aspects of a practical 3D reconstruction framework for dynamic scenes in section 5.4. After analyzing the necessary components, we describe practical challenges and possible solutions in more detail. In section 5.5 we describe a real system which implements these ideas, and in section 5.6 follows a more detailed description of some important components. Section 5.7 shows experimental results. Section 5.8 concludes the chapter.

5.3 Model Selection Review

One of the basic topics in statistical estimation is to fit certain parametric models to data, e.g. fitting a line to noisy point measurements. Among such estimation techniques, Maximum Likelihood Estimation (MLE) [Fis36] is the best known. In the MLE framework, the data noise is modeled with a probabilistic distribution (typically Gaussian or a mixture model for robust estimation), and both the parameters of the model and the original point locations that lie on this parametric model are estimated such that where the likelihood of data given those parameters is maximized. Although it is quite intuitive and relatively simple to formulate, MLE has a certain limitation: when there are competing model types, the estimation process always favors the most general model. To give an example, a second degree polynomial will always give lower residuals, (i.e. higher probability) compared to a line model as the second degree polynomial can generate a line as a special case. Another example is the estimation of the correct adjoint line segments that can describe noisy measurements of points which is illustrated in Fig. 5.1. Three different models are fitted to noisy data, where all of the models consist of contiguous line segments but with different degrees of freedom, i.e.they have different numbers of line segments. The blue model (#1) consists of a single line segment. Although the model is quite simple it is very poor in explaining the data. The red model (#3) explains the data very well, giving a residual error 0, however the model is too complex. The green one (#2) looks like a good compromise between the resid-

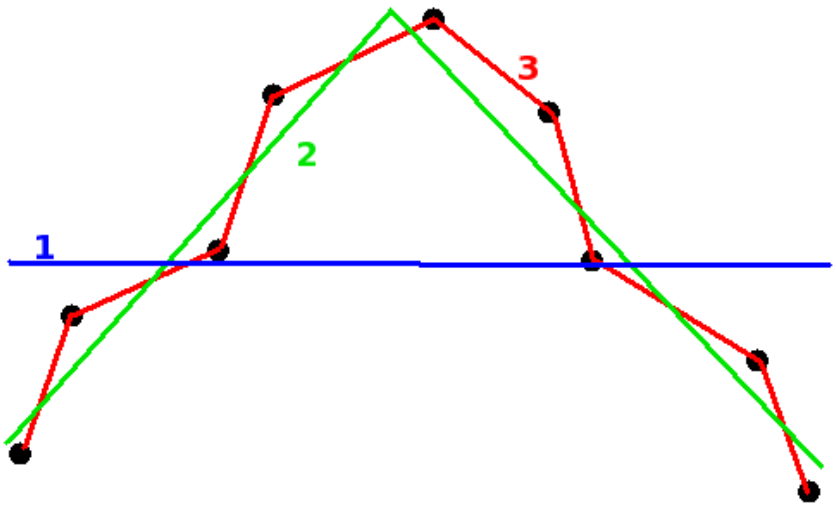


Figure 5.1: An illustration of the model selection problem. The noisy data is fit to 3 different parametric models, each consists of contiguous line segments but with different degrees of freedom. As the number of line segments in a model increases, the residuals decrease.

ual error and the complexity of the model. Considering MLE's bias towards more general models, researchers wanted more effective scoring criteria which not only take account of the residuals but also of the complexity of the model. Various forms of model selection techniques have been proposed. Considering the huge volume of related research, only a crude summary is given here.

One of the earlier answers to the model selection problem has been given by Akaike [Aka74], with what is known as *Akaike Information Criterion* or *An Information Criterion* (AIC). The original method has been devised for time series analysis. AIC aims to minimize the expected residuals of future observations based on the principle that a good model is the one which enables good predictions. The derived criterion is:

$$AIC = (-2)logL + 2k \tag{5.1}$$

where L is the likelihood of the data and k is the number of the parameters of the model. One notable problem of AIC is that, it is not sensitive to the number of the data, hence it is asymptotically inconsistent. Another problem is the dimension of the manifold that the original data lies in. Kanatani [Kan96]

concluded that AIC in that case (dubbed as Geometric AIC) is:

$$GAIC = (-2)logL + 2(dn + k) \tag{5.2}$$

where d is the dimension of the manifold that data lies on and n is the number of the data.

Another approach is Bayesian, where the posterior likelihood of a model is computed by application of the Bayes rule. A complete analytical solution is usually impossible due to integrations over a large number of parameters, so only approximations are made. One such criterion is the Bayesian Information Criterion from Schwarz [Sch78] where the prior distribution of the model parameters is assumed to be a very diffuse gaussian. However it ignores the fact that least squares problems with nuisance parameters have special sparse-diagonal Hessian matrices which contradicts the aforementioned assumption and most structure and motion estimation problems in Computer Vision are of this nature. Indeed, the estimation of the location of a 3D point from a single feature match is only affected by that match, not others. Such a correction is proposed by Torr [BHS00, Tor02] through GRIC (Geometrically Robust AIC), where each degree of freedom of the parametric model and the structure parameters are given different weights. Successful applications of GRIC has been reported in SfM community, e.g. [PVG02, SS06].

A quite different approach to model selection comes in the form of description length of the data. This approach has its intellectual roots in the work of Kolmogorov and Solomov [LV96]. The length of the shortest computer program (A turing machine) that can describe a certain string can be used as a complexity measure (algorithmic complexity) for the string. In the vein of Occam's razor, the shortest program can be thought of as the true theory behind that string. However this Kolmogorov complexity can not be exactly computed hence has to be approximated, such as by MDL (Minimum Descriptive Length) technique of Rissanen [Ris78] where the data is assumed to come from a random process. In this scheme, a code length is approximately computed both for the data and the parameters of the random process that generates the data. The model that generates the shortest over-all code length is selected as the correct model. Since the model parameters need to be coded along with the data, less complex models are favored. The novel work that is going to be presented in this chapter can also be considered in this vein.

Other than the above, there are also various methods that are used in statistics community, such as cross-validation and Structural Risk Minimization. We are skipping them here as they are primarily developed for the learning algorithms.

Model Selection for Segmentation: In Computer Vision, it is very common for the input data to come from separate sources, e.g. multiple lines, multiple motions in the scene etc. In such settings, even though the parametric model type that is describing each source can be assumed to be known,

the number of the existing models and the correct assignment of the observations to the model is not known in general. The role of model selection in this case becomes to choose the correct number of models that describe the data. For instance a trade-off must be made between the number of motion models and the data residuals. A prominent work in that vein is from Torr ([Tor98, TM93]) where a proper segmentation is chosen according to the GRIC score and the initial segmentation is achieved by hierarchical application of RANSAC [FB81]. Kanatani [Kan01] formulates the factorization problem of Costeria and Kanade [CK95] as subspace separation, using his aforementioned Geometric AIC score. Another thread of work is from Schindler *et. al.* [SS06, KSW06] where a general model selection framework is first developed for two views, then multiviews. The work of Schindler *et. al.* plays a significant role for the novel technique that will be described in this chapter. The model selection subroutines that we deploy are derived from that work, which in turn borrows ideas from the article of Leonardis *et. al.* [LGB95]. Following the historical development, first the work of Leonardis *et. al.* will be described, then later the motion segmentation scheme of Schindler *et. al.*.

Leonardis *et. al.* [LGB95] applied model selection and recover-and-select techniques to range image segmentation problems. The basic problem is to segment a range image into simple parametric entities, which are bi-quadratic polynomial in their case. This requires the identification of the models and the data points which belong to them, as typical in any segmentation problem. The initial hypotheses are generated by growing numerous seeds in the image which results in an over-complete description of the scene where a pixel may belong to zero, one or more than one parametric models though only bilateral interactions between the models are taken into account, i.e.a pixel is considered to belong to at most two models in post-processing. The model selection routine is based on an MDL approach where the set of models which describes the data most compactly, i.e. which results in the shortest coding of the data, is selected. In this setting the data points, the residuals and the parameters of the models have different coding lengths so different coefficients are applied to each. A subset of the generated hypotheses are computed as the solution which minimizes a quadratic boolean function of the form:

$$E = \mathbf{m}^T \mathbf{Q} \mathbf{m} = \mathbf{m}^T \mathbf{Q} \mathbf{m} \tag{5.3}$$

where $\mathbf{m}$ is a boolean vector of length n where m_i indicates whether model i is included in the solution or not and n is the number of available hypotheses. The diagonal elements of the $n \times n$ matrix $\mathbf{Q}$ indicates the cost-benefit value for a particular hypothesis and the off-diagonal elements account for the adjustment due to bilateral interaction of the hypotheses: as a pixel may belong to two surface models, however it must not be coded twice. A greedy search can be used to find a reasonable minimum though a global minimum is not guaranteed.

Schindler and Suter [SS06] used the aforementioned machinery for the motion segmentation problem in two perspective images. After generating a large number of hypotheses for two-view relations, using the MDPE technique of

Wang and Suter [WS04] (a kind of random sampling technique where the noise of data is also estimated) a subset of those hypotheses are selected which describe the data best by applying a bi-quadratic boolean optimization scheme like the above. The cost function that is used is an overall GRIC score.

In their following work, Schindler *et. al.* [KSW06] extended their technique to multiviews. The steps they follow can be summarized as: 1. Sampling of two-view essential matrices between every consecutive frame pair, 2. Clustering of those essential matrices to decrease redundancy 3. Creating 3D structure and motion hypotheses after linking essential matrices by using temporal consistency 4. To choose a correct subset of those hypotheses which describes the data most compactly with a bi-quadratic boolean optimization similar to their previous work.

The basic difference with their previous work is the use of full 3D structure and motion models over many frames rather than just 2-view constraints and adopting a MDL inspired approach rather than GRIC. Basically the savings in the coding length of the data caused by SfM representation is maximized. Due to Shannon's theorem, maximizing probability is $\mathcal{P}$ is equivalent to minimizing the codelength since two are related by $\mathcal{L} \sim -log(\mathcal{P})$.

The original paper [KSW06] considers a relatively complex scenario where different rigid objects can be visible during arbitrary sub-sequences and a feature on an object need not be visible during the whole motion of that object. However the novel system presented here applies that mechanism in a time-frame that is small compared to whole sequences. This enables us to safely assume complete feature tracks (for that time window) and the visibility of the moving objects in every frame. Consequently, indexing costs of the features can be ignored and the resultant overall code-length expression is much simpler than the original one. By applying model selection online and in limited time-frame, a globally optimal solution is traded-off for reasonable execution times while keeping the segmentation results correct. The original formulation can be found in the multi-body segmentation work of Schindler *et. al.* [KSW06].

Assuming uniform probability density over the feature search window, the coding length of all the feature tracks without any SfM representation can be written as:

$$\mathcal{L}_{+} = -FN \log \frac{1}{w^2} \tag{5.4}$$

where F is the number of frames, N is the number of feature tracks (or 3D points) and w^2 is the search window size. This is a straightforward coding scheme without using any inherent structure or dependencies in the data. SfM results in a different kind of coding hence coding length. In this scheme, the 3D structure of the points plus the 3D motion of the camera is computed first, so then only the reprojection errors need to be coded. Assuming the residuals have a zero-mean normal distribution with standard deviation σ for both x an

y coordinates and they are independent, the codelength is:

$$\mathcal{L}_{a-} = -\log \prod_{i=1}^{F} \prod_{j=1}^{N} G(r_{ij}, \sigma) = \frac{1}{2\sigma^2} \sum_{i=1}^{F} \sum_{j=1}^{N} r_{ij}^2 + FN \log 2\pi\sigma^2 \tag{5.5}$$

where r_{ij} is the residual of the point j when it is projected on the image i and G is the 0 mean normal distribution with std=σ. However the 3D structure and motion parameters also need to be coded. Similar to Torr [BHS00]'s GBIC approximation, each parameter of the structure and motion can be coded with the number of equations that is used to compute them. There are $3N$ structure parameters where each is computed from $2F$ equations. Also there are $(6F - 7)$ unknown motion parameters (calibrated intrinsics and ambiguity upto a similarity transform) each of which are computed from $2N$ equations, so the coding length for structure and motion becomes:

$$\mathcal{L}_{b-} = \frac{3}{2} N \log 2F + \frac{1}{2} (6F - 7) \log 2N \tag{5.6}$$

Such an approximation of parameter log-likelyhoods with the number of data that is used to estimate them has been documented at various references (Torr [BHS00], Ripley [Rip96], Schwarz [Sch78]). By using the structure and motion representation, the codelength is reduced by $\mathcal{L}_+$ but increased by $\mathcal{L}_{a-} + \mathcal{L}_{b-}$ so two times the total savings is:

$$2\mathcal{D} = 2FN \log \frac{w^2}{2\pi\sigma^2} - \frac{1}{\sigma^2} \sum_{i=1}^{F} \sum_{j=1}^{N} r_{ij}^2 - 3N \log 2F - (6F - 7) \log 2N \tag{5.7}$$

As stated in the original paper [KSW06], the bilateral interactions must also be accounted for. This is handled by assigning an ambiguous feature (a point which belongs to more than one object) to the closest model and correcting overall coding length.

So far the the existing works in the literature have been reviewed. From now on our original contribution will be presented.

5.4 A General 3D Reconstruction Framework for Dynamic Scenes

5.4.1 Requirements

The main task of a SfM framework for dynamic scenes is to identify all major moving objects in each frame, and to compute their 3D structure and motion with reasonable accuracy, while maintaining scalability to realistic recording times (at least several hundred frames). It has to properly handle sequences, where the number of moving objects is a priori unknown, and changes over time. This includes not only objects appearing or disappearing from the field

of view, but also objects appearing due to split operations, and disappearing due to merge operations. Furthermore, long sequences do usually not guarantee long feature tracks: the system has to deal with situations, where feature tracks are short, due to frequent self-occlusions of freely moving foreground objects.

The analysis suggests that feature tracking, segmentation into independent objects, and 3D reconstruction shall not be carried out as independent tasks, but in an interleaved way, as the sequence progresses, so as to be able to use recovered information from previous frames directly. In particular, feature tracking benefits from known 3D motion, and robust reconstruction from reliable segmentation.

For static scenes, the described interleaved process of tracking and reconstruction has been well studied [HZ00, PVV*04, DNB04, BTZ96]. In the case of dynamic scenes, we additionally estimate the number of moving objects online. For initialization, one of the existing methods for short sequences can be used (see below). However, when the number of objects changes in the course of the sequence due to a split or merger, we need to detect the change and react accordingly. In the following, we propose model-scoring methods tailored to the different situations in order to achieve this.

5.4.2 Splitting and merging motions

In the setting described above, splitting and merging of objects are two phenomena that need to be handled carefully. By the term splitting, we mean the phenomenon that several objects, which so far have moved as one rigid body and thus have been covered by a single 3D SfM model, start moving independently. Typically, this happens when a part of the previously static background starts to move in the middle of the sequence, such as a car leaving a parking lot. In a static SfM algorithm, the tracks on the smaller resulting object would simply be labeled as outliers. Instead, we aim to detect such an event, and reconstruct both objects correctly.

Merging is the opposite: independently moving objects rigidly attach to each other and start moving as one. Again, we aim to properly detect such an event and transform the motion models accordingly. One may ask, whether this is necessary, given that separate models should still be correct. However, proper merging will result in more accurate modeling (due to the reduced number of parameters which need to be estimated from the feature tracks), and avoid the problem of assigning new tracks correctly to one of two very similar motions. Furthermore, we will see that it has beneficial effects in resolving scale ambiguities.

5.4.3 Splitting versus Merging

In spite of their apparent relationship, splitting and merging are two significantly different problems, both formally and practically. At first glance, they simply are inverse operations (in the sense that one becomes the other when

playing the sequence in reverse order). This may lead one to believe that they can be treated with the same mathematical model. However, there are remarkable differences between the two, partly due to theoretical properties of the multi-body SfM computation, and partly due to practical issues, which arise in real-world situations.

It is instructive to look at an example to illustrate the subtle theoretical differences: consider a rigid object A, for which the structure and motion are already known. At a certain point in time, an object (a set of 3D points) B split off from A, i.e.starts to move independently of the remaining points A'. Since the 3D structure of B has been reconstructed before splitting (as a part of A), there is no scale ambiguity between the two new objects, so the only problem is to find the new rigid transformation of B relative to the camera, which can be done with simple resection in a single frame. Now assume that we are processing the sequence in reverse order. Initially the objects A' and B are moving independently, and at a certain point in time they merge. This event can only be detected reliably, if we wait long enough: in short sequences, there is a danger of "apparent fusion", because of a near-degenerate configuration. One can often fit a reasonable joint model, if the sequence is short, because it takes time to accumulate enough camera translation (baseline) Furthermore, the two objects have been reconstructed separately, so there is a scale ambiguity between them [OCVV04], which needs to be resolved (this issue is detailed in the next subsection). To this end, we again need to accumulate enough baseline.

Another issue is that merging more than two objects can be safely accomplished by iterative pairwise merging. The contrary is not true for splitting: if an object splits into 3 parts (or into 2 parts and some outliers), there is no split into 2 parts which would produce valid structure and motion estimates. Agreement can be tested greedily, disagreement cannot.

Further differences arise from the practical point of view: when a 3D object is divided, it is desirable to conjure the new 3D models immediately. There are two reasons for that: firstly, it is quite possible that one of the new motions is mostly a rotation around a point close to the object, which will quickly cause loss of features due to self-occlusion. Consequently, a proper motion model must be instantiated as early as possible so as not to loose the object completely. On the contrary, a merge usually means that a smaller object attaches to a large background, and there is no immediate danger of losing large numbers of features. The second practical reason is related to guided tracking. 3D structure, which depends on the availability of a motion model, is an important help for reliable feature tracking. If division is delayed, the tracking will suffer, whereas after merging the old motion is still valid, so a motion model is always available.

5.4.4 Relative Scale Resolution

As described in chapter 2, one subtle problem in 3D reconstruction of dynamic scenes is the relative scale ambiguity between the reconstructions of different

moving objects. If that ambiguity is left unresolved, it results in unrealistic 3D reconstruction. Previously we have suggested how motion constraints or motion consistencies between several views can be used to resolve the ambiguity. In this setting, splitting and merging operations connect 3D objects to each other, which has strong implications on the relative scale without resorting to predefined motion constraints.

Indeed, as stated before, no matter how many splits occur, the relative scales of all objects that stem from the same parent object will always have the correct relative scale. Similarly, if an object is the result of any number of merging operations, the relative scales of all its previously independent components are determined. Splitting and merging propagates the scale between objects in a transitive way. To give an example, if object A splits into A_1 and A_2, the relative scale between them will be correct. The same is true for the object B when it splits into B_1 and B_2. When now A_1 merges with B_1, this sets the relative scale not only between A_1 and B_1, but also between A_2 and B_2. These dependencies require some book-keeping effort, but in many cases resolve most or all scale ambiguities.

5.4.5 Practical Considerations

Although theoretically on equal footing, the nature of the dominant (usually static) scene background is quite different from small moving objects, and the SfM algorithms designed for static scenes only apply well to the dominant background. Foreground objects are generally small, hence have few feature points to track and small aperture angles, which makes them susceptible to noise. In contrast to a static background, for which the motion originates from the moving camera, foreground objects can move quite freely, with more frequent self-occlusion and strong illumination changes, which additionally causes shorter feature tracks. As a consequence, successful feature tracking and exploitation of short tracks are the most crucial factors for successful 3D structure and motion estimation for such objects.

However that important issue has been ignored by previous systems [VSMS02a, CK95, TDP94, SP94, GQZ05] which assume full length or outlier-free feature tracks as input. This effectively limits the number of frames that can be processed, since outlier-free tracks through the entire sequence are all but impossible to obtain with today's feature trackers (such as KLT [TK91]). The strict requirement can be alleviated by assigning small weights to outliers or invisible portions of a track [GQZ05, TDP94], but the question remains, how to deal with novel feature tracks, which are invariably required in order to compensate for tracking loss. To date, a principled treatment of the problem is only possible by interleaved, incremental tracking and segmentation/reconstruction. This allows the feature set to evolve over time, naturally combines the structure and motion from previous frames with new short tracks to robustly utilize all information, and allows for guided tracking with strong, but automatically generated motion constraints, for more accurate and more efficient feature ex-

traction.

5.5 An example implementation

5.5.1 Overview

In this section, we describe a practical system built on the ideas introduced in previous sections. In terms of capabilities the closest work to our system is the one presented in Schindler *et. al.* [KSW06], where an over-complete set of motion hypotheses for the entire sequence is generated and then pruned to an optimal set using model selection. Such a computation is time-consuming because of the combinatorial explosion of potential motions, and quickly becomes intractable, as the number of frames increases. Here, we prefer to sacrifice parsimony over the entire sequence to local parsimony over shorter time windows, to achieve scalable execution times. The model scoring we adopt to solve splitting, merging, and initialization, has been inspired by that paper, but in our system it is carried out online (i.e.each frame is incrementally processed but the system is not real-time yet), rather than as a global batch optimization. Consequently, the system is capable of handling long sequences (more than 200 frames) and capable of processing novel input frames while giving satisfactory segmentation and reconstruction results. As another important consequence, the system is amenable to real-time implementations.

The core SfM routine is based on well-established sequential SfM techniques for static scenes, which are applied to each of the objects in the dynamic scene. Splitting, merging, and the appearance of new objects are handled with task-specific sub-routines, which are based on model scoring.

The proposed system also offers an advantage if the goal is only motion segmentation: compared to approaches where only a limited number of frames is used for motion segmentation, the ability to project 3D points reconstructed in distant frames helps to demarcate a moving object more precisely, including its homogeneous parts, which leads to more complete object descriptions in difficult cases.

5.5.2 System Details

The core engine of the system is a multiple model version of the standard type of SfM framework [BTZ96, DNB04, PVV*04] for static scenes. In order to initialize the algorithm, corner features are detected and tracked over a small number of frames. A 3-view motion segmentation algorithm is applied (see 5.6.1) to the first, middle, and last frames of the initialization sequence. The algorithm yields an initial segmentation of the scene into rigidly moving objects. For each segment, the 3D structure and the camera motion are recovered independently via epipolar geometry decomposition. As new frames arrive, the existing feature points are tracked and new ones instantiated, while incrementally computing the new camera pose w.r.t. each moving object with standard RANSAC

resectioning. Newly added 3D points are assigned to the motion they fit best, assuming normally distributed reprojection errors. If all reprojection errors exceed a threshold, no 3D point is generated, however the track is not rejected immediately, since it could belong to a new motion. As explained above, it is important to detect split events as early as possible. Therefore, if there is a significant number of outliers, which may be caused by a split, a sub-routine is called (see 5.6.2) which tries to detect new motions in the unexplained tracks. Further sub-routines, which are not that critical, run at regular intervals:

- The initialization routine is employed to detect new motions in the set of unexplained feature tracks.
- A sub-routine is called, which checks the set of motions for mergers (see 5.6.3).
- Finally, we periodically run a bundle adjustment to stabilize the global solution.

The algorithm is described in pseudo-code 5.5.2. Waiting periods mentioned indicate that these steps are only carried out periodically to save time. However one can practically disable this parameter by setting it to 1, thus running the related sub-routines at every frame, which would result in no better segmentation but a significant computation time drag. This parameter is directly related to the action speed of the scene, i.e.if the object move fast, it requires less number of frames to resolve mergers and to detect new motions. Another such parameter that needs to be set by the user of algorithm is the number of the unexplained tracks (i.e.the tracks which do not fit to any existing motion model) which hints a possible new motion. This parameter depends on the image size of the target objects and their texture properties (since a higly textured objects give more feature tracks). Fortunately, our experiments showed that the system performs well on a range of afferomentioned parameters.

5.6 Details on the critical Sub-routines

5.6.1 3-View Motion Segmentation

This routine serves to initialize motion models for newly appearing objects. It takes new feature tracks of a predefined length, which have not yet been assigned to any motion. Several possible strategies can be followed here. The minimal solution is to use only the first and last frame and run a two-view motion segmentation [VSMS02b, WS01b, SS06, TM93]. However, SfM for only two views is notoriously unstable. As the other extreme, a full N-view computation distinguishes individual motions very well, albeit with a considerable computation time. Here we strike a balance between those options and choose a 3-View segmentation algorithm on the first, middle, and last frames of the sequence. The method is a simplified version of the N-View segmentation of

Algorithm 1 Overview of dynamic structure and motion pipeline.

1. Instantiate new features, and track all the features. The standard KLT[TK91] algorithm is used here.
2. **If** insufficient frames for SfM initialization (not enough to accumulate parallax for 3D computation):
   - **goto** step 1;

   elseif sufficient parallax **and** no SfM:
   - perform initial motion segmentation and 3D structure computation;
   - **goto** step 1;

   else continue;
3. Try to compute new motion estimates for the active models.
   If too many outliers for a motion model:
   - try to split.
4. **If** waiting period is over **and** the number of unexplained tracks is above a threshold (see the text):
   - try to detect new motion models.
5. **If** waiting period is over:
   - try to fuse active motion models greedily;

   goto step 1;

Schindler *et. al.* [KSW06] which was summarized previously. As stated earlier, an over-complete set of possible motions is generated by random sampling, and the best subset selected with model selection. The overall log-likelihood of the 3D points, camera parameters, and reprojection errors is maximized. As stated before, an additional assumption is used here: during the short object initialization period, the features that are used are visible in all three frames. This assumption causes the indexing cost to drop to 0. With the number of frames $F = 3$, and the number of points of the kth candidate model N_k, the codelength savings by coding with that motion model (Eq. 5.7) becomes:

$$\begin{aligned} 2\mathcal{D}_k =& 6N_k \log \frac{w^2}{2\pi\sigma^2} - \frac{1}{\sigma^2} \sum_{i=1}^{3} \sum_{j=1}^{N_k} r_{ij}^2 \\ & - 3N_k \log(6) - 11 \log(2N_k) \end{aligned} \tag{5.8}$$

where r_{ij} is the reprojection error of point j in frame i. To summarize, the first two terms are the savings in description length, if the image points are coded as the projections of the 3D scene points rather than by their $(3 \times 2 = 6)$ image coordinates. The third term is a first-order approximation for the description length for N_k structure points, estimated from 6 observations, and the fourth term for $(3 \times 6 - 7) = 11$ motion parameters (3 views, each has 6 DOF and 7 is the DOF of unknown similarity transform), each estimated from $2N_k$ observations.

To account for the fact that overlapping motion models compete for image points, the interaction between models is also modeled, by computing the first two terms of Eq.(5.8) for the points in the overlap. This is necessary to eliminate the affect of the features which are inliers to two models. The common feature is coded with the motion model which it fits best, thus it is not coded twice. Interaction costs only arise, if both involved models are selected, leading to a quadratic boolean selection problem that is mentioned in subsection 5.3.

5.6.2 Splitting

Splitting events show up as a sharp increase in the outlier ratio when computing the new camera parameters for the parent object. Consequently, new motion models are searched in those outliers. As stated before, splitting should be decided instantly, because in practice tracks tend to be short due to self-rotation, and because the tracking suffers while the decision is delayed. Since 3D structure is already known at this point, the decision can be made on the basis of a single frame by inspecting resection results[1]. Considering the possibility that more than one object split off at the same time, and that some points may be real outliers, a recover-and-select approach is adopted, similar to the 3-view segmentation routine. Multiple hypothetical camera matrices are generated by random resection, then an optimal set of cameras is selected to explain the 3D-2D correspondences. With the same symbols as above, the saving of model k when using resection on a single frame are

$$\begin{aligned} 2\mathcal{D}_k =& 2N_k \log \frac{w^2}{2\pi\sigma^2} - \frac{1}{\sigma^2} \sum_{i=1}^{N_k} r_{ij}^2 \\ & - 6 \log(2N_k) - 3\frac{N_k}{\widetilde{F}} \log(2\widetilde{F}) \end{aligned} \tag{5.9}$$

The first and second term are the benefit of modeling the points *in the new frame* by 3D structure and motion, rather than as outliers, the third term is a first-order approximation for the coding length of the new camera motion , and the fourth term is an estimate of the change in structure coding length: while during resection the structure does not change, the structure and motion will

[1] In fact, if a sequence of more than one frame is used, and the decision is made for a certain split, then there is a subset of the sequence, which will produce the same split through single-frame tests

be jointly re-estimated once the new model is accepted, and the larger number of observations per 3D point will slightly increase the coding length. This contribution should theoretically be computed separately for each 3D point, depending on the length of its trajectory. Here, we have struck a practical compromise. We count the average number of frames $\widetilde{F}$, during which a feature point on the object remains visible, and equally divide the coding length between these frames.

Again, interaction costs also have to be considered, and the selection is carried out by solving a quadratic boolean problem.

5.6.3 Merging

In the light of previous theoretical arguments we opted for a merge detection algorithm that uses a predefined number of frames. A bottom-up approach is adopted where fusion is carried out greedily. Such a strategy has been used in a different context [LPB05] before. Models with low bilateral reprojection errors are selected as candidates for merging. Then, we again resort to model scoring to decide whether the joint model after a merge is a better explanation of the data than the two separate models. The log-likelihood for *one* model is given by

$$\begin{aligned} 2L &= \tfrac{1}{\sigma^2}\sum_{i=1}^{F}\sum_{j=1}^{N} r_{ij}^2 + 2FN\log(2\pi\sigma^2) \\ &\quad +(6F-7)\log(2N) + 3N\log(2F) \end{aligned} \tag{5.10}$$

where F is the number of frames, N denotes the number of 3D points and the rest of the notation is the same as in Eq. 5.8. The total score of the separate model hypotheses is the sum of the two models' individual scores. Note that when comparing two separate models with the joint model, based on the same data, the second and fourth term cancel out. Please also note that the residuals r_{ij} for separate models are different from the ones for a single model.

In the merging case, it is not necessary to solve a full selection problem. Rather, we only need to compare the coding length of the joint model after the merge with the total coding length for the two separate models. If the joint coding length is shorter, the two models shall be merged, else, they are kept separate.

5.7 Experiments

The system has been tested with four different real image sequences.

The Market Sequence: This sequence consists of 98 frames taken from a longer video that is recorded in a supermarket where the samples were shown in Fig. 3.3. However in order to test the merging and splitting, we manually created a novel (but still realistic) video by taking a sub-sequence of the original video and appending to this sub-sequence a mirrored version of itself (See Fig. 5.2). Although it is somewhat of a toy sequence, it already

Figure 5.2: Market sequence results. Initial segmentation, merging and splitting steps are successfully carried out.

demonstrates the basic capabilities of the method pretty well. A shopping-trolley initially moves backward (frame 1 and 2), stops for a while and merges with the background (frame 3), and then splits off again, moving forward (frame 4). In the mean time, the camera moves arbitrarily. Fig 5.2 shows the segmentation results for the input samples. The results of the initial motion segmentation, merging and splitting steps are clearly visible respectively in the first, the second and the last sample frame. Note that, after the merger the points belonging to the foreground object still continued exist which is a strong proof that the merger operation has successfully transformed the foreground object reconstruction to the proper scale and position. Otherwise those points would get lost immediately due to big reprojection errors.

The Garden Sequence: This experiment is carried out on a 250 frames long garden test set (see Fig. 5.3) where the segmentation results are in Fig. 5.4. The sequence starts with a person who is carrying a paper box while the camera is also moving (frame 1). Later another person shows up from the left (frame 2) and leaves the scene (frame 5), and finally the remaining person merges with the background (frame 6). The segmentation results in Fig. 5.4 demonstrate that the motion detection and merging operations are successfully carried out.

Figure 5.3: 6 frames from the 250 frames long garden sequence in row-wise order.

The Car Sequence: One of the successful application areas for SfM algorithms is movie post-production where artificial 3D objects are added to the original image sequence. One opportunity that comes with the proposed technique is the possibility of augmenting not only the background but also the moving foreground objects while preserving depth consistency, which is usually a problem due to relative scale ambiguities between the components (However this capability is limited unless the dense segmentation maps of the moving

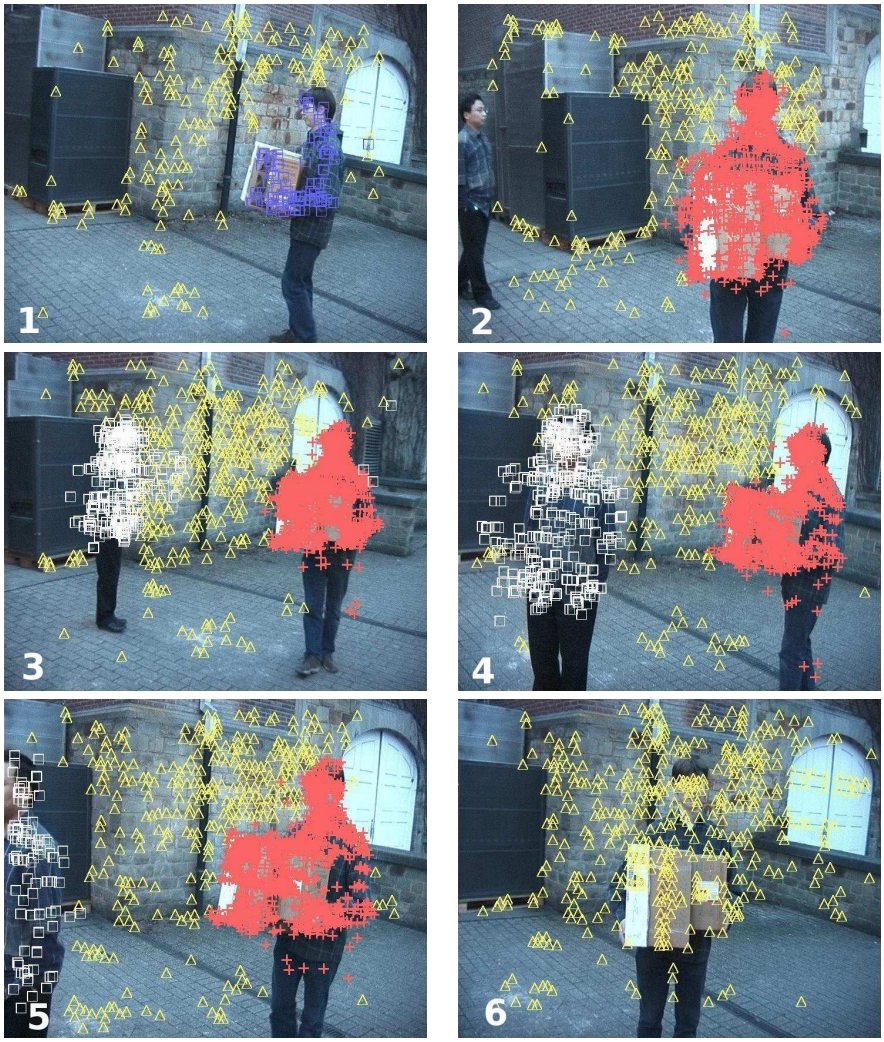


Figure 5.4: Garden sequence results. Initial segmentation, newly appearing motion detection and merging steps are successfully carried out.

objects is available). Thus, we tested our technique on a 107 frames long sequence taken from the movie "2 Fast 2 Furious" (see Fig. 5.5). Although there is no merging or splitting going on, the sequence has all the characteristics of a problematic sequence: freely moving shiny objects with few and short feature tracks. The experimental results show the effectiveness of simultaneous segmentation and reconstruction.

Fig. 5.6 shows two sample frames from the segmentation results where each

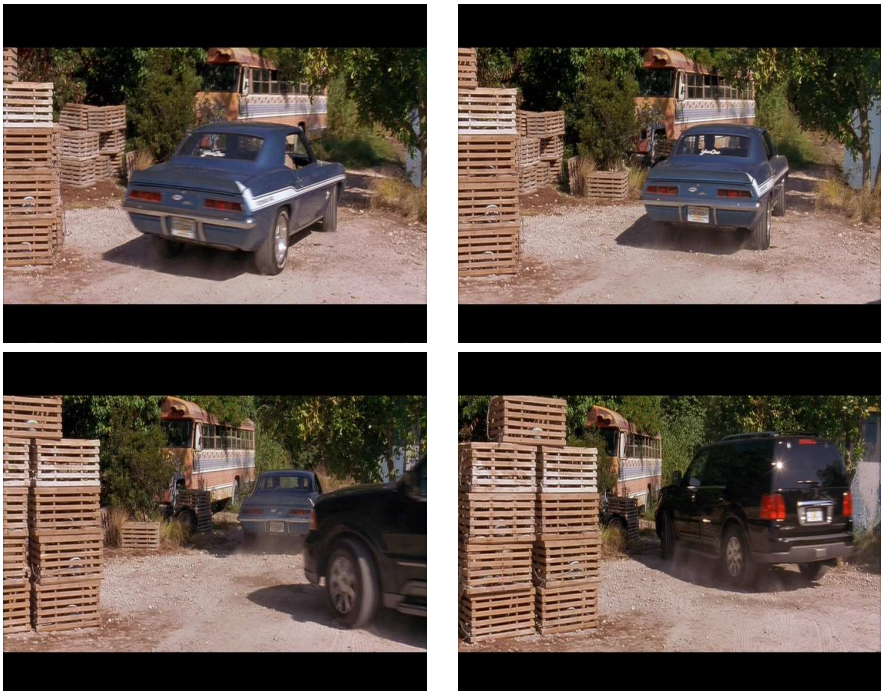


Figure 5.5: 4 frames from the 107 frames long car sequence from the movie "2 Fast 2 Furious".

of the two moving cars and the background are clearly demarcated. To show the quality of the SfM procedure we augmented the original sequence with artificial objects. Their stability in the accompanying video corroborates that SfM was successful. Fig. 5.7 shows a sequence where each object was augmented with calibration patterns.

The Bus-stop Sequence: This is a test set (See Fig. 5.8) where all the capabilities of the technique are demonstrated. A bus enters a scene while the handheld camera moves forward on a more or less linear path. The bus stops near the bus-stop, then pulls off again while a second car appears behind the bus. The bus and the car almost follow a linear path. The lighting conditions are bad as the shadows of the trees are wiping out all features on the moving vehicles, the vehicles occlude many features from the background, and the car has strong specular reflections.

Fig. 5.9 shows sample frames from the segmentation results, where the initial detection of the moving vehicles, the merging of the bus with the background and its split from the background are demonstrated. We also augmented this sequence with artificial patterns (See Fig. 5.10). They move consistently with the objects they are attached to. Note that the pattern that is attached

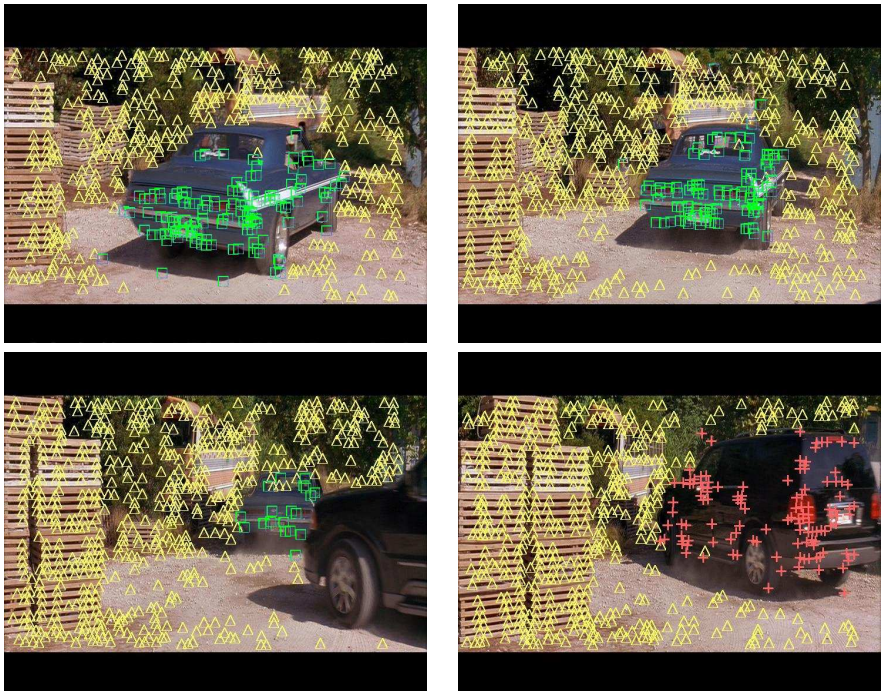


Figure 5.6: The segmentation results for the car sequence. The 3 major motions are detected.

to the bus is rendered with the camera matrices computed for the background after the merging operation, showing that the relative scale resolution worked very well. At first sight this claim looks not very-well grounded, since the objects are rendered from the original camera locations. However note that after the merger, the artificial object on the bus is being rendered by the background camera matrices since the camera matrices for the bus do not exist anymore. Consequently the scale of the object, its location and actual camera motion relative to the background must be in compatible scales.

The 3D reconstruction results (see Fig. 5.11) also show the stability of the technique. The 3D point clouds for the car, the bus and their paths are depicted with different colors and they are rendered from two viewpoints. The first one is more or less a top view and the second one is a side view. Note that when the bus is merged with the background, their relative scales are automatically solved so the position of the bus is correct and its 3D trajectory is almost a line as expected. However the car does not merge with the background nor the bus, so other relative scale resolution techniques as in chapter 2 must be applied, which require additional motion constraints. Here that assumption was linear motion. The resulting car position is quite realistic.

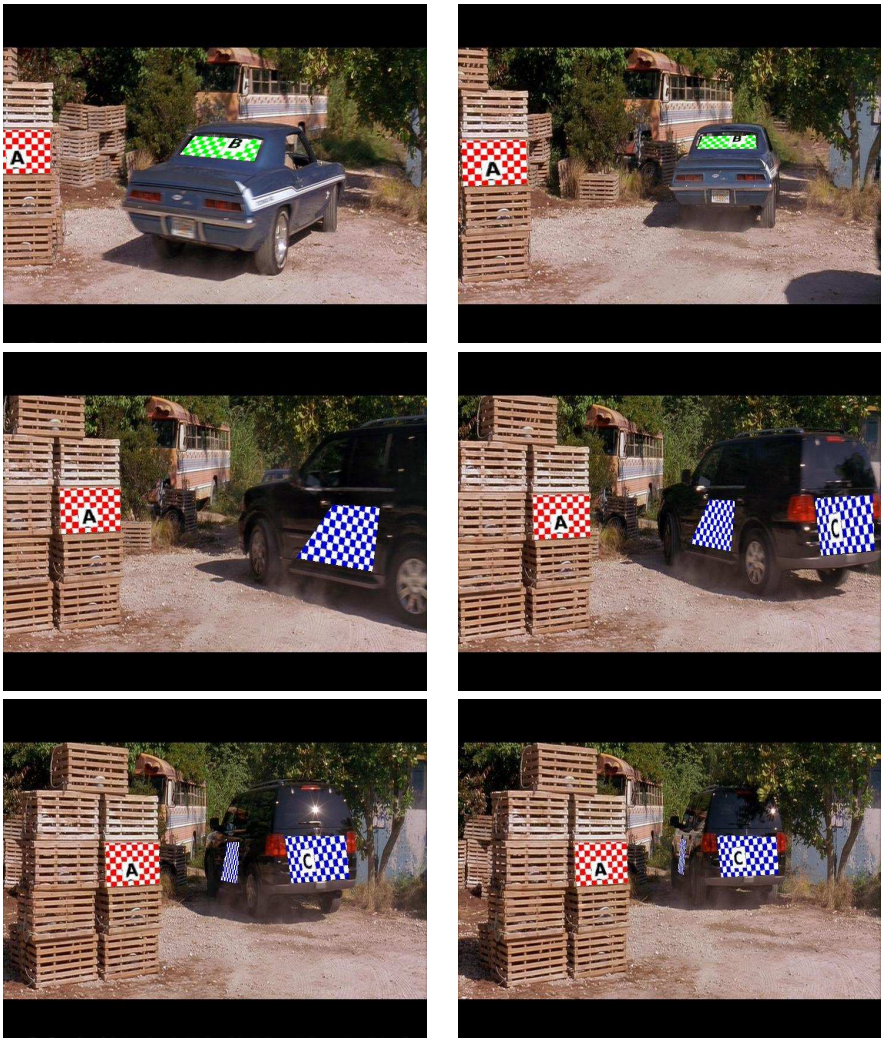


Figure 5.7: The augmented reality results for the car sequence.

The increased accuracy due to merging: One of the expected benefits of the merging operation is the expected increase in the accuracy of the SfM estimation process, as fewer parameters are estimated with the same amount of data. Without ground-truth it is very hard to demonstrate this property solidly but still some known geometric relationships in the scene can be helpful. In this experiment, we exploited known orthogonal lines by reconstructing them in 3D and finding the angle between them. A computed value that is closer to 90 degrees means a more accurate reconstruction.

Figure 5.8: Frame samples from the 175 frames long bus-stop sequence.

The algorithm was run on the bus sequence again but with merging disabled. Later, 4 line-pairs on the bus which are known to be orthogonal are reconstructed in 3D and the angle between them is computed. The error values, which are simply the absolute difference from 90 degrees are shown in table 5.1. The errors with the merging operation enabled are lower as expected.

Figure 5.9: Segmentation results for the bus-stop sequence. The new motion detection, merging and splitting operations are successfully carried out.

Table 5.1: *The error in degrees for the orthogonal lines that are reconstructed in 3D*

With merging	3	8	5	2
Without merging	7	15	9	7

5.8 Conclusion

Compared to the large volume of research on practical systems for static structure from motion, 3D reconstruction of dynamic scenes has so far been investigated mainly theoretically, for short, simple image sequences. However, a real video footage may contain quite challenging phenomena, such as appearing, disappearing, merging and splitting objects and short or corrupted feature tracks due to self-centered rotations or object's appearance characteristics. To achieve robust feature tracking is a key challenge to be able to reconstruct small foreground objects, which demands to do tracking in parallel with 3D segmentation and reconstruction. We have tried to come up with such a system, by identifying which components are required in a general and efficient SfM frame-

Figure 5.10: The augmented reality results for the bus sequence.

work for dynamic scenes. We have unveiled and discussed several subtle issues of large-scale dynamic SfM, and have proposed a novel framework to solve the task. Model selection techniques are deployed to detect changes in the number of independently moving objects. The advantages of such an approach have been demonstrated, and successful experiments have been presented.

The main structure of the proposed system is quite general and allows for different type of multibody SfM techniques (such as algebraic or subspace methods) to be used instead of model selection based recover and select approach

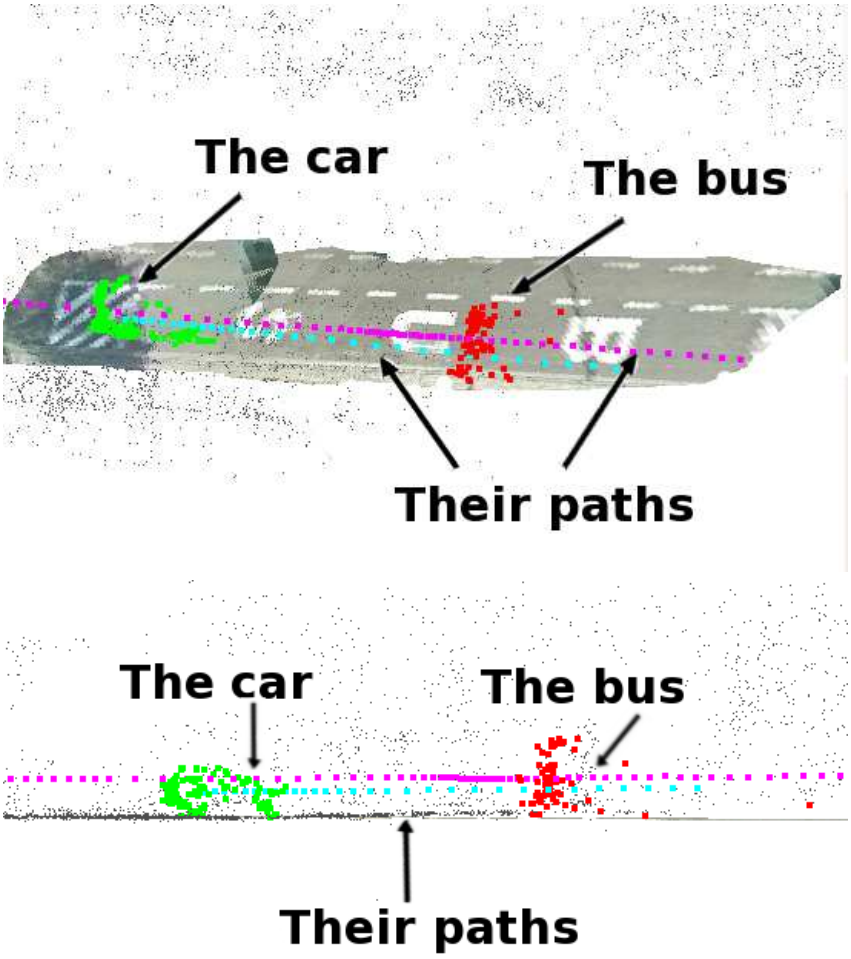


Figure 5.11: 3D reconstruction of the bus-stop sequence from a top view and a side view. The vehicles almost follow a line, proving the accuracy of the SfM estimation.

we adopted. Indeed, our system is open to experimentation in that aspect and only requirement is proper adaptation of such techniques in the light of our findings. The relative scale problem, which has played important role in previous chapters, has also got a different dimension in this chapter since the merger and split operations also fix the relative scales of moving objects.

Chapter 6

Conclusion

To generate the 3D model of a scene and to compute the camera parameters from pure image information has long been the holy grail of computer vision. A large number of papers has been published and a deep insight into the problem has been achieved, however the problem is far from being completely solved. This is partly due to the difficulty of the image registration problem which is a basic requirement in any SfM application and partly due to the degenerate configurations where either the camera does not have enough translational motion or the scene does not have enough 3D variation. Other problems are caused by various factors in a typical scene which complicate the image formation process, including lighting, reflective properties of the materials, the moving entities in the scene - rigid or non-rigid - and the complex interaction of all those factors. Those issues need to be simultaneously taken into account for a complete solution. However, estimating all of those factors from pure image information is an ill-posed problem, hence simplifying assumptions or prior knowledge on the scene are generally incorporated.

One such complicating factor that plagues typical SfM algorithms is the dynamic elements in the scene other than the background. Indeed, in a typical real world scene there are always dynamic elements and those elements are considered as outliers by canonical SfM algorithms at best. However those elements in general contain quite useful information, both for camera calibration and for scene interpretation. Given these observations, the work described in this dissertation is aimed at exploring problems in extending static scene algorithms to dynamic scenes and developing effective solutions. As a reasonable target, we considered only the sequences where the dynamic elements are rigid.

During the course of this PhD work, we initially concentrated on the subtle problem of relative scale ambiguity and proposed solutions for this relatively ignored but significant problem. However, in that initial work we have made three basic assumptions which constrain its applicability. Later, we developed new insights and techniques to lift those assumptions one by one. Other than the introduction section, each chapter in this dissertation is about lifting exactly

one of those assumptions and the presentation order of the related work is in direct correspondence with their chronological development.

The relative scale ambiguity stems from the fact that SfM for a static scene is only possible up to an unknown scale value. Given the video of a static scene, the 3D Euclidean relationships such as the angles between the lines, length ratios etc. are computable whereas the overall scale value can only be given as high level knowledge. In the case of dynamic scenes, the relative scales between the objects and they have to be adjusted properly otherwise the final reconstruction would possibly be quite unrealistic.

In chapter 2 we analyzed the problem and proposed two techniques to solve it. Both of the solutions are based on the idea that for the wrong relative scales the computed foreground motion will always have components from the camera translation; hence the correct relative scale is selected to be the one which removes the effect of the camera translation on the foreground motion. The *Independence* criterion searches for the object motion which is statistically the most independent from the camera motion. To measure this statistical independence, we introduced error measures based on mutual information and classical correlation. This technique is rather general; the downside is it requires many frames (a typical situation in any statistical approach). The other technique is called the *Non-accidentallness* criterion. It is based on the fact that certain geometric simplicities that can be found in the motion of a typical real world object would be destroyed under arbitrary relative scales due to additive camera motion. Hence, the detection of such simple patterns in the family of possible solutions hints at the correct relative scale. We introduced two such motion simplicity criteria, though there are many more that can be conceived. One is the planarity criterion, which is inspired by the fact that in real-life many objects move on planes and consequently the most planar path is selected among the possible solutions. The second one is the heading constraint, which is based on the fact that many objects have certain, locally fixed heading directions and this results in a coupling between the rotation and translation parameters of the object. The relative scale which satisfies this relationship best is selected.

One common assumption we made in the techniques we suggested for relative scale selection is the availability of the high level information concerning which segment is the background. This is highly related to the figure-ground segmentation problem in cognitive science and also in computer vision. There are typical cues that are used in this context, such as symmetry, texture deformation, size, T-Junctions etc. Taking a quite different path, we proposed solutions (chapter 3) based on the observation that the correct background selection would result in the simplest overall scene motion. The proposed solutions are built upon the aforementioned motion simplicity constraints. However in this problem, not only the correct relative scales are found but also the reconstruction that belongs to the static background is identified.

Although the experiments supported our analysis and the proposed solutions based on motion constraints, there are still open issues to be investigated.

One of them is how to select or combine among the different criteria in our repository, since an object may exhibit one or more behaviours at the same time, which may not be known a priori. Another one is the application of those criteria in a multi-object setting, since the type of objects motions are often interrelated, e.g.they move on the same plane or on the same road (hence the same heading direction). Those are questions which do not have immediate answers but can significantly be simplified. For example, SfM techniques are usually designed for certain contexts (such as traffic), where the types of he existing motions can be guessed reasonably well. Another interesting approach would be to incorporate the recently developed object categorization techniques to deduce the category of the foreground object, hence its motion type.

One fundamental aspect of both relative scale resolution and background recognition techniques discussed on the previous chapters is the assumption that the objects follow certain motion constraints while moving. However some counter examples are imaginable where the object motions are quite arbitrary so this assumption is not valid anymore. One way to solve the relative scales in such scenarios is the inclusion of a second, independently moving camera observing the same object. In this case we have two sequences which are reconstructed separately, hence two different relative scales to solve for. The basic idea of our solution (chapter 4) is the fact that since two cameras move independently, both object trajectories contain components from the camera trajectories under the wrong relative scales. Consequently, only at the correct relative scales the foreground motion parameters would be the same for both reconstructions. Hence, the existence of motion constraints is not required anymore.

However, in order to solve for the relative scale values in both view points, both the time synchronization and the similarity transformation between the reconstructions also need to be solved. This results in a more general registration technique where the image sequences are aligned not only by relative scales but also in space and time. The registration parameters are initialized by extending the classical hand-eye calibration techniques to account for relative scales. Later, the parameters are refined iteratively to come up with the most rigid combined foreground motion, since this has to be the case for a good overall registration. The biggest novelty in the proposed approach is that the solution is only based on the motion information of the foreground object, and it is not required to have common feature points between the view points. Such a technique has the potential to be useful in calibrating a multi-camera setting (static, dynamic or both) in a large territory where enough common feature points may not be available and it is hard to conceive a calibration pattern.

However there are still some issues that need to be investigated further. One of them is that since rotation parameters are initialized separately from the translation parameters (which must be a simultaneous solution ideally), the current solution requires at least two different rotation axes for the objects to exist which can be hard to find in cases where the motions are purely

planar on a single plane. However the exploitation of the translation parameters of the foreground object would stabilize the rotation estimation for the registration. The other open issues are the incorporation of multiple cameras and multiple objects, a new bundle adjustment routine that takes into account multiple motions and an automatic segment labeling procedure since currently corresponding moving objects are assumed to be known.

One fundamental limitation of the aforementioned techniques from the application perspective is the lack of a proper motion segmentation routine that would run as a preprocessing step. Indeed, in chapters 2 to 4, we assumed motion segmentation is done beforehand and applied a semi-manual technique to realize it. In order to overcome this problem, a simultaneous segmentation and reconstruction framework is developed in chapter 5.

3D motion segmentation of multiple rigidly moving objects is a problem that attracts increasingly more attention. Although most of the published papers give a good insight into the theoretical depth of the problem, a practical solution for long and realistic image sequences does not exist to the best of our knowledge. The basic aim of the proposed framework is to extend those recently developed multi-body SfM techniques to such hard and long sequences. Such a system has to take into account various complications in the scene, such as splitting, merging and newly arriving objects, short feature tracks, small numbers of features, etc. Those also have to be dealt with in a time efficient manner.

To overcome those problems we proposed a technique where tracking, segmentation, and reconstruction are done simultaneously. In contrast to many approaches which implement tracking as a separate process, our approach enjoys the robustness of guided tracking and consequently gives a very good performance on long and realistic image sequences. Such an approach requires an online detection of split and merging phenomena and those operations are implemented by various model selection techniques. We preferred to apply model selection techniques in an online and local fashion rather than as a global optimization, thus sparing significant computation time while keeping the results satisfactory. The experimental results justified our approach.

However, due to the complex nature of the system, there are several components that need to be investigated/improved further. Model selection routines are based on a recover-and-select approach where many hypotheses are generated and an optimal subset is selected. A smarter way of sampling needs to be implemented to achieve model selection faster and not to miss small moving objects. As typical in a practical SfM application, there are various thresholds in the system that work well in a certain range but still need to be set manually. Those parameters include time-window lengths used in the detection of new motions, merge and split phenomena. An automatic way of adjusting those parameters needs to be implemented.

One open issue is the comparison of our system, to similar other methods. Our technique is not the first SfM system which tries to achieve reconstruction of multi-body scenes. However, it aims at long and realistic sequences

with reasonable computation times and accounts for the changing temporal configuration in a scene. This stands in contrast to many other approaches where mathematical insight rather than practical application has been the focus. Consequently, a comparison is hard to achieve at this point

An interesting research avenue would be to apply sub-space based or algebraic multi-body SfM methods in the core split, merge and new motion instantiation operations of our general framework. In that regard, our framework will serve as a test-bed for various approaches.

The chapters 2, 3, 5 form the sub-components of a possible complete pipeline for multi-body SfM. However, other than a significant engineering effort, such a system requires development of various sub-components including a robust way of combining different motion constraints, a multi-body self-calibration approach (if the camera is not calibrated), a better exploitation of the features especially on the small foreground objects since they are more sensitive to noise, a dense reconstruction module that takes into account the moving objects, a bundle adjustment routine that is aware of multiple motions in a scene, and incorporation of more prior knowledge on both shape and motion for more robust reconstruction, again especially for the hard to reconstruct foreground objects.

Bibliography

[AEM98] Altunbasak Y., Erhan Eren P., Murat Tekalp A.: Region-based parametric motion segmentation using color information. *Graphical models and image processing: GMIP 60*, 1 (1998), 013–023.

[Aka74] Akaike H.: A new look at the statistical model identification. *IEEE Trans. on Automatic Control 19*, 6 (1974), 716–723.

[AP95] Azarbayejani A., Pentland A. P.: Recursive estimation of motion, structure and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence 17*, 6 (June 1995), 562–575.

[AS00] Avidan S., Shashua A.: Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2000), vol. 22, pp. 348–357.

[ASB94] Ayer S., Schroeter P., Birgun J.: Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *European Conference on Computer Vision* (1994), pp. 317–327.

[AW94] Adelson E. H., Wang J. Y. A.: Representing moving images with layers. *IEEE Transactions on Image Processing* (1994).

[BB91] Boult T., Brown L.: Factorization-based segmentation of motions. In *IEEE Workshop on Motion Understanding* (1991), pp. 179–186.

[BCC90] Broida T. J., Chandrashekhar S., Chellappa R.: Recursive 3-D motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems 26*, 4 (1990), 639–656.

[BHB00] Bregler C., Hertzmann A., Biermann H.: Recovering non-rigid 3d shape from image streams. In *Computer Vision and Pattern Recognition* (2000), pp. 690–696.

[BHS00] Bab-Hadiashar A., Suter D. (Eds.): *Data Segmentation and Model Selection for Computer Vision*. Springer-Verlag, 2000, ch. 6.

[Bla92] Black M.: Combining intensity and motion for incremental segmentation and tracking over long image sequences. In *European Conference on Computer Vision* (1992), pp. 485–493.

[Bra01] Brand M.: Morphable 3d models from video. In *Computer Vision and Pattern Recognition* (2001), vol. 1, pp. 456–463.

[BTZ96] Beardsley P., Torr P., Zisserman A.: 3D model acquisition from extended image sequences. In *European Conference on Computer Vision* (1996), vol. 1065, pp. 683–695.

[CB99] Csurka G., Bouthemy P.: Direct identification of moving objects and background from 2d motion models. In *International Conference on Computer Vision* (1999), pp. 566–571.

[CI01] Caspi Y., Irani M.: Alignment of non-overlapping sequences. In *ICCV* (2001), pp. 76–83.

[CI02] Caspi Y., Irani M.: Spatio-temporal alignment of sequences. *PAMI 24*, 11 (2002), 2690–2696.

[CK95] Costeira J., Kanade T.: A multi-body factorization method for motion analysis. In *International Conference on Computer Vision* (1995), pp. 1071–1076.

[Cor04] Cornelis K.: *From uncalibrated video to augmented reality, PhD Thesis*. 2004.

[CS04] C. Strecha R. F. . L. G.: Wide-baseline stereo from multiple views: A probabilistic account. In *Computer Vision and Pattern Recognition* (2004), pp. 552–559.

[CS06] C. Strecha R. F. . L. G.: Combined depth and outlier estimation in multi-view stereo. In *Computer Vision and Pattern Recognition* (2006), pp. 300–307.

[CT90] Caprile B., Torre V.: Using vanishing points for camera calibration. *International Journal of Computer Vision 4*, 2 (1990), 127–140.

[DA98] Debrunner C., Ahuja N.: Segmentation and factorization-based motion and structure estimation for long image sequences. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1998), vol. 20, pp. 206–211.

[Dan99] Daniilidis K.: Hand-eye calibration using dual quaternions. *Int. Journal on Robotics Research 18* (1999), 286–298.

[DC99] Dornaika F., Chung R.: Self-calibration of a stereo rig without stereo correspondence. In *Vision Interface* (1999), pp. 264–271.

[DLR77] Dempster A., Laird H., Rubin D.: Maximum likelihood from incomplete data via the em algorithm. *J. of Royal Statistical Society:Series B(Methodological) 39*, 1 (1977), 1–38.

[DNB04] D. Nister O. N., Bergen J.: Visual odometry. In *Computer Vision and Pattern Recognition* (2004), vol. 1, pp. 652–659.

[EM95] Edelman A., Murakami H.: Polynomial roots from companion matrix eigenvalues. *Mathematics of Computation Archive 64* (1995), 763–776.

[FB81] Fischler M., Bolles R.: Random sampling consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Communications of the Association for Computing Machinery 24* (1981), 381–395.

[FH86] Faugeras O., Hebert M.: The representation, recognition and locating of 3d objects. *Int. Journal on Robotics Research 5*, 3 (1986), 27–52.

[Fis36] Fisher R.: Uncertain inference. In *Proc. Amer. Acad. Arts and Sciences* (1936), vol. 71, pp. 245–258.

[FK98] Faugeras O., Keriven R.: Complete dense stereovision using level set methods. In *European Conference on Computer Vision* (1998), pp. 379–393.

[FL01] Faugeras O., Luong Q.-T.: *The geometry of multiple images*. the MIT Press, 2001.

[FLM92] Faugeras O., Luong Q.-T., Maybank S.: Camera self-calibration: Theory and experiments. In *European Conference on Computer Vision* (1992), vol. 588, Lecture Notes in Computer Science, pp. 321–334.

[FZ00] Fitzgibbon A., Zisserman A.: Multibody structure and motion: 3-d reconstruction of independently moving objects. In *European Conference on Computer Vision* (2000), vol. 1, pp. 891–906.

[Gea94] Gear C.: Feature grouping in moving objects. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects* (1994), pp. 214–219.

[GQZ05] G. Qian R. C., Zheng Q.: Bayesian algorithms for simultaneous structure from motion estimation of multiple independently moving objects. In *IEEE Transactions on Image Processing* (2005), vol. 15, pp. 94–109.

[Gui77] Guiasu S.: *Information Theory with Applications.* McGraw-Hill, 1977.

[Har94] Hartley R. I.: Self-calibration from multiple views with a rotating camera. In *European conference on Computer vision* (1994), vol. 1, pp. s: 471 – 478.

[HBG06] H. Bay T. T., Gool L.: Surf: Speeded up robust features. In *European Conference on Computer Vision* (2006).

[HCON94] Haralick R., C.N.Lee, Ottenberg K., Nolle M.: Review and analysis and solutions of the three points perspective pose estimation problem. In *International Journal Of Computer Vision* (1994), vol. 13, pp. 331–356.

[HD95] Horaud R., Dornaika F.: Hand-eye calibration. *International Journal of Robotics Research 14*, 3 (1995), 195–210.

[HK00] Han M., Kanade. T.: Reconstruction of a scene with multiple linearly moving objects. In *Computer Vision and Pattern Recognition* (2000), vol. 2, pp. 542–549.

[HK03] Han M., Kanade. T.: Multiple motion scene reconstruction from uncalibrated views. In *International conference on Computer Vision* (2003), vol. 25, pp. 884 – 894.

[HKO01] Hyvrinen A., Karhunen J., Oja E.: *Independent Component Analysis.* John Wiley and Sons, 2001.

[HZ00] Hartley R., Zisserman A.: *Multiple view geometry in computer vision.* Cambridge University Press, 2000.

[IAB*96] Irani M., Anandan P., Bergen J., Kumar R., Hsu S.: Efficient representation of video sequences and their representations. *Signal Processing:Image Comm 8*, 4 (1996), 327–351.

[Ich99] Ichimura N.: Motion segmentation based on factorization method and discriminant criterion. In *ICCV* (1999), pp. 600–605.

[Kan96] Kanatani K.: *Statistical optimization for geometric computation: theory and practice.* Elsevier Science, 1996.

[Kan01] Kanatani K.: Motion segmentation by subspace separation and model selection. In *International Conference on Computer Vision* (2001), vol. 2, pp. 301–306.

[Kan05] Kanade X. J. T.: Calibrated perspective reconstruction of deformable structures. In *International Conference on Computer Vision* (2005), vol. 2, pp. 1075 – 1082.

[KSW06] K. Schindler J. U., Wang H.: Perspective n-view multibody structure-and-motion through model selection. In *European Conference on Computer Vision* (2006), vol. 1, pp. 606–619.

[KZ02] Kolmogorov V., Zabih R.: Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision* (2002), vol. 3, pp. 82–96.

[LGB95] Leonardis A., Gupta A., Bajcsy R.: Segmentation of range images as the search for geometric parametric models. *International Journal of Computer Vision 14* (1995), 253–277.

[LH81] Longuet-Higgins H.: A computer algorithm for reconstructing a scene from two projections. In *Nature* (1981), vol. 293, pp. 133–135.

[Li01] Li S.: *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, 2001.

[LKSV07] Li T., Kallem V., Singaraju D., Vidal R.: Projective factorization of multiple rigid-body motions. In *CVPR* (2007).

[Low87] Lowe D.: 3d object recognition from single 2d images. *Artificial Intelligence 31* (1987), 355–295.

[Low04] Lowe D.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.

[LPB05] Langs G., Peloschek P., Bischof H.: Optimal sub-shape models by minimum description length. In *Computer Vision and Pattern Recognition* (2005), vol. 2, pp. 310–315.

[LV96] Li M., Vatanyi P.: *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag, 1996.

[Mac67] MacQueen J.: Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability* (1967).

[MMI02] M. Machline L. Z.-M., Irani M.: Multi-body segmentation: Revisiting motion consistency. In *ECCV Workshop on Vision and Modeling of Dynamic Scenes* (2002).

[Nis03] Nister D.: An efficient solution to the five-point relative pose problem. In *Computer Vision and Pattern Recognition* (2003), vol. 2, pp. 195–202.

[O05] Ozden K. E., . L. V. G.: Background recognition in dynamic scenes with motion constraints. In *International Conference on Computer Vision and Pattern Recognition* (2005), pp. 250–255.

[OCE04] Ozden K. E., Cornelis K., Eycken L. V., . L. V. G.: Reconstructing 3d independent motions using non-accidentalness. In *CVPR* (2004), vol. 1, pp. 819–825.

[OCG06] Ozden K., Cornelis K., Gool L. V.: Space-time-scale registration of dynamic scene reconstructions. In *ECCV* (2006), pp. 173–185.

[OCVV04] Ozden E., Cornelis K., Van Eycken L., Van Gool L.: Reconstructing 3D trajectories of independently moving objects using generic constraints. *CVIU 2004, Special issue on 3D model-based and image-based 3D scene representation for interactive visualization 96*, 3 (December 2004).

[OSG07] Ozden K., Schindler K., Gool L. V.: Simultaneous segmentation and 3d reconstruction of monocular image sequences. In *ICCV* (2007).

[PD91] Pentland A., Darrel A.: *Cooperative robust estimation using layers of support.* Tech. Rep. T.R. 163, MIT Media Lab., 1991.

[PGR99] Pao H., Geiger D., Rubin N.: Measuring convexity for figure/ground separation. In *International Conference on Computer Vision* (1999), pp. 948–955.

[PK97] Poelman C. J., Kanade T.: A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 3 (1997), 206–218.

[PVG02] Pollefeys M., Verbiest F., Gool L. V.: Surviving dominant planes in uncalibrated structure and motion recovery. In *ECCV* (2002), vol. 2, pp. 837–851.

[PVV*04] Pollefeys M., Van Gool L., Vergauwen M., Verbiest F., Cornelis K., Tops J., Koch R.: Visual modeling with a hand-held camera. *IJCV 59*, 3 (2004).

[QC01] Qian G., Chellappa R.: Structure from motion using sequential monte carlo methods. In *International Conference on Computer Vision* (2001), vol. 2, pp. 614–621.

[Rip96] Ripley B. D.: *Pattern Recognition and Neural Networks.* Cambridge University Press, 1996.

[Ris78] Rissanen J.: Modeling by shortest data description. *Automatica 14* (1978), 465–471.

[RLSP06] Rothganger F., Lazebnik S., Schmid C., Ponce J.: 3d object modeling and recognition using affine-invariant patches and multi-view spatial constraints. *International Journal of Computer Vision 66*, 3 (2006).

[RLSP07] Rothganger F., Lazebnik S., Schmid C., Ponce J.: Segmenting, modeling, and matching video clips containing multiple moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29* (2007), 477–491.

[RNS96] Rubin N., Nakayama K., Shapley R.: Enhanced perception of illusory contours in the lower versus upper visual hemifields. *Science 271* (1996), 651–653.

[Rub21] Rubin E.: *Visuell wahrgenommene Figuren.* Copenhagen: Gyldendals, 1921.

[SA89] Shiu Y., Ahmad S.: Calibration of wrist mounted robotic sensors by solving homogenous transform equations of the form ax=xb. *IEEE J. Robot. Automation 5*, 1 (1989), 16–19.

[SA96] Sawhney H., Ayer S.: Compact representation of videos through dominant and multiple motion estimation. *IEEE Pattern Analysis and Machine Intelligence 8*, 4 (1996), 814–830.

[Sch78] Schwarz G.: Estimating dimension of a model. *Ann. Stat. 6* (1978), 461–464.

[SCM*03] Slabaugh G., Culbertson W., Malzbender T., Stevens M., Schafer R.: Methods for volumetric reconstruction of visual scenes. In *International Journal of Computer Vision* (2003), pp. 179 – 199.

[SFZ06] Sivic J., F.Schaffalitzky, Zisserman A.: Object level grouping for video shots. *International Journal of Computer Vision 67* (2006), 189–210.

[Sha48] Shannon C. E.: A mathematical theory of communication. *Bell System Technical Journal 27* (1948), 379–423 and 623–656.

[SL95] Stricker M., Leonardis A.: Figure-ground segmentation using tabu search. In *Proc. of the IEEE Intern. Symposium on Computer Vision* (1995), pp. 605–61.

[SM98] Shi J., Malik J.: Motion segmentation and tracking using normalized cuts. In *ICCV* (1998), pp. 1154–1160.

[Smi01] Smith P.: *Edge-Based Motion Segmentation.* PhD thesis, Univ. of Cambridge, 2001.

[SP94] Soatto S., Perona P.: Three dimensional transparent structure segmentation and multiple 3d motion estimation from monocular perspective image sequences. In *IEEE Workshop on Motion of Nonrigid and Articulated Objects* (1994), pp. 228–235.

[SP04] Sinha S., Pollefeys M.: Synchronization and calibration of camera networks from silhouettes. In *ICPR* (2004), vol. 1, pp. 116–119.

[SPM04] Sinha S., Pollefeys M., McMillan L.: Camera network calibration from dynamic silhouettes. In *CVPR* (2004), vol. 1, pp. 195–202.

[SS02] Scharstein D., Szeliski R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *International Journal of Computer Vision* (2002), pp. 7–42.

[SS06] Schindler K., Suter D.: Two-view multibody structure-and-motion with outliers through model selection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2006), vol. 28, pp. 983 – 995.

[ST96] Sturm P., Triggs B.: A factorization based algorithm for multi-image projective structure and motion. In *European Conference on Computer Vision* (April 1996), vol. 1065, Lecture Notes in Computer Science, Springer Verlag, pp. 709–720.

[Stu02] Sturm P.: Structure and motion for dynamic scenes:the case of points moving in planes. In *European Conference on Computer Vision* (2002), vol. 2, pp. 867–882.

[SW00] Shashua A., Wolf L.: Homography tensors: On algebraic entities that represent three views of static or moving planar points. In *ECCV* (2000), vol. 1, pp. 507 – 521.

[SZ03] Sivic J., Zisserman A.: Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision* (2003), pp. 1470–1477.

[TDP94] T. Darrel A. A., Pentland P.: *Segmentation of Rigidly Moving Objects using Multiple Kalman Filters*. Tech. rep., MIT Media Lab., 1994.

[TK91] Tomasi C., Kanade T.: *Detection and tracking of point features*. Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, April 1991.

[TK92] Tomasi C., Kanade T.: Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision 9*, 2 (1992), 137–154.

[TM93] Torr P., Murray D.: Outlier detection and motion segmentation. In *Proc. SPIE Sensor Fusion Conference* (1993), vol. 6, pp. 432–443.

[Tor98] Torr P. H. S.: Geometric motion segmentation and model selection. In *Phil. Trans. Royal Society of London* (1998), vol. 356, pp. 1321–1340.

[Tor02] Torr P. H. S.: Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision 50*, 1 (2002), 35–61.

[Tri97] Triggs B.: Autocalibration and the absolute quadric. In *Computer Vision and Pattern Recognition* (June 1997), pp. 609–614.

[Tsa87] Tsai R. Y.: A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation 4*, 3 (August 1987), 323–344.

[TV99] Tuytelaars T., Van Gool L.: Content-based image retrieval based on local affinely invariant regions. In *International Conference on Visual Information Systems* (June 1999), pp. 493–500.

[TV07] Tron R., Vidal R.: A benchmark for the comparison of 3D motion segmentation algorithms. In *CVPR* (2007).

[VH04] Vidal R., Hartley R.: Motion segmentation with missing data using powerfactorization and gpca. In *Computer Vision and Pattern Recognition* (2004), pp. 310–316.

[VL97] Vasconcelos N., Lippman A.: Empirical bayesian em-based motion segmentation. In *CVPR* (1997), pp. 527–532.

[VM04] Vidal R., Ma Y.: A unified algebraic approach to 2-d and 3-d motion segmentation. In *European Conference on Computer Vision* (2004), vol. 1, pp. 1–15.

[VMP04] Vidal R., Ma Y., Piazzi J.: A new gpca algorithm for clustering subspaces by fitting, differentiating and dividing polynomials. In *CVPR* (2004), vol. 1, pp. 510–517.

[VMS03] Vidal R., Ma Y., Sastry S.: Generalized principal component analysis (gpca). In *CVPR* (2003), vol. 1, pp. 621–628.

[VSMS02a] Vidal R., Soatto S., Ma Y., Sastry S.: *A factorization method for 3D multi-body motion estimation and segmentation.* Tech. rep., 2002.

[VSMS02b] Vidal R., Soatto S., Ma Y., Sastry S.: Segmentation of dynamic scenes from the multibody fundamental matrix. In *ECCV Workshop on Vision and Modeling of Dynamic Scenes* (2002).

[WA96] Weiss Y., Adelson E.: A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *International Conference on Computer Vision and Pattern Recognition* (1996), pp. 321–326.

[WS01a] Wolf L., Shashua A.: On projection matrices p^k → p^2, k=3,...,6, and their applications in computer vision. In *International Conference on Computer Vision* (2001), pp. 412–419.

[WS01b] Wolf L., Shashua. A.: Two-body segmentation from two perspective views. In *International Conference on Computer Vision* (2001), pp. 263–270.

[WS04] Wang H., Suter D.: A very robust estimator for model fitting and range image segmentation. *International Journal of Computer Vision 59*, 2 (2004), 139 – 166.

[WZ02a] Wolf L., Zomet A.: Correspondence-free synchronization and reconstruction in a non-rigid scene. In *ECCV Workshop on Vision and Modeling of Dynamic Scenes* (2002).

[WZ02b] Wolf L., Zomet A.: Sequence-to-sequence self calibration. In *ECCV* (2002), vol. 2, pp. 370–382.

[XCK04] Xiao J., Chai J., Kanade T.: A closed-form solution to non-rigid shape and motion recovery. In *European Conference on Computer Vision* (2004).

[YP05] Yan J., Pollefeys M.: A factorization-based approach to articulated motion recovery. In *CVPR* (2005), vol. 2, pp. 815–821.

[YP06a] Yan J., Pollefeys M.: Automatic kinematic chain building from feature trajectories of articulated objects. In *CVPR* (2006), vol. 1, pp. 712–719.

[YP06b] Yan J., Pollefeys M.: A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV* (2006), vol. 4, pp. 94–106.

[Zha00] Zhang Z.: A flexible new technique for camera calibration. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2000), vol. 22, pp. 1330–1334.