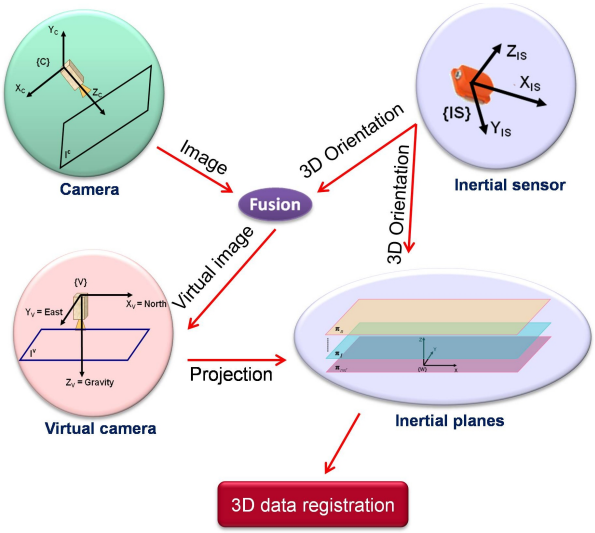
University of Coimbra
Faculty of Science and Technology
Department of Electrical and Computer Engineering

Exploiting Inertial Planes for Multi-sensor 3D Data Registration



PhD Dissertation

Hadi Aliakbarpour

Coimbra, May 2012

University of Coimbra
Faculty of Science and Technology
Department of Electrical and Computer Engineering

Exploiting Inertial Planes for Multi-sensor 3D Data Registration

Dissertation submited:

to the Electrical and Computer Engineering Department of the Faculty of Science and Technology of the University of Coimbra in partial fulfilment of the requirements for the Degree of Doctor of Philosophy.

Hadi Aliakbarpour
Coimbra, 2012

This dissertation is realized under Supervision of

Professor Doctor Jorge Dias
Professor of the
Faculty of Science and Technology, University of Coimbra

This thesis is dedicated to my parents, Hossein and Zahra, my brother Hassan, my son Sadra, and to my wife Shima.

Resumo

Esta dissertação aborda o tema da utilização de planos inerciais para registo de dados tridimensionais de objetos observados por uma rede de camaras dotadas de sensores inerciais. Cada câmera está rigidamente ligada a um sensor inercial (SI). Utilizando as informações provenientes de cada um dos sensores é proposto um método de reconstrução tridimensional, sem necessidade de partir do pressuposto que o chão é plano. Para além disso, em cada par (câmera-SI), o SI é utilizado para definir uma câmera-virtual cujo plano da imagem é horizontal e alinhado com a gravidade terrestre (direções cardinais terrestres). O SI é também utilizado para definir um conjunto de planos Euclidianos inerciais. O plano da imagem de cada câmera virtual é projetado sobre este conjunto de planos-inerciais, que são paralelos e horizontais; usa-se para tal transformações homográficas. Os modelos geométricos são apresentados e a sua visualização é feita em tempo real pela utilização de um algoritmo de reconstrução implementado por um sistema de processamento gráfico. Foram para tal investigadas as relações geométricas entre os diferentes planos de imagem projetados e os planos inerciais Euclidianos, e para cada caso particular foi obtida uma função homográfica paramétrica. Foi ainda proposta uma rquitetura de processamento paralelo a fim de executar em tempo real a reconstrução volumétrica. A capacidade de se processar em tempo real é obtida através da implementação do algoritmo de reconstrução em uma unidade de processamento gráfico (GP-GPU), utilizando Compute Unified Device Architecture (CUDA). Nós aproveitamos o facto de termos cada câmera ligada a um sensor inercial, e propusemos um método para estimar os parâmetros extrínsecos entre as câmeras dentro da rede. Devido à imperfeição das observações provenientes

dos sensores ou dos algoritmos de estimação, os dados obtidos contêm algumas incertezas. Estar ciente de tais incertezas é muito importante na fase de fusão da informação proveniente de diferentes nós de uma rede de sensores, e também em outras aplicações que utilizam os dados recolhidos. Para este efeito, nós usamos a geometria estatística, e modelizamos as incertezas em todas as transformações homográficas envolvidas no âmbito do processo, e também na propagação de erro sobre os dados recolhidos. Além disso, alguns aspetos importantes, tais como uma configuração apropriada da rede de câmeras, filtragem de baixo nível dos dados, e integração de visão móvel com sensores de laser dentro da rede de câmeras, foram também alvo de pesquisa nesta dissertação. Para este estudo existe uma variedade de aplicações práticas, diferentes áreas podem beneficiar da estrutura de registo de dados tridimensional aqui proposta. Estas áreas incluem: vigilância de movimentos humanos, captura e modelos de comportamento humano, realidade virtual, jogos, teleconferências, interação humano-robô, indústrias médicas.

Abstract

This dissertation explores the use of inertial planes for the purpose of scene 3D data registration. The scene is observed by a network of cameras and inertial sensors where each camera is rigidly coupled to an inertial sensor. Taking advantage of inertial sensor (IS), a 3D reconstruction method is proposed with no planar ground assumption. Moreover, IS in each couple is used to define a virtual camera whose image plane is horizontal and aligned with the earth cardinal directions. The IS is furthermore used to define a set of Euclidean inertial planes in the scene. The image plane of each virtual camera is projected onto this set of parallel-horizontal inertial-planes, using homography transformations. Geometric relations among different projective image planes and Euclidean inertial planes of the framework are investigated and for each particular case a parametric homography function is obtained. A parallel processing architecture is proposed in order to perform real-time volumetric reconstruction. The real-time characteristic is obtained by implementing the reconstruction algorithm on a graphics processing unit (GP-GPU) using Compute Unified Device Architecture (CUDA). We take the advantage of having each camera coupled to IS and proposed a method to estimate the extrinsic parameters among the cameras within the network. Due to the imperfectness of the sensor observations or estimation algorithms the obtained data are corrupted and contain some uncertainties. To be aware of such uncertainties can be of importance for the fusion stage of the information coming from different nodes in a sensor network and as well for further applications which will use the registered data. For this purpose we use statistical geometry and modelize the uncertainties in all involved homography transformations within the framework and

their error propagations on the registered data. Moreover, some relevant issues, such as an appropriate camera configuration in the sensor network, low-level data filtering of the scene's dynamic and integration of mobile vision and laser sensor within a camera network, are also investigated in this dissertation. There is a variety of applications from different areas which can benefit from the proposed 3D data registration framework. These areas include surveillance, human motion capturing and behaviour modelling, virtual-reality, games, tele-conferencing, human-robot interaction, medical industries, and scene and object understanding.

Acknowledgment

This thesis could never have been realized without the support of many people. I would like to express my appreciation and gratitude to my advisor, Prof. Jorge Dias, for his support, time, and advice throughout my PhD studies.

My thanks to the professors and colleagues from ISR, like Anìbal T. de Almeida, Jorge Lobo, Paulo Menezes, Rui Rocha, João Barreto, João Filipe Ferreira, Luis Mirisola, Omar Tahri, Luis Santos, João Quintas, Paulo Freitas, Pedro Trindade, Kamrad Khoshhal, Diego Faria, Jose Prado, Mahmoud Tavakoli, Ali Marjovi, Amilcar Ferreira, Ivone Amorim, David Portugal, Seyed Jafar Hosseini, Abed Malti, Tito, Ricardo Martin, Filipe Ferreira, Catia Pinho, Paula Lopes, Rita Catarino, Sirvan Khalighi, Rui Freire, Ricardo Carvalho, Luis Davim, Jörg Rett, José Marinho, Hugo Faria, Alexandre Malhão and Christiana Tsiourti.

I would like to thank Luis Almeida, for his contribution in the real-time implementation of the system. Professor Pedro Manuel Urbano de Almeida Lima from Lisbon Technical University (IST) for his valuable comments. My Spanish colleagues from University of Extremadura like Leandro Serrano, Luis Manso, Pedro Nunez and Professor Antonio Bandeira from University of Malaga. My French colleagues from Probayes® research company like Kamel Mekhnacha and Julien Ros. My colleague from Technical University of Munich, Martin Hofmann.

I would like to thank the FCT-Fundação para a Ciência e a Tecnologia- for supporting my work with the Grant "SFRH/BD/45092/2008".

Special thanks to my dear brother, Hassan, if it were not for his inspiration, I

would not have come half as far. Most of all I would like to thank my wife, Shima, whose love and devotion have been incredible. My lovely son, Sadra, who was just a few month-old, when we came to Portugal and started my PhD studies. My faithful and thoughtful dad, my kind mother and sympathetic sisters. My wife's family who has always been supportive.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In the context of computer vision, three-dimensional (3D) data registration refers to the process of fusing two-dimensional (2D) images, captured by cameras, in order to retrieve 3D coordinates of a scene. A camera captures the visible space as a 2D digital image plane. Inertial sensor (IS) is an electronic device which is able to measure 3D orientation between a rigid body coordinate system and the earth-fixed coordinate system using a combination of magnetometer and accelerometer. In this dissertation, we study the subject of 3D reconstruction of a scene using 2D images of cameras accompanied by 3D orientation provided by inertial sensors. Recovering 3D information is demanded by a variety of applications from different areas including surveillance, human motion and behaviour modelling, virtual-reality, smart-room, health-care, games, teleconferencing, human-robot interaction, medical industries and scene and object understanding.

1.1 Motivation

One of the primary tasks in many of the aforementioned applications is to register 3D data of the scene. Performing 3D data registration and scene reconstruction using a set of planar images is still one of the key challenges of computer vision. A network of cameras, whose usage and ubiquitousness have been increasing in the last decade, can provide such planar images from different views of the scene. Recently, IS has been becoming much cheaper and more available so that nowadays most smart-phones are equipped in both IS and camera sensors. 3D earth cardinal orientation (North-East-Down) is one of the outputs of an IS. The mentioned demand for 3D data registration and availability of cameras already coupled with IS motivated us to investigate how availability of a network of such camera-IS couples can give benefit for the purpose of multi-sensor 3D data registration.

1.2 Related works

Previous works related to the scope of this thesis fall into five categories. First, 3D reconstruction using vision. Second, the use of IS to accompany computer vision. Third, related works to real-time implementation of the computer vision algorithms using GP-GPU. Forth, the modelization of uncertainties of homography transformations and eventually some works related to the issue of extrinsic parameters estimation.

1.2.1 Multi-view 3D reconstruction

There have been many works in the area of 3D reconstruction. Some researchers used homography transformation for this purpose. Khan in [KYS07] proposed a homographic framework for the fusion of multi-view silhouettes. They estimate homographies using the three vanishing points of the reference plane in the scene. In a similar approach, Michoud et al. [MGB07] introduced a marker-less 3D human motion capturing approach using multiple views. Zhang in [ZWW03] introduced an algorithm for 3D projective reconstruction based on infinite homography. Their contribution is improvement of 4 points based method of Hartley and Rother et al. They proposed a linear algorithm based on 3 points on a reference plane which is visible in all views. Homography-based mapping is used to implement a 3D reconstruction algorithm by Zhang and Hanson in [ZH96]. Khan in [Kha08] proposed some algorithms to track, reconstruct and object classification by using a homographic occupancy constrain. Wada et al. [WWTM00] studied a 3D reconstruction method using homography transformation. They presented a parallel volume intersection method based on plane-to-plane homography for real-time 3D volume reconstruction using active cameras where the focus is on the acceleration of back-projection from silhouette images to 3D space. Lai and Yilmaz in [LY08] used images from uncalibrated cameras for performing projective reconstruction of buildings based on Shape From Silhouette (SFS) approach where buildings structure is used to compute vanishing points. The achieved reconstruction is a metric

recovery up to a scale factor and homography transformations are used. Lee et al. in [LY10] applied a 3D reconstruction method using photo consistency in images taken from uncalibrated multiple cameras. A dynamic calibration and 3D reconstruction using homography transformation is proposed by Zhang and Li in [ZL05]. Metric 3D reconstruction for large structures from uncalibrated images and using homography techniques is investigated by Tang et al. [TWH+07]. They used structure lines to obtain vanishing points.

Some researchers use non-homography methods for perform 3D reconstruction. Sorman et al. in [SZB+07] presented a multi-view reconstruction method based on volumetric graph-cuts. A multi-resolution volumetric 3D object reconstruction has been proposed by Guerchouche et al. in [GBZ08]. 3D object reconstruction of an object using uncalibrated images taken by a single camera is proposed by Azevedo et al. in [ATV09]. They used active computer vision method for the 3D reconstruction of objects from image sequences. In their work Structure From Motion (SFM) is used to recover the 3D shape of an object based on the relative involved motion and photo-consistency (voxel coloring) is used to perform the volumetric reconstruction. Jethwa in [Jet04] proposed a method to perform efficient voxel-based reconstruction of urban environments using a large set of images. Color and silhouette information from multiple views are fused by Khan and Shah [KS08] for reconstructing articulated objects in monocular video. Sinha in [Sin09] studied how silhouettes extracted from images and video can help both multi-view camera calibration and 3D surface reconstruction from multiple images. Maitre et al. [MSD08] investigated a method to perform multi-view reconstruction of a scene by using camera arrays. Ruwwe et al. in [RKR+08] proposed an approach for image registration based on reconstructed 3D octrees by voxel carving.

Michoud in [MSEH08] proposed a method to eliminate appearing ghost object in SFS-based 3D reconstructions. Ghost is an extra object which does not exist in the real scene but when there are some cases of visual ambiguities in the silhouettes it can be seen in the reconstructed scene [MSEH08].

There are some related surveys on the different 3D reconstruction techniques.

Kordelas et al. in [KAHD] provided a survey for existing 3D model reconstruction algorithms where the approaches are categorized by the devices by which the data are acquired (laser range finder and camera). Different 3D reconstruction methods based on visual hull approach are compared and evaluated by Fredriksson in [Fre11]. A survey on motion-parallax-based 3-D reconstruction techniques is provided by Lu et al. in [LZWnL04]. Some efficient methods for the 3D reconstruction of static and dynamic scenes from stereo images, stereo image sequences, and images captured from multiple viewpoints are explored in [Leu05] by Leung. Different multi-view stereo reconstruction algorithms are compared and evaluated on a common ground truth by Seitz et al. [SCD+06]. A Comparison between different computer vision methods for real-time 3D reconstruction for the use in mobile robots has been done by Dornauer et al. in [DKBN08].

As mentioned before, 3D reconstruction can be useful for many applications. Feldmann et al. [FMS+10] utilized the volumetric 3D reconstruction for the aim of on-line body motion tracking system. Brice et al. [BES05] investigated the use of multi-view geometry to human model and pose reconstruction. Luo et al. [LBTV10] introduced a method to estimate human pose for multiple persons based on volume reconstruction. The use of 3D information in the field of cultural heritage is investigated by Vergauwen [VVG06] where a web-based 3D reconstruction service is proposed. Capturing of complex human movements from multiple views is studied in [Keh05] by Kehl. 3D reconstruction of natural underwater scenes using an stereo-vision is studied by Brandou et al. [BAP+07]. A human body posture estimation method based on back projection of human silhouette images is proposed by Takahashi et al. [KT07]. Uriol [Uri05] used a camera network to reconstruct human and synthesis an avatar. A system for real-time 3D human visual hull reconstruction and skeleton voxels extraction is proposed by Yang et al. [YZL+09].

Some researchers used more modalities rather than just images for performing 3D reconstruction. Guomundsson et al. in [GPC+10] investigated the improvement of 3D reconstruction in an smart-room by using ToF imaging. Jiang and

Lu [JL06] fused color intensity images and laser range data to perform panoramic 3D reconstruction of scene. A hybrid surface reconstruction method that fuses geometrical information acquired from silhouette images and optical triangulation is presented in [YW07] by Yemez and Wetherilt. Kim et al. in [KTD+09] propose a multi-view sensor fusion approach that combines information from multiple color cameras and multiple ToF depth sensors for the sake of 3D reconstruction. Fusion of laser range and image for the purpose 3D reconstriction is studied by Bok et al.[BHK07]. Guan et al. [GFP08] proposed a method to perform 3D reconstruction by fusion of data from camera and ToF.

1.2.2 Using IS to accompany vision

The use of inertial sensors to accompany compute vision applications is recently attracting attentions of the researchers. Nowadays, IS has become much cheaper and more available. Thanks to the availability of MEMS* chipsets, there are many smart-phones which are equipped with this sensor and camera as well. Dias et al. [DLA02, LAAD03] investigated the cooperation between visual and inertial information. Lobo and Dias [LD07] proposed an efficient method to estimate the relative pose of a camera and an IS. The use of IS with a stereo camera for the purpose of world feature detection is investigated in [LQD03] by Lobo and Dias. Mirisola in [MDdA07] used a rotation-compensated imagery for the aim of trajectory of an airship by aiding inertial data. Fusion of image and inertial data is also investigated by Bleser et al. [BWBS06, BS08] for the sake of tracking in the mobile augmented reality. Ababsa in [Aba09] used inertial sensor orientation and GPS position for performing 3D reconstruction of urban scenes. Zendjebil and Ababsa in [ZADM10] investigated the use of GPS-IS-camera for 3D localization of an outdoor mobile robot and moreover provided some calibration methods among these sensors. In [HGJ07] IS and stereo vision are used for underwater environment reconstruction. In [HYWH10] IS is used to calibrate a camera network with no overlapping FOV (field of view) by Hsieh. Inertial data is augmented with monoc-

*Micro-Electro-Mechanical-Systems

ular video to perform 3D environment reconstruction in [CLT06] by Clark et al. Fusion of image data with location and orientation sensor data streams for the purpose of camera trajectory recovering and scene reconstruction is investigated by Gat et al. [GKN10]. Gat et al. [GKN10] fused inertial information together with geographical data and images from a video stream recorded by a mobile camera in order to reconstruction the camera trajectory for the purpose of consumer video applications. Besdok in [Bes09] proposed a method to calibrate a pair of cameras using IS attached to a calibration pattern, where RBF neural networks are used for training the system. Randeniya et al. [RGSN08] proposed a method to estimate the intrinsic parameters for a camera and the extrinsic parameters among the camera and IS. Their approach is mentioned to be effective and precise for Intelligent Transportation Systems applications with large filed of view and capable of functioning in manoeuvres. In [MR08] a Kalman filter-based algorithm to calibrate an IS and camera couple is proposed by Mirzaei et al. . In [OD02] Okatani et al. demonstrated that how the translation of camera between two images can be robustly estimated by using IS. Based on Okatani's work, Labrie and Hebert in [LH07] showed that how the camera 3D motion recovery can be improved by the using inertial data. Brodie et al. in [BWP08] proposed a re-calibration method for IS in order to noticeably reduce its 3D orientation error. Kalantart et al. in [KHJG11] proposed a solution to the relative orientation problem between two cameras where the accaracy of IS is improved less that 0.001°.

1.2.3 Real-time implementation using GP-GPU

In order to have a real-time processing time many researchers have already started to use GPU-based (GP-GPU and CUDA) parallelization of their algorithms. Joao Filipe and Lobo et al. in [FLD10] proposed a real-time implementation of Bayesian models for perception through multi-modal sensors by using CUDA. Camera calibration and real-time image distortion correction are performed in [MBF12] by Melo and Barreto et al. for the medical endoscopy applications. Almeida and Menezes et al. implemented the stereo vision head vergence using GPU-based

cepstral filtering [AMD11]. A GPU-based background segmentation algorithm is proposed in [GRNG05] by Griesser et al. Ziegler in [Zie10] proposed a GPU data structure for graphic and vision. Real-time space carving using CUDA is investigated in [NNT07] by Nitschke et al. In [SHT+08] CUDA is used to accelerate advanced MRI reconstructions. GPU-based method is used in [WFEK09] by Waizenegger for the purpose of high resoulution and real-time reconstruction using visual hulls. A GPU-based shape from silhouette (SFS) algorithm is implemented in [YLKC07] by Yous et al. An approach for volumetric visual hull reconstruction, using a voxel grid that focuses on the moving target object, is proposed by Knoblauch et al. [KK09]. A real-time 3D reconstruction system is presented in [LBN08] by Ladikos et al. to achieve real-time performance. Yguel et al. in [YAL06] implemented a GPU-based construction of occupancy grids using several laser range-finders. Brisc [Bri08] investigated the issue of Image-based Rendering and Modeling (IBMR) and its implemetation on GPU, where the capturing, geometric and photometric aspects of an IBMR system were studied. A photo consistency based 3D reconstruction is proposed and implemented on GPU by Hornung et al. [HHK06]. Kuhn and Henrich [KH09] proposed a method for reconstructing multiple objects within a known environment which presence of occlusions, where the implementation was done on GPU.

1.2.4 Uncertainty modelling of homography transformations

In the context of modelling the uncertainty in a homography transformation there exist a few works. To the best of our knowledge, all of the carried researches on this subject are for cases where the homography transformation is estimated by using point correspondences.

Criminisi et al. in [CRZ99] discussed about the uncertainty of homography mapping applied for measuring device. In their work the uncertainty is analysed in cases such as number of used point correspondences or the uncertainties in localization of those points. A general geometric reasoning with uncertain 2D point

and lines is mathematically defined in [MBF09] by Meidow et al. Negahdaripour et al. in [NPG05] studies the accuracy of planar homography in applications such as video frame-to-frame homography. Ochoa and Belongie in [OB06] presented an approach to determine a search region for use in guided matching under projective mappings. The problem of finding optiomal point correspondences between images related by homography transformation is addressed by Chum et al. in [CPS05]. They studies that given an homography transformation and a pair of matching points, how to determine a pair of points that are exactly consistent with the homography and also minimize the geometric error. Baker et al. in [BDK06] studied the parametrization of homography to maximize the plane estimation performance. They compared their method with the usual estimation method with a parametrization that combines 4 fixed points in one of the images with 4 variable points in the other image. A method to estimate planar projective transformation is introduced by Chi et al. in [CHY11]. In their work they proposed a method to register 2D points set which reduces the search space for the homographies from eight-dimensional space to a three-dimensional case.

1.2.5 Extrinsic parameters estimation

Estimation of extrinsic parameters in a sensor network is a crucial and demanded issue for many applications such as 3D data registration, tracking, mobile robotics, Human-Computer Interaction (HCI), human behaviour understanding and surveillance. Vasconcelos and Barreto et al. in [VBN12] proposed an algorithm for the extrinsic calibration of a camera and an 2D laser range finder (LRF). The problem of estimating the rigid displacement between the two sensors is formulated as the one of registering a set of planes and lines in the 3D space. The authors proved that the alignment of 3 plane-line correspondences has at most 8 solutions. Images of planar mirror reflections are used to estimate camera pose by Rodrigues and Barreto et al. in [RBN10]. Barreto et al. in [BRSF09] proposed an algorithm to calibrate cameras with lens distortion using a single image of a planar chessboard pattern acquired in general position.

In [WS95] a calibration process based upon a specific calibration pattern is used to identify the transformation between laser range finder and camera. An approach for the extrinsic calibration of a camera with a 3D laser range finder is proposed in [SHS07] by Scaramuzza. Mei in [MR06] presents some methods for estimating the relative position of a central catadioptric camera and a laser range finder in order to obtain depth information in the panoramic image. Schweiger in [SBS08] introduced a plane based approach to calibrate a LRF-camera system in order to determine both intrinsic and extrinsic parameters. Lobo and Dias in [LD07] proposed a novel approach to estimate the relative pose calibration between visual and inertial sensors. Ferreira and Dias in [FPD08] investigated the implementation and calibration of a Bayesian binaural system for the aim of 3D localization. Homographies among image planes of a camera network are used to calibrate a camera network by Cao and Foroosh in [CF04]. The issues of multi-camera calibration and object tracking are jointly investigated by Porikli and Divakaran [PD03] and by Meingast et al. [MOS07]. Localization of a network of non-overlapping surveillance cameras using an optimization method is investigate by Micusik et al. in [MP10]. Similar topic is also studied by Esquivel et al. in [EWK07] and by Kumar et al. in [KIFP08]. Auto-calibration of a network of PTZ cameras with non-overlapping field of view as well is investigated by Ashraf and Foroosh in [AF08]. Beriault in [BPC07] proposed a method for multi-camera network calibration for the sake of human gesture monitoring. In their approach, the relative cameras positions are estimated through waving a red light in a synchronized setup. Chen in [CPMH03] introduced a method to estimate epipole under a pure camera translation. Hu and Tan in [HT06] proposed an approach for depth recovery and affine reconstruction under pure camera translation. In [HL07] vanishing points are used for camera calibration in a vision system by He and Lei. Svoboda in [SMP06] proposed a method for camera network calibration. His method works by waving a bright spot through the working volume in order to make a set of virtual 3D points. Spaan and Lima in [SL09] proposed an approach to dynamic sensor selection in a camera networks. Barreto and Daniilidis in [BD04] investigated the problem of multiple camera calibration and estimation of radial

distortion. Their approach is based on finding correspondences between views. The correspondences are obtained by deliberately moving an LED in thousands of unknown positions in front of the cameras. Meijer in [MLM07] investigated the multi camera calibration problem applied to localization. In his approach a LED is used as calibration object. Faria and Aliakbarpour et al in [FAD09] performed a calibration method to estimate the extrinsic parameters among a Polhemus Tracker and an stereo camera. Calibrating a distributed camera network is deeply investigated in [DR04, DRC06, DR07, Dev07] by Devarajan.

In this thesis the use of inertial planes for the purpose of 3D data registration is explored. The scene is observed by a network of cameras and inertial sensors where each camera is rigidly coupled to an inertial sensor. Taking advantage of inertial sensor (IS), a 3D reconstruction method is proposed with no planar ground assumption. Moreover, IS in each couple is used to define a virtual camera whose image plane is horizontal and aligned with the earth cardinal directions. The IS is furthermore used to define a set of Euclidean inertial planes in the scene. The image plane of each virtual camera is projected onto this set of parallel-horizontal inertial-planes, using homography transformations. Geometric relations among different projective image planes and Euclidean inertial planes of the framework are investigated and for each particular case a parametric homography function is obtained. A parallel processing architecture is proposed in order to perform real-time volumetric reconstruction. The real-time characteristic is obtained by implementing the reconstruction algorithm on a graphics processing unit (GP-GPU) using Compute Unified Device Architecture (CUDA). Due to the imperfectness of the sensor observations or estimation algorithms the obtained data contain some uncertainties. To be aware of such uncertainties can be of importance for the fusion stage of the information coming from different nodes in a sensor network and as well for further applications which will use the registered data. For this purpose we use statistical geometry and modelize the uncertainties in all involved homography transformations within the framework and their error propagations on the registered data. We take the advantage of having each camera coupled to IS and proposed a method to estimate the extrinsic parameters among the cameras

within the network. Moreover some relevant issues, such as an appropriate camera configuration in the sensor network, low-level data filtering of the scene's dynamic and integration of mobile vision and laser sensor within a camera network, are investigated in this dissertation.

1.3 Contributions

This thesis provides a number of novel contributions to the multi-sensor 3D data registration field of research. The primary contributions of this research are as follows:

- A homographic framework is developed for 3D data registration using a network of cameras and inertial sensors. Geometric relations among different projective image planes and Euclidean inertial planes involved in the framework are explored. [AD12a] [AD11c] [AD10b] [AD11b] [AD10a] [AFKD10] [AFQ$^+$11].
- A real-time prototype of the framework is developed which is able to perform fully reconstruction of human body (and objects) in a large scene. The real-time characteristic is achieved by using a parallel processing architecture on a CUDA-enabled GP-GPU [AAMD11].
- A two-point-based method to estimate translations among virtual cameras in the framework is proposed and verified [AD12a] [AD11a] [AD10a] [AFQ$^+$11].
- The uncertainties of the homography transformations involved in the framework and their error propagations on the image planes and Euclidean planes have been modelized using statistical geometry.
- Within the context of the proposed framework, a genetic algorithm is developed to provide an optimal coverage of the camera network to a polygonal object (or a scene) [AD12b].

- A method to estimate extrinsic parameters among camera and laser range finder is developed [ANP+09]. A related toolbox* is prepared.

*SLaRF; available to download at `http://paloma.isr.uc.pt/~hadi`

1.4 Publications

Most of the thesis is based on the following publications and achievements:

1.4.1 Peer-reviewed journal articles

- Multi-sensor 3D Volumetric Reconstruction Using CUDA. Hadi Aliakbarpour, Luis Almeida, Paulo Menezes, and Jorge Dias. Journal of 3D Research, Springer, 2:1-14, 2011.10.1007/3DRes.04(2011)6, 2011 .
- 3D Reconstruction based on Multiple Virtual Planes by Using Fusion-based Camera Network. Hadi Aliakbarpour and Jorge Dias. Journal of Computer Vision (IET), 2012 (accepted).
- Geometric Exploration of Inertial-planes for Multi-layer 3D Data Registration. Hadi Aliakbarpour and Jorge Dias. ACM Transactions on Sensor Networks (TOSN), 2012 (under review).

1.4.2 International awards

- Best Runner-up Paper Award for the paper "IMU-aided 3D Reconstruction based on Multiple Virtual Planes", at DICTA'10 (the Australian Pattern Recognition and Computer Vision Society Conference), IEEE Pr., December 2010, Sydney, Australia.

1.4.3 Peer-reviewed international conference papers

As the first author

- Volumetric 3D reconstruction without planar ground assumption, Aliakbarpour, H. and Dias, J., Distributed Smart Cameras (ICDSC), 2011 Fifth ACM/IEEE International Conference on , pp. 1 -2 , 2011.

- Multi-resolution Virtual Plane based 3D Reconstruction using Inertial-Visual Data Fusion. Aliakbarpour, H. and Dias, J., International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP 2011), 5-7 March 2011, Algarve, Portugal. , 2011.

- Mobile Robot Cooperation with Infrastructure for Surveillance: Towards Cloud Robotics. Hadi Aliakbarpour, João Quintas, Paulo Freitas and Jorge Dias. Accepted by Workshop on Recognition and Action for Scene Understanding (REACTS) in the 14th International Conference of Computer Analysis of Images and Patterns (CAIP), September 2011, Spain.

- Inertial-Visual Fusion For Camera Network Calibration. Aliakbarpour, H. and Dias, J., IEEE 9th International Conference on Industrial Informatics (INDIN 2011), July 2011. , 2011.

- Human Silhouette Volume Reconstruction Using a Gravity-based Virtual Camera Network, Aliakbarpour, H. and Dias, J., Proceedings of the 13th International Conference on Information Fusion, 26-29 July 2010 EICC Edinburgh, UK , 2010.

- IMU-aided 3D Reconstruction based on Multiple Virtual Planes. Aliakbarpour, H. and Dias, J., DICTA'10 (the Australian Pattern Recognition and Computer Vision Society Conference), IEEE Pr., 1-3 December 2010, Sydney, Australia, 2010.

- A Novel Framework for Data Registration and Data Fusion in Presence of Multi-modal Sensors. Aliakbarpour, H.; Ferreira, J. F.; Khoshhal, K. and Dias, J., in Proceedings of DoCEIS2010- Emerging Trends in Technological Innovation, IFIP AICT 314-2010, Springer. , Vol. 314/2010 , pp. 308-315 , 2010.

- HMM-based Abnormal Behaviour Detection Using Heterogeneous Sensor Network. Hadi Aliakbarpour, Kamrad Khoshhal, João Quintas, Kamel Mekhnacha, Julien Ros, Maria Andersson and Jorge Dias. DoCEIS 2011, Technological Innovation for Sustainability, IFIP, Volume 349/2011, Springer.

- An Efficient Algorithm for Extrinsic Calibration between a 3D Laser Range Finder and a Stereo Camera for Surveillance. Aliakbarpour, H.; Nunez, P.; Prado, J.; Khoshhal, K. and Dias, J., 14th International Conference on Advanced Robotics (ICAR 2009) , 2009.

As a co-author

- Parametrizing Interpersonal Behaviour with Laban Movement Analysis; Kamrad Khoshhal, Luis Santos, Hadi Aliakbarpour and Jorge Dias. Accepted at Workshop on Socially Intelligent Surveillance and Monitoring. The 25th Conference on Computer Vision and Pattern Recognition, CVPR, 2012, USA.

- LMA-based Human Behaviour Analysis Using HMM; Kamrad Khoshhal, Hadi Aliakbarpour, João Quintas, Kamel Mekhnacha, Julien Ros and Jorge Dias. DoCEIS 2011,Technological Innovation for Sustainability, IFIP, Volume 349/2011, Springer.

- Fusion of Multi-Modal Sensors in a Voxel Occupancy Grid for Tracking and Behaviour Analysis. Martin Hofmann, Moritz Kaiser, Hadi Aliakbarpour and Gerhard Rigoll. International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2011), University of Delft, 2011, Netherlands.

- Probabilistic LMA-based Human Motion Analysis by Conjugating Frequency and Spatial based Features, Kamrad Khoshhal Roudposhti, Hadi Aliakbarpour, João Quintas, Martin Hofmann and Jorge Dias. In the proceeding of International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2011), University of Delft, Netherlands.

- Probabilistic LMA-based Classification of Human Behaviour Understanding Using Power Spectrum Technique, Kamrad Khoshhal, Hadi Aliakbarpour, João Quintas, Paulo Drews and Jorge Dias. In the Proceedings of the 13th

International Conference on Information Fusion, 26-29 July 2010 EICC Edinburgh, UK.

- Geneva Brain computer Interface for robot steering in virtual and real environments. Rolando Menendez, Jorge Dias, Jose Prado, Hadi Aliakbarpour and Sara Gonzalez Andino. 16th Annual Meeting of the Organization for Human Brain Mapping (HBM2010), June 2010, Barcelona, Spain.

- Using Concurrent Hidden Markov Models to Analyse Human Behaviours in a Smart Home Environment, João Quintas, Kamrad Khoshhal Roudposhti, Hadi Aliakbarpour, Martin Hofmann and Jorge Dias. International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2011), University of Delft, Netherlands.

- Grasping Movements Recognition in 3D Space using a Bayesian Approach, Diego R. Faria, Hadi Aliakbarpour and Jorge Dias. 14th International Conference on Advanced Robotics (ICAR 2009), Munich

- Multi-class Brain Computer Interface Based On Visual Attention, Rolando Menendez, Jorge Dias, Jose Prado, Hadi Aliakbarpour and Sara Gonzalez. European Symposium on Artificial Neural Networks Advances in Computational Intelligence and Learning Bruges, Belgium 2009.

1.5 Dissertation outline

In the next chapter we present the concept of using inertial sensors for 3D data registration. A framework is proposed where 3D orientation provided by IS and the concept of homography transformation are used to define virtual image planes and Euclidean planes for the purpose of data registration.

Chapter 3 specific geometric relations among different Euclidean virtual planes and projective virtual image planes are explored and a parametric equation for each particular case is obtained. Moreover a method to estimate the translation

vectors among virtual cameras within the network is proposed. After that, the uncertainties of the involved homographies and mapped points through them on the planes are modelized in this chapter.

In chapter 4 we present a real-time implementation of the framework proposed. It introduces an architecture where the task of 3D reconstruction for a scene is carried out by implementing the algorithm on GP-GPU (CUDA). It includes several related experiments and a set of performance analysis.

Chapter 5 discusses about topics related to the sensor configurations and their geometry which includes the problem of camera coverage in the network (in the context of the proposed framework) and estimation of extrinsic parameters among cameras and laser range finder. The last discussed issue in this chapter is to consider the scene's dynamic by applying Bayesian techniques.

The overall conclusion, discussions and future works are presented in chapter 6.

After this chapter, the used convention for mathematical notations is presented in appendix A. The next appendix (Appen. B) is provided to extend the issue of multi-sensor calibration for a case where a camera network with no overlap in the field of view can be calibrated jointly with a laser range finder.

Chapter 2

3D Data registration using inertial planes

2.1 Introduction

This chapter presents a method for volumetric 3D reconstruction of an object inside a scene using inertial planes. In order to observe the scene, a sensor network is employed. Each node in the network is comprised of a couple of Inertial Sensor (IS) and camera. In each couple, the IS is used to define a virtual camera whose plane is horizontal and its axes are aligned to the earth cardinal directions. Moreover, a set of inertial-planes, which are parallel to each other and horizontal, is defined in the scene for the purpose of 3D data registration. The image planes of virtual cameras are projected onto these inertial-planes using a method based on the concept of homography. After describing the method, at the end a set of experiments will be presented to demonstrate the practicability and effectiveness of the proposed approach.

2.2 Three dimensional data registration using inertial planes

This section introduces and explains a framework to map and register the 3D data of a scene on to a set of inertial planes. In the next sub-sections we discuss the details of the framework staring by introducing the used camera model in Sec. 2.2.2, following by the basic concept behind homography transformation in sub-Sec. 2.2.3. In sub-Sec. 2.2.4 the problem is stated by showing the schema of a network of cameras and inertial sensor, where the involved reference frames are defined. A macro-view of the approach to map a 3D point from the scene onto an inertial plane is introduced in this section as well, where the details are provided in the sub-Sec. 2.2.5 and sub-Sec. 2.2.6. Eventually the volumetric 3D reconstruction in sub-Sec. 2.2.7 closes this section by providing an algorithm to perform the 3D reconstruction.

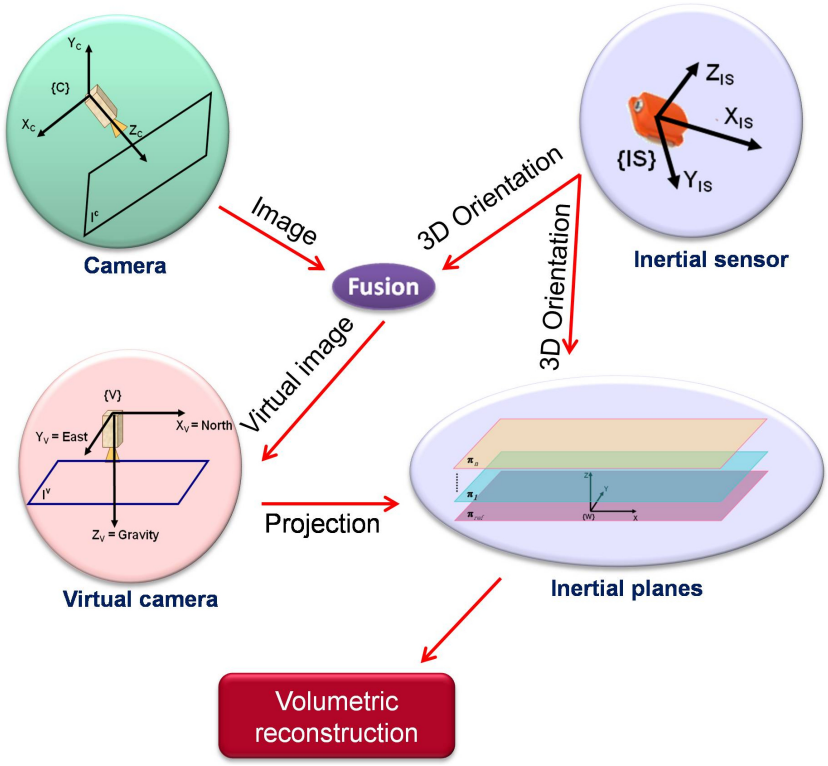


Figure 2.1: Overall scheme of the proposed 3D volumetric reconstruction: 3D orientation from IS and image from camera are fused (using the concept of infinite homography) to define a downward-looking virtual camera whose axes are aligned to the earth cardinal direction (North-East-Down). 3D orientation from IS is as well as used to define a set of inertial-planes (Euclidean) in the scene. The 3D reconstruction can be obtained by projecting the projective virtual images onto this set of parallel inertial planes.

2.2.1 Overall 3D reconstruction scheme

An overall scheme of the proposed volumetric reconstruction approach is depicted in Fig. 2.1. Two types of sensors are used: camera, for image grabbing and IS, for obtaining 3D orientation. Each camera is rigidly coupled to an IS. The outputs of each couple are fused using the concept of infinite homography and leads to have a downward-looking virtual camera whose axes are aligned to the earth cardinal direction (North-East-Down). Moreover, the 3D orientation of IS is used to define a set of inertial planes that are all virtual and parallel. The projective image

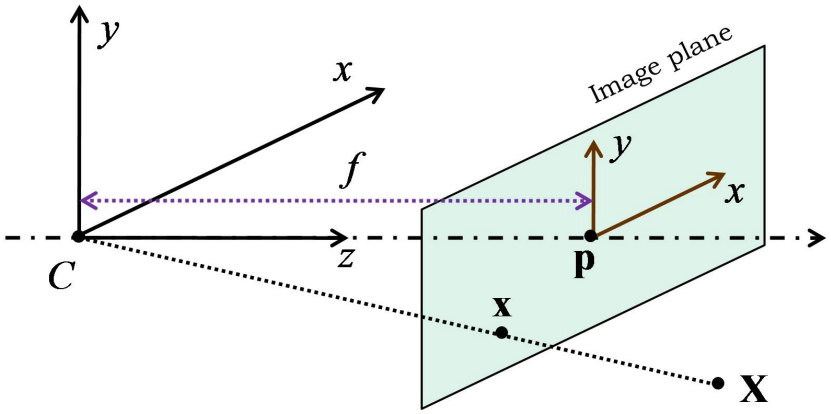


Figure 2.2: Pinhole camera model.

planes of virtual cameras are projected onto this set of inertial-planes (Euclidean) and the 3D volumetric reconstruction of the person (or generally an object) is obtained.

2.2.2 Camera model

Regarding the camera model, we use the pinhole camera model [HZ03]. In the pinhole camera model (Fig. 2.2), a homogeneous 3D point $\mathbf{X} = [\ X \quad Y \quad Z \quad 1\ ]^T$ in the scene and its corresponding projection $\mathbf{x} = [\ x \quad y \quad 1\ ]^T$ on the image plane are related via a 3×4 matrix A, called camera projection matrix, through the following equations (assuming the camera's coordinate frame as the world's coordinate frame):

$$\mathbf{x} = A\mathbf{X} \tag{2.1}$$

A defined as

$$A = K\left[I_{3\times 3}|\mathbf{0}_{3\times 1}\right] \tag{2.2}$$

where K is the camera calibration matrix [HZ03]. The camera matrix K, which is also referred as intrinsic parameter matrix, is defined by:

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

in which f represents the camera's focal length. u_0 and v_0 are the elements of the principal point P. In a pinhole camera model it is assumed that the image coordinates are Euclidean coordinates whose scales in both axes are equal. But normally in CCD cameras this assumption might not be satisfied or in other words the pixels can be non-square [HZ03]. In this case one should consider different focal lengths for x and y directions. Assuming the number of pixels per unit distance in the coordinates of image are respectively m_x and m_y for x and y directions, then $f_x = fm_x$ and $f_y = fm_y$ respectively denote the camera's focal lengths in the scale of pixels for the x and y directions [HZ03]. Based on this the camera matrix of Eq. (2.3) will be updated as following:

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

As mentioned, the camera projection matrix in Eq. (2.3) was for a case where the coordinate frame of camera is assumed as the world coordinate frame but for a general case we should consider a rotation R and translation $\mathbf{t}$ among the two coordinate frames (Fig. 2.3). Considering this, the general camera projection matrix A is expressed [HZ03] as

$$A = K\,[R|\,\mathbf{t}] \tag{2.5}$$

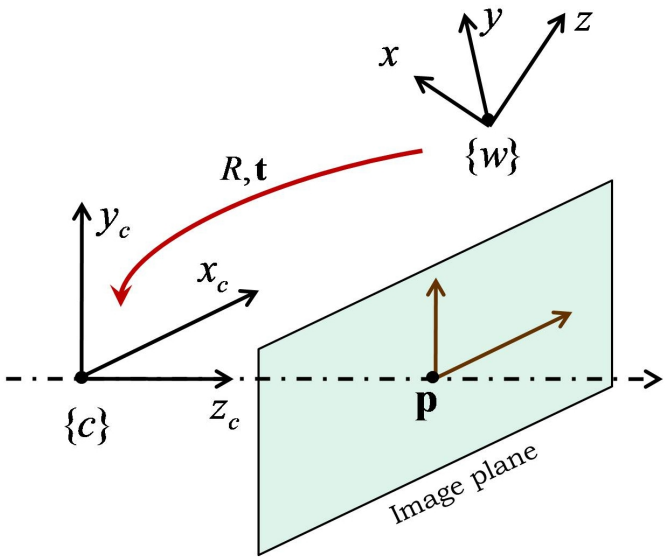


Figure 2.3: Transformation among the coordinate frames of the camera and world.

2.2.3 Multi-view geometry: homography

In order to map points from one plane to another plane (with preserving the collinearity) the concept of homography [HZ03, YMS04] is used. As illustrated in Fig. 2.4, suppose a 3D plane π is observed by two cameras with $A = K[I|0]$ and $A^{'} = K^{'}[R|\mathbf{t}]$ (concerning first camera center as world reference frame) where K and $K^{'}$ are the calibration matrices of the cameras. Also assume that $\mathbf{x}_1$ and $\mathbf{x}_2$ are the imaged points of a 3D point $\mathbf{x}$ lying on the plane π. Then $\mathbf{x}_1$ and $\mathbf{x}_2$ are called a pair of corresponding points and the relation between them can be expressed as $\mathbf{x}_2 = H\mathbf{x}_1$ in which H is a 3×3 transformation matrix called planar homography induced by the plane π [YMS04] and is equal to (up to scale)

$$H = K'(R + \frac{1}{d}\mathbf{t}\mathbf{n}^T)K^{-1} \tag{2.6}$$

where R and $\mathbf{t}$ are respectively rotation matrix and translation vector between the two cameras centres, $\mathbf{n}$ is normal of the 3D plane and d is the orthogonal distance between the 3D plane and the camera center. It it worth to mention that for a

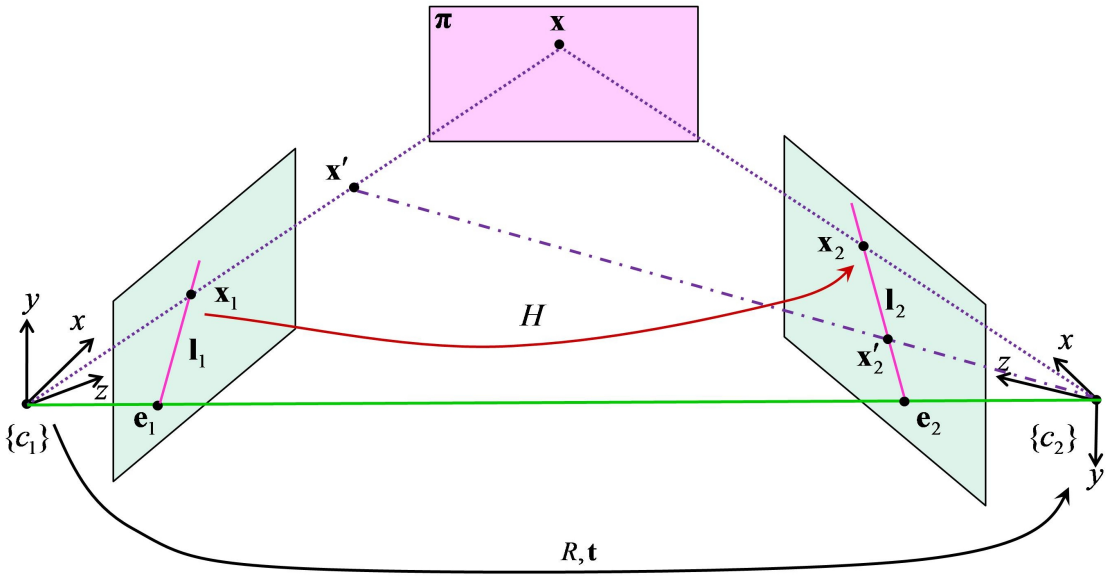


Figure 2.4: Homography among two image planes induced by a plane. π is a plane in the scene. $\mathbf{x}$ is a 3D point on π observed by two cameras C_1 and C_2. $\mathbf{x_1}$ and $\mathbf{x_2}$ are the images of $\mathbf{x}$ on the image planes of the first and second cameras, respectively. H is a 3×3 transformation matrix, called homography matrix, which is able to map $\mathbf{x_1}$ to $\mathbf{x_2}$. Generally such a homography matrix maps all points from the first image plane to the second one where the image points are induced by a plane like π.

point like $\mathbf{x}$ lying on the plane π, its imaged point $\mathbf{x}_2$ on the second camera is uniquely obtained via $\mathbf{x}_1$ as $\mathbf{x}_2 = H\,\mathbf{x}_1$. However for a point like $\mathbf{x}'$ not lying on the plane π (or off-the-plane), then using the homography matrix H as $\mathbf{x}_2 = H\,\mathbf{x}_1$ yields to have the point $\mathbf{x}'_2$ which is just a point lying on $\mathbf{l}_2$ ($\mathbf{l}_2$ being the epipolar line passing through the actual corresponding point $\mathbf{x}'_2$ and the epipole $\mathbf{e}_2$). Whereas the actual imaged point corresponding to $\mathbf{x}'$ on the second camera is $\mathbf{x}'_2$.

2.2.4 Network of cameras and inertial sensors

Fig. 2.5 shows a sensor network setup with a number of cameras. π_{ref} is an Euclidean inertial plane *, defined by the 3D orientation of IS, and is common

*It might appear just as π in the equations

for all cameras. Here $\{W\}$ is the world reference frame (a detailed specification of this reference frame shall be introduced in Sec. 2.2.5). In this setup, as mentioned before, each camera is rigidly coupled with an IS. The intention is to register a 3D point $\mathbf{X}$, observed by camera C, onto the reference plane π_{ref} as ${}^{\pi}\mathbf{x}$ (2D), by the concept of homography and using inertial data. A virtual image plane is considered for each camera. Such a virtual image plane is defined (using inertial data) as a horizontal image plane at a distance f below the camera center, f being the focal length[MDdA07]. In other words, it can be thought that beside of each real camera C in the setup, a virtual camera V exists whose center, $\{V\}$, coincides to the center of the real camera $\{C\}$ (see Fig. 2.8). The transformation matrix among these two reference frames is

$$ {}^{v}T_{c} = \begin{bmatrix} {}^{v}R_{c} & {}^{v}\mathbf{t}_{c} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.7} $$

where ${}^{v}R_{c}$ is the rotation matrix and ${}^{v}\mathbf{t}_{c}$ is equal to $\mathbf{0}_{3\times 1}$.

In order to register a 3D point $\mathbf{X}$ onto the π_{ref} as ${}^{\pi}\mathbf{x}$, three steps can be taken:

- First, the 3D point $\mathbf{X}$ is projected on the camera image plane by ${}^{\mathbf{c}}\mathbf{x} = A\mathbf{X}$ (A is the projection matrix of the camera C).
- Second, ${}^{c}\mathbf{x}$ (the imaged point on the camera image plane) is projected to its corresponding point on the virtual camera's image plane as ${}^{v}\mathbf{x}$. Indeed this operation is a homography transformation among two projective planes and can be expressed as ${}^{v}\mathbf{x} = {}^{v}H_{c}\,{}^{c}\mathbf{x}$, ${}^{v}H_{c}$ being a 3×3 homography matrix[HZ03].
- Third, the projected point on the virtual image plane, ${}^{v}\mathbf{x}$, is reprojected to the world virtual plane, π_{ref}. This operation is among a projective plane and an Euclidean plane and again can be expressed as ${}^{\pi}\mathbf{x} = {}^{\pi}H_{v}\,{}^{v}\mathbf{x}$, where ${}^{\pi}H_{v}$ is a 3×3 homography matrix[HZ03].

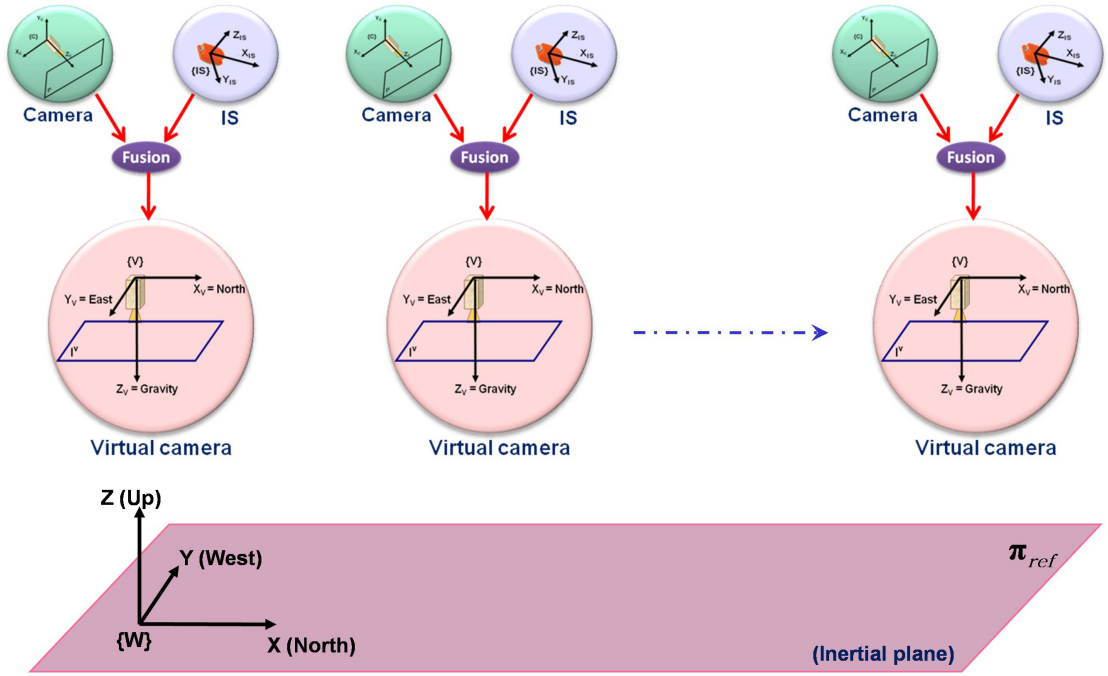


Figure 2.5: A network of sensors observes a scene. The sensor network is comprised of a quantity of IS-camera couples. The inertial and visual information in each couple are fused using the concept of infinite homography which leads to define a virtual camera. π_{ref} is a virtual reference plane (Euclidean) which is defined by using 3D orientation of IS and is common for all virtual cameras.

The first step is done considering the pinhole camera model (previously introduced). The second and third steps are described in the following two sub-sections. Assuming to already have ${}^{v}H_{c}$ and ${}^{\pi}H_{v}$, the final equation for registering a 3D point $\mathbf{X}$ onto the reference plane π_{ref} will be (see Fig. 2.9):

$$\pi_{ref}\mathbf{x} = {}^{\pi}H_{v}\,{}^{v}H_{c}\,A\,\mathbf{X} \tag{2.8}$$

The way of obtaining ${}^{v}H_{c}$ (homography matrix between the real camera image plane and virtual camera image plane) and ${}^{\pi}H_{v}$ (homography matrix between the virtual camera image plane and the world 3D plane π_{ref}) is discussed in the next sub-sections by starting to describe the conventional coordinate systems.

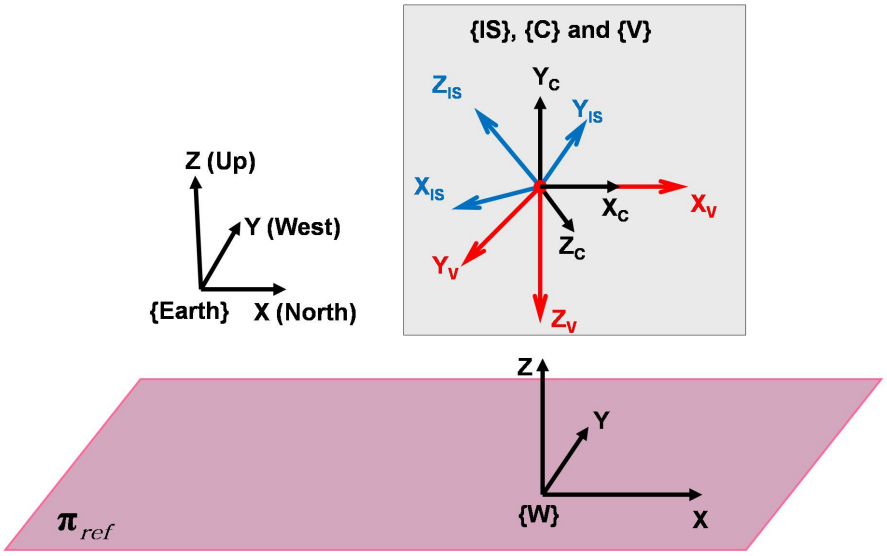


Figure 2.6: Involved coordinate references. $\{C\}$: The local coordinate system of a camera C. $\{E\}$: Earth fixed reference frame having its X axis in the direction of North, Y in the direction of West and Z upward. $\{IS\}$: Local reference frame of the IS sensor which is defined w.r.t. to the earth reference frame $\{E\}$. $\{V\}$: indicates the reference frame of the virtual camera corresponding to C. The centers of $\{C\}$ and $\{V\}$ are coincident.

2.2.5 Image plane of virtual camera

The definition of virtual camera is introduced in this sub-section. We start by presenting the coordinate systems. As seen in Fig. 2.6 and Fig. 2.7, there are four coordinate systems involved in this approach to be explained here:

- Real camera reference frame$\{C\}$: The local coordinate system of a camera C is expressed as $\{C\}$.
- Earth reference frame $\{E\}$: Which is an earth fixed reference frame having its X axes in the direction of North, Y in the direction of West and Z upward.
- Inertial sensor reference frame $\{IS\}$: This is the local reference frame of IS sensor which is defined w.r.t. to the earth reference frame $\{E\}$.

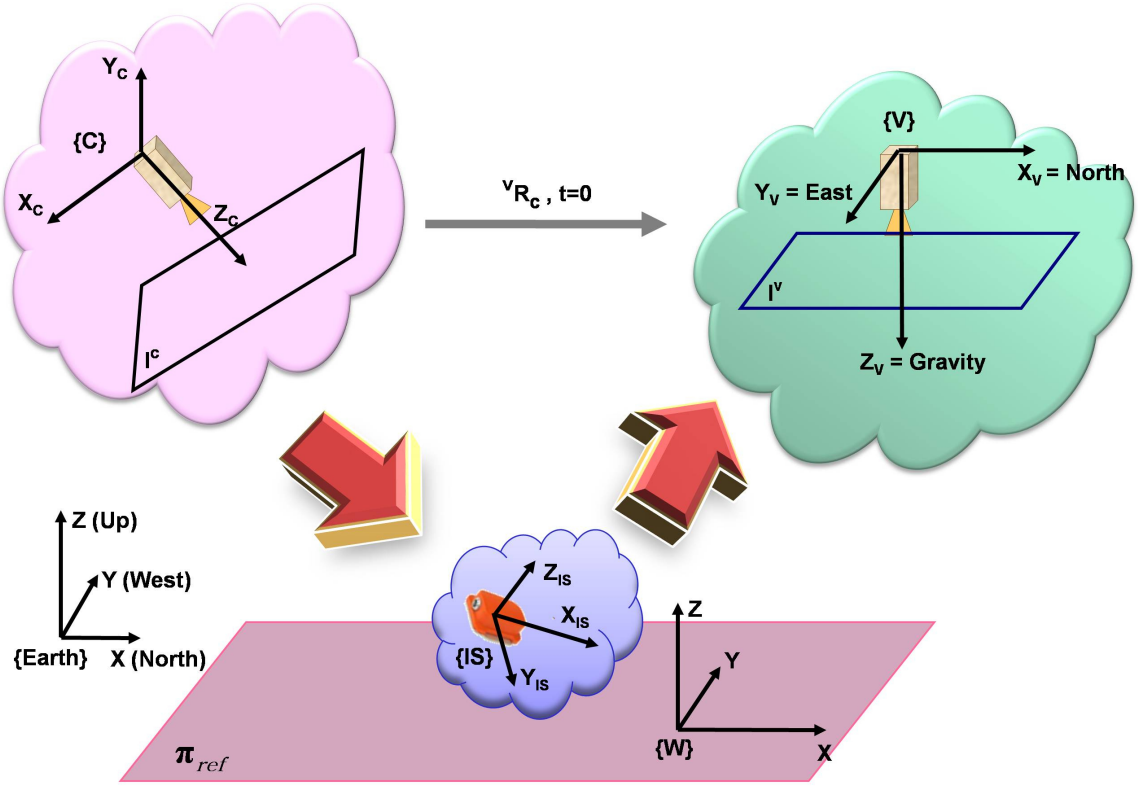


Figure 2.7: Graphical view of virtual camera definition. A virtual camera, whose image plane is horizontal and its axes are aligned to the earth cardinal direction, is defined through using 3D orientation provided by IS. The transformation among $\{C\}$ and $\{V\}$ has a rotation ${}^{v}R_c$ and a translation equal to $\mathbf{0}_{3\times1}$.

- Virtual camera reference frame $\{V\}$: As explained, for each real camera C, a virtual camera V, is considered by the aid of a rigidly coupled IS to that. $\{V\}$ indicates the reference frame of such a virtual camera. The centers of $\{C\}$ and $\{V\}$ coincide and therefore there is just a rotation between these two references.

The idea is to use the 3D orientation provided by IS to register image data on the Euclidean reference plane π_{ref} defined in $\{W\}$ (the world reference frame of this approach). The reference 3D plane π_{ref} is defined such a way that it spans the X and Y axes of $\{W\}$ and it has a normal parallel to the Z (See Fig. 2.6). In this proposed method the idea is to not using any real 3D plane inside the scene for

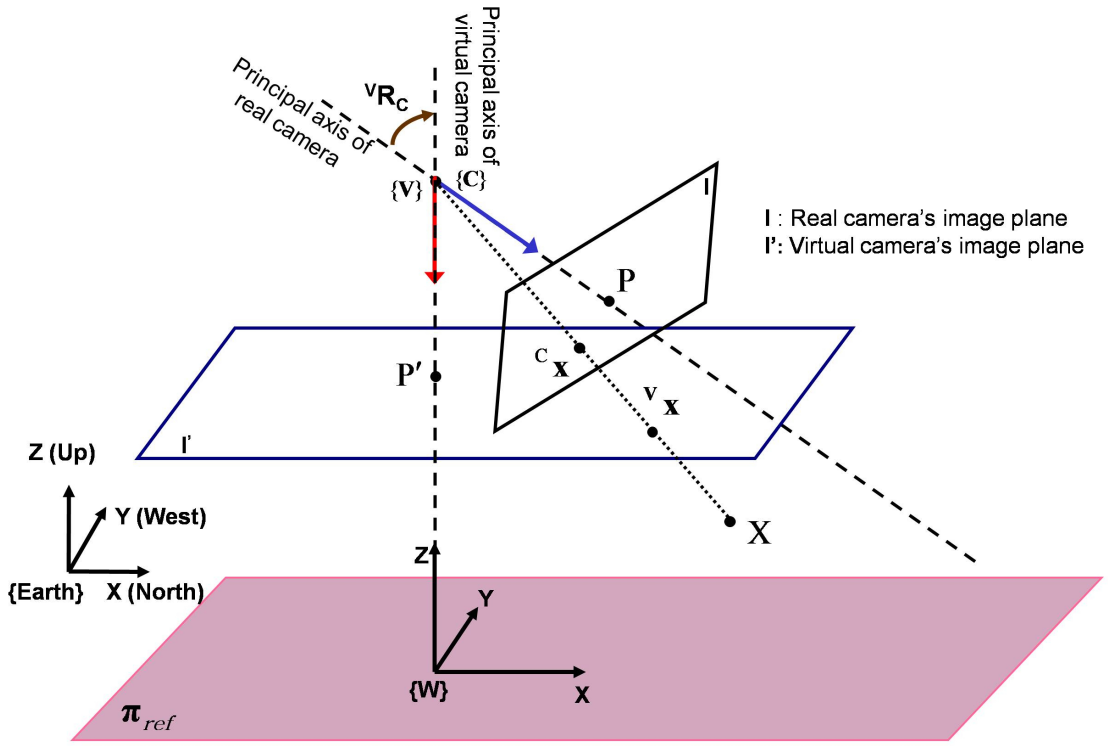


Figure 2.8: Geometrical view of a virtual camera: The concept of infinite homography is used to fuse inertial-visual information and define an earth cardinal aligned virtual camera. Moreover using the inertial information, π_{ref} is defined as a virtual world plane which is horizontal and parallel to the image plane of virtual camera.

estimating homography. Hence we assume there is no a real 3D plane available in the scene so that our $\{W\}$ becomes a virtual reference frame and consequently π_{ref} is a horizontal virtual plane on the fly. Although $\{W\}$ is a virtual reference frame however it needs to be formally specified and fixed in the 3D space. Therefore here we start to define $\{W\}$ and as a result π_{ref}. With no loss of generality we place $\mathbf{O}_W$, the center of $\{W\}$, in the 3D space such a way that $\mathbf{O}_W$ has a height d w.r.t the first virtual camera, V_0. Again with no loss of generality we specify its orientation the same as $\{E\}$ (earth fixed reference). Then as a result we can describe the reference frame of a virtual camera $\{V\}$ w.r.t $\{W\}$ via the following homogeneous transformation matrix

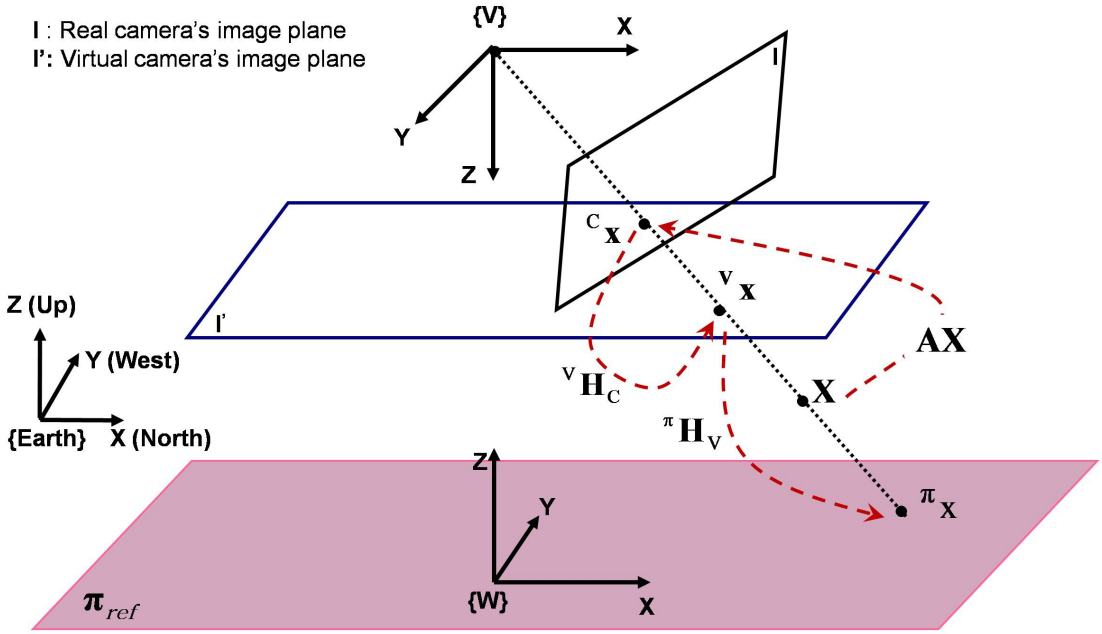


Figure 2.9: One projection and two consecutive homographies are needed to register a 3D point $\mathbf{X}$ from the scene on an Euclidean virtual plane π_{ref} using IS. ${}^{v}H_{c}$: Homography from real camera image plane to the virtual one, ${}^{\pi}H_{v}$: Homography from the image plane of virtual camera to the reference inertial-plane π_{ref}.

$$ {}^{W}T_{V} = \begin{bmatrix} {}^{W}R_{V} & \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.9} $$

where ${}^{w}R_{v}$ is a rotation matrix defined as (see Fig. 2.6):

$$ {}^{W}R_{V} = \begin{bmatrix} \hat{\mathbf{i}} & -\hat{\mathbf{j}} & -\hat{\mathbf{k}} \end{bmatrix} \tag{2.10} $$

$\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ being the unit vectors of the X, Y and Z axes, respectively. Also $\mathbf{t}$ is a translation vector between the centres of $\{V\}$ and $\{W\}$. Obviously using the preceding definitions and conventions, for the first virtual camera we have $\mathbf{t} = [\ 0 \quad 0 \quad d\ ]^{T}$.

We continue the discussion to obtain a 3×3 homography matrix ${}^{v}H_{c}$ which transforms a point ${}^{c}\mathbf{x}$ on the real camera image plane I to the point ${}^{v}\mathbf{x}$ on the virtual camera image plane $I^{'}$ as ${}^{v}\mathbf{x} = {}^{v}H_{c}\,{}^{c}\mathbf{x}$ (see Fig. 2.8). As described, the real camera C and virtual camera V have their centers coincided to each other, so the transformation between these two cameras can be expressed just by a rotation matrix. In this case ${}^{v}H_{c}$ is called infinite homography since there is just a pure rotation between real camera and virtual camera centers [HZ03, Mir09]. Such an infinite homography can be obtained using a limiting process on Eq. (2.6) by considering $d \to \infty$ (as described in [YMS04, HZ03, MD07]):

$$ {}^{V}H_{C} = \lim_{d\to\infty} K({}^{V}R_{C} + \frac{1}{d}\mathbf{t}\mathbf{n}^{T})K^{-1} = K\,{}^{V}R_{C}\,K^{-1} \tag{2.11} $$

where K is the camera matrix and ${}^{v}R_{c}$ is the rotation matrix between $\{C\}$ and $\{V\}$. ${}^{v}R_{c}$ can be obtained through three consecutive rotations which is mentioned in Eq. (2.12) (see the reference frames in Fig. 2.6) as following:

$$ {}^{V}R_{C} = {}^{V}R_{E}\;{}^{E}R_{IS}\;{}^{IS}R_{C} \tag{2.12} $$

The first one is to transform from real camera reference $\{C\}$ to the IS local coordinate $\{IS\}$, the second one transforms from the $\{IS\}$ to the earth fixed reference $\{E\}$ and the last one is to transform from $\{E\}$ to virtual camera reference frame $\{V\}$:

${}^{IS}R_{C}$ can be obtained through an IS-camera calibration procedure. We use Camera Inertial Calibration Toolbox[LD07] is used in order to calibrate a rigid couple of a IS and camera. Rotation from IS to earth, or ${}^{E}R_{IS}$, is given by the IS sensor w.r.t $\{E\}$. Since the $\{E\}$ has the Z upward but the virtual camera is defined to be downward-looking (with a downward Z) then the following rotation is applied to reach to the virtual camera reference frame:

$$ {}^{V}R_{E} = \left[\begin{array}{ccc} \hat{\mathbf{i}} & -\hat{\mathbf{j}} & -\hat{\mathbf{k}} \end{array} \right] \tag{2.13} $$

2.2.6 Projection of 3D data onto a world inertial plane

In this section we describe a method to obtain homography matrix ${}^{\pi}H_{v}$ that transforms points from a projective virtual image plane I' (the image of virtual camera V) to an Euclidean inertial plane π_{ref} (recalling that these two planes are defined to be parallel. See Fig. 2.9). A 3D point $\mathbf{X}$ on π_{ref} is expressed in $\{W\}$ as $\mathbf{X} = [\ X \quad Y \quad 0 \quad 1\ ]^{T}$ in its homogeneous form (recalling that XY-plane of $\{W\}$ corresponds to π_{ref} and therefore any points on this plane has $Z = 0$). For a general case (pinhole camera), $\mathbf{X}$ is projected on the image plane as following:

$$ \mathbf{x} = K \left[\begin{array}{cccc} \mathbf{r}1 & \mathbf{r}2 & \mathbf{r}3 & \mathbf{t} \end{array} \right] \left[\begin{array}{c} X \\ Y \\ 0 \\ 1 \end{array} \right] = K \left[\begin{array}{ccc} \mathbf{r}1 & \mathbf{r}2 & \mathbf{t} \end{array} \right] \left[\begin{array}{c} X \\ Y \\ 1 \end{array} \right] \tag{2.14} $$

where $\mathbf{r}1$, $\mathbf{r}2$ and $\mathbf{r}3$ are the columns of the 3×3 rotation matrix, K is the camera calibration matrix (defined in Eq. 2.4) and $\mathbf{t}$ is the translation vector between π_{ref} and camera center [HZ03]. As can be seen Eq. (2.14) indicates a plane to plane projective transformation and therefore can be expressed like a planar homography:

$$ \mathbf{x} = {}^{v}H_{\pi}\, {}^{\pi}\mathbf{x} \tag{2.15} $$

where

$$ {}^{v}H_{\pi} = K \left[\begin{array}{ccc} \mathbf{r}1 & \mathbf{r}2 & \mathbf{t} \end{array} \right] \tag{2.16} $$

, ${}^{\pi}H_v$ denoting a 3×3 homography matrix and ${}^{\pi}\mathbf{x} = [\ X \quad Y \quad 1\ ]^T$. We recall that for each camera within the network a virtual camera is defined (using inertial data). All such virtual cameras have the same rotation w.r.t world reference frame $\{W\}$. In other words one can think that there is no rotation among the virtual cameras. ${}^{W}R_V$ or the rotation matrix between a virtual camera and $\{W\}$ was described through Eq. (2.10). Then considering ${}^{W}R_V$ from Eq. (2.10), π_{ref} as the interesting world plane and $\mathbf{t} = [\ t_1 \quad t_2 \quad t_3\ ]^T$ as the translation vector and eventually K as camera calibration matrix (K is defined in Eq. 2.4), Eq. (2.16) can be formalized as :

$$ {}^{\pi}H_v^{-1} = K[\ \hat{\mathbf{i}} \quad -\hat{\mathbf{j}} \quad \mathbf{t}\ ] = \begin{bmatrix} f_x & 0 & f_x t_1 + u_0 t_3 \\ 0 & -f_y & f_y t_2 + v_0 t_3 \\ 0 & 0 & t_3 \end{bmatrix} \tag{2.17} $$

In the same way the homography matrices for other inertial planes parallel to π_{ref} can be obtained by using appropriate value for t_3 (the z element of $\mathbf{t}$) in Eq.(2.17). Fig. 2.10 shows a case where a 3D point $\mathbf{X}$ is registered on different Euclidean inertial-planes by using homography transformations.

2.2.7 Volumetric reconstruction

The geometric models for projecting 3D data onto a set of virtual horizontal planes based on the concept of homography was previously introduced. Indeed here the homography transformation can be basically interpreted as shadow on each inertial-based virtual plane created by a light source located at the camera position. Considering several cameras (remembering light sources) which are observing the object then different shadows will appear on the inertial planes. Conceptually, the intersection between each one of these planes and the observed object can be obtained by using the intersections of all shadows. This interpretation is illustrated in the Fig. 2.11.

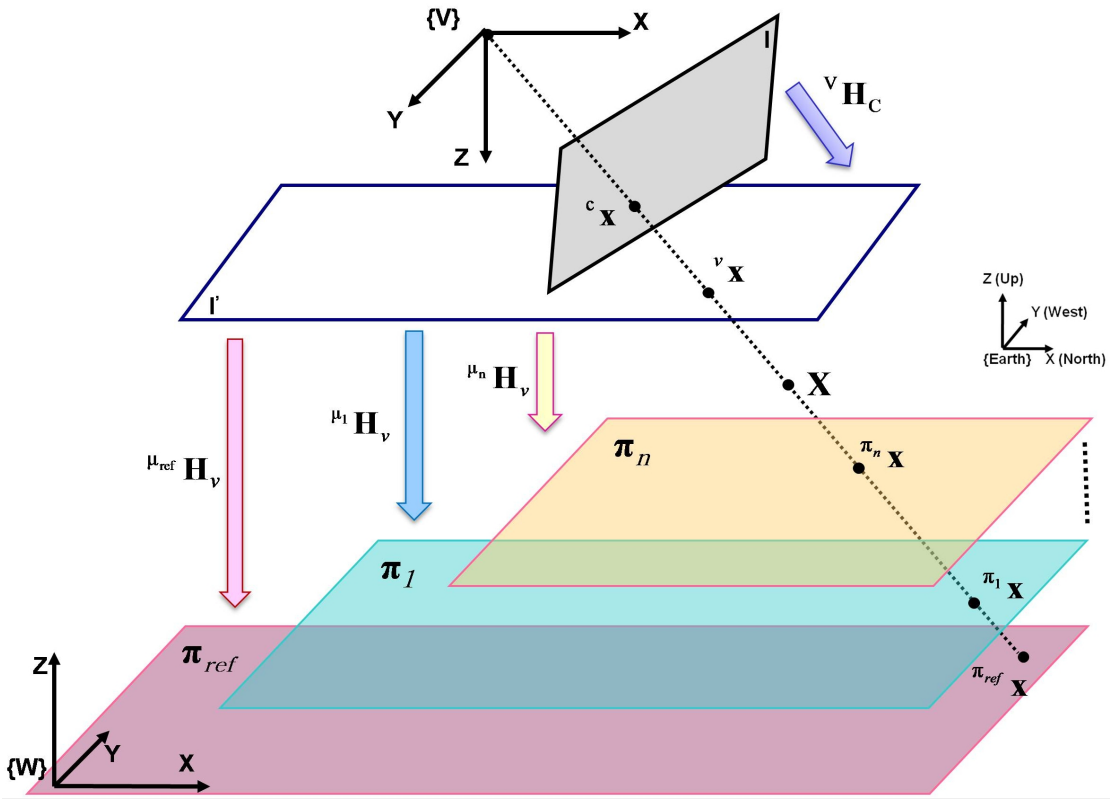
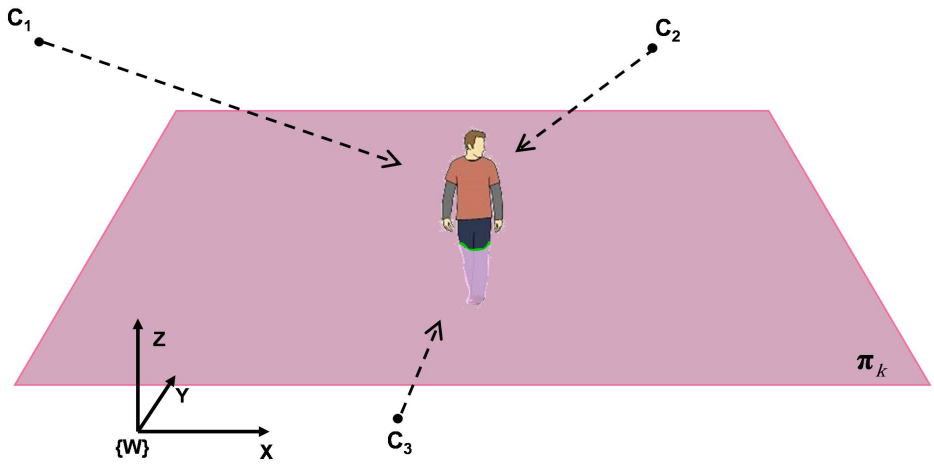
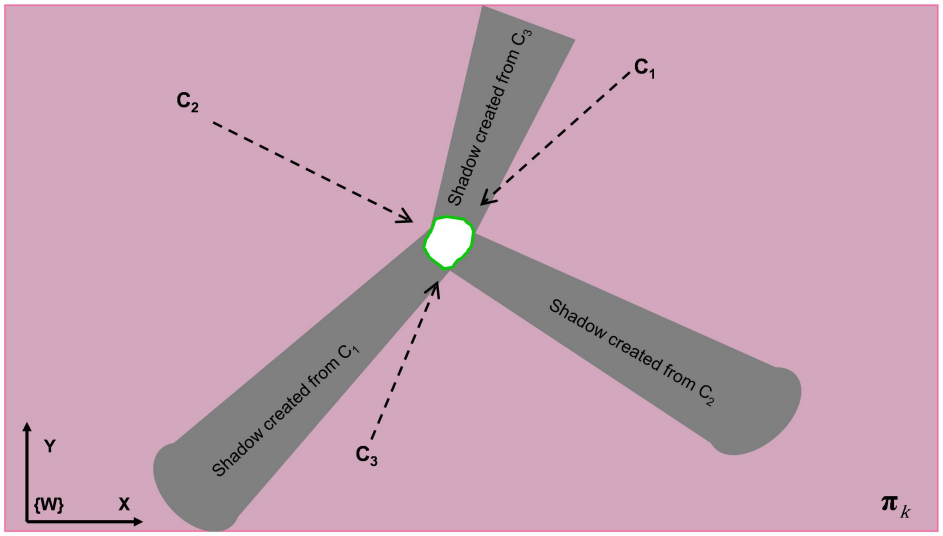


Figure 2.10: Using a set of inertial planes and homography for multi-layer 3D data registration: A 3D point $\mathbf{X}$ in the scene is registered on each one of the inertial planes using an appropriate homography matrix.

There is a geometrical explanation to support this interpretation. Fig. 2.12 demonstrates a person being observed by two virtual cameras V_1 and V_2. π' is an inertial plane which passes across the person. $\mathbf{X}$ and $\mathbf{Y}$ are two 3D points from the person surface. $\mathbf{X}$ lies on the plane π' and $\mathbf{Y}$ is off the plane. The 3D points $\mathbf{X}$ and $\mathbf{Y}$ are imaged as $\mathbf{x}_1$, $\mathbf{y}_1$, $\mathbf{x}_2$ and $\mathbf{y}_2$ on the image planes of V_1 and V_2, respectively (using the proposed homography methods). Suppose $^{\pi'}\mathbf{x}_1$, $^{\pi'}\mathbf{y}_1$, $^{\pi'}\mathbf{x}_2$ and $^{\pi'}\mathbf{y}_2$ are respectively the projections of the imaged points $\mathbf{x}_1$, $\mathbf{y}_1$, $\mathbf{x}_2$ and $\mathbf{y}_2$ onto π'. As seen in Fig. 2.12, for an on the plane point such as $\mathbf{X}$, all three points $\mathbf{X}$, $^{\pi'}\mathbf{x}_1$, $^{\pi'}\mathbf{x}_2$ are coincident and meet on π'. In contrary, for the point $\mathbf{Y}$ which is off the plane, the three points $\mathbf{Y}$, $^{\pi'}\mathbf{y}_1$, $^{\pi'}\mathbf{y}_2$ are distinct. $\mathbf{y}'_2$ denotes the image of $^{\pi'}\mathbf{y}_1$ on the image plane of V_2. The vector between $\mathbf{y}'_2$ and $\mathbf{y}_2$ is called parallax. Indeed



(a)



(b)

Figure 2.11: Illustration of the registration using homography concept. (a): A scene including a human is depicted. π_k is an inertial-based virtual world plane. Three cameras are observing the scene. (b): The registration layer (top view of the plane π_k of figure(a)). Each camera can be interpreted as a light source and the person causes to have a shadow for each camera. Intersection of all shadows on this Euclidean plane gives the cross section of the plane and the person.

Algorithm 1: 3D reconstruction algorithm using a set of inertial based horizontal planes.

for each c involved in $\{camera\}$ begin
  consider v as corresponding virtual camera for c
  obtain projection $I^{'c}$ from c to v
  obtain $\mathbf{t}$ for each v // translation vector
end
for $h = h_{min}$ to h_{max} step Δh begin
  for each v involved in $\{virtual\,camera\}$ begin
    obtain projection $I^{"v}$ from v to π^h
  end
  for each $i \in \{1..height(I^{"v})\}$ begin
    for each $j \in \{1..width(I^{"v})\}$ begin
      $n_c = card(\{virtual\,camera\})$ //cardinality
      $R(h,i,j) = \prod_{v=1}^{n_c} I^{"v}(i,j)$
    end
  end
end
return R // as volumetric 3D reconstruction of the object

the line through $\mathbf{y}'_2$ and $\mathbf{y}_2$ is the image of the ray passing through the center of V_1 and $\mathbf{Y}$ (which is also an epipolar line). For all points off the inertial plane π', the norm of their parallax is bigger than zero and for those points which are on π', there is no parallax (or in other words their parallax's norm is zero).

Based on this explanation, the proposed multi-layer 3D data registration method can be used to perform volumetric 3D scene reconstruction. Such a reconstruction approach is encapsulated and described as an algorithm in Alg. 1. Here $\{camera\}$ and $\{virtual\,camera\}$ are respectively the sets of all cameras and virtual cameras , I indicates the image plane of a real camera, I' indicates the image plane of a

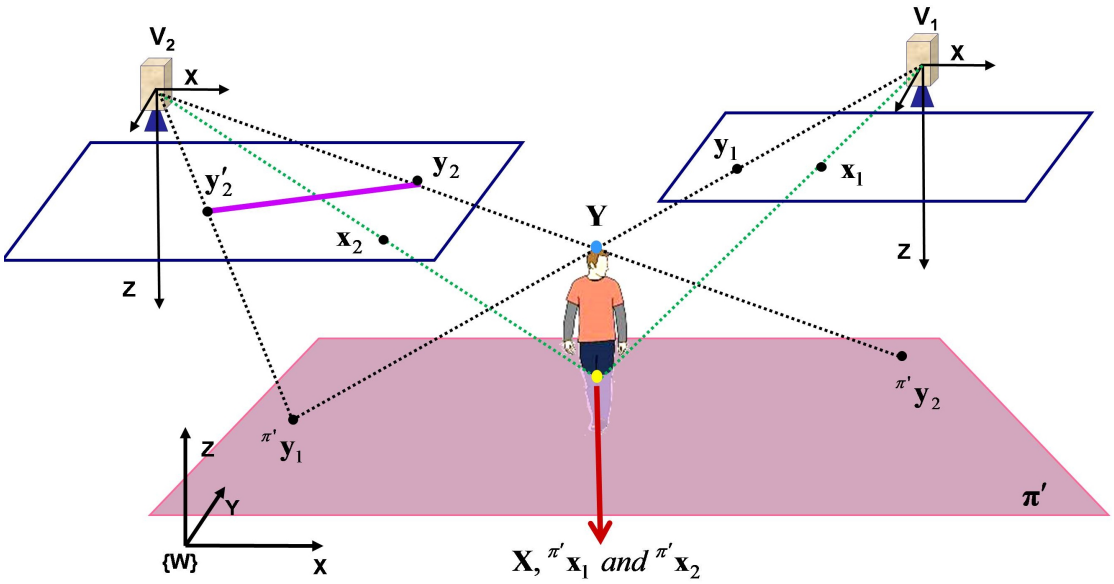


Figure 2.12: Geometric interpretation of the intersection among a person and an inertial plane π': $\mathbf{X}$ and $\mathbf{Y}$ are two exemplary 3D points belonging to the person's body, being observed by two virtual cameras V_1 and V_2. $\mathbf{X}$ lies on π' and $\mathbf{Y}$ is off the plane π'. The 3D points get projected on π'using the proposed homographic method. The homographic projections of the point which are on π' such as $\mathbf{X}$ are coincident ($^{\pi'}\mathbf{x}_1$, $^{\pi'}\mathbf{x}_2$) whereas for 3D points off the plane (such as $\mathbf{Y}$) their projections on π' are distinct ($^{\pi'}\mathbf{y}_1$ and $^{\pi'}\mathbf{y}_2$). In other words, for the points off π' there is a parallax (like the vector through $\mathbf{y}_2$ and $\mathbf{y}'_2$).

virtual camera and I'' indicates a virtual world plane. The algorithm returns a set of Euclidean 2D registration planes. 3D volumetric reconstruction of a human or object is obtained by stacking these virtual planes. Δh can be interpreted as the horizontal resolution for the algorithm.

2.3 Experiments

A set of experiments has been carried out using the proposed 3D reconstruction method. In these experiments, a portable IS-camera couple is placed in different positions and used for data acquisition. The obtained data in these two experi-

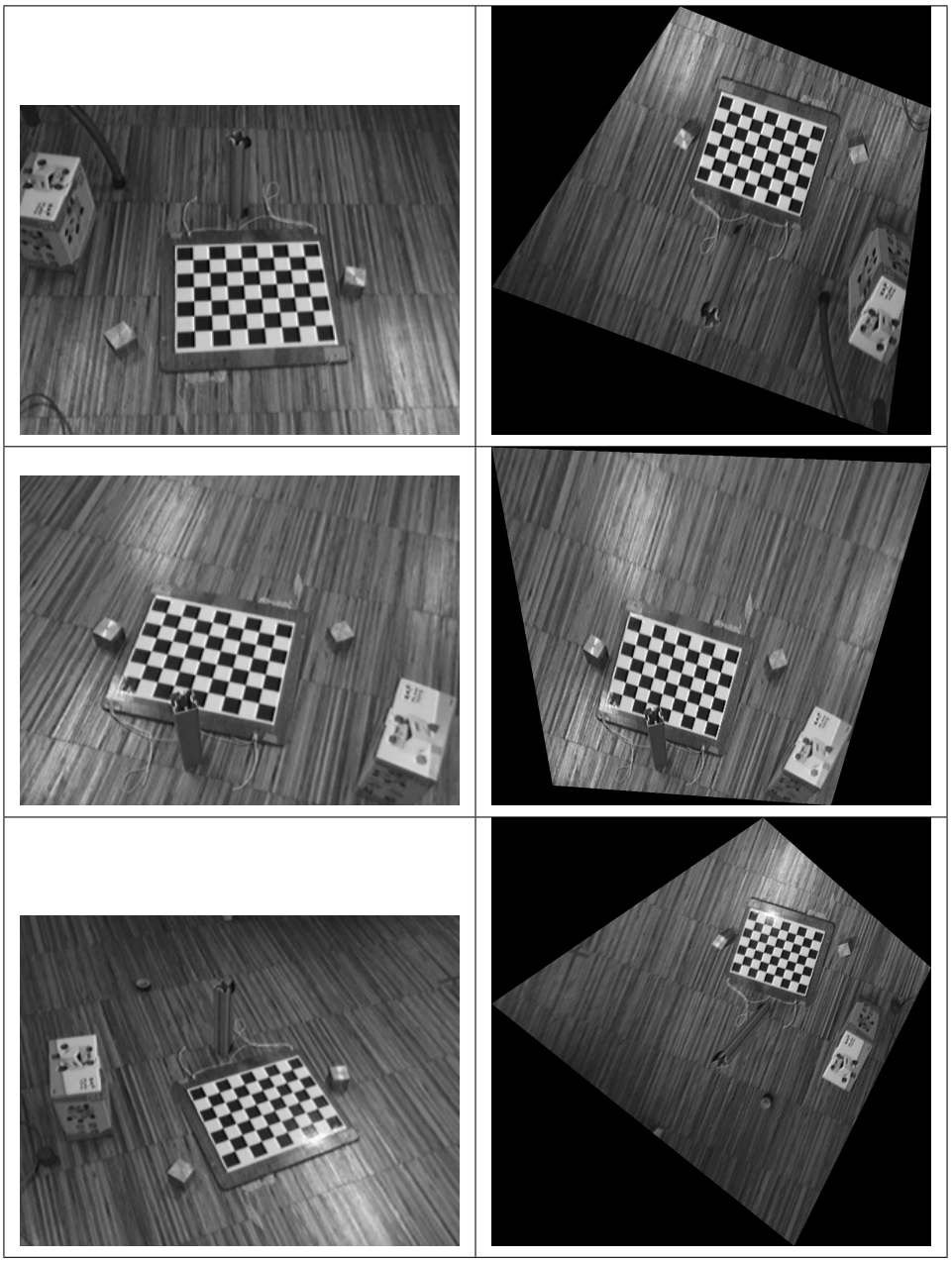


Figure 2.13: An example to demonstrate virtual images. Lefts: Real image planes (grabbed by a real camera within the setup). Rights: Obtained virtual image planes corresponding to the real images shown in the left. These images (right column) are obtained by applying an appropriate homography transformation, described in Sec. 2.2.5, on the original images (left column). As can be seen all three virtual images at the right column seem parallel to the floor (horizontal) and moreover there is no rotation among them.

Figure 2.14: A couple of IS-camera sensors used in the experiments

Figure 2.15: Left: Cat statue. Right: An snapshot of the scene.

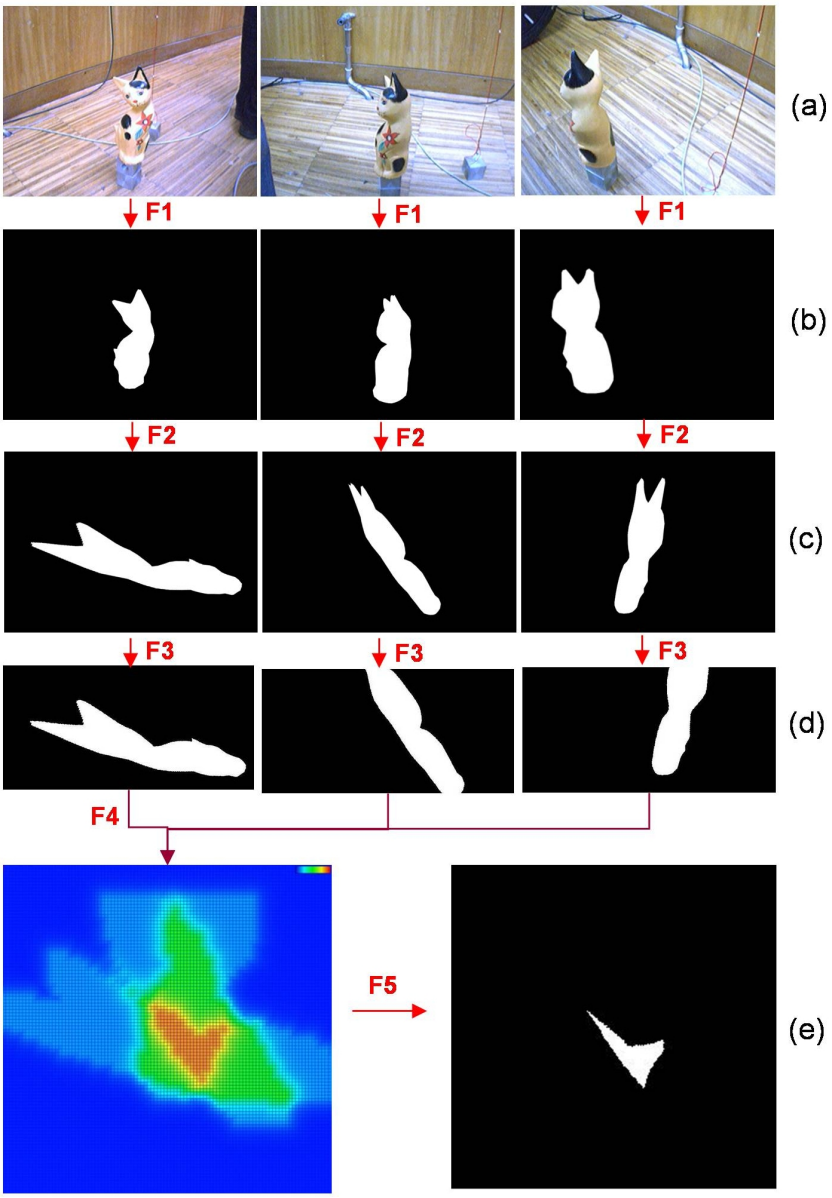


Figure 2.16: Steps to register the cross section of the cat statue on an Euclidean inertial plane. This plane is an exemplary inertial plane among totally 47 Euclidean planes used for 3D reconstruction of the statue. The height of this plane is 380 *mm* with respect to the first camera position. f_1: Extracts silhouettes. f_2: Reprojects black and white images to virtual camera image plane. f_3: Reprojects virtual image plane onto a 2D world virtual (horizontal) plane at a height=380 mm. f_4: Merging of three views (outcome of f_3) : The areas coloured in red indicate that there are overlaps between all three projections. f_5: Final virtual registration plane which is obtained by keeping just the intersections.

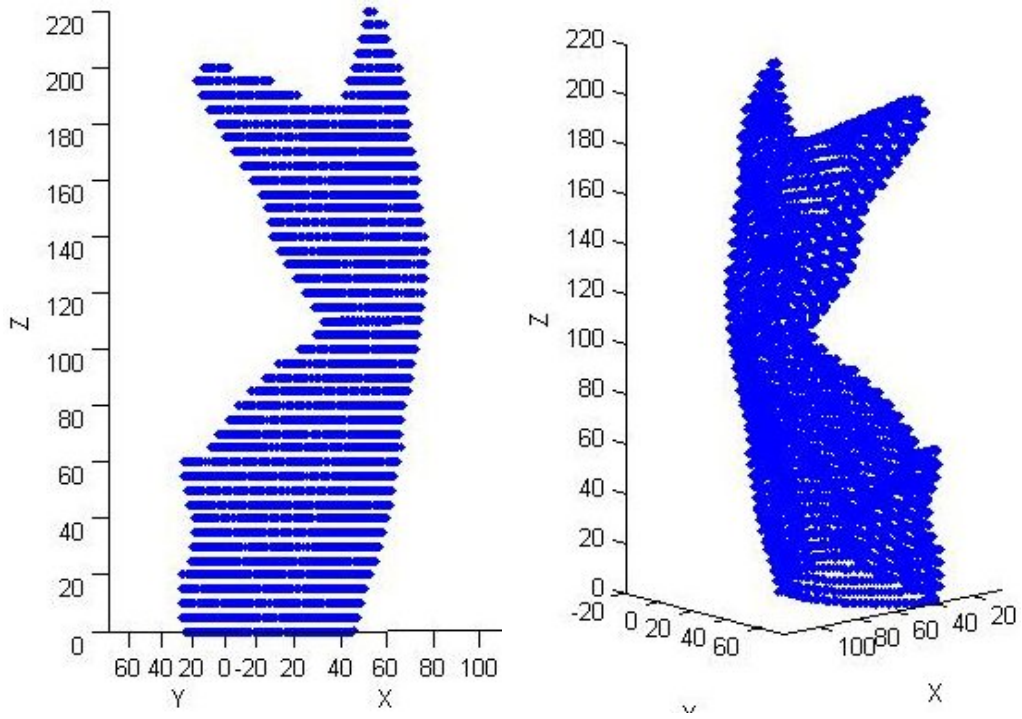


Figure 2.17: Results of 3D reconstruction of the cat statue in true scales. 47 inertial planes with an internal distance equal to 5*mm* are used to cross the object for the aim of reconstruction.

ments are used for 3D volumetric reconstruction of a cat statue and a mannequin. The implementations are performed in Matlab (off-line).

The IS-camera setup used in the experiments is demonstrated in Fig. 2.14. Fig. 2.15 shows a cat statue and an snapshot of the setup. The used camera is a simple FireWire Unibrain camera*. The 3D orientations are obtained using a MTi-Xsens[xse] (as IS). Firstly the intrinsic parameters of the camera is estimated using Bouguet Camera Calibration Toolbox[Bou03]:

$$K = \begin{bmatrix} 750.9819 & 0 & 367.5754 \\ 0 & 751.8286 & 292.6940 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.18}$$

*http://www.unibrain.com

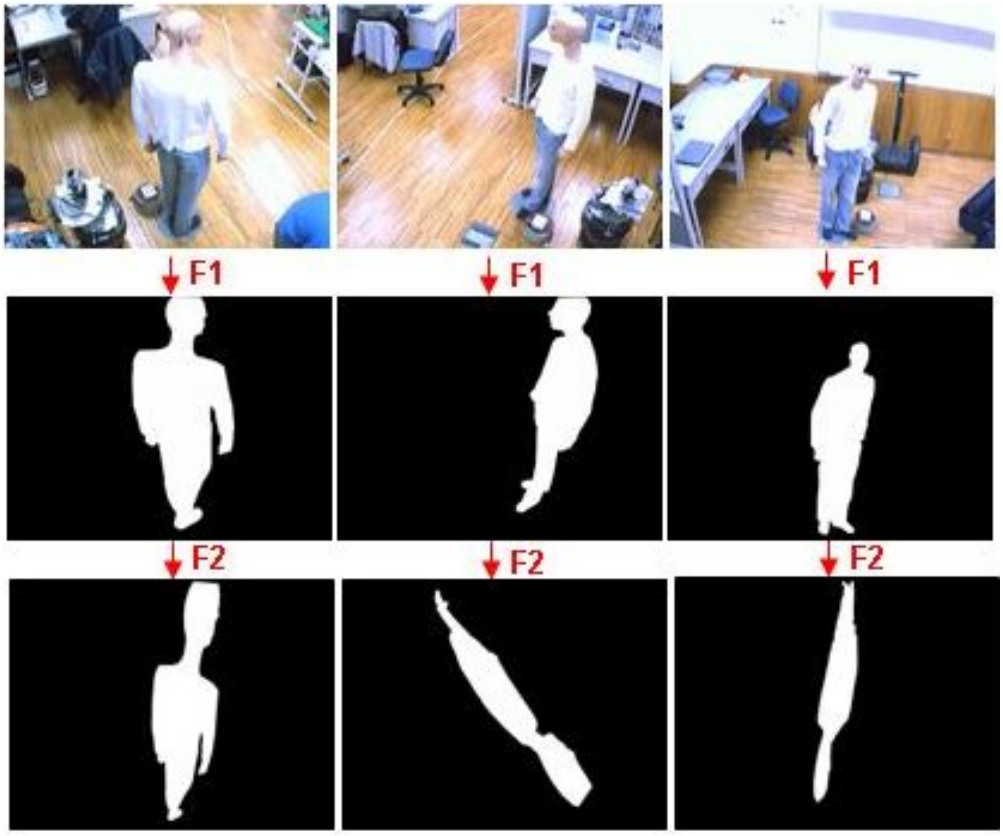


Figure 2.18: Experiment on human silhouette reconstruction using the proposed approach. F1: Background subtraction process. F2: Image planes of virtual cameras.

and then Camera Inertial Calibration Toolbox [LD07] is used for the sake of extrinsic calibration between the camera and IS (to estimate ${}^{IS}R_C$ in equation 2.12):

$$ {}^{C}R_{IS} = \begin{bmatrix} 0.0032 & -0.9996 & -0.0286 \\ 0.0179 & 0.0286 & -0.9994 \\ 0.9998 & 0.0027 & 0.0179 \end{bmatrix} \tag{2.19} $$

Fig. 2.13 demonstrates some examples of virtual images. Real image planes (grabbed by three a real camera within the setup) are shown at the left column. Images at the right column show the obtained virtual image planes corresponding to the real images shown in the left. These images (right column) are obtained

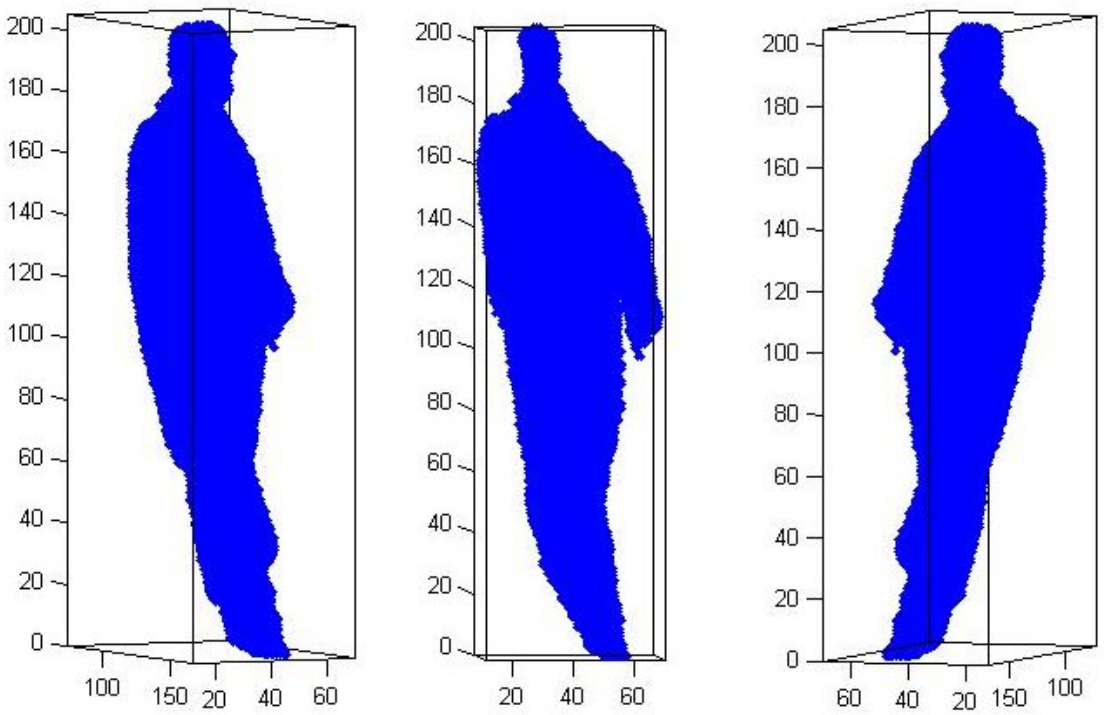


Figure 2.19: Volumetric 3D reconstruction of a mannequin using the proposed multiple virtual parallel planes approach.

by applying an appropriate homography transformation, described in Sec. 2.2.5, on the original images (left column). As can be seen all three virtual images at the right column seem parallel to the floor (horizontal) and moreover there is no rotation among them.

The couple of IS-camera is placed in different positions. In order to estimate the translation among two virtual cameras, we use an approach which is proposed and explained in chapter 3 (Sec. 3.3). This method needs to have the relative heights of two arbitrary 3D points in the scene with respect to one of the cameras within the network. To do so, a simple and thin string is hanged near to the object. Two points of the string are marked. Then the relative heights between these two marked points and the first camera (indeed here the IS-camera couple in the first position) are measured manually. The relative heights can also be

measured using some appropriate devices such as altimeters. Note that these two points are not needed to necessarily be on a vertical line, but since we did not have altimeter available, then we used two points from a vertically hanged string in order to minimize the measuring error. Afterwards, in each position a pair of imagery-inertial data is grabbed (3D orientation of the IS w.r.t earth cardinal direction). Fig. 2.16-a show three exemplary images taken from three different views. Firstly, the silhouettes are interactively extracted (see Fig. 2.16-b). After background subtraction, the corresponding virtual images are obtained based on the method described in 2.2.5. Fig. 2.16-c. shows the mentioned virtual image planes. Using the proposed 2-point-height method (shall be introduced in chapter 3) the translations between cameras in three position are estimated. By now we have the images from the views of virtual cameras (see Fig. 2.16-c). The next step is to consider a set of inertial-based parallel 3D planes in the world and then reproject the virtual camera images onto these horizontal virtual planes. Here 47 horizontal world planes are used. The height of lowest one is $480mm$ w.r.t first camera and the highest one is $250mm$. The interval distance between the inertial-based virtual planes is considered as $5mm$. As an example, Fig. 2.16-d indicates the reprojection of the three virtual camera images onto a virtual world plane at height=$380mm$. In Fig. 2.16-d, the cells with a red color indicate points where all projected virtual cameras have intersection. In other words, in these cells all three images have reported foreground observation. Cells with a color near to green indicate that there are just two foreground observations. Cells with a lighter color means that there is no observation by any of the cameras.

After obtaining the intersection of all projected silhouettes on each each inertial plane, such an intersection is indeed the registration of the cross section of the object and the inertial plane. The result is depicted in Fig. 2.16-e. After performing appropriate operations for all 47 virtual 3D planes and stacking them over together, the result becomes the 3D volumetric reconstruction of the object. Fig. 2.17 shows the result of the 3D reconstruction algorithm. This experiment is implemented in Matlab. It should be mentioned that for the sake of using less memory just the boundaries of the images are fed to the plot (using “bwperim”

Matlab function). Another experiment is also implemented on a mannequin, for the sake of human 3D reconstruction. Fig. 2.18 shows the image planes of real cameras and then image planes of the corresponding virtual cameras. The same algorithm (Alg. 1) is applied for this experiment. The result is shown in Fig. 2.19.

2.4 Conclusion

This chapter presented a method to perform volumetric 3D reconstruction of an object or human inside a scene using a network of cameras and inertial sensors. A set of experiments has been carried out for the proposed volumetric reconstruction algorithm where 3D reconstructions for a cat statue and a mannequin were demonstrated. The data acquisition was done by placing a couple of IS-camera in different places around the object while collecting inertial-image data for each place. Regarding the background subtraction method it should be mentioned that in this work having a good enough background subtraction is assumed. Depends to each application, a suitable background subtraction method should be tailored. In these experiments, the background subtraction was performed interactively.

Chapter 3

Parameter estimation and uncertainty modelling

3.1 Introduction

In this chapter we present three topics, parametric homography among different virtual planes, estimation of translation vectors among cameras and uncertainty modelling of the points mapped through homography transformations.

Geometric relations among the Euclidean virtual planes from the scene and the projective virtual image planes are more specifically explored for the purpose of 3D data registration. A set of mathematical equations are obtained which are capable of parametrically generate homography matrices to transform 2D points from one virtual plane to another within the registration framework.

The proposed use of inertial data in synergy with image from camera in each IS-camera couple within the sensor network leads to relax the rotations among the virtual cameras. From the perspective of extrinsic parameters what remains is the translation among them. We take the advantage of the defined framework to propose a method to estimate the translation vectors among virtual cameras within the network.

In the introduced homographic framework, the data are registered using the homography transformations which are directly obtained by the coupled IS to the camera and the estimated translation vector. Due to imperfection of both the IS observation and the used estimation algorithm for translation (or imperfection of GPS in case of using for outdoor scenario), the obtained geometric entities (2D points) might be corrupted. Therefore it is of importance to be aware of the certainties of the registered data. As the last section of this chapter, the certainties of the homography transformations and their error propagations to the image and Euclidean planes are modelled using statistical geometric analysis.

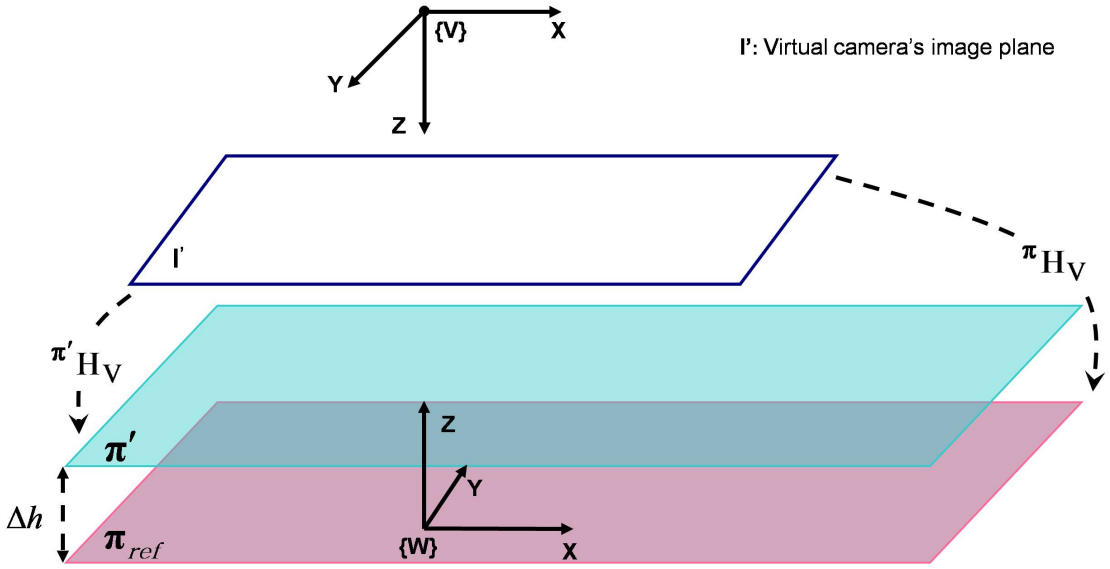


Figure 3.1: Extending homography for planes parallel to π_{ref}. ${}^{\pi}H_V$ is the already available homography matrix among virtual image plane I' and the reference plane π_{ref}. π' is another Euclidean virtual plane, parallel to π_{ref}. Δh is the distance among π and π'. The idea is to obtain ${}^{\pi'}H_V$, the homography between the image plane and π' as a function of ${}^{\pi}H_V$ and Δh (see Eq. (3.2)).

3.2 Parametric homographies among different planes in the framework

In chapter 2, the primary models to perform 3D data registration were presented and supported by presenting some experimental results. Nevertheless, in this section we explore the geometric relations among different involved inertial planes, more specifically. We achieve a set of equations which express the transformations among different Euclidean planes, independent of the cameras intrinsic parameters.

3.2.0.1 Parametric homography between an image plane and Euclidean planes

In chapter 2 the homography matrix from the image plane of a virtual camera V to the world 3D plane $\boldsymbol{\pi}_{\mathbf{ref}}$ was obtained as ${}^{\pi}H_V$ (see Eq. (2.17)) as following:

$$ {}^{\pi}H_v^{-1} = \begin{bmatrix} f_x & 0 & f_x t_1 + u_0 t_3 \\ 0 & -f_y & f_y t_2 + v_0 t_3 \\ 0 & 0 & t_3 \end{bmatrix} \tag{3.1} $$

It is also desired to obtain the homography matrix from a virtual image to another world 3D plane parallel to $\boldsymbol{\pi}_{\mathbf{ref}}$ once we already have ${}^{\pi}H_V$. Lets consider π' as a 3D plane which is parallel to $\boldsymbol{\pi}_{\mathbf{ref}}$ and has a height Δh w.r.t it (see Fig. 3.1). ${}^{\pi'}H_V$ denotes the homography transformation which maps points of the image plane of V onto π'. By substituting t_3 in the equation (2.17) with $t_3 + \Delta h$, ${}^{\pi'}H_V$ can be expressed as a function of ${}^{\pi}H_V$ and Δh as follows:

$$ {}^{\pi'}H_V^{-1}(\Delta h) = {}^{\pi}H_V^{-1} + \Delta h P \hat{\mathbf{k}}^T \tag{3.2} $$

where $P = [\ u_0 \quad v_0 \quad 1]^T$ is the principal point of the camera V and $\hat{\mathbf{k}}$ is the unit vector of the Z axis.

3.2.0.2 Parametric homography relation among Euclidean inertial planes

Suppose π' is an inertial-plane with an Euclidean distance $\triangle h$ to the reference inertial plane π_{ref}. ${}^{\pi'}H_{\pi}$ denotes the homography transformation among the two inertial-planes, induced by the image plane of a virtual camera, and is desired to be obtained (see Fig. 3.2). Such a homography transformation can be expressed by the following equation:

$$ {}^{\pi'}H_{\pi} = {}^{\pi'}H_V {}^{\pi}H_V^{-1} \tag{3.3} $$

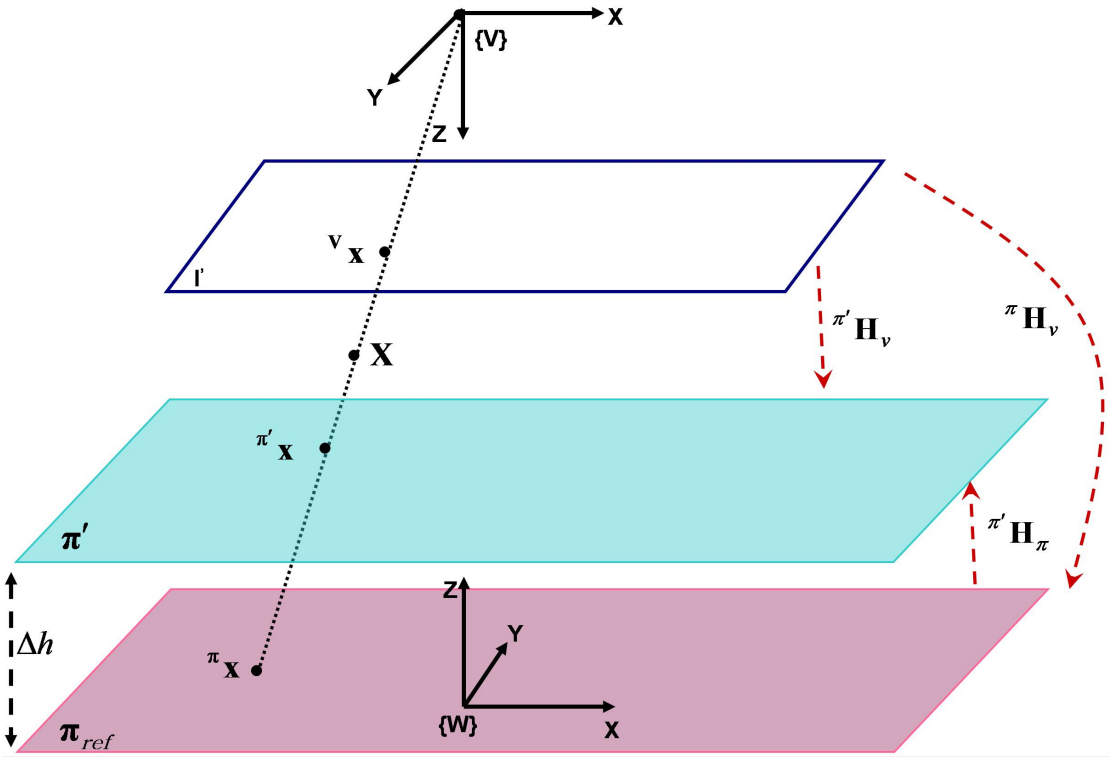


Figure 3.2: Parametric homography among an inertial-plane π' and the reference inertial-plane π_{ref}: The homography transformation ${}^{\pi'}H_{\pi_{ref}}$, induced by image plane of virtual camera V, which maps points from π_{ref} onto π' can be expressed as a function of Δh, Δh being the Euclidean distance among two inertial-planes.

where ${}^{\pi'}H_V$ is the homography transformation among the image plane of a virtual camera V and the inertial-plane π', and ${}^{\pi}H_V$ is the homography transformation between the image plane of V and the reference inertial-plane π_{ref}. By substituting ${}^{\pi'}H_V$ with Eq. (3.2), Eq. (3.3) becomes:

$$ {}^{\pi'}H_{\pi} = ({}^{\pi}H_V^{-1} + \Delta h P \hat{\mathbf{k}}^T)^{-1} {}^{\pi}H_V^{-1} \tag{3.4} $$

The term $({}^{\pi}H_V^{-1} + \Delta h P \hat{\mathbf{k}}^T)^{-1}$ in above equation can be written in an equivalent form using the Sherman-Morrison-Woodbury* formula [Bjo96, Hag89] as follow-

*Considering A as a square matrix and U and V as two column vectors, the Sherman-

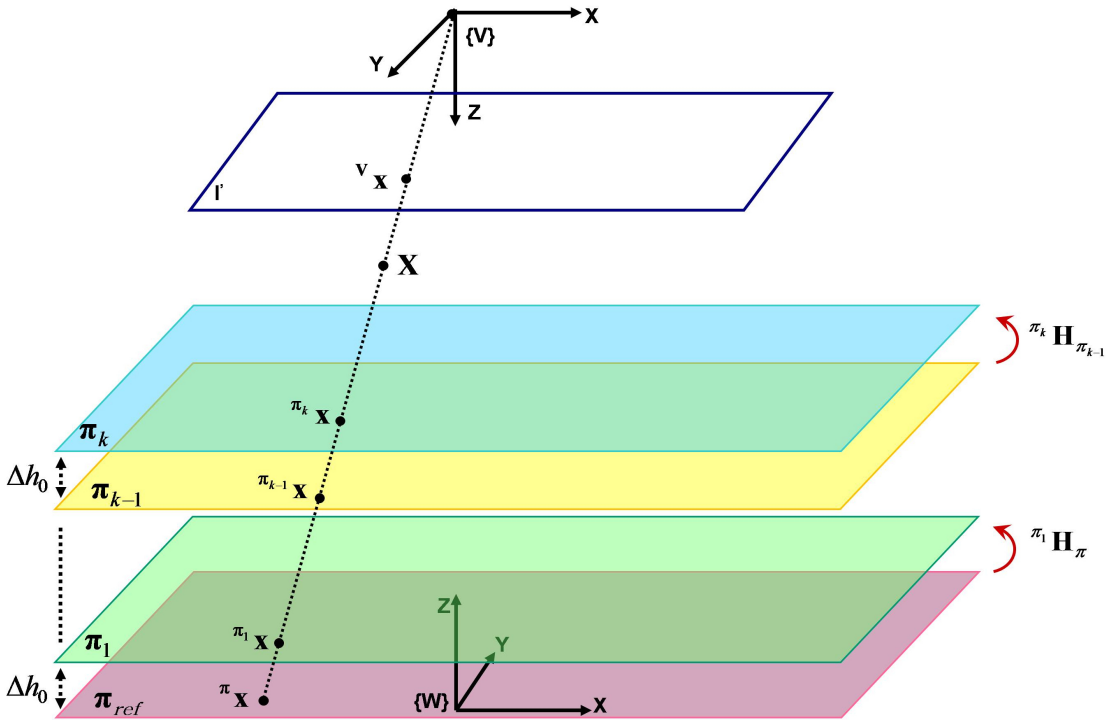


Figure 3.3: Parametric homography among two consecutive inertial-planes induced by a virtual image: The image shows a set of Euclidean inertial-planes where the distance among two consecutive planes is equal to Δh_0. In this case, the homography transformation among any two consecutive inertial-planes can be expressed as a function of Δh_0 and the index of the plane (see Eq. (3.13)).

ing:

$$({}^{\pi}H_V^{-1} + \Delta h P \hat{\mathbf{k}}^T)^{-1} \equiv {}^{\pi}H_V - \frac{{}^{\pi}H_V P \hat{\mathbf{k}}^T {}^{\pi}H_V}{\alpha + \hat{\mathbf{k}}^T {}^{\pi}H_V P} \tag{3.5}$$

where $\alpha = \frac{1}{\triangle h}$. Eventually, Eq. (3.4) after simplifications can be expressed as a function of the distance between two inertial-planes:

$$\boxed{{}^{\pi'}H_{\pi}(\alpha) = \mathbf{I}_{3\times 3} - f(\alpha)\boldsymbol{\Gamma}} \tag{3.6}$$

Morrison-Woodbury formula gives $(A+UV^T)^{-1} = A^{-1} - \frac{A^{-1}UV^TA^{-1}}{1+V^TA^{-1}U}$.

where $f(\alpha)$ is an scalar function of the vertical distance between two inertial plane as following:

$$f(\alpha) = \frac{1}{\alpha t_3 + 1} \tag{3.7}$$

and $\mathbf{\Gamma}$ is a 3×3 matrix equal to

$$\mathbf{\Gamma} = [\ -t_1 \quad t_2 \quad 1\ ]^T\ \hat{\mathbf{k}}^T \tag{3.8}$$

Note that $\mathbf{\Gamma}$ is constant for all inertial planes induced by the camera V (assuming no movement for the cameras) and also independent of the camera intrinsic parameters. Eq. (3.6) is interesting in the sense that once a basic homography ${}^{\pi}H_V$ to project an image to the reference inertial plane π_{ref} is obtained, a direct projection, which is independent to the intrinsic parameters, can be performed from π_{ref} to any arbitrary inertial plane namely π' with just knowing the Euclidean distance (Δh) among them for the purpose of 3D data registration.

While Eq. (3.6) expresses the projective relation among the reference plane π_{ref} and other inertial planes, we are interested to obtain some equation which could express the projective relation among any two consecutive inertial planes, namely ${}^{\pi_k}H_{\pi_{k-1}}$. Fig. (3.3) depicts a set of inertial planes where the Euclidean distance among any two consecutive planes is equal to Δh_0. Suppose ${}^{\pi_k}H_{\pi}$ expresses the homography projection, induced by a virtual image, from the reference plane π_{ref} to kth inertial plane. Such a transformation can be written as:

$${}^{\pi_k}H_{\pi} = ({}^{\pi_k}H_{\pi_{k-1}})\,({}^{\pi_{k-1}}H_{\pi}) \tag{3.9}$$

and then

$${}^{\pi_k}H_{\pi_{k-1}} = ({}^{\pi_k}H_{\pi})\,({}^{\pi_{k-1}}H_{\pi}^{-1}) \tag{3.10}$$

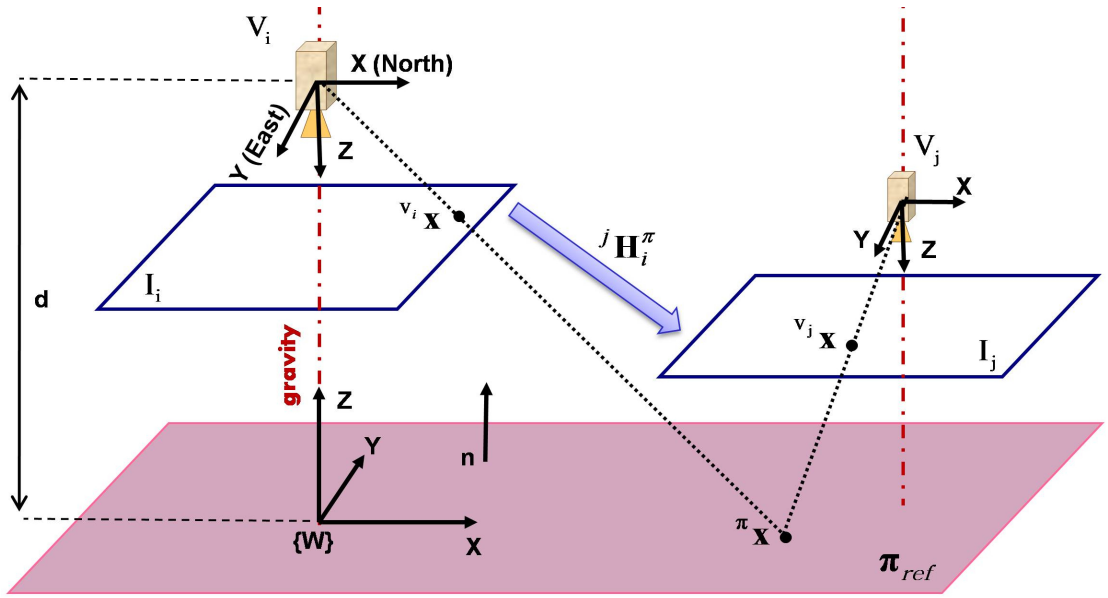


Figure 3.4: Homography between image planes of two virtual cameras.

In Eq. (3.10), by substituting the terms ${}^{\pi_k}H_{\pi}$ with its equivalence from Eq. (3.6) and ${}^{\pi_{k-1}}H_{\pi}$ with Eq.(3.4) we have (considering $\alpha_0 = 1/\Delta h_0$) :

$$ {}^{\pi_k}H_{\pi_{k-1}} = (\mathbf{I}_{3\times3} - \frac{k}{\alpha_0 t_3 + k}\mathbf{\Gamma}) \left[({}^{\pi}H_v^{-1} + \frac{k-1}{\alpha_0} P\hat{\mathbf{k}}^T)^{-1} {}^{\pi}H_v^{-1} \right]^{-1} \tag{3.11} $$

$$ = (\mathbf{I}_{3\times3} - \frac{k}{\alpha_0 t_3 + k}\mathbf{\Gamma})(\mathbf{I}_{3\times3} + \frac{k-1}{\alpha_0 t_3}\mathbf{\Gamma}) \tag{3.12} $$

after simplification:

$$ \boxed{ {}^{\pi_k}H_{\pi_{k-1}}(k, \alpha_0) = \mathbf{I}_{3\times3} - g(\alpha_0, k)\mathbf{\Gamma} } \tag{3.13} $$

where $g(\alpha_0, k)$ is an scalar function whose inputs are the index of inertial plane π_k and the vertical resolution factor α_0 as following:

$$g(\alpha_0, k) = \frac{1}{\alpha_0 t_3 + k} \tag{3.14}$$

As previously mentioned, $\mathbf{\Gamma}$ is a constant 3×3 matrix which is independent of the camera intrinsic parameters. Therefore the obtained homography ${}^{\pi_k}H_{\pi_{k-1}}$ in Eq. (3.13) expresses a homography matrix which transforms the 2D points from inertial plane π_{k-1} to its consecutive inertial plane π_k, independent of the camera intrinsic parameters (as expected), and is a function of k and α_0.

3.2.0.3 Parametric homographic among image planes of virtual cameras

In the previous section, the homography transformation between image plane of a virtual camera and an Euclidean virtual plane (π) was obtained. Here we continue to explain what would be the homography transformation between the images of two virtual cameras in a parametric form. Fig. 3.4 depicts two virtual cameras V_i and V_j with their reference frames. With no lose of generality, we consider V_i as the world reference frame here. The idea is to obtain ${}^jH_i^\pi$, the homography matrix among V_i and V_j, induced by an inertial plane such as π. Based on equation (2.6), ${}^jH_i^\pi$ can be expressed as:

$${}^jH_i^\pi = K_j (R + \frac{1}{d} \Delta\mathbf{t}\mathbf{n}^T) K_i^{-1} \tag{3.15}$$

where K_j and K_i are the camera calibrations matrices, respectively for V_j and V_i. Since there is no rotation among the virtual cameras then R becomes equal to the identity matrix ($I_{3\times3}$) . $\Delta\mathbf{t}$ is a 3-elements vector describing the translation from V_i to V_j. $\mathbf{n} = [\ 0 \quad 0 \quad -1\ ]^T$ is the normal of plane π and d is the distance between π and $\{V_i\}$ along the Z axis of $\{V_i\}$. Therefor, after substitutions and simplifications, Eq. (3.15) can be expressed as:

$${}^jH_i^\pi = K_j [\ \hat{\mathbf{i}} \quad \hat{\mathbf{j}} \quad (\hat{\mathbf{k}} - \frac{\Delta\mathbf{t}}{d})\ ] K_i^{-1} \tag{3.16}$$

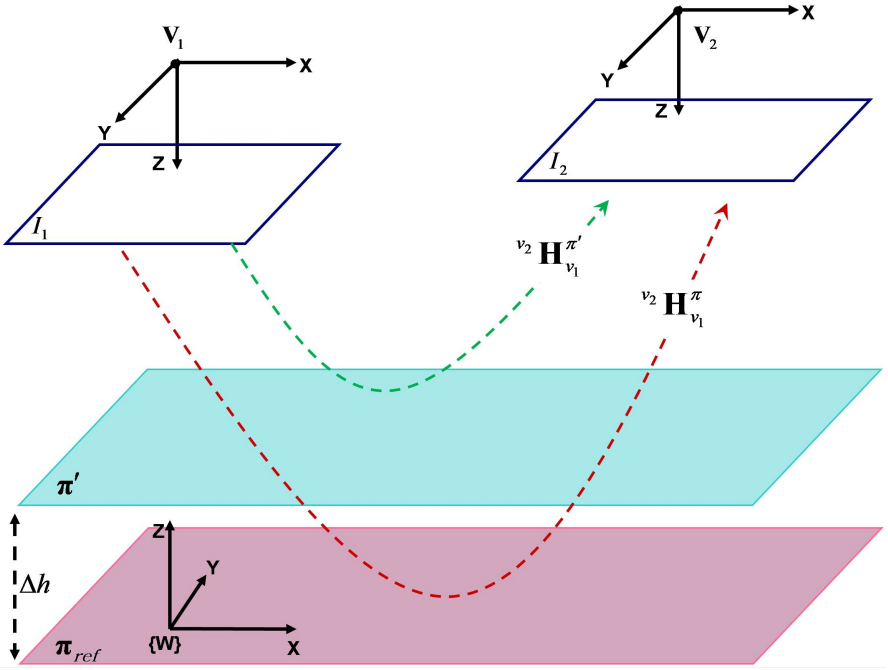


Figure 3.5: Homography between the image planes of two virtual cameras, induced by an inertial plane π' parallel to the reference inertial-plane π_{ref}.

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ are the unit vectors for X, Y and Z axes, respectively. Assuming no changes in camera parameters, Eq. (3.16) generates the homography matrix related to the parameter d, the Euclidean distance between the inertial-plane and $\{V_i\}$.

Eq. (3.16) expresses the homography relation among the image planes of two cameras, induced by the reference inertial plane π_{ref}. It is interesting to obtain the homography among two image planes induced by another inertial plane (π') using the basic relation from Eq. (3.16). Such a homography matrix can be notated as ${}^{v}2H_{v_1}^{\pi'}$ and is depicted in Fig. (3.5). One can write this homography as:

$$ {}^{v}2H_{v_1}^{\pi'} = ({}^{v}2H_{\pi'})({}^{\pi'}H_{v_1}) = ({}^{\pi'}H_{v_2}^{-1})({}^{\pi'}H_{v_1}) \tag{3.17} $$

The terms ${}^{\pi'}H_{v_2}^{-1}$ and ${}^{\pi'}H_{v_1}$ can be replaced by their equivalences using from Eq.

(3.2) and (3.5), respectively:

$$ {}^{v_2}H_{v_1}^{\pi'} = ({}^{\pi}H_{v_2}^{-1} + \frac{1}{\alpha} P_2 \hat{\mathbf{k}}^T)({}^{\pi}H_{v_1} - \frac{{}^{\pi}H_{v_1} P_1 \hat{\mathbf{k}}^T {}^{\pi}H_{v_1}}{\alpha + \hat{\mathbf{k}}^T {}^{\pi}H_{v_1} P_1}) \tag{3.18} $$

where P_1 and P_2 are respectively the principal vectors of the virtual cameras V_1 and V_2. α is equal to the inverse of Δh (the distance among the two inertial planes π_{ref} and π'). After simplification and replacing ${}^{\pi}H_{v_2}^{-1}\,{}^{\pi}H_{v_1}$ with ${}^{v_2}H_{v_1}^{\pi}$ we will have:

$$ \boxed{ {}^{v_2}H_{v_1}^{\pi'} = {}^{v_2}H_{v_1}^{\pi} \,(\, \mathbf{I}_{3\times 3} - f(\alpha)\, P_1\, \hat{\mathbf{k}}^T \,) + f(\alpha)\, P_2\, \hat{\mathbf{k}}^T } \tag{3.19} $$

where $f(\alpha)$ was previously defined in Eq. (3.7) and is an scalar function of the distance among two inertial planes:

$$ f(\alpha) = \frac{1}{\alpha\, t_3 + 1} \tag{3.20} $$

As one can see, the Eq. 3.19 expresses the homography among two virtual cameras induced by an inertial plane π' parallel to π_{ref}, by using a linear equation of the homography among the same virtual cameras but induced through the reference inertial plane π_{ref}.

3.2.1 Volumetric reconstruction: a recursive form

In previous chapter an algorithm to perform 3D data registration of object or human was already proposed. Here, by having the new parametric functions which generate the homographies among different projective and Euclidean planes, we introduce a new version of the algorithm in Alg. 2 which is capable of performing the 3D reconstruction task in a recursive manner. k is the index of inertial plane

Algorithm 2: 3D data registration using inertial-planes in a recursive form. k is the index of inertial plane and α_0 is the inverse of the Euclidean distance between two consecutive inertial planes.

Function `ThreeDimRegistration()`
begin
  /* Initialization */
  for $i \leftarrow 1$ to N_c do
    ${}^{v_i}H_{c_i} \leftarrow K_i \, {}^{v_i}R_{c_i} \, K_i^{-1}$
    $I_{v_i} \leftarrow {}^{v_i}H_{c_i} \; I_{c_i}$
    ${}^{\pi_{ref}}H_{v_i} \leftarrow inv(K_{c_i}[\ \hat{\mathbf{i}} \quad -\hat{\mathbf{j}} \quad \mathbf{t}_{c_i}\ ])$ Eq. (2.17)
    $\pi_{ref}^{(v_i)} \leftarrow {}^{\pi_{ref}}H_{v_i} \, I_{v_i}$
    $\mathbf{\Gamma}_{v_i} \leftarrow [\ -t_1 \quad t_2 \quad 1\ ]^T \, \hat{\mathbf{k}}^T$ Eq. (3.8)
  main function
  /* Performing recursive registration for each camera */
  for $i \leftarrow 1$ to N_c do
    $\pi_{N_\pi-1}^{[v_i]} \leftarrow$ **RegisterRecursive** $(N_\pi - 1 \ , \ \pi_{N_\pi-1}^{[v_i]})$
  /* Performing cell-wise intersection, see Fig. 4.5 */
  for $i \leftarrow 0$ to $N_\pi - 1$ do
    $\pi_i \leftarrow \prod_{j=1}^{N_c} \pi_i^{[j]}$
end

Function **RegisterRecursive** $(k, \pi^{(v_i)})$
begin
  if $k == 0$ then
    return $\pi_{ref}^{(v_i)}$
  Recursively registering
  else
    ${}^{\pi_k}H_{\pi_{k-1}} \leftarrow$ `GenerateNextHomography` $(k-1)$
    $\pi_k^{(v_i)} \leftarrow$ `Warp` $(H \ , \ $ **RegisterRecursive** $(k-1, \pi_{k-1}^{(v_i)}))$
    return $\pi_k^{(v_i)}$
end

Function `GenerateNextHomography` (k)
begin
  ${}^{\pi_k}H_{\pi_{k-1}}(k, \alpha_0) = \mathbf{I}_{3\times3} - g(k, \alpha_0)\mathbf{\Gamma}$ Eq. (3.13)
  return H
end

---

inline Function `g` (k, α_0) return $\frac{1}{\alpha_0 t_3 + k}$ Eq. (3.14)

and α_0 is the inverse of the Euclidean distance between two consecutive inertial planes. In this algorithm, ThreeDimRegistration() is the main function in which firstly the variables are initialized. After initialization, for each camera, RegisterRecursive() function is called. This function recursively projects and registers the image data onto the consecutive inertial planes in the scene. In this function, Warp() is a function which performs the operation of usual homography warping.

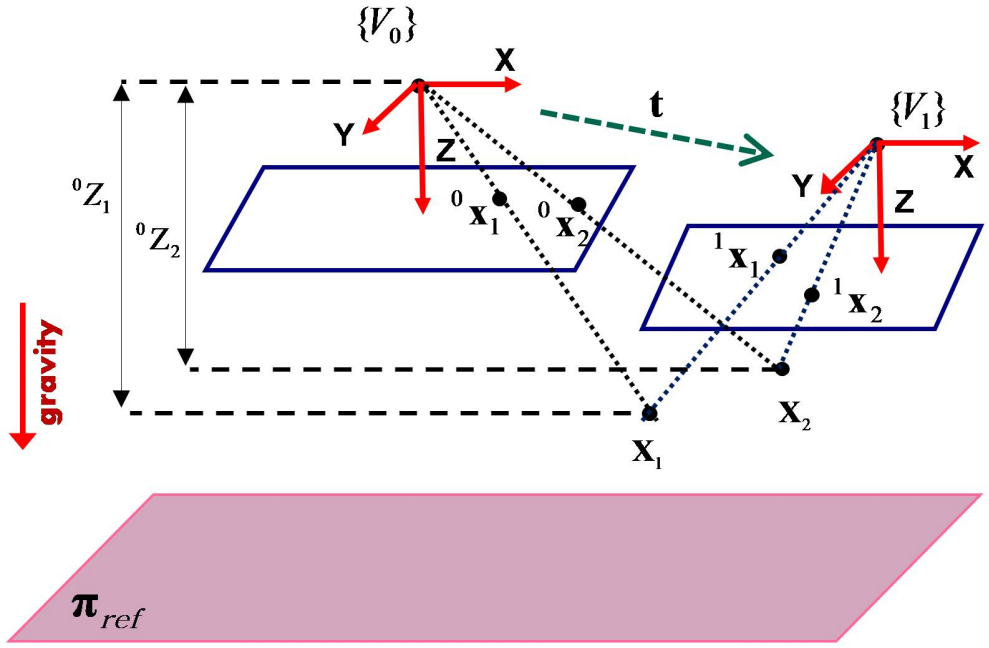


Figure 3.6: Translation between two virtual cameras. $\mathbf{X}_1$ and $\mathbf{X}_2$ are two arbitrary 3D point in the scene. Z_1 and Z_2 are the relative heights of $\mathbf{X}_1$ and $\mathbf{X}_2$ w.r.t. first camera, V_0. $\mathbf{t}$ is the translation vector among two virtual cameras which can be estimated using the proposed method.

3.3 Translation estimation among two virtual cameras

Estimation of extrinsic parameters in a camera network is one of the prerequisites for many computer vision algorithms, including the proposed data registration framework. Extrinsic parameters are comprised of a rotation matrix and a translation vector. Having an IS already coupled with each camera within the network leads to relax the rotation among them. In terms of extrinsic parameters what remains is the translation part. Here we take the advantage of having IS and camera coupled and propose an efficient method to estimate the translation vector $\mathbf{t}$ among virtual cameras.

Our approach is based on having the heights of two arbitrary 3D points in the scene such $\mathbf{X}_1 = [\ X_1 \quad Y_1 \quad Z_1\ ]^T$ and $\mathbf{X}_2 = [\ X_2 \quad Y_2 \quad Z_2\ ]^T$ (see Fig. 3.6) with respect to a camera (namely V_0) within the network and then to have just their

correspondences in the images (Note that a real camera and its correspondent virtual camera have the same centres). Suppose ${}^0\mathbf{X}_1 = [\ {}^0X_1 \quad {}^0Y_1 \quad {}^0Z_1\ ]^T$ and ${}^0\mathbf{X}_2 = [\ {}^0X_2 \quad {}^0Y_2 \quad {}^0Z_2\ ]^T$ are coordinates of the two 3D points $\mathbf{X}_1$ and $\mathbf{X}_2$ expressed in the first virtual camera center, respectively. Based on the assumption, the parameters ${}^0Z_1$ and ${}^0Z_2$ which indicate the heights of $\mathbf{X}_1$ and $\mathbf{X}_2$ in $\{V_0\}$ are known. Recalling that V_0 is downward-looking and has its optical axis parallel to the gravity. Therefore the term height here is equal to the Z component of the 3D point. Then using projective property of a camera we can have all three components of ${}^0\mathbf{X}_1$ and ${}^0\mathbf{X}_2$ numerically obtained in a metric scale:

$$\begin{cases} {}^0\mathbf{X}_1 = {}^0Z_1\ (K_1^{-1}\ {}^0\mathbf{x}_1) \\ {}^0\mathbf{X}_2 = {}^0Z_2\ (K_1^{-1}\ {}^0\mathbf{x}_2) \end{cases} \tag{3.21}$$

where ${}^0\mathbf{x}_1$ and ${}^0\mathbf{x}_2$ are respectively the imaged points of $\mathbf{X}_1$ and $\mathbf{X}_2$ in the first virtual camera image plane. The same can be considered for the second virtual camera. Suppose ${}^1\mathbf{X}_1 = [\ {}^1X_1 \quad {}^1Y_1 \quad {}^1Z_1\ ]^T$ and ${}^1\mathbf{X}_2 = [\ {}^1X_2 \quad {}^1Y_2 \quad {}^1Z_2\ ]^T$ are respectively coordinations of the 3D points $\mathbf{X}_1$ and $\mathbf{X}_2$ expressed in the second virtual camera center ($\{V_1\}$). Then likewise using projective property of a camera we can have the following equation:

$$\begin{cases} {}^1\mathbf{X}_1 = {}^1Z_1\ (K_2^{-1}\ {}^1\mathbf{x}_1) \\ {}^1\mathbf{X}_2 = {}^1Z_2\ (K_2^{-1}\ {}^1\mathbf{x}_2) \end{cases} \tag{3.22}$$

In contrary to the Eq. (3.21), Eq. (3.22) can not be numerically obtained yet, since it has two unknown values for ${}^1Z_1$ and ${}^1Z_2$ (the heights of the 3D points w.r.t $\{V_1\}$). The terms $(K_2^{-1}\ {}^1\mathbf{x}_1)$ and $(K_2^{-1}\ {}^1\mathbf{x}_2)$ in Eq. (3.22) as well express the 3D position of the points ${}^1\mathbf{X}_1$ and ${}^1\mathbf{X}_2$ however up to scale factors ${}^1Z_1$ and ${}^1Z_2$. Here it is desirable to rewrite the Eq. (3.22) as the following:

$$\begin{cases} {}^1\mathbf{X}_1 = {}^1Z_1 \, {}^1\hat{\mathbf{X}}_1 \\ {}^1\mathbf{X}_2 = {}^1Z_2 \, {}^1\hat{\mathbf{X}}_2 \end{cases} \tag{3.23}$$

where ${}^1\hat{\mathbf{X}}_1 = (K_2^{-1} \, {}^1\mathbf{x}_1)$ and ${}^1\hat{\mathbf{X}}_2 = (K_2^{-1} \, {}^1\mathbf{x}_2)$. Then the Eq. (3.21) and Eq. (3.23) can be related through the translation vector between $\{V_0\}$ and $\{V_1\}$ as:

$$\begin{cases} {}^0\mathbf{X}_1 = R \, {}^1\mathbf{X}_1 + \mathbf{t} & = R \, {}^1Z_1 \, {}^1\hat{\mathbf{X}}_1 + \mathbf{t} \\ {}^0\mathbf{X}_2 = R \, {}^1\mathbf{X}_2 + \mathbf{t} & = R \, {}^1Z_2 \, {}^1\hat{\mathbf{X}}_2 + \mathbf{t} \end{cases} \tag{3.24}$$

where R is the rotation matrix between two cameras and $\mathbf{t} = [\ t_1 \quad t_2 \quad t_3\ ]^T$. Since we are considering the virtual cameras and there is not rotation among them then we can simply consider R as an 3×3 identity matrix. In Eq. (3.24) there are five unknown parameters including ${}^1Z_1$, ${}^1Z_2, t_1, t_2, t_3$. Nevertheless there are also six linear equations which are adequate to obtain the unknowns. In order to estimate the five unknowns Eq. (3.24) can be arranged in the form of

$$A\,\mathbf{x} = B \tag{3.25}$$

where

$$A = \begin{bmatrix} {}^1\hat{\mathbf{X}}_1 & \mathbf{0}_{3\times1} & I_{3\times3} \\ \mathbf{0}_{3\times1} & {}^1\hat{\mathbf{X}}_2 & I_{3\times3} \end{bmatrix} \tag{3.26}$$

$$\mathbf{x} = \begin{bmatrix} {}^1Z_1 & {}^1Z_2 & t_1 & t_2 & t_3 \end{bmatrix}^T \tag{3.27}$$

$$B = \begin{bmatrix} {}^0\mathbf{X}_1 \\ {}^0\mathbf{X}_2 \end{bmatrix} \tag{3.28}$$

Therefore $\mathbf{x}$ in Eq. (3.25) can be estimated using the least square approach as follows:

$$\mathbf{x} = (A^T A)^{-1} A^T B \tag{3.29}$$

and consequently the translation vector between the two virtual cameras' references, $\{V_0\}$ and $\{V_1\}$, are estimated. Using the same mentioned method, the translation between other virtual cameras can be estimated.

3.3.1 Error analysis of the translation vector estimation

Here we analyse the accuracy of the proposed method in different cases such as noise in IS observation, error of height measurement of two 3D points, error in extraction of pixel coordinates of two 3D points in the images and effects of relative height (distance) of 3D points w.r.t. camera. In order to have enough data for the analysis, a simulator is prepared which can generate thousands of samples based on the given criteria. The simulated volume has a dimension equal to $500 \times 500 \times 1000\, cm^3$. In each generated sample, two virtual cameras are randomly placed on the ceiling of the volume with a maximum height of $200\, cm$ from the ceiling. Moreover, in each generated sample, two 3D points are randomly selected from the volume. One common criterion for selecting two 3D points is that they need to be inside the visible area by two cameras as well as having a maximum height of $1000\, cm$ to the ceiling. The estimation error has been evaluated under the following conditions

3.3.1.1 IS noise in 3D orientation sensing

An IS has several kind of outputs which among them we use just its 3D orientation output. Normally in MEMS*-IS the accuracy of the rotation angle around the

*Microelectromechanical systems

Algorithm 3: Simulation to evaluate affect of IS noise on translation estimation. 500,000 samples are generated. In each sample an appropriate Gaussian noise is considered for roll, pitch and yaw angles of IS observation. The result of the simulation is the error distributions for three elements of estimated $\mathbf{t}$. The distributions for the input noise (IS observation) and the estimation are shown in Fig. 3.7.

$\mathbf{K}_{1,2} \leftarrow \begin{bmatrix} 500 & 0 & 300 \\ 0 & 505 & 280 \\ 0 & 0 & 1 \end{bmatrix}$ /* Extrinsic parameters of two cameras */

for $i \leftarrow 1$ to $500,000$ do

- /* Gaussian random noise for IS orientation */
- $\varepsilon_{roll} \leftarrow N(0, 0.5/3)$ /* Noise of roll (of IS observation) */
- $\varepsilon_{pitch} \leftarrow N(0, 0.5/3)$ /* Noise of pitch (of IS observation) */
- $\varepsilon_{yaw} \leftarrow N(0, 1.0/3)$ /* Noise of yaw (of IS observation) */
- $\mathbf{R} \leftarrow Eulor2RotationMatrix(\varepsilon_{roll}, \varepsilon_{pitch}, \varepsilon_{yaw})$
- /* translation between two cameras */
- $\mathbf{t} \leftarrow \begin{bmatrix} 100 + 400 * RND \\ 100 + 400 * RND \\ 200 * RND \end{bmatrix}$; /* $0 <= RND <= 1$ (random function)
- /* two random 3D points (ground truth as well) */
- ${}^0\mathbf{X}_0^g \leftarrow \begin{bmatrix} -500 + 1000 * RND \\ -500 + 1000 * RND \\ 50 + 600 * RND \end{bmatrix}$; ${}^0\mathbf{X}_1^g \leftarrow \begin{bmatrix} -500 + 1000 * RND \\ -500 + 1000 * RND \\ 50 + 600 * RND \end{bmatrix}$
- ${}^1\mathbf{X}_0^g \leftarrow {}^0\mathbf{X}_0^g + \mathbf{t}$
- ${}^1\mathbf{X}_1^g \leftarrow {}^0\mathbf{X}_1^g + \mathbf{t}$
- /* heights of 3D points w.r.t. the first camera reference frame */
- ${}^0h_0 \leftarrow {}^0\mathbf{X}_0^g(3)$
- ${}^0h_1 \leftarrow {}^0\mathbf{X}_1^g(3)$
- /* applying the IS' noise on two generated 3D points */
- ${}^1X_0 \leftarrow \mathbf{R} * {}^0\mathbf{X}_0^g + \mathbf{t}$
- ${}^1X_1 \leftarrow \mathbf{R} * {}^0\mathbf{X}_1^g + \mathbf{t}$
- /* points on image planes */
- ${}^0\mathbf{x}_0 \leftarrow \mathbf{K}_1 * {}^0\mathbf{X}_0^g$
- ${}^0\mathbf{x}_1 \leftarrow \mathbf{K}_1 * {}^0\mathbf{X}_1^g$
- ${}^1\mathbf{x}_0 \leftarrow \mathbf{K}_2 * {}^1X_0$
- ${}^1\mathbf{x}_1 \leftarrow \mathbf{K}_2 * {}^1X_1$
- /* estimating t */
- ${}^0\widetilde{\mathbf{X}}_0 \leftarrow {}^0h_0 * \mathbf{K}_1^{-1} * {}^0\mathbf{x}_0$
- ${}^0\widetilde{\mathbf{X}}_1 \leftarrow {}^0h_1 * \mathbf{K}_1^{-1} * {}^0\mathbf{x}_1$
- ${}^1\widetilde{\mathbf{X}}_0 \leftarrow \mathbf{K}_2^{-1} * {}^1\mathbf{x}_0$
- ${}^1\widetilde{\mathbf{X}}_1 \leftarrow \mathbf{K}_2^{-1} * {}^1\mathbf{x}_1$
- $\mathbf{A} = \begin{bmatrix} {}^1\widetilde{\mathbf{X}}_1 & \mathbf{0}_{3\times1} & I_{3\times3} \\ \mathbf{0}_{3\times1} & {}^1\widetilde{\mathbf{X}}_2 & I_{3\times3} \end{bmatrix}$
- $\mathbf{B} = \begin{bmatrix} {}^0\widetilde{\mathbf{X}}_1 \\ {}^0\widetilde{\mathbf{X}}_2 \end{bmatrix}$
- $\mathbf{X} \leftarrow lscov(\mathbf{A}, \mathbf{B})$ /* Least-squares solution for $AX = B$ */
- $\widetilde{\mathbf{t}}(1) \leftarrow \mathbf{X}(3); \widetilde{\mathbf{t}}(2) \leftarrow \mathbf{X}(4); \widetilde{\mathbf{t}}(3) \leftarrow \mathbf{X}(5);$
- $\mathbf{E}_{XYZ}^i \leftarrow \widetilde{\mathbf{t}} - \mathbf{t}$ /* error in X,Y and Z axes in estimated translation */

hist($\mathbf{E}$) /* plot the distribution histogram of $\mathbf{E}$ */

vertical axis (heading direction) is less rather than the other two angles [KHJG11]. For example, an inertial sensor such as Xsens-MTi [xse] has a precision around 0.5° on the roll and pitch angles, and 1.0° on the heading directions. Of course one can use some techniques to improve the accuracy of IS. For example, Kalantart et al. in [KHJG11] discussed this subject and proposed a method to improve the accuracy of IS better than 0.001°. Nevertheless, in the following we discuss the impact of the accuracy of IS observation (orientation sensing) on the proposed method to estimate the translation, where a Xsens-MTi [xse] is used to measure the orientation. Fig. 3.7-top shows the noise distributions for the three angles of IS (roll, pitch and yaw). 500,000 random samples are generated in the simulation. The noise distributions are considered as Gaussian white noise $N(\mu = 0, \delta)$. The standard deviation value (δ) for each one of the angles (roll, pitch and yaw) is supposed as $\frac{1}{3}$ of its corresponding maximum error value (0.5°, 0.5° and 1.0°, respectively), which yields to have $\delta_{roll} = 0.17^\circ$, $\delta_{pitch} = 0.17^\circ$ and $\delta_{yaw} = 0.33^\circ$. These IS measurement noise are applied to the generated data in the simulation and the translation vector $\mathbf{t} = [\ X \quad Y \quad Z\ ]^T$ for each sample is estimated (see Alg. 3)). Fig. 3.7-bottom depicts the error distributions for the three components of $\mathbf{t}$. One can see that the error distributions along three axes have Gaussian shapes as well.

3.3.1.2 Height measurement noise

Error in measurement of the relative heights of two 3D points in the scene can also affect the accuracy of the translation estimation process. Fig. 3.8 depicts an analysis on 110,000 simulated data for this purpose. In this experiment, some noise in measuring the relative heights of two 3D points w.r.t. first camera are injected. By assuming the maximum error value in the height measurement of 3D points in the scene as 10cm, a Gaussian white noise $N(\mu = 0, \delta = \frac{10}{3} = 3.33)$ is applied (plotted in purple) to the proposed algorithm. The error distributions for three elements of the estimated translation vector $\mathbf{t}$ are plotted in blue, red and green.

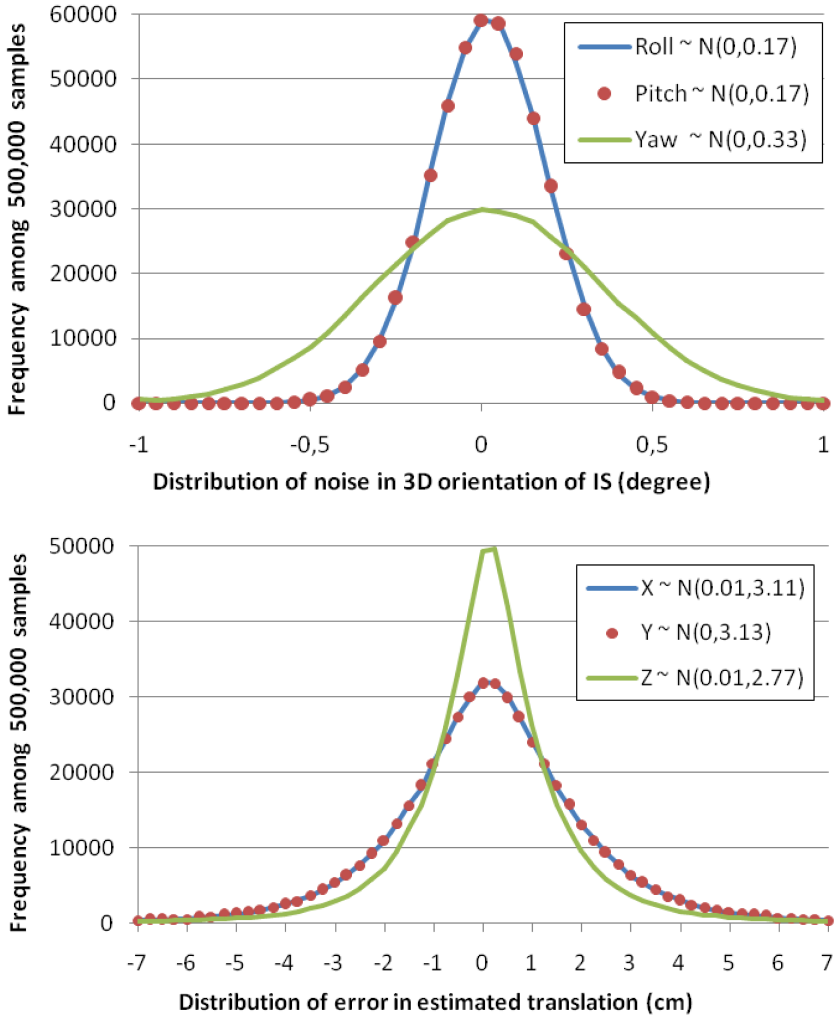


Figure 3.7: Analysis of noise impact in IS orientation for estimating translation among two virtual cameras: Left figure shows the noise distribution for the three angles (roll, pitch and yaw) observed by an IS. 500,000 samples are simulated where the distributions are considered as Gaussian white noise $N(\mu = 0, \delta)$. For a typical IS such as Xsens-MTi the maximum error values for roll, pitch and yaw are 0.5°, 0.5°and 1.0°, respectively [KHJG11]. Thus the value of δ for each angle is considered $\frac{1}{3}$ of its corresponding maximum error value, which yields to have $\delta_{roll} = 0.17$, $\delta_{pitch} = 0.17$ and $\delta_{yaw} = 0.33$. The bottom image indicates the error distributions in cm for the three elements (X,Y and Z) of estimated $\mathbf{t}$ using simulated data (with the noise distributions shown in the top figure).

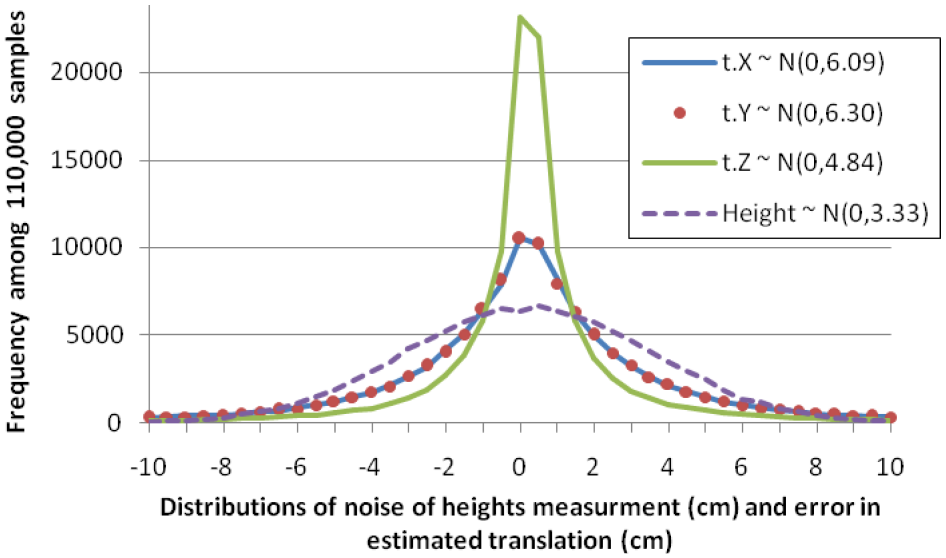


Figure 3.8: Analysis of noise impact in measurement of the heights of two 3D points for estimating translation among two virtual cameras: 110,000 samples are generated in the simulation. By assuming the maximum error value in the height measurement of 3D points in the scene as 10cm, a Gaussian white noise $N(\mu = 0, \delta = \frac{10}{3} = 3.33)$ is applied (plotted in purple). The error distributions for three elements of the estimated translation vector $\mathbf{t}$ is plotted in blue, red and green.

3.3.1.3 Noise in image coordinate extraction of 3D points:

In the proposed translation recovery method, the positions of the 3D points in the image coordinate system (pixel) need to be extracted. Fig. 3.9 demonstrates how errors in extraction of the imaged points can affect the accuracy of the result. 100,000 samples are simulated where the maximum error value in the image coordinates (x and y) of the two 3D points is assumed 5 pixels. The purple plot indicates a Gaussian white noise $N(\mu = 0, \delta = \frac{5}{3} = 1.67)$ applied to image coordinates. The error distributions for three elements of the estimated translation vector $\mathbf{t}$ is plotted in blue, red and green (in cm).

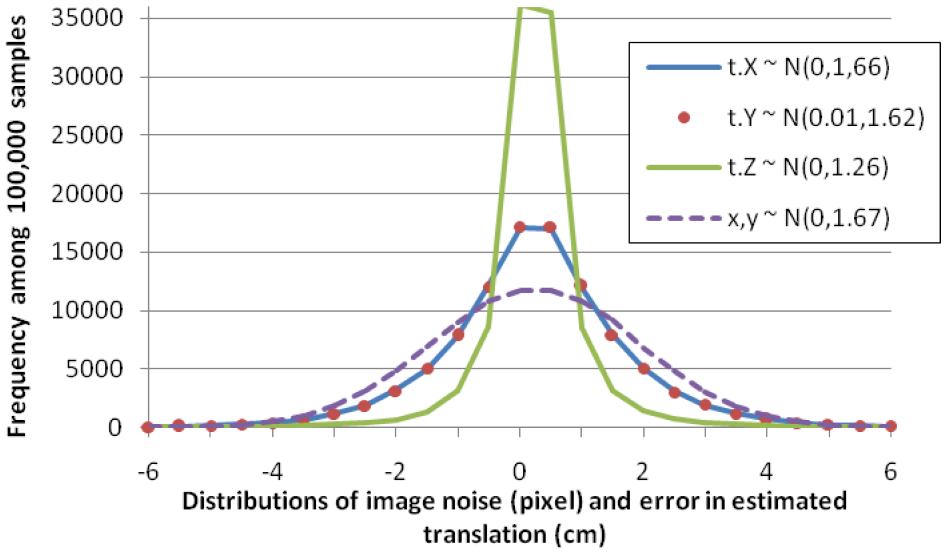


Figure 3.9: Analysis of noise impact in extraction of the image coordinates (in pixel) of the two 3D points for estimating translation among two virtual cameras: 100,000 samples are simulated. The maximum error value in the image coordinates (x and y) of the two 3D points is assumed 5 pixels. A Gaussian white noise $N(\mu = 0, \delta = \frac{5}{3} = 1.67)$ is applied (plotted in purple) to the proposed algorithm. The error distributions for three elements of the estimated translation vector $\mathbf{t}$ is plotted in blue, red and green (in cm).

3.3.1.4 Distance of 3D points to the cameras

In the proposed translation estimation algorithm, two 3D points in the scene need to be selected and their relative heights w.r.t. one camera must be measured. It is worth to analyse the effect of the distances on the accuracy of the result in order to consider it in selection of 3D points from the scene. Fig. 3.10 shows the related analysis when the 3D points are selected from different height w.r.t. first camera in the simulation. The height range is from 250 cm to 1050 cm with the interval of 100 cm yielding totally 9 height values. In each height 10,000 random samples are generated by taking into account the IS noise distributions presented in Fig. 3.7-top. The diagrams of standard deviation and average for the three elements (X, Y and Z) of the estimated translation vector $\mathbf{t}$ are presented in Fig. 3.10 (in cm). As one can see, the distance of the 3D points w.r.t. the cameras has

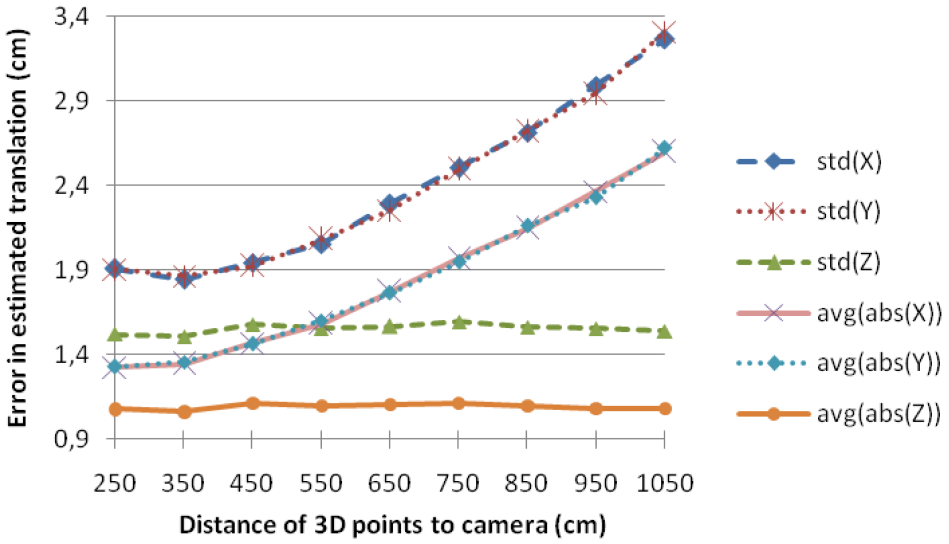


Figure 3.10: Analysis of the relation between the distances of the two 3D points with respect to the first camera and the accuracy of the result in the proposed algorithm to estimate the translation among two virtual cameras: 3D points are placed in different heights with respect to the cameras. The height range is from 250 cm to 1050 cm with the interval of 100 cm yielding totally 9 height values. In each height 10,000 random samples are generated by taking into account the IS noise distributions shown in Fig. 3.7-left. Standard deviation and average for the three elements of the estimated translation vector $\mathbf{t}$ are plotted (in cm).

no significant effect on the accuracy of Z component of the estimated $\mathbf{t}$. However there is an almost linear relation between the distance and the error values for X and Y components of $\mathbf{t}$.

3.4 Uncertainty modelling of inertial-based homography

To be aware of the uncertainties of the registered data in a 3D data registration is important for applications which will use the data, specially when the data are registered by fusion from different sources. The introduced 3D data registration framework uses homography transformations in order to map a 3D point onto an inertial plane as a geometric 2D entity. Such geometric transformations are directly obtained through the principal formula of Eq. (2.6):

$$H = K'(R + \frac{1}{d}\mathbf{t}\mathbf{n}^T)K^{-1} \tag{3.30}$$

The determinant parameters in this formula are the rotation matrix and translation vector. The rotation matrix, R, is obtained from the inertial sensor's observation using Eq. (2.12) and the translation vector can be obtained either by recently introduced two-point-based method or by using a GPS (e.g. in outdoor scenarios). Due to imperfection of measuring device or estimation algorithms the obtained geometric entities (2D points) might be corrupted. In this section we represent the uncertainty of registered data using statistical geometry. Following, first we modelize the uncertainty for the image plane of a virtual camera, then the uncertainties of the points registered on an Euclidean inertial plane are modelized.

3.4.1 Uncertainty of image plane of virtual cameras

The image plane of a virtual camera is obtained by fusion of real camera's image plane and IS observation (orientation) using the concept of infinite homography. We first represent the uncertainty for such a homography and then its uncertainty propagation on the image plane of virtual camera.

The infinite homography presented by Eq. (2.11) depends to the 3D orientation measured by IS. Such an orientation can be presented by a random vector

$$\mathbf{s} = \begin{bmatrix} \theta_r & \theta_p & \theta_y \end{bmatrix}^T \tag{3.31}$$

where θ_r, θ_p and θ_y denote the three elements of the Euler angles (respectively roll, pitch and yaw). We assume that $\mathbf{s}$ has a mean equal to zero and a covariance of

$$\mathbf{\Sigma_s} = diag\{\delta_r^2, \delta_p^2, \delta_y^2\} \tag{3.32}$$

where δ_r, δ_p and δ_y are respectively the standard deviations for θ_r, θ_p and θ_y. In the homography formula of Eq. (2.11), ${}^vH_c$, can be expressed as a linear function of the orientation vector:

$$f : \mathbf{s} \mapsto {}^v H_c \tag{3.33}$$

where it maps the input three angles into the 9-elements homography matrix (R^3 into R^9). For simplicity, we express the homography matrix H as

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \tag{3.34}$$

and assume $\mathbf{h}$ as a vector form of H. Then we consider ${}^v\mathbf{h}_c$ as a random vector and are interested to model its uncertainty. Using a first-order Taylor approximation, as presented in [Fau], the uncertainty of H can be obtained as:

$$\mathbf{\Sigma_h} = \mathbf{J_{h,s}}\, \mathbf{\Sigma}_s\, \mathbf{J_{h,s}^T} \tag{3.35}$$

where $\mathbf{J}$ is a Jacobian matrix:

$$\mathbf{J}_{\mathbf{h},\mathbf{s}} = \begin{bmatrix} \partial h_1/\partial\theta_r & \partial h_1/\partial\theta_p & \partial h_1/\partial\theta_y \\ \partial h_2/\partial\theta_r & \partial h_2/\partial\theta_p & \partial h_2/\partial\theta_y \\ \vdots & \vdots & \vdots \\ \partial h_9/\partial\theta_r & \partial h_9/\partial\theta_p & \partial h_9/\partial\theta_y \end{bmatrix} \tag{3.36}$$

The homography transformation ${}^{v}H_c$ maps points from real camera to virtual camera's image plane. Having the uncertainty of the homography matrix ${}^{v}H_c$, we consequently can characterize the uncertainty for the mapped points. The points on virtual camera's image plane, ${}^{v}\mathbf{x}$, are obtained by the following mapping:

$${}^{v}\mathbf{x} = {}^{v}H_c\,\mathbf{x} \tag{3.37}$$

$\mathbf{x}$ being a point from real camera's image plane. Assuming no uncertainty in real camera's image, the uncertainty of the points on the virtual image plane can be expressed as following (according to [Heu04]):

$$\mathbf{\Sigma}_{{}^{v}\mathbf{x}} = (\mathbf{I} \otimes {}^{v}\mathbf{x}^T)\,\mathbf{\Sigma}_h\,(\mathbf{I} \otimes {}^{v}\mathbf{x}) \tag{3.38}$$

where $\mathbf{I}$ is a 3×3 identity matrix and $\otimes$ denotes Kronecker product.

3.4.2 Uncertainty of Euclidean inertial-planes

Earlier, we used Eq. (2.17), ${}^{\pi}\mathbf{x} = {}^{\pi}H_v\,{}^{v}\mathbf{x}$, in order to project points from image plane of virtual camera onto Euclidean inertial planes. The uncertainty for such projected points is influenced by first the uncertainty of the points from virtual image, and then by the uncertainty of the homography transformation ${}^{\pi}H_v$. We continue to firstly assume the points on virtual image as certain points and just consider uncertainty for the homography matrix ${}^{\pi}H_v$. Afterwards, we take into account the uncertainties of the virtual image's points and propagate them.

3.4.2.1 Uncertainty of homography from virtual image to Euclidean plane

Previously the uncertainty of homography matrix from real camera image plane onto virtual camera image plane was obtained. In the same way, the uncertainty of ${}^{\pi}H_{v}$ (Eq. (2.17)) can be modelled. Such a homography can be considered as a linear function of the translation vector $\mathbf{t} = [\ t_1 \quad t_2 \quad t_3\ ]^T$. We assume the uncertainty of $\mathbf{t}$ with following covariance matrix:

$$\mathbf{\Sigma_t} = diag\{\delta_{t_1}^2, \delta_{t_2}^2, \delta_{t_3}^2\} \tag{3.39}$$

where δ_{t_1}, δ_{t_2} and δ_{t_3} denote the standard deviations for the three elements of the translation vector. Again we use the first-order of Taylor approximation [Fau] and express the uncertainty of ${}^{\pi}\mathbf{h}_{v}$ (the vector form of ${}^{\pi}H_{v}$) as

$$\mathbf{\Sigma}_{\pi_{\mathbf{h}_v}} = \mathbf{J}_{\pi_{\mathbf{h}_v,\mathbf{t}}}\ \mathbf{\Sigma_t}\ \mathbf{J}^T_{\pi_{\mathbf{h}_v,\mathbf{t}}} \tag{3.40}$$

where $\mathbf{J}_{\pi_{\mathbf{h}_v,\mathbf{t}}}$ is a Jacobian matrix:

$$\mathbf{J}_{\pi_{\mathbf{h}_v,\mathbf{t}}} = \begin{bmatrix} \partial h_1/\partial t_1 & \partial h_1/\partial t_2 & \partial h_1/\partial t_3 \\ \partial h_2/\partial t_1 & \partial h_2/\partial t_2 & \partial h_2/\partial t_3 \\ \vdots & \vdots & \vdots \\ \partial h_9/\partial t_1 & \partial h_9/\partial t_2 & \partial h_9/\partial t_3 \end{bmatrix} \tag{3.41}$$

After simplification, Eq. (3.40) becomes as

$$\boldsymbol{\Sigma}_{\pi_{\mathbf{h}_v}} = \frac{1}{t_3^4}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \delta_{t_1}^2 t_3^2 + \delta_{t_3}^2 t_1^2 & 0 & 0 & -\delta_{t_3}^2 t_1 t_2 & 0 & 0 & -\delta_{t_3}^2 t_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\delta_{t_3}^2 t_1 t_2 & 0 & 0 & -\delta_{t_2}^2 t_3^2 + \delta_{t_3}^2 t_2^2 & 0 & 0 & \delta_{t_3}^2 t_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\delta_{t_3}^2 t_1 & 0 & 0 & \delta_{t_3}^2 t_2 & 0 & 0 & \delta_{t_3}^2 \end{bmatrix} \tag{3.42}$$

Having the uncertainty of the homography transformation ${}^{\pi}H_v$, the uncertainty for points to be mapped via this homography becomes:

$$\boldsymbol{\Sigma}'_{\pi_{\mathbf{x}}} = (\mathbf{I} \otimes {}^{v}\mathbf{x}^T)\, \boldsymbol{\Sigma}_{\pi_{\mathbf{h}_v}}\, (\mathbf{I} \otimes {}^{v}\mathbf{x}) \tag{3.43}$$

provided that the points ${}^{v}\mathbf{x}$ are certain.

3.4.2.2 Propagation of uncertainties on Euclidean inertial planes

Previously the uncertainty of the homography transformation ${}^{\pi}H_v$ was obtained by assuming that the points from virtual camera image plane are certain. Now we continue to take into account the uncertainties for these points and their propagations on the Euclidean inertial plane $\boldsymbol{\pi}$. In this case, the two covariance matrices, $\boldsymbol{\Sigma}_{\pi_{\mathbf{h}_v}}$ and $\boldsymbol{\Sigma}_{v_{\mathbf{x}}}$ get augmented [Heu04] and the uncertainty for a registered point on the Euclidean inertial plane, ${}^{\pi}\mathbf{x}$, becomes as following:

$$\boldsymbol{\Sigma}_{\pi_{\mathbf{x}}} = {}^{\pi}H_v\, \boldsymbol{\Sigma}_{v_{\mathbf{x}}}\, {}^{\pi}H_v^T + \boldsymbol{\Sigma}'_{\pi_{\mathbf{x}}} \tag{3.44}$$

3.4.3 Experiments

In this part we introduce some experiments which have been carried out in order to demonstrate the uncertainty values in different situations. In these experiments we simulate a set of IS-camera couples where the cameras have the following calibration matrix:

$$K = \begin{bmatrix} 150 & 0 & 250 \\ 0 & 150 & 250 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.45}$$

3.4.3.1 Analysing uncertainty in virtual camera's image plane

Some experiments are carried out for an exemplary IS-camera couple in order to analyse the uncertainty of the imaged points on the virtual image plane, when the homography transformation is obtained using the inertial sensor.

Fig. 3.11-a,b and c indicate the variation for the elements of the covariance matrix ($\mathbf{\Sigma}_{v_\mathbf{x}}.xx$, $\mathbf{\Sigma}_{v_\mathbf{x}}.yy$ and $\mathbf{\Sigma}_{v_\mathbf{x}}.xy$) for an exemplary pixel $[\ 800 \quad 200 \quad 1\ ]^T$ of the image plane of virtual camera. They are for a case where the homography matrix is obtained from the attached IS with the angles $roll = 0$, $pitch = \pi/4$ and $yaw = 0$. The standard deviation for the roll angle, is assumed zero ($\delta_r = 0$) and the elements of the covariance matrix for the mentioned pixel is plotted with respect to the variations to the values of standard deviations of the other two angles (δ_p and δ_y). As expected, the uncertainties of the point get increased with increasing uncertainties of the IS observation.

In another experiment, Fig. 3.11-d,e and f represent the uncertainty for all the points of the virtual image plane, where its dimension is considered as 500×500 pixels. In this case the IS's observation vector (Eq. (3.31)) is as following

$$\mathbf{s} = [\ \pi/4 \quad 0 \quad \pi/8\ ]^T \tag{3.46}$$

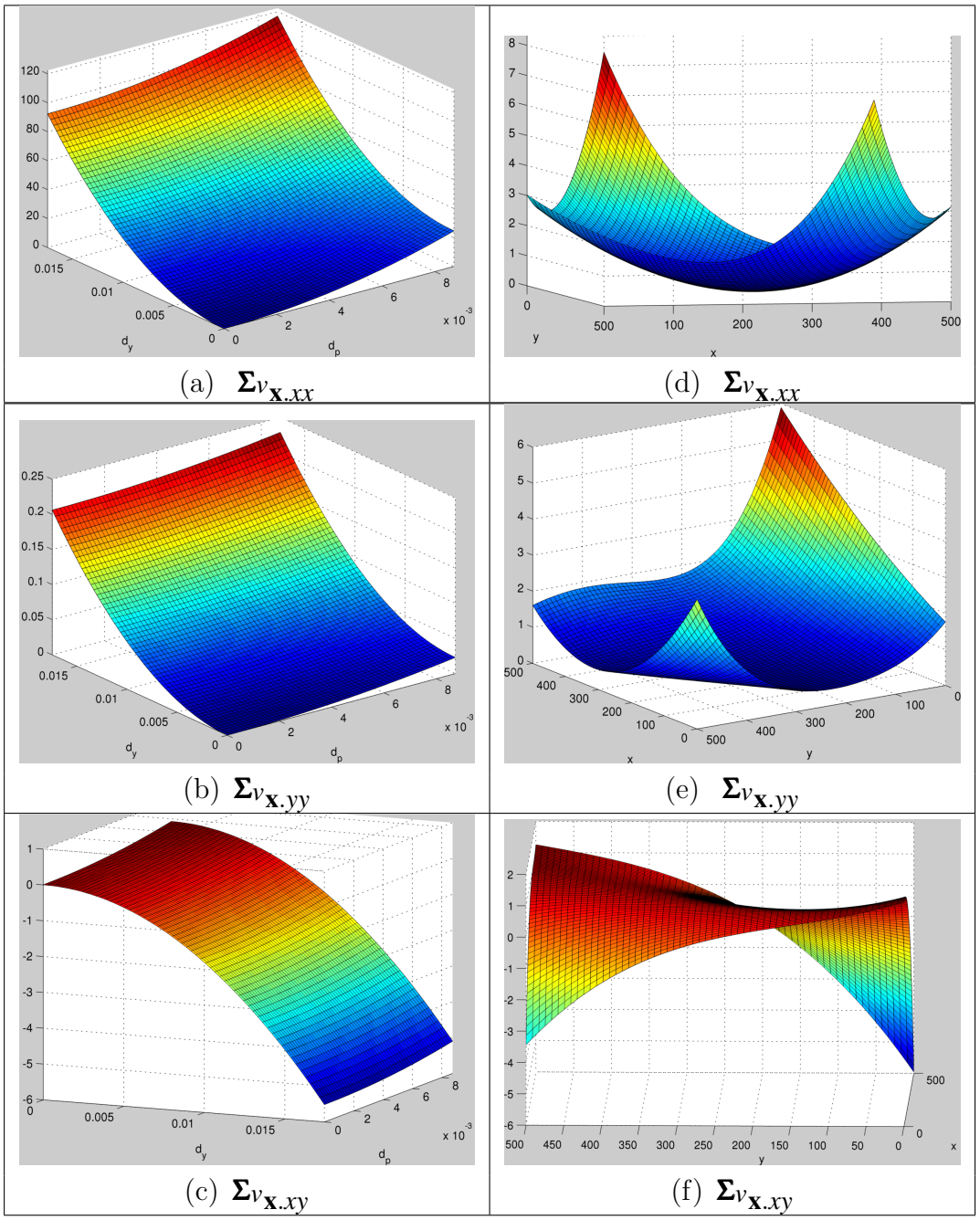


Figure 3.11: Plots for the elements of the covariance matrix of a virtual camera's image plane. (a), (b) and (c): Depict the covariance matrix's elements of for an exemplary pixel $[\ 800 \quad 200 \quad 1\ ]^T$. They correspond to a case where the homography matrix is obtained from the attached IS with the angles $roll = 0$, $pitch = \pi/4$ and $yaw = 0$. The standard deviation for the roll angle (δ_r) is assumed zero. The elements of the covariance matrix for the mentioned pixel ($\Sigma v_{\mathbf{x}}.xx$, $\Sigma v_{\mathbf{x}}.yy$ and $\Sigma v_{\mathbf{x}}.xy$) are plotted with respect to the variations to the value of standard deviations of the other two angles (δ_p and δ_y). (d),(e) and (f): The covariance matrix's elements for the different pixels of the virtual image plane. They correspond to a case where the homography matrix is obtained from the IS with the angles $roll = \pi/4$, $pitch = 0$ and $yaw = \pi/8$. The dimension of the image plane is assumed as 500×500 pixels. The covariance matrix of IS observation is considered as $\Sigma_{\mathbf{S}} = diag\{0.25, 0.25, 1.0\}$ in degrees.

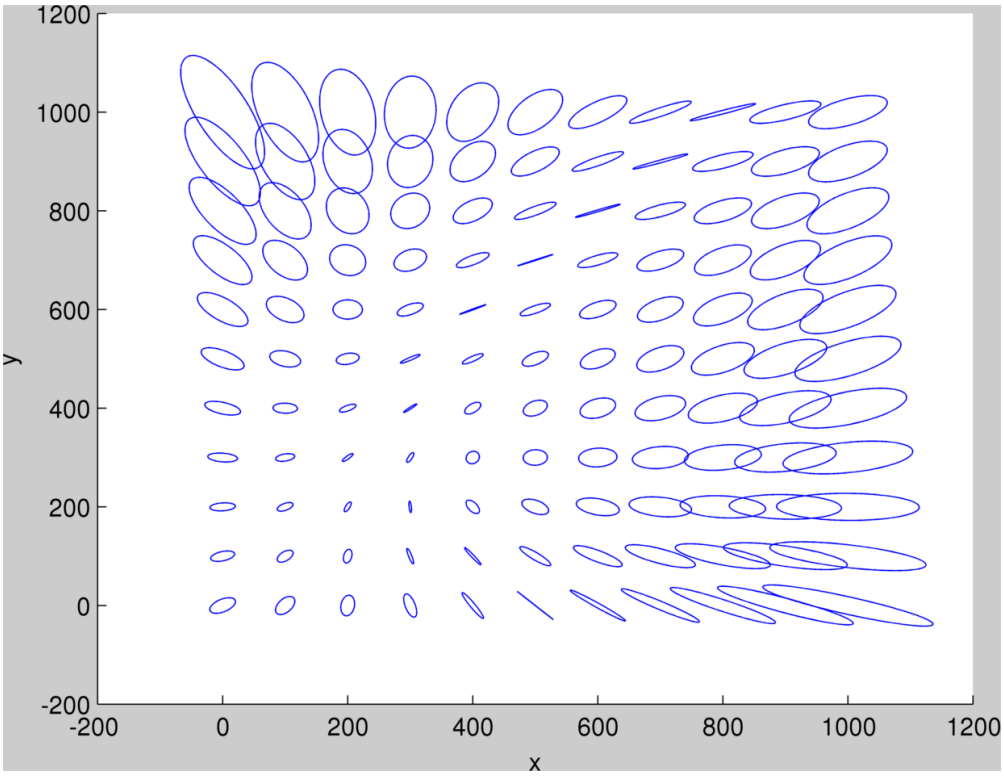


Figure 3.12: The covariance matrices, $\mathbf{\Sigma}_{v_\mathbf{x}}$, for different pixels of the virtual camera's image plane (related to Fig. 3.11-d,e and f) are demonstrated by ellipses, where they are scaled 100 times for clarity.

Based on [KHJG11] we assume the covariance matrix of IS observation as $\mathbf{\Sigma_S} = diag\{0.25, 0.25, 1.0\}$ in degrees. One can see that the uncertainties for the pixels close to the center of the image (principal point) is minimum and they increase in the other image coordinates with respect to the configuration. The same image uncertainties are shown by ellipses in Fig. 3.12 where the values are scaled 100 times for clarity.

The changes of uncertainties on the image plane of two other IS-camera couples have been analysed. Fig. 3.14 shows the progress of the uncertainties on an exemplary point, $\mathbf{x} = [\ 450 \quad 450 \quad 1\ ]^T$, on the virtual camera's image plane. The IS's observation is as following:

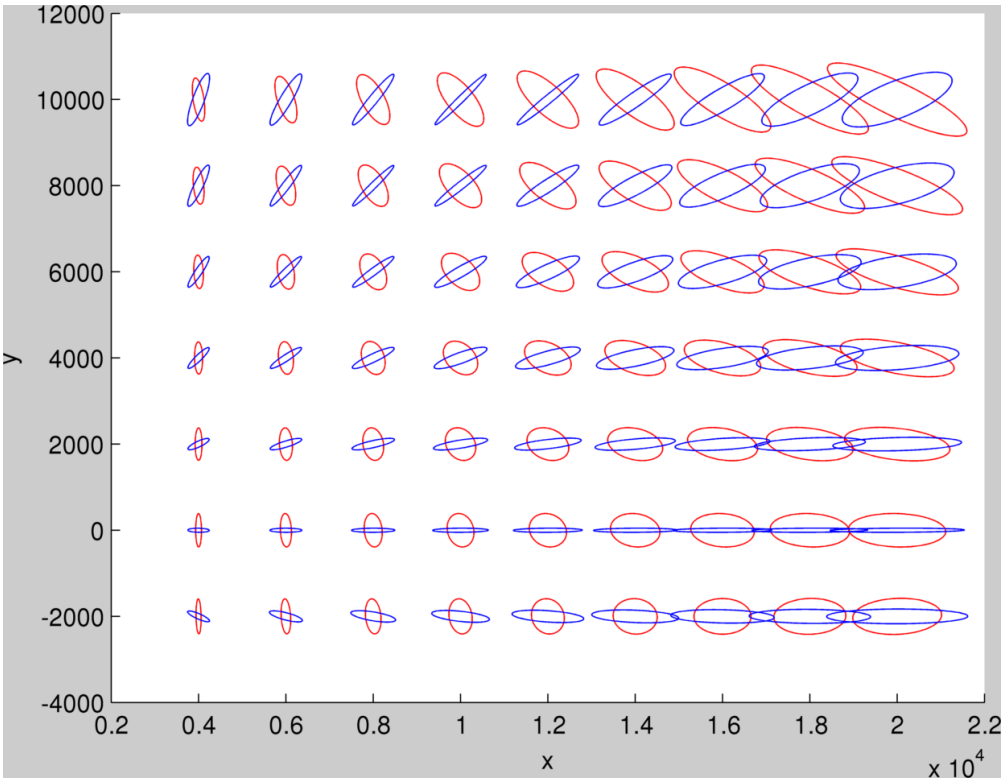


Figure 3.13: The covariance matrices, $\mathbf{\Sigma}_{\pi_\mathbf{x}}$, for different registered points on the Euclidean inertial plane, demonstrated by ellipses. The blue and red ellipses stand for points registered by the first and second camera, respectively. For the sake of clarity the covariance values are scaled 500 and 600 times, respectively for the first and second cameras.

$$\mathbf{s} = [\ 0 \quad \pi/4 \quad \pi/8\ ]^T \tag{3.47}$$

Six incremental covariance matrices for IS observation are considered, starting from $\mathbf{\Sigma_S} = diag(0,0,0)$ and finishing by $\mathbf{\Sigma_S} = diag(0.16^2, 0.16^2, 0.33^2)$. The uncertainty matrices for the particular point $\mathbf{x}$ of the virtual plane are shown by some ellipses in Fig. 3.14-lefts. Also the values for each element of the same covariance matrices are plotted in Fig. 3.14-rights. Similar experiment is carried out for the same IS-camera couple, where the IS observation is as following:

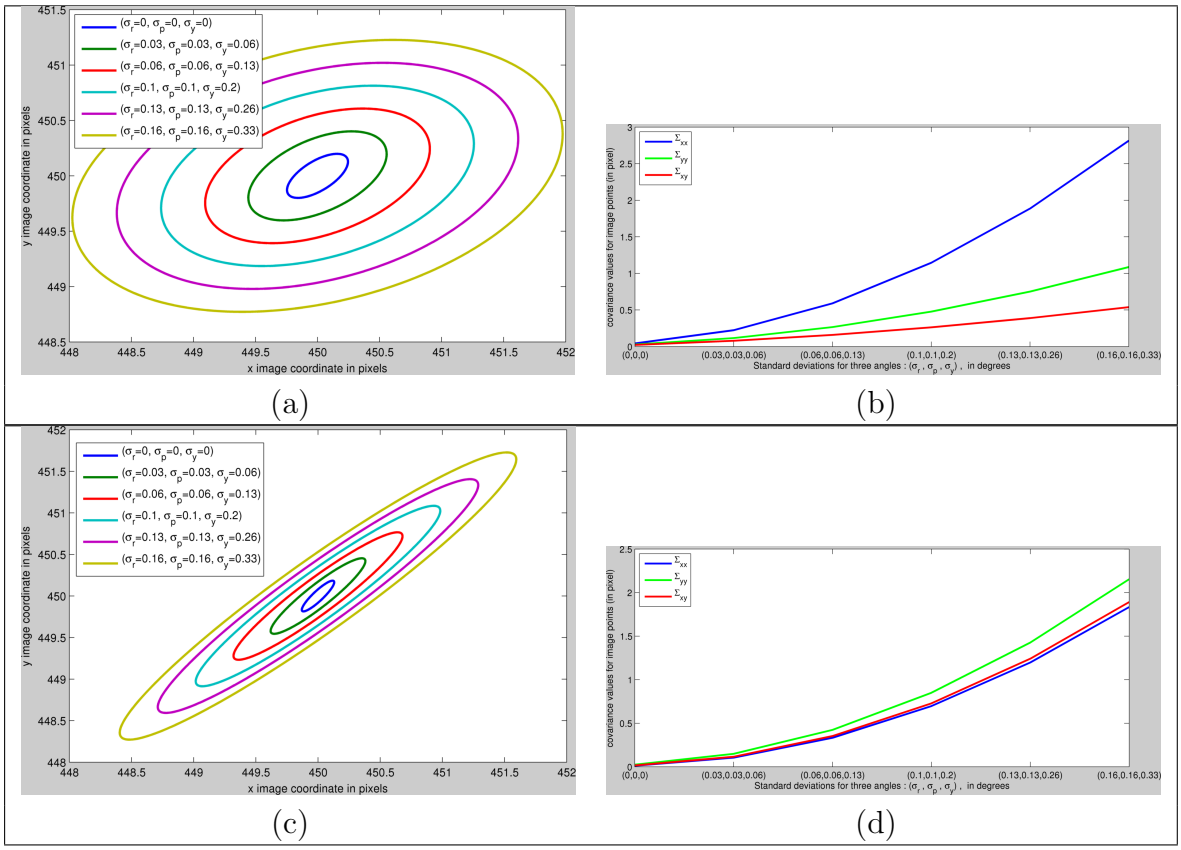


Figure 3.14: Uncertainties for an exemplary pixel, $\mathbf{x} = [\ 450 \quad 450 \quad 1\ ]^T$, of a virtual image plane. The first and second rows correspond to two different values of IS observation: $\mathbf{s} = [\ 0 \quad \pi/4 \quad \pi/8\ ]^T$ and $\mathbf{s} = [\ \pi/2 \quad -\pi/2 \quad 0\ ]^T$, respectively. For each of these two cases, the pixel uncertainties related to different noise level (of IS) are shown.

$$\mathbf{s} = [\ \pi/2 \quad -\pi/2 \quad 0\ ]^T \tag{3.48}$$

As can be seen, the uncertainties of the mapped point increase with increasing the uncertainties in IS observation.

3.4.3.2 Analysing uncertainty in Euclidean inertial plane

Another experiment is performed to analyse the uncertainties of the points registered on an Euclidean inertial plane. Two couples of IS-camera are used. The camera calibration matrices are as previously defined. The translations for first camera and second camera respectively are:

$$\mathbf{t}_1 = [\ 100 \quad -100 \quad 500\ ]^T \tag{3.49}$$

and

$$\mathbf{t}_2 = [\ -1500 \quad 7000 \quad 6500\ ]^T \tag{3.50}$$

in *mm*. The uncertainty covariances for these two translation vectors are assumed as $\mathbf{\Sigma}_{\mathbf{t}_1} = diag\{200^2, 300^2, 500^2\}$ and $\mathbf{\Sigma}_{\mathbf{t}_2} = diag\{100^2, 500^2, 300^2\}$ again in *mm*. The observation vectors for the first and second inertial sensors are respectively as following

$$\mathbf{s}_1 = [\ \pi/4 \quad 0 \quad \pi/8\ ]^T \tag{3.51}$$

and

$$\mathbf{s}_2 = [\ \pi/4 \quad \pi/2 \quad 0\ ]^T \tag{3.52}$$

where the covariance matrices for both IS_1 and IS_2 are assumed as $\mathbf{\Sigma}_{\mathbf{s}_1} = \mathbf{\Sigma}_{\mathbf{s}_2} = diag\{0.25, 0.25, 1.0\}$ in degrees [KHJG11].

For this experiment, the uncertainty values for the registered points on the Euclidean plane are calculated by using Eq. (3.44). The obtained uncertainties are presented by covariance ellipses in Fig. 3.13. The blue ellipses are for points which are mapped though the first IS-camera couple and the red ones are for the

point mapped through the second couple. The covariance ellipses for the first and second cameras are respectively scaled 500 and 600 times for clarity. The Euclidean plane is also scaled to 10^4 times and is shown in *mm*.

3.5 Conclusion

In this chapter the geometric relations among different projective and Euclidean virtual planes involved in the framework have been more specifically explored. A set of mathematical equations was obtained which are able to generate the homography matrix among two inertial planes for different cases. Majority of the obtained equations express the relation between two Euclidean planes without an explicit involving of intrinsic parameters of the cameras. They show that the point mapped onto only one of the inertial planes (particularity π_{ref}) is sufficient to propagate the mapping on the other Euclidean planes which are parallel to the reference plane, independent of involving the camera intrinsic parameters (as expected). Using the obtained models, an alternative version of the 3D data registration algorithm was introduced which recursively registers the data on the virtual planes.

Translation among two virtual cameras is one of the prerequisite for many computer vision applications including the proposed data registration framework. Thus we took the advantage of having an IS coupled to each camera and proposed a method to estimate the translation vectors among the cameras within the network. A set of experiments to evaluate the quality of the translation estimation method was performed.

The uncertainties of the involved homography transformations in the framework and their propagation of errors onto the registered data have been modelled using statistical geometric analysis. The obtained achievement can be of importance for fusion of information which are coming from different nodes in a sensor network.

Chapter 4

Real-time implementation using GPU-CUDA

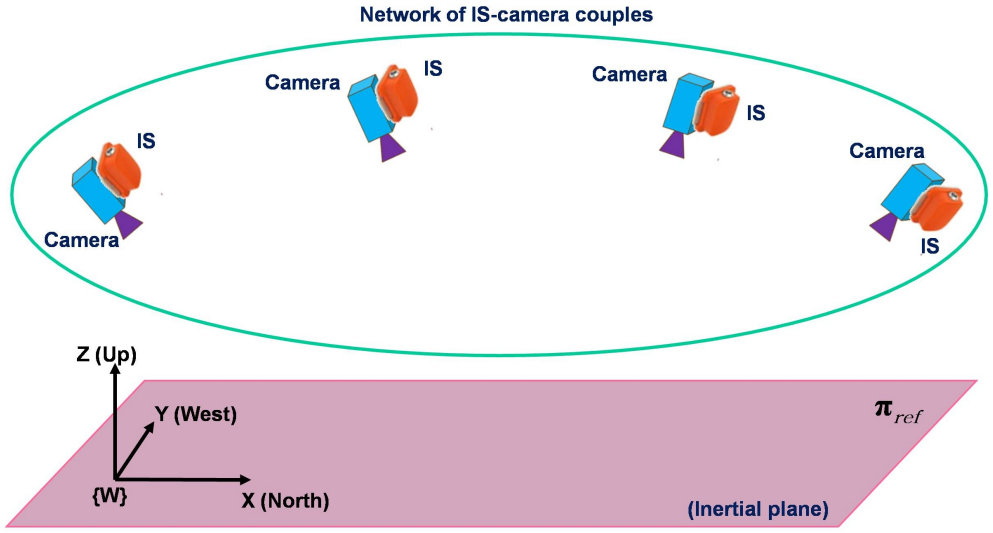


Figure 4.1: A distributed network of cameras and inertial sensors. An inertial Euclidean plane, π_{ref}, is defined as a virtual reference plane in the scene.

4.1 Introduction

This chapter presents a full-body volumetric reconstruction of a person (or object) in a scene using a sensor network. The concept used in chapter is based on the previous chapters, however a brief summarization is presented here. The sensor network is comprised of couples of camera and inertial sensor (IS), as seen in Fig. 4.1. Taking advantage of IS, the 3D reconstruction is performed using no planar ground assumption. Moreover, IS in each couple is used to define a virtual camera whose projective image plane is horizontal and aligned with the earth cardinal directions (see Fig. 4.2). The IS is furthermore used to define a set of Euclidean inertial planes in the scene. The image plane of each virtual camera is projected onto this set of parallel-horizontal inertial-planes, using some adapted homography functions. A parallel processing architecture is proposed in order to perform human real-time volumetric reconstruction. The real-time characteristic is obtained by implementing the reconstruction algorithm on a general purpose processing unit (GP-GPU) using Compute Unified Device Architecture (CUDA). In order to show the effectiveness of the proposed algorithm, a variety of human

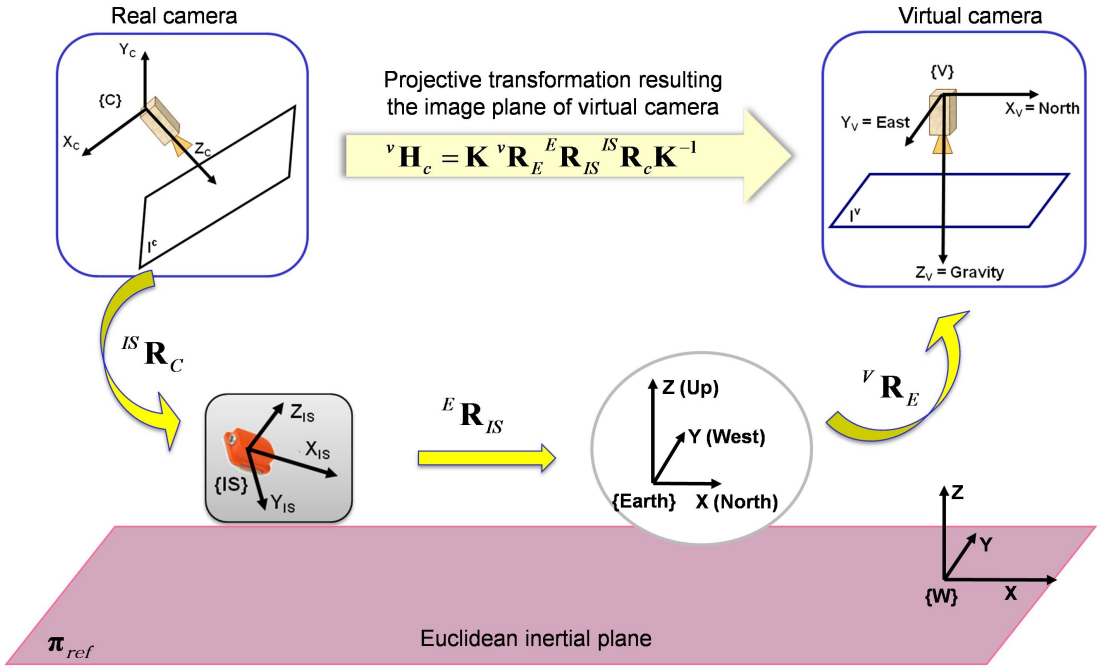


Figure 4.2: Schematic of a virtual camera: A virtual camera is created from a IS-camera pair by using infinite homography. Different coordinate systems are involved in this definition. $\{Earth\}$: Earth cardinal coordinate system, $\{IS\}$: Inertial reference frame expressed in $\{Earth\}$, $\{W\}$: world reference frame of the framework, $\{C\}$: camera reference frame, $\{V\}$: reference frame of the virtual camera corresponding to $\{C\}$. Based on the definition, the virtual camera has a horizontal image plane (projective), parallel to the Euclidean inertial plane π_{ref}.

postures and some objects in the scene are reconstructed and demonstrated. Some analysis have been carried out to measure the performance of the algorithm in terms of processing time in different configurations.

4.2 Parallel processing using GPU

CUDA and GPU Hardware Architecture

In CUDA terminology, the GPU is called the device and the CPU is called the host (see Fig. 4.4). A CUDA device consists of a set of multi-core processors.

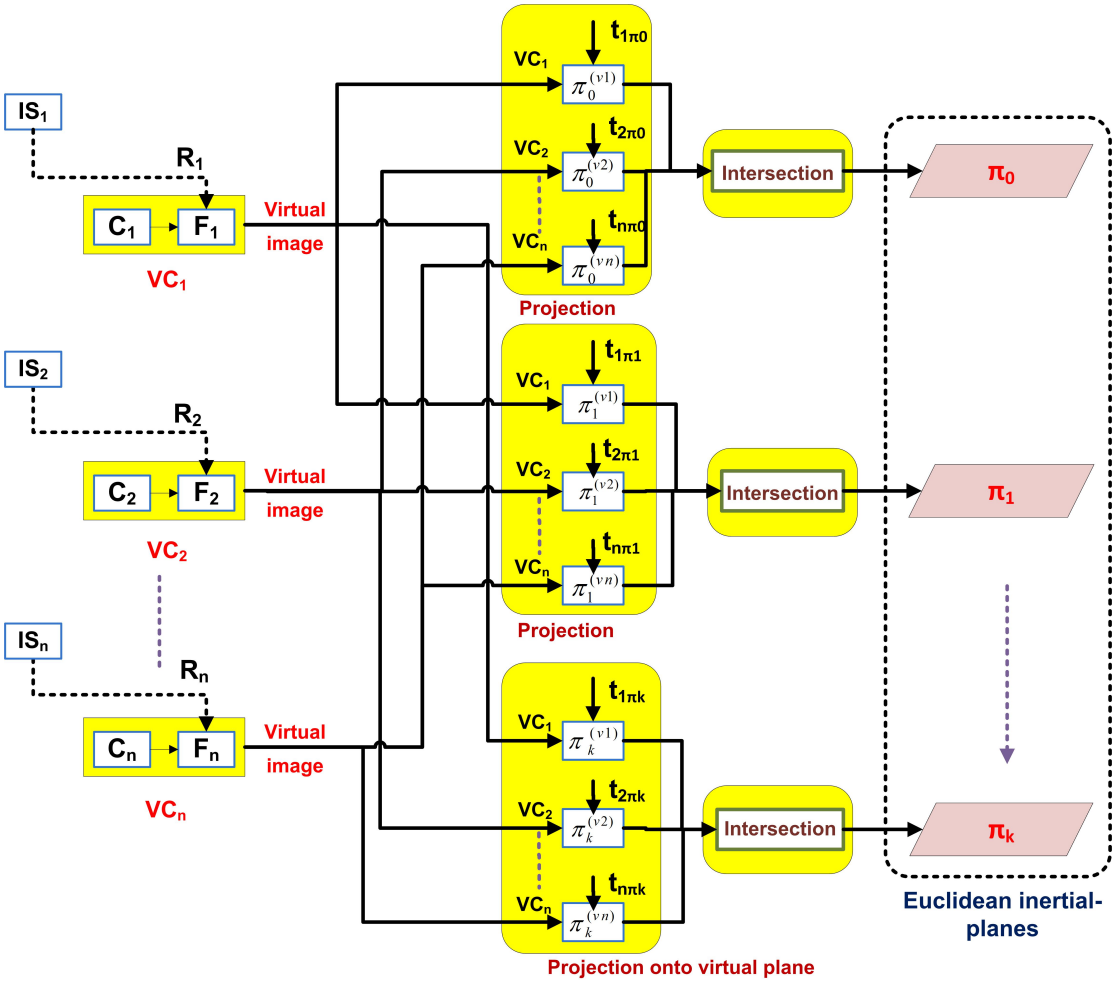


Figure 4.3: The architecture corresponding to the proposed algorithm. First in each IS-camera couple the 3D rotations provided by the IS is fused with the camera image to create a horizontal virtual image plane. The projective image planes get projected onto different Euclidean inertial planes in the scene. By performing the intersection of the projected silhouettes the registration on each inertial plane is obtained. The parts coloured in yellow are implemented on GP-GPU.

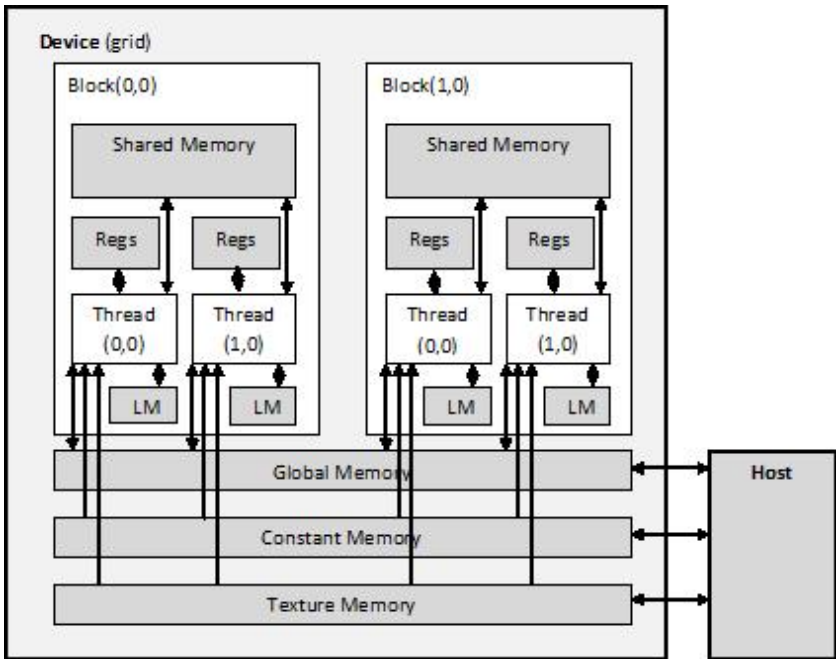


Figure 4.4: CUDA architecture [AMD11].

Each multicore processor is simply referred to as a multiprocessor. Cores of a multiprocessor work in a single instruction, multiple data (SIMD) fashion. All multiprocessors have access to three common memory spaces (globally referred to as device memory but with different access time). The CUDA program is organized into a host program, consisting of one sequential thread running on the host CPU, and several parallel kernels executed on the parallel processing device (GPU). A kernel executes a scalar sequential program on a set of parallel threads. The program organizes these threads into a grid of thread blocks.

3D reconstruction using GPU-CUDA

Normally a full-body volumetric reconstruction of human is time consuming due to the huge amount of data to be processed. In order to have a real-time processing (which is necessary for many applications) we propose a parallelizing of the 3D reconstruction algorithm. The previously proposed 3D volumetric reconstruction

approach is adapted for this implementation and described as an algorithm in Alg. 4. First, the image plane of each virtual camera is obtained. Then the image of each virtual camera is projected onto a set of inertial-planes. N_c and N_π indicate the number of cameras and number of inertial-planes, respectively. I_{c_i} and I_{v_i} respectively are the image plane of camera C_i and its corresponding virtual camera V_i. Δh is the Euclidean distance among inertial-planes which also can be interpreted as the vertical resolution of the algorithm. The labels 'Gpu_Warping', 'Gpu_Project2VirtualPlane' and 'Gpu_Plane_Intersection' correspond to the labels in the flow-chart of Fig. 4.6.

Algorithm 4: Algorithm of 3D data registration using inertial-planes: First, the image plane of each virtual camera is obtained. Note that the background-subtracted images are binary. Then the image of each virtual camera is projected onto a set of Euclidean inertial-planes. N_c and N_π indicate the number of cameras and number of inertial-planes, respectively. I_{c_i} and I_{v_i} respectively are the image planes of camera C_i and its corresponding virtual camera V_i. Δh is the Euclidean distance among the inertial-planes which also can be interpreted as the vertical resolution of the algorithm. (The labels 'Gpu_Warping', 'Gpu_Project2VirtualPlane' and 'Gpu_Plane_Intersection' correspond to the labels in flowchart of Fig. 4.6). At the end the algorithm returns a set of inertial planes with data registered over them.

/* generating image planes of virtual cameras */

for $i \leftarrow 1$ to N_c do

"Gpu_Warping"

$^{v_i}H_{c_i} \leftarrow K_i\, {}^{v_i}R_{c_i}\, K_i^{-1}$

$I_{v_i} \leftarrow {}^{v_i}H_{c_i}\, I_{c_i}$

$^{v_i}H_{\pi_{ref}} \leftarrow K_{c_i}[\ \hat{\mathbf{i}} \quad -\hat{\mathbf{j}} \quad \mathbf{t}_{c_i}\ ]$ /* Eq. (2.17) */

/* projecting virtual images onto inertial-planes */ $h \leftarrow 0$

for $j \leftarrow 0$ to $N_\pi - 1$ do

for $i \leftarrow 1$ to N_c do

"Gpu_Project2VirtualPlane"

$^{\pi_h}H_{v_i} \leftarrow inv(^{v_i}H_{\pi_{ref}} + hP_i\hat{\mathbf{k}}^T)$ /* using Eq. (3.2) */

$\pi_h^{(v_i)} \leftarrow {}^{\pi_h}H_{v_i}\, I_{v_i}$

$h \leftarrow h + \Delta h$

/* obtaining intersection of the projected virtual images for each inertial plane */

for $j \leftarrow 0$ to $N_\pi - 1$ do

"Gpu_Plane_Intersection"

/* cell-wise binary AND. Please see Fig. 4.5 */

$\pi_j \leftarrow \prod_{i=1}^{N_c} \pi_h^{(v_i)}$

return $\{\pi_0, \pi_1 \cdots \pi_{(N_\pi - 1)}\}$

For each Euclidean inertial plane in the framework a set of temporary planes (also Euclidean) is considered. For instance for π_h (the inertial plane at the height h) a set of temporary planes $\{\pi_h^{(v1)}, \pi_h^{(v2)} \cdots \pi_h^{(vN_c)}\}$ is defined. $\pi_h^{(v_i)}$ in-

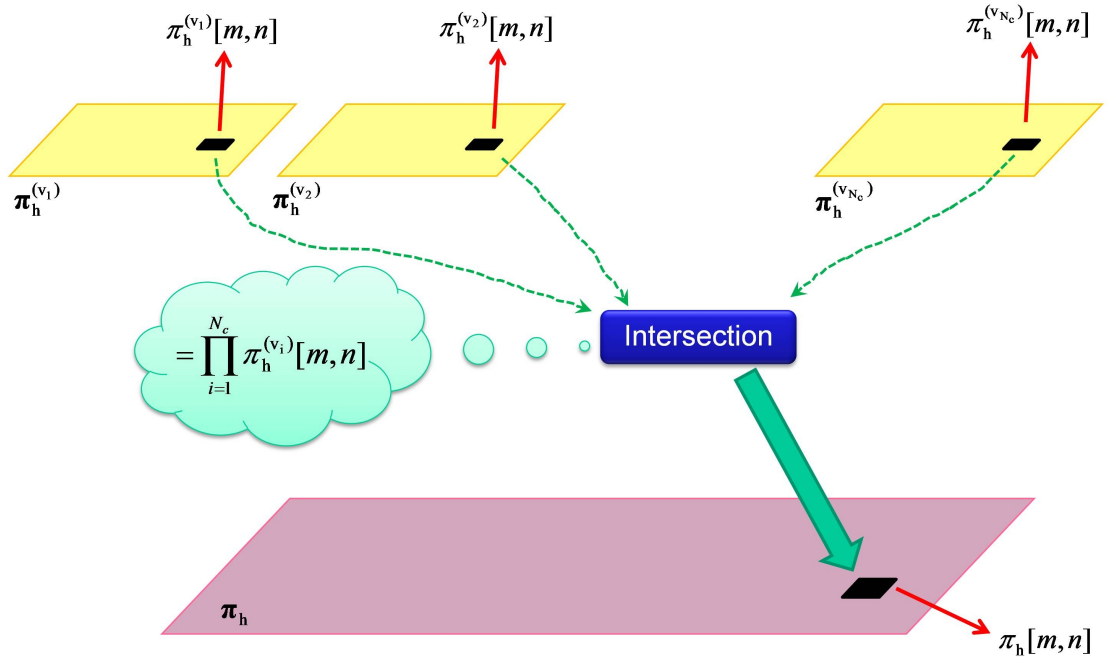


Figure 4.5: Cell-wise intersection of the projections of the virtual images onto an exemplary inertial-plane π_h: Firstly the images of all virtual cameras get projected onto a temporary inertial plane. $\pi_h^{(v_i)}$ indicates the temporary inertial-plane corresponding to the virtual camera V_i. Then the corresponding cells of all temporary inertial-planes are fused using an AND operator in order to provide the final registration on the inertial-plane π_h. (m and n indicate the indices of a cell). Note that the silhouettes are considered as binary.

dicates the temporary inertial-plane corresponding to the virtual camera V_i. The images planes of each virtual camera V_i initially gets projected onto $\pi_h^{(v_i)}$, for the inertial plane π_h (see Fig. 4.5). Then the corresponding cells of all temporary planes (belonging to the same inertial plane) are fused using an AND operator in order to provide the final registration on the inertial-plane π_h. Note that the images are considered as binary. This part of the algorithm is labelled as 'Gpu_Plane_Intersection' and illustrated in Fig. 4.5. In this figure, m and n indicate the indices of a cell.

Fig. 4.3 depicts an architectural view corresponding to the algorithm. The parts colored in yellow are implemented on CUDA. Fig. 4.6 demonstrates the

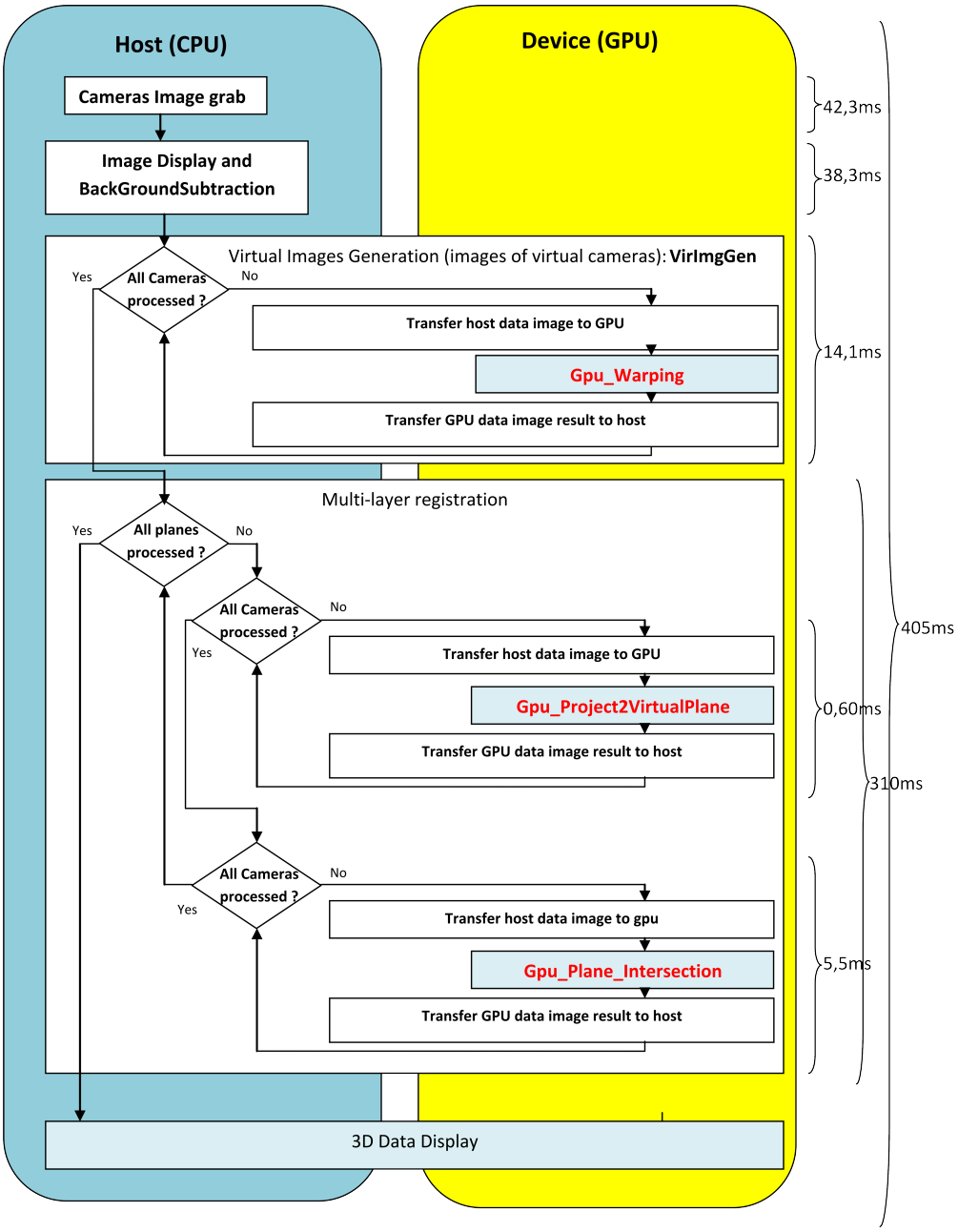


Figure 4.6: Flowchart of CUDA implementation of the proposed inertial-based 3D reconstruction. The left block (coloured in aqua) is the processes which are executed in CPU in a traditional serial fashion. The right block (coloured in yellow) indicates the processes which are implemented on GP-GPU using parallel processing. The corresponding algorithm is presented in Alg. 4.

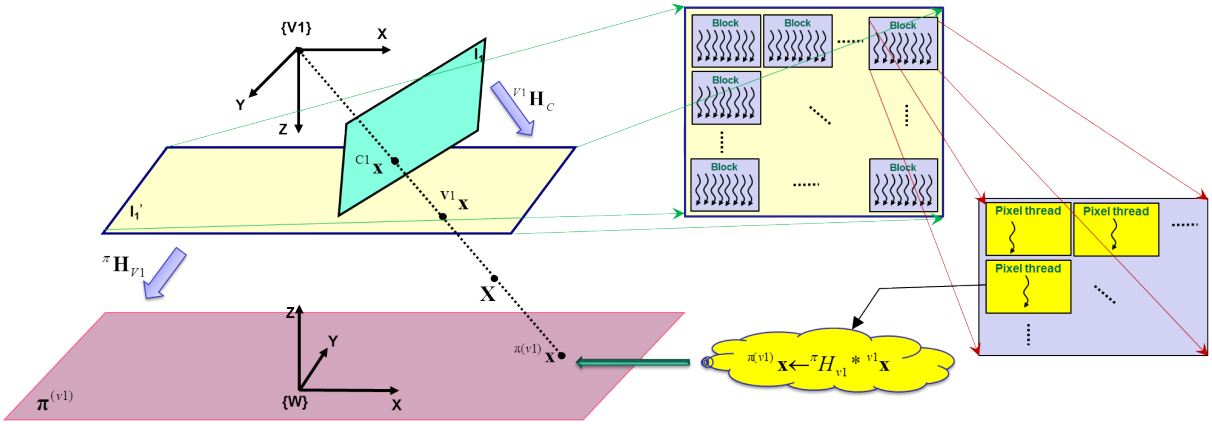


Figure 4.7: CUDA implementation of inertial plane projection. The process labelled by GPU_Project2VirtualPlane in Fig. 4.6 and Alg. 4 is performed on CUDA. $V1$ denotes a virtual camera and I' is its corresponding virtual image. The homography transformation ${}^{\pi}H_{v1}$ is applied on each pixel thread (of I'), independently. The results are stored on a temporary inertial plane (see Fig. 4.5).

flowchart of the parallel implementation using CUDA. In the beginning the images are grabbed and then the silhouettes are extracted. After that the silhouettes are loaded on the GPU memory in order to be processed by CUDA. The loaded images on GPU memory are warped to generate the images of virtual cameras (labelled as VirImgGen). After having the images of the virtual cameras generated, the images are projected on different inertial-planes in order to register the 3D data on them (labelled as GPU_Project2VirtualPlane). Once images of all cameras get projected onto the inertial-planes, a pixel-wise AND operator is applied to them in order to obtain the intersections (labelled as Gpu_Plane_Intersection). In this point the 3D volumetric reconstruction has been obtained. Eventually the registered data are passed to a visualizer to show the result.

In Fig. 4.6 and Alg. 4, the processes labelled by VirImgGen, GPU_Project2VirtualPlane and Gpu_Plane_Intersection are the parts which are performed on CUDA using a parallel implementation. As shown in Fig. 4.7, the virtual image plane is divided into a set of blocks. Each block has a set of pixel threads which run in parallel. I' is the image plane of virtual camera $V1$. The homography transformation ${}^{\pi}H_{v1}$ is applied on each pixel thread (of I'), independently. The results are stored on a

temporary inertial plane, as described in Fig. 4.5. I', the image plane of virtual camera, was obtained using a similar parallelization process.

As described in Fig. 4.5, the intersections of the obtained temporary inertial planes need to be obtained by applying an 'AND' operation. This part, which corresponds to the process labelled by Gpu_Plane_Intersection in Fig. 4.6 and Alg. 4, is also implemented on CUDA. As shown in Fig. 4.8, the 'AND' operation for each pixel thread is performed in parallel. It is for an exemplary case where there are two virtual cameras. The result is the Euclidean inertial plane with the data registered on.

4.3 Experiments

4.3.1 Infrastructure

Fig. 4.9 shows the smart-room of the laboratory of mobile robotic in the University of Coimbra [MRL], used in our experiments. The superimposed area in this figure is observed by a camera network. The cameras are AVT Prosilica GC650C GigE Color [Proa], synchronized by hardware. Each camera is rigidly coupled with an IS (we used Xsens MTx [xse]). Fig. 4.10 depicts an exemplary IS-camera couple. The purpose of using IS is to have 3D orientation with respect to earth, obtain virtual camera and define virtual horizontal planes. First the intrinsic parameters of the cameras are estimated using Bouguet Camera Calibration Toolbox [Bou03] and then Camera Inertial Calibration Toolbox [LD07] is used for the sake of extrinsic calibration between the camera and IS (to estimate ${}^{C}R_{IS}$). After acquiring image from each camera, a color-based background subtraction step is performed. The human silhouette is separated from the background through color segmentation using the HSV (hue, saturation, value) model. This model is less sensible to illumination changing conditions [KMB07, Bra98]. A 1-D Hue histogram is sampled from the human area and stored for future use. During frame acquisition, the stored color histogram is used as a model, or look-up table, to convert incoming

video pixels to a corresponding probability of body image. Using this method, probabilities range in discrete steps from zero (0.0) to the maximum probability pixel (1.0). Later it is multiplied by a binary mask.

The reconstruction algorithm is developed using the C++ language, OpenCV library [Ope] and NVIDIA's CUDA software [Nvi] for Ubuntu Linux v10.10. The visualization is carried out using openGL library. The processing unit responsible for all the sensory and vision algorithm (including CUDA processing) is composed by a PC (Intel Core2 Quad processor Q9400, 6 MB Cache, 4 GB RAM, 1333 MHz and a PCI-Express NVIDIA GeForce 9800 GTX+).

4.3.2 Reconstruction results

Different sets of experiments have been carried out using the proposed inertial-based 3D reconstruction method by a GPU-based implementation. 16 samples are demonstrated in Fig. 4.12 and 4.13 where an acting person is fully reconstructed in 3D. One of the samples is separately shown in Fig. 4.11 in order to have a more detailed view. In these examples, 48 Euclidean inertial-planes are used for the purpose of 3D data registration. The interval distance among two consecutive inertial-plane is $5\,cm$. Although the area of the scene in these experiments is small however in the computation the area is considered as $384 \times 384\,cm^2$ which is relatively large. Using a parallel implementation of the algorithm (using GPU), we managed to have a frequency close to $2.5\,Hz$ for reconstruction of the mentioned area (using the hardware stated in sub-section 4.3.1). The number of layers and their intervals can be adjusted depending to the need of an application and available hardware.

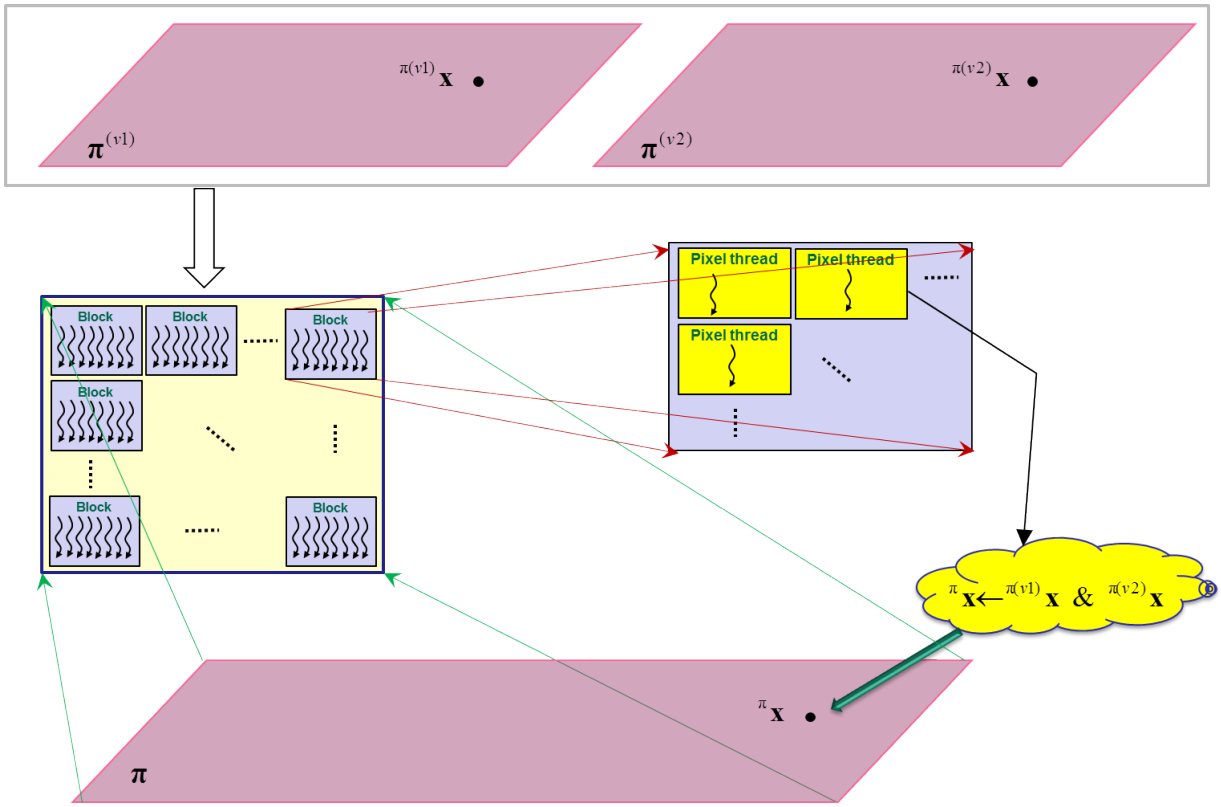


Figure 4.8: CUDA implementation of temporary inertial planes intersection. The process labelled by Gpu_Plane_Intersection in Fig. 4.6 and Alg. 4 is performed on CUDA to be processed in parallel. $\pi^{(v1)}$ and $\pi^{(v2)}$ are two temporary inertial planes which are obtained through using a parallel implementation described in Fig. 4.7. Each 'AND' operation is independently applied on a pair of corresponding points, from two temporary planes. The results are the final data registration for the Euclidean plane π.

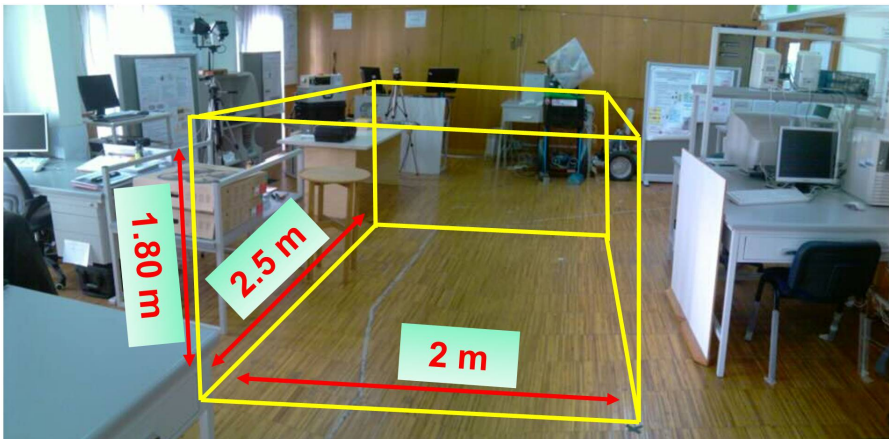


Figure 4.9: The scene used in the 3D reconstruction experiments. The superimposed area indicates where all cameras have overlap in their field of view.

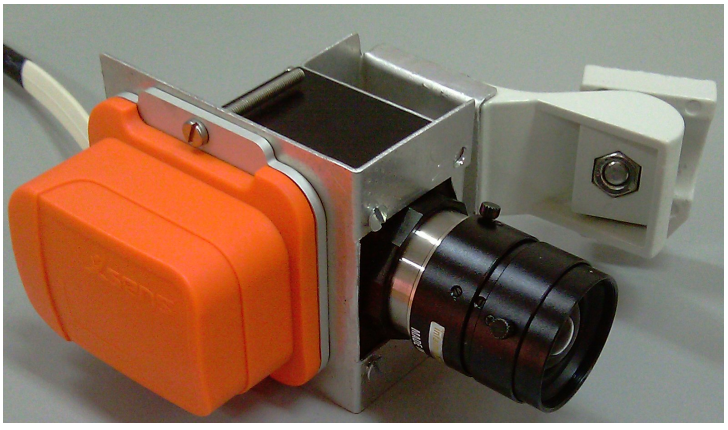


Figure 4.10: The IS-camera couple used in the real-time 3D reconstruction experiment.

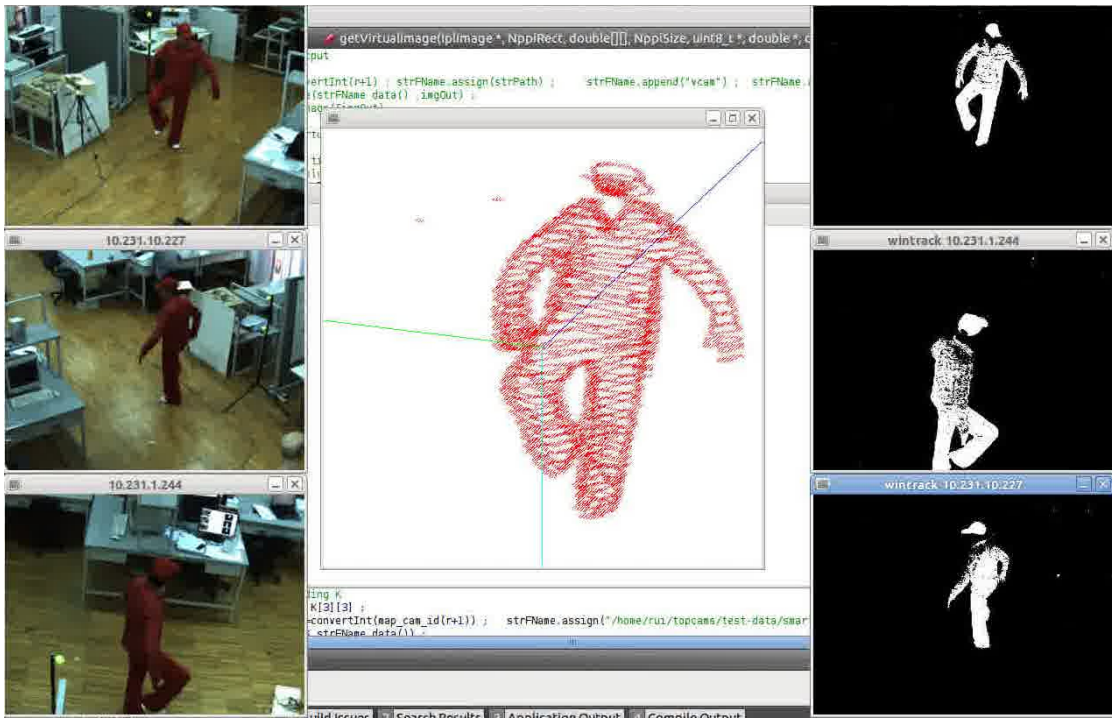


Figure 4.11: Results of 3D volumetric reconstruction using the proposed framework: The camera images before and after background subtraction (silhouette) are respectively shown in the left and right columns. The result of volumetric reconstruction using the silhouettes is illustrated in the middle. A network of IS-camera is used to observe the scene. 48 inertial-planes are used to register 3D data from the scene. The interval distance among two consecutive inertial-plane is $5\,cm$.

In order to demonstrate the applicability of the proposed framework for some other applications such as scene understanding a set of experiments have been carried out on some objects (see Fig. 4.14). In Fig. 4.14-(a) a semi rectangular blue box object is reconstructed. Fig. 4.14-(b) demonstrates a case where an cylindrical object, placed on the top of the box, is reconstructed. A chair which is partially covered in red is registered (the red part) in Fig. 4.14-(c). Fig. 4.15-(a) and Fig. 4.15-(b) show the result for a scene including a person and a mannequin. The person seated on a chair is reconstructed and shown in Fig. 4.15-(c).

4.3.2.1 Statistical analysis on the processing times

Some performance statistics are carried out in order to show the time which each part of the algorithm takes to run. In Fig. 4.6, processing time for each part of the algorithm is imprinted. The times refer to the case where 48 inertial-planes, each one having a size of $384 \times 384\,cm^2$, have been used. The infrastructure and hardware are as stated in sub-section 4.3.1. The total processing time for a full 3D reconstruction is $405\,ms$ which leads to have a frequency close to $2.5Hz$.

Fig. 4.16 depicts the average processing time in ms for different size of inertial-planes (the scale is $10^4\,cm^2$). The number of inertial-planes in this experiments is a constant equal to 48. The blue line demonstrates the processing time for generating the images of virtual cameras. Since the number of cameras are fixed in all tests, the execution time for that is almost constant. The red line indicates a part where images of all virtual cameras get projected onto a set of inertial-planes. Eventually the total processing time is shown in green color. As it is visible in the diagram, the processing time has a linear proportion related to size (area) of inertial-planes.

Another diagram showing the processing time versus number of inertial-planes is shown in Fig. 4.17. The size of inertial-planes (they are equal in the sizes) is considered as a constant equal to $384 \times 384\,cm^2$. Similar to Fig. 4.16, the colors

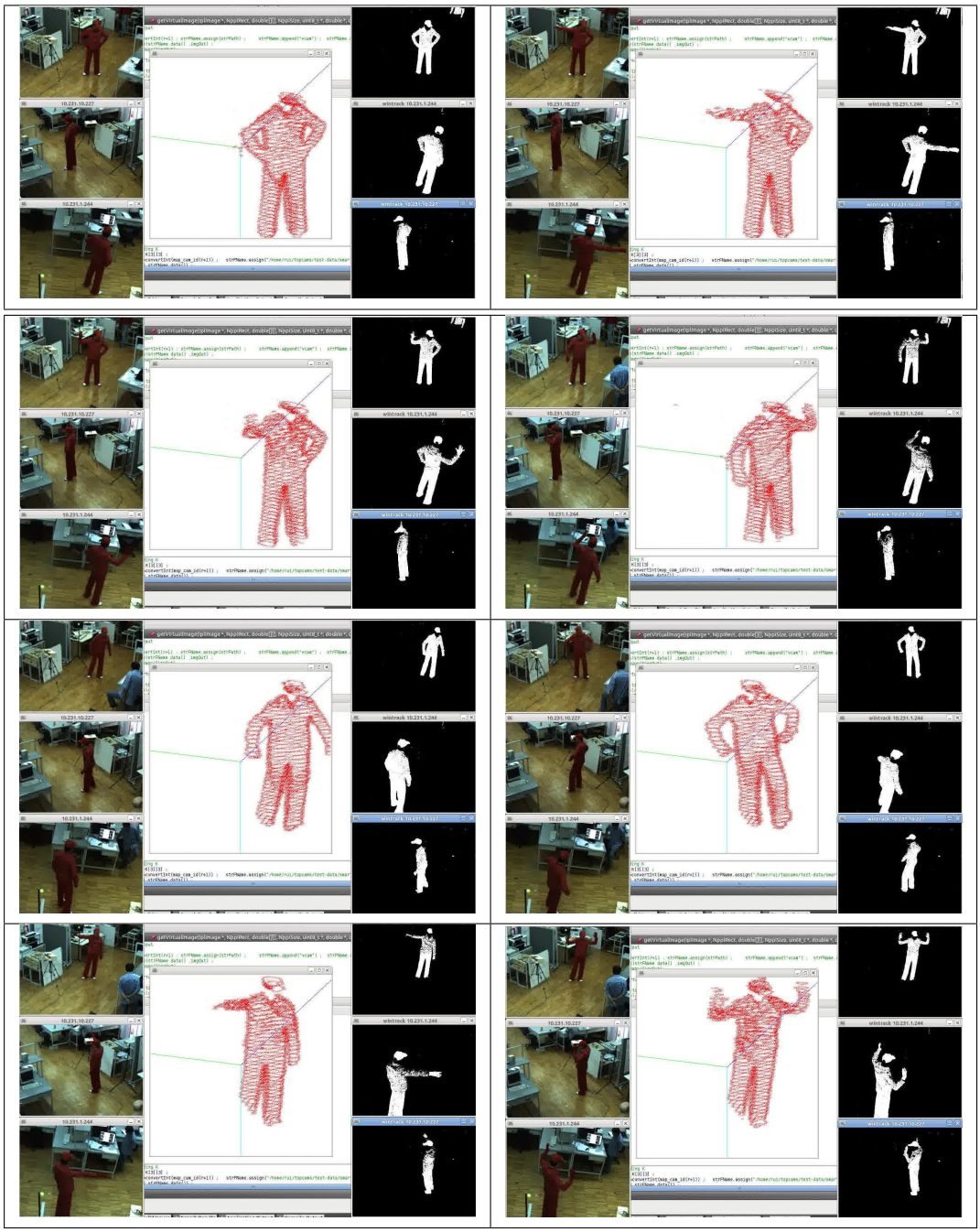


Figure 4.12: Results of 3D volumetric reconstruction using the proposed framework: 12 samples have been illustrated. In each sample, the camera images before and after background subtraction (silhouette) are respectively shown in the left and right columns. The result of volumetric reconstruction using the silhouettes is illustrated in the middle column for each sample. A network of IS-camera is used to observe the scene. 48 inertial-planes are used to register 3D data from the scene. The interval distance among two consecutive inertial-plane is $50mm$.

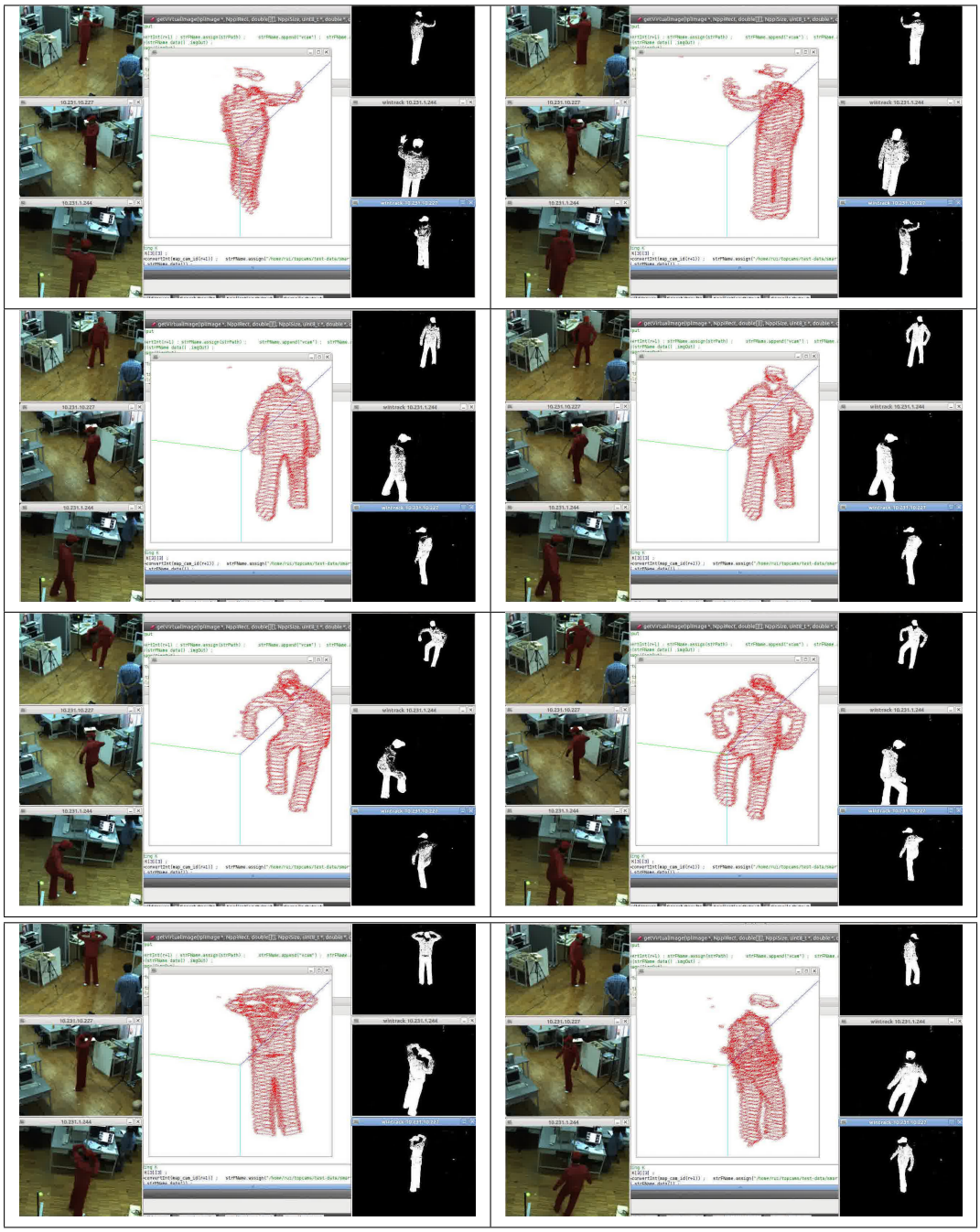


Figure 4.13: Results of 3D volumetric reconstruction using the proposed framework: 12 samples have been illustrated. In each sample, the camera images before and after background subtraction (silhouette) are respectively shown in the left and right columns. The result of volumetric reconstruction using the silhouettes is illustrated in the middle column for each sample. A network of IS-camera is used to observe the scene. 48 inertial-planes are used to register 3D data from the scene. The interval distance among two consecutive inertial-plane is $50mm$.

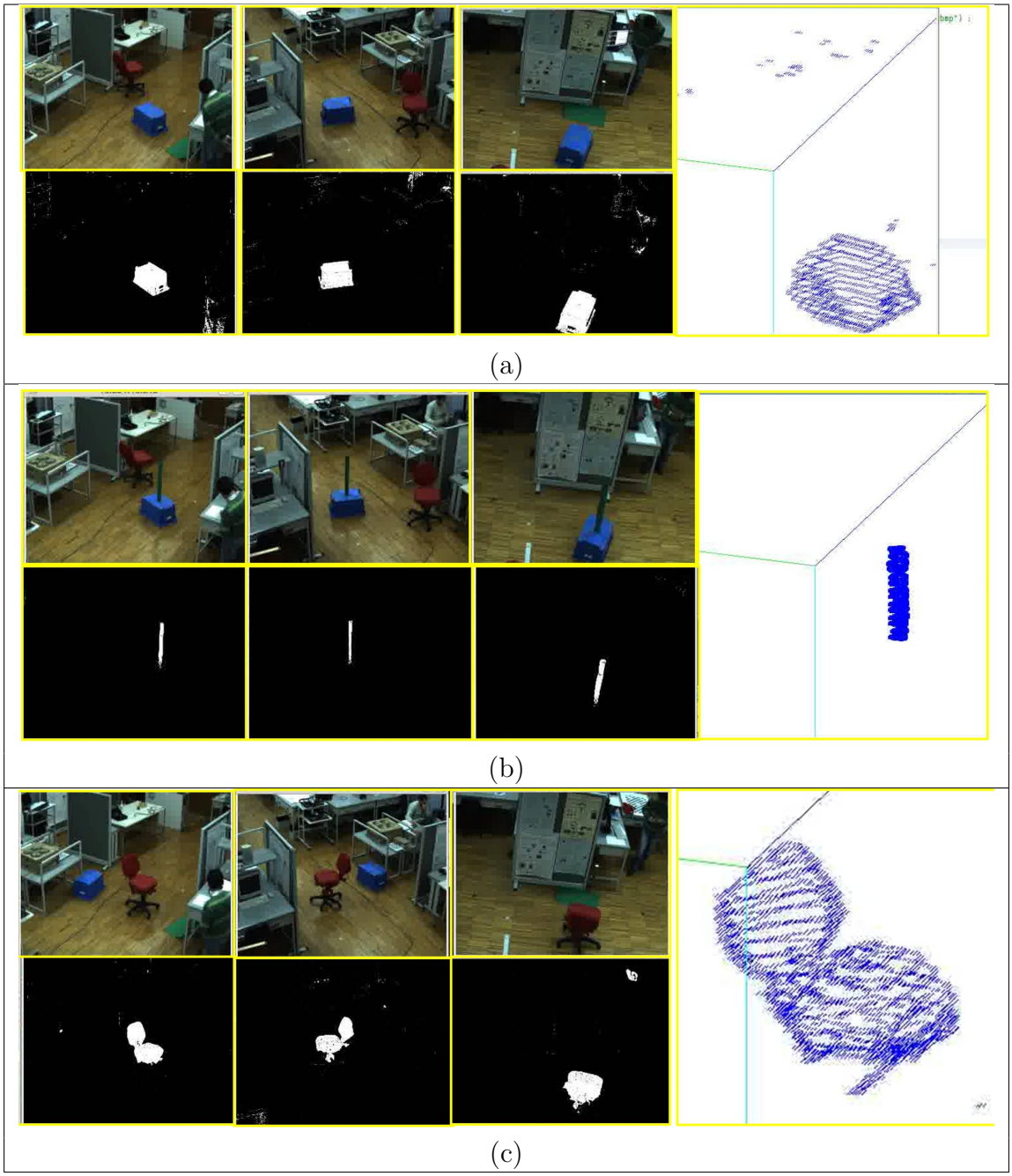


Figure 4.14: Results of the proposed multi-layer 3D data registration: Three experiments, each one for an object, are carried out. (a) represents the result for a semi-rectangular blue box. (b) depict the result for a small cylindrical green object on top of a box. (c) demonstrates the result for the red covered parts of a chair. In all these experiments a color-based background subtraction is performed based.

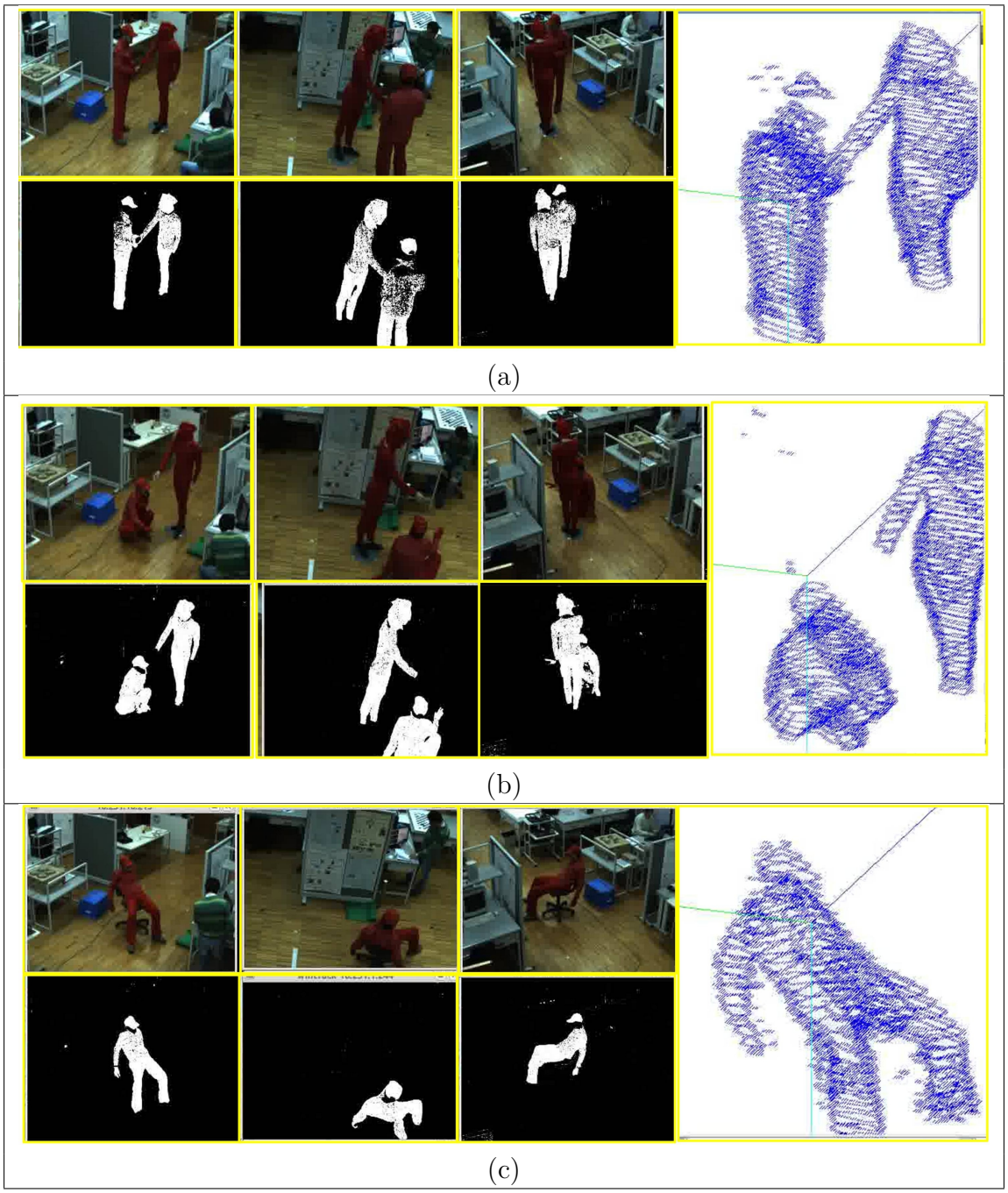


Figure 4.15: Results of the proposed multi-layer 3D data registration: The first experiment, (a), stands for a scene where a person is hand-shaking with a mannequin. In the second one, (b), the person is seated in front of the mannequin. (c) shows a case where the person is seated on a chair. A set of euclidean inertial planes are used (with interval of $35\,mm$) to register the data in 3D. The image at the right demonstrate the visualized result.

blue, red and green respectively indicate the processing time of virtual images generation, projection of generated virtual imaged onto a set of inertial-planes and the total algorithm cycle, respectively. Also in this diagram the processing time has a linear proportion related to number of allocated inertial-planes.

From these two analysis, shown in Fig. 4.16 and Fig. 4.17, the slopes for two cases are respectively 0.002 ms/cm^2 and 6.43 ms/cm^2. Having this, it can be concluded that the performance of the system regarding the processing time much more depends to the number of inertial planes than their areas.

As shown in Fig. 4.6, a big amount of running time of the algorithm is spent where we project a virtual image plane onto a set of inertial planes. This part is a bottleneck for the system. Such a time consumption can have two reasons. First because of increasing number of planes and consequently the volume of data to be processed. The second reason is due to number of block-copy actions which are carried out between the host and the device.

As can be seen, for each Euclidean inertial plane, we once transfer the data from host to the device memory (upload) and then after applying the homography warping and intersection operations on the device the results get transferred on the host memory (download). These operations are repeated for each one of the inertial planes. In this particular implementation of the algorithm the unit which displays (visualizes) the data runs on the host and needs to have the data on the memory of the host. If there was not such a need, the processed (registered) data on the device would not need to be downloaded to the host. Indeed this depends too much to an application which will use the registered data. If the application is also implemented on GPU and has the capability of using the registered data directly from the device memory, then the downloading operation can be eliminated and leads to have a higher speed.

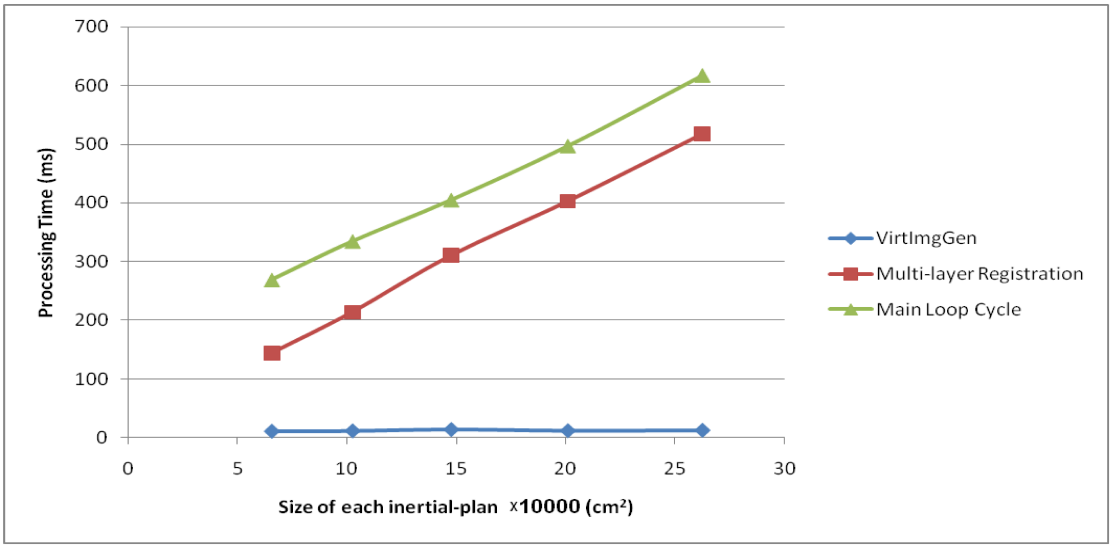


Figure 4.16: Average processing times in *ms* for different size of inertial-planes. The notations are related to the flowchart shown in Fig. 4.6. Number of 2D inertial-planes used in this statistic is 48.

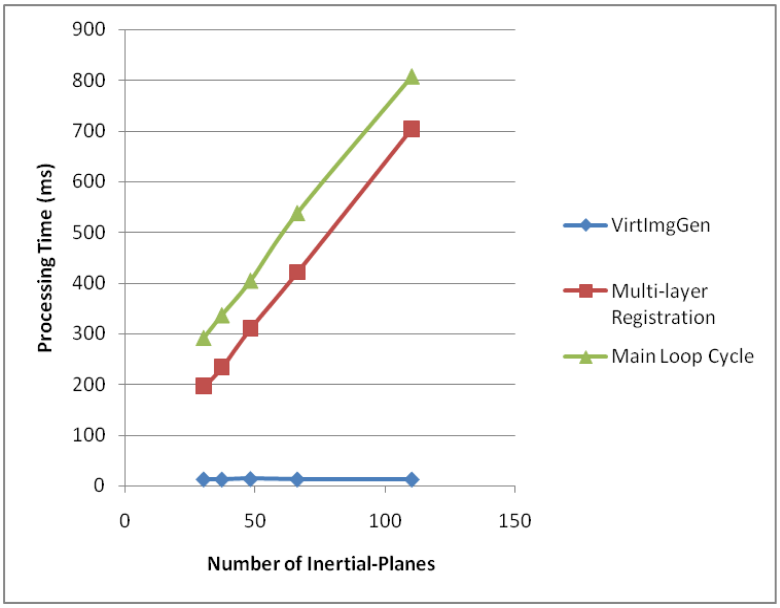


Figure 4.17: Average processing times in *ms* for different number of inertial-planes. The notations are related to the flowchart shown in Fig. 4.6. The size of each 2D inertial-planes used in this statistic is $384 \times 384\, cm^2$.

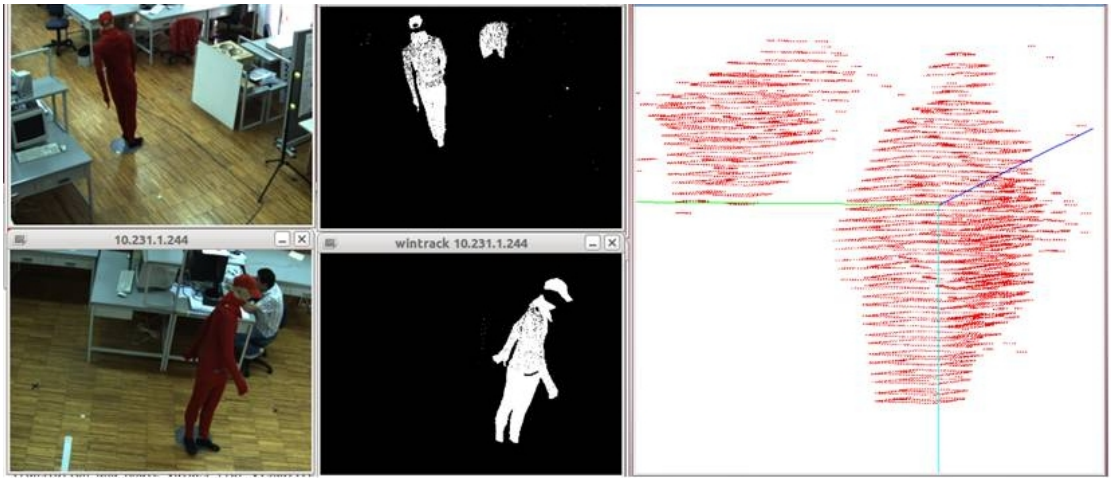


Figure 4.18: Mobile sensor experiment: Result of 3D reconstruction when just two IS-camera couples are used. The other cameras are intentionally blinded. The result is shown in the right column. Because of lack of views, the details are not clear and moreover a ghost object has appeared.

Figure 4.19: Result of 3D reconstruction when a mobile sensor is augmented to the network (corresponding to Fig. 4.18); In order to have more details of the scene, a mobile sensor is navigated close to the mannequin and its view is integrated as a new node in the network. The left two columns are the images corresponding to the two fixed cameras and the third column from left is the image corresponding to the mobile camera. The results of the 3D reconstruction by using two fixed IS-camera couples and a new augmented couple is demonstrated in the right column.

4.3.3 Extension for mobile sensor

The previously shown experiments were carried out by using static sensors. In some scenarios, it would be very useful to have a mobile sensor which could move inside the scene and collect data from an arbitrary point of view. The data provided by it can be used as a regular node of the sensor network. Such a mobile sensor has two main advantages: Firstly, always it is not possible to have many cameras (specially in large areas) to have all details of the different parts of the scene. Secondly, in some cases one of the main nodes (IS-camera couples) could be occluded or in any reason stop to work. In such situations, a mobile sensor could approach to an appropriate position in the scene, gather and transmit close-view information to the infrastructure. The proposed framework has the ability to integrate the data coming from a mobile sensor. The localization and navigation of a mobile sensor are the two old topics in the area of robotics and computer vision and there can be found many papers in the literatures which proposed different solutions for these problems. Therefore we do not enter in these areas and just assume that we have these techniques already available. In following, an experiment is provided to show the advantage of using a mobile sensor. In order to localize the mobile sensor, the method proposed in [AD11a] is used. Fig. 4.18 shows a case where just two cameras from the infrastructure is used for the 3D reconstruction of a mannequin (we intentionally blinded the other cameras). As can be seen, in such situations that there is not enough views to see the scene, the result of 3D reconstruction is not good enough. As seen, there is no enough detail about the reconstructed person and moreover a ghost object [MSEH08] has appeared as noise. In order to have more details of the scene, a mobile sensor is navigated close to the mannequin. Then after localizing the mobile sensor, its view is integrated as a new node in the network. The results of the 3D reconstruction by using two fixed IS-camera couples and a new added couple is demonstrated in fig. 4.19. This figure shows the advantage of having a mobile sensor which could cooperate with the infrastructure.

4.4 Conclusion

Having real-time volumetric reconstruction of scene (human and object) is demanding by many applications such as human motion and behaviour modelling, teleconferencing, human-robot interaction, smart-room, health-care, medical industries, virtual reality, scene understanding, surveillance, game industries etc. Nowadays, camera network is frequently deployed for public or even private observations for different purposes depending to the application. Recently, IS is becoming much cheaper and more available. Even many smart phones can be found equipped in both IS and camera. Taking advantage of this, we used a network of IS-camera couples to observe the scene and then a method for 3D reconstruction of a person using inertial data and with no planar ground assumption was proposed. In order to achieve a real-time execution, a parallel processing architecture was proposed and implemented on CUDA. Different real-time experiments were provided in this chapter to demonstrate the applicability and effectiveness of the proposed method for many applications. The experiments include 3D reconstruction for a single person, a person and a mannequin and some objects. The presented results are quite promising.

Chapter 5

Contribution on sensor configuration and tracking

5.1 Introduction

In this chapter first we discuss about the issue of having an appropriate coverage for cameras within the proposed data registration framework and a genetic algorithm is proposed to improve this issue. Synergy among several heterogeneous sensors can provide more precise result specially when some sensors are mounted on a mobile robot. On this context we discuss about how to estimate the extrinsic parameters among cameras and laser sensor and propose a method for that. Afterwards we discuss about how the dynamic state of a scene can be considered in the proposed framework and for this purpose the Bayesian techniques are applied on the registration plane.

5.2 Edge visibility criteria and camera configuration

Algorithm 5: Criteria to check the edges visibility for a given polygon. k is number of polygon's edges and e_j is the j'th edge. $\mathbf{n}_j$ is the normal vector corresponding to e_j. $\mathbf{b}_i$ is the bisecting vector for camera i. Each edge is checked and will be labelled as either 'visible' or 'invisible'. Labelled as 'invisible' for an edge means that it is invisible for all the cameras.

```
for j = 1 to k do
    if ∃b_i, i ∈ 1..n_c where ∠(n_j, b_i) > π/2 then
        consider the edge e_j as < visible >
    else
        consider the edge e_j as < invisible >
```

The proposed volumetric reconstruction method uses silhouettes of an object and provides its volumetric reconstruction. The completeness of the reconstructed volume depends to some parameters such as the positions of cameras within the network, number of cameras and the shape of the object. Fig. 5.1-left shows an exemplary case where a convex polygon is observed by two cameras (top view). In this case, the polygon has five edges and five vertices (pentagon) however as

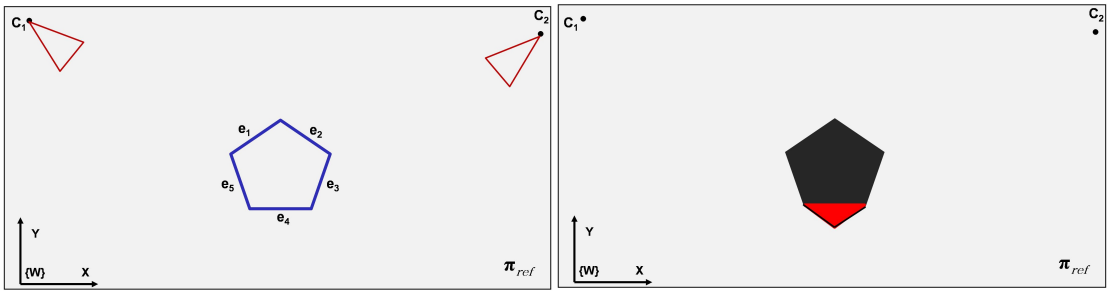


Figure 5.1: Investigation of the criteria for visibility of a general convex polygon. Left: An exemplary convex polygon is being observed by two cameras. The images are shown from the top view of the inertial reference plane π_{ref}. Right: The registration of the polygon corresponding to left picture. The registration includes the object and some extra areas (coloured in red) which does not belong to the polygon. This red area has appeared because of not having visibility on the lowest edge of the polygon.

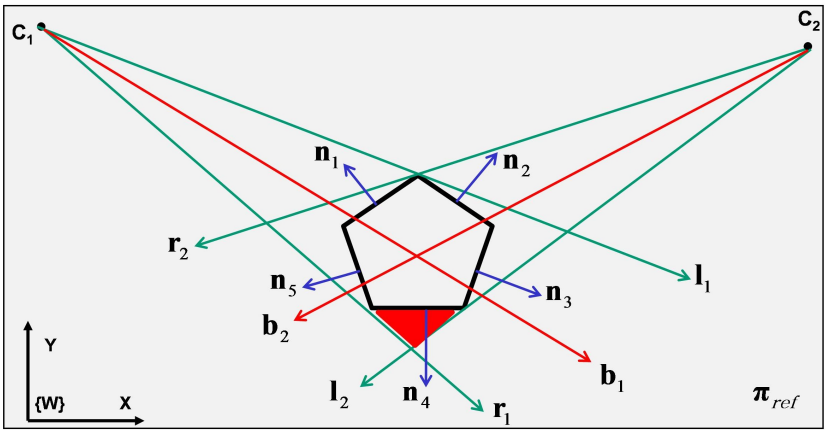


Figure 5.2: Registration plane corresponding to Fig. 5.1. The figure shows the involved vectors. Green vectors, $\mathbf{l}_i$ and $\mathbf{r}_i$, respectively indicate the left and right tangents (bounding vectors) of a camera c_i. The bisector vector for each camera bounding pair (the tangents) of $\mathbf{l}_i$ and $\mathbf{r}_i$ is shown in red ($\mathbf{b}_i$). $\mathbf{n}_i$ stands for the normal of the edge e_i. After performing the registration process based on the proposed algorithm, the area coloured in red also become registered as a part of the object.

is shown in Fig. 5.1-right it is registered on the inertial plane as a six edges polygon, due to the effect of the mentioned parameters (number of cameras and their positions). The extra part of the polygon after registration is shown in red in the figure. As previously mentioned, registration of the cross section of an object with an inertial plane can be thought as the intersection among all shadows created by cameras, through considering each camera as a light source. Based on this interpretation the appearance of the red part can be justified: the red part is the area which can not be seen by any camera and is shadowed in all views. We intend to introduce a geometric approach to realize the visibility or invisibility of an edge. Assume a general convex polygon including k vertices $V = \{v_1, v_2, ... v_k\}$ and k edges $E = \{e_1, e_2, ... e_k\}$ (e.g. consider Fig. 5.2 as an exemplary polygon corresponding to Fig. 5.1). A normal vector can be considered for each edge resulting to have $N = \{\mathbf{n}_1, \mathbf{n}_2, ... \mathbf{n}_k\}$. Moreover assume a set of n_c cameras $\{c_1, c_2, ... c_{n_c}\}$. Each camera c_i has a pair of tangents (bounding vectors) $(\mathbf{l}_i, \mathbf{r}_i)$ to the polygon and for each tangents pair a bisecting vector $\mathbf{b}_i$ is considered. Having this, the visibility criteria for the edges can be expressed as following: an edge e_j is visible if and only if there is a $\mathbf{b}_i$ where $\angle(\mathbf{n}_j, \mathbf{b}_i) > \frac{\pi}{2}$ (see Alg. 5).

5.2.1 Optimal Camera placement using Genetic Algorithm

The visibility criteria defined in Alg. 5 can be used for obtaining an optimal solution for camera placement. The question to be solved is as following: given a convex polygon with k vertices and k edges and n_c number of cameras, what would be an optimal solution for placements of cameras in order to have the best observation of the polygon for applying the proposed reconstruction method. This question can be considered in another form: Given a polygonal space to be monitored by a camera network, what would be an optimal solution to place n cameras around the space. GA is a bio-inspired algorithm which is known as an appropriate mechanism to solve such a problem. We continue to describe our GA-based algorithm to solve the mentioned problem.

Algorithm 6: Algorithm to generate a valid gene. **V** is the matrix of vertices of the polygon. max_fov is the maximum possible FOV for each gene (camera) and 'space' is the search space. Having these as the inputs, the algorithm generates a valid gene with its properties. The position of each gene signifies the position of the corresponding camera. The function getTangentsToPolygon(V,p) receives the matrix of the vertices of the polygon (**V**) and the position (**p**) of the camera (gene) and returns two vectors (**l** and **r**) which are tangents to the given polygon. Then the angular bisecting vector is stored in **b**. This bisecting vector will be used to compute the cost value of the gene. It can be also interpreted as the looking direction of the camera. Then the generated gene is returned as the result of the function createGene()

Function createGene()

Input: {**V**, max_fov, space}

Output: {gene}

begin

  repeat

    p ← random 2D position in space

    [**l**,**r**] ← getTangentsToPolygon(**V**,**p**)

    $\text{fov} \leftarrow \arccos(\frac{\mathbf{l}.\mathbf{r}}{|\mathbf{l}||\mathbf{r}|})$

  until fov <= max_fov

  gene ← generate_an_empty_gene()

  gene.**p** ← **p**

  gene.fov ← fov

  gene.**b** ← |**l**| **r** + |**r**| **l**

end

return gene

Population in GA is a set of members called chromosome. Each chromosome includes a number of elements named gene. In our case a gene is equivalent to a camera and its properties and a chromosome is synonymous to a set of cameras. The structure of a chromosome string is defined in the Fig. 5.3. In this structure **p** and **b** stands for the vectors of position and bisector of the camera, fov is the angle of FOV and cost is an scaler value corresponding to the gene's cost. Based on these definitions, an algorithm to generate a valid gene is provided in Alg. 6. The inputs of the algorithm are the vector of vertices of the given polygon, **V**, the search space of the camera positions and maximum possible FOV. Alg. 7 presents a function to generate a chromosome including N_c genes (a chromosome string with length=N_c).

Algorithm 7: Algorithm to generate a chromosome. **V** is the matrix of vertices of the polygon. N_c indicates the chromosome's length or in other words the number of cameras. max_fov is the maximum possible FOV for each gene (camera) and 'space' is the search space. Given these as inputs the algorithm generate a chromosome with N_c genes and returns it (using Alg. 6).

```
Function generate_chromosome()
Input: {V, max_fov, N_c, space}
Output: {chromosome}
begin
    // creating an empty chromosome with length=N_c
    chromosome ← empty_chromosome(N_c)
    for i ← 1 to N_c do
        gene(i) ← createGene(V, max_fov, space)   // from Alg. 6
    getChromosomeCost(chromosome)   // from Alg. 8
end
return chromosome
```

One of the crucial points in GA-based algorithms is to have a suitable cost function in order to evaluate the fitness of a member in the population. In the case of our coverage problem, a cost function $f(\alpha = \angle(\mathbf{b},\mathbf{n})) : [0..\pi] \rightarrow [0..\lambda]$ is

Figure 5.3: Structure of a chromosome string.

chromosome															
gene(1)				gene(2)				...			gene(n_c)				
p	fov	**b**	cost	**p**	fov	**b**	cost	...			**p**	fov	**b**	cost	

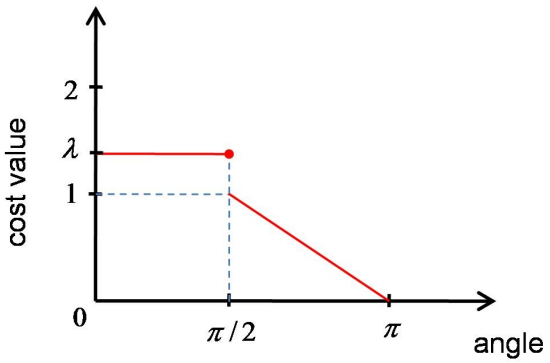


Figure 5.4: Defined function to measure the cost between a camera and a polygon edge. The maximum cost is equal to λ and happens when $\alpha <= \pi/2$ or in other words the edge is invisible by the camera.

defined (see Fig. 5.4) for a bisector **b** and the normal of an edge **n** as following:

$$f(\alpha = \angle(\mathbf{b},\mathbf{n})) = \begin{cases} |\frac{2}{\pi}(\alpha - \pi)| & ; \quad \frac{\pi}{2} < \alpha \leq \pi \\ \\ \lambda & ; \quad \textit{others} \end{cases} \tag{5.1}$$

where $1 < \lambda < 2$ and α is the angle among the two vectors **b** and **n**. An algorithm to compute the cost of each gene in a chromosome string using the defined cost function in Eq.(5.1) is proposed in Alg. 8. Firstly the cost among each individual gene of the chromosome and each edge of the polygon is computed (whole algorithm except the lines 12-20). The primary costs for each gene and the polygon's edges were obtained regardless of considering the other genes in the chromosome. Secondly, the cost of each gene gets updated by taking into account the previous genes in the chromosome, in order to avoid getting trapped in a local minima (lines 12-20). Fig. 5.5 shows an exemplary case where a triangular polygon is supposed

to be optimally observed by three cameras. Based on the cost function in Eq. (5.1) (the second part of Alg. 8) an optimal arrangement for the cameras is when all three cameras observe the edge e_{12}*. In this situation the cost value for each camera is close to zero (using Eq. (5.1), since the angle among the bisector vector of the cameras and $\mathbf{n}_1$ (normal of e_{12}) is straight (π). This case is considered as a local minima for the camera placement. In order to avoid the GA algorithm to fall into such a local minima, an update on the cost of each gene in a chromosome with regards to the other genes in the same chromosome is proposed in the lines 12-20 of Alg. 8. In this part, the cost of each gene-edge gets increased (penalized) if the same edge was previously observed by another antecedent gene in the chromosome. In this case, the unidirectionality between the bisector vectors of such two genes ($\mathbf{b}_c$ and $\mathbf{b}_p$ for the current gene and the antecedent one) determines the penalty value to be augmented to the cost value of the second gene. The more aligned in the same directions, the more penalty value is applied by using the following equation:

$$aug(\mathbf{b}_c, \mathbf{b}_p) = |1 - \frac{\angle(\mathbf{b}_c, \mathbf{b}_p)}{\pi}| \tag{5.2}$$

This causes the genes who are observing the same edges get far from each other and converge other edges.

A genetic algorithm (Alg. 9) to search for an optimal solution is proposed in this section by using the defined cost function and the introduced sub-functions (Alg. 6, Alg. 7 and 8). The inputs for this algorithm are: V, the matrix of vertices; max_fov, maximum for the FOV of a camera (a gene); N_c, number of cameras (number of genes in a chromosome or chromosome's length); and 'space', the space to be searched by GA for placing cameras (search space). Number of population (number of chromosomes) is considered as 100. First a new generation is initialized. After applying a fitness function on each individual (chromosomes) in this population, 20% of them are selected as elites for the new generation.

*e_{12} denotes the edge connecting vertix v_1 to vertix v_2

Algorithm 8: Algorithm to compute the cost of a chromosome and its genes. The inputs are $\mathbf{V}$, the vertices's matrix and chromosome. The cost value among each individual gene in the chromosome and each edge of the polygon is computed using the Eq. (5.1). The cost value gets penalized for the genes which are visiting an edge that was previously visited by an antecedent gene of the chromosome (lines 12-20). The penalty value is obtained using Eq. (5.2).

```
   Function getChromosomeCost() Input: {V, chromosome}
   Output: {cost of chromosome}
1  begin
2      λ ← 1.2
3      l ← length(chromosome)   // number of genes in chromosome
4      for i ← 1 to l do
5          gene(i).cost ← 0
6          for j ← 1 to k do
7              h ← mod(j,k) + 1
8              b ← gene(i).b
               α ← arccos( b.e_jh / (|b||e_jh|) )
9
10             if α > π/2 and α <= π then
11                 gene(i).e_jh.cost ← (2/π)|α − π|
12                 //check if e_jh was previously visited by an antecedent gene
13                 for prev_gene ← 1 to i − 1 do
14                     if gene(prev_gene).e_jh <= 1 then
15                         //yes visited, so penalize it!
16                         b_i ← gene(i).b
17                         b_p ← gene(prev_gene).b
                           α ← arccos( b_p.b_i / (|b_p||b_i|) )
18
19                         augmented_cost ← |1 − α/π|
20                         gene(i).e_jh ← gene(i).e_jh + augmented_cost
21             else
22                 gene(i).e_jh.cost ← λ
23             gene(i).cost ← gene(i).cost + gene(i).e_jh.cost
24 end
25 return chromosome.cost
```

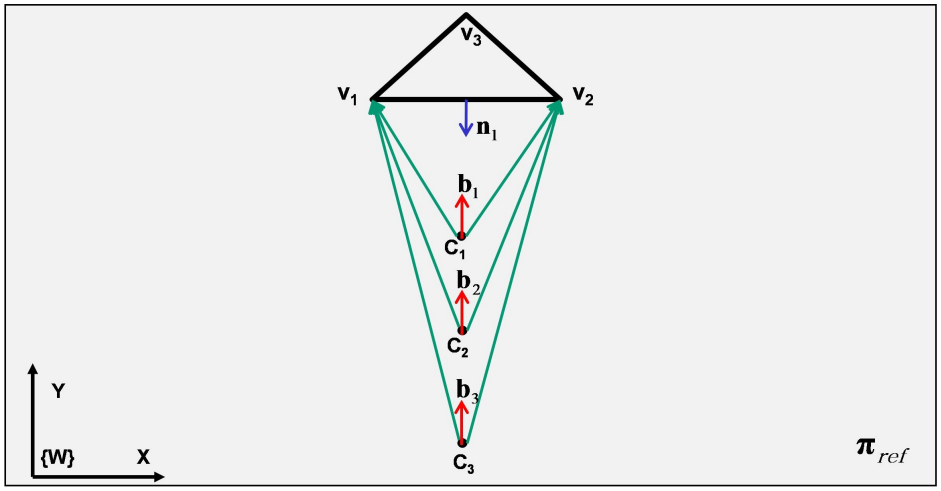


Figure 5.5: The local minima problem for a triangular polygon and three cameras. Using just the cost value for each gene (camera) regardless of the other genes (cameras) in the same chromosome (camera network) can leads to have one edge perfectly observed by many cameras and other edges starving. In this case all three cameras are observing the edge e_{12} (the line between v_1 and v_2) with cost values are zero since $\mathbf{n}_1$ is opposite to their bisector vectors ($\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{b}_3$), whereas the two other edges (e_{23} and e_{31}) are not observed at all since their cost value can not be zero. The second part of Alg. 8 is dedicated to eliminate this problem using the penalty function in Eq. (5.2).

The rest of the population (80%) are created by applying crossover and mutation operations on the elites. For doing so, every time two parents are selected randomly from the elites. Then a crossover operation is applied in this selected couple and as the result two children (new member of the society or chromosomes) are added to the population. On some of the newly created children, a mutation is applied as well. The probability of happening a mutation on the children is considered as 0.50. In each cycle, the cost value of the best member (best fitted chromosome) of the elites is saved as the minimum cost value of that generation. If after a n_{stop} number of consecutive trials the cost value does not get improved, or the algorithm reaches its maximum iteration (n_trial_max) then it stops. The answer of the algorithm is an optimal chromosome. This optimal solution includes a set of genes (N_c genes) and each gene signifies a camera with its properties such as position, direction vector, etc.

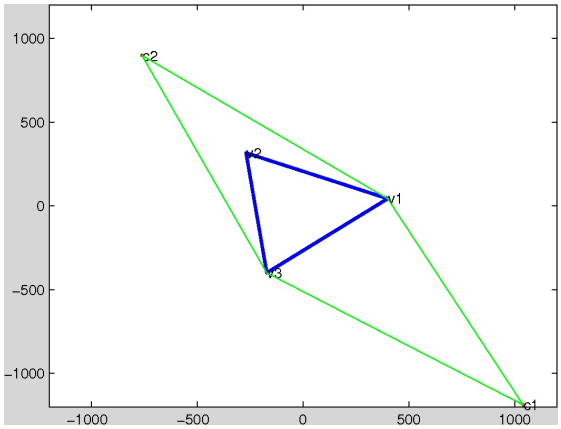
Algorithm 9: Genetic algorithm to search for an optimal solution for camera placement problem.

```
   Function GA()
   Input: {V, max_fov, N_c, space}
   Output: {optimal_chromosome}
1  begin
2      /* Initialization */
3      chrom_length ← N_c   //number of genes in a chromosome
4      n_elites ← 20   // number of elites to be selected
5      n_pop ← 100   //number of population
6      n_stop ← 150   //stop after no change in a consecutive n_stop iterations
7      search_space ← space
8      fov_max ← π/4   // maximum possible FOV for a camera.
9      n_trial_max ← 2000   //maximum number of iterations
10     /* First generation */
11     for i ← 1 to n_pop do
12         pop(i) ← generate_chromosome(V, fov_max, chrom_length) //Alg. 7
13     evaluate_fitness()   // using Alg. 8
14     elites ← pop(1..n_elites)
15     cost_history(1) ← getChromosomeCost(V, elites(1))   //Alg. 8
16     t ← 0
17     n_repeated ← 0
18     /* Iterations */
19     while t < n_trial_max and n_repeated < n_stop do
20         t ← t + 1
21         pop(1 : n_elites) ← elites
22         pop(n_elites + 1 : n_pop) ← crossover and mutation on elites
23         evaluate_fitness()   // using Alg. 8
24         elites ← pop(1..n_elites)
25         cost_history(t) ← getChromosomeCost(V, elites(1))   //Alg. 8
26         if t > 1 and cost_history(t) == cost_history(t − 1) then
27             n_repeated ← n_repeated + 1
28         else
29             n_repeated ← 0
30     optimal_chromosome ← elites(1)
31     return optimal_chromosome
32 end
```

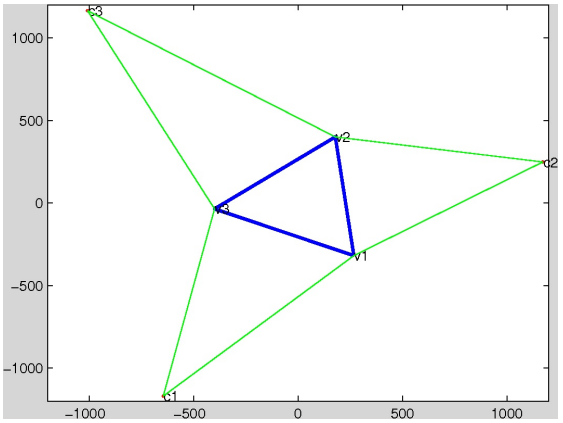
Although the proposed algorithm to optimize the camera coverage is discussed in 2D as a case-study, however it has the potential to be used in 3D with some small modifications. The first necessary modification in the algorithm to deal with a 3D case is that instead of using the normal vectors of the edges, the normal vectors of the faces have to be used. This counts all faces except the bottom face which is not needed to be observed. The second needed modification is to consider the camera position as 3D instead of 2D. E.g. in Alg. 6, the place where a camera position in space is randomly created it should be generated as 3D vector. The rest of the algorithm would be the same as the studied 2D case.

5.2.2 Camera placement optimization using GA: simulation

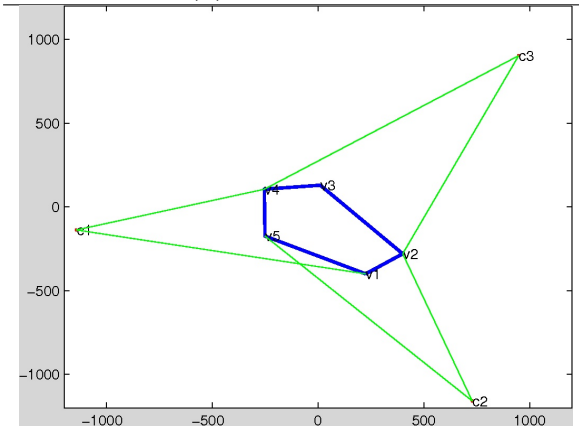
In this sub-section a set of experiments to demonstrate the efficiency and effectiveness of the proposed GA-based algorithm for camera placement is demonstrated. Totally nine samples are shown in Fig. 5.6, Fig. 5.7 and Fig. 5.8. In each sample a convex polygon with k number of edges and n_c cameras are considered. The polygons are randomly generated and the space to be searched by camera placement has a dimension equal to 1200×1200cm. The convergence plots of the algorithm for the samples in each figure has been depicted in the corresponding (d) section. The vertical axes in the plots show the cost value of the best fittest chromosome in each iteration where the value is divided by number of genes for each sample (number of cameras n_c). The condition to stop the GA loop is when the cost values of the found solution in 150 consecutive trials do not get improved. As mentioned, the polygon can be either considered as an object to be reconstructed or an area to be observed by the camera network, however for our case it is considered as the first case. The proposed GA-based algorithm tries to find an optimal placements (position and direction) of the cameras within the network in such a way that gives the best coverage on the polygon for the purpose of proposed 3D reconstruction method.



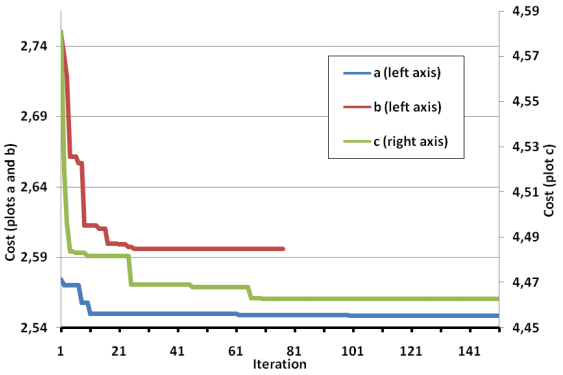
(a) $k = 3, n_c = 2$



(b) $k = 3, n_c = 3$

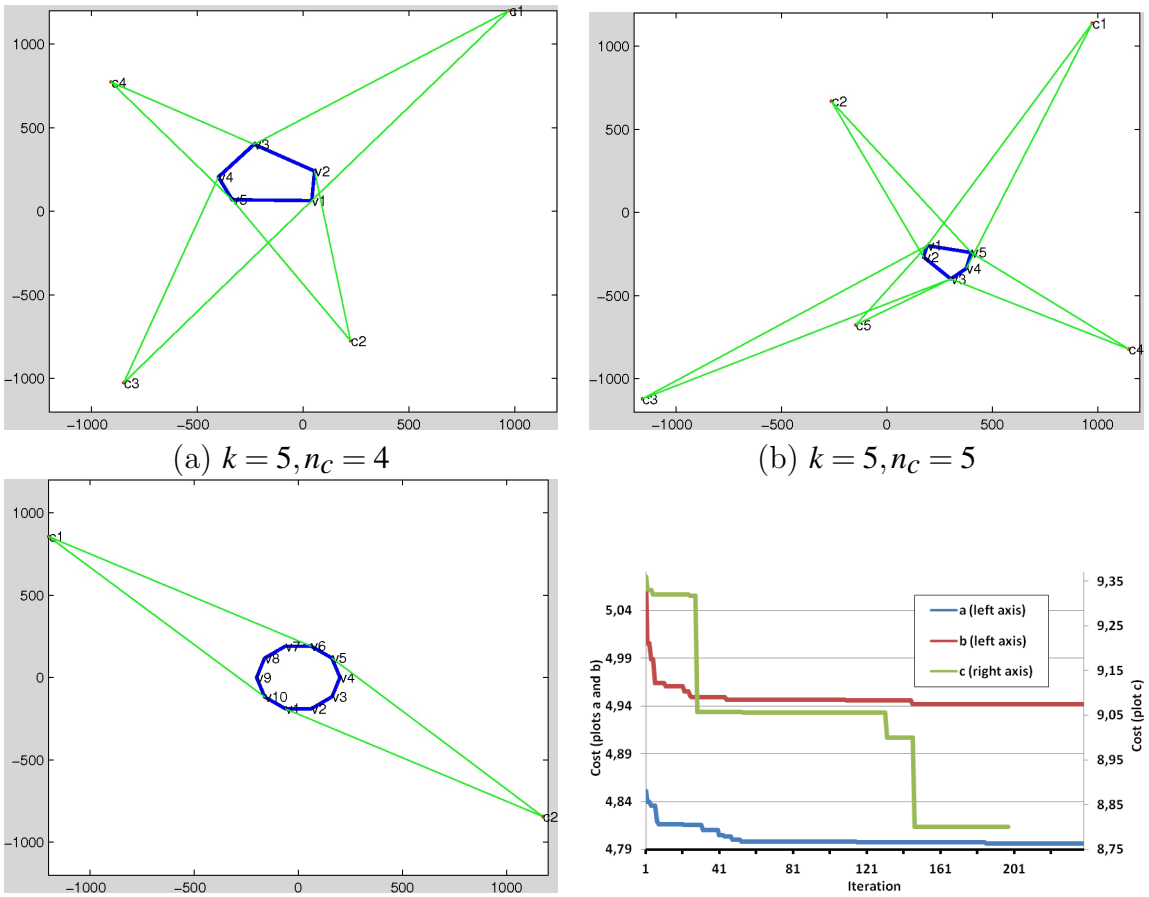


(c) $k = 5, n_c = 3$



(d) Convergence plot

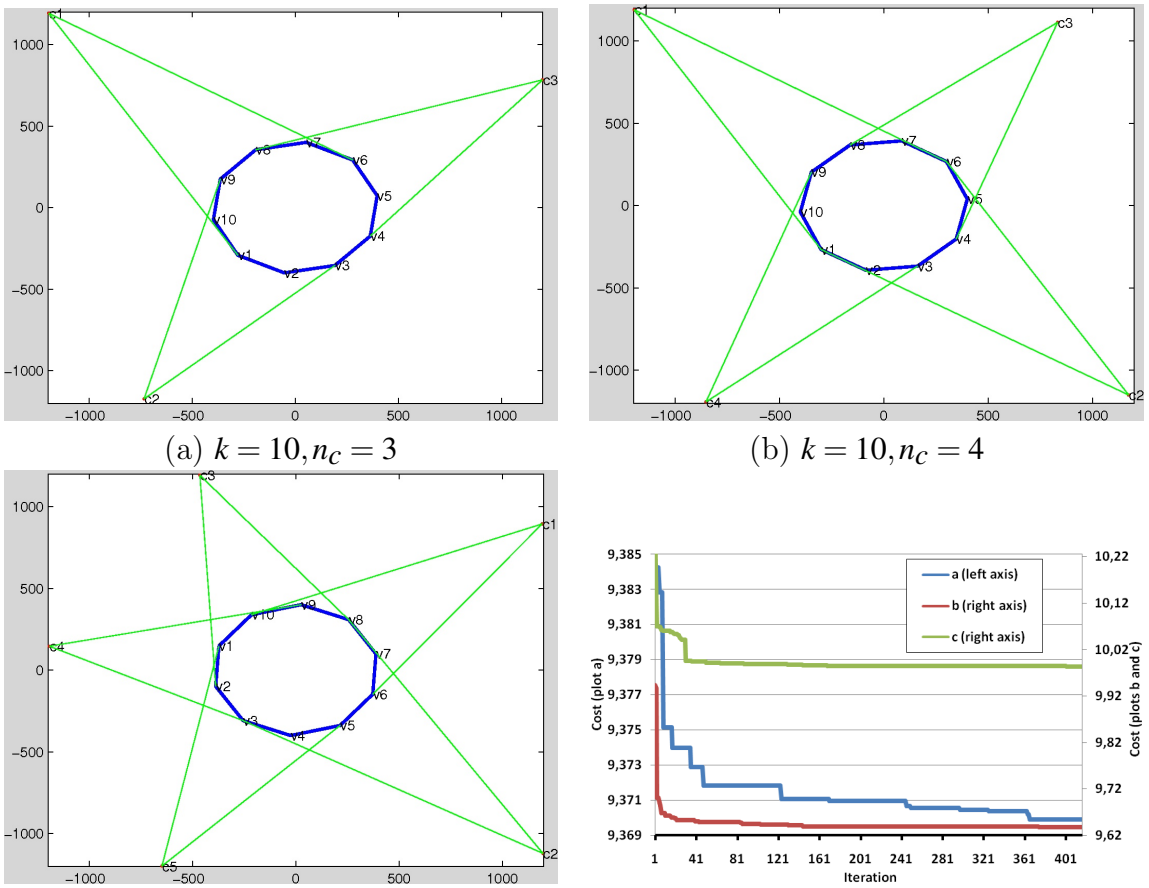
Figure 5.6: Results for camera placement optimization using the proposed GA algorithm. (a), (b) and (c) depict three samples. In each sample a polygon with k vertices is randomly generated and the purpose of the algorithm is to search for an optimal coverage using n_c number of cameras. The convergences for the samples are plotted in (d). The vertical axis depicts the cost value for the fittest chromosome in each iteration, once gets divided to the number of genes (n_c). The dimension of the search space is $1200 \times 1200\, cm^2$.



(a) $k = 5, n_c = 4$ (b) $k = 5, n_c = 5$

(c) $k = 10, n_c = 2$ (d) Convergence plot

Figure 5.7: Results for camera placement optimization using the proposed GA algorithm. (a), (b) and (c) depict three samples. In each sample a polygon with k vertices is randomly generated and the purpose of the algorithm is to search for an optimal coverage using n_c number of cameras. The convergences for the samples are plotted in (d). The vertical axis depicts the cost value for the fittest chromosome in each iteration, once gets divided to the number of genes (n_c). The dimension of the search space is $1200 \times 1200\, cm^2$.



(a) $k = 10, n_c = 3$ (b) $k = 10, n_c = 4$

(c) $k = 10, n_c = 5$ (d) Convergence plot

Figure 5.8: Results for camera placement optimization using the proposed GA algorithm. (a), (b) and (c) depict three samples. In each sample a polygon with k vertices is randomly generated and the purpose of the algorithm is to search for an optimal coverage using n_c number of cameras. The convergences for the samples are plotted in (d). The vertical axis depicts the cost value for the fittest chromosome in each iteration, once gets divided to the number of genes (n_c). The dimension of the search space is $1200 \times 1200\, cm^2$.

5.3 Integration of mobile vision and laser sensor within a camera network - the estimation of extrinsic parameters

Earlier we proposed a data registration framework using a network of cameras and inertial sensors. In this framework camera was used as the main passive sensor to observe the scene. Although vision is one of essential modality in register and later perception of a scene, however it has some weaknesses. In the context of 3D data registration, cameras are capable of providing depth readings, with the further advantages of being passive sensors and of yielding additional information, such as surface colour. However, data registration using these sensors is highly dependent on light conditions, shadows and homogeneous textures. Conversely, a precise active sensor like laser range finder is able to provide 3D information of a scene with a much lesser degree of dependency on texture, but they are more expensive and less common than traditional cameras, and do not yield colour information. Therefore fusion of these two different modalities in a synergistic manner removes each of their individual shortcomings, while allowing for the overall harvesting of their advantages. Thus, integrating a range sensor within the proposed framework can be helpful. Before being able to integrate the range and image data, we have to know the extrinsic parameters among the reference frames of these sensors in order to perform data registration. Having this motivation, in this section a method to estimate the extrinsic parameters among a 3D-LRF and an stereo camera is proposed. It is worth to mention that depending to a further application (which will be benefiting from the proposed data registration framework), having a mobile sensor can grant some profits, as briefly discussed in Sec. 4.3.3. As another application, a mobile robot can be used in a smart-room (see Fig. 5.9). Smart rooms are sensor equipped areas that are able to perceive and understand what is happening in them. These systems can be applied to homes, offices, factories. Mobile robots appear as natural agents in the physical world to carry out smart room actions. A mobile agent, within an intelligent space, comprises several tasks such as his localization, localization and reconstruction of the person in front, identification, interaction with human, etc. To carry out these tasks, the mobile

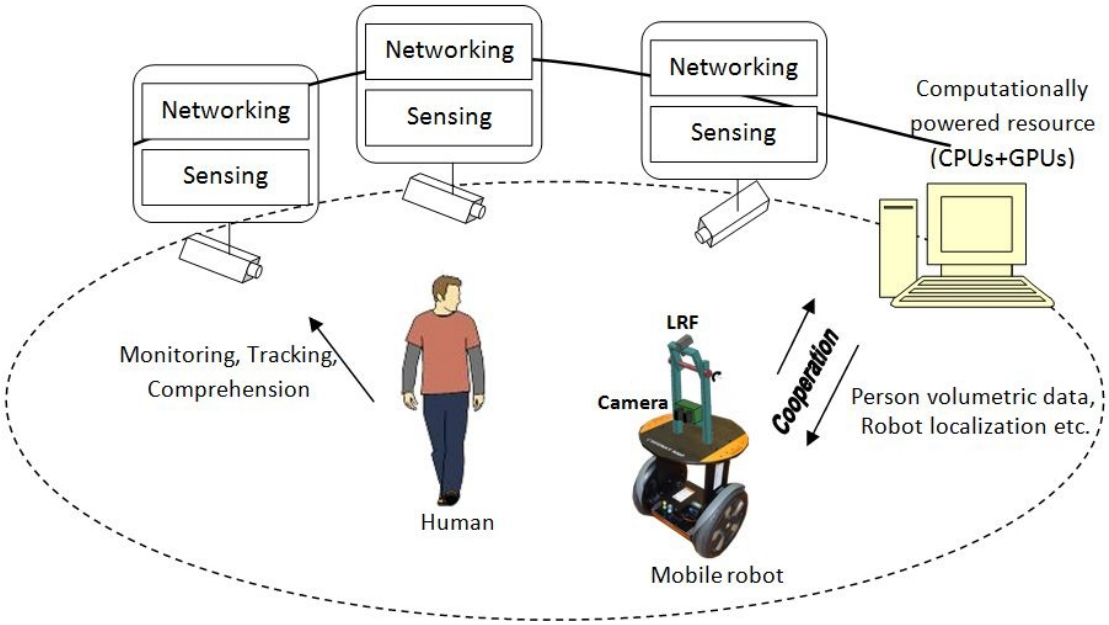


Figure 5.9: Schematic of a smart-room including a mobile robot

robot needs to be equipped with many sensors and have high power computation in order to achieve real-time performance. The information provided by the mobile agent is egocentric ($2D\frac{1}{2}$) which limits the robot's perception due to its field of view's constraint. This topic is far from the focus of our thesis and we do not go through more details on this, however in this section we give the schemes in the context of a mobile robot where the robot carries a set of heterogeneous sensors (vision, range and orientation).

This sub-section introduces a method to estimate the extrinsic parameters among a 3D-LRF and an stereo camera. A freely moving bright spot is the only calibration object which is needed here to collect data. A set of virtual 3D points is made by waving the bright spot through the working volume in three different planes. Its projections onto the images are found with sub-pixel precision. The same points are extracted according to the laser scan data and are corresponded to the virtual 3D points in the stereo pair.

5.3.1 LRF model

Our 3D laser range finder is built by moving a 2D LRF along one of its axes (tilt). By rotating the 2D scanner around its tilt axes, α, it is possible to obtain the spherical coordinates of the measured points. This type of configuration for the 3D laser can be modelled as following:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c_i c_j & c_i d_x + s_i d_z \\ s_j & 0 \\ -s_i c_j & -s_i d_x + c_i d_z \end{bmatrix} \begin{bmatrix} \rho_{ij} \\ 1 \end{bmatrix} \tag{5.3}$$

$$c_i = \cos(\alpha_i), c_j = \cos(\theta_j), s_i = \sin(\alpha_i), s_j = \sin(\theta_j),$$

where ρ_{i_j} is the j-th measured distance with corresponding orientation θ_j in the i-th scan plane, which makes the angle α_j with the horizontal plane. The offset of the rotation axis from the center of the mirror has components d_x and d_z. $[x\, y\, z]^T$ is the coordinates of each point measured relative to the center of rotation of the laser, with the x axis pointing forward and the z axis pointing up.

5.3.2 Problem definition

A setup with a LRF, a stereo vision system and inertial sensor is illustrated in Fig. 5.10-(a). The goal is to estimate the homogeneous transformation between the reference frames of the stereo camera and LRF. As shown in the figure, three coordinate frames, namely stereo camera $\{C\}$, laser range finder $\{L\}$ and the center of the rotation axis $\{I\}$ have been defined. The $\{C\}$ is located in the left camera center of the stereo pair. Furthermore, the IS is strongly coupled to the laser range finder and is used in order to measure the angle α_i. Let ${}^{C}T_{L(\alpha)}$ be the homogeneous transformation between the stereo camera and the laser range finder for each angle of the rotation axis α, which is described as:

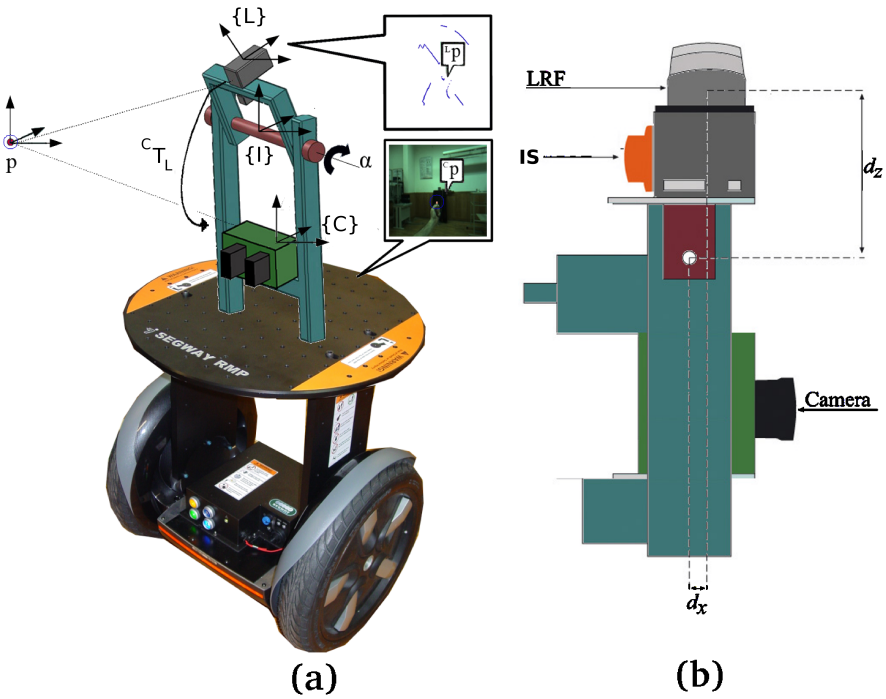


Figure 5.10: a) Schematic of the problem of calibration among a LRF and a stereo camera. The goal is to estimate the rigid transformation between the reference frames of the LRF and stereo camera. b) Sketch of the measurement system (dx and dz are the offset distances from the rotation axis to the center of the laser mirror).

$$ {}^{C}T_{L(\alpha)} = \begin{bmatrix} {}^{C}R_{L(\alpha)} & {}^{C}t_{L(\alpha)} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{5.4} $$

where ${}^{C}R_{L(\alpha)}$ is the rotation matrix between the LRF and the stereo camera, and ${}^{C}t_{L(\alpha)}$ is the translation vector. The coordinates of a 3D point P in $\{C\}$ and $\{L\}$ are respectively denoted by ${}^{c}P$ and ${}^{l}P$ and the relation among them can be expressed as

$$ {}^{c}P = {}^{C}T_{L}\,{}^{l}P \tag{5.5} $$

The intention is to estimate ${}^{C}T_{L(\alpha)}$. By having such a transformation matrix it will be possible to transform 3D points between two coordinate systems $\{L\}$ and $\{C\}$.

5.3.3 Approach

The proposed calibration procedure, to estimate the extrinsic parameters between a tilt 3D-LRF and a stereo camera, is divided into the following three consecutive stages.

1. Estimating ${}^{C}T_{L(\alpha_0)}$, ${}^{C}T_{L(\alpha_1)}$ and ${}^{C}T_{L(\alpha_2)}$, which stand for the transformation matrices from the $\{L\}$ to $\{C\}$ when the LRF is placed in three different orientations around its rotation axis, α_0, α_1 and α_2, respectively.
2. Obtaining ${}^{I}T_{L(\alpha)}$. This matrix defines the transformation between $\{L(\alpha)\}$ and the center of rotation, $\{I\}$ ($\{I\}$ is considered as an auxiliary intermediate reference frame).
3. Calculating the final transformation ${}^{C}T_{L(\alpha)}$ as the extrinsic parameters between the tilt-LRF in any arbitrary angle and the stereo-camera.

These stages are explained in the next sub-sections.

A) Obtaining ${}^{C}T_{L(\alpha_j)}$ for three different values of α.

Firstly the LRF is placed in three different angles α_0, α_1 and α_2 making three reference frames for the LRF, namely $\{L(\alpha_0)\}$, $\{L(\alpha_1)\}$ and $\{L(\alpha_2)\}$ (see Fig. 5.11). The idea is to estimate ${}^{C}T_{L(\alpha_0)}$, ${}^{C}T_{L(\alpha_1)}$ and ${}^{C}T_{L(\alpha_2)}$ (the transformation matrices among each of these three reference frames and $\{C\}$).

For each one of the three angles, a set of 3-D corresponding points are collected. It leads to have ${}^{c}P^{\alpha_j} = \{{}^{c}P_i^{\alpha_j} \mid i = 1...N, j = 0...2\}$ and ${}^{l}P^{\alpha_j} = \{{}^{l}P_i^{\alpha_j} \mid i = 1...N, j =$

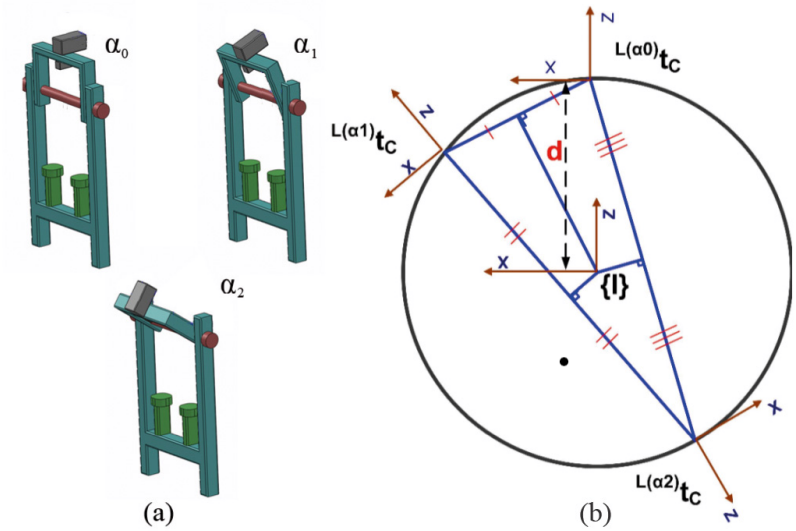


Figure 5.11: a) The LRF is placed (tilted) in three different angles α_0, α_1 and α_2. b) Using geometrical concepts in LRF and stereo camera calibration process: O_0, O_1 and O_2 denote the centers of LRF in these three angles. These three points make a triangle whose circumcircle indeed is the center of rotation, $\{I\}$. Note that the plane containing the triangle conventionally has a normal parallel to Y axis of $\{I\}$.

$0...2\}$ where N is the number of points and j is the angle's index. In each set the corresponding points must satisfy the following equation:

$$ {}^{c}P^{\alpha_j} = {}^{C}R_{L(\alpha_j)}\,{}^{l}P^{\alpha_j} + {}^{C}\mathbf{t}_{L(\alpha_j)} \tag{5.6} $$

being ${}^{C}R_{L(\alpha_j)}$ the rotation matrix, and ${}^{C}\mathbf{t}_{L(\alpha_j)}$ the translation vector of the homogeneous transformation ${}^{C}T_{L(\alpha_j)}$. In order to estimate ${}^{C}R_{L(\alpha_j)}$ and ${}^{C}t_{L(\alpha_j)}$ we use Arun's method [KSAB87]. This method tries to estimate the rotation matrix and translation vector in such a way that the following equation gets minimized:

$$ E = \sum_{i=1}^{N} |{}^{c}P^{\alpha_j} - ({}^{C}R_{L(\alpha_j)}\,{}^{l}P^{\alpha_j} + {}^{C}\mathbf{t}_{L(\alpha_j)})|^2 \tag{5.7} $$

In order to perform the collection of some corresponding points between LRF and camera, a simple laser pointer as a bright spot has been used. The idea of using such as tool for the calibration is originally inspired from an auto-calibration method between multi cameras by Svoboda in [SMP05] and by Barreto et al. in [BD04]. Their methods are extended in the proposed approach for LRF-camera calibration. The procedure is achieved in three steps for each α_j angle:

1. LRF data acquisition and pre-processing. A simple method is used to distinguish the bright spot from the background. Let ${}^{l}P_{i_{back}}^{\alpha_j} = \{P_{i_{back}}^{\alpha_j}(\theta,\rho)_i \mid i = 1...n_l\}$ be the range data of the background (before putting the bright spot inside the LRF view field) and ${}^{l}P_{i}^{\alpha_j} = \{P_{i}^{\alpha_j}(\theta,\rho)_i \mid i = 1...n_l\}$ be the range data at the moment, in which n_l is number of point read by the LRF, then to detect the laser pointer as a foreground abject we can use

$$|{}^{l}P_{i}^{\alpha_j} - {}^{l}P_{i_{back}}^{\alpha_j}| >= U_{th} \tag{5.8}$$

   where U_{th} is a threshold. Thus, in order to obtain the ${}^{l}P_{i}^{\alpha_j}$ set, the scan data is acquired with the laser pointer located out of the LRF´s field of view, which is considered to be planar. Therefore, meanwhile that the LRF is capturing range signals, the bright spot is slowly raising until to hit the LRF's plane. More pairs of 3D points can be collected by repeating this process.

2. Stereo-camera data acquisition and pre-processing. As soon as a foreground point is detected by the LRF, the stereo-camera is triggered to take two images (left and right) from the scene. Using triangulation the 3D position of the point is obtained from its images in two cameras.

3. Homogeneous transformation estimate.

   In this stage, firstly RANSAC can be used to remove outliers from the point sets ${}^{l}P_{i}^{\alpha_j}$ and ${}^{c}P_{i}^{\alpha_j}$. The valid ${}^{l}P_{i}^{\alpha_j}$ and ${}^{c}P_{i}^{\alpha_j}$ are used in the Eq. 5.7 which

is a least-squares solution to find ${}^{C}R_{L(\alpha_j)}$ and ${}^{C}t_{L(\alpha_j)}$ based on singular value decomposition (SVD) as is described in [KSAB87].

B) Obtaining ${}^{I}T_{L(\alpha)}$:

In order to obtain the transformation between $\{L(\alpha)\}$ and the center of rotation of the LRF, $\{I\}$, the following geometrical concepts have been used, which are summarized in the Fig. 5.11-b. Consider the three points O_0, O_1 and O_2 as the origins for $\{L(\alpha_0)\}$, $\{L(\alpha_1)\}$ and $\{L(\alpha_2)\}$, respectively. These points define a triangle in a 3D space. As is shown in the figure, the center for the circumcircle of such a triangle is as well the center of rotation for the LRF, which has been named $\{I\}$. Therefore, the radius of this circle which is also the distance d between $\{I\}$ and $\{L(\alpha)\}$ can be obtained by

$$d = \frac{|O_0\,O_1||O_1\,O_2||O_2\,O_0|}{4|\triangle O_0\,O_1\,O_2|} \tag{5.9}$$

where $\triangle$ denotes the area of the triangle. Finally, the transformation matrix ${}^{I}T_{L(\alpha)}$ can be described as

$${}^{I}T_{L(\alpha)} = \begin{bmatrix} cos(\alpha) & 0 & sin(\alpha) & d\,sin(\alpha) \\ 0 & 1 & 0 & 0 \\ -sin(\alpha) & 0 & cos(\alpha) & d\,cos(\alpha) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.10}$$

C) Calculating ${}^{L(\alpha)}T_{C}$:

Lets consider the transformation matrix from $\{L\}$ to $\{C\}$ as

$${}^{C}T_{L(\alpha)} = {}^{C}T_{I}\,{}^{I}T_{L(\alpha)} \tag{5.11}$$

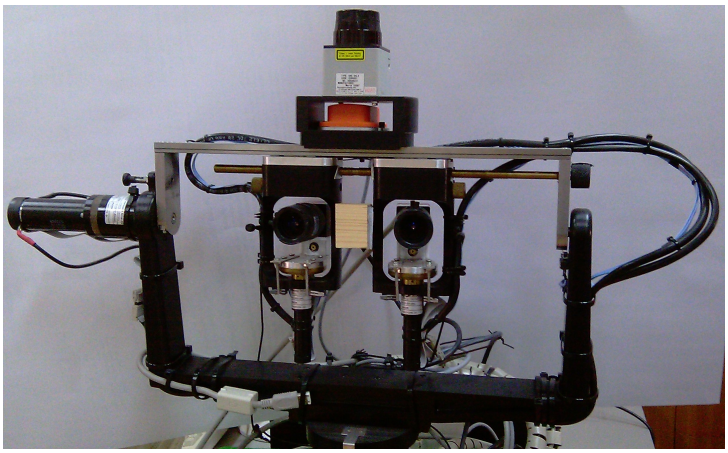


Figure 5.12: Setup used in the experiments

where ${}^{C}T_{I}$ corresponds to the transformation between the $\{C\}$ and the center of rotation $\{I\}$. In order to obtain ${}^{C}T_{L(\alpha)}$, the transformation ${}^{C}T_{I}$ has to be beforehand obtained. Eq. (5.11) represents a homogeneous transformation which is defined for each α angle. Therefore, ${}^{C}T_{L(\alpha)}$ can be replaced by the already estimated ${}^{C}T_{L(\alpha_0)}$ (obtained in the previous subsection). On the other hand, ${}^{I}T_{L(\alpha)}$ can be replaced by the matrix in Eq. (5.10) (by concerning $\alpha = \alpha_0$ in this equation). Having these the ${}^{C}T_{I}$ is obtained as:

$$ {}^{C}T_{I} = {}^{C}T_{L(\alpha_0)} \, {}^{I}T^{-1}_{L(\alpha_0)} \tag{5.12} $$

Once ${}^{C}T_{I}$ has been obtained, the desired transformation ${}^{C}T_{L(\alpha)}$ can be estimated according to Eq. (5.11).

5.3.4 Experiments

The proposed approach is tested using the sensor platform shown in Fig. 5.12. The dimensions of the provided images by the cameras are 320×240 pixels. The LRF mounted on the tilt unit is an Hokuyo URG-04LX [hok], a compact laser sensor which has a resolution of 0.36°and the field of view of 240°. Furthermore,

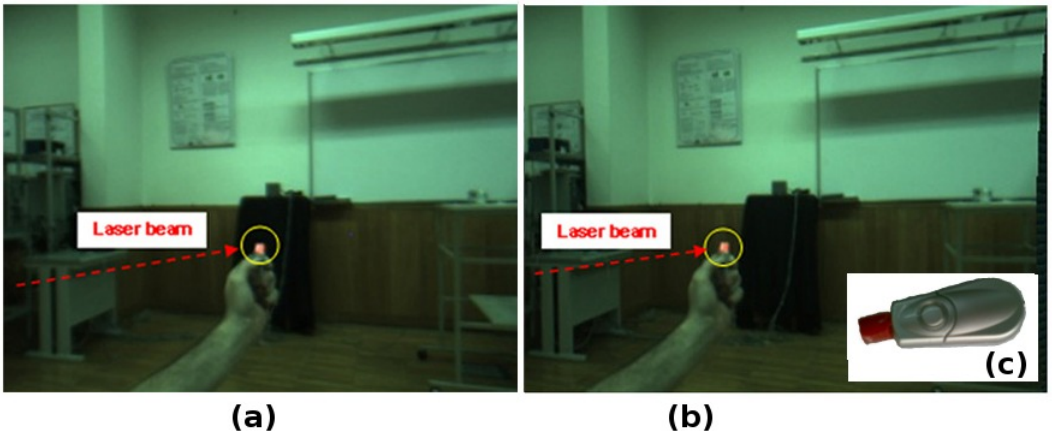


Figure 5.13: a) and (b) Real example (left and right images) for one step in collecting corresponding points in stereo camera and tilt-LRF with the $\alpha = 23.2^o$. (c) A simple laser pointer with a red-color plastic is the only calibration object.

an Xsens-MTi[xse] inertial sensor is strongly coupled to the LRF.

Based on the described procedure, a set of virtual 3D points has been generated (using a low cost bright spot shown in Fig. 5.13-c) while the LRF is placed in three different planes. These planes correspond to the angles $\alpha = 12.1^o$, $\alpha = 23.2^o$ and $\alpha = 50^o$ for the tilt (the angles are measured using the IS). Fig. 5.13 illustrates a real stereo capture, where the virtual point has been superimposed ($\alpha = 23.2^o$). The acquired corresponding data set is used to estimate the extrinsic parameters among the LRF and stereo camera based on the proposed method.

Fig. 5.15 shows the projection of the range data onto the camera image for three exemplary tilt angles (green, red and blue points respectively correspond to $\alpha_0 = 2^o$, $\alpha_1 = 12^o$ and $\alpha_2 = 23.2^o$). In another experiment, a full 3D range data and also an image are taken from a mannequin and then the 3D laser range data are reprojected on the image (see Fig. 5.16). The reprojection error values, in pixels, according to the number of 3-D points used in the method are presented in Fig. 5.14. As can be seen the error of the proposed calibration method decreases when the number of corresponding points increases.

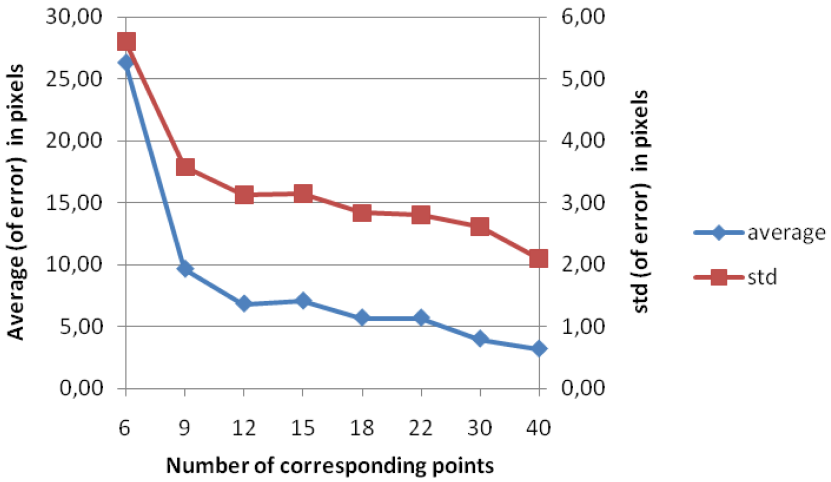
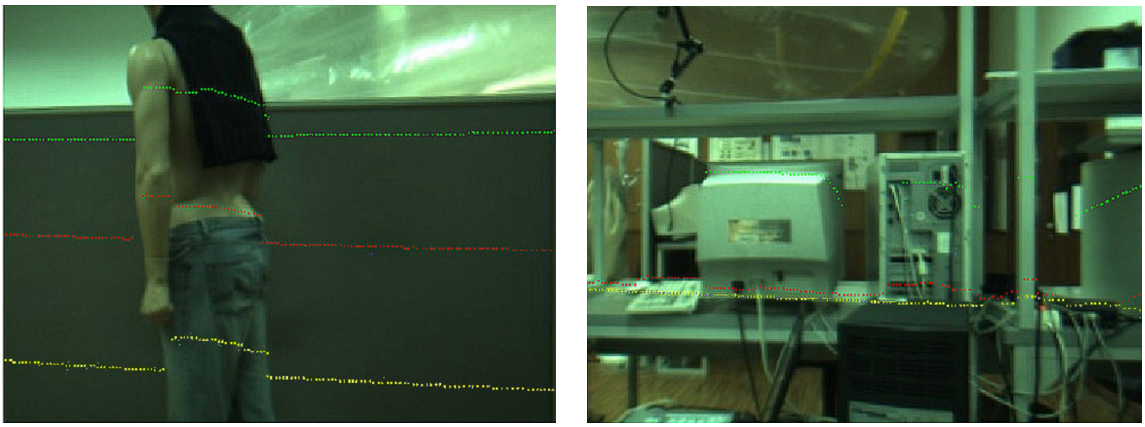


Figure 5.14: Evaluation of LRF-stereo camera calibration method with respect to number of used corresponding points in the experiments. The average of absolutes and the standard deviations are plotted.



(a) (b)

Figure 5.15: Scan data acquired by the laser range finder in three different planes (green, red and yellow points correspond to $\alpha_0 = 2^o$, $\alpha_1 = 12^o$ and $\alpha_2 = 23.2^o$, respectively) are reprojected onto the left images of two different scenarios.

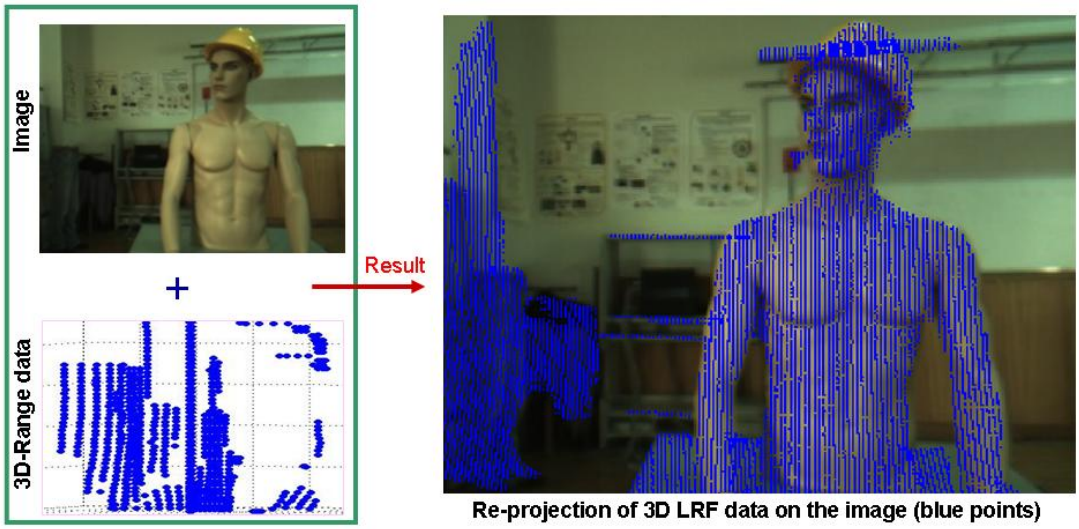


Figure 5.16: Reprojection of 3D range data on the image using the proposed calibration method between LRF and camera.

5.4 Low-level data filtering and tracking using Bayesian approach

The 3D data registration framework was previously introduced. The intention of such a framework is to provide low-level data registration which could be used by different applications. Rather than providing such a data in an static fashion, one can also consider the dynamic of a scene using a filtering approach.

5.4.1 Concept of Bayesian filtering

Bayesian technique is one of the classical approaches which is used to probabilistically estimate the state of a dynamic system from noisy observations. Basically, signals obtained by sensors carry information related to some physical phenomenon. As a matter of fact, the acquired signals are noisy and, moreover, the relationship of mapping between the state and observations (the model of the signal) is never known precisely. Hence, in order to infer the true state of nature, it is necessary to find the most appropriate model to describe the obtained data, and then estimate its parameters. The random nature of noise as well as uncertainty associated with the model can make it extremely difficult to determine what exactly is occurring. In order to deal with uncertainties and dynamic we turn to a method which originates from the 18th century mathematician T. Bayes [Bay63] [Pun99].

Suppose that I denotes all relevant background knowledge, "$sensory\ input$" denotes the observations from the sensors and $state_i$ denotes the probability of the state i , between the state space S, $S = \{state_1, state_2, \ldots, state_n\}$, which is interesting to be known. Then based on the Bayesian theory we will have:

$$p(state_i|sensory\ input, I) = \frac{p(sensory\ input|state_i, I).p(state_i|I)}{p(sensory\ input|I)} \tag{5.13}$$

in which $p(sensory\ input|state_i, I)$ stands for the likelihood, $p(state_i|I)$ comes from the previous knowledge of the $state_i$ (without implying new sensors observations) and $p(sensory\ input|I)$ is known as the evidence and actually is a normalizing constant which can be written as an integral and then:

$$p(state_i|sensory\ input, I) = \frac{p(sensory\ input|state_i, I).p(state_i|I)}{\sum_{j=1}^{n} p(sensory\ input|state_j, I).p(state_j|I)} \tag{5.14}$$

As can be seen in the equation 5.14, which is known as the Bayesian equation, one can calculate the probability of a hypothesis ($state_i$) based on both the observations and the previous state of that hypothesis. This equation can be turned up in a recursive form to perform a Bayesian filtering. Suppose x_t is the state of our system at the time t which can encompass e.g. the position of a person or an object. Also suppose that $z_{0:t}$ denotes the observations of the system from the times 0 to t . Then using Bayesian rule, the posterior distribution of x_t can be represented as:

$$p(x_t|z_{0..t}) = p(x_t|z_{0..t-1}, z_t) = \frac{p(z_t|x_t, z_{0..t-1})p(x_t|z_{0..t-1})}{p(z_t|z_{0..t-1})} \tag{5.15}$$

in which $p(z_t|z_{0..t-1})$ can be considered as a normalization factor, α, then:

$$= \alpha p(z_t|x_t, z_{0..t-1})p(x_t|z_{0..t-1}) \tag{5.16}$$

Using the Markov assumption in which x_t has all of information of $0..t-1$, so the observations $z_{0..t-1}$ must be also inside x_t, so:

$$p(z_t|x_t, z_{0..t-1}) = p(z_t|x_t) \tag{5.17}$$

then:

$$p(x_t|z_{0..t}) = \alpha\, p(z_t|x_t)\, p(x_t|z_{0..t-1}) \tag{5.18}$$

where $p(z_t|x_t)$ is the likelihood function and $p(x_t|z_{0..t-1})$ can be considered as a predictive step to predict the current state of x_t based on the all previous observations. Using marginalization we can insert the term x_{t-1}in a part of the equation 5.18:

$$p(x_t|z_{0..t-1}) = \int p(x_t|x_{t-1}, z_{0..t-1})p(x_{t-1}|z_{0..t-1})dx_{t-1} \tag{5.19}$$

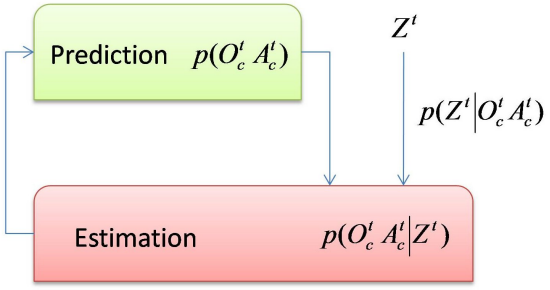


Figure 5.17: Two stages in BOF to estimate occupancy and velocity distribution

Here again using the Markov assumption we would have:

$$p(x_t|x_{t-1}, z_{0..t-1}) = p(x_t|x_{t-1}) \tag{5.20}$$

and consequently:

$$p(x_t|z_{0..t-1}) = \int p(x_t|x_{t-1}) p(x_{t-1}|z_{0..t-1}) dx_{t-1} \tag{5.21}$$

and finally using that to rewrite the equation 5.18:

$$p(x_t|z_{0..t}) = \alpha\, p(z_t|x_t) \int p(x_t|x_{t-1})\, p(x_{t-1}|z_{0..t-1}) dx_{t-1} \tag{5.22}$$

in which $p(x_t|x_{t-1})$ is the model of system (or the state transition model), $p(x_{t-1}|z_{0..t-1})$ is the prior distribution (which can be considered as the posterior of the previous step) and $p(z_t|x_t)$ is called perceptual model, likelihood function or sensor model. The equation (5.22), which is a typical formulation of the Bayesian filtering, presents how to compute the state of x at the moment t having the observations from the period of $[0 \ldots t]$ by using just the previous state of x (x_{t-1}), and the last observation (z_t), in a recursive way.

5.4.2 Applying Bayesian Occupancy Filtering

2D Bayesian Occupancy Filter (BOF) is an approach suitable to perform low-level data filtering. As mentioned earlier, the idea is to have a framework which could be used by different applications. For this purpose the filtering process should be applied in a low-level form in order to preserve the data as much as possible for further applications. Thus we propose to apply a BOF [MMRL08] on each Euclidean virtual plane after having the data registered on them. BOF [CTML06, MMRL08, RM09] is a special implementation of the Bayesian filtering approach. It represents the environment as a two dimensional planar grid based decomposition. In such a grid, two probability distribution is considered for each cell. One is to indicate the occupancy probability distribution and the other to represent the velocity probability distribution of the cell. BOF recursively estimates the probability distributions of each cell using the sensor observation which can be thought as $p(X_t|Z_{0..t})$ in which, as it was described before, X is the system state and Z is the sensor observation. Analogue to the traditional filtering algorithms, BOF has also two stages to obtain the posterior distribution $p(X_t|Z_{0..t})$: prediction and estimation (see Fig. 5.17). In prediction stage a priori prediction of the state is computed by using the defined model, without counting on the current sensor observation. Then in the estimation stage, the posteriori distribution of the state is computed using the priori distribution and the current sensor observation. Here the used BOF model is based on the definition in [CTML06, MMRL08, RM09]. For a cell c^l in the l-th virtual plane of the framework, ${}^lc \in y$, we have the following variables:

- $A_{c^l} \subset y$: represents antecedent set for the cell c^l.
- $A^t_{c^l} \in A_{c^l} \subset y$: indicates the antecedent of the cell ${}^lc$ at the current step.
- $A^{t-1}_{c^l} \in A_{c^l} \subset y$: indicates the antecedent of the cell ${}^lc$ at the previous step.
- $O^t_{c^l}$: a boolean variable to indicate whether the cell c^l at the time t is occupied ($O^t_{c^l} = 1$) or not ($O^t_{c^l} = 0$).

- $Z_1, ..., Z_S$: represents measurements taken by S sensors.

Having these definitions, the joint distribution for the model is:

$$\begin{aligned} &P(A_{c^l}^{t-1}\, A_{c^l}^{t}\, O_{c^l}^{t}\, Z_1^t ... Z_S^t) = \\ &P(A_{c^l}^{t-1}) P(A_{c^l}^{t} | A_{c^l}^{t-1}) P(O_{c^l}^{t} | A_{c^l}^{t-1}) \prod_{i=1}^{S} P(Z_i^t | A_{c^l}^{t}\, O_{c^l}^{t}) \end{aligned} \tag{5.23}$$

where

- $P(A_{c^l}^{t-1})$ is the probability for a given neighbouring cell A_{c^l} to be antecedent of the cell c^l, belonging to the layer l, at the time $t-1$.
- $P(A_{c^l}^{t} | A_{c^l}^{t-1})$: for a cell c^l, it is the distribution over antecedents at time t.
- $P(O_{c^l}^{t} | A_{c^l}^{t-1})$: for a cell c^l, it is the distribution over over occupancy given the antecedents of c^l.
- $P(Z_i^t | A_{c^l}^{t}\, O_{c^l}^{t})$ is the observation model for sensor i.

As mentioned two main stages are involved in estimation of occupancy and velocity of the cells in the BOF (see Fig. 5.17). In any time, t, a prediction $P(O_{c^l}^{t} A_{c^l}^{t})$ of the system state's probability distribution is made as a priori. Then the predicted distribution is updated by using the current observation $\prod_{i=1}^{S} P(Z_i^t | A_{c^l}^{t} O_{c^l}^{t}$ which leads to have a estimation $P(O_{c^l}^{t} A_{c^l}^{t} | Z^t)$ of the system state's probability distribution.

5.4.3 Experiments on BOF and tracking

A set of experiments has been carried out where to prove the applicability of BOF in the registration framework. In Fig. 5.19, the left and right columns demonstrate

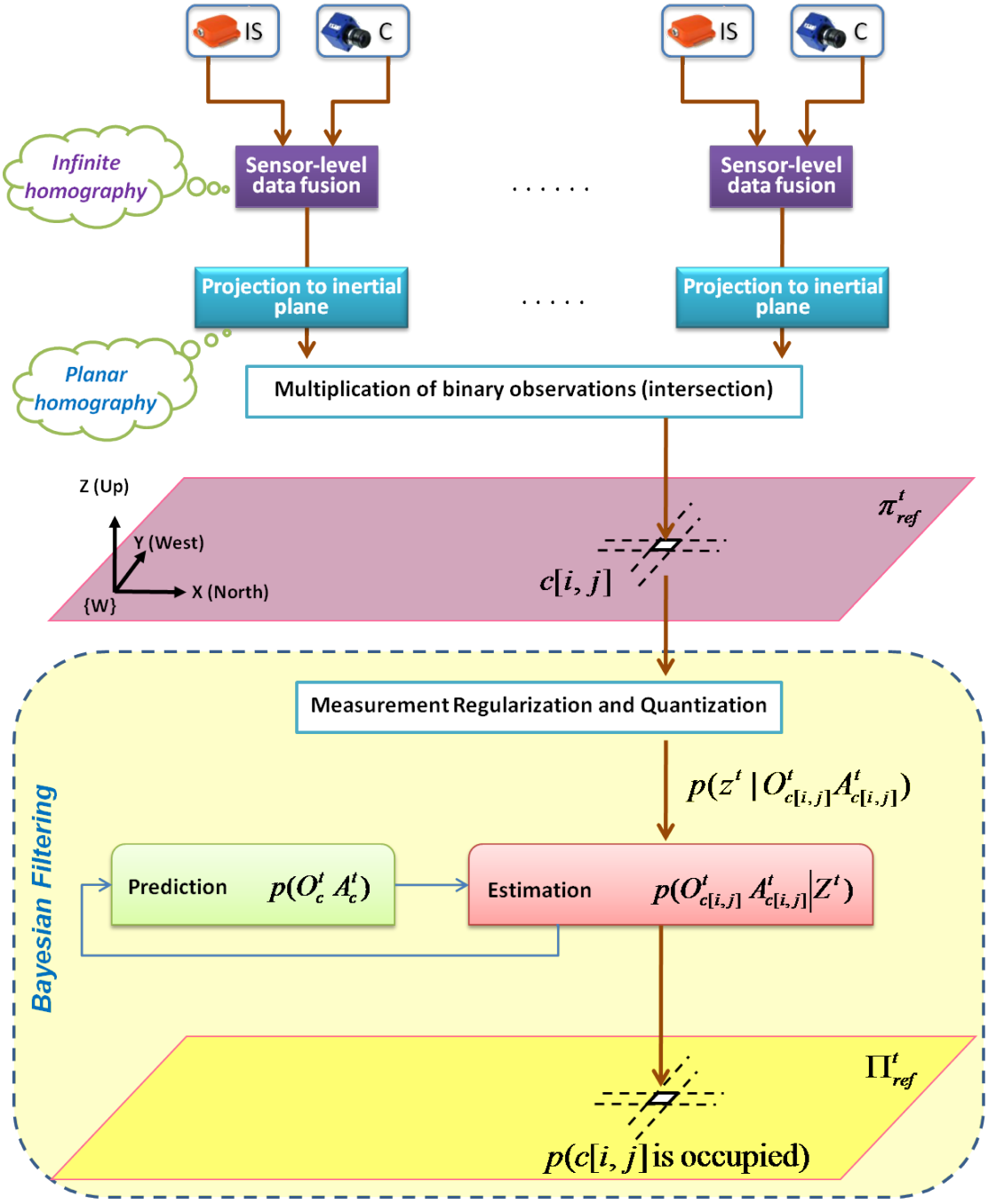


Figure 5.18: Applying Bayesian Occupancy Filtering to deal with the dynamic of scene in the proposed registration framework.

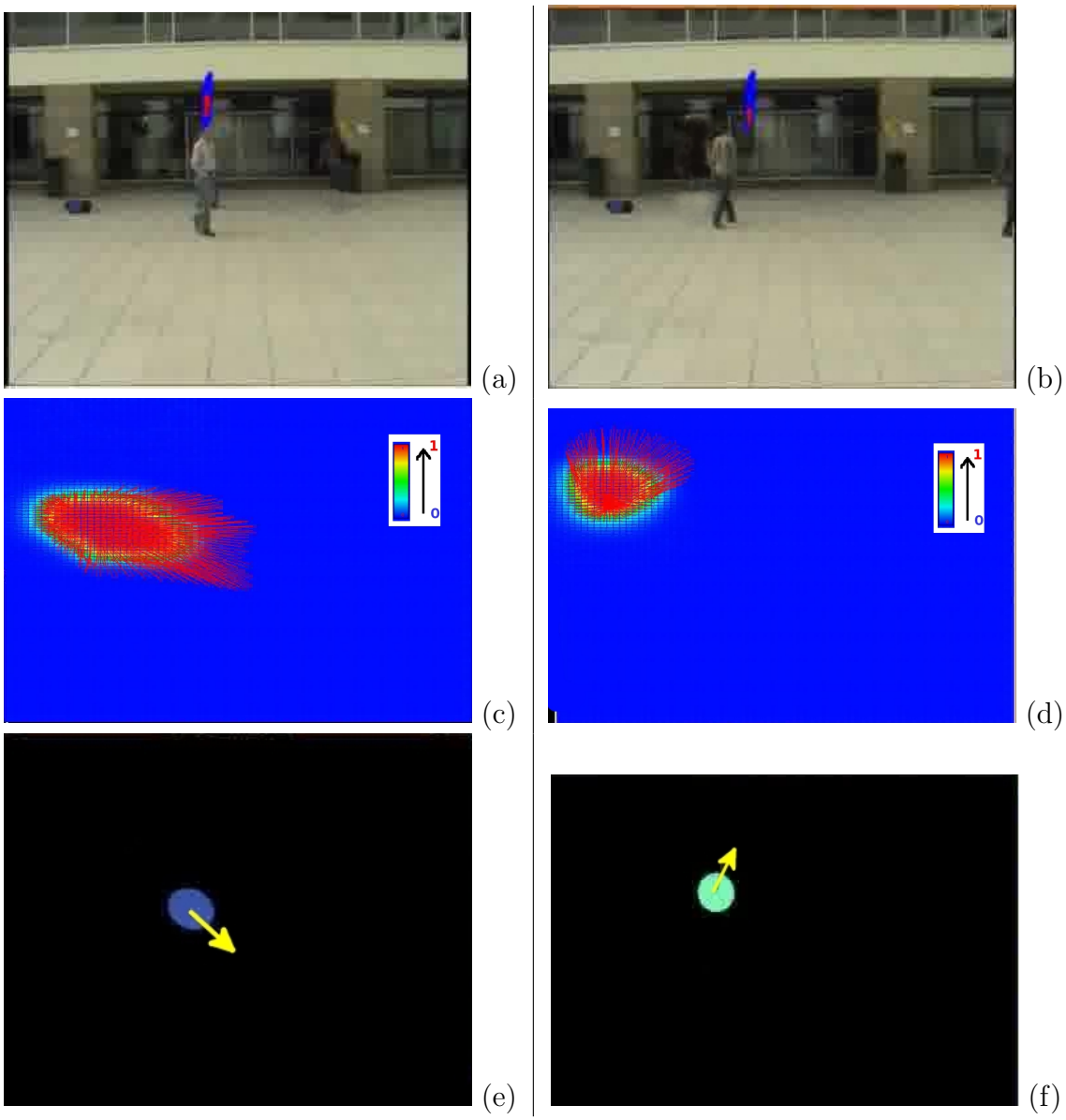


Figure 5.19: Applying Bayesian occupancy filtering and Tracking over the framework: The two left and right columns demonstrate two moments of the scene and system. The first row shows one of the three views . After performing background subtraction, the obtained silhouettes are used as inputs for the data registration framework. The output of the framework is the inertial-plane with registered data over which. The second row show the BOF applied over the inertial plane. The probabilities of being empty or being occupied for each cell is demonstrated by an spectrum from blue to red. The third row depicts a tracking applied over the BOF.

two moments of a scene where a person is walking in. For each moment three different views are used as the inputs to the data registration framework. After performing background subtraction the obtained silhouettes are projected onto the ground floor. Subsequently the binary intersections of the projected silhouettes are obtained and the result is fed to the low-level filtering algorithm (BOF). Fig. 5.19-c and Fig. 5.19-d depict the obtained occupancy grids respectively corresponding to Fig. 5.19-a and Fig. 5.19-b. Blue colour means that the probability of the corresponding cell to be occupied is zero and conversely the red color means that the corresponding cell has the highest probability (close to one) of being occupied. Moreover the movement direction for moving cells is characterized by red arrows where the arrow's size means the magnitude of the movement. Afterwards, using ProBT® library [prob] we have applied clustering and tracking algorithms over previously obtained low-level occupancy grids and the results are shown in Fig. 5.19-e and Fig. 5.19-f. In these figures the yellow arrows characterize the overall vector sum of the movement arrows of Fig. 5.19-c and Fig. 5.19-d.

5.5 Conclusion

In this chapter the coverage problem of cameras within a network was investigated, in the context of the proposed method. Using a geometric cost function, a genetic algorithm was proposed to find an optimal camera configuration in the network. Integration of mobile vision and laser sensor within a camera network was discussed and a method to estimate the extrinsic parameters among cameras and LRF was proposed. Eventually in this chapter it was shown that how the dynamic state of a scene can be taken into the account in the framework by using a Bayesian filtering approach.

Chapter 6

Overall Conclusions and Future Work

In this thesis we investigated the use of IS for 3D data registration by using a network of cameras and inertial sensors. 3D orientation provided by IS in each IS-camera couple was used to define a virtual camera. Moreover, the IS was used to define a set of Euclidean virtual planes in the scene. These Euclidean planes were used to register the data in the scene in 3D. Based on these, we presented a multi-sensor 3D data registration framework. A set of of experiments, in which some objects were reconstructed in off-line mode, were proposed. Additionally, geometric relations among different projective image planes and Euclidean inertial planes in the framework were investigated and for each particular case a parametric homography function was achieved.

Normally the volumetric reconstruction of a scene is time consuming due to the huge amount of data to be processed. The speed of the reconstruction process decreases with increasing the size and resolution of the volume to be reconstructed. Having a real-time reconstruction system is demanding for many applications. In order to achieve a real-time processing we proposed a parallelizing of the 3D reconstruction algorithm. Using GP-GPU and CUDA a prototype was built with ability to perform 3D reconstruction process with high speed. A set of comparisons was performed to demonstrate the performance of the system for different configurations. A large set of human postures and objects were reconstructed in 3D using this prototype.

In the proposed framework, thanks to IS, the rotations among all virtual cameras are relaxed. Therefore in aspect of having extrinsic parameters of the camera network, what remains is to have the translation vector among cameras. For an outdoor scenario the translation part can be obtained using a GPS coupled to each sensor. In this case, as mentioned in [KHJG11] the use of GPS can improve the accuracy of IS in its orientation angle until 0.01°. In this thesis, we took the advantage of having IS coupled to camera and proposed a novel method to estimate the extrinsic parameters (translation vector) among the cameras within the network. The proposed method estimates translation vectors among virtual cameras which can be used in cases of not having a coupled GPS available or us-

ing an indoor scenario. This method is upon on having the relative heights of two 3D points in the scene with respect to one of the cameras. Normally IS is error prone in sensing the 3D orientation. Effects of the IS noise in its 3D orientation measurements were simulated and analysed in this thesis. Apart of IS 3D orientation, some other parameters such as error in the height measurement of two 3D points, error in extraction of the coordinates of two 3D points in the images and the relative height (distance) of 3D points with respect to the cameras can effect the accuracy of the proposed translation estimation method. Effects of all these parameters were analysed by a generating thousands of data in simulation. The proposed method to translation recovery is fairly accurate and fast and does not need having a planar ground or a specific calibration pattern. This translation estimation approach has two requirements: The selected two 3D points have to be visible by all cameras and moreover their relative heights must be possible to measure. For many cases the first requirement can be satisfied like by hanging a simple string on the scene and marking two points on that. The second restriction can be eliminated by grouping the cameras within the network. It should be mentioned that in our experiments detection of the two points in the images has been interactively done.

Regarding the 3D reconstruction, as discussed in [MSEH08], in some circumstances a phenomenon called ghost can appear in the result. Ghost is an extra object which does not exist in the real scene but when there are some cases of visual ambiguities in the silhouettes it can be seen in the reconstructed scene [MSEH08]. In our experiments such a phenomenon did not occur and since the focus of the work was to prove the proposed concept we did not go through solving the problem of ghost phenomena. However one can refer to [MSEH08] where the authors proposed a technique to eliminate ghost objects from the result. In case of having possibility to segment silhouettes before feeding them to the algorithm, then for each segmented silhouette a separate instance of the proposed reconstruction algorithm can be ran. Another issue in the proposed 3D reconstruction algorithm is that it requires having intersection among coverages in the field of the views of all cameras. This drawback might be eliminated if instead of the proposed technique

some more sophisticated ones could be prepared to find the intersection among the views (e.g. using a probabilistic method instead of the proposed deterministic one).

The quality of reconstruction using a camera network depends to mainly three parameters: (1)-Number of cameras, (2)- The cameras configurations (e.g. positions) and (3)-The quality of the applied background subtraction technique. The first parameter is upon to the application and also the budget. In this thesis, we did not go through the details of background subtraction methods since we saw it a bit far from our problem. However the second parameter, the camera configuration, specifically their positions in the scene was investigated and a geometric method to find an optimal configuration was proposed using genetic algorithm.

Although vision is one of the essential modalities in data register and scene perception, having the further advantages of being passive sensors and of yielding additional information, such as surface colour, however it has some weaknesses such as error prone being in range sensing and data registration using these sensors is highly dependent on light conditions, shadows and homogeneous textures. In the other hand a precise active sensor like laser range finder is able to provide 3D information of a scene with a much lesser degree of dependency on texture, but they do not yield colour information. As a result, using range data as another strong modality in a synergistic manner can improve the process of 3D data registration. Based on this we took a primary step toward using these two modalities together and investigated the problem of extrinsic parameter estimation among them. Integration of range data within the proposed inertial-based data registration framework, using probabilistic technique remains as our future work.

The proposed framework intends to provide low-level data registration which could be used by different applications. Moreover than providing such a data in an static fashion, one can also consider the dynamic of the 3D data of a scene using a filtering approach. For the purpose of this framework we should perform the filtering process in a low-level form in order to preserve the data as much as possible for further applications. Supporting this, we demonstrated the possibility of

providing low-level data fusion by applying a Bayesian Occupancy Filtering which is able to deal with the dynamic of the scene and be useful for some applications such as tracking. As a future work we intend to investigate a multi-layer 3D tracking of articulated objects using Bayesian techniques. In this future investigation, we will provide a scientific contribution to model, predict and recognize the state of scene and to analyse the crowd behaviour using probabilistic approaches.

In the context of data registration it is important to have the uncertainty of each registered geometric entity. In the introduced framework we used homography transformations to map and register the data. For each IS-camera couple, the 3D orientation acquired from the IS and the translation vector, obtained by either an estimation method or GPS (in case of outdoor scenarios), are directly used to compute the homography transformations. Due to imperfection of sensors observation and estimation algorithms, the obtained homographies contain some uncertainties. As a consequence these uncertainties (of homographies) will get propagated to the points which are mapped through. To be aware of the degree of uncertainty for each data which is registered by an IS-camera couple is very important specially where a network of sensors is used. For this purpose, we also modelized the uncertainties of the mapped points in the framework by using statistical geometry analysis. In this thesis the fusion of the observations obtained from different nodes was performed by using a simple multiplication of the values (see Fig. 4.5). However having the uncertainties for the points, which are projected onto the Inertial Euclidean planes through different nodes, grants the possibility of using a more sophisticated and appropriate method, such as probabilistic approach, with ability of taking into the account the covariance matrix (uncertainty) of each point. As our future work we intend to develop the probabilistic registration and fusion of heterogeneous 3D data using the concept of homography by taking into the account the uncertainty of each hybrid-node.

Appendices

Appendix A

Mathematical notations

Throughout of this thesis, we use the convention listed in table A.1 for mathematical symbols:

Table A.1: Mathematical notations.

d, u, v	Scalars are typeset in regular lower-case
$\mathbf{t}$, $\mathbf{n}$, ...	Vectors are typeset in regular lower-case
$\mathbf{A}$, $\mathbf{B}$, ...	Matrices are typeset in boldface capitals. The matrices for homographies and rotations and also the camera calibration matrices are excepted from this rule.
$\mathbf{x}$, $\mathbf{y}$	Denote 2D points. If applicable, A right-subscript and a left-super-script denote the index and the system of reference, respectively.
$\mathbf{X}$, $\mathbf{Y}$	Denote 3D points. If applicable, A right-subscript and a left-super-script denote the index and the system of reference, respectively.
${}^{b}H_{a}$	A homography matrix which transform from system a to system b. It is typeset in regular capitals
${}^{b}R_{a}$	A rotation matrix which transform from system a to system b. It is typeset in regular capitals.
K	Camera calibration matrix, typeset in regular capital.
$\mathbf{I}_{m\times n}$	Identity matrix with $m \times n$ dimension.
$\mathbf{0}_{m\times n}$	Zero matrix with $m \times n$ dimension.
$diag(\mathbf{a})$	Diagonal matrix composed by a vector $\mathbf{a}$.
$\mathbf{\Sigma}$	Covariance matrix.
$\mathbf{J}$	Jacobian matrix.
δ	Standard deviation.
$\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$	The unit vectors of the X, Y and Z axes, respectively.
π	Denotes an Euclidean plane

Appendix B

Extrinsic parameter estimation among a 2D-LRF and a mono-camera

This appendix introduces a method to estimate geometric transformation among a 2D-LRF and a camera using 3D-2D pose estimation approach. Afterwards, the approach is extended for a case that a LRF is used to estimate the extrinsic parameters among a set of cameras within a network even with no overlap among their FOV.

Fig. B.1 shows the reference frames of a 2D-LRF and a camera, $\{L\}$ and $\{C\}$, respectively. The aim is to estimate the ${}^{C}R_{L}$ and ${}^{C}t_{L}$ which respectively indicate the rotation matrix and translation vector between $\{L\}$ and $\{C\}$. Given a set of observed (non-collinear) 3D points ${}^{L}X = \{{}^{L}X_i | \, i = 1..np\}$ by LRF expressed in $\{L\}$, np being number of correspondences in the set, the corresponding points in camera reference frame ${}^{C}X = \{{}^{C}X_i | \, i = 1..np\}$, are related by a rigid transformation such as:

$$ {}^{C}X = {}^{C}R_{L} \, {}^{L}X + {}^{C}\mathbf{t}_{L} \tag{B.1} $$

where ${}^{C}R_{L} = [\ \mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3\ ]^t$ and $\mathbf{t} = \begin{array}{cccc} t_x & t_y & t_z \end{array}$ are rotation matrix and translation vector, respectively. In the case of availability of corresponding 3D points in camera reference frame, we will have a set of corresponding pairs such as $({}^{L}X, {}^{C}X)$ which are related by Eq. B.1 where ${}^{C}R_{L}$ and ${}^{C}\mathbf{t}_{L}$ can be solved by applying a least-squares approach [KSAB87] to minimize the following function

$$ min_{R,t} \sum_{i=1}^{n} \left\| {}^{C}R_{L} \, {}^{L}X_i + {}^{C}\mathbf{t}_{L} \right\|^2 \tag{B.2} $$

This is known also as 3D-3D pose estimation or absolute orientation problem in the literatures[LHM00]. But the restriction in this approach is the need of reconstructing the 2D image points to 3D, which is not always possible. Thus, here we are interested to estimate the rotation matrix and translation vector by considering 3D points in the LRF reference frame and just 2D corresponding image points (mono-camera). This last case is known as 3D-2D pose estimation [LHM00, AD03] and for which we use a solution from Lu et al. [LHM00]. Their

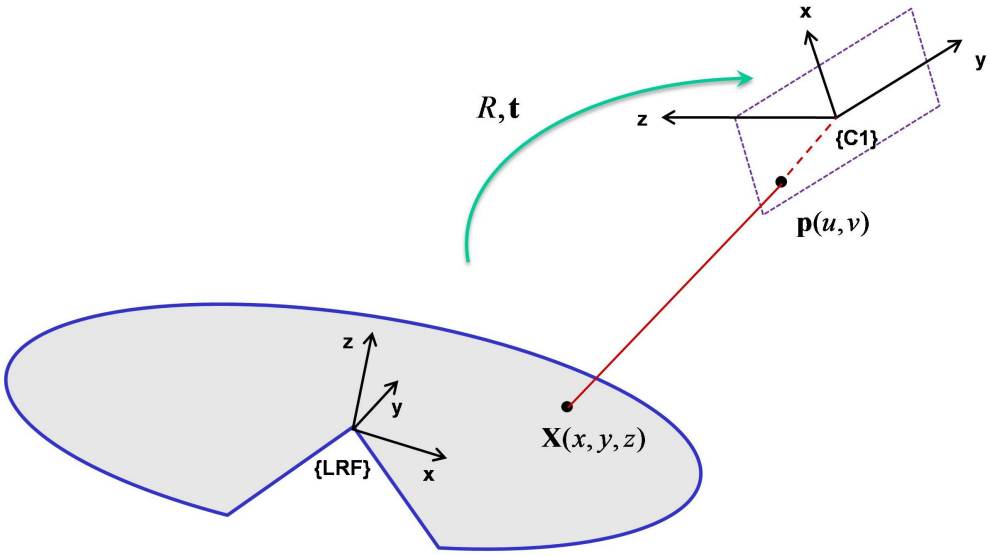


Figure B.1: 2D-LRF and a camera

approach assumes to have a normalized image plane which is defined as a plane with $z = 1$. Then the projection of 3D point ${}^{C}X_i$ (expressed in the camera reference) on the image plane will be $\mathbf{p_i} = (u_i, v_i, 1)^t$. Using collinearity of the center of projection, $\mathbf{p_i}$ and ${}^{C1}X_i$ we will have the following equations [LHM00]:

$$u_i = \frac{\mathbf{r}_1^t \, {}^{L}X_i + t_x}{\mathbf{r}_3^t \, {}^{L}X_i + t_z} \tag{B.3}$$

$$v_i = \frac{\mathbf{r}_2^t \, {}^{L}X_i + t_y}{\mathbf{r}_3^t \, {}^{L}X_i + t_z} \tag{B.4}$$

and the collinearity equation [LHM00] can be written as

$$\mathbf{v_i} = \frac{1}{\mathbf{r}_3^t \, {}^{L}X_i + t_z}(R \, {}^{L}X_i + \mathbf{t}) \tag{B.5}$$

Then the line-of-sight projection matrix can be defined as [LHM00]

$$\mathbf{V_i} = \frac{\mathbf{v_i}\,\mathbf{v_i^t}}{\mathbf{v_i^t}\mathbf{v_i}} \tag{B.6}$$

and an error vector can be defined as following [LHM00]

$$\mathbf{e}_i = (\mathbf{I} - \mathbf{V}_i)(R^L X + \mathbf{t}) \tag{B.7}$$

After, R and $\mathbf{t}$ can be estimated by minimizing the sum of the squared error over them, based on Lu's [LHM00]'s method:

$$E(R,\mathbf{t}) = \sum_{i=1}^{n} \|\mathbf{e}_i\|^2 = \sum_{i=1}^{n} \left\| (\mathbf{I} - \mathbf{V}_i)(R^L X + \mathbf{t}) \right\|^2 \tag{B.8}$$

and therefore, the transformation among LRF reference frame and camera center is estimated and it leads to have:

$$^{C}T_L = \begin{bmatrix} ^{C}R_L & ^{C}\mathbf{t}_L \\ \mathbf{0}_{1x3} & 1 \end{bmatrix} \tag{B.9}$$

B.0.0.1 Extension: Synergy of LRF to estimate extrinsic parameters in a camera network

This recently introduced method to estimate the transformation among a camera and 2D-LRF, can be extended to jointly estimate the extrinsic parameters in a network of cameras laser range finder. There is even an advantage that the cameras can have no overlap in their FOV. Fig. B.2 shows coordinate references of a 2D-LRF, $\{L\}$ and two cameras, $\{C1\}$ and $\{C2\}$. X^1 indicates 3D points which are observed by the LRF and C1, and X^2 indicates 3D point which are visible and common for the LRF and C2. This novel approach is suitable also for a camera network even if there is not any overlap between cameras, provided that the 2D-LRF could span its FOV to the cameras. Figure B.3 shows a exemplary scenario in

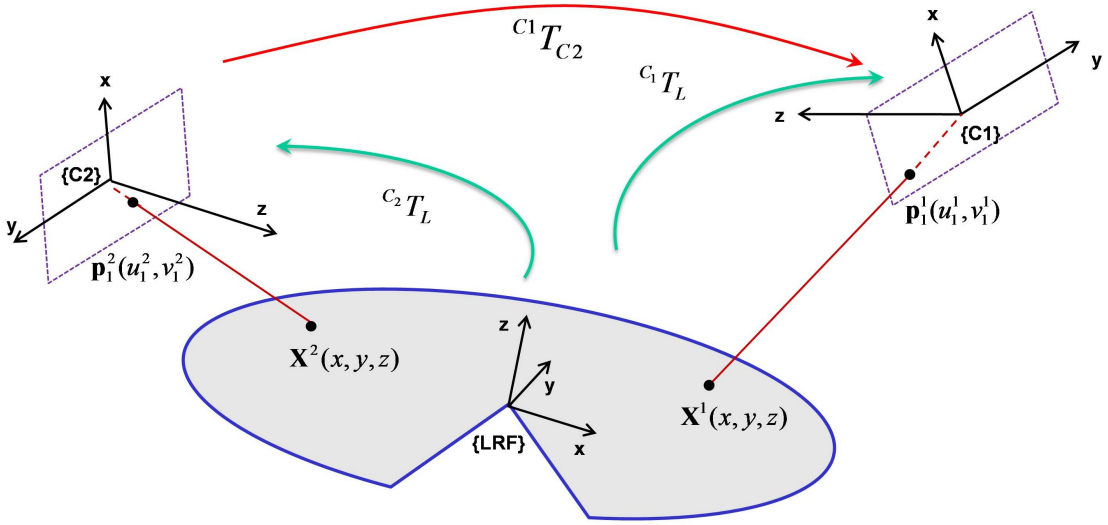


Figure B.2: 2D-LRF and two cameras

which the LRF mounted on a mobile agent is used to calibrate a distributed camera network where some of them do not have any overlap with the other cameras. Note that the Fig. B.3 is for the case where having an overlap between cameras' FOV is not necessary. If the cameras C1 and C2 have a common FOV, then the 3D points X^1 and X^2 can be coincided. Then the same early mentioned approach can be used to estimate the transformations between LRF and C2, namely$^{C2}T_L$. Then the transformation between C2 and C1 can be easily expressed as:

$$^{C1}T_{C2} = {}^{C1}T_L \, {}^{C2}T_L^{-1} \tag{B.10}$$

and obviously it can be repeated for any other cameras in the network. It means that we already have performed the conjugate calibration of the camera network and 2D-LRF.

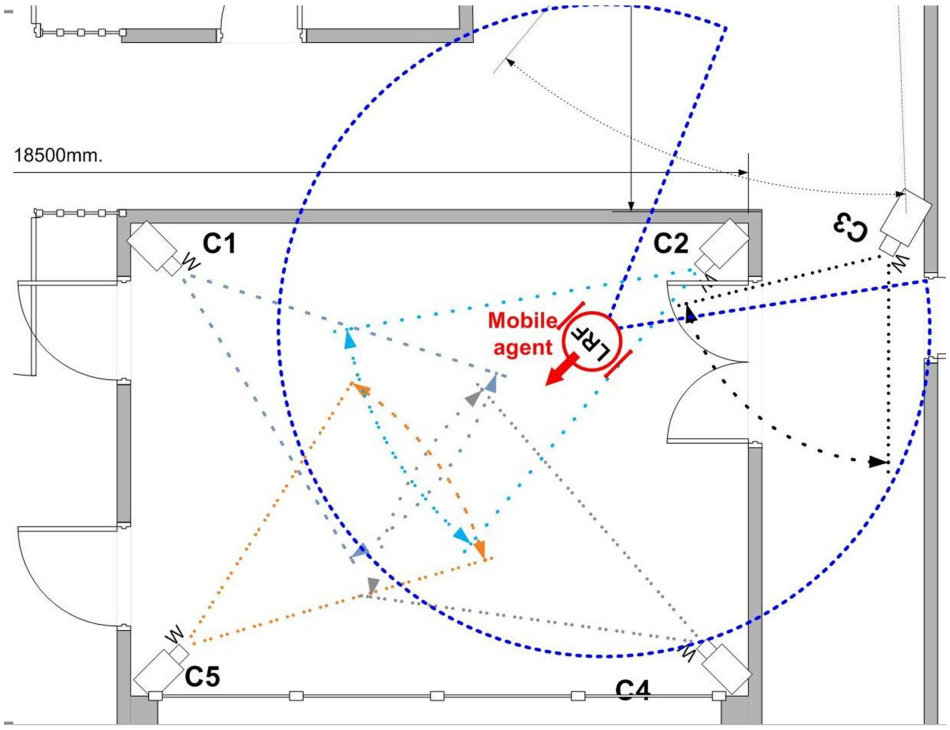


Figure B.3: Scheme of a camera network and a LRF equipped robot agent: As seen, although C3 does not have any overlap with the rest of cameras, but thanks to the proposed approach, the camera network and LRF can be calibrated.

Chapter 7

Bibliography

Bibliography

[AAMD11] Hadi Aliakbarpour, Luis Almeida, Paulo Menezes, and Jorge Dias. Multi-sensor 3d volumetric reconstruction using cuda. Journal of 3D Research, Springer., 2:1–14, 2011. 10.1007/3DRes.04(2011)6.

[Aba09] F. Ababsa. Toward a real-time 3d reconstruction system for urban scenes using georeferenced and oriented images. In Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on, volume 1, pages 75 –79, 2009.

[AD03] Adnan Ansar and Kostas Daniilidis. Linear pose estimation from points or lines. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 25:578–589, 2003.

[AD10a] Hadi Aliakbarpour and Jorge Dias. Human silhouette volume reconstruction using a gravity-based virtual camera network. In Proceedings of the 13th International Conference on Information Fusion, 26-29 July 2010 EICC Edinburgh, UK, 2010.

[AD10b] Hadi Aliakbarpour and Jorge Dias. Imu-aided 3d reconstruction based on multiple virtual planes. In DICTA'10 (the Australian Pattern Recognition and Computer Vision Society Conference), IEEE Pr., 1-3 December 2010, Sydney, Australia., 2010.

[AD11a] Hadi Aliakbarpour and Jorge Dias. Inertial-visual fusion for camera network calibration. In IEEE 9th International Conference on Industrial Informatics (INDIN 2011), July 2011., 2011.

[AD11b] Hadi Aliakbarpour and Jorge Dias. Multi-resolution virtual plane based 3d reconstruction using inertial-visual data fusion. In International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP 2011), 5-7 March 2011, Algarve, Portugal., 2011.

[AD11c] Hadi Aliakbarpour and Jorge Dias. Volumetric 3d reconstruction without planar ground assumption. In Distributed Smart Cameras (ICDSC), 2011 Fifth ACM/IEEE International Conference on, aug. 2011.

[AD12a] Hadi Aliakbarpour and Jorge Dias. 3d reconstruction based on multiple virtual planes by using fusion-based camera network. Journal of Computer Vision (IET), 2012.

[AD12b] Hadi Aliakbarpour and Jorge Dias. Geometric exploration of inertial-planes for multi-layer 3d data registration. Journal of ACM Transactions on Sensor Networks (TOSN), (submitted in 2012).

[AF08] N. Ashraf and H. Foroosh. Robust auto-calibration of a ptz camera with non-overlapping fov. In Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, pages 1 –4, 2008.

[AFKD10] Hadi Aliakbarpour, J. F. Ferreira, K. Khoshhal, and Jorge Dias. A novel framework for data registration and data fusion in presence of multi-modal sensors. In in Proceedings of DoCEIS2010- Emerging Trends in Technological Innovation, IFIP AICT 314-2010, Springer., volume 314/2010, pages 308–315, 2010.

[AFQ+11] Hadi Aliakbarpour, Paulo Freitas, JoÃ£o Quintas, Christiana Tsiourti, and Jorge Dias. Mobile robot cooperation with infrastructure for surveillance: Towards cloud robotics. In Workshop on Recognition and Action for Scene Understanding (REACTS) in the 14th International Conference of Computer Analysis of Images and Patterns (CAIP) , September 2011, Spain., 2011.

[AMD11] Luis Almeida, Paulo Menezes, and Jorge Dias. Stereo vision head vergence using gpu cepstral filtering. In Proceedings of the Fifth International Conference on Computer Vision Theory and Applications (VISAPP), Vilamoura, Algarve, Portugal, March 5-7, 2011., 2011.

[ANP+09] Hadi Aliakbarpour, Pedro Nunez, Jose Prado, Kamrad Khoshhal, and Jorge Dias. An efficient algorithm for extrinsic calibration between a 3d laser range finder and a stereo camera for surveillance. In 14th International Conference on Advanced Robotics (ICAR 2009), 2009.

[ATV09] Teresa C. S. Azevedo, Joao Manuel R. S. Tavares, and Mario A. P. Vaz. 3d object reconstruction from uncalibrated images using an off-the-shelf camera. Advances in Computational Vision and Medical Image Processing; in series of Computational Methods in Applied Sciences, Springer Netherlands, 13:117–136, 2009. Universidade do Porto.

[BAP+07] V. Brandou, A. G. Allais, M. Perrier, E. Malis, P. Rives, J. Sarrazin, and P. M. Sarradin. 3d reconstruction of natural underwater scenes using the stereovision system iris. In OCEANS 2007 - Europe, 2007.

[Bay63] T. Bayes. An essay towards solving a problem in the doctrine of chances. Biometrika, 45:293–315, 1763.

[BD04] Joao P. Barreto and Kostas Daniilidis. Wide area multiple camera calibration and estimation of radial distortion. In Int. Work. on Omnidirectional Vision, Camera Networks and Non-Classical Cameras. Prague, May 2004., 2004.

[BDK06] Simon Baker, Ankur Datta, and Takeo Kanade. Parameterizing homographies. Technical Report CMU-RI-TR-06-11, Robotics Institute, Pittsburgh, PA, March 2006.

[BES05] Michoud Brice, Guillou Erwan, and Bouakaz SaÃ¯da. Human model and pose reconstruction from multi-views. 2005.

[Bes09] Erkan Besdok. 3d vision by using calibration pattern with inertial sensor and rbf neural networks. Journal of Sensors, 9:4572–4585, 2009.

[BHK07] Yunsu Bok, Youngbae Hwang, and In So Kweon. Accurate motion estimation and high-precision 3d reconstruction by sensor fusion. In 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10-14 April 2007, 2007.

[Bjo96] Ake Bjorck. Numerical Methods for Least Squares Problems. SIAM (Society for Industrial and Applied Mathematics), 1996.

[Bou03] Jean-Yves Bouguet. Camera calibration toolbox for matlab. In www.vision.caltech.edu/bouguetj, 2003.

[BPC07] Silvain Beriault, Pierre Payeur, and Gilles Comeau. Flexible multi-camera network calibration for human gesture monitoring. In ROSE 2007 - IEEE International Workshop on Robotic and Sensors Environments, Ottawa - Canada, 12-13 October 2007., 2007.

[Bra98] Gary R. Bradski. Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal, (Q2), 1998.

[Bri08] Felicia Brisc. Accelerated volumetric reconstruction from uncalibrated camera views. PhD thesis, Dublin City University, 2008.

[BRSF09] J. P. Barreto, J. Roquette, P. Sturm, and F. Fonseca. Automatic camera calibration applied to medical endoscopy. In British Machine Vision Conference. London, September 2009, 2009.

[BS08] G. Bleser and D. Stricker. Advanced tracking through efficient image processing and visual-inertial sensor fusion. In Virtual Reality Conference, 2008. VR '08. IEEE, pages 137 –144, 2008.

[BWBS06] Bleser, Wohlleber, Becker, and Stricker. Fast and stable tracking for ar fusing video and inertial sensor data. pages 109–115. Short Papers Proceedings. Plzen: University of West Bohemia, 2006.

[BWP08] M. A. Brodie, A. Walmsley, and W. Page. The static accuracy and calibration of inertial measurement units for 3d orientation. Computer Methods in Biomechanics and Biomedical Engineering, 11:641–648, 2008.

[CF04] Xiaochun Cao and Hassan Foroosh. Easy camera calibration from inter image homographies. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 04), IEEE, 2004.

[CHY11] Yu-Tseh Chi, Jeffrey Ho, and Ming-Hsuan Yang. A direct method for estimating planar projective transform. In Proceedings of the 10th Asian conference on Computer vision - Volume Part II, ACCV'10, pages 268–281, Berlin, Heidelberg, 2011. Springer-Verlag.

[CLT06] R. Robert Clark, Michael H. Lin, and Colin J. Taylor. 3d environment capture from monocular video and inertial data. 2006.

[CPMH03] Zezhi Chen, Nick Pears, John McDermid, and Thomas Heseltine. Epipole estimation under pure camera translation. In Proc. VIIth Digital Image Computing: Techniques and Applications, Sydney., 2003.

[CPS05] Ondrej Chum, Tomas Pajdla, and Peter Sturm. The geometric error for homographies. Comput. Vis. Image Underst., 97(1):86–102, January 2005.

[CRZ99] A. Criminisi, I. Reid, and A. Zisserman. A plane measuring device. Image and Vision Computing, 17(8):625 – 634, 1999.

[CTML06] C. Chen, C. Tay, K. Mekhnacha, and C. Laugier. Dynamic environment modeling with gridmap: a multiple-object tracking application. 9th International Conference on Control, Automation, Robotics and Vision, 2006. ICARCV '06., 2006.

[Dev07] Dhanya Devarajan. DISTRIBUTED LOCALIZATION OF CAMERA NETWORKS. PhD thesis, Rensselaer Polytechnic Institute, Troy, New York., 2007.

[DKBN08] Joachim Dornauer, Gabriele Kotsis, Christian Bernthaler, and Michael Naderhirn. A comparison of different computer vision methods for real time 3d reconstruction for the use in mobile robots. In MoMM '08: Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia, pages 136–141, New York, NY, USA, 2008. ACM.

[DLA02] Jorge Dias, Jorge Lobo, and Luis A. Almeida. Cooperation between visual and inertial information for 3d vision. In Proceedings of the 10th Mediterranean Conference on Control and Automation - MED2002 Lisbon, Portugal, July 9-12, 2002., 2002.

[DR04] D. Devarajan and R. J. Radke. Distributed metric calibration of large camera networks. In BaseNets workshop in conjunction with BroadNets, Oct 25-27, 2004, San Jose, California., 2004.

[DR07] Dhanya Devarajan and Richard J. Radke. Calibrating distributed camera networks using belief propagation. EURASIP Journal on Applied Signal Processing, pages 221–221, 2007.

[DRC06] Dhanya Devarajan, Richard J. Radke, and Haeyong Chung. Distributed metric calibration of ad hoc camera networks. ACM Transactions on Sensor Networks (TOSN), 2:380–403, 2006.

[EWK07] Sandro Esquivel, Felix Woelk, and Reinhard Koch. Calibration of a multi-camera rig from non-overlapping views. In Fred Hamprecht, Christoph Schnorr, and Bernd Jahne, editors, Pattern Recognition, volume 4713 of Lecture Notes in Computer Science, pages 82–91. Springer Berlin - Heidelberg, 2007.

[FAD09] Diego R. Faria, Hadi Aliakbarpour, and Jorge Dias. Grasping movements recognition in 3d space using a bayesian approach. In 14th International Conference on Advanced Robotics (ICAR 2009), 2009.

[Fau] Olivier Faugeras. Three-Dimensional Computer Vision.

[FLD10] Joao Filipe Ferreira, Jorge Lobo, and Jorge Dias. Bayesian real-time perception algorithms on GPU — Real-time implementation of Bayesian models for multimodal perception using CUDA. Journal of Real-Time Image Processing, Special Issue, 26 February 2010.

[FMS+10] Tobias Feldmann, Ioannis Mihailidis, Sebastian Schulz, Dietrich Paulus, and Annika Worner. Online full body human motion tracking based on dense volumetric 3dÂ reconstructions from multi camera setups. In Rudiger Dillmann, Jurgen Beyerer, Uwe Hanebeck, and Tanja Schultz, editors, KI 2010, Advances in Artificial Intelligence, volume 6359 of Lecture Notes in Computer Science, pages 74–81. Springer Berlin - Heidelberg, 2010.

[FPD08] JoÃ£o Filipe Ferreira, CÃ¡tia Pinho, and Jorge Dias. Implementation and calibration of a bayesian binaural system for 3d localisation. In IEEE International Conference on Robotics and Biomimetics (ROBIO 2008), Bangkok, Tailand, December 14-17, 2008, 2008.

[Fre11] Linus Fredriksson. Evaluation of 3d reconstructing based on visual hull algorithms. Technical report, Faculty of Engineering and Sustainable Development, University of Gavle, 2011.

[GBZ08] R. Guerchouche, O. Bernier, and T. Zaharia. Multiresolution volumetric 3d object reconstruction for collaborative interactions. Pattern Recognition and Image Analysis, 18:621–637, 2008. 10.1134/S1054661808040147.

[GFP08] Li Guan, Jean-Sebastien Franco, and Marc Pollefeys. 3d object reconstruction with heterogeneous sensor data. In International Symposium on 3D Data Processing, Visualization and Transmission. INRIA, 2008.

[GKN10] Y. Gat, I. Kozintsev, and O. Nestares. Fusing image data with location and orientation sensor data streams for consumer video applications. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, pages 1 –8, 2010.

[GPC+10] Sigurjon Arni Guomundsson, Montse Pardas, Josep R. Casas, Johannes R. Sveinsson, Henrik AanÃ¦s, and Rasmus Larsen. Improved 3d reconstruction in smart-room environments using tof imaging. Computer Vision and Image Understanding, 114(12):1376–1384, 2010. Special issue on Time-of-Flight Camera Based Computer Vision.

[GRNG05] A. Griesser, S. D. Roeck, A. Neubeck, and L. V. Gool. Gpu-based foreground-background segmentation using an extended colinearity criterion. In In Proc. Vision, Modeling, and Visualization (VMV)

2005. Amsterdam, The Netherlands: IOS, Nov. 2005., pages 319–326, 2005.

[Hag89] William W. Hager. Updating the inverse of a matrix. SIAM Rev. 31,, pages 221–239, 1989.

[Heu04] Stephan Heuel. Uncertain Projective Geometry: Statistical Reasoning for Polyhedral Object Reconstruction. Springer, 2004.

[HGJ07] A. Hogue, A. German, and M. Jenkin. Underwater environment reconstruction using stereo and inertial data. In Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on, pages 2372–2377, 2007.

[HHK06] Alexander Hornung, Er Hornung, and Leif Kobbelt. Robust and efficient photo-consistency estimation for volumetric 3d reconstruction. In In ECCV, vol. II, pages 179–190, 2006.

[HL07] B.W. He and Y.F. Li. Camera calibration from vanishing points in a vision system. Optics & Laser Technology, Elsevier., 40:555–561, 2007.

[hok] www.hokuyo-aut.jp.

[HT06] Zhaozheng Hu and Zheng Tan. Depth recovery and affine reconstruction under camera pure translation. Pattern Recognition, Elsevier, 40:2826–2836, 2006.

[HYWH10] Hsiang-Wen Hsieh, Hung-Hsiu Yu, Chin-Chia Wu, and Jwu-Sheng Hu. Concurrent multiple cameras calibration and robot localization from visual and 3d inertial measurements. In SICE Annual Conference 2010, Proceedings of, pages 1914–1919, aug. 2010.

[HZ03] Richard Hartley and Andrew Zisserman. Multiple View Geometry in Computer Vision. CAMBRIDGE UNIVERSITY PRESS, 2003.

[Jet04] Manish Jethwa. Efficient Volumetric Reconstruction from Multiple Calibrated Cameras. PhD thesis, Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), 2004.

[JL06] Wei Jiang and Jian Lu. Panoramic 3d reconstruction by fusing color intensity and laser range data. In Robotics and Biomimetics, 2006. ROBIO '06. IEEE International Conference on, pages 947 –953, dec. 2006.

[KAHD] G. Kordelas, J. D. Perez-Moneo Agapito, J. M. Vegas Hernandez, and P. Daras. State-of-the-art algorithms for complete 3d model reconstruction. Technical report, Engage Summer School, Sept. 2010.

[Keh05] Roland Kehl. Markerless Motion Capture of Complex Human Movements from Multiple Views. PhD thesis, SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH, 2005.

[KH09] Stefan Kuhn and Dominik Henrich. Multi-view reconstruction of unknown objects within a known environment. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Yoshinori Kuno, Junxian Wang, Jun-Xuan Wang, Junxian Wang, Renato Pajarola, Peter Lindstrom, AndrÃ© Hinkenjann, Miguel EncarnaÃ§Ã£o, ClÃ¡udio Silva, and Daniel Coming, editors, Advances in Visual Computing, volume 5875 of Lecture Notes in Computer Science, pages 784–795. Springer Berlin-Heidelberg, 2009.

[Kha08] Saad M. Khan. Multi-view Approaches to Tracking, 3D Reconstruction and Object Class Detection. PhD thesis, School of Electrical Engineering and Computer Science, College of Engineering and Computer Science, University of Central Florida, Orlando, Florida., 2008. Major Professor: Mubarak Shah.

[KHJG11] Mahzad Kalantari, Amir Hashemi, Franck Jung, and Jean-Pierre Guedon. A new solution to the relative orientation problem using only 3 points and the vertical direction. Journal of Mathematical Imaging and Vision, 39:259–268, March 2011.

[KIFP08] R.K. Kumar, A. Ilie, J-M. Frahm, and M. Pollefeys. Simple calibration of non-overlapping cameras with a mirror. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1 –7, 2008.

[KK09] Daniel Knoblauch and Falko Kuester. Focused volumetric visual hull with color extraction. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Yoshinori Kuno, Junxian Wang, Renato Pajarola, Peter Lindstrom, Andre Hinkenjann, Miguel Encarnacao, Claudio Silva, and Daniel Coming, editors, Advances in Visual Computing, volume 5876 of Lecture Notes in Computer Science, pages 208–217. Springer Berlin-Heidelberg, 2009.

[KMB07] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. Pattern Recogn., 40:1106–1122, March 2007.

[KS08] S.M. Khan and M. Shah. Reconstructing non-stationary articulated objects in monocular video using silhouette information. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008.

[KSAB87] T. S. Huang K. S. Arun and S. D. Blostein. Least-squares fitting of two 3-d point sets. 1987.

[KT07] Masafumi Hashimoto Kazuhiko Takahashi, Yusuke Nagasawa. Remarks on 3d human body's feature extraction from voxel reconstruction of human body posture. pages 121–126. Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics, December 15 -18, 2007, Sanya, China., 2007.

[KTD+09] Young Min Kim, C. Theobalt, J. Diebel, J. Kosecka, B. Miscusik, and S. Thrun. Multi-view image and tof sensor fusion for dense 3d reconstruction. In Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, pages 1542 –1549, 272009-oct.4 2009.

[KYS07] Saad M. Khan, Pingkun Yan, and Mubarak Shah. A homographic framework for the fusion of multi-view silhouettes. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007.

[LAAD03] J. Lobo, L. Almeida, J. Alves, and J. Dias. Registration and segmentation for 3d map building - a solution based on stereo vision and inertial sensors. In Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, volume 1, pages 139 – 144 vol.1, 2003.

[LBN08] A. Ladikos, S. Benhimane, and N. Navab. Efficient visual hull computation for real-time 3d reconstruction using cuda. In Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on, pages 1 –8, jun. 2008.

[LBTV10] Xinghan Luo, Berend Berendsen, Robby T. Tan, and Remco C. Veltkamp. Human pose estimation for multiple persons based on volume reconstruction. In Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR '10, pages 3591–3594, Washington, DC, USA, 2010. IEEE Computer Society.

[LD07] Jorge Lobo and Jorge Dias. Relative pose calibration between visual and inertial sensors. International Journal of Robotics Research,

Special Issue 2nd Workshop on Integration of Vision and Inertial Sensors, 26:561–575, 2007.

[Leu05] Carlos Leung. Efficient Methods for 3D Reconstruction from Multiple Images. PhD thesis, University of Queensland, 2005.

[LH07] M. Labrie and P. Hebert. Efficient camera motion and 3d recovery using an inertial sensor. In Computer and Robot Vision, 2007. CRV '07. Fourth Canadian Conference on, pages 55 –62, May 2007.

[LHM00] Chien-Ping Lu, Gregory D. Hager, and Eric Mjolsness. Fast and globally convergent pose estimation from video images. 2000.

[LQD03] Jorge Lobo, Carlos Queiroz, and Jorge Dias. World feature detection and mapping using stereovision and inertial sensors. Robotics and Autonomous Systems, 44(1):69–81, 2003. Best Papers of the Eurobot '01 Workshop.

[LY08] Po-Lun Lai and Alper Yilmaz. Projective reconstruction of building shape from silhouette images acquired from uncalibrated cameras. In ISPRS Congress Beijing 2008, Proceedings of Commission III, 2008.

[LY10] Heewon Lee and Alper Yilmaz. 3d reconstruction using photo consistency from uncalibrated multiple views. In VISAPP 2010 - The International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2010.

[LZWnL04] Ye Lu, Jason Z. Zhang, Q. M. Jonathan Wu, and Ze nian Li. A survey of motion-parallax-based 3d reconstruction algorithms. IEEE Trans. on Systems, Man, and Cybernetics, 34:532–548, 2004.

[MBF09] Jochen Meidow, Christian Beder, and Wolfgang FÃ¶rstner. Reasoning with uncertain points, straight lines, and straight line segments in 2d. ISPRS Journal of Photogrammetry and Remote Sensing, 64(2):125–139, 2009.

[MBF12] Rui Melo, Joao P. Barreto, and Gabriel Falcao. A new solution for camera calibration and real-time image distortion correction in medical endoscopy - initial technical evaluation. IEEE Trans. in Biomedical Engineering, 2012.

[MD07] Luiz G. B. Mirisola and Jorge M. M. Dias. Exploiting inertial sensing in mosaicing and visual navigation. In In 6th IFAC Symposium on Inteligent Autonomous Vehicles (IAV07), Toulouse, France, Sep. 2007., 2007.

[MDdA07] Luiz G. B. Mirisola, Jorge Dias, and A. Traca de Almeida. Trajectory recovery and 3d mapping from rotation-compensated imagery for an airship. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems San Diego, CA, USA, Oct 29 - Nov 2, 2007, 2007.

[MGB07] Brice Michoud, Erwan Guillou, and Saida Bouakaz. Real-time and markerless 3d human motion capture using multiple views. Human Motion-Understanding, Modeling, Capture and Animation, Springer Berlin/Heidelberg., 4814/2007:88–103, 2007.

[Mir09] Luiz Gustavo Bizarro Mirisola. Exploiting attitude sensing in vision-based navigation, mapping and tracking including results from an airship. PhD thesis, 2009.

[MLM07] P.B.L. Meijer, C. Leistner, and A. Martiniere. Multiple view camera calibration for localization. In Distributed Smart Cameras, 2007. ICDSC '07. First ACM/IEEE International Conference on, 2007.

[MMRL08] Kamel Mekhnacha, Yong Mao, David Raulo, and Christian Laugier. Bayesian occupancy filter based "fast clustering-tracking" algorithm. In IROS 2008, 2008.

[MOS07] M. Meingast, Songhwai Oh, and S. Sastry. Automatic camera network localization using object image tracks. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pages 1–8, oct. 2007.

[MP10] B. Micusik and R. Pflugfelder. Localizing non-overlapping surveillance cameras under the l-infinity norm. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 2895–2901, june 2010.

[MR06] C. Mei and P: Rives. Calibration between a central catadioptric camera and a laser range finder for robotic applications. Proceedings of the IEEE International Conference on Robotics and Automation, 2006.

[MR08] F.M. Mirzaei and S.I. Roumeliotis. A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. Robotics, IEEE Transactions on, 24(5):1143–1156, oct. 2008.

[MRL] Mrl, http://paloma.isr.uc.pt/mrl/.

[MSD08] M. Maitre, Y. Shinagawa, and M.N. Do. Symmetric multi-view stereo reconstruction from planar camera arrays. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8, June 2008.

[MSEH08] Brice Michoud, Bouakaz Saida, Guillou Erwan, and Briceno Hector. Largest silhouette-equivalent volume for 3d shapes modeling without ghost object. In M2SFA2 2008: Workshop on Multi-camera and Multi-modal Sensor Fusion, Marseille, France., 2008.

[NNT07] Christian Nitschke, Atsushi Nakazawa, and Haruo Takemura. Real-time space carving using graphics hardware. IEICE - Trans. Inf. Syst., E90-D:1175–1184, August 2007.

[NPG05] S. Negahdaripour, R. Prados, and R. Garcia. Planar homography: accuracy analysis and applications. In Image Processing, 2005. ICIP 2005. IEEE International Conference on, volume 1, pages 1089–92, sept. 2005.

[Nvi] Nvidia. http://www.nvidia.com/.

[OB06] Benjamin Ochoa and Serge Belongie. Covariance propagation for guided matching. In Statistical Methods in Multi-Image and Video Processing (SMVP) 2006, 2006.

[OD02] T. Okatani and K. Deguchi. Robust estimation of camera translation between two images using a camera with a 3d orientation sensor. In Pattern Recognition, 2002. Proceedings. 16th International Conference on, volume 1, pages 275 – 278 vol.1, 2002.

[Ope] Opencv. http://opencv.willowgarage.com/.

[PD03] F. Porikli and A. Divakaran. Multi-camera calibration, object tracking and query generation. In Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on, volume 1, pages I – 653–6 vol.1, july 2003.

[Proa] Prosilica, http://www.1stvision.com/cameras/prosilica/gc650-gc650c.html.

[prob] www.probayes.com.

[Pun99] Olena Punska. Bayesian Approaches to Multi-Sensor Data Fusion. PhD thesis, University of Cambridge, Signal Processing and Communications Laboratory, 1999.

[RBN10] Rui Rodrigues, Joao P. Barreto, and Urbano Nunes. Camera pose estimation uisng images of planar mirror reflections. In European Conference on Computer Vision. Heraklion, September 2010., 2010.

[RGSN08] D.I.B. Randeniya, M. Gunaratne, S. Sarkar, and A. Nazef. Calibration of inertial and vision systems as a prelude to multi-sensor fusion. Transportation Research Part C: Emerging Technologies, 16:255–274, 2008.

[RKR+08] C. Ruwwe, B. Keck, O. Rusch, U. Zolzer, and X. Loison. Image registration by means of 3d octree correlation. In Multimedia Signal Processing, 2008 IEEE 10th Workshop on, pages 515 –519, oct. 2008.

[RM09] Julien Ros and Kamel Mekhnacha. Multi-sensor human tracking with the bayesian occupancy filter. In IEEE Proceedings of the 16th international conference on Digital Signal Processing, 2009.

[SBS08] Florian Schweiger, Ingo Bauermann, and Eckehard Steinbach. Joint calibration of a camera triplet and a laser rangefinder. In IEEE 2008, 2008.

[SCD+06] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 1, pages 519 – 528, 2006.

[SHS07] D. Scaramuzza, A. Harati, and R. Siegwart. Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, 2007.

[SHT+08] S.S. Stone, J.P. Haldar, S.C. Tsao, W. m.W. Hwu, B.P. Sutton, and Z.-P. Liang. Accelerating advanced mri reconstructions on gpus. Journal of Parallel and Distributed Computing, 68(10):1307 – 1318, 2008. General-Purpose Processing using Graphics Processing Units.

[Sin09] Sudipta N. Sinha. Silhouettes for Calibration and Reconstruction from Multiple Views. PhD thesis, University of North Carolina, Chapel Hill, 2009.

[SL09] Matthijs T.J. Spaan and Pedro U. Lima. A decision-theoretic approach to dynamic sensor selection in camera networks. In Proc. of ICAPS 2009 - 19th International Conference on Automated Planning and Scheduling, Thessaloniki, Greece, 2009, 2009.

[SMP05] Tomas Svoboda, Daniel Martinec, and Tomas Pajdla. A convenient multi-camera self-calibration for virtual environments. PRESENCE: Teleoperators and Virtual Environments, MIT Press., 14:407–422, 2005.

[SMP06] T. Svoboda, D. Martienc, and T. Pajdla. A convenient multi-camera self-calibration for virtual environments , vol. 14(4), pp. 407–422, 2004. PRESENCE: Teleoperators and Virtual Environments, Massachusetts Institute of Technology., 14:407–422, 2006.

[SZB+07] Mario Sormann, Christopher Zach, Joachim Bauer, Konrad Karner, and Horst Bishof. Watertight multi-view reconstruction based on volumetric graph-cuts. In Bjarne Ersball and Kim Pedersen, editors, Image Analysis, volume 4522 of Lecture Notes in Computer Science, pages 393–402. Springer Berlin, Heidelberg, 2007.

[TWH+07] Cheng-Yuan Tang, Yi-Leh Wu, Pei-Ching Hu, Hsien-Chang Lin, and Wen-Chao Chen. Self-calibration for metric 3d reconstruction using homography. In IAPR CONFERENCE ON MACHINE VISION APPLICATIONS MVA2007, Institute of Industrial Science (IIS), The University of Tokyo., 2007.

[Uri05] Maria Cruz Villa Uriol. Video-Based Avatar Reconstruction and Motion Capture. PhD thesis, Electrical and Computer Engineering, UNIVERSITY OF CALIFORNIA, IRVINE., 2005.

[VBN12] Francisco Vasconcelos, Joao P. Barreto, and Urbano Nunes. A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder. IEEE Trans. in Pattern Analysis and Machine Intelligence, 2012.

[VVG06] Maarten Vergauwen and Luc Van Gool. Web-based 3d reconstruction service. Mach. Vision Appl., 17:411–426, October 2006.

[WFEK09] W. Waizenegger, I. Feldmann, P. Eisert, and P. Kauff. Parallel high resolution real-time visual hull on gpu. In Image Processing (ICIP), 2009 16th IEEE International Conference on, pages 4301 –4304, 2009.

[WS95] Wasielewski and Strauss. Calibration of a multi-sensor system laser rangefinder/camera. In Proceedings of the Intelligent Vehicles '95. Symposium, IEEE, pages 472 – 477, 1995.

[WWTM00] Toshikazu Wada, Xiaojun Wu, Shogo Tokai, and Takashi Matsuyama. Homography based parallel volume intersection: Toward

real-time volume reconstruction using active cameras. In Computer Architectures for Machine Perception, 2000. Proceedings. Fifth IEEE International Workshop on 11-13 Sept. 2000, pages 331–339, 2000.

[xse] Xsens motion technologies. http://www.xsens.com.

[YAL06] Manuel Yguel, Olivier Aycard, and Christian Laugier. Efficient gpu-based construction of occupancy grids using several laser range-finders. Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, Oct. 2006.

[YLKC07] Sofiane Yous, Hamid Laga, Masatsugu Kidode, and Kunihiro Chihara. Gpu-based shape from silhouettes. In Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia, GRAPHITE '07, pages 71–77, New York, NY, USA, 2007. ACM.

[YMS04] Jana Kosecka Yi Ma, Stefano Soatta and S. Shankar Sastry. An invitation to 3D vision. Springer, 2004.

[YW07] Y. Yemez and C. J. Wetherilt. A volumetric fusion technique for surface reconstruction from silhouettes and range data. Comput. Vis. Image Underst., 105:30–41, January 2007.

[YZL+09] Tao Yang, Yanning Zhang, Meng Li, Dapei Shao, and Xingong Zhang. A multi-camera network system for markerless 3d human body voxel reconstruction. In Fifth International Conference on Image and Graphics, 2009. ICIG '09., 2009.

[ZADM10] I.M. Zendjebil, F. Ababsa, J. Didier, and M. Mallem. A gps-imu-camera modelization and calibration for 3d localization dedicated to outdoor mobile applications. In Control Automation and Systems (ICCAS), 2010 International Conference on, pages 1580–1585, oct. 2010.

[ZH96] Zhongfei Zhang and Allen R. Hanson. 3d reconstruction based on homography mapping. In In ARPA Image Understanding Workshop, 1996.

[Zie10] Gernot Ziegler. GPU Data Structures for Graphics and Vision. PhD thesis, Max-Planck-Institut fÃ¼r Informatik, 2010.

[ZL05] B. Zhang and Y.F. Li. An efficient method for dynamic calibration and 3d reconstruction using homographic transformation. Sensors and Actuators A: Physical, 119(2):349 – 357, 2005.

[ZWW03] Quan-Bing Zhang, Hai-Xian Wang, and Sui Wei. A new algorithm for 3d projective reconstruction based on infinite homography. In Machine Learning and Cybernetics, 2003 International Conference on, IEEE, 2003.