SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

Local and Global Methods for Registering 2D Image Sets and 3D Point Clouds

DANDA PANI PAUDEL

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

THÈSE présentée par

DANDA PANI PAUDEL

pour obtenir le

Grade de Docteur de

l'Université de Bourgogne

Spécialité : **Instrumentation et Informatique de l'image**

Local and Global Methods for Registering 2D Image Sets and 3D Point Clouds

Soutenue publiquement le 10 Décembre 2015 devant le Jury composé de :

PETER STURM	Rapporteur	INRIA Grenoble, France
ANDREA FUSIELLO	Rapporteur	Università degli Studi di Udine, Italie
ADRIEN BARTOLI	Examinateur	Université d'Auvergne, France
DAVID FOFI	Examinateur	Université de Bourgogne, France
CÉDRIC DEMONCEAUX	Directeur de thèse	Université de Bourgogne, France
PASCAL VASSEUR	Co-encadrant	Université de Rouen, France
ADLANE HABED	Co-encadrant	Université de Strasbourg, France

Dedicated to my parents.

(आमा-बुवाप्रति समर्पित)

ACKNOWLEDGMENTS

"We are like dwarfs on the shoulders of giants, so that we can see more than they, and things at a greater distance, not by virtue of any sharpness of sight on our part, or any physical distinction, but because we are carried high and raised up by their giant size."

- Bernard de Chartres, 12^{th} *century AD*

I would like to express my sincere gratitude to my supervisors, Cédric Demonceaux, Adlane Habed, and Pascal Vasseur, who taught me the most crucial lesson of asking the right questions that define the problems well and lead to a clear vision of the issues involved. In addition to that, they also helped me through the tough questions, the answers of which we have been looking in this thesis. I have always admired the passion and productive discussions with them related to academics, research, or life matters.

My thanks to Cédric are endless – not only my research inspiration did come form him, but most of the contributions presented in this thesis are the consequence of the freedom that he has allowed me in conducting research on the topic and in choosing the workplace. On the other hand, my interaction with Adlane remained very special. Among many aspects I admire of him, I cannot forget to mention the hours long discussions on every aspect ; may that be research, writing, or thinking out of the box. His attitude towards perfection – pushing all possible boundaries – has made seemingly impossible achievements plausible.

I would also like to thank the jury members for their kindness of considering to participate in my Ph.D. defense committee. Many thanks to Peter Sturm and Andrea Fusiello for carefully reading the thesis and their valuable remarks, Adrien Bartoli and David Fofi for investing their precious time on playing the role of the president of defense and the thesis examiner respectively. Their involvement has undoubtedly widened the understanding and added new perspectives on the research issues appeared during this thesis.

This research was funded by the International Project NRF-ANR DrAACaR : ANR-11-

ISO3-0003, the Regional Council of Burgundy and European Regional Development Fund. Not only this funding supported me financially, but created the opportunities to travel for conferences, summer school, and partner universities. My visit to Korean Advance Institute of Technology (KAIST) as a visiting scholar in 2013, although only for a month, remained very memorable and productive. During my stay in KAIST, I was fortunate to have insightful discussions with Prof. In So Kweon. Furthermore, I would also like to thank all my colleagues at RCV laboratory, KAIST, who made my stay in Korea very pleasant and unforgettable experience.

Besides the realm of research, I also had opportunities to interact with graduate students as a teacher, a tutor, or a thesis co-supervisor during these last three years. I really enjoyed the time working with Deepak and Ran as their thesis co-supervisor. Teaching MsCV/Vibot students was no less fun. This thesis has direct or indirect inputs of the ideas that grew from the discussions I had with my students in different occasions. I would like to thank the faculty members in Condorcet (especially Fabrice, Yohan, Désiré, and Ralph) for creating these opportunities.

I have received tremendous support and kindness from the MsCV programme team (especially Alice, David, and Herma) in Le Creusot since the first day of my Master's study, which continued throughout my Ph.D. with the involvement of Nathalie Ravey – many thanks to them. It would have been impossible to manage my way through the subtleties of the French bureaucracy without their help.

I am very thankful to all my friends and colleagues in Le Cruesot and Strasbourg for the support they have given me during the past few years. To name a few : Ran, Deepak, Suman, Lijia, Maya, Ajad, Chinmay, Priyanka, Natalia, Qinglin, François, Cansen, Guillaume, Mojdeh, Joan, Mazen, Vincent, Muhammad, Sharib, Abinash, Bishesh, Abhilash, Kedir, Alper, Ozan, Sam, Rienier, Jeffrey, Andreea, Darshan, Pablo, Taman, Andru, Dadhichi, Chakon, Sai, Juan, Jilliam, Sebastian, Vikram, Maximillian, Fanny, Rahim, Xavier, Lionel ..., the list is long and grows longer everytime. To say the least, being together with them has surely made my time a lot more fun and fruitful than it would have been otherwise.

Finally and above all, I would like to thank my family for giving me the support and kindness and making everything I have done possible. Their love and continued kindness, despite the distance, have always provided me the reasons to persevere.

CONTENTS

1

INTRODUCTION

"The three R's of Vision : Recognition, Reconstruction, Reorganization."

- Jitendra Malik, *UC Berkeley*

1.1/ CONTEXT AND MOTIVATION

A grand goal of Computer Vision is to provide computers with the ability to recover the three-dimensional structure of a scene and to understand its content while relying on visual data. Such attributes would allow machines to recognize objects, identify people and actions, build environment maps for navigation, and deduce semantic relationships between objects in the scene. Traditionally, the visual information was limited to 2D images mainly obtained from conventional imaging sensors such as consumer cameras. The recent surge in consumer and high-end visual sensors of various modalities, along with powerful computer hardware, makes it viable in the present time to consider acquiring and jointly processing large amounts of data obtained from different sources such as 3D scanners, color, thermal or multispectral images. In particular, 2D consumer cameras are now capable of capturing high quality texture information while 3D sensors, such as lidar sensors and RGB-D cameras, provide 3D range data of various resolutions and quality in the form of 3D point coordinates. Jointly acquiring and processing 2D and 3D data may undoubtedly yield exciting and potentially revolutionary applications beyond the realm of Computer Vision. Indeed, the measurements obtained from such modalities are complementary to one another and therefore may leverage high quality 3D scene modeling with texture mapping as well as scene understanding algorithms. Such a fusion may turn also useful for change detection, filling scene gaps, camera pose correction or refinement, and

visual odometry.

A common practice for acquiring 2D and 3D visual information involves the use of a "packaged solution" consisting of a rigidly coupled pair of fully calibrated sensors. In its simplest form, such solution comprises a 2D camera and a 3D sensor capable of carrying out synchronous acquisitions. Maintaining such setup may turn out to be both tedious and difficult due to differences in acquisition frequencies, possible changes in the calibration parameters and the presence of a dedicated hardware required for synchronization. Note that good acquisition conditions, for example in terms of viewing angle and lighting, for capturing quality measurements with one modality may not be compatible with those necessary for obtaining decent measurements with the other. In this regard, the ability of acquiring 2D images and 3D data independently from one another adds a great deal of flexibility and freedom in order to achieve higher quality modeling results. Furthermore, one may be interested in processing corpora of data in which 2D images and 3D scans have been obtained independently, possibly at different times or because of application-specific restrictions. However, when images and scans are acquired independently, the problem of registering the data obtained from both modalities becomes very challenging and difficult to solve. The present thesis is precisely concerned with the automatic registration of independently captured data emanating from 2D and 3D cameras.

1.2/ SCOPE AND CHALLENGES

In this thesis we address the problem of registering 2D image sets and a 3D point cloud, i.e. a set of 3D point coordinates, of the same rigid scene. By registering such data, we mean finding "enough" correspondences between 3D points and image pixels and retrieving the unknown rigid transformation relating the 3D sensor (or reference frame) and each 2D camera. In particular, we have investigated solutions to the following three cases in which one or more requirements of packaged solutions, namely calibration, rigid coupling or synchronous acquisitions, are dropped :

1. **Fully calibrated coupled acquisition set-up** : in this case, a rigidly attached set of sensors comprised of 2D cameras and a 3D sensor is used. Such acquisition set-up is assumed to be fully calibrated, i.e. the 2D cameras are internally calibrated and the rigid transformation between the 3D sensor and cameras are known and

unchanged at all time. However, unlike in “packaged solutions”, the acquisitions obtained from individual sensors are allowed to be asynchronous.

2. **Uncoupled acquisition set-up with internally calibrated 2D cameras** : a 3D sensor and two or more 2D cameras (or a moving camera) are independently employed to capture the scene. The 2D cameras are again assumed to be internally calibrated but the rigid motion relating the 3D sensor and any given 2D camera is unknown and possibly changing (e.g. cameras in motion). Acquisitions are allowed to be synchronous and asynchronous.
3. **Uncoupled acquisition set-up with uncalibrated 2D cameras** : a 3D sensor and two or more 2D cameras (or a moving camera) independently capture the scene. The rigid motion relating the 3D sensor and any given 2D camera is unknown and possibly changing. However, this time, the 2D cameras are assumed to be uncalibrated. Again, acquisitions may be either synchronous or asynchronous.

These three cases are further discussed below.

FULLY CALIBRATED COUPLED ACQUISITION SET-UP

When a fully calibrated coupled acquisition set-up is used to capture a scene, the 3D and 2D measurements may or may not be acquired synchronously. In the case in which the acquisition set-up is not moving, synchronization of all sensors is irrelevant so long as the scene is static : the 3D-2D relationship between 3D points and pixels can be established via the known rigid transformation relating the two considered sensors. However, when the acquisition set-up is moving, all sensors require perfect synchronization : any motion of the set-up between acquisitions from a 2D camera and a 3D sensor undermines the use of the sensors’ rigid transformation as a basis for establishing 2D and 3D correspondences. Furthermore, most common synchronization methods use time-stamps from the synchronized clocks of each sensor. The closest frames in time are considered to be synchronous. This solution is not accurate enough for applications that involve fast motion, especially when the acquisition frame-rate of any sensor is considerably low. Other systems use dedicated hardware that synchronizes the frames by monitoring the sensors’ readout time. When the readout times of sensors vary significantly, as in 2D cameras and 3D sensors, accurate synchronization requires more advanced hardware. Therefore, low-cost fast moving systems very often suffer from synchronization issues, particularly when

the mechanical parts – as in Lidar sensors – are involved during acquisition. Note that for fast moving cameras, a small synchronization delay may lead to a big error in pose.

In view of the above synchronization issues, as part of the present thesis, we investigate the problem of registering 3D data and 2D images emanating from a moving fully calibrated, yet asynchronous, coupled acquisition set-up. Doing so offers the potential to avoid the burden of resorting to extra hardware altogether. We regard this problem as that of registering a set of coupled acquisition set-ups, each with vaguely known or inaccurate rigid relationships between its internally calibrated 2D cameras and its 3D sensor. The interest in addressing this problem manifests in accurately estimating the pose of 2D cameras in the 3D camera coordinate frame or vice versa. Note that, because the acquisition set-up is fully calibrated, the initial relationships between the 2D cameras and the 3D sensor are known. However, the exact pose information is lost due to the synchronization delay and our goal is to recover the pose corresponding to the delayed acquisition.

Another aspect that we are concerned with in this work is to accurately estimate the camera motion using both 2D and 3D information. Estimating camera motion from visual data is also known as visual odometry. Once the asynchronous 3D sensor and cameras are registered, they can be considered as synchronized. Fusing information from both modalities for accurate motion estimation becomes highly desirable. However, synchronizing the acquisition set-up is not sufficient to estimate the motion of the acquisition set-up and inter-frame motion estimation is required.

UNCOUPLED ACQUISITION SET-UP WITH INTERNALLY CALIBRATED 2D CAMERAS

Some application-specific requirements prohibit the rigid coupling constraint of the 2D cameras and the 3D sensor. This situation may arise either because of the difference in suitable acquisition conditions or the nature of the problem itself. Several applications concerned with uncoupled camera set-ups are discussed below.

Applications such as photo-realistic rendering demand very high quality of both 2D and 3D data. Usually, the acquisition of high quality 3D data is carried out by using one or multiple slow moving 3D cameras over a long duration of time. However, when illumination is changing, typical to outdoor scenes, investing the same amount of time for acquiring a 2D image may intricate a problem. Under such conditions, 2D images need to be rather

captured using a fast (or at least faster than the 3D sensor) moving camera covering the whole scene. If rendering under different lighting conditions is required, it is recommended to use one set of images for each lighting condition. This happens mainly because images captured under different illumination conditions can deteriorate the photo-realistic rendering effect. Moreover, this also creates difficulty in establishing correspondences across images. Therefore, for such applications, 2D and 3D cameras are better kept uncoupled.

Acquiring shape and texture information in parallel using independently moving 2D cameras and a 3D sensor (such as a 2D camera on a drone and a 3D sensor on a terrestrial vehicle) may also prohibit us from establishing the direct relationship between 2D and 3D acquisitions. The main difficulty in registering 2D and 3D acquisitions lies in establishing reliable correspondences between 2D and 3D data. This process is highly undermined by unreliable 3D feature descriptors, especially when the 3D data lacks the texture information or when the 2D images and the 3D data are captured under different lighting conditions. Additionally, in the case of uncoupled cameras, it is very difficult to find a good 2D camera pose initialization with respect to a 3D sensor such that locally convergent registration methods – typically Iterative Closest Point (ICP) or its variants – could provide satisfactory results. Consequently, in the absence of feature correspondences, the problem of registering uncoupled 2D cameras and a 3D sensor demands devising methods that are insensitive to local minima traps. In other words, we are interested in registering a set of images, obtained from internally calibrated, possibly moving, 2D cameras and a 3D point cloud originating from a 3D sensor while all cameras and the sensors are uncoupled. Note that in such set-up, whether the acquisitions are synchronous or asynchronous is irrelevant since no initial information on sensors' relationships are known.

UNCOUPLED ACQUISITION SET-UP WITH UNCALIBRATED 2D CAMERAS

We consider the case in which a scene is captured by a 3D sensor and a set of possibly moving 2D cameras. All sensors, whether 2D or 3D, are uncoupled. Observe that if one manages to register a 2D image and the 3D scene, then the intrinsic parameters of the 2D camera at hand can be recovered. This is very much as if the 2D camera was calibrated using a known calibration pattern. This observation indicates that it may not be necessary to use internally calibrated 2D cameras after all in order to register 2D and 3D data. Doing without internal calibration certainly provides an additional unprecedented level

of flexibility and freedom as the cameras are allowed to vary their intrinsic parameters through zooming and focusing to capture better quality images.

When cameras undergo changes in their intrinsic parameters, the registration between 2D images and 3D data faces extra challenges. Local refinement methods such as Bundle Adjustment (BA) may correct these changes provided these are small. However, there is no indication whatsoever in the set-up under consideration that changes in the internal calibration of any camera ought to be small. There remains the improbable option of a pattern-based re-calibration of the cameras which is clearly impractical along with that of camera autocalibration that, due to numerous critical motions rendering such task inapplicable, cannot unfortunately be relied upon. The registration problem we address in this thesis is that of uncoupled 2D cameras and 3D sensors with altogether unknown internal calibration of all 2D cameras. Again, whether the sensors are synchronous or asynchronous has no relevance in this case.

1.3/ CONTRIBUTIONS

In this thesis, several contributions related to all three aforementioned acquisition set-ups are proposed. These contributions appeared in [1, 2, 3, 4, 5, 6]. In the following, we provide a brief summary of contributions for each of the three cases separately.

FULLY CALIBRATED COUPLED ACQUISITION SET-UP

In the case of an asynchronous fully calibrated coupled acquisition set-up, we propose a joint synchronization and motion refinement framework based on alternating minimization. The synchronization part uses the coupled set-up's known rigid transformation relating 2D and 3D sensors for initialization whereas the motion parameters are initialized using up to scale rigid motion computed from 2D-to-2D correspondences across images. The proposed method does not require an accurate set of 2D-to-3D correspondences, handles occlusions, and works for partially known scenes. It goes without saying that if the data are already synchronized (or acquired by a synchronized setup), our method refines only the motion parameters. In such cases, initialization of motion parameters is obtained from approximate 2D-to-3D correspondences. Our contributions with regard to

this acquisition set-up have been published as follows :

— our work dealing with a synchronous acquisition set-up was published in [1] : *Danda Pani Paudel, Cédric Demonceaux, Adlane Habed, Pascal Vasseur, and In So Kweon. 2d-3d camera fusion for visual odometry in outdoor environments. In the Porceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), pages 157-162. IEEE, 2014* ($\sim$ 45% acceptance rate) ;

— the case of an asynchronous acquisition set-up was reported in our papers [2, 3] :

- *Danda Pani Paudel, Cédric Demonceaux, Adlane Habed, and Pascal Vasseur. Estimation de la pose d'une caméra dans un environnement connu à partir d'un recalage 2D-3D. In Reconnaissance de Formes et Intelligence Artificielle (RFIA).* 2014.
- *Danda Pani Paudel, Cédric Demonceaux, Adlane Habed, and Pascal Vasseur. Localization of 2d cameras in a known environment using direct 2d-3d registration. In the Proceedings of the 22nd IAPR International Conference on Pattern Recognition (IAPR ICPR 2014), pages 196-201.* IEEE, 2014.
  The paper was orally presented at the conference ($\sim$ 15% acceptance rate for oral papers).

UNCOUPLED ACQUISITION SET-UP WITH INTERNALLY CALIBRATED 2D CAMERAS

We propose a deterministic globally optimal method for registering a set of images and associated 3D data under the assumption that the scene can be well-represented by planes or planar patches. The planar segmentation assumption is particularly valid when dealing with man-made environments, including (but not limited to) Manhattan World, urban and indoor scenes that are abundant with planes. This entails that the 3D point set is sufficiently dense to be segmented efficiently and robustly into planes. We also assume that sparse pixel correspondences across images are provided and reconstructed via a Structure-from-Motion (SfM) approach up to an unknown scale using the internally calibrated cameras. In particular, the proposed method optimally aligns planes extracted from the 3D point-set with the SfM-induced 3D points. We use a robust inlier set maximization approach within a branch-and-bound framework to explore the registration parameters' space. In this regard, our contribution is threefold : (a) a novel formulation of the point-to-plane correspondence problem using polynomial sum-of-squares optimization theory ; (b)

incorporating constraints that are specific to the problem (plane visibility, camera position, etc.) so as to reduce the search space ; and (c) globally optimal point-to-plane inlier set maximization with or without putative correspondences.

This work has been accepted for publication in - and oral presentation at - the IEEE/CVF ICCV conference [4] :

— *Danda Pani Paudel, Adlane Habed, Cédric Demonceaux, and Pascal Vasseur. Robust and optimal sum-of-squares-based point-to-plane registration of image sets and structured scenes. In Proceedings of the IEEE Conference on Computer Vision (IEEE/CVF ICCV 2015), 2015.*(∼ 3% acceptance rate for oral papers)

UNCOUPLED ACQUISITION SET-UP WITH UNCALIBRATED 2D CAMERAS

We also propose a method for the direct 2D-3D registration of a set of uncalibrated images of a scene and the latter's Euclidean 3D point-set. Our method assumes that an unknown subset of Euclidean scene points have their projections detected and matched across the images. The correspondences obtained from the image-based matching are used to obtain a consistent set of camera matrices up to a projective ambiguity and expressed in some arbitrarily chosen projective frame. For the sake of direct 2D-3D registration, we propose a Linear Matrix Inequality (LMI) framework that facilitates the process of estimating the projective homography relating the cameras' projective frame and the Euclidean scene coordinate frame. This process requires neither 2D-to-3D correspondences nor explicit image-based reconstruction of the points. This LMI framework allows one to establish reconstruction-free putative correspondences between 2D matched points and a 3D volume represented by a convex polyhedron (a cuboid in all our experiments, also referred to as a "Box"). Furthermore, an extra set of LMIs are also derived to check the reconstruction-free so-called cheirality conditions.

Based on the proposed LMI framework, we develop two algorithms for 2D-3D registration that employ a Branch-and-Prune(BnP) paradigm. In our first algorithm, all points detected and matched across images are assumed to have a 3D counterpart in the 3D point could. Branching is carried out in the scene's space and the LMI conditions allow to associate 2D points to non-empty 3D boxes. This is done while taking the structure of the scene into account. Our second algorithm is more concerned with robustness : it allows image

points not to have counterparts in the 3D point cloud as well as mismatches across the images. This algorithm branches on the homography parameters' space and allows a predefined fraction of 2D points to correspond to empty boxes in the scene. Both algorithms provide the guarantee of convergence to a globally optimal solution under some mild initial bounding conditions.

This work has been published in IEEE/CVF CVPR 2015 [5] and has been filed for US patent [6] :

- *Danda Pani Paudel, Adlane Habed, Cédric Demonceaux, and Pascal Vasseur. LMI-based 2d-3d registration : From uncalibrated images to euclidean scene. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4494-4502, 2015.* (acceptance rate ~ 28%)
- *Danda Pani Paudel, Adlane Habed, Cédric Demonceaux, and Pascal Vasseur. Vasseur. Method for free Registration of a Euclidean 3-Dimensional Scanned Scene and Image Sets. US Provisional Patent, No 62/165,433, United States, May 22, 2015.*

1.4/ ORGANIZATION

This thesis is divided into seven different chapters. Chapter 2 introduces the geometric concepts and existing 2D-3D registration approaches. In Chapter 3, we discuss the local and global optimization techniques. This chapter also provides different tools that we use to devise our registration methods. The registration methods for fully calibrated coupled acquisition set-ups have been proposed in Chapter 4. Similarly, the registration methods for acquisition set-ups with internally calibrated and uncalibrated 2D cameras are proposed in Chapter 5 and Chapter 6, respectively. Finally, Chapter 7 concludes our work, and also summarizes its future prospective.

2

The Geometry of Image Sets and 3D Scenes

"There is no royal road to geometry."

- Euclid, *325 BC – 265 BC*

This chapter introduces the basic notations and the geometric concepts that are necessary to both understand and reproduce this thesis' contributions. In addition to an overview of the geometry of single and multiple cameras, we derive the relationships between a 3D scene and 2D images for asynchronous, uncoupled, and uncalibrated camera setups. In this context, we discuss the classical approaches from the literature for solving the 2D-3D registration problem.

2.1/ 3D and 2D Acquisitions

Acquiring object's shape and texture using optical devices has for long been a major topic of interest for accurate data representation, reasoning, and communication. Inventions that allowed one to successfully capture objects' surface and structure date back to as early as the beginning of Photography. The first known method is traced back to the 1860's decade when François Willème (1830-1905) used a process for producing portrait sculptures using 24 synchronized photo projections to create photo-sculptures. So much has happened since then to now. Indeed, the sensational and steady progress in various fields of science and technology (electronics, photonics, optics, computer vision, computer graphics) has enabled the emergence of reliable high resolution 3D sensors. These

sensors can be broadly categorized in to two categories discussed below ; namely, active and passive sensors.

2.1.1/ ACTIVE SENSING

Active optical 3D sensors use external projecting devices that emit light patterns targeting the object of interest. The reflection of each pattern on the object's surface back onto a camera is measured and converted into the 3D location. This conversion can be carried out in many ways depending upon the type of sensor used. Based on their working principles, there are mainly two types of active sensors : (a) structured-light and (b) time-of-flight. The active sensors are called active because they capture the 3D information using their own light source rather than the ambient lighting.

Structured-light based sensors project bi-dimensional patterns to estimate the dense depth information of the object surface points. Depending upon the application, the projected patterns can be of single or multiple frames. The main role of the projected patterns is to establish correspondences between the known pattern and camera measurements in a relatively easy manner. Note that the location of the camera measurements varies according to the surface structure. This variation encodes the scene depth information, which can be recovered using a triangulation diagram similar to the one discussed in Section 2.3 – as the geometry of the structured-light system remains equivalent to that of a par of 2D cameras. As far as the projection patterns are concerned, a large number of strategies have been developed in the literature. The most popular patterns include grids [7], dots [8], multiple slits with cuts [9] and colored patterns [10]. However, the fringe patterns [11, 12] are considered to be the most suitable for maximum reconstruction density. More detailed information regarding structured light for 3D surface imaging can be found in [13].

Time-of-flight cameras consist of emitter and receiver units. The emitter unit generates a laser pulse that impinges on the targeted object surface. The reflected laser pulse is then detected by the receiver unit along with the roundtrip travel time from transmitter to the receiver. This round-trip travel time provides the 3D position of the object surface point using the speed of light and the ray projection angle. Time-of-flight cameras in principle perform point-to-point reconstruction. However, multi-sensor arrangements allow to cover

large bi-dimensional scenes as well. These cameras can cover large distances and also provide accurate depth estimation though the coverage is limited by the allowed laser power. Additionally, time-of-flight cameras are rather costly and face scanning difficulties with surfaces exhibiting certain reflective, gloss or color properties. To improve the sensitivity and accuracy, both amplitude and frequency modulated strategies have been adopted in [14, 15] for close range distance measurements.

Other active sensing technologies include interferometry, laser triangulation, Moiré fringe range contours, etc. A detailed study about existing active as well passive sensors can be found in [16]. Some of the recent commercial systems use active sensing to capture dense 3D along with its texture information, also known as RGB-D cameras. One example of such cameras include the know well-established and known Microsoft Kinect sensor.

2.1.2/ Passive Sensing

Passive cameras use the ambient light source to capture 3D measurements from the scene. This includes shape from focus, texture gradient based reconstruction, stereo or monocular structure from motion. Most of these methods use 2D cameras as a sensing device as these are allow capturing high-quality details. Most common methods, namely stereo vision and monocular or multiple-view structure from motion, recover depth information from two or multiple such projections using the concept of image parallax. In the case of a stereo camera pair, the image parallax is generated due to their positioning. However, monocular cameras must go under motion to generate the parallax. This approach is referred to as Structure-from-Motion (SfM). In recent days, the SfM-based techniques are very popular because they offer a simple, inexpensive, and accurate solution. Additionally, it allows us to reconstruct large scenes by moving the cameras covering the whole scene or using acquisitions from many cameras [17].

The high quality images acquired from 2D cameras are due to a combined technological progress in both optics and CCD or CMOS sensors. These advanced sensors allow us to capture high-quality texture information, therefore making them suitable for parallax-based scene reconstruction. Nowadays, these cameras are very popular and widespread as they are affordable and reliable. The reconstruction from 2D cameras has recently become even more appealing as an unprecedented amount of data are being collected

and made accessible by common users from all over the world every day.

Although their exists varieties of acquisition models for 2D cameras, some have limited usage scope for professionals (fisheye, panoramic, etc.). The principle of image formation for most off-the-shelf consumer camera follows, to a large extent, the so-called pinhole or perspective camera model. In the following sections, we discuss the geometry of the image formation and image to scene registration for the perspective camera model.

2.2/ PERSPECTIVE CAMERA MODEL

We consider a static scene represented by Euclidean 3D point coordinates expressed in a common arbitrary world coordinate frame O, acquired using either offline 3D acquisition systems or online one or multiple 3D cameras. In the case of online 3D acquisitions, we assume, without loss of generality, that the scene and first 3D camera coordinate frames coincide.

A perspective 2D camera P, with intrinsic camera matrix K, captures an image of the scene points as shown in Figure 2.1. The position of the camera in world coordinate frame is defined by a 3-space rotation matrix R and a translation vector t. The camera coordinate frame is attached to a special point C, also known as camera center. If C is the homogeneous coordinate vector of C and P is the 3×4 camera matrix representation of camera P, then C is the null vector of P such that $PC = 0$ is satisfied. In the world frame, the coordinates of the camera center are given by $C = -R^T t$. Without loss of generality , we assume that the image coordinate frame is attached to a special point p, also known as principal point representing the projection of the camera center C onto the image plane $\mathcal{I}$. Note that, the matrix K encodes, among other parameters, the location of p. For unknown K, the location of p is also unknown. If Y and X are the homogeneous coordinate vectors of a scene point Y and its representation X in camera frame C, the relationship between them is given by $X = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} Y$. Hence, the projection of Y in image coordinates can be expressed as

$$x \sim K[R|t]Y, \tag{2.1}$$

where, x is the homogeneous coordinate vector of the 2D image point x and ' $\sim$' refers to the equality up to an unknown scale factor. The camera P is represented by its projection

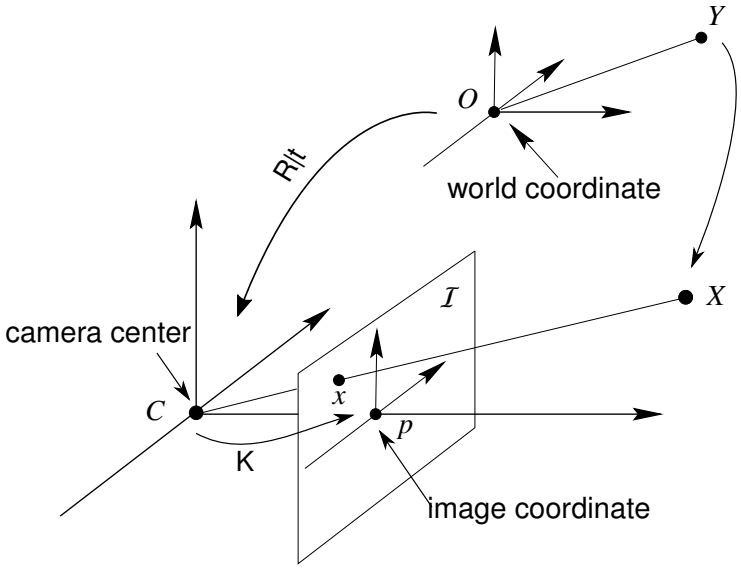

FIGURE 2.1 – Single view geometry.

matrix P = K[R|t]. The terms K and [R|t] are referred to as, respectively, the intrinsic and extrinsic parameters of a camera.

Within the scope of the present thesis, the three acquisition set-ups introduced in Section are now briefly discussed in the context of this camera model.

Fully calibrated coupled acquisition set-up : For jointly moving coordinate frames attached to O and C, 2D-3D camera setups are called synchronous if the scene points and images are acquired simultaneously in time such that the extrinsic parameters [R|t] are preserved. On the contrary, asynchronous cameras fail to preserve their relative extrinsic parameters coupling due to change in pose during acquisition time gap.

Uncoupled acquisition set-up with internally calibrated 2D cameras : When the extrinsic parameters [R|t] between coordinate frames attached to O and C are unknown, the 2D and 3D cameras are considered to be uncoupled. This could happen mainly in two different scenarios : (a) 2D image and 3D scene are captured independently, e.g. : online 2D and offline 3D acquisitions ; (b) even when the acquisitions are started with known extrinsic parameters, allowing 2D and 3D cameras to have independent and unknown motions leaves no guess opportunity on extrinsic parameters anymore.

Uncoupled acquisition set-up with internally calibrated 2D cameras : In general, ca-

mera with unknown intrinsic matrix K is called an uncalibrated camera. However, in the context of 2D-3D registration, we consider a camera to be uncalibrated if both extrinsic [R|t] and intrinsic K parameters are unknown. In practice, the problem of unknown extrinsic parameters appears to be similar to that of uncoupled cameras. Meanwhile, the problem of unknown intrinsic parameters appears either because the available images are captured by a camera with unknown or changing internal geometry during acquisitions due to zooming and/or focusing.

2.3/ CALIBRATED GEOMETRY

A primary interest in 3D vision is to recover both scene structure and camera motion using single or multiple moving cameras. Given point correspondences between at least two 2D cameras and their intrinsic parameters, it is possible to recover both the motion between cameras and the scene structure. In this context, we discuss the geometry for reconstruction from two views as shown in Figure 2.2. Let x_1 and x_2 be corresponding image feature points given in camera coordinate frames attached to C_1 and C_2 respectively. If $[R_{12}|t_{12}]$ is the motion from C_1 to C_2, the special points e_1 and e_2 are called epipoles. The epipole e_1 and e_2 are the projections of C_2 on $\mathcal{I}_1$ and C_1 on $\mathcal{I}_2$ respectively. From the projection model Equation (2.1), the coordinate vectors of the epipoles in their respective camera frames can be expressed as $e_1 \sim t_{21}$ and $e_2 \sim R_{21}^T t_{21}$.

2.3.1/ EPIPOLAR GEOMETRY

The two-view imaging model is based on the fact that, given a point in one image, its corresponding point in another image must lie in a one-dimensional space known as epipolar line. Every epipolar line intersects in the epipole. This is shown in Figure 2.2, where x_2 (the point corresponding to x_1) lies on the epipolar line l_2 that also passes through the epipole e_2. This happens because the camera centers, epipoles, image points, together with the observed 3D point X, lie on the same plane such that back-projected rays from image points intersect in 3-space. Since the line l_2 must contain e_2 and x_2, l_2 must also lie on the epipolar plane defined by C_1, C_2, and X. It is straightforward to express the normal vector of the epipolar plane as $[t_{21}]_\times R_{21} x_1$. The symbol $[.]_\times$ denotes a skew-symmetric matrix constructed from its vector argument and representing the cross-product between

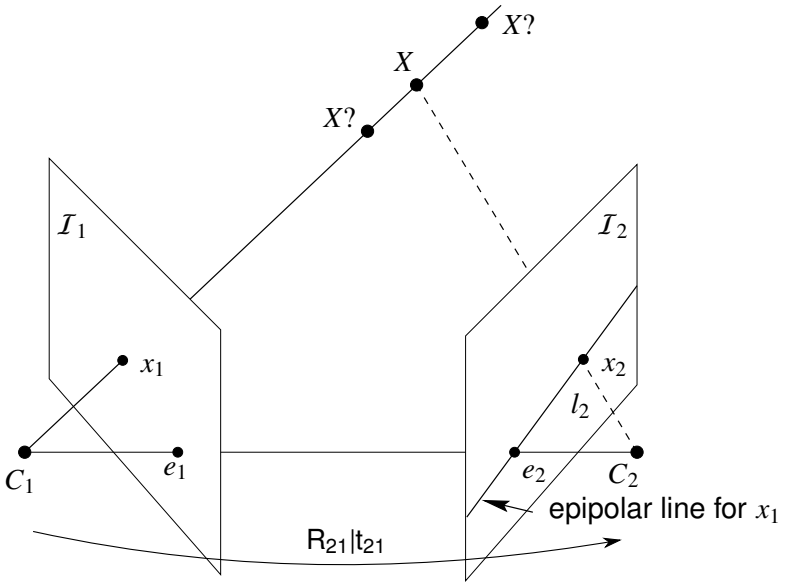

FIGURE 2.2 – Two view geromety.

vectors as a linear operation. Considering, without loss of generality, the reference frame attached to C_1, the vector emanating from C_1 and heading to x_1 and the epipolar plane normal must be orthogonal to one another. In such case, their inner product must vanish giving rise to the so-called epipolar relationship between the corresponding points x_1 and x_2 as follows :

$$\mathsf{x}_2{}^T \underbrace{[\mathsf{t}_{21}]_\times \mathsf{R}_{21}}_{\mathsf{E}_{21}} \mathsf{x}_1 = 0. \tag{2.2}$$

The matrix E_{21} is also known as the Essential matrix. By construction, the Essential matrix is of rank 2. Geometrically, this degeneracy brings non-unique mapping of a point, i.e. point in one image being mapped to its corresponding epipolar line in another. In fact, the vector $[\mathsf{t}_{21}]_\times \mathsf{R}_{21}\mathsf{x}_1$ is the homogeneous representation of the epipolar line l_2. In other words, the coordinates of the epipolar line can be safely expressed as : $\mathsf{l}_2 = \mathsf{E}_{21}\mathsf{x}_1$. The essential matrix has five degrees of freedom – three from the rotation matrix and two from the translation vector. Therefore, it can be estimated from at least five point correspondences, as every pair of correspondences provides one equation of the form of Equation (2.2). Then, the rotation and translation can be recovered by direct decomposition of the Essential matrix. The estimation and decomposition of Essential matrix are out of the scope of this work. The reader may refer [18] for more information on its estimation.

The entries of E_{21} can be estimated only up to an unknown scale factor using Equa-

tion (2.2). In other words, if any matrix $\widehat{\mathrm{E}}_{21}$ satisfies Equation (2.2), the solution $\delta\widehat{\mathrm{E}}_{21}$ is equally valid for every scalar δ. This eventually prohibits us from recovering the translation's scale. In fact, by the virtue of rotation matrix, the rotation matrix can be recovered uniquely from a finite possible solution by exploiting its orthogonality along with point visibility conditions. However, the exact scale associated with the translation vector can never be known unless extra information are fed into the problem. In this context, the extra information of scale is always necessary for its unique recovery. Therefore, the translation vector is usually computed with a unit norm hence losing the true scale of the scene.

2.3.2/ TRIANGULATION

Once the rotation and translation are recovered, the scene structure can be reconstructed by back-projecting image points to 3D-space. The 3D point X can be recovered by solving two sets of linear equations in X, each set representing a ray passing through an image point and one of the camera center. If these two rays do not intersect, the optimal X is obtained by searching for a point that minimizes the projection error in both images. Various triangulation methods can be found in [19]. In fact, the ray back-projection is performed in a fixed coordinate frame, usually attached to one of the cameras. Therefore, the obtained reconstruction is often represented in (or with respect to) camera coordinate frame.

It is important to notice that the scale associated with translation vector is arbitrarily chosen, allowing the camera centers to move along the translation vector. More specifically, for a different scale factor and fixed C_1, C_2 can move anywhere along the vector containing C_1 and e_1. This means that the back-projected ray containing C_2 and x_2 can slide over the ray containing C_1 and x_1 resulting in a different X for every scale factor. Therefore, the scale of the reconstruction depends upon the scale of the translation vector. If this reconstruction is compared against the scene represented in the world coordinate frame, neither the reconstruction scale nor the rotation and translation (between world to camera) is known. If α is the unknown scale factor, the scene and reconstructed points can be related as follows :

FIGURE 2.3 – An image illustrating the loss in scale information. It is impossible to tell whether this an image of a real or a toy car. (source : `www.123hdwallpapers.com`)

$$X = \underbrace{\begin{bmatrix} \frac{1}{\alpha}R & \frac{1}{\alpha}t \\ 0^T & 1 \end{bmatrix}}_{H_M} Y. \tag{2.3}$$

The transformation defined by H_M in Equation (2.3) is called a “similarity transform”. H_M is also known as a metric homography matrix. It goes without saying that the image-based reconstruction differs from the true Euclidean scene by a metric ambiguity. Intuitively, the reason behind this can be interpreted in two aspects : (i) loss in camera absolute pose [R|t] due to unknown world coordinate frame ; (ii) loss of scale information due to camera projection model. More informally, images of two different scales of the same object may look exactly the same. For example, as shown in Figure 2.3, a real car and a toy car can have exact same images, making it impossible to differentiate them.

2.4/ UNCALIBTRAED GEOMETRY

This section introduces the epipolar and multiple view geometry for uncalibrated cameras. Our interest focuses on obtaining the (projective) camera matrices of uncalibrated

cameras and discussing some related concepts. In particular, we will also discuss about methods that provide bounds on the plane at infinity, a key ingredient for upgrading a projective structure and in the contributions we detail in the subsequent chapters.

2.4.1/ TWO-VIEW RELATIONSHIP

When the cameras are not calibrated, the feature points can be measured only in the image coordinate frame. Hence, the measured coordinates of points x_1 and x_2 are given (in pixels) by K_1x_1 and K_2x_2, respectively. The relationship between these two measurements can be established using Equation (2.2) in the following form :

$$(K_2x_2)^T \underbrace{K_2^{-T}[t_{21}]_\times R_{21}K_1^{-1}}_{F_{21}} K_1x_1 = 0. \tag{2.4}$$

The matrix F_{21} is also known as the Fundamental matrix. Similar to the Essential matrix, the Fundamental matrix is, by construction, of rank 2. Geometrically, the fundamental matrix establishes exactly the same relationship as the Essential matrix, but only in a different measurement system – the image coordinate frames. In fact, it is straightforward to express the relationship between Fundamental and Essential matrices in the following form :

$$E_{21} = K_2^T F_{21} K_1. \tag{2.5}$$

2.4.2/ MULTIPLE VIEW RECONSTRUCTION

The Fundamental matrix encapsulates all the necessary (projective) geometric relationships for the two-view imaging model. However, when more than two views are involved, more sophisticated relationships (analogous to the Fundamental matrix), involving measurements from all the views, are required. These relationships are known as N-view multilinear tensors such as the trifocal tensor for three views and the quadrifocal tensor for four.

Although N-view tensors successfully encapsulate the geometric relationships upto 4 views, their usage is limited due to their computational complexities. Therefore, a common

practice of incorporating measurements from multiple views involves the projective factorization method. The process of projective factorization takes 2D point measurements from multiple views and decomposes it into a scene structure and camera matrices that are consistent with this structure.

PROJECTIVE FACTORIZATION

Consider 3D points $\{X_j\}_{j=1}^m$ observed by cameras $\{P^i\}_{i=1}^n$. The observed image points are given by $\{x_j^i\}$. For given point correspondences $\{x_j^1 \leftrightarrow x_j^i\}_{i=1}^n$ across images $\mathcal{I}^1, \mathcal{I}^2, \ldots, \mathcal{I}^n$, the reconstruction task is to find 3D point coordinates X_j and camera matrices P^i such that

$$\mathsf{x}_j^i \sim \mathsf{P}^i \mathsf{X}_j, \quad \text{for all } i \text{ and } j. \tag{2.6}$$

If we write this equation explicitly by introducing scale variables (or Projective depth), we have, $\lambda_j^i \mathsf{x}_j^i = \mathsf{P}^i \mathsf{X}_j$. Provided that the points are visible in all views (i.e. x_j^i is known for all i and j), the complete set of equations may be written by stacking the vectors and matrices in the following form

$$\begin{bmatrix} \lambda_1^1 \mathsf{x}_1^1 & \lambda_2^1 \mathsf{x}_2^1 & \ldots & \lambda_m^1 \mathsf{x}_m^1 \\ \lambda_1^2 \mathsf{x}_1^2 & \lambda_2^2 \mathsf{x}_2^2 & \ldots & \lambda_m^2 \mathsf{x}_m^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^n \mathsf{x}_1^n & \lambda_j^i \mathsf{x}_2^n & \ldots & \lambda_m^n \mathsf{x}_m^n \end{bmatrix} = \begin{bmatrix} \mathsf{P}^1 \\ \mathsf{P}^2 \\ \vdots \\ \mathsf{P}^n \end{bmatrix} \begin{bmatrix} \mathsf{X}_1 & \mathsf{X}_2 & \ldots & \mathsf{X}_m \end{bmatrix}. \tag{2.7}$$

Note that the matrix on the left-hand side is known as the measurement matrix, say M. By construction, the matrix M is of rank 4. This equation involves the scale variables λ_j^i, which are not part of the measurement, for each measured point x_j^i. Furthermore, note that the decomposition on the right-hand side of the above equality is not unique. To see this, observe that with any non-singular 4×4 matrix H, we have that $\mathsf{x}_j^i \sim \mathsf{P}^i \mathsf{H}^{-1} \mathsf{H} \mathsf{X}_j$ is also satisfied. Such reconstruction $\{P^i, X_j\}$ is a projective reconstruction and the matrix H is called a projective homography matrix. There are several approaches that allow decomposing the measurement matrix M in the form of Equation (2.7).

Sturm/Triggs Factorization : The first solution was proposed by *Sturm and Triggs* [20]

where the initial estimate of projective depths λ_j^i is assumed to be known. This may be obtained either from initial Projective reconstruction (for example, using fundamental matrix) or simply setting all $\lambda_j^i = 1$. Once the Projective depths are known, the measurement matrix M is complete. In case of noisy measurements, the M can be enforced to have rank 4 using Singular Value Decomposition. Thus, if $M = UDV^T$, all except the largest four diagonal entries of D are forced to zero resulting in $\hat{D}$. Then, the rank constrained measurement matrix is $M = U\hat{D}V^T$. The camera matrices and the scene points are retrieved from

$$\begin{bmatrix} P^{1T} & P^{2T} & \dots & P^{nT} \end{bmatrix}^T = U\hat{D} \text{ and } \begin{bmatrix} X_1 & X_2 & \dots & X_m \end{bmatrix} = V^T. \tag{2.8}$$

CIESTA : *Oliensis and Hartley* [21] have shown that the Sturm/Triggs Factorization method (or its variants) can converge to trivial false solution, if constraints on the projective depths are not imposed. Especially when the measurement matrix is built using noisy measurements, they showed that the Sturm/Triggs Factorization method may converge to a wrong solution or exhibit undesirable convergence behavior. To overcome this problem, CIESTA minimizes a regularized target objective

$$\min_{\Lambda,\ \text{rank}(Z)\leq 4} \frac{\|M(\Lambda) - Z\|_{fro}^2}{\|M(\Lambda)\|_{fro}^2} + \mu \sum_{i=1}^{n}\sum_{j=1}^{m} \|x_j^i\|^2 (1 - \lambda_j^i)^2, \tag{2.9}$$

where, $\mu > 0$ is the regularization constant. The operation $\|.\|_{fro}$ stands for Frobenius norm and the symbol Λ is a $n \times m$ matrix with λ_j^i entries. The optimization problem of Equation (2.9) is solved by alternately minimizing the error with respect Z and Λ. The first term of the error aims to estimate the closest rank-4 matrix Z to the measurement matrix M (as a function of unknown protective depths). However, the regularization term favors the projective depths that are close to 1. Hence, although CIESTA has proof for convergence, its solution, whether globally or locally optimal, does not necessarily converge to the correct solution even in the absence of noise. In practice, however, this problem is not very critical. CIESTA behaves very well in the presence of noise. One major disadvantage of using CIESTA is that it cannot handle the case of missing data, at least in its original form.

Element-wise Factorization : *Dai et. al.* [22] recast the problem of projective factorization as semi-definite programming one by relaxing the original problem. The offered solution is globally optimal up to some relaxation gap. This method estimates the projective

depths by minimizing the rank of the measurement matrix M. In this case, the measurement matrix is represented by the depth matrix Λ and a matrix W (constructed only from the image points) such that $\mathsf{M} = \Lambda \odot \mathsf{W}$ is satisfied. Here, $\odot$ denotes the element-wise matrix multiplication. More formally, the element-wise factorization methods attempts to solve the following problem :

$$\begin{aligned} \min_{\Lambda} \quad & \text{rank}(\mathsf{M}(\Lambda)) \\ \text{subject to} \quad & \mathsf{M}(\Lambda) = \Lambda \odot \mathsf{W}, \\ & \sum_{i}^{n} \lambda_j^i = n, \quad \text{for } \; j = 1, \ldots, m, \\ & \sum_{j}^{m} \lambda_j^i = m, \quad \text{for } \; i = 1, \ldots, n, \\ & \lambda_j^i > 0, \quad \text{for all } \; i, j. \end{aligned} \tag{2.10}$$

This problem is solved using semi-definite programming after converting it to a relaxed dual problem by minimizing the nuclear norm of the measurement matrix rather than is rank. Similar to Sturm/Triggs factorization method, the final projective reconstruction is obtained by decomposing the optimal matrix $\mathsf{M}(\Lambda^*)$. A variant of this method can also handle missing data and outliers. Although the variant behaves well for low levels of noise and few outliers, it is very sensitive to moderate and higher levels of noise, outliers, and missing data.

Other methods for projective reconstruction use unit row norm [23] or unit column norm [24] constraints. Some methods also fix row and column norms [25], row and column elements [26], or row and column sums [22]. A comparative discussion about these methods can be found in [27].

GEOMETRIC INTERPRETATION

A projective reconstruction does not preserve the aspect ratio, angle, and parallelism of the scene due to the lack of knowledge of cameras' intrinsic parameters. Figure 2.4 shows the imaging model for the two-view case, where as Figure 2.5 illustrates the geometry of the projective reconstruction (the two images have been made available at `http://www.robots.ox.ac.uk/`). In fact, for some homography matrix H, the reconstruction $\{\mathsf{H}\mathsf{X}_j\}_{j=1}^n$ and the projection matrices $\{\mathsf{P}^i\mathsf{H}^{-1}\}_{i=1}^m$ represent their Euclidean counterparts, say $\{\mathsf{X}_j^{\mathsf{E}}\}_{j=1}^n$ and $\{\mathsf{P}_{\mathsf{E}}^i\}_{i=1}^m$ respectively. Let H_{E} be the homography matrix that relates

the projective reconstruction to the Euclidean scene. Figure 2.6 shows the relationship established by this homography matrix. Furthermore, as per the definition of H_E, the following must be satisfied

$$P_E^i \sim P^i H_E^{-1} \text{ and } X_j^E \sim H_E X_j \quad \text{for all } i \text{ and } j. \tag{2.11}$$

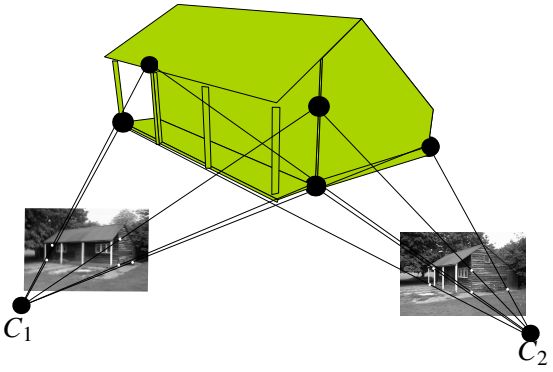

FIGURE 2.4 – Two-view imaging.

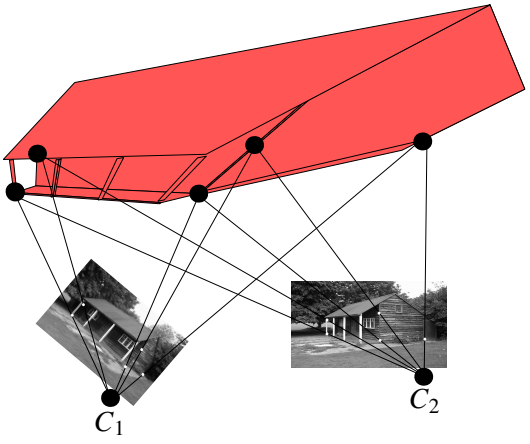

FIGURE 2.5 – Projective reconstruction.

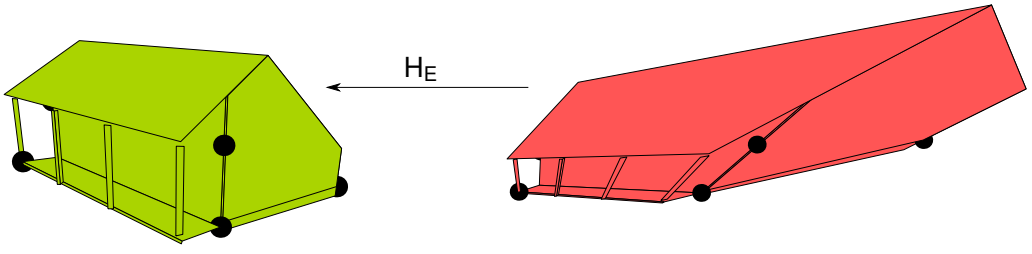

FIGURE 2.6 – Homography relating a projective reconstruction to the Euclidean scene.

2.4.3/ PROJECTIVE-EUCLIDEAN RELATIONSHIP

We are interested in deriving the analytical expression for H_E such that the criteria defined in Equation (2.11) are satisfied. For simplicity and without loss of generality, we assume that the coordinate frame of the first camera in the projective space coincides with the world frame such that $P^1 = [I|0]$, where I is a 3×3 identity matrix. Similarly, recall that Equation (2.1), the projection matrix of the first camera in the Euclidean space, is given by $P_E^1 = K^1[R^1|t^1]$. Using Equation (2.11), one can establish the following relationship :

$$K^1[R^1|t^1] \sim [I|0]H_E^{-1}. \tag{2.12}$$

For simplicity, we consider that the inverse of the homography matrix has the form

$$H_E^{-1} = \begin{bmatrix} A & b \\ c^T & d \end{bmatrix}. \tag{2.13}$$

From Equations (2.12) and (2.13), it is straightforward to establish the following relationships :

$$A \sim K^1R^1 \text{ and } b \sim K^1t^1. \tag{2.14}$$

However, the expressions for c and d require extra-knowledge.

One of the most distinctive properties of a projective reconstruction can be described with the help of parallel lines. In the general case, it fails to preserve parallelism. In other words, the parallel lines in the scene do not remain parallel in the projective reconstruction. All the parallel lines meet at a point, the so-called ideal point. Figure (2.7) shows three sets of parallel lines meeting at their respective ideal points. The plane that passes through all the ideal points is called the plane at infinity, say Π_∞. In fact, H_E establishes the relationship

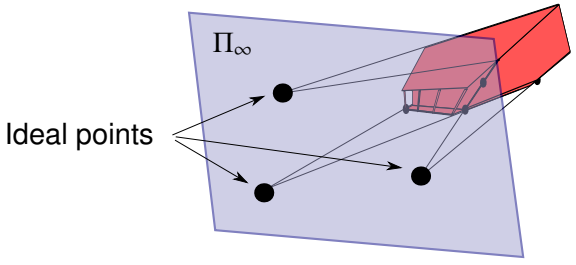

FIGURE 2.7 – Projective reconstruction showing the plane at infinity and ideal points.

between Π_∞ and the canonical plane at infinity with the help of point-to-plane duality.

Definition 1 : Projective transformation of points and planes

Consider a homogeneous vector $\Pi = (\pi^T \; 1)^T$ representing the coordinates of the plane Π. For a point Y that lies on the plane Π, $\Pi^T Y = 0$ must be satisfied. If Y gets transformed via the 4×4 transformation matrix H such that $Y \rightarrow HY$, the plane must go under the transformation $\Pi \rightarrow H^{-T}\Pi$ such that $(H^{-T}\Pi)^T HY = 0$ is satisfied.

Let the coordinates of Π_∞ be $\Pi_\infty = (\pi_\infty^T \; 1)^T$. For the known coordinates of the canonical plane at infinity $\Pi_\infty^E = (0\,0\,0\,1)^T$, with the help of Equation (2.11) and Definition 1 the following relationship can be established :

$$\Pi_\infty^E \sim H_E^{-T}\Pi_\infty. \tag{2.15}$$

Now, the terms c and d of Equation (2.13) can be expressed using Equations (2.15) and (2.14) as follows :

$$c \sim -A^T\pi_\infty = -(K^1R^1)^T\pi_\infty \text{ and } d \sim 1 - b^T\pi_\infty = 1 - \pi_\infty^T K^1 t^1. \tag{2.16}$$

Therefore, the homography matrix H_E and its inverse are written as

$$H_E \sim \begin{bmatrix} R^{1T}K^{1-1} - R^{1T}t^1\pi_\infty^T & -R^{1T}t^1 \\ \pi_\infty^T & 1 \end{bmatrix} \text{ and } H_E^{-1} \sim \begin{bmatrix} K^1R^1 & K^1t^1 \\ -\pi_\infty^T K^1R^1 & 1 - \pi_\infty^T K^1 t^1 \end{bmatrix}. \tag{2.17}$$

2.4.4/ CHEIRALITY

When a projective reconstruction is obtained using Equation (2.6), the constraint of a camera seeing only in the front direction is ignored. In the general case, this allows the reconstructed scene to split across the plane at infinity as demonstrated in Figure 2.8. Correction of such reconstruction so that the split parts correctly stick together as one object is extremely simple, if one neglects the corresponding correction for the cameras. Note that the reconstruction obtained after such correction is also called “quasi-affine” reconstruction.

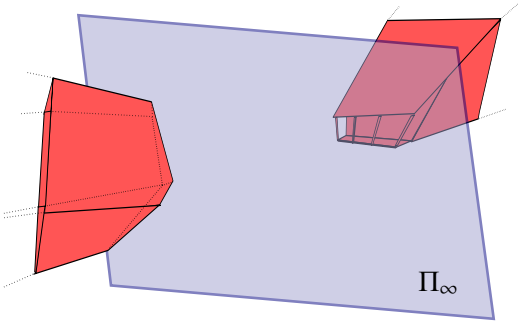

FIGURE 2.8 – Projective reconstruction when the plane at infinity passes through scene.

Without loss of generality, we start from an assumption that all points $\{X_j\}_{j=1}^n$ are visible in all the cameras. Consider the corrected projection matrices and reconstructed points are $\hat{\mathsf{P}}^i = \pm\mathsf{P}^i$ and $\hat{\mathsf{X}}_j = \pm\mathsf{X}_j$, respectively. If we introduce the implied scalar constant ζ_j^i explicitly in the Equation (2.6), since all the points are visible in all the cameras, the following must be true :

$$\zeta_j^i \mathsf{x}_j^i = \hat{\mathsf{P}}^i \hat{\mathsf{X}}_j, \quad \zeta_j^i > 0, \quad \text{for all } i \text{ and } j. \tag{2.18}$$

Note that we are interested in correcting the reconstruction so that the positive scalar constants exist, rather than knowing their exact values. One can always perform this correction, multiplying projection matrices and/or reconstructed points by -1, if necessary. These multiplication factors are also called the signatures. To find all signatures, one can first fix the signature of one of the cameras, say $\hat{\mathsf{P}}^1 = \mathsf{P}^1$. Then, the signatures of all the points can be chosen such that $\zeta_j^1 \mathsf{x}_j^1 = \hat{\mathsf{P}}^1 \hat{\mathsf{X}}_j, \quad \zeta_j^1 > 0$, is satisfied. Once the point signatures are obtained, the camera signatures can be easily obtained in a similar manner using Equation (2.18). In what follows, we refer as a projective reconstruction the one obtained after the sign correction. Furthermore, we will be using P^i for $\hat{\mathsf{P}}^i$ and X_j for $\hat{\mathsf{X}}_j$, unless mentioned otherwise.

QUASI-AFFINE UPGRADE

Now, we are interested in finding the homography H that transforms the projective reconstruction to a quasi-affine frame. Since only the relative camera orientations can be measured in the projective space, it is in fact impossible to enforce the constraint of front-seeing cameras for the quasi-affine upgrade. At best, one can enforce all cameras poin-

ting towards or away from the scene. In this regard, we wish to simultaneously transform cameras and points such that the plane at infinity passes through neither the scene points nor the cameras centers. In other words, the plane at infinity must be transferred to a new location. Doing so, is very straightforward, if the exact location of the plane at infinity is known. However, finding the location of the plane at infinity in the projective reconstruction is, at best in the absence of scene knowledge and under some favorable conditions, a challenging non-linear problem. Therefore, we seek a surrogate plane at infinity that does not cut through the scene and cameras. For the sought plane, the following must be true

$$(\Pi_\infty^T \mathsf{X}_j)(\Pi_\infty^T \mathsf{C}^i)\delta > 0, \quad \text{for all } i \text{ and } j. \tag{2.19}$$

where δ is the sign of the determinant of homography matrix H. Since we are free to multiply Π_∞ by -1 if necessary, one can assume that $\Pi_\infty^T \mathsf{C}^1 > 0$ for the center C^1 of the camera P^1. Therefore, the following inequalities can be expressed easily :

$$\begin{aligned} &\delta \Pi_\infty^T \mathsf{C}^i > 0, \quad \text{for all } i \\ &\Pi_\infty^T \mathsf{X}_j > 0, \quad \text{for all } j. \end{aligned} \tag{2.20}$$

The Equations (2.20) are also called “cheiral inequalities”. The existence of a feasible Π_∞ satisfying these inequalities is a necessary condition to upgrade the reconstruction from projective to Euclidean (or more specifically a quasi-affine). In practice, the value of δ is not known in prior. Therefore, it is necessary to seek the solution for both $\delta = \pm 1$. If the solutions exist for both cases, two oppositely-oriented reconstruction realizations are possible.

Any feasible solution of Equations (2.20), say $\widetilde{\Pi}_\infty$, is a surrogate plane at infinity. As soon as $\widetilde{\Pi}_\infty$ is a solution, so is $\alpha\widetilde{\Pi}_\infty$ for any positive α. Therefore, the addition constraints may be added to bound the solution space. If $\Pi_\infty = (\pi_1, \pi_2, \pi_3, \pi_4)^T$, then inequalities $-1 < \pi_k < 1$ for $k = 1, 2, \ldots, 4$, are sufficient for bounding the solution space as a polyhedron. While searching for a unique solution, it is a good idea to look for the farthest possible plane from all the point and cameras (as the real plane at infinity obeys this property in affine space).

To do so, we introduce a new variable, say d. Then, the quasi-affine plane at infinity $\widetilde{\Pi}_\infty$ is obtained by solving the following optimization problem using Linear Programming.

$$\begin{aligned} \max_{\Pi_\infty} \quad & d \\ \text{subject to} \quad & \delta \Pi_\infty^T \mathsf{C}^i > d, \quad \text{for all } i, \\ & \Pi_\infty^T \mathsf{X}_j > d, \quad \text{for all } j, \\ & 1 > \pi_k > -1, \quad \text{for all } k. \end{aligned} \tag{2.21}$$

Once the quasi-affine plane at infinity is recovered, one can choose the desired transformation matrix H such that its last row is $\widetilde{\Pi}_\infty^T$, and the sign of its determinant is same as that of δ. In fact, the first three rows of H can be chosen to have a simple form of $\pm[\mathsf{I}\,|\,0]$. One instance of a quasi-affine reconstruction obtained after such upgrade is shown in Figure 2.9. It shows the sign corrected scene points and the cameras before the upgrade, together with estimated quasi-affine plane at infinity.

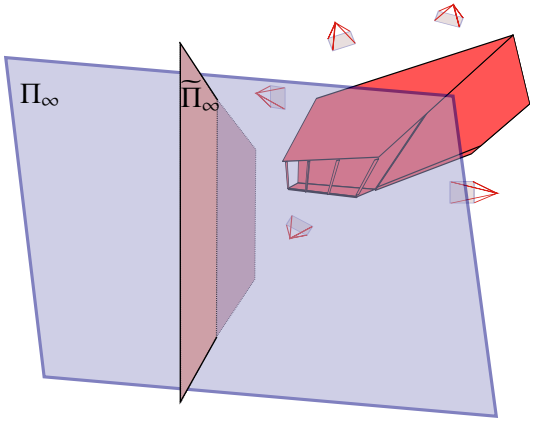

FIGURE 2.9 – An instance of the quasi-affine upgrade.

BOUNDING THE PLANE AT INFINITY

One may often be interested in finding the bounds of plane at infinity. To do so efficiently, the origin of reconstruction is first moved at the centroid (or at least inside the convex hull of scene points) of the reconstructed scene. As the plane at infinity cannot pass through the origin anymore (otherwise, it will cross the convex hull of reconstruction), we can safely fix the last entry of plane at infinity to one, such that $\Pi_\infty = (\pi_1, \pi_2, \pi_3, 1)^T$. Note that unlike the problem of Equation (2.21), the entries π_k must not be bounded in this case.

Then, the bounds on the entries of Π_∞ can be obtained by solving the following optimization problem six times (twice for each entry, one to minimize and another to maximize) using Linear Programming

$$\begin{aligned} \min/\max_{\Pi_\infty} \quad & \pi_k && \text{for } k = 1,2,3, \\ \text{subject to} \quad & \delta \Pi_\infty^T \mathsf{C}^i > 0, && \text{for all } i, \\ & \Pi_\infty^T \mathsf{X}_j > 0, && \text{for all } j. \end{aligned} \tag{2.22}$$

ROBUST CHEIRALITY

A robust algorithm for quasi-affine upgrade was first proposed by *Nistér* [28] where he argues that the upgrade process using all scene points, as in [29], is very sensitive to mismatched points. Rather than using both scene points and cameras, *Nistér* suggests to use only the camera centers – so called quasi-affine with respect to the cameras (QUARC). Although, QUARC preserves the convex hull of only the cameras, which is weaker than simultaneously preserving the separate convex hulls of both points and cameras, it is very robust because the cameras are computed from many points with robust algorithms. In contrast, relying on the correctness on every single reconstructed point is not recommended, due higher chances of incorrect point reconstruction. More importantly, if any point does not satisfy the constraint of Equation (2.20), the problem of Equation (2.21) becomes infeasible.

The QUARC method first corrects the sign of cameras such that the majority of points lie one side. This is carried out by multiplying each camera P^i by η^i defined for $i = 2,3,\ldots,n$, as

$$\eta^i = sgn[\frac{1}{2} + \sum_{j=1}^{m} sgn\{(\mathsf{P}^i\mathsf{X}_j)_3(\mathsf{P}^{i-1}\mathsf{X}_j)_3\}], \tag{2.23}$$

where the operation $(.)_3$ selects the third entry of the vector. Then, the quasi-affine plane at infinity is obtained by solving the following optimization problem

$$\begin{aligned} \max_{\Pi_\infty} \quad & d \\ \text{subject to} \quad & \frac{\Pi_\infty^T \mathsf{C}^i}{|\mathsf{C}^i|} > d, \quad \text{for all } i, \\ & 1 > \pi_k > -1, \quad \text{for all } k. \end{aligned} \tag{2.24}$$

2.5/ 2D AND 3D REGISTRATION METHODS

The problem of 2D and 3D registration appears in several contexts, including scene modeling [30], robotics [31], medical imaging [32], and virtual reality [33]. In these applications, 2D and 3D cameras need to be registered either (or both) to localize the cameras or fuse 2D and 3D information. Usually, camera localization is the major interest for robotics and virtual reality applications, whereas scene modeling and medical imaging attempt to benefit form both modalities. Nevertheless, the existing registration methods may vary largely depending upon the quality of 2D-3D measurements and their adequacy. Here, we discuss how common techniques handle the various scenarios.

2.5.1/ SINGLE IMAGE REGISTRATION

Single image based registration aims to estimate the absolute pose of a 2D camera in the 3D scene (equivalently, 3D camera with respect to 2D). A generalized 2D-3D absolute pose problem aims to estimate the location of the camera such that the mapping of 2D and 3D measurements are respected. These mappings can be geometric features, scene and image intensities, or higher level information. If Z_j and z_j are corresponding 3D and 2D measurements respectively, whose mapping is given by as function Φ, the objective of single image registration problem is to find the camera matrix P such that :

$$\begin{aligned} &\text{find} && \mathrm{P}, \\ &\text{subject to} && z_j = \Phi(\mathrm{P}, Z_j), \quad \text{for all } j. \end{aligned} \tag{2.25}$$

Feature-based registration : One of the oldest paper considering this problem dates back to 1841 by *Grunert* [34]. However, a commonly used 2D-3D registration method method, also known as Direct Linear Transformation (DLT), was proposed by *Aziz et Karara* [35] which solves the perspective-n-point problem to register a 2D image to the 3D scene or vice versa. This method minimizes an algebraic error using a simplified camera model. The algebraic error minimization is performed using a linear least square method, where a system of linear equations is constructed from multiple matched feature points between the image and the scene. The perspective-n-point problem attempts to recover the absolute pose of the camera, from where the image is captured.

The first analytical solution to the absolute pose problem was proposed by *Horaud et al.* [36] by solving the Prespective-4-Point (P4P) problem. This method offers the solution by replacing the four points with a pencil of three lines, while exploring the geometric constraints available with the perspective camera model. The final solution of P4P problem is obtained by casting it into a biquadratic polynomial equation in one unknown. However, the minimal absolute pose problem was solved by *Haralick and Lee* [37] by designing it as the prespective-3-point (P3P) problem. The extension and detailed analysis of P3P problem can be found in [38, 39]. More recent fast and robust 2D-3D registration based on point correspondences can be found in [40, 41, 42]. Furthermore, a similar method for uncalibrated cameras was proposed in [43] using 4 or 5 point correspondences.

As the above mention methods rely upon the feature point correspondences between 2D images and 3D scene, they are also called feature based registration methods. However, the category of features are not limited only to points. Many geometric features such as lines [44], colinear points [45], spheres [46], cylinder [47], and hybrid features [48], are also used to register an image to the scene. For feature based registration methods, the registration parameters (or the camera pose) are obtained using geometric relationships between the matched features.

Feature based method are widely used for many computer vision applications such as scene modeling and camera localization. In this context, *Yang et al.* [49] estimate the location of a camera with respect to a 3d model using SIFT feature descriptors [50]. *Liu et al.* [30] have used building bounding boxes represented by lines for the photo-realistic rendering of laser acquired 3D models. A fast version of feature matching methods, for feature descriptors known both in 2D and 3D, is proposed in [51]. Feature based methods require the 3D scene along with the 2D as well as 3D representation invariant feature descriptors so that a set of putative correspondences could be established. The registration process is performed using these putative correspondences, which is usually supported by Random consensus maximization (RANSAC) [52] based methods.

Intensity-based registration : An image can also be registered to the scene using the information of raw intensity [53], intensity gradient [54], or their hybrid [55]. In contrast to the feature-based registration, these methods solely depend upon the information contained all 3D points and 2D pixels. The points that coincide to pixels are considered to be correspondences, and the registration is performed by computing the pixelwise similarity

measure. A major breakthrough in this category proposed by *Viola et al.* [56] aligns a 2D image with respect to the 3D model by maximizing the mutual information between them. In fact, starting from a good initialization on the registration parameters, these methods in practice result into highly accurate registrations.

Advanced parameterization-based registration : Other single image based methods use more complex parameterizations such as skyline detection [57] and scene segmentation [58]. The registration tasks have also been addressed using the image edges [59], active contour [60] and polyhedral-silhouette [61] models, in the context of object tracking. More recently, complete 3D model specific registration methods (not necessarily for rigid motion) have also been proposed [62, 63, 64] for the object detection, recognition, and reconstruction.

2.5.2/ IMAGE-SET REGISTRATION

When a set of images acquired from different viewpoints are registered using single image registration methods, each image is treated individually. Doing so, introduces two major problems : (i) for local methods, each image requires its own initialization ; many initializations make the problem more difficult, and (ii) registered images may fail to satisfy the multiview geometric constraints (such as the epipolar constraint in two images). Furthermore, methods relying on higher level features, such as lines, and building bounding boxes, are generally suitable for Manhattan World scenes (or the like) and hence applicable only in such environments. Skylines-based methods are restricted to outdoor environment. Methods relying on a predefined surface model are, likewise, have limited applicability. Therefore, the problem of registering an image set must be treated differently. In literature, the image set to 3D scene registration problem is addressed two different ways.

Direct vs. indirect registration methods : We refer as a direct registration method when the registration process establishes the direct relationships between 2D and 3D measurements. One example of direct method includes 2D and 3D feature matching based registration. On the otherhand, registration methods that involve intermediate step, such as the relationship between SfM induced reconstruction and the scene, are called the indirect 2D-3D registration. The work flow for these two registration methods is shown

in Figure 2.10. Based on the application suitability, direct and indirect methods perform the registration to match the scene representing features of the complete geometry.

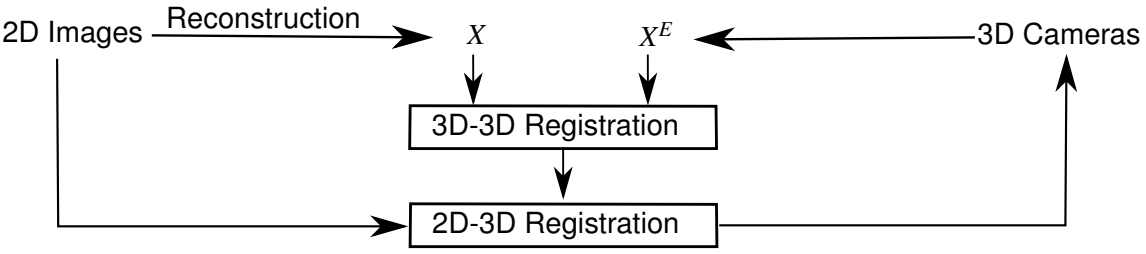

FIGURE 2.10 – Direct and indirect 2D-3D registration pipeline. Methods that use 3D reconstruction in the process are called indirect methods. The method is direct, if direct 2D and 3D relationshps are established.

FEATURE-BASED REGISTRATION

When 2D-to-2D as well as 2D-to-3D correspondences are known (or can be obtained), the image set to 3D scene registration problem can be address in two steps : SfM-based reconstruction and 3D-3D (reconstruction-to-scene) registration. It is trivial to obtain the reconstruction from image correspondence for calibrated cameras. Then the registration can be carried out by estimating the metric Homography between the reconstruction and the scene as discussed in Section 2.3. This process can also be supported by RANSAC-based method, in the presence of outliers. An efficient way of doing so has been proposed in [65]. Basically, given m point correspondences $\mathsf{X}_j \leftrightarrow \mathsf{Y}_j$ between reconstruction and the scene, the registration process minimizes the following objective function

$$\min_{\mathsf{H_M}} \sum_{j=1}^{m} d(\mathsf{X}_j, \mathsf{H_M}\mathsf{Y}_j). \tag{2.26}$$

Where $\mathsf{H_M}$ is the metric Homography matrix as in Equation (2.3) and $d(\mathsf{X}, \mathsf{Y})$ represents the distance between points X and Y. The success of feature-based single or multiple image methods is often undermined by the absence of reliable 3D descriptors and their lack of compatibility with 2D descriptors. They may also be undermined by the likewise unreliable descriptors for certain image features such as lines. For example, intensity based descriptors may change significantly if the illumination conditions differ during the

2D and 3D acquisitions. To address the problem with intensity based feature descriptors, *Zheng et al*.[66] represent spatial geometric features using local geodesic descriptors and refined the final results by maximizing the mutual information [56]. The feature descriptors extraction and matching to establish the feature correspondences was avoided with the help of manual intervention to establish the coarse alignment in [67, 68]. *Pintus et al*.[67] use Sparse Bundle Adjustment [69] (SBA) technique to refine their results, whereas, *Neugebauer and Klein* [68] minimize a blend of objective functions of texture and geometric information.

POINT-SETS-BASED REGISTRATION

When the feature correspondences are not known (or cannot be established), the image-set and the 3D scene are usually registered using SfM-induced 3D points. Two point sets, one from 3D cameras and another from SfM pipeline, are aligned using the appropriate registration parameters. The point-sets-based methods aim to find registration parameters and point correspondences simultaneously.

Most commonly used point-sets registration methods are iterative in nature. Each iteration alternates between finding point correspondences and estimating the registration parameters. While finding the correspondences, a point in one set is assigned to the closest point in another. Once the correspondences are known, the registration parameters are then estimated by minimizing the cumulative distance between all the corresponding points. If $\{\mathsf{Y}_k\}_{k=1}^{p}$ is the set of 3D points, the indirect registration method minimizes the following objective (cost) function,

$$\min_{\mathsf{H_M}} \sum_{j=1}^{m} \min_{k=1,2,\ldots,p} d(\mathsf{X}_j, \mathsf{H_M}\mathsf{Y}_k). \tag{2.27}$$

Finding the global optimal solution of the problem in Equation (2.27) is very difficult. Therefore, almost all the methods require a good initialization on the registration parameters to find a satisfactory solution.

scaled-ICP /With initialization : Note that the registration parameters in Equation (2.3) is encapsulated by the metric Homography matrix $\mathsf{H_M}$, as in Equation (2.3). Unlike the classical Iterative Closest Point (ICP) methods [70], it also includes the scale parameter

along with rotation and translation. The required extension of the ICP algorithm (considering the scale factor) has been proposed in [71, 72]. *Zhao et al.* [72] use sensor/GPS data to align a video sequence onto the 3D scene. This is performed by estimating H_M (using ICP) between 3D data from laser scanner and the SfM reconstruction, obtained from the video sequence. Given a good initialization, these methods work very well in practice, however, their success is highly biased upon the registration parameters' initialization.

Go-ICP /Without initialization : One of the very first (and only known to us) globally optimal point-sets registration method was proposed by *Yang et al.* [73] (Go-ICP). Go-ICP performs the registration on the same-scale point-sets by searching for the optimal rigid transformation parameters. If R and t are the rotation matrix and translation vector, the sough registration parameters (of rigid body transformation), the Go-ICP method minimizes the following objective function in a globally optimal manner.

$$\min_{\mathrm{R,t}} \sum_{j=1}^{m} \min_{k=1,2,\ldots,p} \|\mathrm{R}Y_k + \mathrm{t} - X_j\|^2. \tag{2.28}$$

Note that since SfM reconstructions suffer from a scale ambiguity. Therefore, Go-ICP can be used for image-set registration, only if the scale of the reconstruction is also known.

Alternative Methods

In the absence of feature correspondences, scale knowledge, and registration parameters' initialization, other methods of image-set registration appear in various flavors. In this context, image photo-consistency property has been exploited to register two or multiple images to the 3D surface model of the face in [74]. To address the unknown scale problem, *Pham et al.* [75] perform registration using mean-shift in the scale invariant space. Similarly, the method proposed by *Crosini et al.* [76](RISAG) employs a RANSAC-based inlier set maximization, in which the scale problem is handled by an extension of the 4-point congruent sets (4PCS) algorithm.

4PCS : The 4PCS algorithm was first used by *Aiger et al.* [77] to align pairs of same-scale point-sets in arbitrary initial poses, under the RANSAC framework. The main idea of 4PCS is to represent the target point-set by a set of 4 coplanar points, defining a fixed number of quadruples, such that their approximately congruent quadruples (i.e. two

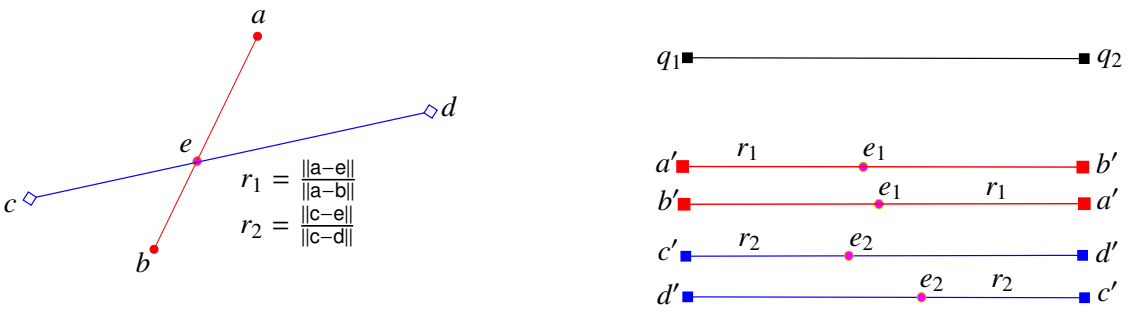

FIGURE 2.11 – A set of four coplanar points with their intersection and ratios(left). Four possible intersecting points, two for each assigments two assignments $\{a, b\}$ and $\{c, d\}$.

quadruples differing under a rigid transformation) could be extensively searched in the data point-set. For every candidate quadruple in the data, the potential transformations is computed and then applied to the whole point-set. The final transformation parameters are select among all the candidate transformation based on the maximum consensus of the transformed points.

A set of four coplanar points (not all collinear), say a, b, c, and d from the target set as shown Figure 2.11(left), define two independent ratios of three points including there intersecting point e as follows

$$r_1 = \frac{\|a - e\|}{\|a - b\|} \quad \text{and} \quad r_2 = \frac{\|c - e\|}{\|c - d\|}. \tag{2.29}$$

The ratios r_1 and r_2 are invariant that uniquely define four points upto affine transformations. Now given a pair of points, say q_1 and q_2 from the data (transformed space), two possible intersecting points can be be computed as

$$\begin{aligned} e_1 &= q_1 + r_1(q_2 - q_1), \\ e_2 &= q_1 + r_1(q_2 - q_1) \end{aligned} \tag{2.30}$$

For any data point-pair, there can be two assignments corresponding to $\{a, b\}$ and and another two corresponding to $\{c, d\}$ leading to four possible intersecting points, as shown in Figure 2.11(right). If two pairs of data points have coinciding intersecting points (i.e. $e_1 \approx e_2$), two sets of points (each set with four points), one from data and another from target, are considered to be the potential corresponding point sets – also known as congruent sets. For every pair of congruent sets, 4PCS computes the rigid transformation and counts

the number of remaining points that respect the computed transformation. If any point successfully finds it corresponding point for the given transformation, the point is called inlier for the given transformation parameters. 4PCS algorithm aims to maximize the set of inlier points using the RANSAC framework, were each iteration of RANSAC randomly selects four coplanar points from the target point-set.

RISAG : Although the 4PCS relationships are invariant upto affine transformation, algorithm *Aiger et al.* handles the points-sets only under the rigid transformation. The main problem of applying it on scaled point-sets aeries due to the unbalanced number of quadruple sampling in source and target point-sets, because of the unknown scale. To overcome sampling problem, *Crosini et al.*, in [76], express the point cloud as a set of planar regions and resample them using the object shape information. The object shape information is represented by a set of quasi planar regions using the Variational Shape Approximation algorithm [78] followed by uniform resampling with respect their area.

Calibrated vs. Uncalibrated Setups

All the feature matching-free methods described above are designed and tested only for the calibrated camera setups. Even most of the feature-based methods require the calibrated cameras. When the cameras are not calibrated, only the methods presented in [43, 30, 67] are extensively tested for uncalibrated setups. It is needless to say that given sufficient number of feature correspondences, most of the feature based methods can be extended to uncalibrated case. However, correspondence-free 2D-3D registration has not received enough attention until recently. Given a good initialization on registration parameters, it may not be very difficult to offer local methods that can handle the registration for uncalibrated case. However, upto our knowledge, there does not exist any globally optimal method (direct or indirect) which can register uncalibrated image sets to the 3D scene, without requiring the feature correspondences.

We will be discussing the related works specific to three different camera setups in their respective chapters. These discussions will also include the potential alternative methods, along with their application (demonstrated in this report) related previous works. The related works will also cover the literature review on the techniques that we will be using to solve the above mentioned problems.

3

OPTIMIZATION

"Local optimization methods are more art than technology."

- Boyd and Vandenberghe, *Convex Optimization*

In this chapter, we introduce the optimization tools and techniques that we use throughout this thesis work. Using a standard optimization problem formulation, we discuss both local and global optimization methods. Our discussion mainly focuses on global and robust techniques. In the context of global optimization, search-based methods are discussed in detail. On the other hand, robust optimization methods, commonly used to solve geometric problems, are also presented.

3.1/ MATHEMATICAL OPTIMIZATION

Definition 2 : The Optimization Problem

A mathematical optimization problem (or optimization problem) on a vector of optimization variables $\mathrm{x} = (x_1,\ x_2,\ \ldots,\ x_n)^T$ is generally defined, for an objective $f : \mathbb{R}^n \to \mathbb{R}$ and constraints $g_j : \mathbb{R}^n \to \mathbb{R}$, in the following form :

$$\begin{aligned} \min_{\mathrm{x}} \quad & f(\mathrm{x}) \\ \text{subject to} \quad & g_j(\mathrm{x}) \geq b_j, \quad \text{for} \ \ j = 1, \ldots, m. \end{aligned} \tag{3.1}$$

A vector x^* is an optimal solution of the problem 2 if the objective function satisfies $f(\mathrm{z}) \geq f(\mathrm{x}^*)$ for any z satisfying $g_1(\mathrm{z}) \geq b_1, \ldots, g_m(\mathrm{z}) \geq b_m$. In other words, the solution to an

optimization problem is a vector that minimizes the objective function while satisfying all the constraints.

The difficulty of solving problem 2 depends upon many factors : the nature of the objective and constraints, the number of variables and constraints, etc. Even in the cases where the objective and constraints are smooth polynomials, solving the optimization problem 2 in its generic form remains surprisingly difficult. This very often leads to a scenario where one needs a compromise between speed and optimality. Broadly speaking, local methods that compromise global optimality are often, although not always, only guaranteed to provide a local minimum. On the other hand, other methods are designed to find a globally optimal solution, very often at the cost of speed, are called global methods.

3.2/ LOCAL VS. GLOBAL OPTIMIZATION

Local optimization methods seek a solution that minimizes the objective only in the local region. Even when the solution reached is globally optimal, such methods cannot provide an optimality certificate. These methods generally require an initialization of the optimization variables. The initialization is very critical because it can greatly affect the final solution. Therefore, local optimization methods are suitable for applications where finding the globally optimal solution is either not of interest, or a good initialization is already known. In fact, local optimization methods are usually fast, can handle large-scale problems and are widely applicable. In many cases, local optimization methods are the only choice available due to the difficulty of the problem at hand.

Global optimization methods seek a solution that best minimizes the objective function throughout the search space. These methods do not generally require an initialization on the optimization variables. Global optimization methods are more suitable for problems that are small in size (in terms of both variables and constraints), when the value of finding the best solution is very high, and when the computation time is not critical. The main problem with global optimization is that there are no effective methods for solving a generic problem 2.

When both objective and constraints are linear in the optimization variables, a globally optimal solution can be found efficiently using the Linear Programming (LP) technique. However, the problem becomes difficult even when both objective and constraints are

convex in nature – so called convex optimization problems. Although generic convex optimization problems do not have any analytical solutions, there exist very effective methods (e.g. interior-point methods) for solving them. The main challenge of global optimization is the appropriate problem formulation. If any problem formulation fits into some given technique, such as Quadratic Programming (QP), Geometric Programming (GP), or Semi-Definite Programming (SDP), it can be solved efficiently as well [79].

When the objective or constraint function are non-linear, there does not exist any effective globally optimal method that solves generic non-linear optimization problems. Even a simple looking non-linear problem can become extremely challenging and very often becomes numerically intractable with a small increase in the number of variables or constraints. Therefore, the global optimization of non-linear optimization problems demands special care and is very problem-specific. In practice, non-linear optimization problems are solved using several different approaches. A common practice includes Branch-and-Bound(BnB) or Branch-and-Prune(BnP) search paradigms which we employ in this work and discuss in the following Section.

3.3/ SEARCH METHODS

Optimization methods based on search methods proceed by enumerating possible solutions. Depending upon the problem, the enumeration process can either be implicit or explicit. When the solution space is continuous, the search-based methods discretize the space for implicit enumeration. The BnB and BnP paradigms perform a hierarchical discretization on the optimization variables via a dynamically-built search-tree. The process of hierarchical discretization is also know as branching. The branching process requires maximum and minimum possible values of the optimization variables. The maximum and minimum values are called the upper and lower bounds and the enclosed interval is the subspace that contains the sought solution. In practice, variable bounds can initially be obtained either from a rough guess or the vague knowledge. The initial subspace defined by the given bounds can in fact be significantly large. In some cases, the optimization variables are intrinsically bounded.

While seeking the optimal solution, the search methods recursively divide the solution space into non-overlapping subspaces. During this process, every subdivision (or bran-

ching) produces two or more smaller subspaces, each of them represented by a node in the tree. The aim of the branching-based search methods is to reduce the size of the potential solution-subspace, in a recursive manner. The process of reduction involves the rejection of unpromising subspaces. This is carried out by pruning the unpromising branches as soon as possible during branching. However, such optimization methods largely rely on finding efficient pruning conditions. In most cases, the pruning is carried out either by comparing the objective function bounds or by checking the solution feasibility within the variable bounds. These two approaches are broadly categorized as Branch-and-Bound and Branch-and-Prune paradigms, respectively.

3.3.1/ Dynamic Tree Construction

Every node of the dynamic tree represents a closed convex subspace of the sought optimization variables. It is defined by the lower and upper bounds of the variables in the form of two vectors $\underline{\mathrm{x}}$ and $\overline{\mathrm{x}}$ in $\mathbb{R}^n$, respectively. The lower and upper bound entries $\underline{\mathrm{x}}_i$ and $\overline{\mathrm{x}}_i$ satisfy the conditions $\underline{\mathrm{x}}_i \leq \overline{\mathrm{x}}_i$ for all $i = 1, \ldots, n$. Thus a node is defined by the variables' interval $[\underline{\mathrm{x}}, \overline{\mathrm{x}}]$. For a given level of the tree, there exists no interval that overlaps with another. Starting from a known interval, say $\mathcal{B}_0 = [\underline{\mathrm{x}}_{init}, \overline{\mathrm{x}}_{init}]$, the tree is constructed by recursively dividing the interval into two or more tighter intervals. If $\mathcal{B}_0$ is divided into $\mathcal{B}_k$, $k = 1, \ldots, p$ intervals, it must satisfy $\mathcal{B}_k \subset \mathcal{B}_0$ for all $k = 1, \ldots, p$ and $\mathcal{B}_0 = \mathcal{B}_1 \cup \mathcal{B}_2 \ldots \cup \mathcal{B}_p$.

We illustrate the tree construction process with an example of two variables, say $\mathrm{x} = (x_1, x_2)^T$, scenario. Let a_0 be the first node representing the interval $\mathcal{B}_0$. The interval $\mathcal{B}_0$ can be divided into smaller intervals by breaking each variable bounds into two. Two variables, each with two bounds produce four combinations of smaller intervals, which are represented by first level nodes a_1, a_2, a_3, and a_4. The complete tree construction process is then carried out by similar branching in a recursive manner. Figure 3.1 provides the graphical illustration of two level branching. The nodes created in this fashion are dynamically stored in a tree structure as shown in Figure 3.2. In case of higher dimensional search space, trees are constructed in a very similar fashion. However, the number of branching per node can vary as per the convenience.

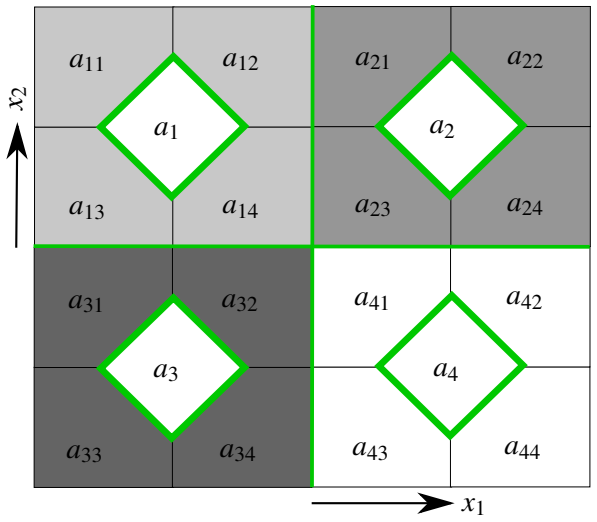

FIGURE 3.1 – Hierarchical barnching in two dimensional space. The subspaces represented by nodes a_i and a_{ij} are the results of the first and second level branching respectively.

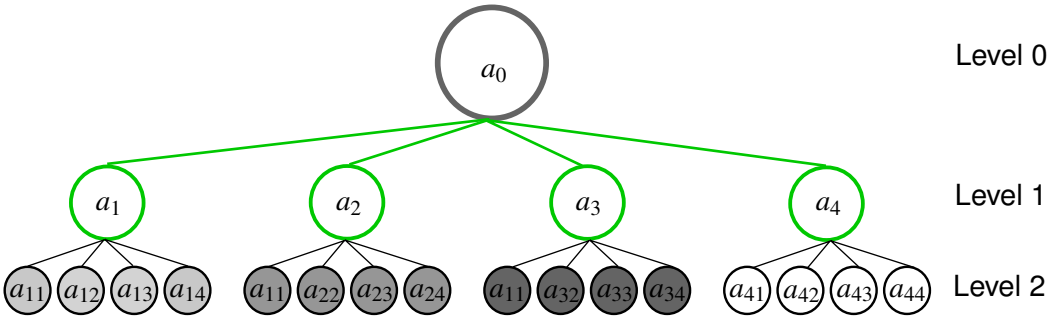

FIGURE 3.2 – Two level branching results represented as a tree structure.

3.3.2/ BRANCH-AND-BOUND PARADIGM

The Branch-and-Bound search paradigm relies on finding the best and worst possible values of the objective function (while respecting the constraints) within the variables' interval. While minimizing the objective function, the potentially best possible value is also called the lower bound. Similarly, the worst possible value is known as the upper bound of the objective. Although the bounds on the objective function do not need to be strict, the performance of BnB highly depends on its effectiveness. In fact, the worst value (the upper bound) can, very often, be found using a simple local optimization method. Any solution obtained using a randomly picked sample from the given interval, as a starting

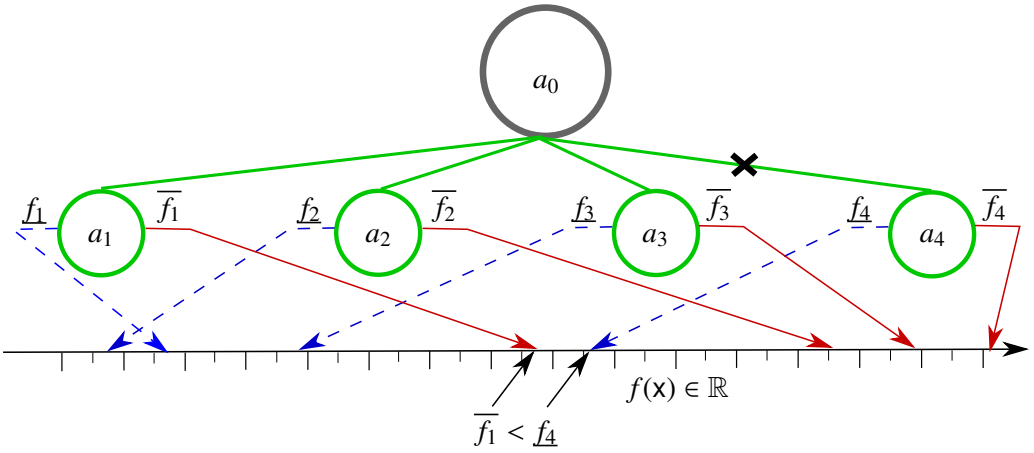

FIGURE 3.3 – Branch-and-Bound pruning based on the objective function bounds. The node a_4 is pruned because the lower bound of its objective function is larger than the upper bound of the objective function of a_1.

point, can serve as the upper bound. However, the most challenging part is that of finding the lower bound of the objective in an efficient manner. We will be discussing the specific way of finding the upper and lower bounds in its related chapter. Here, we discuss the BnB search process with an example while assuming that the bounds can be estimated for any given interval (or the node). Note that the objective function can also be maximized (if required) in a very similar manner, with the straightforward change in conventions.

Let us assume $\underline{f}$ and $\overline{f}$ are, respectively, the lower and upper bounds of the objective function $f(x)$, estimated for the interval $[\underline{x}, \overline{x}]$. We represent the node-specific bounds using the subscript corresponding to that node. For the first level nodes, mapping of both bounds to the real number axis are shown in Figure 3.3. One can observe from this diagram that the minimum possible value of node a_4 is greater than the maximum possible value of a_1 (i.e. $\overline{f_1} < \underline{f_4}$). Due to this condition, the optimal solution cannot lie in the subspace defined by node a_4. Therefore, in this tree, a_4 can be safely pruned. In fact, in general, if the lower bound of any node is greater than the upper bound of any other node, the node with bigger lower bound can always be pruned. This process can be applied repeatedly in a recursive manner to obtain the globally optimal solution. In case of multiple optimal solutions, the BnB search method allows us to obtain all the possible solutions. In practice, the branching process is carried out until the variables' interval (or the bound gap) becomes small enough such that the local method safely converges to the desired solution. The complete BnB search process is summarized in Algorithm 1.

Algorithm 1 Branch-and-Bound Search

Input : $\mathcal{B}_0$ **Output :** $f^* := f(\mathrm{x}^*)$

1: $\overline{f}_0$ = computeUpperBound($\mathcal{B}_0$) ▷ Intialization
2: f^* = processNodeBnB($\mathcal{B}_0, \overline{f}_0$) ▷ Recursive function

3: **function** PROCESSNODEBNB($\mathcal{B}, \overline{f}$)
4: $\underline{t}$ = computeLowerBound($\mathcal{B}$) ▷ Lower bound of objective function
5: $\overline{t}$ = computeUpperBound($\mathcal{B}$) ▷ Upper bound of objective function
6: **if** $\overline{t} < \overline{f}$ **then**
7: $\overline{f} \leftarrow \overline{t}$ ▷ Objective function value update
8: **end if**
9: **if** $(|\underline{t} - \overline{t}| < \epsilon) \vee (\underline{t} > \overline{f})$ **then** ▷ BnB stopping criteria
10: **return** $\overline{f}$
11: **else**
12: $(\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n)$ = divideBranches($\mathcal{B}$) ▷ Branching step
13: **for** $i = 1, 2, \ldots, n$ **do**
14: **return** processNodeBnB($\mathcal{B}_i, \overline{f}$) ▷ Recursive call
15: **end for**
16: **end if**
17: **end function**

3.3.3/ BRANCH-AND-PRUNE PARADIGM

The Branch-and-Prune search paradigm relies on finding a certificate of feasibility for the constraints within the variables' interval. During the search process, every node is examined with the feasibility test to obtain a certificate. A negative certificate (infeasibility) for any node ensures that there exist no feasible solution within its represented interval. Therefore, every infeasible node can safely be pruned. In fact, the BnP search method is solely driven by the negative feasibility certificates because any node with affirmative certificate can never be pruned. Therefore, the efficiency of BnP search depends upon strict feasibility conditions. Early detection of infeasible nodes leads to a faster search. However, finding efficient feasibility conditions is the most challenging part of BnP methods. We will discuss about our application-specific feasibility conditions in the chapter dedicated to our registration using uncalibrated cameras.

We now illustrate the working principle of BnP on an example under the assumption that the feasibility conditions for every node can be derived and tested. In this context, Figure 3.4 the pruning process for the first-level nodes of the tree is presented in Figure 3.2. In this example, the node a_4 does not qualify the feasibility test. Therefore, it can be be safely pruned following the discussion presented above. If this process is carried out re-

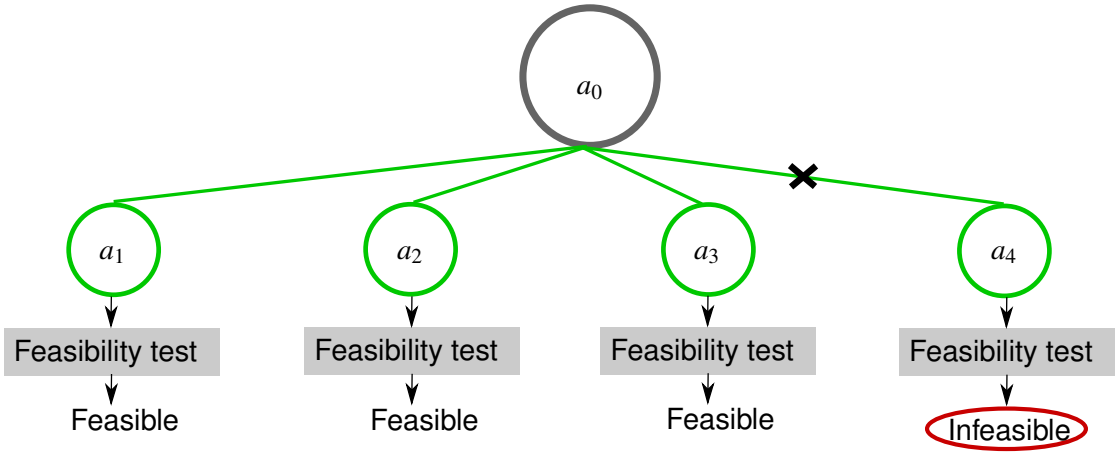

FIGURE 3.4 – An example illustrating the Branch-and-Prune pradigm. In this example, the node a_4 is prunned because it doesn't qualify the feasibility test.

peatedly in a recursive manner, the feasible region is better represented by the union of leaf nodes in every next level. Figure 3.5 shows an example of a four-level BnP process for finding a region that is feasible for two constraints. Note that the performed branching is similar to that of Figure 3.1.

FEASIBILITY REGION SEARCH

Depending upon the problems, one can be interested only in identifying the feasibility regions. In such cases, BnP alone can efficiently provide the desired solution without taking care of the objective function value. However, the standard optimization problem 2 demands a feasible solution that minimizes the objective function. BnP-based approach for objective function minimization has its own share of problem-specific difficulties. In this regard, we will be deriving an extra set of implicit constraints from the object function which, when not satisfied, certifies the nonexistence of the optimal solution within that interval. Although the details will be discussed in its corresponding chapter, these methods are considered to be very suitable for goal attainment problems.

GOAL ATTAINMENT

Goal attainment problems are optimization problems whose objective function has a targeted minimum value. These methods are particularly suitable when the objective func-

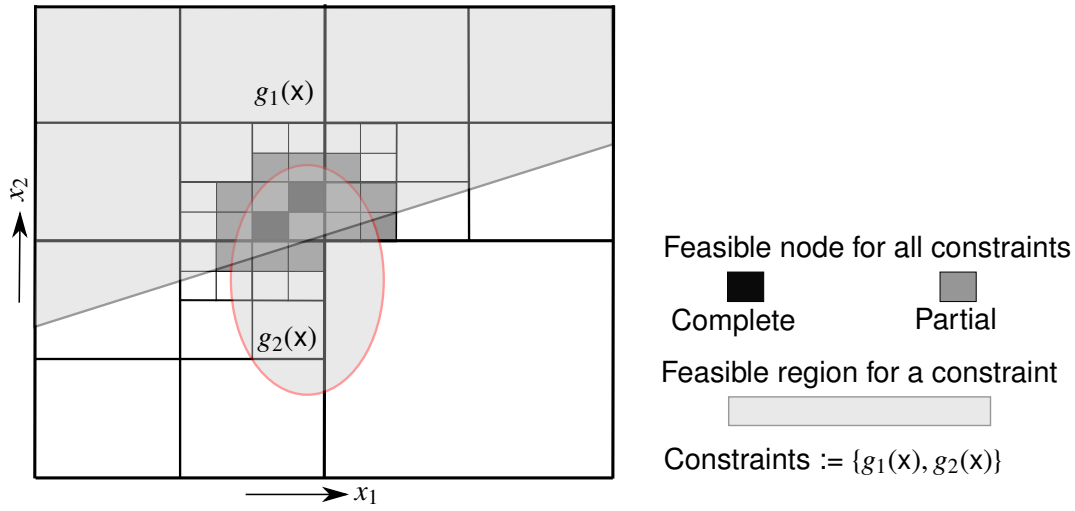

FIGURE 3.5 – Four level branching using Branch-and-Prune pradigm. Both "partial" and "complete" nodes are feasible as forth level leaves. Further branching of "partial" nodes result into tighter estimation of feasible region for all the constraints.

tions are derived from the equality constraints. When the equality constraints are perfectly satisfied, the objective function stays at zero. Otherwise, any deviation from the equality increase its value. In such cases, setting the targeted minimum value to zero is always a safe choice. In fact, the implicit constraints derived from the objective function, when not satisfied, guarantee that the solution with targeted minimum value does not lie within the processed interval.

Typically, during a BnP search, if a node qualifies the feasibility test, it is further processed using a local method. While doing so, the local solution is enforced to lie within the interval of the current node. Every local solution is then stored before proceeding to the next step. At the end, if there exists no solution with the targeted minima, the final solution is the best local solution obtained throughout. Otherwise, the method automatically returns, as soon as the target is reached. In case of multiple branches reaching the targeted value, all the solutions are returned. One of the main advantage of using BnP search is that, unlike when using BnB, the search process can be parallelized because each branch is processed independently from the others. Algorithm 2 presents the workflow of BnP-based search methods.

Although the algorithms based on the BnB and BnP paradigms have been discussed here under the prism of a Depth-First-Search tree traversal, other methods such as Breadth-First-Search and Best-First-Search can likewise be used depending upon their suitability.

Algorithm 2 Branch-and-Prune Search

Input : $\mathcal{B}_0$ **Output :** $f^* := f(x^*)$

1: $\overline{f}_0$ = feasibilityTest($\mathcal{B}_0$) ▷ A feasible objective function value
2: f^* = processNodeBnP($\mathcal{B}_0, \overline{f}_0, \tau$) ▷ Recursive function

3: **function** PROCESSNODEBNP($\mathcal{B}, \overline{f}, \tau$)
4: ($\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n$) = divideBranches($\mathcal{B}$) ▷ Branching
5: **for** $i = 1, 2, \ldots, n$ **do**
6: (f_i, feasFlag) = feasibilityTest($\mathcal{B}_i$) ▷ Feasibility flag
7: **if** feasFlag **then**
8: **if** $\overline{f} > f_i$ **then**
9: $\overline{f} \leftarrow f_i$ ▷ Objective function value update
10: **end if**
11: **if** boundGap($\mathcal{B}_i$)$< \epsilon$ **then** ▷ BnP stopping criteria
12: **return** $\overline{f}$
13: **else**
14: **return** processNodeBnP($\mathcal{B}_i, \overline{f}, \tau$) ▷ Recursive call
15: **end if**
16: **else**
17: **return** $\overline{f}$ ▷ Return if infeasible
18: **end if**
19: **end for**
20: **end function**

Moreover, the recursive operations, especially when the memory is limited, can be replaced using iterative methods.

3.4/ LINEAR MATRIX INEQUALITY

Definition 3 : Linear Matrix Ineqality (LMI)

When dealing with matrices, $A > 0$ (resp. $A \geq 0$) means that the symmetric matrix A is positive-definite (resp. positive semi-definite). A Linear Matrix Inequality (LMI) is a constraint on a real-valued vector $x = (x_1, x_2, \ldots, x_n)^T$ such that

$$A(x) = A_0 + \sum_i^n x_i A_i > 0. \tag{3.2}$$

The matrix $A(x)$ is an affine function of x involving symmetric matrices $A_0, A_1, A_2 \ldots, A_n$.

The Equation (3.2) is a convex function on x because for any $A(x_1) > 0$ and $A(x_2) > 0$, if

there exists a solution set, then $A(\frac{\alpha x_1+\beta x_2}{\alpha+\beta}) > 0$ is always true for $\alpha, \beta \in \mathbb{R}^+$. Such solution is a convex subset of $\mathbb{R}^n$ and it is called the feasible set.

A set of LMIs $\{A^j(x) = A_0^j + \sum_{i=1}^n x_i A_i^j > 0\}_{j=1}^m$ can always be written as a single LMI, using the block-diagonal matrices, in the following form :

$$\begin{bmatrix} A_0^1 & 0 & \dots & 0 \\ 0 & A_0^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_0^m \end{bmatrix} + \sum_{i=1}^{n} x_i \begin{bmatrix} A_i^1 & 0 & \dots & 0 \\ 0 & A_i^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_i^m \end{bmatrix} > 0. \tag{3.3}$$

When a LMI $A(x) > 0$ arises in **homogeneous form**, i.e. $A(x) = \sum_i x_i A_i$, it is replaced by a non-homogeneous counterpart $A(x) \geq I$ as to avoid numerical issues since $A(0) = 0$.

3.4.1/ THE LMI FEASIBILITY PROBLEM

A LMI feasibility problem consists in finding x that satisfies the considered LMIs or determining that no solution exists. In other words, it searches for the existence of non-empty subspace defined by the intersection of the cone of positive semidefinite matrices within the affine space. Establishing the feasibility of LMIs is a convex optimization problem that can be efficiently solved using interior-point methods [80]. By duality, it is a problem of finding a nonzero $G < 0$ such that $\text{Trace}(GA_i) = 0$ for $i = 1, 2, \dots, n$ and $\text{Trace}(GA_0) > 0$. The interested reader can refer [79] for the details. The feasibility test tells us whether there exists a convex subset of $\mathbb{R}^n$ for which the given constraints are always satisfied. A feasible solution is any sample from this subset which when discovered justifies the existence of such convex subset. In fact, the feasibility problem is of concern only to LMI constraints. The problem of minimizing objective function under the LMI constraints is discussed below.

3.4.2/ SEMIDEFINITE PROGRAMING

Definition 4 : Semidefinite Programing (SDP)

Semidefinite Programming (SDP) consists in minimizing or maximizing a linear objective function subject to LMI constraints. Among its many varieties, a typical SDP solves the following problem :

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{subject to} \quad & A(x) > 0. \end{aligned} \tag{3.4}$$

Here, $c \in \mathbb{R}^n$ is a vector whose entries define the weight assigned to their corresponding decision variables being multiplied.

Note that the objective function can also be maximized by inverting the sign of the entries of the vector c. Furthermore, any optimization variable is minimized or maximized by changing the sign of its corresponding coefficient in c. As the SDP is a special case of cone programming [79], its globally optimal solution can be obtained using the interior-point method. An efficient interior-point method for SDP is presented in [81].

In our work, the following lemma plays a key role in expressing some geometric constrains as a LMIs system :

Lemma 3.4.1 (Finsler's)**.** *Let* Y *be a vector,* Q *a symmetric matrix,* B *a rectangular matrix - all real-valued and of appropriate dimensions - and* γ *a scalar. The following statements are equivalent :*

(i) $Y^T Q Y > 0 \ \forall Y \neq 0 : BY = 0.$

(ii) $\exists \gamma : Q + \gamma B^T B > 0.$

Lemma 3.4.1 is due to Paul Finsler [82]. It allows converting the problem of checking the sign of a quadratic form over a subspace into one of solving a LMI problem.

	Number of Variables						
Degree	1	2	3	4	5	6	7
2	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✗	✗	✗	✗	✗
6	✓	✗	✗	✗	✗	✗	✗
8	✓	✗	✗	✗	✗	✗	✗

TABLE 3.1 – Polynomial degree vs. number of variables for PSD and SoS equivalence.

3.5/ SUM-OF-SQUARES THEORY

Definition 5 : SoS and PSD

Let $\mathbb{R}[x]$ be the ring of polynomials in n variables, $x = (x_1, x_2, \ldots, x_n)^T$, with real-valued coefficients. A polynomial $f(x) \in \mathbb{R}[x]$ is

- Positive Semi-Definite (PSD) (or nonnegative) if $f(x) \geq 0$ for all $x \in \mathbb{R}^n$;
- Sum-of-Squares (SoS) if there exist polynomials $f_k(x) \in \mathbb{R}[x]$ such that

$$f(x) = \sum_{k=1}^{p} f_k(x)^2. \tag{3.5}$$

A SoS is obviously always PSD and the converse is generally untrue. Establishing the conditions for the equivalence of classes of SoS and PSD polynomials is known as Hilbert's 17^{th} problem [83]. Indeed, Hilbert [84] proved that, for some classes of polynomials including quadratic ones, a polynomial is PSD if and only if it is SoS. Equivalence between SoS and PSD for other classes of polynomials is shown in Table 3.1. Checking whether a polynomial is PSD is NP-hard (though decidable) while checking whether a polynomial is SoS is computationally tractable using Semidefinite Programming. Semidefinite Programming employs a matrix to represent the polynomial – the so-called Gram matrix.

Definition 6 : Gram matrix [85]

Consider a polynomial $f(x) \in \mathbb{R}[x]$ of degree $2d$. Let $\mathbf{Z}_d(x)$ be the vector of monomials of $f(x)$ up to monomials of degree d. The matrix G such that $f(x) = \mathbf{Z}_d(x)^T G \mathbf{Z}_d(x)$ is a Gram matrix of $f(x)$.

The following theorem shows the role of the Gram matrix representation for expressing the condition for a polynomial to be a SoS.

Theorem 3.5.1 ([86, 85]). *A polynomial $f(\mathrm{x}) \in \mathbb{R}[\mathrm{x}]$ of degree $2d$ is SoS if and only if there exists a real symmetric positive semidefinite Gram matrix of $f(\mathrm{x})$.*

Note that since odd-degree polynomials cannot be SoS, only even-degree polynomials are concerned by such test. Checking for the existence of a positive semidefinite Gram matrix G boils down to solving a LMI feasibility problem. Recall that LMI feasibility can be efficiently checked using interior-point methods. Theorem 3.5.1 allows us to check whether a polynomial $f(\mathrm{x})$ is nonnegative for every $\mathrm{x} \in \mathbb{R}^n$. One is often interested in checking whether $f(\mathrm{x})$ is nonnegative in a semi-algebraic set $\mathcal{K}$ defined by polynomials $g_j(\mathrm{x}) \in \mathbb{R}[x]$ such that

$$\mathcal{K} = \{\mathrm{x} \in \mathbb{R}^n : g_j(\mathrm{x}) \geq 0, j = 1 \dots m\}. \tag{3.6}$$

This can be answered via the so-called Stengle's *Positivstellensatz* (Psatz) [87].

Theorem 3.5.2 (Stengle's Positivstellensatz [87]). *A polynomial $f(\mathrm{x})$ is nonnegative on $\mathcal{K}$ (defined by (3.6)), if there exist SoS polynomials $\sigma_v(\mathrm{x})$ such that*

$$f(\mathrm{x}) = \sum_{v \in \{0,1\}^m} \sigma_v(\mathrm{x}) g_1(\mathrm{x})^{v_1} g_2(\mathrm{x})^{v_2} \dots g_m(\mathrm{x})^{v_m}. \tag{3.7}$$

In fact, Stengle's Psatz can be thought of as an ordered analogue of Hilbert's Nullstellensatz. Nullstellensatz is considered to be the seminal work on algebraic geometry that establishes a fundamental relationship between geometry and algebra. However, exploiting Stengle's Psatz is difficult and may turn numerically intractable in practice because (3.7) requires 2^m SoS σ_v polynomials. But on the other hand, Putinar [88] provides a much simpler Psatz under Archimedean condition on the so-called quadratic module of the $g_j(\mathrm{x})$ polynomials. Before presenting the Putinar's Psatz, we first define the quadratic module and Archimedean condition.

Definition 7 : Quadratic module [89]

The quadratic module $M(g) = M(g_1, \dots, g_m) \subset \mathbb{R}[x]$ of polynomials $g_1(\mathrm{x})$, $g_2(\mathrm{x})$, $\dots$, $g_m(\mathrm{x})$ is the set

$$M(g) = \{\sigma_0(\mathrm{x}) + \sum_{j=1}^{m} \sigma_j(\mathrm{x}) g_j(\mathrm{x}) : \ each\ \sigma_j\ is\ SoS\}. \tag{3.8}$$

Definition 8 : Archimedean [89]

The quadratic module $M(g)$ of polynomials $g_1(\mathrm{x}), g_2(\mathrm{x}), \ldots, g_m(\mathrm{x})$ is Archimedean if $N - \|\mathrm{x}\|^2 \in M(g)$ for some $N \in \mathbb{N}$.

In this work, we are interested in finding the zeros of multivariate polynomials – which is also a major interest of algebraic geometry. Although the details of our work will be presented later, the following theorem is a key ingredient for us.

Theorem 3.5.3 (Putinar's Positivstellensatz [88])**.** *Assume the quadratic module $M(g)$ is Archimedean. If $f(\mathrm{x}) > 0$ on $\mathcal{K}$ (defined by (3.6)), then $f(\mathrm{x}) \in M(g)$.*

3.6/ Robust Optimization Methods

The primary reason for solving an optimization problem is to find the optimal solution, rather than evaluating the minimum objective function value. In many geometric problems, the optimization variables are the transformation parameters whose estimation allows one to perform tasks such as detection, registration, and reconstruction of geometric objects. In such cases, the optimization problems are designed such that the optimal transformation parameters minimize (or maximize) the objective function. Very often, the objective function is derived from multiple measurements. Multiple measurements result into multiple residual errors (usually, one for each measurement). The objective function is built by combining these residuals. However, doing so is not always straightforward. Here, we discuss some techniques of building an objective function from multiple measurements, which we will be using in the following chapters.

3.6.1/ Least-squares Approximation

Least-squares minimization method – the most commonly used approach – combines the residuals such that their sum of squares is minimized. Let $r_1(\mathrm{x}), r_2(\mathrm{x}), \ldots, r_p(\mathrm{x})$ be the residuals that are expected to be minimized by the sought optimal solution. Then, the least-squares optimization method minimizes an objective function of the following form :

$$f(\mathrm{x}) = \sum_{k=1}^{p} r_k(\mathrm{x})^2. \tag{3.9}$$

In general, least-squares minimization methods are considered to be robust towards noise under the assumption that the noise follows Gaussian distribution with zero mean. For linear residual functions, this formulation allows us to obtain the globally optimal solution in a very efficient manner. However, least-squares methods are unstable in the presence of outlier measurements or a non-Gaussian distribution of residuals. This happens mainly because the outlier measurements (or non-Gaussian residuals) affect the objective function in a way that distorts the optimal solution. More robust methods address this problem by introducing penalty functions so that the affect of outliers is minimized.

3.6.2/ PENALTY FUNCTION APPROXIMATION

Penalty function approximation problems reduce the effect of outliers by introducing a penalty function of the residuals, yielding

$$f(\mathrm{x}) = \sum_{k=1}^{p} \rho(r_k(\mathrm{x})). \tag{3.10}$$

where $\rho : \mathbb{R} \rightarrow \mathbb{R}$ is called the (residual) penalty function. In many cases, the penalty function ρ is symmetric and nonnegative with unique minima at zero, satisfying $\rho(0) = 0$. Although minimizing the objective function of Equation (3.10) produces a more robust estimation of optimization variables, solving this problem is not straightforward. The possibility of finding its globally optimal solution largely depends upon the property of residuals, and also remains problem specific. Therefore, the objective function is usually minimized locally using an iterative reweighted least-squares method. This is carried out by iteratively solving the weighted least-squares problems where the weights are generated using the following vanishing condition of first order derivative.

$$\sum_{k=1}^{p} \psi(r_k(\mathrm{x})) \frac{\partial r_k(\mathrm{x})}{\partial \mathrm{x}_i} = 0, \quad \text{for} \quad i = 1, 2, \ldots, n. \tag{3.11}$$

Here, $\psi(x) = \frac{\partial \rho(x)}{\partial x}$ is called the influence function. The weight function is defined as

$$\omega(x) = \frac{\psi(x)}{x}. \tag{3.12}$$

Hence, Equation (3.11), after introducing the weight function, takes the following form

$$\sum_{k=1}^{p} \omega(r_k(\mathsf{x}))r_k(\mathsf{x})\frac{\partial r_k(\mathsf{x})}{\partial \mathsf{x}_i} = 0, \quad \text{for} \quad i = 1, 2, \ldots, n. \tag{3.13}$$

One can solve the exact problem of Equation (3.13) using the iterative reweighted least-squares problem whose l^{th} iteration solves the following weighted least-squares problem [90]

$$\min_{\mathsf{x}} \quad \sum_{k=1}^{p} \omega(r_k(\mathsf{x})^{l-1})r_k(\mathsf{x})^2 \tag{3.14}$$

It can be seen from Equation (3.14) that the weights for the current iteration must be computed in the previous one. Therefore, this method also requires the initialization on weights or optimization variables. Note that the special class of penalty function approximation methods discussed here are also known as M-estimators.

The influence function $\psi(x)$ measures the residuals' influence on the optimization variables. For example, the penalty function of least-squares is given by $\rho(x) = x^2/2$ whose influence function is $\psi(x) = x$. This behavior of linear increase in influence with the increase in residual makes the least-squares method susceptible to outliers. A penalty function is considered to be robust if a single residual cannot make significant influence on the optimization variables. Table 3.6.2 shows a list of commonly used penalty functions (for M-estimators) along with their influence and weight functions. We will be using the Tukey penalty function, whose graphical representation is shown in Figure 3.6.

Name	$\rho(x)$	$\psi(x)$	$\omega(x)$
Least-squares	$x^2/2$	x	1
L_1-norm	$\lvert x\rvert$	$sgn(x)$	$1/\lvert x\rvert$
L_p-norm	$\lvert x\rvert^p/p$	$sgn(x)\lvert x\rvert^{p-1}$	$\lvert x\rvert^{p-2}$
Fair	$\xi^2(\frac{\lvert x\rvert}{\xi} - log(1 + \frac{\lvert x\rvert}{\xi}))$	$\frac{x}{1+\lvert x\rvert/\xi}$	$\frac{1}{1+\lvert x\rvert/\xi}$
Cauchy	$\frac{\xi^2}{2}log(1 + x^2/\xi^2)$	$\frac{x}{(1+x^2/\xi^2)}$	$\frac{1}{(1+x^2/\xi^2)}$
Huber $\begin{cases}\lvert x\rvert \le \xi \\ \lvert x\rvert > \xi\end{cases}$	$\begin{cases}x^2/2 \\ \xi(\lvert x\rvert - \xi/2)\end{cases}$	$\begin{cases}x \\ \xi sgn(x)\end{cases}$	$\begin{cases}1 \\ \xi/\lvert x\rvert\end{cases}$
Tukey $\begin{cases}\lvert x\rvert \le \xi \\ \lvert x\rvert > \xi\end{cases}$	$\begin{cases}\frac{x^6}{6} - \frac{\xi^2 x^4}{2} + \frac{\xi^4 x^2}{2} \\ \frac{\xi^6}{6}\end{cases}$	$\begin{cases}x\left(\xi^2 - x^2\right)^2 \\ 0\end{cases}$	$\begin{cases}\left(\xi^2 - x^2\right)^2 \\ 0\end{cases}$

TABLE 3.2 – Commonly used penalty functions along with their influence and weight functions. The variable ξ stands for the defined threshold.

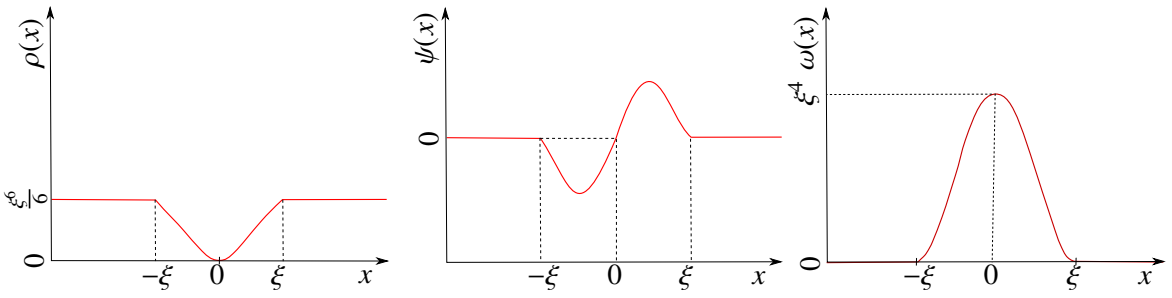

FIGURE 3.6 – Tukey : penalty function (left), influence function (middle), and weight function (right), for the threshold value of ξ.

3.6.3/ CONSENSUS-SET MAXIMIZATION

The penalty function approximation-based methods work well in practice when the residuals are computed from noisy measurements with few outliers. Besides the difficulty of obtaining the initialization, these methods are fast, efficient, and easy to implement. However, in the presence of an overwhelmingly high number of outliers, the penalty-based objective functions fail to serve their purpose. In such cases, direct detection and categorization of the inlier and outlier measurements is required. Basically, for any given configuration of the optimization variables, each measurement can be assigned either into an inlier or an outlier set, based on its residual. The configuration that results into maximum consensus of the inlier set is considered as the optimal solution – also known as consensus-set maximization.

The problem of consensus-set maximization can be formulated as an optimization problem by introducing a set of binary variables. Let $\mathcal{Z} = \{z_k\}_{k=1}^{p}$ be a set whose entry $z_k \in \{0, 1\}$ decides whether the measurement that generates residual r_k is an inlier or an outlier. More precisely, if $z_k = 1$, the measurement with residual r_k is an inlier, and vice-versa. Now, the consensus-set maximization problem can be formulated as follows :

$$\begin{aligned} \max_{\mathcal{Z},\,\mathrm{x}} \quad & \sum_{k=1}^{p} z_k \\ \text{subject to} \quad & z_k |r_k(\mathrm{x})| \leq \epsilon, \quad \forall k, \\ & z_k \in \{0, 1\}, \quad \forall k. \end{aligned} \tag{3.15}$$

A measurement is considered to be an inlier if its residual lies within the interval of $[-\epsilon, \epsilon]$. This problem is, however, very difficult to solve. One major difficulty arises due to the

binary constraints on the optimization variables z_k. The problem becomes more difficult for a non-linear residual function. If the residual functions are non-convex, this problem becomes even more difficult. In most cases, the optimization problem of Equation 3.15 is solved using Random SAmple Consensus (RANSAC) paradigm [52].

Random sample consensus : RANSAC-based methods assume that the optimization variables can always be estimated from a subset of measurements. If all the members of this subset are inliers, the optimization variables are correctly estimated, therefore the remaining inliers must also satisfy the solution. Any measurement that satisfies the solution is assigned to the inlier set of the offered solution. Then, the solution that maximizes the consensus-set is chosen as the optimal solution. The inlier set corresponding to the optimal solution is accepted as the inlier set for the original optimization problem.

The process of selecting an initial subset of measurements is the main bottleneck of consensus-based optimization methods. Therefore, RANSAC-based methods construct the subset by randomly selecting the minimum number of required measurements, at the cost of optimality. The process of random selection is performed several times, until the desired target or upper limit on the number of iterations is reached. For the measurements $\mathcal{H} = \{h_k\}_{k=1}^{p}$ associated with residuals $\{r_k\}_{k=1}^{p}$, (i.e. h_k measurement results r_k residual error), RANSAC's work flow is given in Algorithm 3. When the measurements consist of a moderate number of outlier measurements, RANSAC-based methods are both fast and robust. Thanks to its simplicity, it is also very easy to implement/adopt for a wide range of problems. The main drawback of these methods lies in the exponential complexity with the increase in number of outliers.

Algorithm 3 Random Sample Consensus Search

Input : $\mathcal{H}$, numIter **Output :** $\mathcal{Z}$

Initialization : $\overline{f} = 0, \mathcal{Z} = \emptyset$ ▷ Initial count and inlier set
for $i = 1, 2, \ldots,$ numIter **do**
  $\mathcal{A}_i$ = selectRandomSamples($\mathcal{H}$) ▷ $\mathcal{A}_i \subset \mathcal{H}$
  x_i = fitModel($\mathcal{A}_i$)
  ($\mathcal{Z}_i, f_i$) = countNumInliers($x_i, \mathcal{H}$)
  if $f_i > \overline{f}$ **then**
    $\overline{f} \leftarrow f_i$ and $\mathcal{Z} \leftarrow \mathcal{Z}_i$ ▷ Update
  end if
end for
return $\mathcal{Z}$ ▷ The maximum consensus set

4

ASYNCHRONOUS CAMERAS

"If everything seems under control, you're just not going fast enough."

- Mario Andretti, *Formula One*

This chapter is dedicated to fully calibrated coupled acquisition set-ups – asynchronous cameras. The problems associated with asynchronous cameras predominantly arise in robotic applications. In robotics, the 2D and 3D cameras whenever used, are, usually calibrated, rigidly attached on the robot, and endeavored to stay synchronized. However, either due to the hardware limitations or due to fast motion, it is difficult to remain synchronized all the time. This introduces errors in extrinsic parameters, hence the extrinsics need to be corrected while fusing 2D and 3D information, or preferably before. Therefore, this chapter is mainly focused on solving the 2D-3D registration problem in the context of robot localization and map building using asynchronous/synchronous 2D and 3D cameras.

4.1/ INTRODUCTION

The problem of accurately localizing cameras is of prime importance in many application involving visual Simultaneous Localization and Mapping (vSLAM). An accurate environment map is generally required for an accurate localization. In turn, building an accurate environment map is not possible without an accurate localization, hence, making it a paradoxical "chicken and egg" problem.

Contemporary mobile robots are, or can easily be, equipped with either or both 2D and 3D cameras [91][92][93][94][95]. As far as 3D cameras are concerned, the Iterative Closest

Point (ICP) algorithm (or one of its variants), applied on neighboring 3D point cloud measurements, is overwhelmingly used for robot localization. However, in the case of abrupt or long run displacements, localization based on 3D information alone is difficult mainly because of local minima traps (typical to ICP) and unreliable 3D feature descriptors. When a robot is equipped with both 3D and 2D sensors, 2D images are used to estimate the motion of the cameras (visual odometry) whereas the mapping is obtained directly from the 3D cameras. Indeed, the emergence of reliable 2D image feature descriptors (such as the Scale-Invariant Feature Transform (SIFT)), 2D-to-2D matching, generally supported by Random Sample Consensus (RANSAC), has become more reliable. However, the accuracy of the camera motion estimation from images, on which the robot localization relies, is undermined by the error amplitude of the extracted 2D features. When localization is based on 2D-to-3D correspondences and 2D-2D based refinement, it may suffer from significant error accumulation. One example of such error accumulation is shown in Figure 4.1. This error is usually minimized by a loop closing technique as described in [96]. However, in particular when robots travel long distances, loop closing is not always possible and may not adequately compensate for error accumulation thus leaving visible artifacts in the map. Performing small and frequent loops is recommended as to keep the accumulated error under control. In practice, making such small loops while building large maps is undoubtedly a burden for the task at hand and often impossible. Though incorporating information from extra sensors such as GPS has been proposed [95][97], it is often argued that such information is neither accurate nor reliable enough.

In this work, we propose a method for direct 2D-3D registration when 3D and 2D cameras are asynchronous. Once the asynchronous images are registered with the scene, they can be treated as synchronous acquisitions for which we propose a complete visual odometry framework that combines both 2D and 3D data. The proposed asynchronous 2D-3D registration method demands only a rough knowledge of the pose of only one of the cameras and, apart from 3D scene point coordinates, requires no other knowledge regarding the geometry of the input scene. We assume that point correspondences across images are available but 2D-to-3D correspondences are unknown. To our knowledge, there is no method that makes use of both 2D and 3D information without 2D-to-3D correspondences. Note that methods employing Bundle Adjustment (BA) with known scene [69] and PnP [35] require such 2D-to-3D correspondences to be established. In practice, good 2D correspondences between instantaneously captured images can be

obtained by using state-of-the-art feature descriptors such as SIFT. The proposed method does not require a precise set of 2D-to-3D correspondences, handles occlusions, and works even when only a part of the 3D scene is known. This framework computes the pose by localizing a set of cameras at once with respect to the 3D scene acquired in the previous frame using a minimum of three corresponding points among all the views. Furthermore, a constrained nonlinear optimization framework is also proposed for pose refinement. The first step of visual odometry uses only the known part of the scene whereas our refinement process uses the constraints that arise from the unknown part as well. The refinement step minimizes the projection errors of 3D points while enforcing the existing relationships between images. Both steps handle the problem of occlusion and that of missing scene parts by confronting the image-based reconstruction and the 3D sensor measurements. They also minimize the effect of data inaccuracies by using an M-estimator based technique. Unlike [98], our method makes no prior assumption regarding the geometry of the scanned scene.

This chapter is organized as follows. Related works are presented in Section 4.2. Asynchronous cameras related setups and the background are introduced in Section 4.3. We formulate the optimization problem to obtain the optimal odometry parameters in Section 4.4. The solution to this problem is presented in the form of an algorithm in the same section. In Section 4.5, experiments with synthetic and four real datasets are presented and discussed. Section 4.6 concludes this part of our work.

4.2/ RELATED WORKS

The problem of asynchronous cameras usually appears in the robotic applications involving visual Simultaneously Localization and Mapping (vSLAM). With the ongoing surge in affordable high quality 3D and 2D capture technologies, many mobile robots are, or can easily be, equipped with either or both vision modalities [91, 92, 93, 94, 95]. Although these methods assume the setup to be synchronized, the asynchronous case may easily arise due to the various reasons discussed in Section 4.1.

In literature, visual odometry is generally carried out by relying on 2D-2D, 3D-3D, or 2D-3D information. 2D-2D based methods typically track features in monocular or stereo images and estimate the motion between them [99, 100]. Some of these methods im-

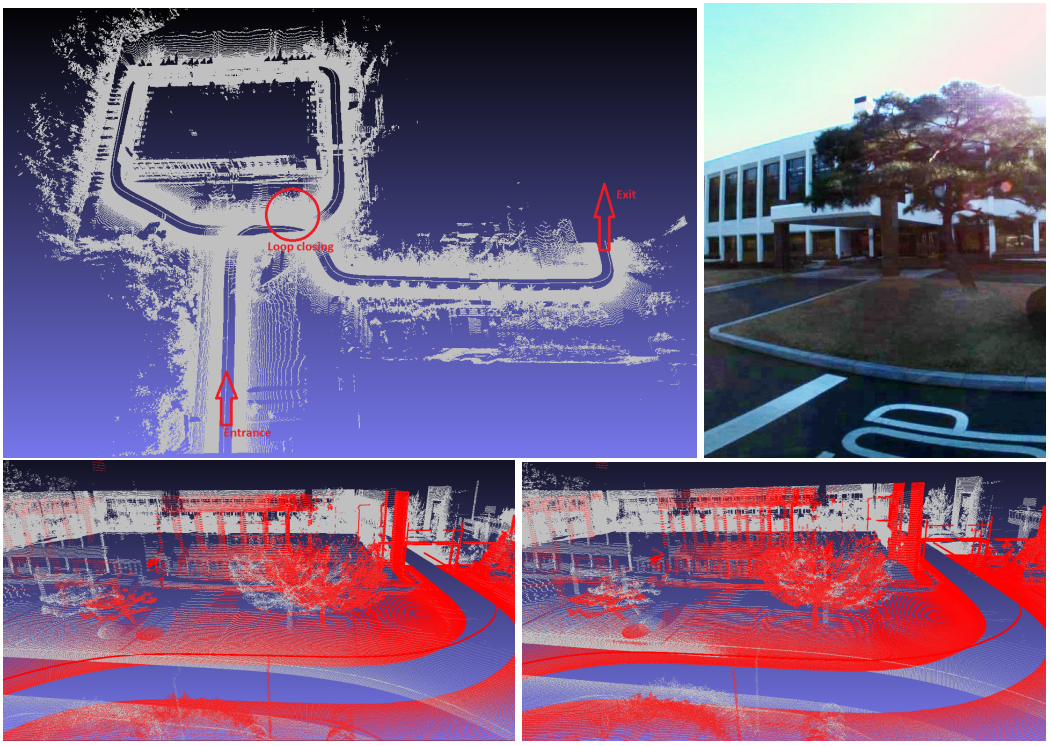

FIGURE 4.1 – An example of error accumulation around a loop : Map built by a Laser-Camera system around a large structure (top-left). Image taken at a loop closing point with only one tree at the corner (top-right). Map built before (red) and after (white) the visit around the loop using 2D-2D based refinement [95] (bottom-left). Refined map obtained using our method (bottom-right) : the scan of the same tree come significantly closer after refinement.

prove the localization accuracy by simultaneously processing multiple frames, while using Sparse Bundle Adjustment (SBA) for refinement. Some other methods obtain the motion parameters by registering images such that the photometric error between them is minimized [101, 102]. For the same purpose, most 3D-3D based methods use ICP or its variants [103, 70, 104] between consecutively acquired point clouds obtained from the 3D camera [105, 106]. However, ICP-based methods are computationally expensive due to the calculation of the nearest neighbors for every point at each iteration. Both of these methods use the information from either camera only and, hence, do not fully exploit all the available information. Methods that rely upon only one camera type may suffer from significant error accumulation during the localization process. This error is usually minimized by a loop closing technique as described in [96]. Though incorporating information from extra sensors such as GPS has been proposed [95, 97], it is often argued that such information is neither accurate nor reliable enough.

Recent works [98, 107] propose the use of information provided from both cameras du-

ring the process of localization. The work in [98] refines the camera pose obtained from Structure-from-Motion (SfM) using an extra constraint of a plane-induced homography via scene planes. This method provides a very good insight for a possibility to improve the camera pose when the partial 3D is known. However, it uses only the information from planes that are in the scene. The methods presented in [106, 107, 108] have been tested in indoor environments mainly with a Kinect sensor. Extension of these methods to outdoor environments with possibly different kinds of 3D cameras is not trivial due to various unhandled situations that may arise. Typical issues arising in outdoor scenes and/or different camera setups occur, for example, when 2D and 3D cameras do not share the exact same field of view, when the 3D points are sparse (as opposed to pixel-to-pixel mapping of RGB-D cameras), in the absence of required scene structures, and in the event of low frame rates and/or large displacements of the cameras. Note that other existing 2D-3D based refinement methods, such as SBA and loop closing, are not applicable under these circumstances because they require precise 2D-to-3D correspondences across frames.

4.3/ NOTATION AND BACKGROUND

The setup consists of a 3D camera and multiple calibrated 2D cameras as shown in Figure 4.2. At any given instant, the 3D camera captures the scene points $Y_k, k = 1, 2, \ldots, p$ in its coordinate frame O^1. A set of calibrated cameras at $\mathrm{R}_i|\mathrm{t}_i, i = 1, 2, \ldots, n$, not necessarily overlapping field of views, capture n images, from which a set of 2D feature points are extracted. Let $x^1_{ij}, j = 1, 2, \ldots, m$ represent those feature points in the i^{th} image. $P(\mathrm{R}, \mathrm{t}, \mathrm{Y})$ is the projection function that maps a point Y to its 2D counterpart in the image captured from R|t. When the system moves by R′|t′ to next position, corresponding variables are represented by the same notations with change in superscript. The poses of the second set of cameras with respect to O^1 are expressed as $\bar{\mathrm{R}}_i|\bar{\mathrm{t}}_i$. Similar to Equation (2.2), the Essential matrix between two views of the same camera in different frames is expressed as

$$E_i(\mathrm{R}', \mathrm{t}') = [\mathrm{t}'_i]_\times \mathrm{R}'_i, \tag{4.1}$$

where $\mathrm{R}'_i|\mathrm{t}'_i$ is the pose of i^{th} camera in the second frame with respect to the first one. For synchronous setups, it is related to R′|t′ as follows

$$\begin{pmatrix} \mathsf{R'}_i & \mathsf{t'}_i \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathsf{R}_i & \mathsf{t}_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathsf{R'} & \mathsf{t'} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathsf{R}_i & \mathsf{t}_i \\ 0 & 1 \end{pmatrix}^{-1}. \tag{4.2}$$

If x^1_{ij} and x^2_{ij}, $j = 1 \dots m$ are corresponding feature points in two consecutive images taken by the i^{th} camera, their 2D-to-3D correspondences are specified by a function ϕ. Let $\phi_i(j)$ be a function that maps each pair of 2D points $x^1_{ij} \leftrightarrow x^2_{ij}$ to the corresponding 3D point Y_k. Every rotation matrix R is represented by a 4×1 vector of quaternions q unless mentioned otherwise (similarly, q′ for R′). Both 3D and 2D points are represented by 3×1 vectors, the latter being the homogeneous representation in the camera coordinate system. The distance between two rotation matrices is measured by computing the spectral norm of their difference. For a matrix A, its spectral norm is denoted as |||A|||.

4.4/ 2D-3D Visual Odometry

In this section, we establish the relationships between a set of image pairs and scene points. Using these relationships, we propose an optimization framework whose optimal solution is the required odometry parameters. A complete algorithm for solving this optimization problem is also discussed. The proposed method deals with both the asynchronous and synchronous cases separately. In the asynchronous case, the camera's extrinsic parameters $\mathsf{R'}_i|\mathsf{t'}_i$ are assumed to be unknown. In the synchronous case these parameters are known and fully exploited during the motion estimation process. We also assume that the 2D-to-2D correspondences between image pairs acquired by the same camera are known.

4.4.1/ Problem Formulation

The relationship between 2D and 3D points is depicted in the ray diagram given in Figure 4.2. The projection error of points on the first set of cameras is given by

$$e^1(\mathsf{R}_i, \mathsf{t}_i, \phi_i(j)) = \|\mathsf{x}^1_{ij} - P(\mathsf{R}_i, \mathsf{t}_i, \mathsf{Y}_{\phi_i(j)})\|^2. \tag{4.3}$$

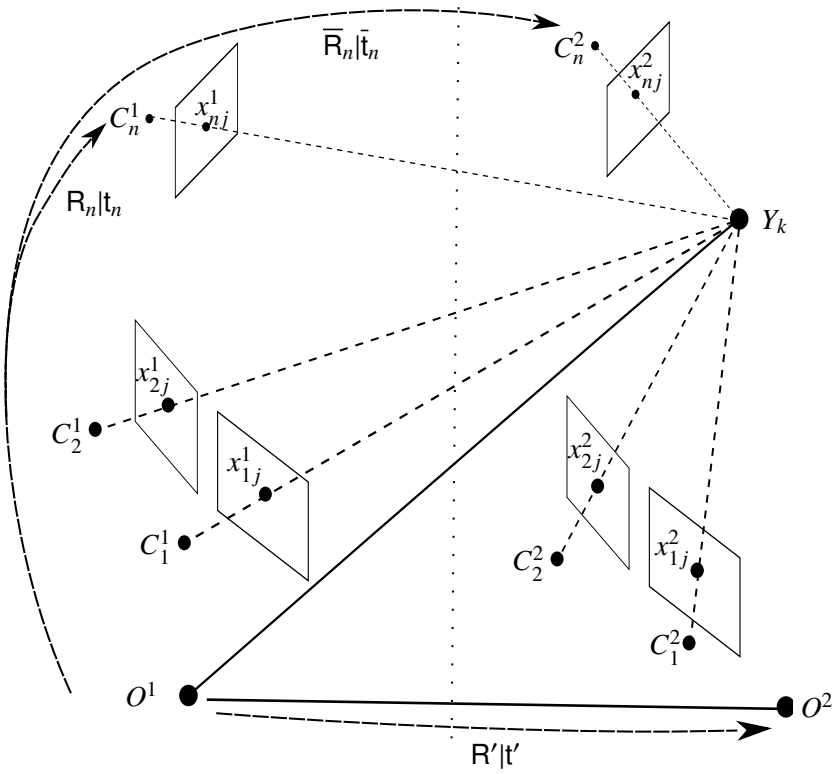

FIGURE 4.2 – Ray diagram of the experimental setup.

Similarly, for the second set of cameras, the projection error is given by

$$e^2(\mathrm{R}_i, \mathrm{R}', \mathrm{t}_i, \mathrm{t}', \phi_i(j)) = \|\mathrm{x}_{ij}^2 - P(\mathrm{R}_i\mathrm{R}', \mathrm{R}'\mathrm{t}_i + \mathrm{t}', \mathrm{Y}_{\phi_i(j)})\|^2. \tag{4.4}$$

Furthermore, the epipolar constraint that relates the points in two views of different frames can be written as

$$(\mathrm{x}_{ij}^2)^T E_i(\mathrm{R}', \mathrm{t}')\mathrm{x}_{ij}^1 = 0. \tag{4.5}$$

While (4.3) locates the first camera, (4.4) locates the second camera with respect to the world reference frame while preserving its relationship to the first one. Similarly, (4.5) localizes the second camera with respect to the first one. Equations (4.3), (4.4) and (4.5) are obviously redundant. However, in the presence of noise in the data and unknown correspondences all constraints must be enforced : satisfying only the non-redundant conditions does not necessarily satisfy all of them. In addition, (4.5) makes use of the unknown part of the scene as well. Therefore, all three equations will be incorporated in our optimization framework in which (4.4) is chosen to be the objective (as it includes the pose of both the cameras) while the rest are used as constraints.

Our problem is to localize a set of 2D cameras with known 2D-to-2D $(x^1_{ij} \leftrightarrow x^2_{ij})$ and unknown 2D-2D-to-3D $(x^1_{ij} \leftrightarrow x^2_{ij} \leftrightarrow Y_{\phi_i(j)})$ correspondences in the presence of noise. Hence, finding the optimal ϕ_i itself is part of the optimization process. Therefore, the optimization framework can be written as

$$\begin{aligned} \min_{\mathsf{q}_i,\mathsf{t}_i,\mathsf{q}',\mathsf{t}',\phi} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} \|\mathsf{x}^2_{ij} - P(\mathsf{R}_i\mathsf{R}', \mathsf{R}'\mathsf{t}_i + \mathsf{t}', \mathsf{Y}_{\phi_i(j)})\|^2, \\ \text{subject to} \quad & \|\mathsf{x}^1_{ij} - P(\mathsf{R}_i, \mathsf{t}_i, \mathsf{Y}_{\phi_i(j)})\|^2 = 0, \\ & (\mathsf{x}^2_{ij})^T E_i(\mathsf{R}', \mathsf{t}')\mathsf{x}^1_{ij} = 0, \\ & \|\mathsf{q}_i\|^2 = 1, \|\mathsf{q}'\|^2 = 1, \ i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, m. \end{aligned} \tag{4.6}$$

The optimization problem (4.6) considers that every image point has its corresponding 3D point in the scene. In practice, there could be extra 2D or missing 3D points resulting in invalid 2D-to-3D correspondences. We address this problems by assigning the weights derived from a scale histogram to each correspondence. Furthermore, we also relax the strict equality of constraints to avoid the infeasibility that would arise due to the noisy data (or the discretisation during the image formation process).

If X_{ij} is the two-view reconstruction, the relative scale of reconstruction for known 3D-to-3D correspondences $X_{ij} \leftrightarrow Y_{\phi_i(j)}$ can be computed as

$$s_i(j) = \frac{\|\mathsf{R}_i^T\mathsf{X}_{ij} - \mathsf{R}_i^T\mathsf{t}_i\|}{\|\mathsf{Y}_{\phi_i(j)}\|}, \quad j = 1, 2, \ldots, m. \tag{4.7}$$

Since the reconstructed points from each pair share a common scale, in the ideal case, we have $s_i(j) = c_i, \ \forall j \in 1, 2, \ldots, m$ (for constants c_i-s). In practice, when the histograms $H_i(u)$, $u = 1, 2, \ldots, b$ of these scales are built, they hold the highest number of samples in the bin corresponding to the true scale. If those bins are U_i, then the weights are distributed as follows :

$$w_i(j) = \begin{cases} 1 & s_i(j) \in H(U_i) \\ 0 & \text{otherwise.} \end{cases} \tag{4.8}$$

Furthermore, the effect of data inaccuracies is reduced by introducing a robust estimation technique. Hence, the optimization problem (4.6) with robust estimation and histogram-based weighting can be re-written as

	Input	Estimation
Asynchronous	2D-2D corresp.	R_i, t_i and $\bar{\mathsf{R}}_i$, $\bar{\mathsf{t}}_i$
Synchronous	2D-2D corresp., R_i, t_i	R′, t′

TABLE 4.1 – Known and estimated parameters.

$$\begin{aligned} \min_{\mathsf{q}_i,\mathsf{t}_i,\mathsf{q}',\mathsf{t}',\phi} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} w_i(j)\rho(\|\mathsf{x}^2_{ij} - P(\mathsf{R}_i\mathsf{R}', \mathsf{R}'\mathsf{t}_i + \mathsf{t}', \mathsf{Y}_{\phi_i(j)})\|), \\ \text{subject to} \quad & w_i(j)\rho(\|\mathsf{x}^1_{ij} - P(\mathsf{R}_i, \mathsf{t}_i, \mathsf{Y}_{\phi_i(j)})\|) = 0, \\ & \rho((\mathsf{x}^2_{ij})^T E_i(\mathsf{R}', \mathsf{t}')\mathsf{x}^1_{ij}) = 0, \\ & \|\mathsf{q}_i\|^2 = 1, \|\mathsf{q}'\|^2 = 1, \ i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, m. \end{aligned} \tag{4.9}$$

where $\rho(.)$ is Tukey bi-weighted potential function as discussed in Section 3.6, defined in Table 3.6.2, and illustrated in Figure 3.6.

Note that any 2D-to-3D correspondence that does not vote for the valid scale is considered to be an outlier. Here, the derived cost depends only upon the known part of the scene whereas the constraint includes the unknown part as well. The optimal odometry parameters are obtained by iteratively solving this optimization problem. Each iteration breaks the problem down into two subproblems : (a) 2D-to-3D registration and (b) Camera pose refinement.

4.4.2/ 2D-TO-3D REGISTRATION

The registration step coarsely localizes the cameras with respect to the scene. Here, we discuss the registration methods for asynchronous and synchronous cases as two separate subproblems. In the asynchronous case, finding the 2D-to-3D correspondences required for registration is not trivial. This is done by iterating between camera poses and the correspondence estimation. On the other hand, finding the precise cross-frame correspondences for the synchronous case is not easy either. Cross-frame image-to-scene registration in synchronous acquisition is carried out by using minimal point RANSAC-based pose estimation. The choice of registration methods depends upon the experimental setup. The known input and estimated parameters for two different cases are summarized in the Table 4.1.

ASYNCHRONOUS CASE

The main problem in the asynchronous acquisition is that the poses of the camera with respect the scene are unknown. This makes solving 2D-to-3D correspondence problem very challenging. Since these correspondences are unknown, the reconstruction that can be obtained from images is related to the scene by an unknown scale factor. To avoid the role of this unknown scale, we minimize a cost function which is independent of it, while imposing the epipolar constraint between images. The proposed optimization problem for asynchronous cameras registration is as follows :

$$\begin{aligned} \min_{\mathsf{q}_i,\mathsf{t}_i,\phi} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} w_i(j)\rho(\|(\mathsf{x}_{ij}^2)^T E_i(\mathsf{R}',\mathsf{t}')P(\mathsf{R}_i,\mathsf{t}_i,\mathsf{Y}_{\phi_i(j)})\|), \\ \text{subject to} \quad & w_i(j)\rho(\|\mathsf{x}_{ij}^1 - P(\mathsf{R}_i,\mathsf{t}_i,\mathsf{Y}_{\phi_i(j)})\|) = 0, \\ & \|\mathsf{q}_i\|^2 = 1,\ \ i = 1,2,\ldots,n,\ j = 1,2,\ldots,m. \end{aligned} \tag{4.10}$$

The initial estimate of R_i', t_i' is obtained using the SfM-based relative pose estimation method. Note that $R'|t'$ is the motion between the 3D cameras, whereas $R_i'|t_i'$ are the same for 2D cameras. Although $E_i(R',t')$ is shown as the function of R' and t' in Equation (4.1), it is actually the function of R_i' and t_i', which are again dependent upon both (R',t') and (R_i,t_i) as shown in Equation (4.2). We choose ϕ such that it maps every pair of image points to a 3D point that respects the constraint while minimizing the cost. The constraint violation is penalized by a simple but effective static penalty function as discussed in [109]. Therefore,

$$\phi_i(j) = \underset{k\in\{1,\ldots,p\}}{\arg\min} \quad \|\mathsf{x}_{ij}^1 - P(\mathsf{R}_i,\mathsf{t}_i,\mathsf{Y}_k)\| + \|(\mathsf{x}_{ij}^2)^T E_i(\mathsf{R}',\mathsf{t}')P(\mathsf{R}_i,\mathsf{t}_i,\mathsf{Y}_k)\|. \tag{4.11}$$

Hence, the optimal poses of the first set of cameras are obtained, for each camera i separately, by solving

$$\begin{aligned} \arg\min_{\mathsf{q}_i,\mathsf{t}_i} \quad & \sum_{j=1}^{m} w_i(j)\rho(\|(\mathsf{x}_{ij}^2)^T E_i(\mathsf{R}',\mathsf{t}')P(\mathsf{R}_i,\mathsf{t}_i,\mathsf{Y}_{\phi_i(j)})\|), \\ \text{subject to} \quad & w_i(j)\rho(\|\mathsf{x}_{ij}^1 - P(\mathsf{R}_i,\mathsf{t}_i,\mathsf{Y}_{\phi_i(j)})\|) = 0, \\ & \|\mathsf{q}_i\|^2 = 1,\ \ i = 1,2,\ldots,n,\ j = 1,2,\ldots,m. \end{aligned} \tag{4.12}$$

This is a constrained nonlinear optimization problem on the quaternion parameters whose local optimal solution can be obtained by the iteratively re-weighted least-squares (ILRS)

technique. In fact, depending upon one's choice, it can also be solved linearly on R and t using singular value decomposition. However, the linear solution does not constrain R to be a rotation matrix. Therefore, the obtained solution needs to be enforced as a rotation matrix before extracting the quaternion parameters.

For each pair of images, the scale of the reconstruction is finally estimated by averaging the scales of inliers as follows

$$\mu_i = \frac{\sum_{j=1}^{m} w_i(j) s_i(j)}{\sum_{j=1}^{m} w_i(j)}, \quad i = 1, 2, \ldots, n. \tag{4.13}$$

Finally, the absolute poses of the second set of cameras in O^1 can be obtained through

$$\begin{pmatrix} \bar{\mathsf{R}}_i & \bar{\mathsf{t}}_i \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathsf{R'}_i & \mu_i \mathsf{t'}_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathsf{R}_i & \mathsf{t}_i \\ 0 & 1 \end{pmatrix}. \tag{4.14}$$

Once the cameras are fully registered, they can be thought as synchronized ones. This is because the second set of cameras can be localized in the first coordinate frame. Henceforth, we assume that the unsynchronized cameras are synchronized once registered to the scene.

SYNCHRONOUS CASE

It is trivial to find the 2D-to-3D correspondences $x_{ij} \leftrightarrow P(\mathsf{R}_i, \mathsf{t}_i, \mathsf{Y}_k)$ in one frame. However, cross-frame correspondences are required in order to estimate the motion R′|t′. Such correspondences can be obtained by matching the 2D feature points between images. Note that most $P(\mathsf{R}_i, \mathsf{t}_i, \mathsf{Y}_k)$, when considered as feature points, are unlikely to result in reliable feature descriptors for matching. Therefore, we extract a separate set of 2D feature points to obtain better 2D-2D correspondences $x^1_{ij} \leftrightarrow x^2_{ij}$. Methods based on relative pose require at least 5 such correspondences to compute the motion with an unknown scale. On the other hand, if 2D-to-3D correspondences $x^2_{ij} \leftrightarrow Y_k$ can be found, it would require only 3 points to estimate the motion including the scale. In order to benefit from this, the required 2D-to-3D correspondences are computed for each image which is established

by the mapping function $\phi_i(j)$ computed as

$$\phi_i(j) = \underset{k\in\{1,\ldots,p\}}{\arg\min} \quad \|\mathrm{x}_{ij}^1 - P(\mathrm{R}_i, \mathrm{t}_i, \mathrm{Y}_k)\|, \quad j = 1, 2, \ldots, m. \tag{4.15}$$

It is important to notice that the correspondences obtained in this manner are not perfect. We make a strong consideration of this restriction while refining the estimated motion. The search required to minimize (4.15) can be performed using a KD-tree like structure where the projections of all 3D points build one tree in each image. The detected feature points traverse these trees in search for the best possible match. Once the required correspondences are obtained, the set of cameras in the second frame can be localized with respect to previously acquired 3D scene using the method presented in [110]. The advantage of using this method is that it requires a minimum of 3 correspondences among all the views and does not require a complex scene as demanded by ICP or SfM. For example, even a planar scene with sufficient texture can be processed. For low frame rates and/or large displacements, feature matching methods still work better than tracking them. Since only 3 correspondences are needed, finding them from already matched 2D-2D to sparse 3D is very much achievable in practice.

4.4.3/ Camera Pose Refinement

Recall that in both asynchronous and synchronous cases the final result is the registration of next frame images to the previous scene. In fact, the obtained registration parameters are the absolute poses of the cameras. However, in practice, the motion obtained in this manner is not very accurate. In this step, we refine these coarse motion/registration parameters while making use of scene information. The refinement process optimizes the motion parameters such that the SfM reconstruction is the closest to the known scene. During this process, the asynchronous setups are refined by directly solving the equation presented in (4.9) for the known correspondence function ϕ. The correspondences required in this step are obtained directly from the registration process. However, the synchronous setups are refined by solving the following optimization problem :

$$\begin{aligned} \min_{\mathsf{q}',\mathsf{t}'} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} w_i(j)\rho(\|\mathsf{x}_{ij}^2 - P(\mathsf{R}_i\mathsf{R}', \mathsf{R}'\mathsf{t}_i + \mathsf{t}', \mathsf{Y}_{\phi_i(j)})\|), \\ \text{subject to} \quad & \rho((\mathsf{x}_{ij}^2)^T E_i(\mathsf{R}', \mathsf{t}')\mathsf{x}_{ij}^1) = 0, \\ & \|\mathsf{q}'\|^2 = 1, \ i = 1, 2, \ldots, n, j = 1, 2, \ldots, m. \end{aligned} \tag{4.16}$$

Note that the refinement process uses all the cameras simultaneously to refine R' and t', unlike in Equation (4.12) of the asynchronous case. This is again a constrained nonlinear optimization problem that can be solved by ILRS technique. Each iteration of IRLS uses the interior-point method to solve the constrained nonlinear least-squares problem.

4.4.4/ The Algorithm

Starting from known 2D-to-2D correspondences, the algorithm iteratively estimates the odometry parameters mentioned in Table 4.1. Every iteration reduces the cost function (4.9) in two steps while satisfying its constraints. Here, we present two different algorithm for asynchronous and synchronous cases separately.

Algorithm 4 Asynchronous case

For known initial guess on $\mathsf{R}_i|\mathsf{t}_i$ and $\mathsf{R}'_i|\mathsf{t}'_i$ obtained from relative pose estimation, refine them through the following two steps :

1. **Camera alignment** : iteratively align the cameras to scene until convergence,
   - **(a)** estimate the relative pose using 2D-to-2D correspondences ;
   - **(b)** compute 2D-to-3D correspondences using (4.11) ;
   - **(c)** build multiple scale histogram $H_i(u)$ and compute weights $w_i(j), j = 1, 2, \ldots, m$;
   - **(d)** update the pose of the first set of cameras using (4.12).
2. **Simultaneous pose refinement** : starting from the results obtained in the "Camera alignment" step, refine poses of both sets of cameras by solving (4.9).

Obtain real scale μ_i and compute the absolute pose using (4.14).

Discussion : The problem addressed here is similar to that of scaled-ICP /with initialization described in Section 2.5.2. The solution to Equation (4.10) provides the scaled-ICP-like registration of image-sets in a direct manner. Our solution can be thought at that of [72]. However unlike [72], where the 3D-to-3D correspondences are searched, we established 2D-to-3D direct correspondences using Equation (4.11). Once the correspondences are found, the Equation (4.12) is basically refining the registration parameters, in a very usual ICP-based methods. Algorithm 4 describes the steps for Asynchronous case. Here, step

Algorithm 5 Synchronous case

1. **2D-3D registration** : for known extrinsics $\mathsf{R}_i|\mathsf{t}_i, i = 1 \dots n$, iterate over the following steps until convergence :
   For each Camera $i = 1, 2, \dots, n$
   - **(a)** compute $P(\mathsf{R}_i, \mathsf{t}_i, \mathsf{Y}_k), k = 1, 2, \dots, p$ and build a KD-tree ;
   - **(b)** find 2D-to-3D correspondences maps $\phi_i(j), j = 1, 2, \dots, m$ using (4.15).

   Using all Cameras : perform 2D-3D-based RANSAC and estimate $\mathsf{R}'_{i,0}|\mathsf{t}'_{i,0}$ using [110].
2. **2D-2D-to-3D based refinement** : starting from $\mathsf{R}'_{i,0}|\mathsf{t}'_{i,0}$, iterate until convergence,
   - **(a)** Reconstruct the scene $X_{ij}, j = 1, 2, \dots, m$ and compute scales $s_i(j)$ for each point ;
   - **(b)** Build a combined scale histogram $H(u), u = 1, 2, \dots, b$ for all cameras ;
   - **(c)** Compute weights $w_i(j), j = 1, 2, \dots, m$ using $H(u)$;
   - **(d)** Update the pose by optimizing (4.9) for known $\phi_i(j)$ obtained from 2D-3D registration.

1 (Camera alignment) only aligns the image-sets with respect the 3D scene, whereas step 2 (Simultaneous pose refinement) refines the pose using coarse alignment obtained from step 1.

Regarding the choice of R_i and t_i, once the essential matrix is fixed, for a dense 3D scene, one can always find a 3D point that lies on the ray back-projected from the image point. However, for any R_i and t_i, the 3D point lying on the ray does not share a common scale with rest of the others. Thanks to the scale histogram, a 3D point belonging to common scale with rather some error (due to inaccurate current R_i and t_i estimates) is selected. Now, since the 3D point is not error free, its projection on the image doesn't necessarily satisfy either the cost or the constraint. Furthermore, due to such tread off between scale and the point on back-projection ray, satisfying the constraint doesn't necessarily satisfy the cost, or vice versa.

Normalization and Pose Recovery

For the sake of numerical stability, the 3D scene points are normalized such that the distance between the scene's centroid to the first camera is approximately equal to 1. If the initial estimate of the i^{th} camera pose is $\{\mathsf{R}_{i,0}, \mathsf{t}_{i,0}\}$, such normalization corresponds to $\hat{\mathsf{Y}}^i_k = (\mathsf{R}_{i,0}\mathsf{Y}_k + \mathsf{t}_{i,0})/||\mathsf{t}_{i,0}||, i = 1, 2, \dots, n, k = 1, 2, \dots, p$. After this transformation, $\mathsf{R}_{i,0}$ and

$\mathrm{t}_{i,0}$ simplify to $I_{3\times3}$ and $0_{3\times1}$ respectively. We also normalize the data during the robust estimation is scaled with twice of its median value and ξ for Tukey penalty function is set to 1 whenever it is used. The iterations are terminated when the improvement of the pose between two consecutive iterations $l-1$ and l of both cameras becomes insignificant. The improvements on the rotational (R) and transnational (t) components are computed using

$$e_R = |||\mathrm{R}_l - \mathrm{R}_{l-1}||| \text{ and } e_t = cos^{-1}\left(\frac{\mathrm{t}_l^T \mathrm{t}_{l-1}}{||\mathrm{t}_l||||\mathrm{t}_{l-1}||}\right). \tag{4.17}$$

Improvements on R' and t' are also computed similarly. The algorithm terminates when $e_R < \tau_1$, $e_{R'} < \tau_1$, $e_t < \tau_2$, and $e_{t'} < \tau_2$ for some given thresholds τ_1 and τ_2.

4.5/ EXPERIMENTS

We tested our methods using both synthetic and real datasets. Our results with synthetic data were compared against those of ICP with classical SfM. For real data, experiments with four different datasets captured under different setups were performed. In all the cases, the constrained nonlinear least-squares optimization problem was solved by using MATLAB-R2012a Optimization Toolbox with interior-point method.

4.5.1/ SIMULATIONS

We generated a set of 800 random 3D points scattered on the surface of four faces of a $[-10\ 10]^3$ cube. The cameras were placed about 20 ± 2 units away from the origin with randomly generated rotations while roughly looking towards the centroid of the scene. All scene points were projected onto 256×256 images with zero-skew, 100 pix. focal length and an image-centered principal point. The 2D data were obtained by adding various levels of zero-mean Gaussian noise to the pixel coordinates. 400 out of 800 projected points were randomly selected and used to localize the second camera with respect to the first one using classical SfM. During this process, half of the points are rejected to minimize the effect of outliers thus leading to the reconstruction of only 200 points. The same data were used in our method to perform the registration and the refinement. We ran 100 tests for each noise level of standard deviation from 0 to 2.0 with a 0.25 step. The simulation results are presented for the two-view case only.

The roughly known R was generated by introducing an error of $[0.05\,0.075]^c$ in roll, pitch and yaw each. We introduced these relatively small errors in R to observe the improvement when the iterative scheme converges. Similarly, a small error of $\pm 5\%$ was introduced in each translation axis. Nevertheless, these errors are very significant since the scene is relatively far from the cameras. The histogram was built with auto adjustable 10 bins after discarding the scales of less than 0.1 and greater than twice its median. First, we obtained the best possible R, t, R′, and t′ using classical SfM [18] and ICP[111]. As ICP cannot be performed without the knowledge of relative scale, the extra information of scale is recovered with the assumption of the image-based reconstruction being spread all over the provided 3D scene. Note that, our method does not require this extra information of scale. To analyze the improvement on camera pose, we computed the deviation of these results from their ground truth values. The errors ΔR, Δt, $\Delta R'$, and $\Delta t'$ correspond to the residuals computed as in Equation (4.17). Figure 4.3 shows the Root-Mean Square (RMS) plots of the computed errors for various levels of noise. It can be seen that our method performs significantly better than SfM with ICP even when the ICP is favored with extra information of scale.

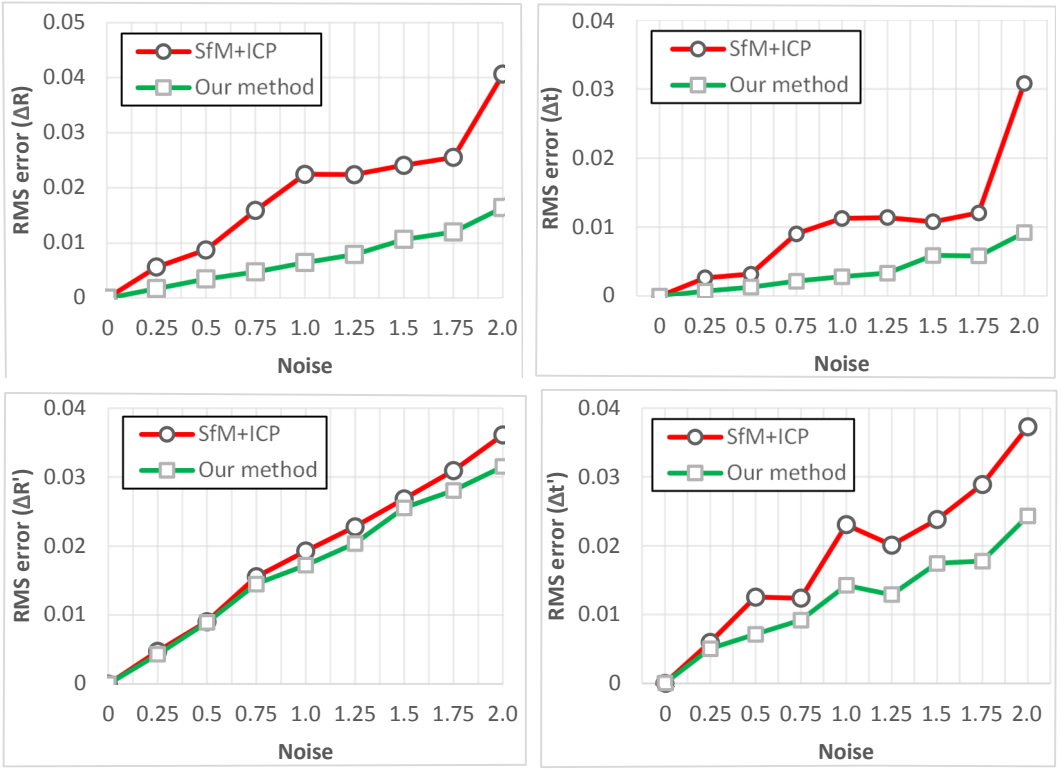

FIGURE 4.3 – SfM+ICP vs. Our method with noise ; ΔR (left-top), Δt (right-top), $\Delta R'$ (left-bottom), and $\Delta t'$ (right-bottom).

4.5.2/ Real Data

Three benchmark and one in-house real datasets were used to test the proposed algorithms. Two out of these four datasets were acquired asynchronously and the other two synchronously. Each of these datasets were acquired by very different setups as discussed below. The results obtained were compared against the ground truth (whenever available) or the known desired output. Required 2D-to-2D correspondences for all the experiments were obtained by the SURF descriptor based matching.

Asynchronous case

Scene and images were captured by two different devices. The first dataset was captured by a Kinect sensor and a separate 2D camera. The second dataset consists of two different scenes scanned by a laser-scanner and multiple images captured by a camera. Results for the second dataset were compared against the provided ground-truth values. However, the results of the first dataset were compared against the desired reconstruction.

Kinect Dataset : For the first experiment with real data, we built the prior 3D scene by registering multiple frames acquired from a 3D sensor (Kinect). This scene was then down-sampled to about 50,000 points as shown in Figure 4.4 (left). After the 3D scene is acquired, a standard-sized football was placed in the same scene and two 1080×1920 images were captured by a moving camera. These images and their 1198 correspondences are shown in Figure 4.4 and Figure 4.5. 14 manually selected points from the corners of the Truncated Icosahedron (TI) (Figure 4.5 (right)) were retained for assessing the quality of the reconstruction. To overcome the problem of initialization, the first views of both 2D and 3D cameras are captured approximately from the same location while facing towards the same part of the scene.

The final metric reconstruction of the scene is upgraded to Euclidean for the measured length of polygon sides equal to 4.5 cm. Reconstructed TI from two views is placed in the given 3D scene and shown in Figure 4.6. We have approximated the circumference of the football by fitting a sphere passing through the vertices of the reconstructed TI. For a quantitative analysis, the following geometric parameters of reconstructed TI are computed : (i) LS : RMS error of the length of sides. (ii) AH : RMS error of the internal

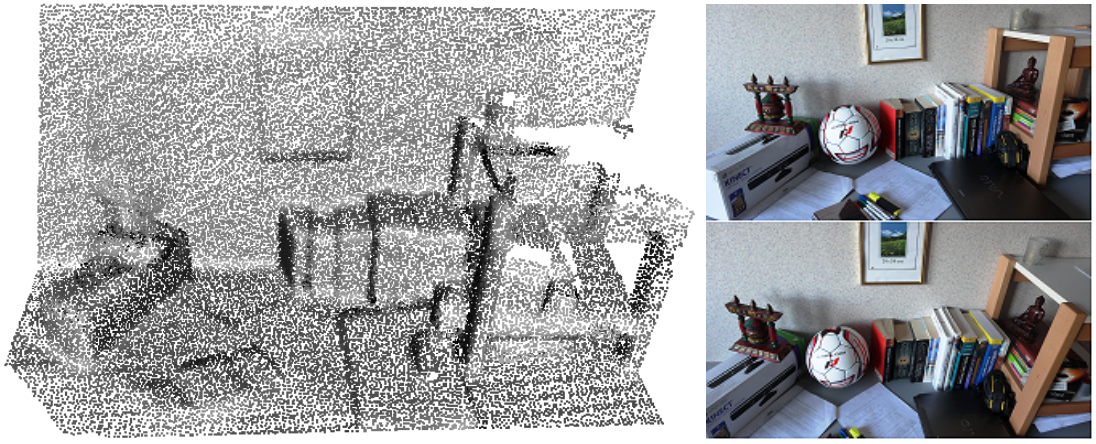

FIGURE 4.4 – Left : Kinect 3D scene ; Right : image pair.

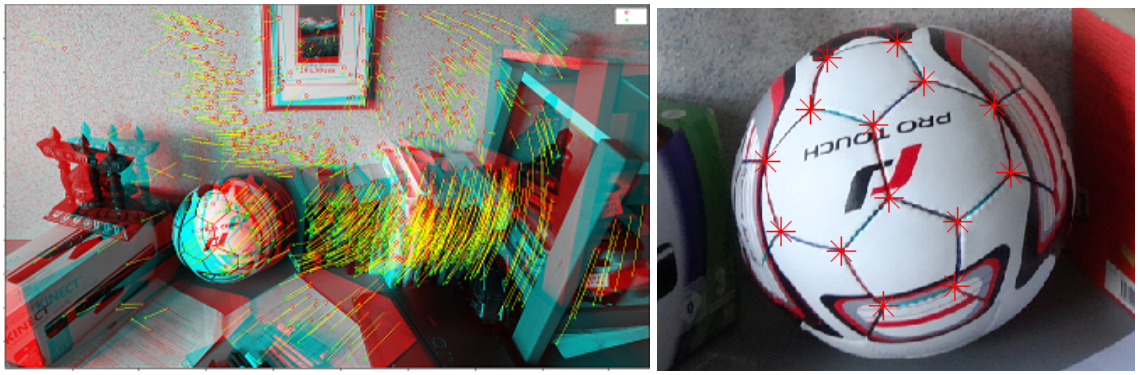

FIGURE 4.5 – Left : Correspondences ; Right : feature points.

	LS (cm)	AP	AH	A-HP	A-HH	CS (cm)
SfM	0.201	4.267	2.008	6.195	140.19	76.25
Our method	0.117	2.943	0.863	3.342	139.20	73.10

TABLE 4.2 – Geometric parameters.

angles of hexagons. (iii) AP : RMS error of the internal angles of pentagons. (iv) A-HP : RMS error of Dihedral angles between hexagons and the pentagons. (v) A-HH : Dihedral angle between two hexagons (expected : 138.19). (vi) CS : Circumference of the sphere (expected : 68-70 cm). Table 4.2 compares these parameters against FIFA's standard. This is an example of 2D-to-3D data fusion where the reconstruction from two views is added to the 3D scene. This example also demonstrates the handling of occlusion problem because of the football placed in the scene after the 3D acquisition. Furthermore, even when the 3D data is not very accurate, as it is the case in this case, it shows that our method still benefits from the scene information.

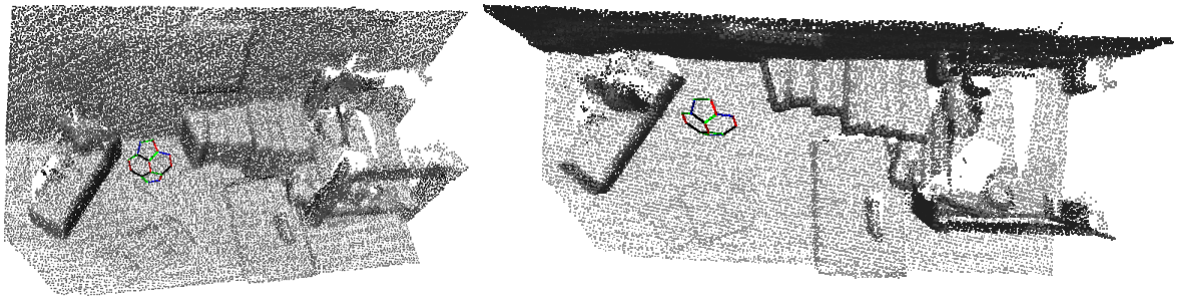

FIGURE 4.6 – Two views of the 3D scene with TI.

FIGURE 4.7 – Left : Fountain-P11 ; Right : Herz-Jesu-K7.

EPFL dataset : We also tested our method with the public datasets Fountain-P11 and Herz-Jesu-K7 (Figure 4.7 from http://cvlabwww.epfl.ch/~strecha). These datasets consist, respectively, of 11 and 7 images of size 3072×2048 along with ground truth partial 3D point clouds of the scenes. To validate the ground truth, the texture was mapped on the scene by back-projecting images using their ground truth projection matrices. Figure 4.8 shows that the provided camera poses are very satisfactory (unlike M. Corsini et al. reported in [76]). First, the 3D reconstructions for every consecutive pair of images are obtained using classical SfM. All these results are then refined separately using our method. Results before and after the refinement are compared against the ground truth in Table 4.3. The 3D errors shown here are the mean 3D RMS error of all the pairs. During the implementation, we have decimated the 3D scenes to about 50,000 points by uniform down-sampling for a faster computation. About 2000-3000 feature points were selected in each pair of views for the reconstruction.

For the multiview case, reconstructions from each consecutive pair of views are registered. Such registration undergoes error accumulation and scale factor drift. We separately refined these results using our method and sparse BA [112]. The results using our method

FIGURE 4.8 – Texture mapping of Herz-Jesu-K7.

	Method	Fountain	Herz-Jesu
$\Delta R'$(RMS)	SfM	0.0044	0.0072
	Our method	8.49e-4	0.0013
$\Delta t'$(RMS)	SfM	0.0404	0.0757
	Our method	0.0031	0.0052
3D error	SfM	0.0011	0.0025
	Our method	5.95e-4	0.0018

TABLE 4.3 – SfM vs. our method (two views).

were found to be significantly better than those of BA. We also considered refining our results using BA. Results obtained from BA, our method, and BA performed to refine our results are shown in Table 4.4. It is observed that BA performed on our results diverges from the ground truth instead of further refinement. Since BA takes only the image information into account and cannot incorporate the 3D knowledge, noise present in the image might be the reason for BA to diverge. For qualitative analysis, results obtained from BA as well as our method were used to map the texture (Figure 4.9). Texture mapping using BA contains many artifacts the most visible of which has been circled in this figure. Note that, as the scene being relatively far from the cameras, even a small error in pose can significantly affect the texture mapping. It clearly shows the pose refinement using our method is very accurate and visually no different from the ground truth.

Discussion : In 2-views case, in Table 4.2–4.3, the results are significantly different, as the proposed method uses extra knowledge of 3D during the pose refinement (because the given 3D scenes are highly accurate than that could be obtained from 2-views reconstruction). In case of multiple views, it is really difficult to conclude its significance, shown in Table 4.4. In fact, the experiments for multiple-views asynchronous cameras are

	Method	Fountain	Herz-Jesu
$\Delta R'$(RMS)	BA	0.0436	0.0123
	Our method	0.0020	0.0067
	Refined	0.0251	0.0080
$\Delta t'$(RMS)	BA	0.0311	0.0402
	Our method	0.0019	0.0224
	Refined	0.0172	0.0241
3D error	BA	0.0020	0.0069
	Our method	0.0015	0.0068
	Refined	0.0020	0.0069

TABLE 4.4 – BA vs. Our method and unsuccessful refinement of our results using Bundle Adjustment - BA (multiview).

FIGURE 4.9 – Texture mapping : Bundle Adjustment (left), our method (right).

not conducted extensively to make very strong conclusions. Although 2-views asynchronous case has extensive simulations in Figure 4.3, further investigation on asynchronous cameras in multiple-views case remains as a future work.

SYNCHRONOUS CASE

We have also tested our method using two different real and synchronous datasets. Both datasets were acquired by a moving vehicle equipped with a laser-camera system. However, these two setups greatly differ from one another.

KAIST Dataset : We conducted our first Synchronous experiment using data obtained from a Laser-Camera system dedicated to reconstructing very large outdoor structures. This system uses two 2D laser scanners and four 2D cameras which are synchronized and calibrated for both intrinsic and extrinsic parameters. Laser scanners used here provide a wide angle of view of the scanning plane so that the system can observe tall objects as well as the ground making its suitable to scan the environment from a close distance.

The 3D map (reconstruction) of the environment is made by collecting these 2D scans at their proper location. Therefore, this system requires a very precise localization for a good reconstruction. Extrinsic parameters of 2D cameras were estimated by laser points and a pattern-based calibration method. However, it still possesses a mean projection error of about 0.5 pixels. The interested reader may refer to [95] for details regarding the experimental setup. The dataset we have tested is a continuous trip of the Laser-Camera scanning system within the compound of KAIST (Korea) for a distance of about 3 KM. The system made seven different loops during its travel. The original reconstruction and the loops are shown in Figure 4.10. The lengths of the loops, as shown in Table 4.5, range from about 200 meters to 1.5 KM. Each camera captured 480×640 pix. images with a rate of about 20 frames/sec. The 2D-to-2D correspondences are computed between images escaping each 10 frames. The original reconstruction obtained by the Laser-Camera system was used as the required 3D information for our method. Note that this reconstruction was not very accurate. Nevertheless, we were still able to refine the motion using such inaccurate data.

The qualitative and quantitative results are presented in Figure 4.11 and Table 4.5 respectively. The errors were computed by performing the ICP between two point clouds captured at the loop closing point before and after the loop travel. Note that loop closing methods are not applied to the presented results. Our goal is to obtain a better localization so that it would be suitable for the loop closing methods. We strongly believe that the localization with such accuracy can be a very suitable input for loop closing. Our experiments clearly show significant improvement in loop closing errors by our method for all the loops tested. Since, most of the loop closing methods used in practice provide only the local optimal solution ; these improvements contribute to their convergence to the desired one. It can also be seen that the error reduction is independent of the loop length. In fact, the improvement is dependent upon the quality of feature points. The remaining residual error is the combined effect of the errors in calibration, matching, and measurements.

To analyze reconstruction accuracy, we fitted the surface on the reconstructed points cloud using an algorithm that we have developed in-house. This algorithm takes advantage of the camera motion and the order of scanned points. The reconstructed surface was mapped with texture from the same images that were used for localization. The textured scene with its various stages is shown in Figure 4.12 for only one side of the

Loop	Size (m)	Bok *et al.* (m)	Our method (m)
1	351.76	4.063	1.548
2	386.38	4.538	1.469
3	224.37	4.765	4.398
4	242.87	1.696	1.077
5	931.14	3.884	2.858
6	1496.4	7.182	6.381
7	546.05	5.502	2.115

TABLE 4.5 – Loop size and loop closing errors in meters for Bok *et al.* [95] and our method.

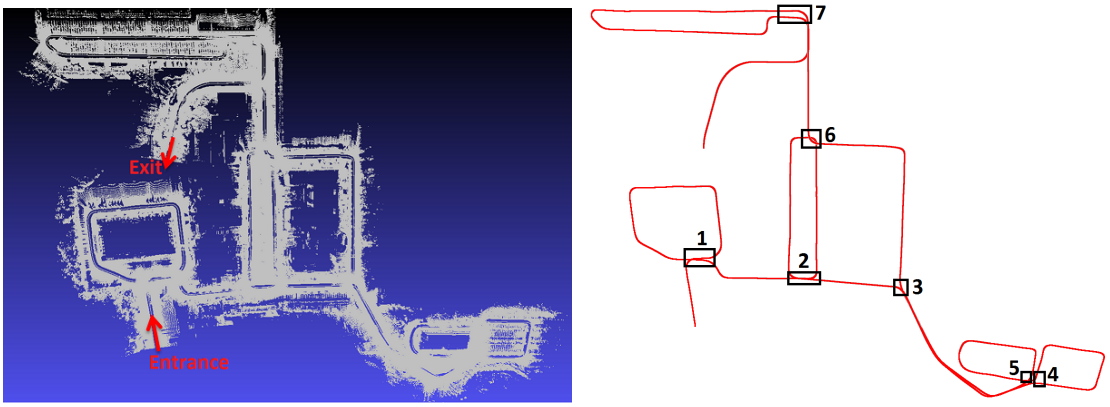

FIGURE 4.10 – Large map reconstructed using Laser-Camera system in a single trip shown with starting and end points (left). Closed loops made during the travel. Boxes shown are the loop closing locations of seven different loops (right).

reconstruction around the first loop (about 350 meters). This part of the reconstruction consists of about 1.3×10^6 3D points and 2.5×10^6 triangles.

KITTI Dataset : The proposed method was also tested on the benchmark dataset available at (http ://www.cvlibs.net/datasets/kitti/). The details of the experimental setup is described in [113]. We have used the stereo pair of gray images and the 3D data scanned from a Velodyne laser scanner. The results obtained before and after refinement for 5 different sequences were compared against the provided ground truth. Errors in rotation and translation were computed by using the evaluation code provided along with the dataset which uses the ground truth obtained using GPS and other odometry sensors. Although this ground truth might not be very accurate for local poses comparison, it is relevant over a long sequence due to no error accumulation process. Therefore, the errors were measured at the sequence steps of (100,200,...,800) and are presented in Table 4.6. Figure 4.13 shows the map obtained for the fifth sequence. A close observation shows that the localization before the refinement is already quite satisfactory. Its further refine-

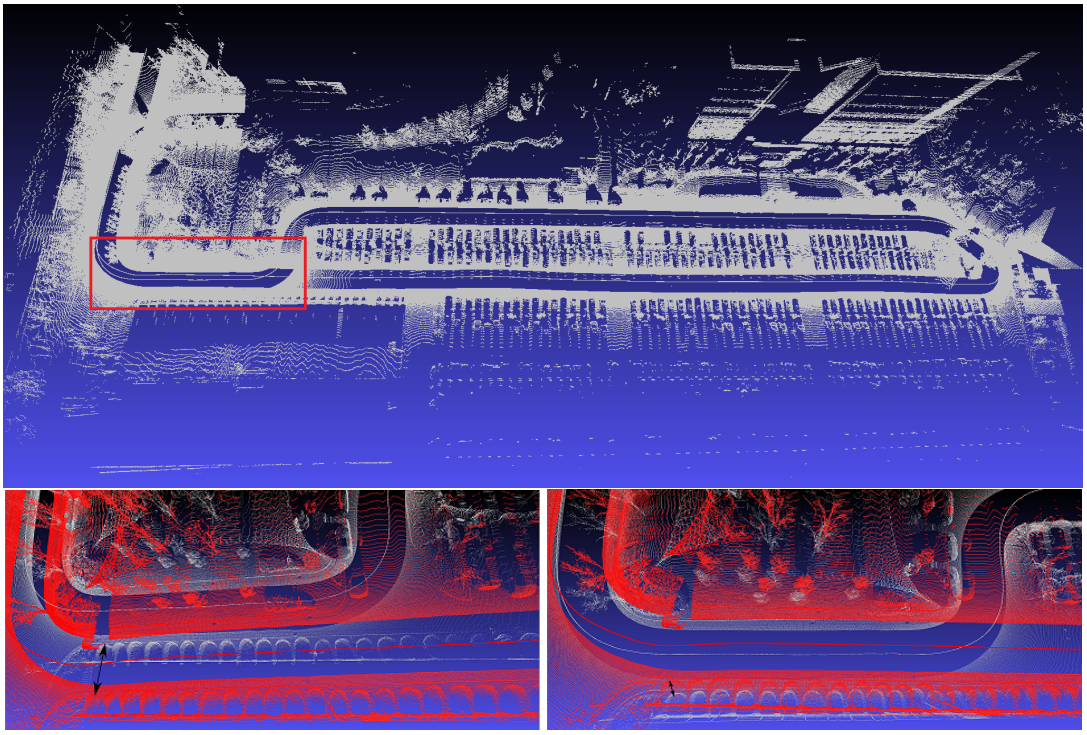

FIGURE 4.11 – Results similar to Figure 4.1 for seventh Loop. Reconstruction with a red box at the loop closing location (top), obtained using Bok *et al.* (bottom-left) and our method after refinement (bottom-right). The double sided arrows show the gap between two different reconstructions of the same scene.

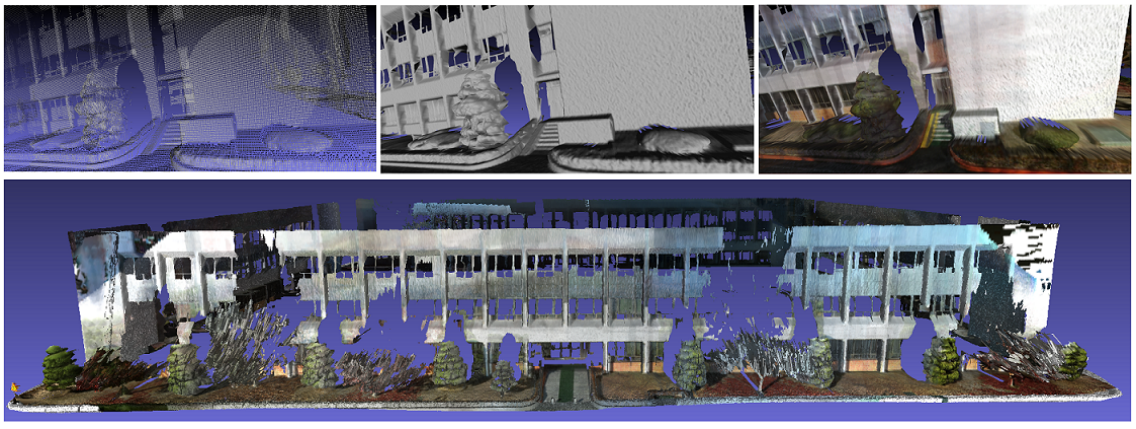

FIGURE 4.12 – Surface reconstruction and texture mapping showing the accuracy of localization. Reconstructed 3D, fitted surface, and texture mapping in a close view (top row, left to right). Texture mapping of the structure scanned around loop 1 (bottom).

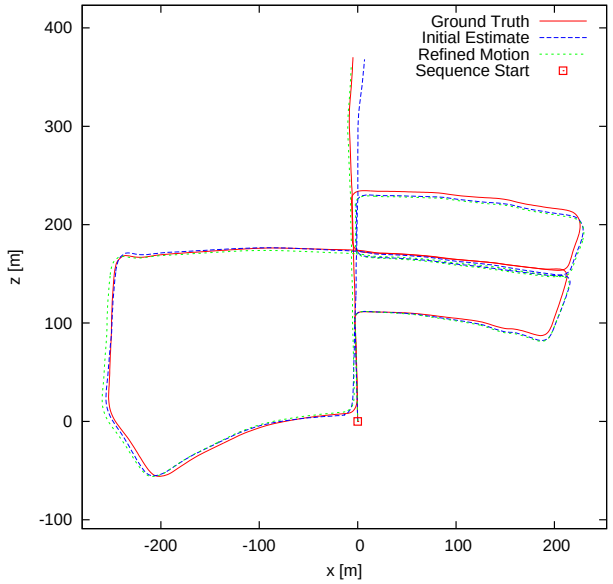

FIGURE 4.13 – Map built by our method (Initial Estimate and Refined Motion) vs. Ground Truth for the fifth sequence.

Sq.N	N.Frames	Initial Estimate		Refined	
		$\Delta T(\%)$	$\Delta R(^\circ/m)$	$\Delta T(\%)$	$\Delta R(^\circ/m)$
3	801	1.6774	0.000432	1.6398	0.000216
5	2761	1.9147	0.000245	1.8679	0.000162
7	1101	2.3410	0.000231	1.5689	0.000192
8	4071	2.3122	0.000447	1.9799	0.000196
9	1591	1.7562	0.000270	1.5604	0.000197

TABLE 4.6 – Translation (ΔT) and Rotation (ΔR) errors in Initial and Refined results for five different sequences.

ment makes the result very close to the ground truth itself. Here again, the results are presented without the loop closing.

ASYNCHRONOUS-TO-SYNCHRONOUS CASE

We also processed the results obtained by asynchronous method using the synchronous data processing algorithm. Basically, the camera poses in asynchronous case are obtained by using Equation (4.2). Starting from the obtained poses of the first set of cameras, we used Algorithm 5 under the assumption that Algorithm 4 synchronizes the camera pairs. Results obtained in each step for Fountain and Herz-Jesu sequences are shown in Table 4.7. Figure 4.14 shows cameras in the scene for one of the sequences. It can be observed that the camera poses obtained after the synchronous assumption

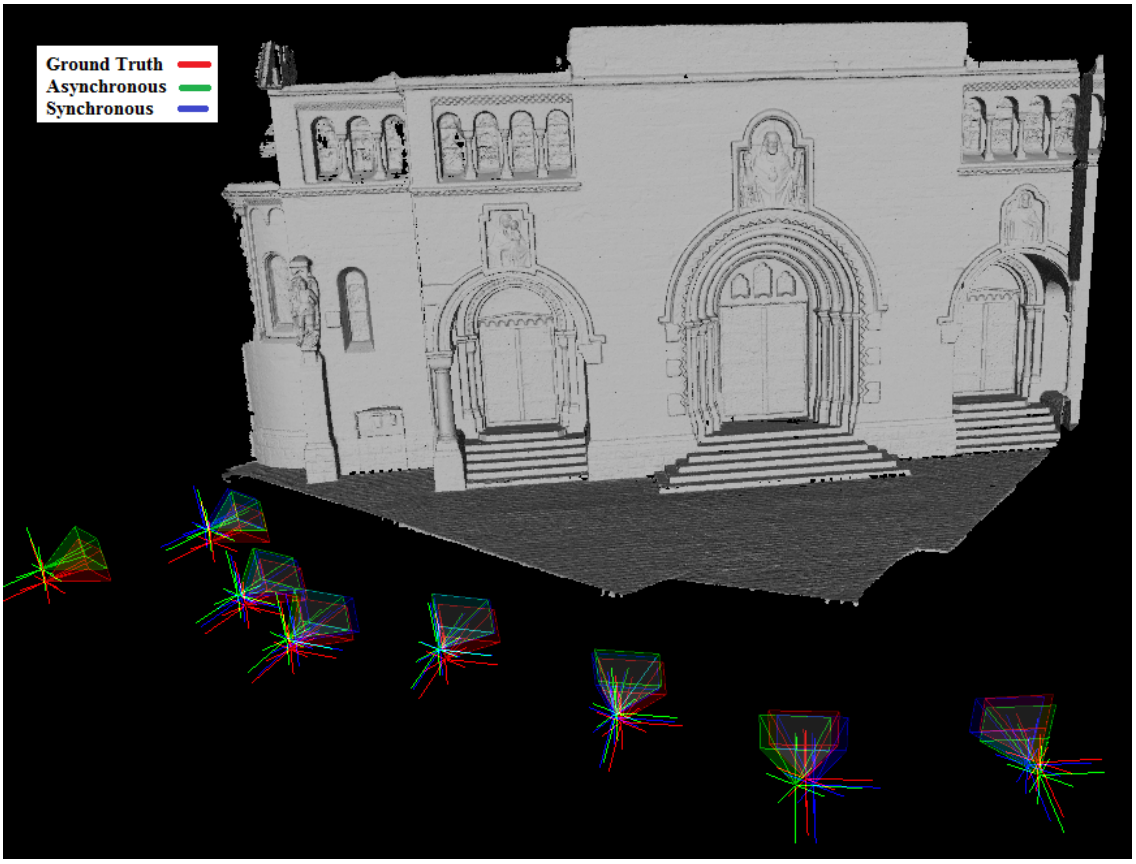

FIGURE 4.14 – Ground truth, Asynchronous-to-Synchronous Cameras in the scene.

	Asynchronous		Synchronous	
	ΔR(mean)	Δt(mean)	ΔR(mean)	Δt(mean)
Fountain-P11	0.0214	0.0074	0.0230	0.0111
Herz-Jesu	0.0222	0.0182	0.0196	0.0191

TABLE 4.7 – Error measured during Asynchronous-to-Synchronous case.

are very satisfactory. However, they are not always as good as the ones obtained from asynchronous algorithm. This happens mainly because the synchronous algorithm is relatively more sensitive to the pose gaps. In few cases, when the asynchronous algorithm does not produce results very close to ground truth, the synchronous algorithm rather deteriorates the results instead of further improvement. Nevertheless, the absolute poses obtained from the asynchronous algorithm remains unaffected.

4.6/ CONCLUSION

In this chapter, we have proposed an optimization framework to accurately localize two or more cameras in a known environment. We have demonstrated the possibility of precisely registering 2D images to 3D scene using only feature points. Usage of a known 3D scene to refine the camera pose is key to achieve such accuracy. To make it possible, a direct 2D-to-3D registration method has also been integrated in the optimization process. When the 3D scene is known, even up to some inaccuracies, it is better to use this information for the refinement of the SfM reconstruction than using general-purpose techniques such as Bundle adjustment.

5

UNCOUPLED CAMERAS

"The art of doing mathematics is finding that special case that contains all the germs of generality."

- David Hilbert, *1862 AD – 1943 AD*

In this chapter, we present a globally optimal method for 2D-3D registration in the case of uncoupled acquisition set-up with internally calibrated 2D cameras. Our method, one presented here, registers a set of images to the scene using SfM-induced reconstruction. Therefore, it belongs to the indirect 2D-3D registration methods. We search for the optimal Metric homography, similar to Equation (2.27), using RANSAC framework, defined by Equation (3.15), under the Branch-and-Bound paradigm, discussed in Section 3.3. Upper bounds required for BnB-based search are derived using Sum-of-Squares conditions, presented in Section 3.5 , whereas, the lower bounds are obtained using scaled-ICP (a local method) registration discussed in Section 2.5.

5.1/ INTRODUCTION

2D-3D registration problem for asynchronous cameras has become evident with the emergence of affordable 3D and high quality 2D cameras. Indeed, 3D cameras allow us to obtain faithful 3D scene models in the form of dense 3D point clouds while images can be used to extract texture information. High quality 3D models with mapped texture can be obtained provided the 2D and 3D cameras are registered in a common reference frame. The two imaging modalities are generally registered off-line and the 2D and 3D sensors kept rigidly attached at all time during acquisition. Doing so may, however, be

either impractical or impossible. On the one hand, suitable acquisition conditions for one sensor may not be adequate for the other (e.g. lighting conditions for cameras, surface orientation for 3D sensor, etc.) and, on the other hand, some application-specific requirements (e.g. camera on a drone and a 3D scanner on a vehicle) may altogether prohibit the sensors to be rigidly attached. When the cameras and the 3D sensor are free, reliable methods for registering the two modalities are highly desirable. This consists in establishing feature correspondences between the two modalities and estimating the rigid transformation aligning their respective reference frames.

Structure-from-Motion (SfM) techniques allow us to compute 3D point coordinates from pixel correspondences across images. It is thus tempting to regard the registration of 3D and 2D sensors as that of two 3D point sets : one set induced by the images and the other obtained from scanner measurements. Registering 3D point clouds is a well-studied problem. Most methods use the Iterative Closest Point (ICP) algorithm (or its variants) [114, 70, 115]. While ICP is a local method, recent work by *Yang et al.* [73] (Go-ICP) provides the very first globally optimal solution to same-scale point set registration. However, because SfM reconstructions suffer from a scale ambiguity, methods devised for registering same-scale data cannot be employed.

Most methods handling the scale ambiguity rely on establishing correspondences either between the 3D measurements obtained by both modalities or directly between scanned data and images [30, 116, 117]. The sought transformation parameters are then obtained either by minimizing the registration loss function or maximizing the consensus set of inliers. Note that Random Sample Consensus (RANSAC) [52] is the most widely used method for finding the maximum set of inliers. Methods based on loss function minimization are more prone to outliers than their inlier-set-maximization counterparts [118]. Some methods exploit scene knowledge or the Manhattan World assumption. In this regard, methods have been devised based on line segment matching [30], target segmentation [58], repeated patterns detection [119], mutual information maximization [120], and extended Chamfer matching [121]. Registration methods that are based on establishing correspondences may be undermined by unreliable visual feature descriptors. Alternative methods, not establishing initial correspondences, have also been proposed [76, 1, 71]. The methods in [1, 71] use variants of the ICP algorithm and hence remain susceptible to partial scene overlap, scene occlusion, and high levels of outliers. The method in [76] employs

a RANSAC-based inlier set maximization in which the scale problem is handled by an extension of the 4-point congruent sets algorithm.

As far as the problem of maximizing the set of inliers is concerned, RANSAC is non-deterministic and provides no guarantee with respect to the optimality of its solution. Globally optimal inlier set maximization methods [122, 118] have recently been proposed for problems that can be described using linear equations. However, extensions to problems with nonlinear equations [123] is problem-specific, difficult and may result in much more complicated (possibly numerically intractable) mathematical formulations. Note that a variety of methods for solving systems of nonlinear polynomial equations exist. While some are based on Gröbner bases or homotopy continuation [124], others use Sum-of-Squares (SoS) polynomial optimization [125, 126, 127]. However, such methods are dedicated to solving outlier-free nonlinear systems and dealing with outliers is carried out through RANSAC.

In this work, we address the problem of registering the 3D scan and a set of images of a structured scene captured by calibrated cameras. Our assumption is that the scene is structured in the sense that it can be segmented into and represented by planes (or planar patches). Such representation is compact [128] and can also be useful for scene knowledge-based refinement methods [98]. The plane-based assumption is particularly valid when dealing with man-made environments, including (but not limited to) Manhattan World, urban and indoor scenes that are abundant with planes. In our approach, we seek the metric transformation relating the scene's planes and SfM-induced 3D points. Note that point-to-plane registration methods are known to perform better than their point-to-point counterparts [129]. We rely on the fact that, under metric ambiguity, a point-to-plane assignment can be expressed as a second degree polynomial in scaled-quaternion and translation parameters. Our approach aims at maximizing the set of point-to-plane inliers with guaranteed optimality of the consensus set. The consensus set maximization methods [122, 118] discussed above are not applicable because of the nonlinearity of the problem at hand. In our approach, we use the Branch-and-Bound (BnB) algorithmic paradigm to explore the scaled-quaternion and translation parameter space. As in [122, 118], we rely on establishing optimistic and pessimistic sets of point-to-plane inliers for pruning branches whose most optimistic sets are worse than the best pessimistic one.

5.2/ Related Works

We address the problem of image set to 3D scene registration using an indirect approach. Given the calibrated cameras and correspondences across a set of images, we first reconstruct the 3D scene using SfM technique. The registration is then carried out by estimating the metric Homography between the reconstruction and the scene. Most methods handling the scale ambiguity rely on establishing correspondences either between the 3D measurements obtained by both modalities or directly between scanned data and images [30, 116, 117]. The sought transformation parameters are then obtained either by minimizing the registration loss function or maximizing the consensus set of inliers. Note that Random Sample Consensus (RANSAC) [52] is the most widely used method for finding the maximum set of inliers. Methods based on loss function minimization are more prone to outliers than their inlier-set-maximization counterparts [118]. Some methods exploit scene knowledge or the Manhattan World assumption. In this regard, methods have been devised based on line segment matching [30], target segmentation [58], repeated patterns detection [119], mutual information maximization [120], and extended Chamfer matching [121]. Registration methods that are based on establishing correspondences may be undermined by unreliable visual feature descriptors. Alternative methods, not establishing initial correspondences, have also been proposed [71, 76]. The methods in [71] use variants of the ICP algorithm and hence remain susceptible to partial scene overlap, scene occlusion, and high levels of outliers. The method in [76] employs a RANSAC-based inlier set maximization.

As far as the problem of maximizing the set of inliers is concerned, RANSAC is non-deterministic and provides no guarantee with respect to the optimality of its solution. Globally optimal inlier set maximization methods [122, 118] have recently been proposed for problems that can be described using linear equations. However, extensions to problems with nonlinear equations [123] is problem-specific, difficult and may result in much more complicated (possibly numerically intractable) mathematical formulations. Note that a variety of methods for solving systems of nonlinear polynomial equations exist. While some are based on Gröbner bases or homotopy continuation [124], others use Sum-of-Squares (SoS) polynomial optimization [125, 126, 127]. However, such methods are dedicated to solving outlier-free nonlinear systems and dealing with outliers is carried out through RANSAC. Note that the RANSAC based methods fail to provide the certificate for

global optimality.

5.3/ SoS Point-to-Plane Assignment Conditions

We consider a set of two or more calibrated cameras observing a scene consisting of a set $\mathcal{P}$ of at least four distinct planes in general positions. The scene has been scanned by a 3D sensor and segmented into these planes. We also consider the set $\mathcal{Y}$ of seven or more points (lying on at least four distinct scene planes) whose projections are matched across two or more cameras. Let $\mathsf{Y} \in \mathbb{R}^3$ be the SfM-induced [130] cartesian coordinate vector of a point $Y \in \mathcal{Y}$ (with change in notation for reconstructed points). The coordinates of the SfM-reconstructed points and those of the scene planes are represented in two distinct reference frames. A plane $\Pi \in \mathcal{P}$ is given by its normal 3-vector π and signed distance to the origin d. Recall that the image-induced reconstruction is metric, the transformation aligning the SfM-reconstructed points and the scanned scene is given by the metric Homography, as in Equation (2.3). We represent the sought metric Homography by a 3×3 scaled-rotation matrix Q and a translation 3-vector t (representing the inverse Homography, i.e. ${\mathsf{H}_\mathsf{M}}^{-1}$, of Equation (2.3)). A quaternion representation with no enforcement of unit quaternion $\mathsf{q} = (\begin{array}{cccc} z & u & v & w \end{array})^\mathsf{T}$ is used to represent the scaled-rotation matrix Q as follows :

$$\mathsf{Q} = \begin{pmatrix} z^2+u^2-v^2-w^2 & 2uv-2wz & 2uw+2vz \\ 2uv+2wz & z^2-u^2+v^2-w^2 & 2vw-2uz \\ 2uw-2vz & 2vw+2uz & z^2-u^2-v^2+w^2 \end{pmatrix}.$$

Let $\mathcal{A} \subset \mathcal{Y} \times \mathcal{P}$ be the set of putative point-to-plane assignments ($\times$ refers to the cartesian product) and $a = (Y, \Pi) \in \mathcal{A}$ is one such assignment. Furthermore, we denote by $\mathsf{x} \in \mathbb{R}^7$ the vector $\mathsf{x} = (\mathsf{q}^\mathsf{T}, \mathsf{t}^\mathsf{T})^\mathsf{T}$ and let $f_a(\mathsf{x})$ be the polynomial in $\mathbb{R}[\mathsf{x}]$ induced by a such that :

$$f_a(\mathsf{x}) := \pi^\mathsf{T}(\mathsf{QY} + \mathsf{t}) - d. \tag{5.1}$$

If x is the true registration parameter vector, then for every correct assignment $a \in \mathcal{A}$, $f_a(\mathsf{x}) = 0$. Our goal is to simultaneously estimate the registration parameters x and associated set of correct point-to-plane assignments. For $\mathcal{Z} = \{z_a | \ a \in \mathcal{A}\}$ be a set whose

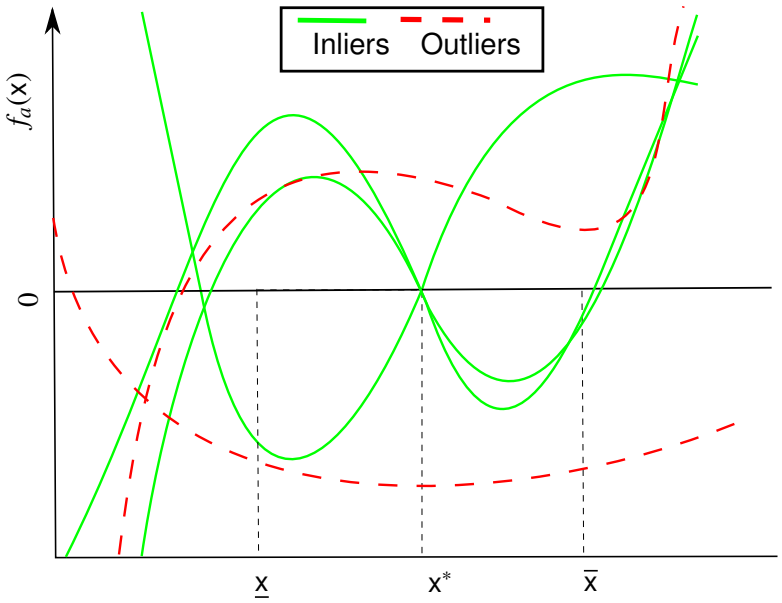

FIGURE 5.1 – Polynomials from inlier and outlier assignments. All polynomials with common zero crossing x^* are from the correct point-to-plane assignments. If any plynomial $f_a(x)$ crosses zero within the interval $[\underline{x}, \overline{x}]$, it is considered as an inlier for that interval.

entry $z_a \in \{0, 1\}$ decides whether the assignment a is an inlier or an outlier (i.e. if $z_a = 1$, assignment a is an inlier and vice versa), the desired registration can be formulated as an optimization problem as follows :

$$\begin{array}{lll} \max\limits_{\mathcal{Z},\, x} & \sum\limits_{a \in \mathcal{A}} z_a & \\ \text{subject to} & z_a f_a(x) = 0, & \text{for all } a \in \mathcal{A}, \\ & z_a \in \{0, 1\}, & \text{for all } a \in \mathcal{A}. \end{array} \tag{5.2}$$

We solve this problem using BnB algorithmic paradigm, where branching is carried out on the space of registration parameters x. At each iteration, we are given parameter intervals, in the form of two vectors $\underline{x}$ and $\overline{x}$ in $\mathbb{R}^7$ whose respective entries $\underline{x}_k$ and $\overline{x}_k$ satisfy $\underline{x}_k \leq \overline{x}_k$ for $k = 1 \dots 7$. Although the full approach is detailed further in the thesis, the idea is that such intervals are to be probed for point-to-plane potential assignments by attempting to solve the following problem :

Problem 5.3.1. *For a given $a \in \mathcal{A}$, is there a vector $x \in \mathbb{R}^7$ satisfying $\underline{x}_k \leq x_k \leq \overline{x}_k$, $k = 1 \dots 7$ such that $f_a(x) = 0$?*

In other words, one would like to know whether the polynomial crosses zero within the

considered bounds. The point-to-plane assignment would then qualify as a potential inlier, i.e. possible correct assignment, within the considered bounds. This is however difficult to answer, unless the zero crossing are searched based on the optimization variable intervals, as shown in Figure 5.1. Therefore, we consider the following alternative problem :

Problem 5.3.2. *For a given $a \in \mathcal{A}$, is there a $\lambda_a \in \mathbb{R}$ such that $\lambda_a f_a(\mathrm{x}) > 0$ for every* x *satisfying $\underline{\mathrm{x}}_k \leq \mathrm{x}_k \leq \overline{\mathrm{x}}_k \; for \; k = 1 \ldots 7$?*

If $\lambda_a f_a(\mathrm{x}) > 0$, then the assignment a is definitely an outlier, i.e. incorrect assignment, within the bounds. Otherwise, it is a potential inlier. Indeed, if the question of Problem 5.3.2 is answered in the affirmative, the one of Problem 5.3.1 is answered in the negative : i.e. there exist no x in the interval with which $f_a(\mathrm{x})$ crosses zero. Furthermore, one can rely on Putinar's Theorem 3.5.3 to solve Problem 5.3.2. To do so, assume we are given a set of polynomials $g_i(\mathrm{x})$ whose quadratic module $M(g)$ is Archimedean : if, for λ_a a scalar, $\lambda_a f_a(\mathrm{x}) > 0$ for all $\mathrm{x} \in \mathcal{K} = \{\mathrm{x} \in \mathbb{R}^7 : g_i(\mathrm{x}) \geq 0, i = 1 \ldots m\}$, then $\lambda_a f_a(\mathrm{x}) \in M(g)$. Hence, there must exist SoS polynomials σ_i such that :

$$\lambda_a f_a(\mathrm{x}) - \sum_{i=1}^{m} \sigma_i(\mathrm{x}) g_i(\mathrm{x}) \;\; \text{is} \;\; \text{SoS}. \tag{5.3}$$

Note that, in general, if Equation (5.3) is satisfied, then $\lambda_a f_a(\mathrm{x})$ may not be necessarily positive in $\mathcal{K}$ since $\mathcal{K}$ could possibly be empty. However, so long as $\mathcal{K}$ is not empty and σ_i SoS polynomials can be found, one is guaranteed that $\lambda_a f_a(\mathrm{x}) > 0$ everywhere in $\mathcal{K}$ since $\sum_{i=1}^{m} \sigma_i(\mathrm{x}) g_i(\mathrm{x}) > 0$ in $\mathcal{K}$.

There are two main pending issues before one is able to use Equation (5.3). First, one needs to find a set of polynomials $g_i(\mathrm{x})$, representative of the parameter intervals, whose quadratic module $M(g)$ is Archimedean. Second, it is so far unclear how the σ_i SoS polynomials can be found. Let us explore now the first of these issues. Note that the Archimedean property is a matter of representation and the quadratic module of the set constructed from the linear interval constraints $\mathrm{x}_k - \underline{\mathrm{x}}_k \geq 0$ and $\overline{\mathrm{x}}_k - \mathrm{x}_k \geq 0$ is not Archimedean. In the following, we show that quadratic polynomial inequalities derived from such bound constraints yield an Archimedean quadratic module.

Proposition 5.3.3. *Consider the polynomials $g_k(\mathrm{x}) = (\mathrm{x}_k - \underline{\mathrm{x}}_k)(\overline{\mathrm{x}}_k - \mathrm{x}_k)$, $k = 1 \ldots 7$. The quadratic module $M(g)$ of these polynomials is Archimedean.*

Proof As per Definition 8, for $M(g)$ to qualify as Archimedean, one must show that $N - \|\mathsf{x}\|^2 \in M(g)$ for some $N \in \mathbb{N}$. In other words, there exist SoS $\sigma_0(\mathsf{x})$ and $\sigma_k(\mathsf{x})$, $k = 1 \dots 7$, such that

$$N - \sum_{k=1}^{7} \mathsf{x}_k^2 = \sigma_0(\mathsf{x}) + \sum_{k=1}^{7} \sigma_k(\mathsf{x}) g_k(\mathsf{x}). \tag{5.4}$$

Equivalently, one needs to show that

$$N - \sum_{k=1}^{7} \mathsf{x}_k^2 - \sum_{k=1}^{7} \sigma_k(\mathsf{x}) g_k(\mathsf{x}) \text{ is SoS.} \tag{5.5}$$

Upon expanding and factorizing the latter polynomial, we obtain

$$\sum_{k=1}^{7} (\sigma_k(\mathsf{x}) - 1)\mathsf{x}_k^2 - \sum_{k=1}^{7} \sigma_k(\mathsf{x})(\underline{\mathsf{x}}_k + \overline{\mathsf{x}}_k)\mathsf{x}_k + (N + \sum_{k=1}^{7} \sigma_k(\mathsf{x}) \underline{\mathsf{x}}_k \overline{\mathsf{x}}_k). \tag{5.6}$$

Using zero-degree SoS polynomials σ_k, i.e. nonnegative real scalars, one can always find $\sigma_k > 1$ and sufficiently large value of N such that this polynomial is always positive. Notice that the polynomial is quadratic in which case PSD and SoS are equivalent [84].

Let us now consider the problem of checking whether or not Equation (5.3) is SoS when considering the polynomials $g_k(\mathsf{x})$, $k = 1 \dots 7$ of Proposition 5.3.3. If so the assignment a is definitely an outlier within the bounds. If one knows beforehand that $\lambda_a f_a(\mathsf{x})$ must be positive, a sequence of $\sigma_k(\mathsf{x})$ of increasing degree can be used until a positivity certificate is obtained. However, for the problem at hand, when a set of $\sigma_k(\mathsf{x})$ of some degree fails to deliver such certificate, it is either because $\lambda_a f_a(\mathsf{x})$ indeed crosses zero (inlier) or the required degree for a positivity certificate has not been reached. The good news here is that, within a BnB search, the considered bound intervals $[\underline{\mathsf{x}}, \overline{\mathsf{x}}]$ get smaller and we show in the following that using nonnegative scalars σ_k rather than SoS polynomials of higher degree suffices. To see this, consider the following proposition :

Proposition 5.3.4. *Let $\hat{\mathsf{x}} \in \mathbb{R}^7$ with known entries. The following statements are equivalent*
(i) $\lambda_a f_a(\hat{\mathsf{x}}) > 0$.

(ii) $\exists$ *nonnegative scalars* $\sigma_k \in \mathbb{R}$, $k = 1 \dots 7$ *:*

$$\lambda_a f_a(\mathsf{x}) + \sum_{k=1}^{7} (\mathsf{x}_k - \hat{\mathsf{x}}_k)^2 \sigma_k > 0. \tag{5.7}$$

Proof (ii) $\Longrightarrow$ (i) is straightforward. For (i) $\Longrightarrow$ (ii), consider $f_a(\mathsf{x})$'s Gram matrix G_f and G_x that of $\sum_{k=1}^{7}(\mathsf{x}_k - \hat{\mathsf{x}}_k)^2$. These matrices are defined by : $f_a(\mathsf{x}) = \mathsf{x}^\mathsf{T}\mathsf{G}_f\mathsf{x}$ and $\sum_{k=1}^{7}(\mathsf{x}_k - \hat{\mathsf{x}}_k)^2 = \mathsf{x}^\mathsf{T}\mathsf{G}_\mathsf{x}\mathsf{x}$. Note that G_x is PSD and can be written as $\mathsf{G}_\mathsf{x} = \mathsf{U}^\mathsf{T}\mathsf{U}$ with $\mathsf{U}\hat{\mathsf{x}} = 0$. The Gram matrix of the polynomial in Equation (5.7) is then written as $\lambda_a \mathsf{G}_f + \mathsf{U}^\mathsf{T} diag(\sigma_1, \sigma_2, \dots, \sigma_7)\mathsf{U}$. A direct application of Finsler's lemma [82] is that the latter matrix is positive-definite if and only if $\lambda_a \hat{\mathsf{x}}^\mathsf{T}\mathsf{G}_\mathsf{f}\hat{\mathsf{x}} > 0$. This not only shows (i) $\Longrightarrow$ (ii) but also proves the equivalence.

We now state the following preliminary result :

Result 5.3.5. *Consider two vectors* $\underline{\mathsf{x}}$ *and* $\overline{\mathsf{x}}$ *in* $\mathbb{R}^7$ *whose respective entries* $\underline{\mathsf{x}}_k$ *and* $\overline{\mathsf{x}}_k$ *satisfy* $\underline{\mathsf{x}}_k \leq \overline{\mathsf{x}}_k$ *for* $k = 1 \dots 7$*. Let* $\mathcal{K}_b$ *be the set*

$$\mathcal{K}_b = \{\mathsf{x} \in \mathbb{R}^7 : g_k(\mathsf{x}) := (\mathsf{x}_k - \underline{\mathsf{x}}_k)(\overline{\mathsf{x}}_k - \mathsf{x}_k) \geq 0\}. \tag{5.8}$$

If $\exists$ *a scalar* λ_a *and nonnegative scalars* σ_k *such that*

$$\lambda_a f_a(\mathsf{x}) - \sum_{k=1}^{7} g_k(\mathsf{x})\sigma_k \quad \textit{is SoS}, \tag{5.9}$$

then $\lambda_a f_a(\mathsf{x}) > 0$ *for every* $\underline{\mathsf{x}}_k \leq \mathsf{x}_k \leq \overline{\mathsf{x}}_k$*. In this case, the assignment* a *is guaranteed to be an outlier (a point-to-plane mismatch) within the considered bounds. Otherwise,* a *is a potential inlier. Furthermore, a consequence of Proposition 5.3.4 is that when* $\overline{\mathsf{x}}_k - \underline{\mathsf{x}}_k$ *tends towards zero, we are guaranteed that any outlier within the bound is detected. Indeed, this can be seen by noticing that when* $\overline{\mathsf{x}}_k = \underline{\mathsf{x}}_k = \hat{\mathsf{x}}_k$*, polynomial (5.9) turns into (5.7).*

Whether (5.9) is SoS can be tested by converting it into its corresponding Gram matrix LMI feasibility problem for the λ_a and σ_k indeterminates. Although the guarantee of identifying outliers using scalar σ_k multipliers is demonstrated with a zero-gap bound, in practice, outliers are detected very early in the process. As demonstrated in our experiments, the ability to detect outliers is improved with every size reduction of the investigated bounds. It may be tempting to use higher degree $\sigma_k(\mathsf{x})$ SoS polynomials to boost the process. However, this is unnecessary and yields slower performances compared to

branching.

5.4/ REGISTRATION

Our goal is to register a SfM-induced reconstruction and a plane-segmented scene. Unlike when dealing with 3D-3D registration, additional constraints emanating from the cameras can be exploited. Some may be implicit such as plane visibility, others, as vague camera locations, may be obtained from extra knowledge. In addition, when dealing with segmented scenes, one is given planar patches rather than infinite planes. Such additional constraints can augment the set $\mathcal{K}_b$ derived from the bound constraints for earlier outlier detection. Note that adding new polynomials inequalities in $\mathcal{K}_b$ has no effect on the Archimedean property of its quadratic module and Proposition 5.3.4 still holds.

Patches : Consider a scene plane Π and three or more planes Φ_k, not necessarily from the scene, orthogonal to it. The Φ_k planes must be chosen such that their intersection with Π defines a convex region on Π. The set of points on Π within this convex region is a patch. In practice, four such planes are adequate to represent meaningful patches in man-made environments. Each Φ_k is described by its normal vector ϕ_k and signed distance d_k. Let us denote by Φ the set $\{\Phi_k\}_{k=1}^4$ and let $\delta_k = \pm 1$ be the known sign, with respect to Φ_k, of a scanned point lying within the considered region. One can then identify outliers by checking whether $f_a(\mathsf{x})$ is positive everywhere within $\mathsf{x}'s$ bounds and in the set

$$\mathcal{K}_a^{\Phi} = \{\mathsf{x} \in \mathbb{R}^7 : p_k(\mathsf{x}) := (\phi_k^{\mathsf{T}}(\mathsf{Q}\mathsf{y} + \mathsf{t}) - d_k)\delta_k \geq 0, k = 1 \ldots 4\}. \tag{5.10}$$

Plane visibility : Consider a point Y (not lying on the corner) on a scene plane Π. If this point is imaged by two cameras, then these can only observe the same side of the plane : the one on which the point lies. In order for the cameras to observe the same side of the plane, their camera centers must lie on one side with respect to Π. Camera centers can easily be obtained from the SfM-calculated camera matrices : they are their right null space. Let C_k be the camera centers of $n \geq 2$ cameras with cartesian coordinates c_k. We define the set $\mathcal{K}_{\Pi}^{\delta}$ such that

$$\mathcal{K}_{\Pi}^{\delta} = \{\mathsf{x} \in \mathbb{R}^7 : v_k(\mathsf{x}) := (\pi^{\mathsf{T}}(\mathsf{Q}\mathsf{c}_k + \mathsf{t}) - d)\delta \geq 0, k = 1 \ldots n\} \tag{5.11}$$

where $\delta = \pm 1$. We denote $\mathcal{K}_\Pi^+$ the set $\mathcal{K}_\Pi^\delta$ obtained using $\delta = +1$ and $\mathcal{K}_\Pi^-$ otherwise. A given assignment a is a definite outlier if $f_a(\mathsf{x}) > 0$ in $\mathcal{K}_\Pi^+$ and in $\mathcal{K}_\Pi^-$ (in addition to patch and bounds conditions). Furthermore, planes for which $v_1(\mathsf{x})$ and $v_2(\mathsf{x})$ (for two cameras 1 and 2) always have opposite signs within $\mathsf{x}'s$ bounds cannot be assigned any points visible in those cameras. This would indicate that the plane always cuts the base-line of the two camera and cannot contain points visible in both cameras. Testing this can be carried out by checking, for $\delta = \pm 1$, whether

$$\begin{cases} \exists \sigma_k : \; v_1(\mathsf{x}) - \sum_{k=1}^{7} g_k(\mathsf{x})\sigma_k \; is \; SoS \\ \exists \sigma_k : -v_2(\mathsf{x}) - \sum_{k=1}^{7} g_k(\mathsf{x})\sigma_k \; is \; SoS \end{cases} \tag{5.12}$$

If for both values of δ, each polynomial in Equation (5.12) is SoS, plane Π shall not be considered for assigning SfM points emanating from those cameras.

Camera bounds : A camera center C may lie within a box delimited by six planes in the set $\Psi = \{\Psi_k\}_{k=1}^6$ defined by their normal vectors ψ_k and signed distances d_k. Such information can be obtained from application-specific knowledge (GPS, moving vehicle, etc.). This knowledge can be used for further enforcing the search for point-to-plane outliers and turns very useful when no putative point-to-plane correspondences are initially known. Consider the cartesian coordinate vector c of the camera center and let

$$\mathcal{K}_c = \quad \{\mathsf{x} \in \mathbb{R}^7 : h_k(\mathsf{x}) := (\psi_k^\intercal(\mathsf{Qc} + \mathsf{t}) - d_k)\delta_k \geq 0, k = 1 \ldots 6\} \tag{5.13}$$

where δ_k is the known sign, with respect to Φ_k, of any point within the considered box. If $h_k(\mathsf{x})$ are positive, the camera center is within the box. One can now test if $\lambda_a f_a(\mathsf{x}) > 0$ whenever the camera center is in the box defined by $\mathcal{K}_c$.

Quaternions and scale : In the absence of scale indeterminacy, quaternion parameters demand that $\mathsf{q}^\intercal\mathsf{q} = 1$. When dealing with a scaled scene, the rotation is represented by a scaled quaternion matrix and one can only enforce that $\mathsf{q}^\intercal\mathsf{q} > 0$. It is understood that, in order to keep the problem numerically tractable via the Archimedean property, all registration parameters need to be bounded. The scale of the scene is no exception. When a better lower bound $\underline{s} > 0$ on the scale s is available, it is preferable to enforce that $\mathsf{q}^\intercal\mathsf{q} \geq \underline{s}$. This condition does not appear in the set $\mathcal{K}_b$ and hence must be accounted for. Assuming the entries x_k, $k = 1 \ldots 4$ of x correspond the quaternion parameters, we

consider the set

$$\mathcal{K}_q = \{\mathsf{x} \in \mathbb{R}^7 : q(\mathsf{x}) := -\underline{s} + \sum_{k=1}^{4} \mathsf{x}_k^2 \geq 0\} \tag{5.14}$$

Furthermore, since both q and −q yield the same rotation matrix, the initial lower bound of one of the quaternion parameters may arbitrarily be chosen nonnegative. The rest of the quaternion parameters may be initially bounded between $-\sqrt{\overline{s}}$ and $\sqrt{\overline{s}}$ where $\overline{s}$ is the scale's upper bound.

We now state our main result :

Result 5.4.1. *Assume we are given a putative point-to-plane assignment $a = (Y, \Pi) \in \mathcal{A}$, a patch on Π delimited by the planes in the set $\Phi = \{\Phi_k\}_{k=1}^4$, lower $\underline{\mathsf{x}}$ and upper $\overline{\mathsf{x}}$ bounds on the registration parameter vector x, bounds $\underline{s}$ and $\overline{s}$ on the scale of the scene, and (optionally) bounds defined by planes $\Psi = \{\Psi_k\}_{k=1}^6$ on the location of the camera centers of one (possibly more) camera. One would like to know whether or not the SfM-reconstructed point Y may lie on Π, while Π is visible by the cameras observing Y, within the patch Φ with registration parameters in the bounds $\underline{\mathsf{x}}$ and $\overline{\mathsf{x}}$. In order to establish whether such assignment is possible, we consider the set*

$$\mathcal{K} = \{\mathsf{x} \in \mathbb{R}^7 : \mathsf{x} \in \mathcal{K}_b \cap \mathcal{K}_a^{\Phi} \cap \mathcal{K}_{\Pi}^{\delta} \cap \mathcal{K}_c \cap \mathcal{K}_q) \tag{5.15}$$

resulting from the intersection of all the sets defined by (5.8),(5.10),(5.11), (5.13) and (5.14). If there exist a scalar λ_a and nonnegative scalars σ_k, σ'_k, σ''_k, σ'''_k and σ such that

$$\lambda_a f_a(\mathsf{x}) - \sum_{k=1}^{7} g_k(\mathsf{x})\sigma_k - \sum_{k=1}^{4} p_k(\mathsf{x})\sigma'_k - \sum_{k=1}^{n} v_k(\mathsf{x})\sigma''_k - \sum_{k=1}^{6} h_k(\mathsf{x})\sigma'''_k - q(\mathsf{x})\sigma \tag{5.16}$$

is SOS, then $\lambda_a f_a(\mathsf{x}) > 0$ in $\mathcal{K}$ and the assignment a is a definite outlier. It is a potential inlier otherwise. Recall that this can be solved as a LMI feasibility problem.

Our registration approach is based on Result 5.4.1. We use in the following the term point-to-plane to refer to both point-to-plane and point-to-patch assignments. The goal of the BnB algorithm is to estimate the registration parameters yielding the largest number of inliers. Our algorithm is provided a set of putative point-to-plane correspondences. In the absence of such correspondences, we consider every point to be putatively assigned to all the planes. A dynamically-built search tree, whose nodes are registration parameters' bounds, allows to explore the space of parameters. Given point-to-plane assignments

and bounds on the registration parameters, the algorithm (see Algorithm 1) estimates the optimistic number of potential inliers using Result 5.4.1. A local method, a variant of the scaled-ICP algorithm [71] (please refer, point-set-based registration in Section 2.5), is used to obtain a pessimistic number of inliers for each given node. The local algorithm is started in the mid-values of the registration parameters' bounds. It's variation from [71] resides in constraining the registration solution to be within the investigated bounds in order to be representative of the node. We keep track of the highest (bestPessimistic in Algorithm 1) of the pessimistic number of potential inliers over all bound intervals. Any node whose optimistic number of inliers is worse than bestPessimistic is rejected. Otherwise, the node is qualified and branched along its longest edge resulting in two new nodes to be processed. The node corresponding to the bestPessimistic number of inliers is processed first. The algorithm terminates when no node has an optimistic number of inliers that is better than bestPessimistic.

Algorithm 6 Node processing

Input : bestPessimistic
Bounds on the registration parameters
Output : bestPessimistic

1. Compute the Optimistic number of inliers using Result 5.4.1.
2. If Optimistic $<$ bestPessimistic, reject the bounds.
3. Compute the Pessimistic number of inliers using a local method.
4. If bestPessimistic $<$ Pessimistic, then bestPessimistic $\leftarrow$ Pessimistic.

To qualify a point as an inlier, we distinguish two cases :

1. Putative point-to-plane correspondences are provided : a point qualifies as a potential inlier if (5.16) is not proven SoS when assigned to the considered plane.
2. No putative correspondences are provided : the point is considered a potential inlier as soon as (5.16) is not proven SoS when the point is assigned to one plane.

Discussion : In general, our method converges while the explored bounds are still quite large. The solution maximizing the inlier consensus set is the one returned by the local method. When the bounds are large enough, polynomials constructed from noisy data would still cross zero within the bounds allowing inliers, although affected by noise, to be accounted for. Therefore, the robustness to noise is more influenced by the local method

than it is by the SoS tests. In our implementation, no special care was taken to further deal with noise when using SoS tests. However, in the case of highly noisy data, the proposed SoS framework may allow to deal more efficiently with noise by incorporating an extra bounded variable ϵ (bounded by the allowed threshold), accounting for noise, in each point-to-plane assignment polynomial $f_a(\mathrm{x})$. In other words, though the assignment polynomial does not cross zero at the sought solution, $f_a(\mathrm{x}) + \epsilon$ (for some value of ϵ) would. Furthermore, we have assumed throughout that the camera information fed to our algorithm is, to some extent, reliable. Should incorrect/noisy information about a camera be used, it may cause, especially with small camera bounding boxes, the registration to fail. In such cases, it is advised to include the camera-to-box constraints in the set of assignments to be accounted for when maximizing the consensus set.

5.5/ EXPERIMENTS

We conducted experiments with seven different benchmark real datasets shown in Figure 5.2 and whose details can be found in [131] and [132]. Our algorithm was implemented in MATLAB2014b and the SoS problems solved using the LMI Control Toolbox. All experiments were carried out on a 8GB RAM Pentium i7/3.40GHz. The SfM reconstructions and segmented scene planes were obtained using the openMVG Toolbox [133] and Hough Transform based plane detector [128]. For all the experiments, the initial bound on reconstruction scale was set to 0.2–5.0 (five times in scale in both directions). Four different error measurement metrics were defined to evaluate the registration quality : the RMS 3D error on normalized point sets, errors in rotation R, translation t, and scale s. For N experiments, these are defined as follows :

$$\Delta R = \sqrt{\frac{1}{3N}\sum_{i=1}^{N} ||\mathrm{r}_i^* - \mathrm{r}||^2}, \Delta T = \sqrt{\frac{1}{N(||\mathrm{t}||^2)}\sum_{i=1}^{N} ||\mathrm{t}_i^* - \mathrm{t}||^2}, \Delta S = \sqrt{\frac{1}{N(s^2)}\sum_{i=1}^{N} (s_i^* - s)^2},$$

where r is a vector obtained by stacking three rotation angles in degrees. The estimated variables are represented with * and variables without it are their ground truth.

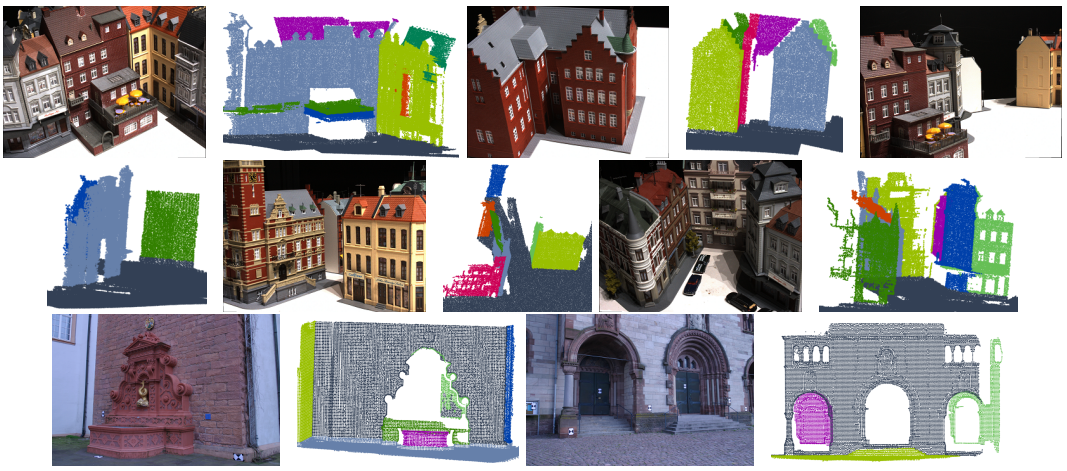

FIGURE 5.2 – Sample image and corresponding segmented scene next to each other shown in different colors for each plane (In order : Scene23, Scene24, Scene27, Scene29, Scene73, Fountain, and Herz-Jesu).

5.5.1/ INLIER SET MAXIMIZATION WITH CORRESPONDENCES

The method was first tested for known putative correspondences where the synthetic inliers/outliers were generated under real data setups. No bounds on cameras were used in these experiments. To test the robustness, we varied the number of outliers up to 90% for Scene73 and compared the results against the linear 12-point RANSAC. Figure 5.3 shows that our method consistently detects 21 inliers for every experiment while RANSAC fails to detect the least number of required inliers starting from 45% of outliers. Note that the numbers of inliers reported here are true-positive inliers. Furthermore, our method does not detect any false positive inliers. Figure 5.5(left) shows the errors in rotation, translation, and scale for the same scene with various levels of outliers. The convergence graph of our method with 50% outliers is shown in Figure 5.5(right) for Scene23, Scene73, and Fountain whose quantitative results are shown in Table 5.1. Figure 5.4 shows the evolution of the volume and the number of nodes remaining to be processed for the first 50 iterations on Scene23 with 50% outliers. The qualitative results for scene73 is shown in Figure 5.6.

5.5.2/ INLIER SET MAXIMIZATION W/O CORRESPONDENCES

In the absence of initial correspondences, each point was assigned to all available planes. We conducted several experiments with bounded cameras by changing the number of

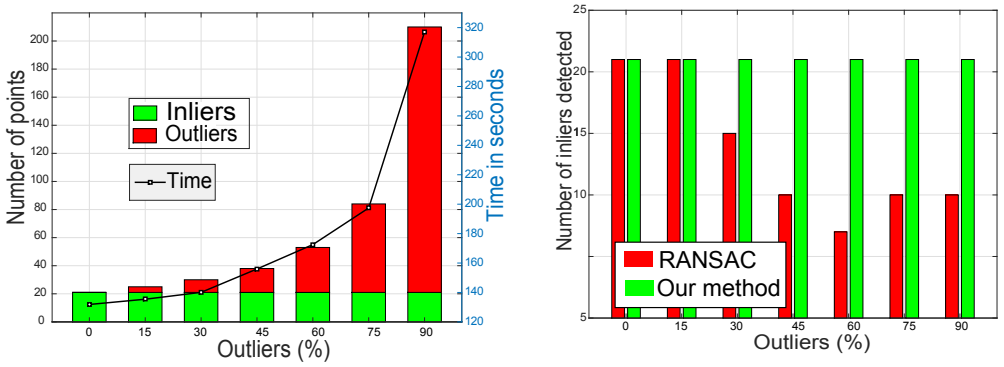

FIGURE 5.3 – Experiments on Scene73 with correspondences and no camera bounds. Left : no. of processed points and time taken for various levels of outliers. Right : no. of detected inliers using RANSAC and our method.

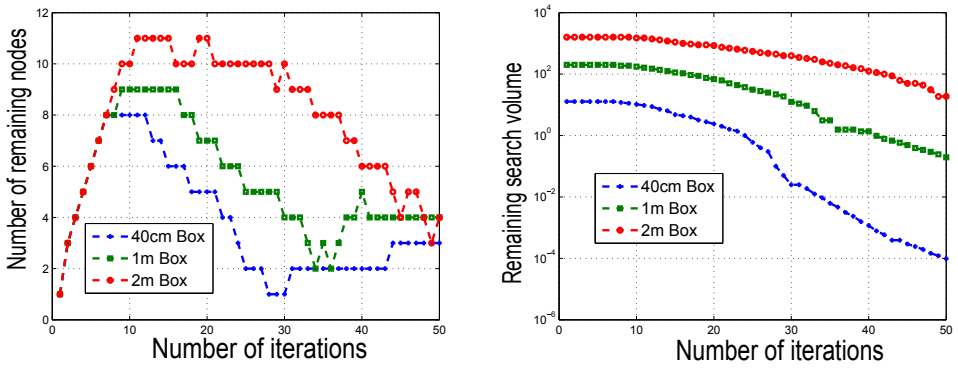

FIGURE 5.4 – Experiments on Scene23 with correspondences, no camera bounds, and 50% outliers. Left : remaining nodes. Right : remaining volume.

	ΔR (degree)	ΔT (%)	ΔS (%)	3D error	Time (sec)
Scene23	0.785	1.75	0.21	0.0163	168.95
Scene73	1.263	4.63	1.68	0.0219	153.39
Fountain	0.524	1.21	0.53	0.0056	546.41

TABLE 5.1 – Experiments with correspondences and no camera bounds : quantitative results obtained with 50% outliers.

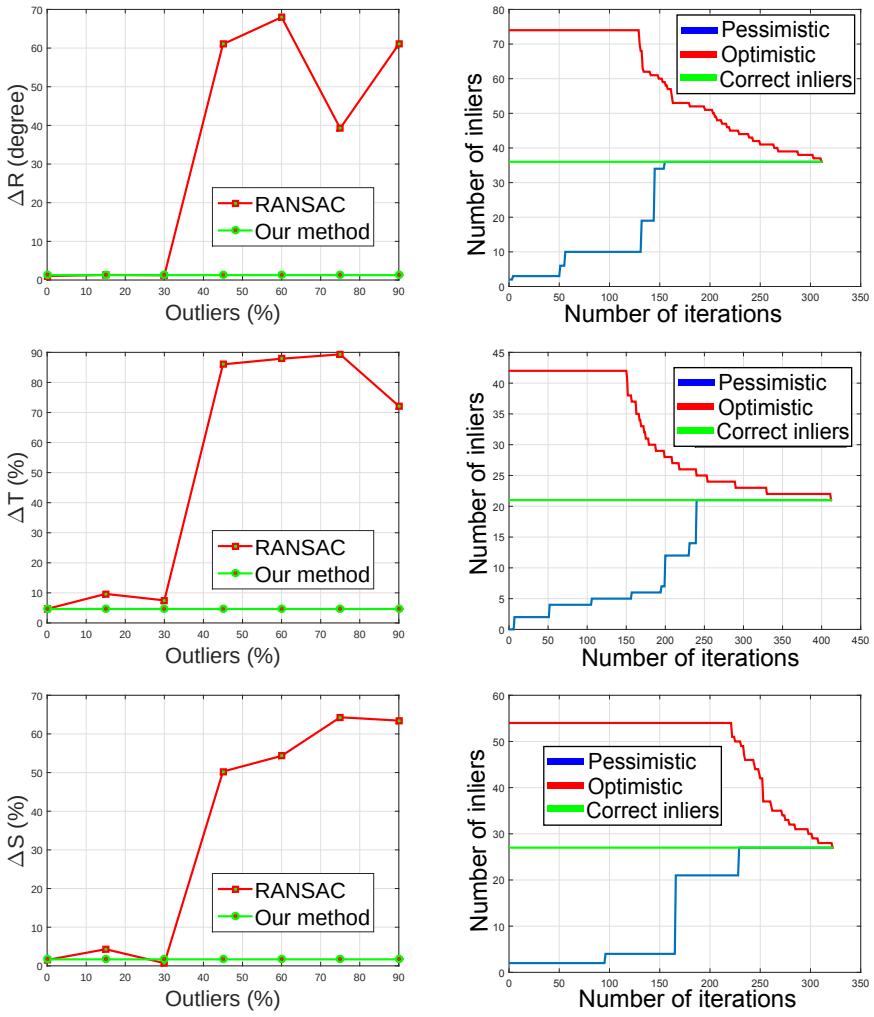

FIGURE 5.5 – Experiments with correspondences and no camera bounds. Left : Error in rotation, translation, and scale for Scene73 vs. no. of outliers. Right : Convergence graph for 50% outliers (Top to bottom : Scene23, Scene73, Fountain).

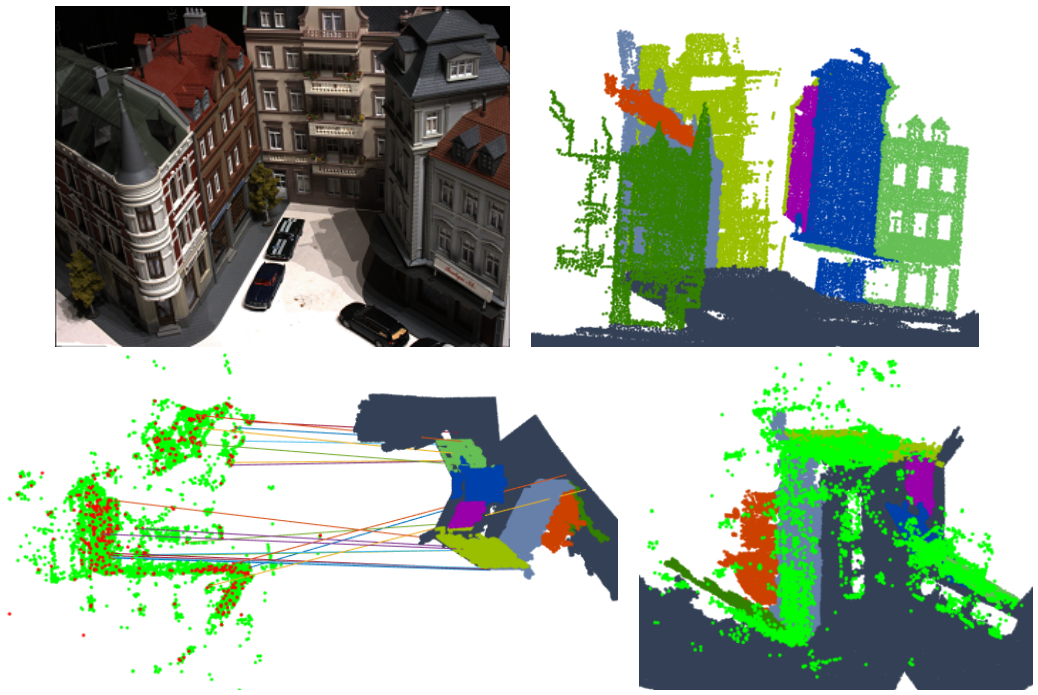

FIGURE 5.6 – Top : Sample image and segmented scene ; bottom : point-to-plane correspondences (left) and registered reconstruction (in green) and scene .

bounded cameras and camera bounding box size. The number of iterations taken for these configurations are shown in Figure 5.7(left) for Scene23. The average time per iteration is 1.15sec. In the same figure, we also provide the number of iterations taken for the “with correspondences” case with 50% outliers and 2m camera bounding boxes. The case of a single bounded camera is equivalent to unbounded cameras but bounded translation : plane visibility criterion cannot be used in this case. We recall that initial bounds on all the registration parameters are indispensable to ensure an Archimedean quadratic module of the constraints set and hence employ Putinar’s Psatz. Figure 5.7(right) shows the convergence graph, using Scene23, obtained with 3 1m-box bounded cameras. It also shows how the residual error on the registration parameters varies with the increase in the number of pessimistic inliers. The reported box size is for a normalized scene size of about 10 meters. In Figure 5.8, we report the results obtained on Scene23 (with 3 1m-box bounded cameras) using our method and a randomly started scaled ICP (RS-ICP) for 100 independent trials. In each trial, the scaled ICP was started at randomly picked registration parameter values satisfying bound and visibility constraints. The results show that, unlike RS-ICP which provides very large 3D errors, our method consistently detects the same number of inliers with the same 3D error.

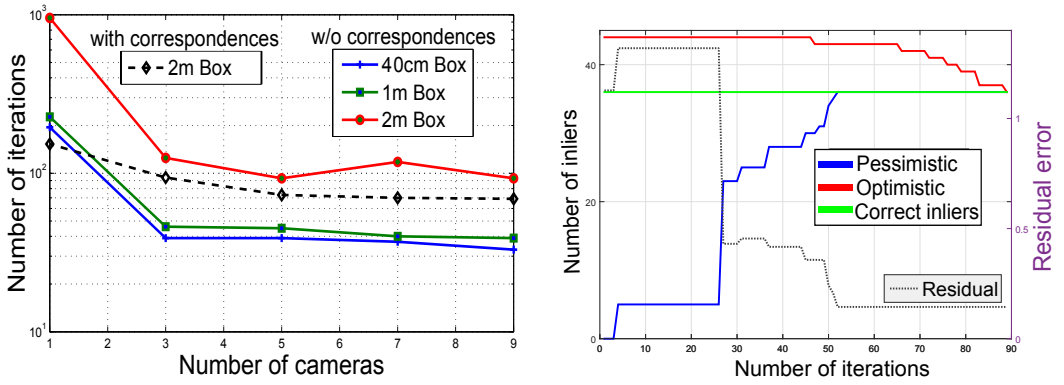

FIGURE 5.7 – Experiments on Scene 23. Left : no. of iterations vs. no. of cameras. Right : Convergence graph for the case w/o correspondences with 3 1m-box bounded cameras.

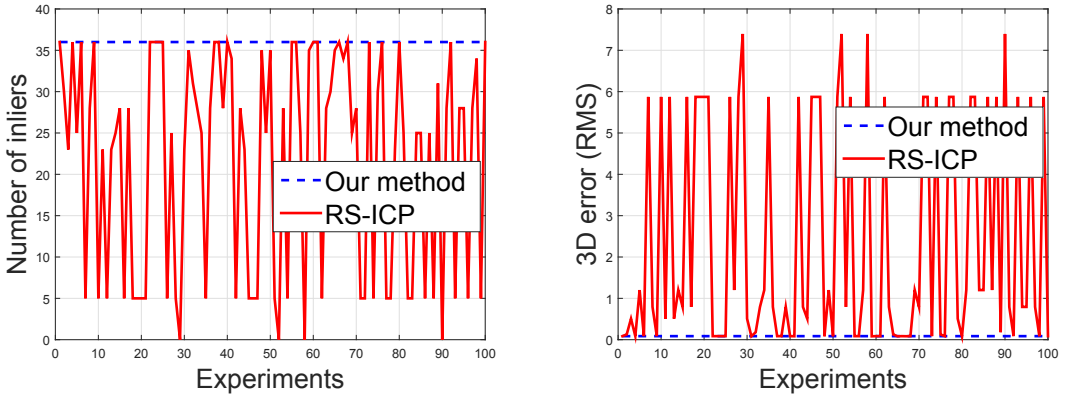

FIGURE 5.8 – Experiments on Scene23 w/o correspondences with 3 1m-box bounded cameras (100 independent trials). Our method vs. randomly started scaled ICP (RS-ICP). Left : no. of inliers detected. Right : 3D RMS error.

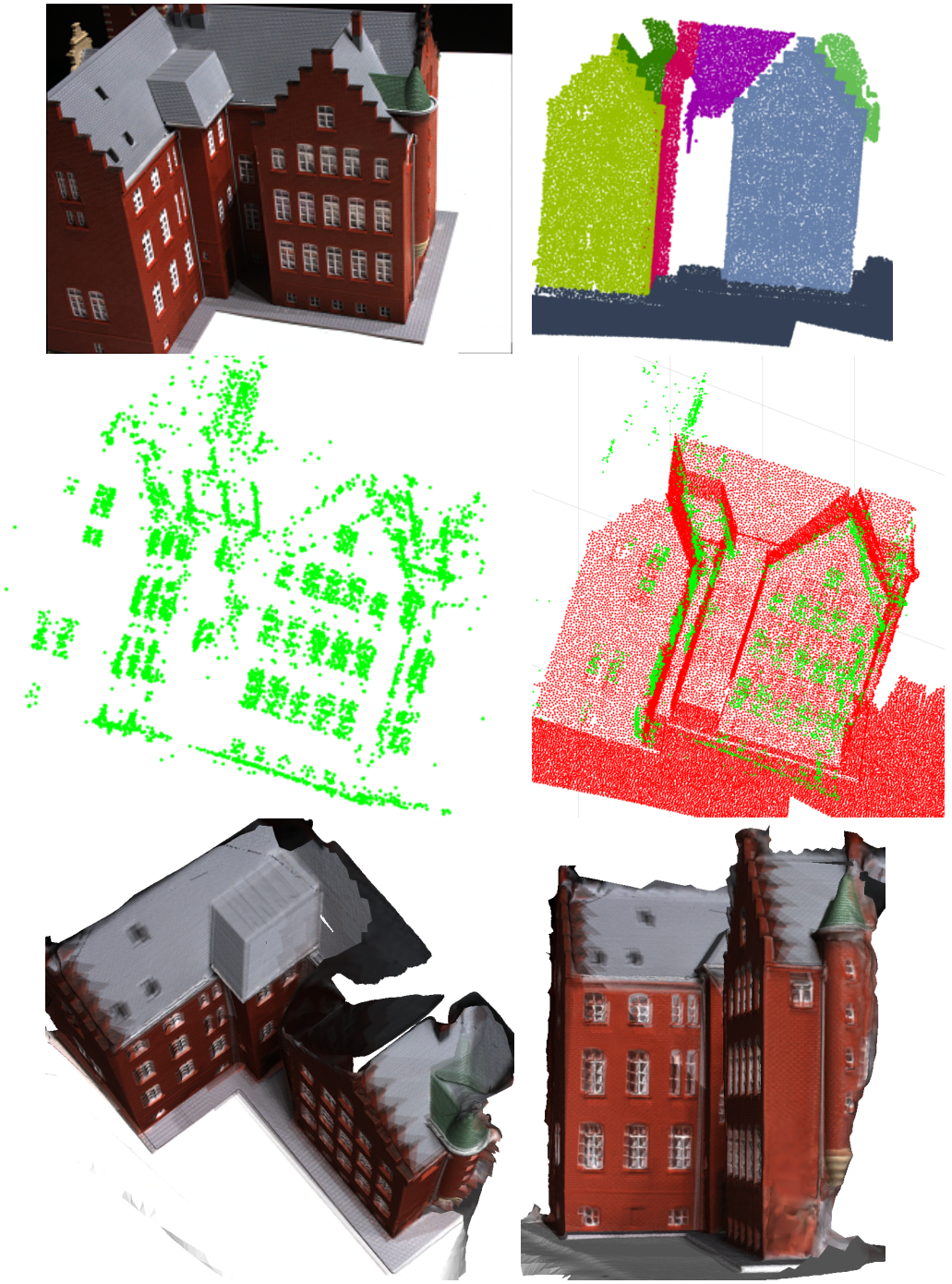

FIGURE 5.9 – Top : Sample image and segmented scene ; middle : reconstruction, and registered pointsets ; bottom : two views of texture mapped scene for Scene24.

Scene	Points	Planes	Recon.	Rep.	Camera	Box	Inlier	ΔR	$\Delta T(\%)$	$\Delta S(\%)$	3D error	Iter	Time (sec)
Scene23	90	8	2.08134	0.8373	3	2m	41	3.4147	1.95	2.48	0.0619	482	599.738
Scene24	47	7	2.62791	0.8756	3	40cm	31	0.9591	2.31	2.09	0.0424	81	51.572
Scene27	49	4	1.61906	0.8127	8	50cm	20	2.5759	3.96	1.68	0.0131	133	141.837
Scene29	90	8	1.77408	0.9226	5	1m	45	2.9995	5.41	2.41	0.081	209	277.351
Scene73	71	8	3.21913	0.8654	5	1m	45	3.3463	4.78	2.73	0.0654	223	271.226
Fountain	29	7	0.81293	0.8495	4	40cm	11	2.8639	3.18	4.74	0.0570	102	55.730
Herz-Jesu	129	5	2.08134	0.6402	8	40cm	101	7.1958	4.02	1.99	0.0464	103	137.766

TABLE 5.2 – Experiments w/o correspondences : quantitative results for seven different datasets.

Scene	Method	Time (sec)	ΔR	$\Delta T(\%)$	3D error
Fountain	RISAG	805.680	8.6825	14.08	0.3275
	Go-ICP	529.415	0.7225	1.63	0.0348
	Our method	55.730	2.8639	3.18	0.0570
Herz-Jesu	RISAG	160.064	17.6378	5.70	0.1830
	Go-ICP	31.254	3.2618	16.9	0.0725
	Our method	137.766	7.1958	4.02	0.0464

TABLE 5.3 – Results using RISAG, Go-ICP and our method.

The results of our method for all scenes (with their corresponding configurations) are summarized in Table 5.2. In the reported parameters, Points, Planes, Iter, and Inlier represent their numbers. “Recon.” is the quality of the SfM reconstruction measured as the median reprojection error in pixels while “Rep.” is the fraction of the scene points represented by fitted planes. Observe that the registration quality depends upon the reconstruction quality, representation factor, and the number and size of the camera boxes. For a qualitative evaluation, the results obtained for Scene24 are shown in Figure 5.9 along with the registered point sets and textured scene (after further refinement using [56]).

We also provide the results for two datasets obtained using RISAG [76], Go-ICP [73], and our method in Table 5.3. Our method was used without correspondences in the setting given in Table 5.2. Note that Go-ICP requires an Euclidean reconstruction, which was obtained by upgrading the metric reconstruction using ground truth measurements. Comparison of these methods may be unfair because each requires different initial conditions. Note that the poor performance of RISAG could be due to its RANSAC-driven nature (we used 10^4 RANSAC iterations). Nevertheless, both RISAG and Go-ICP were conducted in their favorable conditions.

5.6/ CONCLUSION

We proposed a method for registering a 3D scan and a set of images of a structured scene represented by planes (or planar patches). Using the Branch-and-Bound algorithmic paradigm and SoS theory, we were able to devise a robust and optimal method for inlier set maximization of point-to-plane correspondences. Although the problem at hand is nonlinear and combinatorial, our method has provided outstanding results in terms of robustness : it worked with as many as 90% outliers. In the absence of initial assignments, the proposed method still remains non-combinatorial and can incorporate additional constraints that arise from plane visibility criterion and optional vague constraints on the positions of the camera. The optimization framework used in our approach has the potential to be efficiently applied to several other nonlinear problems in Computer Vision.

6

UNCALIBRATED CAMERAS

"Projective geometry is all geometry."

- Arthur Cayley, *1821 AD – 1895 AD*

In this chapter, we present a globally optimal method for 2D-3D registration in the case of uncoupled acquisition set-up with uncalibrated 2D cameras. We address this problem by directly registering two or more uncalibrated 2D images to the scene. The proposed approach assumes the cameras only known in some arbitrary projective frame as discussed in Section 2.4. Our solution is based on a Linear Matrix Inequality framework presented in Section 3.4. We assume that the readers are familiar with cheirality conditions of Equation (2.20) and Branch-and-Prune search paradigm discussed in Section 3.3.

6.1/ INTRODUCTION

We investigate the problem of registering a scanned scene, represented by Euclidean 3D point coordinates, and two or more uncalibrated cameras. An unknown subset of the scanned points have their image projections detected and matched across images. The proposed approach assumes camera matrices to be calculated in some arbitrarily chosen projective frame and no calibration or autocalibration is required. We argue here that camera calibration may turn out to be impractical due to possible changes in the cameras' internal geometry when zooming and focusing. As for camera autocalibration, although globally convergent methods [134, 135, 136, 137] do exist, the process fails for numerous critical motions of the cameras and is generally sensitive to 2D pixel localization errors. When cameras are uncalibrated, the transformation relating the cameras to the scene is

projective. Our proposed registration solution is based on a Linear Matrix Inequality (LMI) framework that allows simultaneously estimating this unknown projective transformation and establishing 2D-3D correspondences without triangulating image points. The proposed LMI framework allows both deriving triangulation-free LMI cheirality conditions and establishing putative correspondences between 3D volumes (boxes) and 2D pixel coordinates. Directly using raw 2D points in lieu of triangulated 3D points is believed to yield more accurate motion computation [138]. In practice, triangulation results are rather uncertain in the depth direction. Using a small set of such reconstructed points for alignment may have a devastating effect on the results [100].

Two registration algorithms, one exploiting the scene's structure and the other concerned with robustness, are presented. Both algorithms employ the Branch-and-Prune paradigm and guarantee convergence to a global solution under some mild initial bounding conditions. Our algorithms require initial box-2D correspondences with 5 non-overlapping boxes to guarantee convergence to a global solution. Alternatively, non-overlapping bounds on camera centers can also be used. Finding initial bounds on camera positions is relatively easy as far as hand-held or GPS-equipped cameras are concerned. The results of our experiments, on both simulated and real data, are also presented.

6.2/ Background and Notations

Recall the notations used for uncalibrated geometry in Section 2.4. Here, we briefly discuss the concepts of triangulation and Cheirality for uncalibrated reconstruction. These concepts will later be used derive the LMI conditions for direct 2D-3D registration.

Triangulation : Any point X_j can be triangulated in a 3D coordinate frame given camera matrices and 2D pixel correspondences $\{x_j^i\}_{i=1}^n$ across images. So long as at least two 2D points are matched in at least two images, if a x_j^i is unknown in one given image (no corresponding feature point detected and/or matched in that image), it can safely be replaced by the null vector without prejudice for what follows. Let S_j be the $3n \times 3n$ block-diagonal matrix

$$\mathsf{S}_j = \begin{bmatrix} [\ \mathsf{x}_j^1\ ]_\times & 0 & \dots & 0 \\ 0 & [\ \mathsf{x}_j^2\ ]_\times & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & [\ \mathsf{x}_j^n\ ]_\times \end{bmatrix} \tag{6.1}$$

with matrices $[\ \mathsf{x}_j^i\ ]_\times$, $i = 1 \dots n$, on the diagonal blocks and zeros elsewhere. $[\ \mathsf{x}_j^i\ ]_\times$ denotes the 3×3 skew-symmetric matrix associated with the cross-product and constructed using the projection x_j^i of X_j on camera P^i. Let M be the $3n \times 4$ matrix obtained by stacking all camera matrices :

$$\mathsf{M}^\intercal = [\ \mathsf{P}^{1\intercal}\ \mathsf{P}^{2\intercal}\ \mathsf{P}^{3\intercal} \dots\ \mathsf{P}^{n\intercal}]. \tag{6.2}$$

The coordinate vector of X_j can then be obtained by solving $\mathsf{S}_j\mathsf{M}\mathsf{X}_j = 0$ to least squares. Note that matrix $\mathsf{S}_j\mathsf{M}$ must be of rank-3, or else assumed to be enforced as such throughout this Chapter.

Cheirality : As far as the true Euclidean camera matrices and 3D points are concerned, the depth of any scene point, relative to a camera in which it is visible, must be positive. The sign of this depth is referred to as the *cheirality* of the point with respect to the considered camera [130, 139]. However, in addition to the projective ambiguity, projective points and cameras are each retrieved up to a different unknown scale generally not preserving cheirality. It is possible though to assign signatures $\zeta_i = \pm 1$ to cameras and signatures $\eta_j = \pm 1$ to points to ensure that : (i) each point has a consistent cheirality with respect to all cameras in which it is visible, and (ii) all points have a consistent cheirality with respect to any one camera in which they are visible.

Camera signatures : Let X be a point visible in camera P. The cheirality of X with respect to any camera P^i in which it is also visible must be identical to that of its cheirality with respect to P. This can be enforced by considering the signatures ζ and ζ_i of, respectively, P and P^i such that

$$(\zeta \mathsf{P}\mathsf{X})_3(\zeta_i \mathsf{P}^i\mathsf{X})_3 > 0 \text{ for } X \text{ visible in } P \text{ and } P^i. \tag{6.3}$$

Note that (6.3) can be used to deduce the signature of one camera given the signature of the other. Indeed, this can be done by initially assigning an arbitrarily chosen signature ζ to one given camera P and iteratively assigning signatures to all cameras observing X. Every P^i with assigned signature can in turn be used to deduce signatures of cameras

sharing visible points with it. A robust version of such algorithm may be found in [28].

Point signatures : Correcting the signatures of cameras suffices to enforce identical cheirality for any given point in all the views in which it is visible. It, however, remains that any two points X and X_j, visible in the same camera P, may have different cheiralities, $(\mathrm{PX})_3(\mathrm{PX}_j)_3 < 0$, with respect to that camera. To make such points share the same cheirality relative to one such camera, one seeks the signatures η and η_j of these points such that

$$(\eta\mathrm{PX})_3(\eta_j\mathrm{PX}_j)_3 > 0 \text{ for } X \text{ and } X_j \text{ visible in } P. \tag{6.4}$$

Using (6.4), one may arbitrarily assign a signature η to one of the points X and recover the signatures of the remaining visible points. Once a signature is assigned to a point, it can be used to assign signatures to points visible in other views.

Cheirality inequalities : Note that (6.3) allows to assign signatures to cameras independently from the homogeneous representation of the considered visible points. Likewise, signatures are assigned to points through (6.4) independently from the camera signatures. However, this suffices to guarantee that the cheirality of any point to be identical with respect to all cameras in which it is visible. It also guarantees that all points visible by one camera carry the same cheirality with respect to it. As in the Euclidean frame, once signatures are assigned to cameras and points, the plane at infinity Π_∞ neither cuts through the convex hull of scene points nor does it cut through the convex hull of camera centers. The projective coordinates of Π_∞ must satisfy :

$$\eta_j\Pi_\infty^\mathsf{T}\mathsf{X}_j > 0 \text{ for } j = 1\ldots m, \tag{6.5}$$

$$\delta\zeta_i\Pi_\infty^\mathsf{T}\mathsf{C}_i > 0 \text{ for } i = 1\ldots n \tag{6.6}$$

for some $\delta = \pm 1$. Note that the coordinate vectors C_i referred to in (6.6) ought to be obtained exactly through the identity $\mathsf{C}_i^\mathsf{T}\Pi = \det([\mathsf{P}^{i\mathsf{T}} \mid \Pi])$ for some 4-vector Π.

Upgrade : The plane at infinity plays a key role in upgrading a projective reconstruction to its Euclidean or affine counterpart. For instance, the Euclidean coordinates X_j^E of points X_j and Euclidean camera matrices $\mathsf{P}^{\mathsf{E}i}$ of P^i, satisfying $\mathsf{x}_j^i \sim \mathsf{P}^{\mathsf{E}i}\mathsf{X}_j^\mathsf{E}$, are only a projective transformation, say H, away from their projective counterparts : $\mathsf{X}_j^\mathsf{E} \sim \mathsf{H}\mathsf{X}_j$ and $\mathsf{P}^{\mathsf{E}i} \sim \mathsf{P}^i\mathsf{H}^{-1}$. The full-rank 4×4 matrix H is the matrix representation of H. Unless the cameras are calibrated and their pose calculated, H is unknown. However, the last row of H is the homogeneous coordinate vector Π_∞ of the plane at infinity in the projective frame. If the

latter is known, for arbitrarily chosen remaining rows of H, points at infinity in the true scene are mapped back onto the canonical plane. In this case, the scene and cameras are said to be reconstructed in an affine frame. Π_∞ being generally unknown, one may use a surrogate plane, say $\widetilde{\Pi}_\infty$ whose coordinate vector $\tilde{\Pi}_\infty$ in the projective frame satisfies (6.5) and (6.6). The resulting reconstruction is then said to be quasi-affine with respect to the considered points and camera centers.

Further notations : Additional notations are used throughout the paper : the canonical vectors are denoted e_k, $k = 1,2,3$, such that $e_1 = (1\ 0\ 0)^\intercal$, $e_2 = (0\ 1\ 0)^\intercal$ and $e_3 = (0\ 0\ 1)^\intercal$. The superscript * refers to the symmetric part of a square matrix. For example, the symmetric part $Q^\star$ of a square matrix Q is given by $Q^\star = \frac{1}{2}(Q + Q^\intercal)$.

6.3/ LMI-BASED 2D-3D REGISTRATION

In this section, we first introduce a set of LMI and bounding conditions that constitute the backbone of our 2D-3D registration algorithms. The proposed algorithms are also presented in this section. We consider the scene imaged by a sequence of uncalibrated cameras and scanned by a 3D sensor. In addition to 2D point correspondences across images, the scanned scene points are given by their Euclidean coordinates X_j^E, $j = 1 \dots m$. In the absence of Euclidean-to-projective (3D-3D) point correspondences and Euclidean-to-image (3D-2D) point correspondences, the scanned points are an unknown projective transformation away, $X_j^E \sim HX_j$, from the image-induced 3D points X_j. Recall, Equation (2.17), hereafter we refer H_E of Equation (2.17) by H only. Note that H can be linearly calculated if 3D-3D point correspondences are available. It can also be estimated from 3D-2D point correspondences via $x_j^i \sim P^i H^{-1} X_j^E$. It goes without saying that, if H is known, then the correspondences can be established. However, when neither H nor 3D-3D or 3D-2D correspondences are known, the problem is particularly challenging and difficult to solve. Our goal is precisely to simultaneously establish such unknown correspondences and estimate H while using only 2D pixel coordinates and the Euclidean coordinates of the scanned points : i.e. without triangulating image points in 3-space. Once the correspondences established and H estimated, the Euclidean matrices P^{Ei}, camera pose and internal calibration parameters can be extracted. Our proposed solution heavily depends upon finding a surrogate plane at infinity $\widetilde{\Pi}_\infty$ that wouldn't cross the scene and

cameras. This however traditionally requires the so-called cheirality inequalities involving image points to be triangulated. Therefore, prior to presenting our registration conditions and methods, we first provide a LMI formulation of the cheirality inequalities for obtaining such "quasi-affine" plane without triangulating image points.

6.3.1/ CHEIRALITY LMIS

Consider a point X visible in camera P. The signature ζ of P and that of any camera P^i in which X is also visible must satisfy (6.3). Note that (6.3) can be rewritten as $\zeta\zeta_i \mathsf{X}^\intercal \mathsf{P}^\intercal \mathsf{e}_3 \mathsf{e}_3^\intercal \mathsf{P}^i \mathsf{X} > 0$. One can only notice that the latter inequality is equivalent to $\zeta\zeta_i \mathsf{X}^\intercal (\mathsf{P}^\intercal \mathsf{e}_3 \mathsf{e}_3^\intercal \mathsf{P}^i)^\star \mathsf{X} > 0$ when employing the symmetric part of the involved matrix. Finsler's lemma can then be used to deduce the LMI

$$\exists \gamma^i : \zeta\zeta_i (\mathsf{P}^\intercal \mathsf{e}_3 \mathsf{e}_3^\intercal \mathsf{P}^i)^\star + \gamma^i (\mathsf{SM})^\intercal \mathsf{SM} > 0 \tag{6.7}$$

for X visible in P and P^i. In (6.7), γ^i is a scalar and matrices S and M are constructed as in (6.1) and (6.2) from the image projections of point X and camera matrices. Note that LMI (6.7) is equivalent to (6.3). It allows to correct the signature of a camera given the signature of another camera. Unlike (6.3), LMI (6.7) does not require triangulating any point X. As in (6.3), an arbitrary signature ζ can initially be assigned to camera P and every matrix whose signature is recovered can be used to deduce the signatures of other cameras.

An alternative to (6.4) would be to enforce that all points X_j visible in some camera P^i have positive cheirality, i.e. $\zeta_i \eta_j \mathsf{e}_3^\intercal \mathsf{P}^i \mathsf{X}_j > 0$ as demanded when using the true Euclidean points and cameras. From (6.5), one can deduce that η_j and $\mathsf{X}_j^\intercal \Pi_\infty$ must carry the same sign. Because Π_∞ is homogeneous, we can choose the plane at infinity such that $\zeta_i \mathsf{X}_j^\intercal \Pi_\infty \mathsf{e}_3^\intercal \mathsf{P}^i \mathsf{X}_j > 0$. The latter inequality remains true when considering the symmetric part of the matrix involved : $\mathsf{X}_j^\intercal (\zeta_i \Pi_\infty \mathsf{e}_3^\intercal \mathsf{P}^i)^\star \mathsf{X}_j > 0$. Using Finsler's lemma and accounting for homogeneity, we deduce that LMI

$$(\zeta_i \Pi_\infty \mathsf{e}_3^\intercal \mathsf{P}^i)^\star + \gamma_j^i (\mathsf{S}_j \mathsf{M})^\intercal \mathsf{S}_j \mathsf{M} > \mathtt{I} \tag{6.8}$$

must hold for any point X_j visible in P^i for some scalar γ_j^i and the true Π_∞. Given the signatures of all cameras obtained via (6.7), LMI (6.8) is an equivalent alternative to using (6.5)

to calculate a "quasi-affine" plane $\widetilde{\Pi}_\infty$, satisfying (6.8), not cutting through the convex-hull of visible points. Unlike (6.5), LMI (6.8) neither requires the calculation of point signatures nor does it require the reconstruction of the observed points in 3-space. A surrogate plane at infinity $\widetilde{\Pi}_\infty$ can be obtained by solving the LMIs (6.8) along with inequalities (6.6) (with $\delta = \pm 1$) for all cameras and visible points.

6.3.2/ BOUNDING LMIS

Definition 9 : Positive vs. negative sides of a plane

Consider a plane Π with Euclidean coordinate vector Π^{E}. We say that a point X with coordinates X^{E} in this frame lies on the positive side with respect to this plane if and only if $\mathsf{X}^{\mathsf{E}\intercal}\Pi^{\mathsf{E}}_\infty\Pi^{\mathsf{E}\intercal}\mathsf{X}^{\mathsf{E}} > 0$. The coordinate vector $\Pi^{\mathsf{E}}_\infty = (0\ 0\ 0\ 1)^\intercal$ is that of the plane at infinity in the Euclidean frame. Points on the negative side with respect Π satisfy $\mathsf{X}^{\mathsf{E}\intercal}\Pi^{\mathsf{E}}_\infty\Pi^{\mathsf{E}\intercal}\mathsf{X}^{\mathsf{E}} < 0$.

Definition 10 : Boxing a point

Let $\mathcal{B} = \{(\underline{\Pi}_k, \overline{\Pi}_k)\}_{k=1}^3$ be a set of three pairs of planes with Euclidean coordinate vectors $\underline{\Pi}^{\mathsf{E}}_k = (\mathsf{e}_k^\intercal \ -\underline{d}_k)^\intercal$ and $\overline{\Pi}^{\mathsf{E}}_k = (\mathsf{e}_k^\intercal \ -\overline{d}_k)^\intercal$ such that the signed distances $\underline{d}_k$ and $\overline{d}_k$ of the planes to the origin of the frame satisfy $\underline{d}_k < \overline{d}_k$. Without loss of generality, the normal vectors e_k of the planes are assumed to be the canonical basis vectors. We say that a point X *is boxed by* $\mathcal{B}$ if for each pair $(\underline{\Pi}_k, \overline{\Pi}_k)$, X is on the positive side with respect to $\underline{\Pi}_k$ and negative side with respect to $\overline{\Pi}_k$.

Let S_j be the matrix constructed as in (6.1) from 2D matches. Let M^{E} (resp. M) be, as in (6.2), the stack of Euclidean (resp. projective) camera matrices $\mathsf{P}^{\mathsf{E}i}$ (resp. P^i) , $i = 1 \ldots m$. Based on the above definitions, the following corollary can be directly deduced from Finsler's lemma.

Corollary 6.3.1. *A point X_j projecting onto x^i_j in cameras $\{P^i\}_{i=1}^n$ is boxed by $\mathcal{B}_j = \{(\underline{\Pi}_k, \overline{\Pi}_k)\}_{k=1}^3$ if and only if the following LMIs are simultaneously feasible for some scalars $\underline{\gamma}_{jk}$ and $\overline{\gamma}_{jk}$:*

$$(\Pi^{\mathsf{E}}_\infty\underline{\Pi}^{\mathsf{E}\intercal}_k)^\star + \underline{\gamma}_{jk}(\mathsf{S}_j\mathsf{M}^{\mathsf{E}})^\intercal\mathsf{S}_j\mathsf{M}^{\mathsf{E}} > 0 \ k = 1,2,3 \tag{6.9}$$

$$\overline{\gamma}_{jk}(\mathsf{S}_j\mathsf{M}^{\mathsf{E}})^\intercal\mathsf{S}_j\mathsf{M}^{\mathsf{E}} - (\Pi^{\mathsf{E}}_\infty\overline{\Pi}^{\mathsf{E}\intercal}_k)^\star > 0 \ k = 1,2,3. \tag{6.10}$$

Remark 6.3.2. *Note that if any of LMIs (6.9) and (6.10) is feasible for some γ_{jk}, then the same LMI is also feasible for any $\gamma > \gamma_{jk}$. Hence, one can seek a single γ simultaneously satisfying (6.9) and (6.10) rather than six scalars $\underline{\gamma}_{jk}$, $\overline{\gamma}_{jk}$ ($k = 1, 2, 3$) for each point. This also means that a single γ can be sought for the LMIs induced by multiple points X_j. We henceforth express all our LMIs using a common γ.*

Corollary 6.3.1 allows to express the correspondence between a box in 3D and 2D point matches. It basically states that if the 3D point X_j was to be triangulated from 2D correspondences $\{x_j^i\}_{i=1}^m$, then it would be within the box $\mathcal{B}_j$ if LMIs (6.9) and (6.10) were feasible and outside this box otherwise. However, LMIs (6.9) and (6.10) depend upon the unknown Euclidean camera matrices and the true plane at infinity. Let us now consider the block-diagonal matrix

$$\mathsf{B}_j = diag(\underline{\mathsf{B}}_j^1, \underline{\mathsf{B}}_j^2, \underline{\mathsf{B}}_j^3, \overline{\mathsf{B}}_j^1, \overline{\mathsf{B}}_j^2, \overline{\mathsf{B}}_j^3) \tag{6.11}$$

whose blocks $\underline{\mathsf{B}}_j^k = (\widetilde{\Pi}_\infty \underline{\Pi}_k^{\mathsf{E}\intercal}\mathsf{H})^\star + \gamma(\mathsf{S}_j\mathsf{M})^\intercal\mathsf{S}_j\mathsf{M}$ and $\overline{\mathsf{B}}_j^k = \gamma(\mathsf{S}_j\mathsf{M})^\intercal\mathsf{S}_j\mathsf{M} - (\widetilde{\Pi}_\infty \overline{\Pi}_k^{\mathsf{E}\intercal}\mathsf{H})^\star$ are expressed using projective camera matrices, an unknown 4×4 transformation matrix H, and the surrogate plane at infinity $\widetilde{\Pi}_\infty$ (calculated as in Section 6.3.1). The following holds for visible scene points :

Proposition 6.3.3. *Let $\mathcal{S}x = \{(X_j, \mathcal{B}_j)\}_{j=1}^m$ be a set of putative point-to-box correspondences (i.e. each point X_j, projecting onto image points $\{x_j^i\}_{i=1}^n$, is assigned to a box $\mathcal{B}_j$). If $\mathcal{S}x$'s correspondences are correct then LMIs*

$$\mathsf{B}_j \geq \mathtt{I}, \;\; j = 1 \ldots m \tag{6.12}$$

must be simultaneously feasible for a scalar γ and at least the true transformation matrix H *satisfying* $\mathsf{X}_j^{\mathsf{E}} \sim \mathsf{H}\mathsf{X}_j$.

Proof The proof relies on Corollary 6.3.1. Recall that $\mathsf{M}^{\mathsf{E}} \sim \mathsf{M}\mathsf{H}^{-1}$ and $\Pi_\infty = \mathsf{H}^\intercal\Pi_\infty^{\mathsf{E}}$. It is well-known that congruence transformations preserve definiteness. Hence, pre- and post-multiplying the left-hand side of each of LMIs (6.9) and (6.10) by H and $\mathsf{H}^\intercal$, these can be respectively rewritten as $(\Pi_\infty \underline{\Pi}_k^{\mathsf{E}\intercal}\mathsf{H})^\star + \gamma(\mathsf{S}_j\mathsf{M})^\intercal\mathsf{S}_j\mathsf{M} > 0$ and $\gamma(\mathsf{S}_j\mathsf{M})^\intercal\mathsf{S}_j\mathsf{M} - (\Pi_\infty \overline{\Pi}_k^{\mathsf{E}\intercal}\mathsf{H})^\star > 0$. As per Remark 6.3.2, a common γ is used. Because $\mathsf{X}_j^\intercal\Pi_\infty\widetilde{\Pi}_\infty^\intercal\mathsf{X}_j$ carry the same sign for all points X_j, one can replace Π_∞ by $\widetilde{\Pi}_\infty$ thus leading to all $\underline{\mathsf{B}}_j^k$ and $\overline{\mathsf{B}}_j^k$ being simultaneously

either positive or negative definite. Since H is unknown, one may choose to enforce positive rather than negative definiteness. $\mathsf{B}_j > 0$ then arises naturally since a block-diagonal matrix is positive-definite if and only if each of its diagonal blocks is positive-definite. Because $\mathsf{B}_j > 0$ is homogeneous, it is replaced by (6.12).

Similarly, consider the matrix

$$\mathsf{D}_i = diag(\underline{\mathsf{D}}_i^1, \underline{\mathsf{D}}_i^2, \underline{\mathsf{D}}_i^3, \overline{\mathsf{D}}_i^1, \overline{\mathsf{D}}_i^2, \overline{\mathsf{D}}_i^3) \tag{6.13}$$

with blocks $\underline{\mathsf{D}}_i^k = \delta(\widetilde{\Pi}_\infty \underline{\Pi}_k^{\mathsf{E}\mathsf{T}}\mathsf{H})^\star + \gamma \mathsf{P}^{i\mathsf{T}}\mathsf{P}^i$ and $\overline{\mathsf{D}}_i^k = \gamma \mathsf{P}^{i\mathsf{T}}\mathsf{P}^i - \delta(\widetilde{\Pi}_\infty \overline{\Pi}_k^{\mathsf{E}\mathsf{T}}\mathsf{H})^\star$. Given $\widetilde{\Pi}_\infty$ and δ, both obtained by solving (6.6) and cheirality LMIs (6.8), the following holds :

Proposition 6.3.4. *Let $\mathcal{S}c = \{(C_i, C_i)\}_{i=1}^n$ be a set of putative camera-to-box correspondences (i.e. each camera center C_i is assigned to a box C_i). If $\mathcal{S}c$'s correspondences are correct, then LMIs*

$$\mathsf{D}_i \geq \mathtt{I}, \ i = 1 \ldots n \tag{6.14}$$

must be simultaneously feasible for a scalar γ and at least the true transformation matrix H *satisfying* $\mathsf{X}_j^{\mathsf{E}} \sim \mathsf{H}\mathsf{X}_j$.

Proof The proof, omitted here, is along the lines of that of Proposition 6.3.3. It employs Finsler's lemma while relying on the fact that $\mathsf{P}^i\mathsf{C}_i = 0$ and that $\delta \mathsf{C}_i^{\mathsf{T}} \Pi_\infty \widetilde{\Pi}_\infty^{\mathsf{T}} \mathsf{C}_i > 0$.

When a set of points and/or camera centers are putatively assigned to bounding boxes $\mathcal{B}_j$ and/or C_i, LMIs (6.12) and (6.14) can be simultaneously tested for feasibility. Should they be infeasible, one is guaranteed that at least one point or one camera center has wrongly been assigned to a box. Alternatively to assigning multiple points to boxes, one may use bounds on the entries of the sought matrix H to check whether or not a single point (or camera center)-to-box hypothesis is viable. Assuming the origin of the projective scene/cameras frame coincides with the centroid of the camera centers and SfM-deduced points, δ and $(\mathsf{H})_{44}$ can both be set to 1 (the last row of H being the plane at infinity - see [130] p. 526). The following corollary can be deduced :

Corollary 6.3.5. *Let* $\underline{\mathsf{H}}$ *and* $\overline{\mathsf{H}}$ *be the* 4×4 *matrices whose entries are valid, respectively, lower and upper bounds on the entries of the sought matrix* H. *If a point X (resp. camera*

center C) is boxed by $\mathcal{B}$ (resp. C), the LMI problem

$$\begin{aligned} &\mathsf{B} > 0 \text{ (resp. } \mathsf{D} > 0), \ (\mathsf{H})_{44} = 1 \\ &(\underline{\mathsf{H}})_{k\ell} < (\mathsf{H})_{k\ell} < (\overline{\mathsf{H}})_{k\ell} \ k, \ell = 1, 2 \dots 4 \end{aligned} \tag{6.15}$$

is feasible for a scalar γ and the true matrix H*.*

6.3.3/ REGISTRATION

We have devised two algorithms for registering 2D corresponding points across images with their 3D scanned counterparts. The first algorithm, named here SSR (Scene Structure Registration), is based on Propositions 6.3.3 and 6.3.4 and exploits the scene's structure. SSR is relatively fast, considering the problem at hand, but requires that matched 2D features have their corresponding 3D points scanned. This requirement is relaxed in our second registration method, named RR (Robust Registration), that allows a predefined number of 2D matches not to have scanned 3D counterparts. RR is based on Corollary 6.3.5 and considers each point-to-box assignment independently from the others. Both algorithms exploit the Branch-and-Prune (BnP) paradigm but explore different spaces. On the one hand, SSR subdivides non-empty bounding boxes to which points are assigned in order to iteratively obtain tighter boxes. This algorithm exploits the fact that scanned scenes consist of surface points and much of the explored space is void. A point that can only be assigned to an empty box indicates that the correspondence hypotheses for such assignment are surely incorrect. On the other hand, RR subdivides the space of parameters defined by the 15 bounded entries of the sought transformation matrix in order to obtain tighter bounds on this matrix while guaranteeing that at least a predefined number of points are assigned to non-empty boxes.

Initialization : In both SSR and RR algorithms, all scanned points are initially assigned to the scene's bounding box. Some applications and/or setups may allow to assign some of the points to smaller boxes. Camera centers are initially assigned to bounding boxes obtained either from GPS information or a good guess (possibly application-specific). Because estimating H requires 5 pairs of 3D-3D correspondences (no 4 points on one plane), 5 distinct non-overlapping bounding boxes in general position are required for the boundedness of the optimization problems at hand. These could be non-overlapping boxes on 4 cameras in addition to the scene's bounding box, or boxes around 3 came-

ras and 2 boxes in the scene, etc. Such assumption is considered satisfied throughout. In principal, this equivalent to already having a weak (or quasi) Euclidean reconstruction which lies somewhere between Projective and Euclidean reconstructions. Based on Corollary 6.3.5, the initial bounds on the entries of H can be obtained by solving a series of SDPs. That is, for each entry $(\mathsf{H})_{k\ell}$, solve

$$\max_{\mathsf{H},\gamma} / \min_{\mathsf{H},\gamma} (\mathsf{H})_{k\ell} \; s.t. \; \mathsf{B}_j > 0, \; \mathsf{D}_i > 0, \; (\mathsf{H})_{44} = 1. \tag{6.16}$$

In the absence of sufficient bounding boxes, one can also use the structure of H given by Equation (2.17). Note that the entries of H consists of the first camera intrinsic K^1, rotation R^1, translation t^1, and the plane at infinity π_∞. Given the bounds on all these entries, the bounds on H can be obtained using interval analysis technique[140]. In fact, the entries of rotation matrix is always bounded between -1 and 1. It is trivial to find the bounds on intrinsic using an informed guess (as in [134],[137],[141]). The bounds on plane at infinity can be estimated by solving the optimization problem of Equation (2.21). Similarly, the translation bounds can also be obtained as in Chapter 5, from the vague knowledge about the location of first camera.

SSR : At any given iteration of the SSR algorithm, one is given the sets $\mathcal{S}x = \{(X_j, \mathcal{B}_j)\}_{j=1}^m$ and $\mathcal{S}c = \{(C_i, C_i)\}_{i=1}^n$ of respectively point-to-box and camera-to-box assignments. The set $\mathcal{S}x \cup \mathcal{S}c$ defines a node in a dynamically-built search tree. The point or camera-to-box assignments therein have feasible H and γ simultaneously satisfying their corresponding LMIs (6.12) and (6.14). Algorithm 7, that requires solving Problem 1 below, is used to reassess the boxes of all points and camera centers such that smaller boxes contribute to shrinking larger ones and all boxes best fit the scanned points within. If any box assigned to a point turns out to be empty, the branch is marked for dismissal and the hypothetical assignments are dropped. If a branch is not dismissed, then the feasible H for LMIs (6.12) and (6.14) is used to initialize a projective ICP-like refinement (discussed below). The branch with the lowest cost (6.19) is processed first. The box in $\mathcal{S}x$ with the longest edge is subdivided (along the latter edge) into two boxes resulting in two new branches to be explored.

Problem 1 : Let $\mathcal{S}x = \{(X_j, \mathcal{B}_j)\}_{j=1}^m$ and $\mathcal{S}c = \{(C_i, C_i)\}_{i=1}^n$ be sets of putative, respectively, point-to-box and camera-to-box assignments. Let $X \in \{X_j\}_{j=1}^m$ be boxed by

Algorithm 7 [$\mathcal{S}x$,$\mathcal{S}c$] = SSR-NodeProcessing($\mathcal{S}x$,$\mathcal{S}c$)

for each $a \in \mathcal{S}x \cup \mathcal{S}c$ **do**
  ($a \in \mathcal{S}x \cup \mathcal{S}c$ consists in a tuple $(a.X, a.\mathcal{B})$)
  Refine $a.\mathcal{B}$ by solving Problem 1
  if $a \in \mathcal{S}x$ (i.e. $a.X$ is a point) **then**
    if refined $a.\mathcal{B}$ is empty (i.e. no scanned points) **then**
      $\mathcal{S}x \leftarrow \varnothing$; $\mathcal{S}c \leftarrow \varnothing$ (branch to dismiss)
    else
      Shrink $a.\mathcal{B}$ to best fit scanned points within
      Update a in $\mathcal{S}x$
    end if
  else
    Update a in $\mathcal{S}c$
  end if
end forreturn [$\mathcal{S}x$, $\mathcal{S}c$]

$\mathcal{B} = \{(\underline{\Pi}_k, \overline{\Pi}_k)\}_{k=1}^{3} \in \{\mathcal{B}_j\}_{j=1}^{m}$ for which a possibly tighter box may exist. Recalling that $\underline{\Pi}_k^{\mathsf{E}} = (\mathsf{e}_k^{\mathsf{T}} \ -\underline{d}_k)^{\mathsf{T}}$ and $\overline{\Pi}_k^{\mathsf{E}} = (\mathsf{e}_k^{\mathsf{T}} \ -\overline{d}_k)^{\mathsf{T}}$, a new upper bound $\overline{d}_k$ for some fixed k can be obtained by solving

$$\begin{array}{ll} \max\limits_{\mathsf{H},\gamma,d_k} & d_k \\ s.t. & (\widetilde{\Pi}_\infty(\mathsf{e}_k^{\mathsf{T}} \ -d_k)\mathsf{H})^\star + \gamma(\mathsf{SM})^{\mathsf{T}}\mathsf{SM} > \mathtt{I}, \\ & \mathsf{B}_j \geq \mathtt{I} \ j = 1 \ldots m, \ \mathsf{D}_i \geq \mathtt{I} \ i = 1 \ldots n. \end{array} \tag{6.17}$$

This can be solved by binary search over d_k in the range $[\underline{d}_k, \overline{d}_k]$. Intuitively, this is equivalent to pushing $\underline{\Pi}_k^{\mathsf{E}}$ towards $\overline{\Pi}_k^{\mathsf{E}}$ until either the two planes coincide (no smaller bound on $\overline{d}_k$) or X cannot be mapped on the positive side of $(\mathsf{e}_k^{\mathsf{T}} \ -d_k)^{\mathsf{T}}$. This latter case means that X can only be mapped on the negative side of $(\mathsf{e}_k^{\mathsf{T}} \ -d_k)^{\mathsf{T}}$ thus making the resulting d_k the new upper bound $\overline{d}_k$. A new lower bound $\underline{d}_k$ can be obtained by solving a similar problem to (6.17) by minimizing d_k while $\gamma(\mathsf{S}_j\mathsf{M})^{\mathsf{T}}\mathsf{S}_j\mathsf{M} - (\widetilde{\Pi}_\infty(\mathsf{e}_k^{\mathsf{T}} \ -d_k)\mathsf{H})^\star > \mathtt{I}$ and subjected to points and cameras' bounding LMIs.

RR : At any given iteration of the RR algorithm, one is given bounds on the 15 entries of H (given $(\mathsf{H})_{44} = 1$) and a set $\mathcal{S}x$ of point-to-box putative assignments. The set $\mathcal{S}x$ and H's bounds define a node in a dynamically-built search tree. Algorithm 8 refines the box assigned to each point based on the bounds on H it has been provided. This algorithm returns a new set $\mathcal{S}x$ with updated boxes and, more importantly, empty box assignments taken away. The cardinality of $\mathcal{S}x$ hence provides the number of points actually assigned to non-empty boxes. The node is dropped if the number of such point-to-box assign-

ments is below a predefined threshold or LMIs (6.15) are infeasible when considered simultaneously for all assignments in the refined $\mathcal{S}x$. Otherwise, the feasible H satisfying LMIs (6.15) due to $\mathcal{S}x$ is used to initialize the projective ICP-like refinement. The branch with the lowest cost (6.19) is processed first. In this case, H is branched along its longest edge thus creating two new branches (inheriting the refined $\mathcal{S}x$) to explore.

Algorithm 8 $\mathcal{S}x$ = RR-NodeProcessing($\mathcal{S}x$,$\underline{\mathsf{H}}$,$\overline{\mathsf{H}}$)

for each $a \in \mathcal{S}x$ **do**
  Refine $a.\mathcal{B}$ by solving Problem 2
  if $a.\mathcal{B}$ is empty (i.e. no scanned points) **then**
    Remove a from $\mathcal{S}x$
  else
    Update a in $\mathcal{S}x$
  end if
end forreturn $\mathcal{S}x$

Problem 2 : Now consider bounds on H are given and $\mathcal{B} = \{(\underline{\Pi}_k, \overline{\Pi}_k)\}_{k=1}^{3}$ be the box to which X is assigned. With tighter bounds on H one can obtain new tighter bounds on X. A new upper bound $\overline{d}_k$ can be obtained by solving

$$\begin{aligned} \max_{\mathsf{H},\gamma,d_k} \quad & d_k \\ s.t. \quad & (\widetilde{\Pi}_\infty(\mathsf{e}_k^\intercal - d_k)\mathsf{H})^\star + \gamma(\mathsf{SM})^\intercal\mathsf{SM} > 0, \\ & (\underline{\mathsf{H}})_{k\ell} < (\mathsf{H})_{k\ell} < (\overline{\mathsf{H}})_{k\ell}\ k,\ell = 1,2\ldots 4, \\ & (\mathsf{H})_{44} = 1 \end{aligned} \tag{6.18}$$

assuming the SfM-scene and cameras centered at the origin of the projective frame. As in (6.17), this can be solved by binary search over d_k in the range $[\underline{d}_k, \overline{d}_k]$. The largest d_k is the new $\overline{d}_k$. A lower bound $\underline{d}_k$ can very much be obtained in the same manner as discussed for Problem 1.

Termination : Both SSR and RR algorithms terminate when the cost of the projective ICP-like refinement reaches a predefined objective or when all branches have been processed (up to bound gap in the branching parameters). In the latter case the best solution is returned.

Projective ICP-like refinement : Let $\mathcal{X}_j$ be the set of scanned 3D points boxed by some

box $\mathcal{B}_j$. Given initialization on H, it can be refined by minimizing the cost,

$$\mathbb{C}(\mathsf{H}) = \sum_{i=1}^{n}\sum_{j=1}^{m} \min_{X \in \mathcal{X}_j} \mathbf{d}(\mathsf{x}_j^i, \mathsf{P}^{\mathsf{i}}\mathsf{H}^{-1}\mathsf{X}^{\mathsf{E}})^2 \tag{6.19}$$

where $\mathbf{d}(.,.)$ is the Euclidean distance. This is carried out by alternating matching 3D scanned points in bounding boxes and 2D points (based on re-projection error) and re-estimating H.

Discussion : Both algorithms (proposed above) search the optimal registration parameters using the goal attainment-based BnP paradigm. If there exists any solution that minimizes the cost below desired threshold, these methods safely return the first available solution. Otherwise, the methods return the best solution found so far. In the context of outliers, we recommend to use a variant of RR that maximizes the inlier set, similar to the problem stated in Equation (5.2). The variant of RR uses BnB paradigm (unlike RR), and maximizes the inlier assignment while searching for the optimal registration parameters. This is important mainly because it provide the guarantee of finding optimal solution for a different objective function (of the previous Chapter). The implementation of this method provides similar solution to that of RR, while guaranteeing that there exists no other better solution. In terms of processing time, the time it takes to reach to the optimal solution is very similar to that of RR, however the finding the optimal certificate requires extra time. The extra time again depends upon the experimental setups.

6.4/ EXPERIMENTS

We tested the proposed methods using synthetic and real images. Projective reconstruction was obtained using [21] and refined via Bundle Adjustment [69] in [112] using Rabauds SfM Toolbox [142]. The algorithms were implemented in MATLAB2012a and the LMI problems were solved using the LMI Control Toolbox. All experiments were carried out on a Pentium i7/2.50GHz with 8GB RAM.

Synthetic data : We generated a set of 800 random 3D points scattered on the surface of four faces of a $20m \times 20m \times 20m$ scene box. The cameras were placed about 20 ± 2m away from the scene's centroid with randomly generated rotations while looking towards the

scene. 800 additional points were also generated on the surface of a hemisphere placed at a corner of the box. Of these points, 1000 were randomly selected and projected onto 512×512 images with zero-skew, 200 pix. focal length and an image-centered principal point. The projected points were imposed 0.0 to 2.0 pixels random noise (with a step of 0.4). Only 20 image points were assumed to be matched across the image sequence. The SSR method was tested by changing various parameters while conducting 50 experiments for each setup. The number of views was varied from 5 to 15 (with a step of 2) while bounding camera centers inside cubic bounding boxes (denoted Bbx) of different sizes (sides of 20cm, 2m, and 4m), with no constraints on the scene points. The number of branching was allowed to be no more than 50 to restrict the maximum processing time. The 2D projection error threshold was set to 10^{-2}.

The median time taken for various experiments against the number of bounded cameras and image noise are shown in Figure 6.1. Similarly, Figure 6.2 shows the success count over 50 experiments. 2D-to-3D registration accuracy was measured by computing the 3D registration error of all 1000 reconstructed points to the scene. Measured 3D RMS registration error is shown in Figure 6.3. An experiment is assumed to be successful if it produces less than 0.1 3D error. The estimated camera intrinsics and pose were compared against that of ground truth. The Euclidean projection matrix of the first camera was recovered using $\mathsf{P}^{\mathsf{E}1} = \mathsf{K}^1[\mathsf{R}^1\ \mathsf{t}^1] = \mathsf{P}^1\mathsf{H}^{-1}$. For the evaluation, error measurement metrics for N number of experiments are defined as follows

$$\Delta f = \sqrt{\frac{\sum_{i=1}^{N} (\alpha_i^1 - \alpha)^2 + (\beta_i^1 - \beta)^2}{N(\alpha^2 + \beta^2)}}, \quad \Delta R = \sqrt{\frac{\sum_{i=1}^{N} \|\mathsf{r}_i^1 - \mathsf{r}\|^2}{3N}},$$

$$\Delta uv = \sqrt{\frac{\sum_{i=1}^{N} (u_i^1 - u)^2 + (v_i^1 - v)^2}{N(u^2 + v^2)}}, \quad \Delta t = \sqrt{\frac{\sum_{i=1}^{N} \|\mathsf{t}_i^1 - \mathsf{t}\|^2}{N(\|\mathsf{t}\|^2)}},$$

where α^1, β^1 represent two focal lengths, and (u^1, v^1) is the principal point. r^1 is a vector obtained by stacking three rotation angles in degrees. These angles are obtained from R^1 after enforcing its orthogonality. The corresponding variables without subscript represent the ground truth. The errors in camera intrinsics and pose are shown in Figure 6.4. The success, speed, and accuracy improve with the increase in number of views and decrease in the box size.

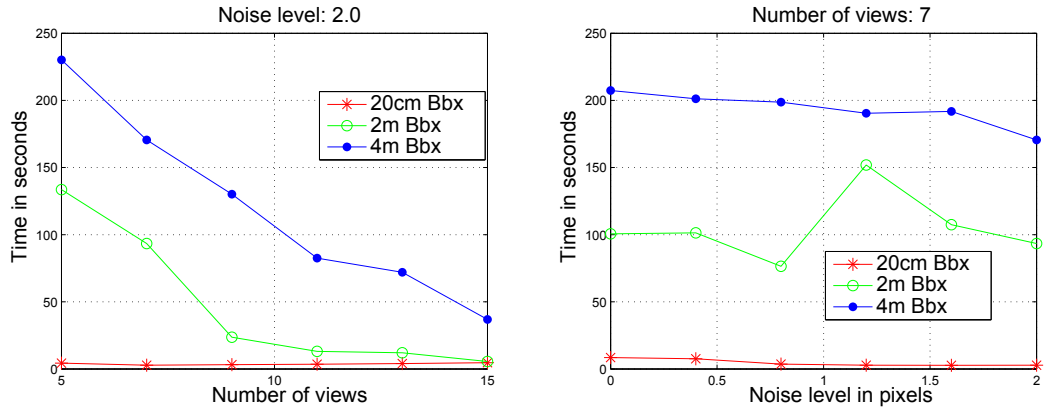

FIGURE 6.1 – Time vs. number of views and noise.

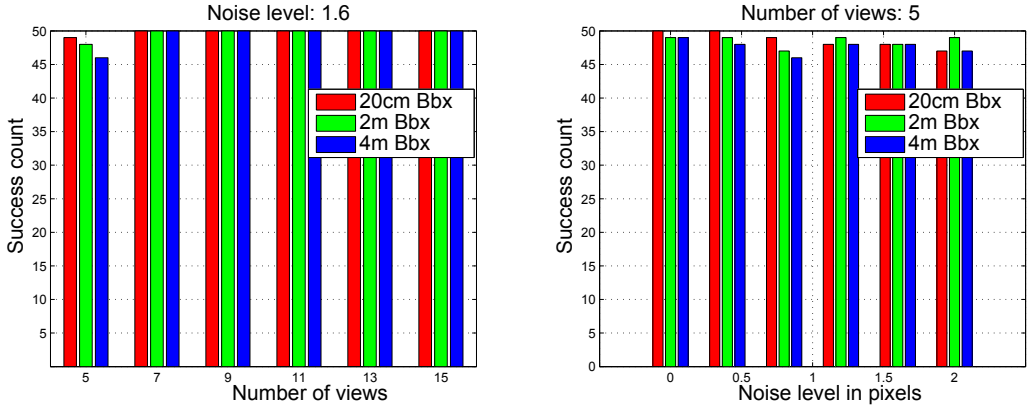

FIGURE 6.2 – Success count vs. number of views and noise.

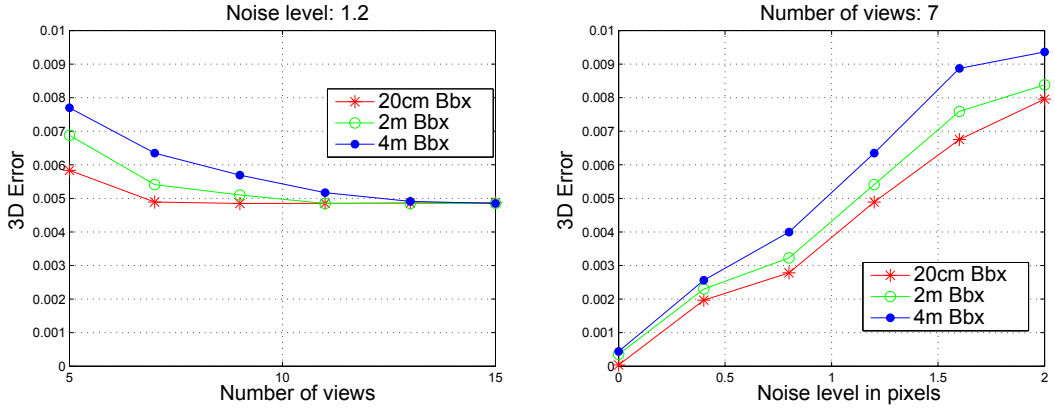

FIGURE 6.3 – Registration error vs. number of views and noise.

Critical motion sequence : we also tested our method on one particular critical motion sequence for camera auto-calibration. Tested sequence consists cameras moving in a cir-

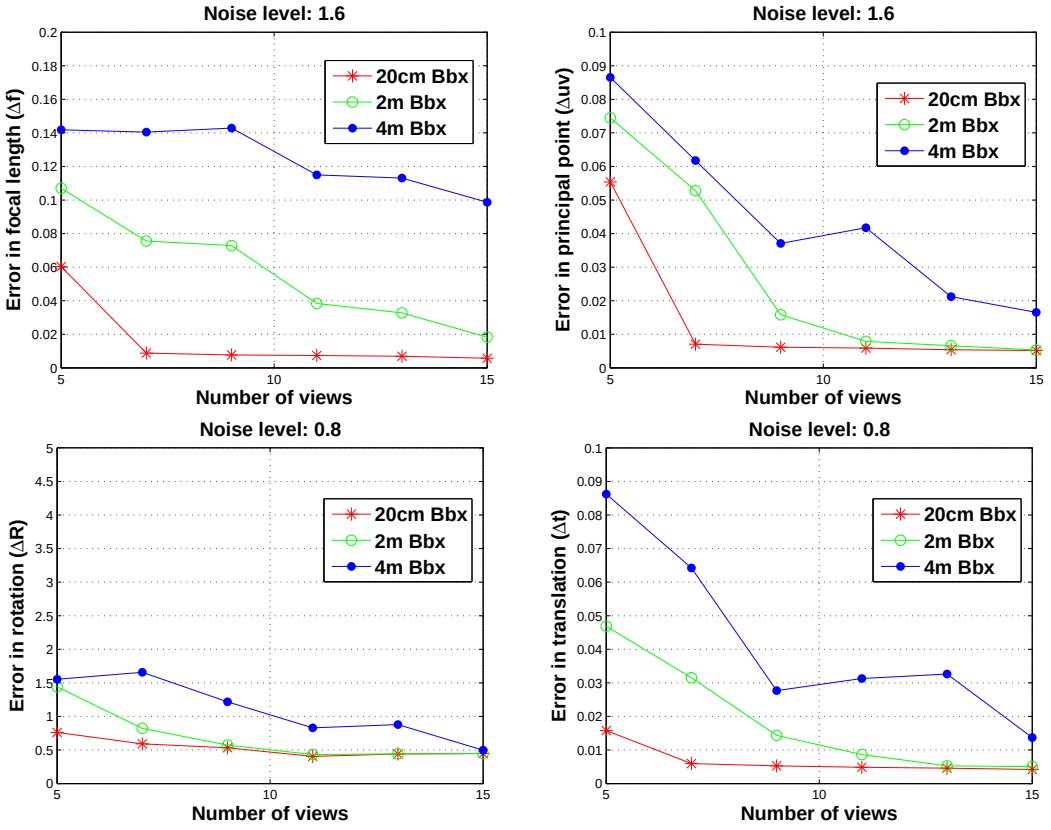

FIGURE 6.4 – Intrinsic and pose errors vs. number of views.

cular motion around the scene. These cameras maintain a constant distance of 20m from the centroid of the scene, while going thought the rotation about only z-axis. We conducted two sets of experiments (20 each set) for 15 cameras bounded inside the bounding boxes of 20cm and 2m. All the experiments with 20cm bounding boxes were successfully converged to the desired solution in 82.38 seconds of median time. On the otherhand, 17 out of 20 experiments were converged to the desired solution within the median time of 232.21 seconds. All the experiments were ended with failure when conducted using both [136] and [137]. Note that the tested motion is not critical motion for the problem at hand because of the 3D knowledge of the scene.

Real data : We tested our method with two real datasets : Fountain-P11 and Herz-Jesu-P8 (from [132]). These datasets consist, respectively, of 11 and 8 images of size 3072×2048 captured by a moving camera of $\alpha = 2759.5$, $\beta = 2764.2$, $u = 1520.7$ and $v = 1006.8$, along with the laser scanned 3D scenes. Our results were compared against two methods : RISAG [76] and Go-ICP [73]. RISAG requires metric reconstruction, hence works only for the calibrated case. Likewise, Go-ICP requires an Euclidean reconstruc-

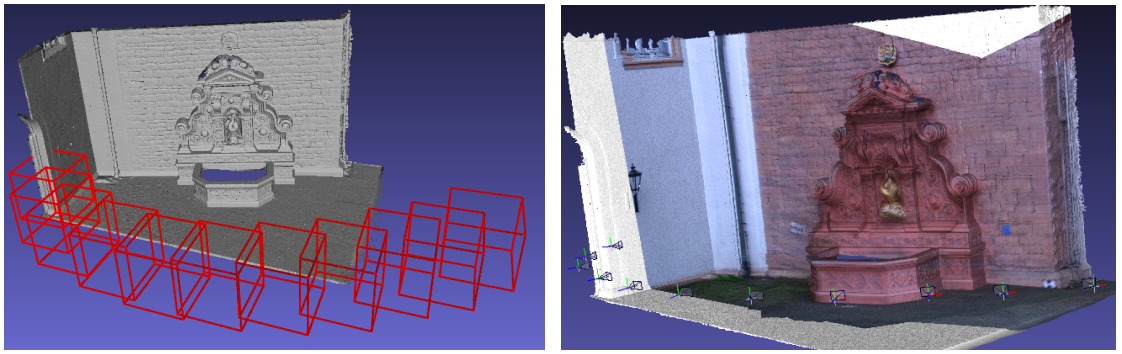

FIGURE 6.5 – Fountain : (left) 11 cameras 2m Bbx and scene, (right) estimated cameras in textured scene using SSR.

FIGURE 6.6 – Herz-Jesu : (left) matched 2D features with outliers in red, (right) texture-mapped scene using RR.

tion, which was obtained by upgrading the metric reconstruction using ground truth projection matrices. The metric reconstruction was obtained using openMVG [133]. The results obtained for all four methods are shown in Table 6.1. For qualitative analysis, estimated projection matrices were used for texture mapping. The obtained results using our methods were very accurate. These are shown in Figures 6.5-6.6 which also provide the results after further refinement using [56]. Note that a small error in pose can significantly affect the texture mapping. For the Fountain sequence, both SSR and RR converged to the same solution. RR, however, converged to a better solution for Herz-Jesu.

Sequence	Method	Points Bbx.		Cameras Bbx.		Time (sec)	Δf	Δuv	ΔR	Δt	3D Error
		Num.	Size (m)	Num.	Size (m)						
Fountain	SSR	0/36	-	11	2.00	78.423	0.0162	0.0275	0.2471	0.0196	0.0164
		18/36	2.00	11	2.00	3.912	0.0162	0.0275	0.2471	0.0196	0.0164
	RR	10/36	1.00	11	1.00	52.796	0.0162	0.0275	0.2471	0.0196	0.0164
	RISAG	-/4601	-	11	-	805.680	-	-	8.6825	0.1408	0.3275
	Go-ICP	-/4601	-	11	-	529.415	-	-	0.7225	0.0163	0.0348
Herz-Jesu	SSR	0/29	-	8	2.00	57.442	0.0207	0.0280	0.1441	0.0137	0.0221
		18/29	2.00	8	2.00	3.999	0.0207	0.0280	0.1441	0.0137	0.0221
	RR	5/29	1.00	8	1.00	8.212	0.0039	0.0089	0.3223	0.0061	0.0019
	RISAG	-/4024	-	8	-	160.064	-	-	17.6378	0.0570	0.1830
	Go-ICP	-/4024	-	8	-	31.254	-	-	3.2618	0.169	0.0725

TABLE 6.1 – Four different methods with real data. Points Bbx. p/q means p number of points are bounded out of q sought.

6.5/ CONCLUSION

We have presented a novel approach for registering two or more uncalibrated cameras to a 3D scanned scene. The proposed approach only assumes point correspondences across images. Our solution allows estimating the unknown projective transformation relating the cameras to the scene and establishing 2D-3D correspondences. A LMI framework was used to overcome the image-induced point triangulation requirement. Using this framework, we have derived triangulation-free LMI cheirality conditions and LMI constraints for establishing putative correspondences between 3D boxes and 2D points. Two globally convergent algorithms, one exploiting the scene's structure and the other concerned with robustness, have been presented.

7

Conclusion and Future Work

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

- Alan Turing, *Computing Machinery and Intelligence*

In this thesis, we studied three different cases of 2D and 3D registration. When a moving set-up of 2D cameras and 3D sensor are calibrated, synchronized and rigidly attached, we showed that the 2D and 3D information can be fused in a relative ease. Fusing 2D and 3D information not only allowed us to obtain better scene modeling (as in texture-mapped scenes), but also helped us to recover the accurate motion parameters (as in visual odometry). We also showed that even when the 2D cameras and 3D sensor are not synchronized (as in asynchronous set-up), it is possible to register 2D and 3D data so that they could still be fused together. We presume that the failure to synchronization demands the re-calibration of extrinsic parameters. Re-calibration was carried out using a local refinement method, under the assumption that the extrinsic parameters obtained from the set-up at rest serves as an initialization.

The problem of registering 2D and 3D data becomes more difficult when the 2D cameras and 3D sensor are allowed to move freely. This results into no reliable guess on registration parameters, making it impossible to use local registration methods. For the uncoupled camera set-ups, we have devised a globally optimal method of 2D-3D registration. Our method registers a set of calibrated images to the structured scene, with the help of SfM-induced reconstruction. In this context, the metric homography that relates the reconstruction to the scene is sought using the Branch-and-Prune paradigm. Our registration method assumes that the scene can be segmented and meaningfully re-

presented by planar patches, and the bounds on the position of one camera (translation with respect to the 3D scene coordinate frame) are known. The registration process was carried out by assigning reconstructed points to the segmented planar patches. While doing so, we searched for the registration parameters such that it maximizes the set of inlier assignments using sum-of-squares-based outlier detection conditions. With several experiments, we showed that our method is very robust to outliers, so much so that it worked even when each point was assigned to all the available planes.

The challenge of registering 2D and 3D data becomes even more difficult when the 2D images are captured by uncalibrated cameras. The possible reconstruction that can be obtained in this case relates to the 3D scene by a projective homography. In this work, we proposed a global method of finding that homography by establishing the direct relationships between 2D and 3D measurements, under the framework of Branch-and-Prune search paradigm. This process of registration allowed us to devise two kinds of algorithms : one that relies on scene structure and the other concerned with robustness. Our experiments showed that although the scene structure-based method demands strong criteria of complete 3D scene parts with no 2D point outliers, it successfully exploits the fact that the scene points are from a surface, for a faster registration process. On the other hand, another proposed algorithm that doesn't require complete scene parts and handles the presence of 2D point outliers, provided very robust results in the practical scenarios, at the cost of extra computation time.

Inline with [138, 100], our experiments showed that the 2D-3D registration (or camera pose estimation) is more accurate when the 2D re-projection error is minimized, compared to registration based on 3D-to-3D relationships. When given good initialization (like in asynchronous case), it is safe to assume that local method serves the purpose of registration and fusion. In fact, we have also demonstrated that the local method can also play an important role even inside the global optimization framework. Furthermore, one can always refine the results from global registration methods, if necessary, in the light of the local method developed in Chapter 4. Based on our observations, the key of accurate registration is either because of the final refinement process or the local refinement conducted inside the global framework. Despite the fact that the global methods are of very high value for approximating the correct solution, local methods remain important when highly accurate solutions are desired.

Although the optimization method developed in Chapter 5 has been discussed and tested for inlier set maximization of point-to-plane assignments, it is not limited to the calibrated 2D-3D registration problem alone. In fact, one must be able to successfully exploit the proposed method for solving many more problems, as long as the problem can be modeled as in Equation (5.2). As a prospect, it would be interesting to test our method on inlier set maximization for other polynomial problems. Furthermore, it would also be interesting to explore the cases when the polynomials do not belong to the category of PSD and SoS equivalence (please, refer Table 3.1). The holy grail of consensus set maximization for non-linear polynomial systems would be to develop a generic framework that does not require PSD to SoS equivalence for the optimality certificate. This indeed may require more understanding and exploitation of algebraic geometry concepts and more efficient search techniques.

In case of uncalibrated camera set-ups, we look forward to incorporate the camera auto-calibration LMI conditions, as in [137], during the registration process. This will allow us to jointly benefit from both auto-calibration and registration constraints, simultaneously. As the current version of the registration method requires the rank-3 enforcement on the measurement matrix ($S_j M$) before using Finsler's lemma 3.4.1, one other interesting direction will be to consider the effect of noisy data during triangulation process, as discussed in [19] for the uncalibrated case. We we have so far tested the uncalibrated registration using the projective reconstruction method [21] that does not consider any missing entry in the measurement matrix. One other interesting direction will be to develop the registration method that successfully considered the presence of outliers during the projective reconstruction itself, as in [22].

REFERENCES

[1] Danda Pani Paudel, Cédric Demonceaux, Adlane Habed, Pascal Vasseur, and In So Kweon. 2d-3d camera fusion for visual odometry in outdoor environments. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 157–162. IEEE, 2014.

[2] Danda Pani Paudel, Cédric Demonceaux, Adlane Habed, and Pascal Vasseur. Estimation de la pose d'une caméra dans un environnement connu à partir d'un recalage 2d-3d. In *Reconnaissance de Formes et Intelligence Artificielle (RFIA) 2014*, 2014.

[3] Danda Pani Paudel, Cédric Demonceaux, Adlane Habed, and Pascal Vasseur. Localization of 2d cameras in a known environment using direct 2d-3d registration. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 196–201. IEEE, 2014.

[4] Danda Pani Paudel, Adlane Habed, Cedric Demonceaux, and Pascal Vasseur. Robust and optimal sum-of-squares-based point-to-plane registration of image sets and structured scenes. In *Proceedings of the IEEE Conference on Computer Vision (ICCV)*, 2015.

[5] Danda Pani Paudel, Adlane Habed, Cedric Demonceaux, and Pascal Vasseur. Lmi-based 2d-3d registration : From uncalibrated images to euclidean scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4494–4502, 2015.

[6] Danda Pani Paudel, Adlane Habed, Cedric Demonceaux, and Pascal Vasseur. Method for free registration of a euclidean 3-dimensional scanned scene and image sets. In *No 62/165,433, United States, US Provisional Patent, May 22*, 2015.

[7] Jacqueline Le Moigne and Allen M Waxman. Projected light patterns for short range navigation of autonomous robots. In *Proc. Int. Conf. on Pattern Recognition*, volume 1, pages 203–206, 1984.

[8] Hiroyoshi Morita, Kaanyasn Yajima, and Shojiro Sakata. Reconstruction of surfaces of 3-d objects by m-array pattern projection method. In *Computer Vision., Second International Conference on*, pages 468–473. IEEE, 1988.

[9] Minoru Maruyama and Shigeru Abe. Range sensing by projecting multiple slits with random cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(6) :647–651, 1993.

[10] Kim L Boyer and Avinash C Kak. Color-encoded structured light for rapid active ranging. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1) :14–28, 1987.

[11] V Srinivasan, HC Liu, and Maurice Halioua. Automated phase-measuring profilometry of 3-d diffuse objects. *Applied optics*, 23(18) :3105–3108, 1984.

[12] Shouhong Tang and Yau Y Hung. Fast profilometer for the automatic measurement of 3-d object shapes. *Applied Optics*, 29(20) :3012–3018, 1990.

[13] Jason Geng. Structured-light 3d surface imaging : a tutorial. *Advances in Optics and Photonics*, 3(2) :128–160, 2011.

[14] David Nitzan. Three-dimensional vision structure for robot applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(3) :291–309, 1988.

[15] T Nielsen, F Bormann, S Wolbeck, H Spiecker, MD Burrows, and P Andresen. Time-of-flight analysis of light pulses with a temporal resolution of 100 ps. *Review of scientific instruments*, 67(5) :1721–1724, 1996.

[16] Giovanna Sansoni, Marco Trebeschi, and Franco Docchio. State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors*, 9(1) :568–601, 2009.

[17] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10) :105–112, 2011.

[18] David Nistér. An efficient solution to the five-point relative pose problem. In *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 756–777, June 2004.

[19] Richard I Hartley and Peter Sturm. Triangulation. volume 68, pages 146–157. Elsevier, 1997.

[20] Peter Sturm and Bill Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Computer Vision—ECCV'96*, pages 709–720. Springer, 1996.

[21] John Oliensis and Richard Hartley. Iterative extensions of the sturm/triggs algorithm : Convergence and nonconvergence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12) :2217–2233, 2007.

[22] Yuchao Dai, Hongdong Li, and Mingyi He. Element-wise factorization for n-view projective reconstruction. In *Computer Vision–ECCV 2010*, pages 396–409. Springer, 2010.

[23] Anders Heyden, Rikard Berthilsson, and Gunnar Sparr. An iterative factorization method for projective structure and motion from image sequences. *Image and Vision Computing*, 17(13) :981–991, 1999.

[24] Shyjan Mahamud, Martial Hebert, Yasuhiro Omori, and Jean Ponce. Provably-convergent iterative methods for projective structure from motion. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–1018. IEEE, 2001.

[25] Bill Triggs. Factorization methods for projective structure and motion. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 845–851. IEEE, 1996.

[26] Toshio Ueshiba and Fumiaki Tomita. A factorization method for projective and euclidean reconstruction from multiple perspective views via iterative depth estimation. In *Computer Vision—ECCV'98*, pages 296–310. Springer, 1998.

[27] Behrooz Nasihatkon, Richard Hartley, and Jochen Trumpf. A generalized projective reconstruction theorem and depth constraints for projective factorization. *International Journal of Computer Vision*, pages 1–28, 2015.

[28] David Nistér. Calibration with robust use of cheirality by quasi-affine reconstruction of the set of camera projection centres. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 116–123. IEEE, 2001.

[29] Richard Hartley, Eric Hayman, Lourdes de Agapito, Ian Reid, et al. Camera calibration and the search for infinity. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 510–517. IEEE, 1999.

[30] Lingyun Liu and Ioannis Stamos. Automatic 3d to 2d registration for the photorealistic rendering of urban scenes. In *CVPR*, pages 137–143, 2005.

[31] Michel Dhome, Ali Yassine, and Jean-Marc Lavest. Determination of the pose of an articulated object from a single perspective view. In *BMVC*, pages 1–10, 1993.

[32] Primoz Markelj, D Tomaževič, Bostjan Likar, and F Pernuš. A review of 3d/2d registration methods for image-guided interventions. *Medical image analysis*, 16(3) :642–661, 2012.

[33] Stéphane Nicolau, Xavier Pennec, Luc Soler, and Nicholas Ayache. Evaluation of a new 3d/2d registration criterion for liver radio-frequencies guided by augmented reality. In *Surgery Simulation and Soft Tissue Modeling*, pages 270–283. Springer, 2003.

[34] JA Grunert. Das pothenot'sche problem, in erweiterter gestalt, nebst bemerkungen über seine anwendung in der. *Archiv der Mathematik und Physik*, page 238248, 1841.

[35] Abdel Y. I. Aziz and H. M. Karara. Direct linear transformation into object space coordinates in close-range photogrammetry. In *Proc. of the Symposium on Close-Range Photogrammetry*, pages 1–18, Urbana, Illinois, 1971.

[36] Radu Horaud, Bernard Conio, Olivier Leboulleux, and LIFIA Bernard Lacolle. An analytic solution for the perspective 4-point problem. In *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR'89., IEEE Computer Society Conference on*, pages 500–507. IEEE, 1989.

[37] Robert M Haralick, Chung-nan Lee, K Ottenburg, and Michael Nölle. Analysis and solutions of the three point perspective pose estimation problem. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 592–598. IEEE, 1991.

[38] Daniel DeMenthon and Larry S Davis. Exact and approximate solutions of the perspective-three-point problem. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11) :1100–1105, 1992.

[39] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(8) :930–943, 2003.

[40] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp : An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2) :155–166, 2009.

[41] Luis Ferraz, Xavier Binefa, and Francesc Moreno-Noguer. Very fast solution to the pnp problem with algebraic outlier rejection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 501–508. IEEE, 2014.

[42] Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8) :774–780, 1999.

[43] Bill Triggs. Camera pose and calibration from 4 or 5 known 3d points. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 278–284. IEEE, 1999.

[44] Michel Dhome, Marc Richetin, Jean-Thierry Lapreste, and Gerard Rives. Determination of the attitude of 3d objects from a single perspective view. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(12) :1265–1278, 1989.

[45] Robert M Haralock and Linda G Shapiro. *Computer and robot vision*. Addison-Wesley Longman Publishing Co., Inc., 1991.

[46] Yiu Cheung Shiu and Shaheen Ahmad. 3d location of circular and spherical features by monocular model-based vision. In *Systems, Man and Cybernetics, 1989. Conference Proceedings., IEEE International Conference on*, pages 576–581. IEEE, 1989.

[47] Marc Richetin, M Chome, Jean-Thierry Lapresté, and Gerard Rives. Inverse perspective transform using zero-curvature contour points : Application to the localization of some generalized cylinders from a single view. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2) :185–192, 1991.

[48] Florent Nageotte, Philippe Zanne, Christophe Doignon, and Michel de Mathelin. Visual servoing-based endoscopic path following for robot-assisted laparoscopic surgery. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2364–2369. IEEE, 2006.

[49] Gehua Yang, Jacob Becker, and Charles V Stewart. Estimating the location of a camera with respect to a 3d model. In *3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on*, pages 159–166. IEEE, 2007.

[50] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[51] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 667–674. IEEE, 2011.

[52] Martin A Fischler and Robert C Bolles. Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395, 1981.

[53] John H Hipwell, Graeme P Penney, Robert McLaughlin, Kawal Rhode, Paul Summers, Tim C Cox, James V Byrne, J Alison Noble, David J Hawkes, et al. Intensity-based 2-d-3-d registration of cerebral angiograms. *Medical Imaging, IEEE Transactions on*, 22(11) :1417–1426, 2003.

[54] Dejan Tomaževič, Bostjan Likar, Tomaž Slivnik, and Franjo Pernuš. 3-d/2-d registration of ct and mr to x-ray images. *Medical Imaging, IEEE Transactions on*, 22(11) :1407–1416, 2003.

[55] Dejan Toma Zevi, Bostjan Likar, and Franjo Pernu. 3-d/2-d registration by integrating 2-d information in 3-d. *Medical Imaging, IEEE Transactions on*, 25(1) :17–27, 2006.

[56] Paul Viola and William M. Wells, III. Alignment by maximization of mutual information. *IJCV*, pages 137–154, September 1997.

[57] Srikumar Ramalingam, Sofien Bouaziz, Peter Sturm, and Matthew Brand. Skyline2gps : Localization in urban canyons using omni-skylines. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3816–3823. IEEE, 2010.

[58] Aparna Taneja, Luca Ballan, and Marc Pollefeys. Registration of spherical panoramic images with cadastral 3d models. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 479–486. IEEE, 2012.

[59] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1508–1515. IEEE, 2005.

[60] Tom Drummond and Roberto Cipolla. Real-time visual tracking of complex structures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7) :932–946, 2002.

[61] Eric Marchand, Patrick Bouthemy, François Chaumette, and Valérie Moreau. Robust real-time visual tracking using a 2d-3d model-based approach. In *IEEE Int. Conf. on Computer Vision, ICCV'99*, volume 1, pages 262–268, 1999.

[62] Thomas J Cashman and Andrew W Fitzgibbon. What shape are dolphins ? building 3d morphable models from 2d images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1) :232–244, 2013.

[63] Abhishek Kar, Shubham Tulsiani, João Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. *arXiv preprint arXiv :1411.6069*, 2014.

[64] Vincent Gay-Bellile, Adrien Bartoli, and Patrick Sayd. Direct estimation of nonrigid registrations with image-based self-occlusion reasoning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1) :87–104, 2010.

[65] Rahul Raguram and Jan-Michael Frahm. Recon : Scale-adaptive robust estimation via residual consensus. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1299–1306. IEEE, 2011.

[66] Hongwei Zheng, Ioan Cleju, and Dietmar Saupe. Highly-automatic mi based multiple 2d/3d image registration using self-initialized geodesic feature correspondences. In *Computer Vision–ACCV 2009*, pages 426–435. Springer, 2010.

[67] Enrico Gobbetti Ruggero Pintus and Roberto Combet. Fast and robust semi-automatic registration of photographs to 3d geometry. 2011.

[68] Peter J Neugebauer and Konrad Klein. Texturing 3d models of real world objects from multiple unregistered photographic views. In *Computer Graphics Forum*, volume 18, pages 245–256. Wiley Online Library, 1999.

[69] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision algorithms : theory and practice*, pages 298–372. Springer, 2000.

[70] Andrew W Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(13) :1145–1153, 2003.

[71] Shaoyi Du, Nanning Zheng, Shihui Ying, Qubo You, and Yang Wu. An extension of the icp algorithm considering scale factor. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 5, pages V–193. IEEE, 2007.

[72] Wenyi Zhao, David Nistér, and Steve Hsu. Alignment of continuous video onto 3d point clouds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8) :1305–1318, 2005.

[73] Jiaolong Yang, Hongdong Li, and Yunde Jia. Go-icp : Solving 3d registration efficiently and globally optimally. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1457–1464. IEEE, 2013.

[74] Matthew J Clarkson, Daniel Rueckert, Derek LG Hill, and David J Hawkes. Using photo-consistency to register 2d optical images of the human face to a 3d surface model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11) :1266–1280, 2001.

[75] Minh-Tri Pham, Oliver J Woodford, Frank Perbet, Atsuto Maki, Bjorn Stenger, and Roberto Cipolla. A new distance for scale-invariant 3d shape recognition and registration. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 145–152. IEEE, 2011.

[76] Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, Riccardo Gherardi, Andrea Fusiello, and Roberto Scopigno. Fully automatic registration of image sets on approximate geometry. *International journal of computer vision*, 102(1-3) :91–111, 2013.

[77] Dror Aiger, Niloy J Mitra, and Daniel Cohen-Or. 4-points congruent sets for robust pairwise surface registration. In *ACM Transactions on Graphics (TOG)*, volume 27, page 85. ACM, 2008.

[78] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational shape approximation. *ACM Transactions on Graphics (TOG)*, 23(3) :905–914, 2004.

[79] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[80] Sanjay Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on optimization*, 2(4) :575–601, 1992.

[81] Christoph Helmberg, Franz Rendl, Robert J Vanderbei, and Henry Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6(2) :342–361, 1996.

[82] P. Finsler. Uber das vorkommen definiter und semidefiniter formen in scharen quadratischer formen. *Comment. Math. Helv., 9*, pages 188–192, 1936/37.

[83] David Hilbert et al. Mathematical problems. *Bulletin of the American Mathematical Society*, 8(10) :437–479, 1902.

[84] David Hilbert. Über die darstellung definiter formen als summe von formenquadraten. *Mathematische Annalen*, 32(3) :342–350, 1888.

[85] V. Powers and T. Wörmann. An algorithm for sums of squares of real polynomials. *Journal of Pure and Applied Algebra*, 127(1) :99 – 104, 1998.

[86] M.D. Choi, T.Y Lam, and B. Reznick. Sums of squares of real polynomials. *Proceedings of Symposia in Pure Mathematics*, 2(58) :103–126, 1995.

[87] Gilbert Stengle. A nullstellensatz and a positivstellensatz in semialgebraic geometry. *Mathematische Annalen*, 207(2) :87–97, 1974.

[88] Mihai Putinar. Positive polynomials on compact semi-algebraic sets. In *Indiana Univ. Math. J.*, pages 969–984, 1993.

[89] S. Wagner. Archimedean quadratic modules : A decision problem for real multivariate polynomials. *Ph.D. thesis, Universität Konstanz*, August 2009.

[90] William Rey. *Introduction to robust and quasi-robust statistical methods*. Springer Science & Business Media, 2012.

[91] Dirk Holz, Christopher Lörken, and Hartmut Surmann. Continuous 3d sensing for navigation and slam in cluttered and dynamic environments. In *Information Fusion, 2008 11th International Conference on*, pages 1–7. IEEE, 2008.

[92] Jan W Weingarten, Gabriel Gruener, and Roland Siegwart. A state-of-the-art 3d sensor for robot navigation. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2155–2160. IEEE, 2004.

[93] Yasuhiro Taguchi, Yong-Dian Jian, Srikumar Ramalingam, and Chen Feng. Point-plane slam for hand-held 3d sensors. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5182–5189. IEEE, 2013.

[94] Alexander JB Trevor, John G Rogers III, Henrik Christensen, et al. Planar surface slam with 3d and 2d sensors. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3041–3048. IEEE, 2012.

[95] Yunsu Bok, Yekeun Jeong, Dong-Geol Choi, and In So Kweon. Capturing village-level heritages with a hand-held camera-laser fusion sensor. *International Journal of Computer Vision*, 94(1) :36–53, 2011.

[96] Brian Williams, Mark Cummins, José Neira, Paul Newman, Ian Reid, and Juan Tardós. A comparison of loop closing techniques in monocular slam. *Robotics and Autonomous Systems*, 57(12) :1188–1197, 2009.

[97] Maxime Lhuillier. Incremental fusion of structure-from-motion and gps using constrained bundle adjustments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(12) :2489–2495, 2012.

[98] Mohamed Tamaazousti, Vincent Gay-Bellile, Sylvie Naudet Collette, Steve Bourgeois, and Michel Dhome. Nonlinear refinement of structure from motion reconstruction by taking advantage of a partial knowledge of the environment. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3073–3080. IEEE, 2011.

[99] Alessandro Chiuso, Paolo Favaro, Hailin Jin, and Stefano Soatto. 3-d motion and structure from 2-d motion causally integrated over time : Implementation. In *Computer Vision—ECCV 2000*, pages 734–750. Springer, 2000.

[100] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages 652–659. IEEE, 2004.

[101] Reinhard Koch. Dynamic 3-d scene analysis through synthesis feedback control. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(6) :556–568, 1993.

[102] Andrew I Comport, Ezio Malis, and Patrick Rives. Real-time quadrifocal visual odometry. *The International Journal of Robotics Research*, 29(2-3) :245–266, 2010.

[103] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[104] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

[105] Andreas Nüchter, Kai Lingemann, Joachim Hertzberg, and Hartmut Surmann. 6d slam—3d mapping outdoor environments. *Journal of Field Robotics*, 24(8-9) :699–722, 2007.

[106] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion : Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[107] Christian Kerl, Jurgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2100–2106. IEEE, 2013.

[108] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping : Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research*, 31(5) :647–663, 2012.

[109] Alice E. Smith and David W. Coit. Penalty functions. In *University of Pittsburgh*, September 1995.

[110] David Nistér and Henrik Stewénius. A minimal solution to the generalised 3-point pose problem. *Journal of Mathematical Imaging and Vision*, 27(1) :67–79, 2007.

[111] K. Martin and Jakob W. Iterative closest point. In *Technical University of Denmark*, 2012.

[112] M.I. A. Lourakis and A.A. Argyros. Sba : A software package for generic sparse bundle adjustment. In *ACM Trans. Math. Software*, 2009.

[113] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics : The kitti dataset. *The International Journal of Robotics Research*, page 0278364913491297, 2013.

[114] Umberto Castellani and Adrien Bartoli. 3d shape registration. In *3D Imaging, Analysis and Applications*, pages 221–264. Springer, 2012.

[115] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

[116] Stéphane Christy and Radu Horaud. Iterative pose computation from line correspondences. *Computer vision and image understanding*, 73(1) :137–144, 1999.

[117] Luis Ferraz, Xavier Binefa, and Francesc Moreno-Noguer. Very fast solution to the pnp problem with algebraic outlier rejection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 501–508. IEEE, 2014.

[118] J Bazin, Hongdong Li, In So Kweon, Cédric Demonceaux, Pascal Vasseur, and Katsushi Ikeuchi. A branch-and-bound approach to correspondence and grouping problems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(7) :1565–1576, 2013.

[119] Grant Schindler, Panchapagesan Krishnamurthy, Roberto Lublinerman, Yanxi Liu, and Frank Dellaert. Detecting and matching repeated patterns for automatic geotagging in urban environments. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008.

[120] Andrew Mastin, Jeremy Kepner, and Jonathan Fisher. Automatic registration of lidar and optical images of urban scenes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2639–2646. IEEE, 2009.

[121] Xi Zhang, Gady Agam, and Xin Chen. Alignment of 3d building models with satellite images using extended chamfer matching. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 746–753. IEEE, 2014.

[122] Hongdong Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1074–1080. IEEE, 2009.

[123] Jiaolong Yang, Hongdong Li, and Yunde Jia. Optimal essential matrix estimation via inlier-set maximization. In *Computer Vision–ECCV 2014*, pages 111–126. Springer, 2014.

[124] Jan Verschelde. Algorithm 795 : Phcpack : A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software (TOMS)*, 25(2) :251–276, 1999.

[125] Gerald Schweighofer and Axel Pinz. Globally optimal o (n) solution to the pnp problem for general camera models. In *BMVC*, pages 1–10, 2008.

[126] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3) :796–817, 2001.

[127] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

[128] Dorit Borrmann, Jan Elseberg, Kai Lingemann, and Andreas Nüchter. The 3d hough transform for plane detection in point clouds : A review and a new accumulator design. *3D Research*, 2(2) :1–13, 2011.

[129] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics : Science and Systems*, volume 2, 2009.

[130] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[131] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 406–413. IEEE, 2014.

[132] Christoph Strecha, Wolfgang von Hansen, L Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[133] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Adaptive structure from motion with a contrario model estimation. In *Computer Vision–ACCV 2012*, pages 257–270. Springer, 2012.

[134] Andrea Fusiello, Arrigo Benedetti, Michela Farenzena, and Alessandro Busti. Globally convergent autocalibration using interval analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(12) :1633–1638, 2004.

[135] Manmohan Chandraker, Sameer Agarwal, Fredrik Kahl, David Nistér, and David Kriegman. Autocalibration via rank-constrained estimation of the absolute quadric. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[136] Manmohan Chandraker, Sameer Agarwal, David Kriegman, and Serge Belongie. Globally optimal algorithms for stratified autocalibration. *International journal of computer vision*, 90(2) :236–254, 2010.

[137] Adlane Habed, Danda Pani Paudel, Cédric Demonceaux, and David Fofi. Efficient pruning lmi conditions for branch-and-prune rank and chirality-constrained estimation of the dual absolute quadric. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 493–500. IEEE, 2014.

[138] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *Robotics & Automation Magazine, IEEE*, 18(4) :80–92, 2011.

[139] Richard I. Hartley. Chirality. *International Journal of Computer Vision (IJCV)*, pages 41–61, January 1998.

[140] Ramon E Moore. *Interval analysis*, volume 4. Prentice-Hall Englewood Cliffs, 1966.

[141] Riccardo Gherardi and Andrea Fusiello. Practical autocalibration. In *Computer Vision–ECCV 2010*, pages 790–801. Springer, 2010.

[142] Vincent Rabaud. Vincent's Structure from Motion Toolbox. http://vision.ucsd.edu/~vrabaud/toolbox/.

Document réalisé avec LaTeX et :
le style LaTeX pour Thèse de Doctorat créé par S. Galland — http://www.multiagent.fr/ThesisStyle
la collection de paquets `tex-upmethodology` — http://www.arakhne.org/tex-upmethodology/

Abstract:

In this thesis, we study the problem of registering 2D image sets and 3D point clouds under three different acquisition set-ups. The first set-up assumes that the image sets are captured using 2D cameras that are fully calibrated and coupled, or rigidly attached, with a 3D sensor. In this context, the point cloud from the 3D sensor is registered directly to the asynchronously acquired 2D images. In the second set-up, the 2D cameras are internally calibrated but uncoupled from the 3D sensor, allowing them to move independently with respect to each other. The registration for this set-up is performed using a Structure-from-Motion reconstruction emanating from images and planar patches representing the point cloud. The proposed registration method is globally optimal and robust to outliers. It is based on the theory Sum-of-Squares polynomials and a Branch-and-Bound algorithm. The third set-up consists of uncoupled and uncalibrated 2D cameras. The image sets from these cameras are registered to the point cloud in a globally optimal manner using a Branch-and-Prune algorithm. Our method is based on a Linear Matrix Inequality framework that establishes direct relationships between 2D image measurements and 3D scene voxels.

Keywords: Registration, Camera Calibration, Structure-from-Motion, Projective Geometry, Mathematical Optimization

SPIM

■ École doctorale SPIM - Université de Bourgogne/UFR ST BP 47870 F - 21078 Dijon cedex
■ tél. +33 (0)3 80 39 59 10 ■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

UNIVERSITÉ DE BOURGOGNE