Exploiting Inertial Planes for Multi-sensor 3D Data Registration

PhD Thesis

Hadi Aliakbarpour

Coimbra, 2012
University of Coimbra
Faculty of Science and Technology
Department of Electrical and Computer Engineering

Exploiting Inertial Planes for Multi-sensor 3D Data Registration

Thesis submitted:

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 thesis 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

...
Abstract

This thesis 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. 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. 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 Aníbal 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, Christiana Tsiourti, ...

I would like to thank Luis Almeida, for his contribution in the real-time implementation of the system. 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-Fundaçao 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 has 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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resumo</td>
<td>i</td>
</tr>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgment</td>
<td>v</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Contributions</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Related works</td>
<td>3</td>
</tr>
<tr>
<td>1.3.1 Multi-view 3D reconstruction</td>
<td>3</td>
</tr>
<tr>
<td>1.3.2 Using IS to accompany vision</td>
<td>6</td>
</tr>
<tr>
<td>1.3.3 Real-time implementation using GP-GPU</td>
<td>7</td>
</tr>
<tr>
<td>1.3.4 Other relevant topics</td>
<td>8</td>
</tr>
<tr>
<td>1.4 Publications</td>
<td>13</td>
</tr>
<tr>
<td>1.4.1 Peer-reviewed journal articles</td>
<td>13</td>
</tr>
<tr>
<td>1.4.2 International awards</td>
<td>13</td>
</tr>
<tr>
<td>1.4.3 Peer-reviewed international conference papers</td>
<td>13</td>
</tr>
</tbody>
</table>
1.5 Dissertation outline ........................................... 16

2 3D Data registration using inertial planes ........................................ 19
  2.1 Introduction .................................................. 20
  2.2 Three dimensional data registration using inertial planes ............... 20
     2.2.1 Overall 3D reconstruction scheme .......................... 20
     2.2.2 Image plane of virtual camera ............................... 28
     2.2.3 Projection of 3D data onto a world inertial plane ............. 32
     2.2.4 Volumetric reconstruction ................................. 33
  2.3 Experiments .................................................. 41
  2.4 Conclusion .................................................. 45

3 Real-time implementation using GPU-CUDA .................................... 47
  3.1 Introduction .................................................. 48
  3.2 Parallel processing using GPU ................................... 49
  3.3 Experiments .................................................. 56
     3.3.1 Infrastructure .............................................. 56
     3.3.2 Reconstruction results .................................... 57
     3.3.3 Extension for mobile sensor ............................... 66
  3.4 Conclusion .................................................. 67

4 Parametric homography and translation estimation ............................ 69
  4.1 Introduction .................................................. 70
  4.2 Parametric homographies among different planes in the framework . .. 70
4.2.1 Volumetric reconstruction: a recursive form .................. 80
4.3 Translation estimation among two virtual cameras ............. 80
  4.3.1 Error analysis of the translation vector estimation ............. 84
4.4 Conclusion ................................................................. 89

5 Contribution on sensor configuration and tracking ................. 91
  5.1 Introduction .............................................................. 92
  5.2 Edge visibility criteria and camera configuration ................. 92
    5.2.1 Optimal Camera placement using Genetic Algorithm ........... 94
    5.2.2 Camera placement optimization using GA: simulation ........ 102
  5.3 Integration of mobile vision and laser sensor within a camera network - the estimation of extrinsic parameters .................. 102
  5.4 Low-level data filtering and tracking using Bayesian approach ... 115
    5.4.1 Concept of Bayesian filtering ................................. 115
    5.4.2 Applying Bayesian Occupancy Filtering ...................... 120
    5.4.3 Experiments on BOF and tracking .............................. 124
  5.5 Conclusion ................................................................. 124

6 Overall Conclusions and Future Work .................................. 125

A Extrinsic parameter estimation among a 2D-LRF and a mono-camera 129

Bibliography ................................................................. 136
# List of Figures

2.1 Overall scheme of the proposed 3D data registration .......................... 21
2.2 Pinhole camera model ....................................................................... 22
2.3 Pinhole camera model; transformation among the coordinate frames of the camera and world ................................................. 23
2.4 Homography among two image planes induced by a plane ................. 24
2.5 Distributed network of cameras and inertial sensors .......................... 26
2.6 Involved coordinate references in the framework ............................... 27
2.7 Graphical view of virtual camera definition ....................................... 28
2.8 Geometrical view of the virtual camera ............................................. 29
2.9 Data registration on an inertial Euclidean plane ............................... 31
2.10 Using a set of inertial planes for multi-layer 3D data registration ........ 34
2.11 Prospective Registration Plane ....................................................... 35
2.12 Geometric interpretation of the intersection among a person and an inertial plane ................................................................. 37
2.13 An example to demonstrate virtual images ....................................... 38
2.14 An exemplary IS-camera couple used in the experiments .................. 39
2.15 Setup for cat statue experiment ........................................ 39
2.16 Steps to obtain one 2D layer for 3D reconstruction .................. 40
2.17 Results of 3D Reconstruction of a cat statue .......................... 41
2.18 3D Reconstruction of a manikin; the process .......................... 43
2.19 3D Reconstruction of a manikin; results .............................. 44
3.1 Distributed network of cameras and inertial sensors .................. 48
3.2 Schematic of a virtual camera ........................................... 49
3.3 Parallelization architecture ............................................. 50
3.4 Architecture of CUDA .................................................. 51
3.5 Cell-wise intersection of the projections of the virtual images onto an inertial-plane .................................................. 54
3.6 Flowchart of GP-GPU(CUDA) implementation ......................... 55
3.7 Smart-room scene ....................................................... 56
3.8 The IS-camera couple used in the real-time 3D reconstruction experiment. .................................................. 58
3.9 Result for real-time 3D reconstruction (1) .............................. 59
3.10 Result for real-time 3D reconstruction (2) ............................. 60
3.11 Result for real-time 3D reconstruction (3) ............................. 61
3.12 Reconstruction result: objects in a scene ................................ 62
3.13 Reconstruction result: two persons .................................... 63
3.14 Average processing times for different size of inertial-planes ...... 64
3.15 Average processing times for different number of inertial-planes .. 65
3.16 3D Reconstruction result-mobile robot extension ....................... 65
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.17</td>
<td>3D Reconstruction result-mobile robot extension(2)</td>
<td>66</td>
</tr>
<tr>
<td>4.1</td>
<td>Extending homography for planes parallel to $\pi_{ref}$</td>
<td>71</td>
</tr>
<tr>
<td>4.2</td>
<td>Parametric homography among an inertial-plane $\pi'$ and the reference inertial-plane $\pi_{ref}$</td>
<td>72</td>
</tr>
<tr>
<td>4.3</td>
<td>Parametric homography among two consecutive inertial-planes, induced by a virtual image</td>
<td>73</td>
</tr>
<tr>
<td>4.4</td>
<td>Homography between image planes of two virtual cameras.</td>
<td>75</td>
</tr>
<tr>
<td>4.5</td>
<td>Homography between the image planes of two virtual cameras, induced by an inertial plane $\pi'$ parallel to the reference inertial-plane $\pi_{ref}$.</td>
<td>77</td>
</tr>
<tr>
<td>4.6</td>
<td>Translation between two virtual cameras.</td>
<td>81</td>
</tr>
<tr>
<td>4.7</td>
<td>Analysis of noise impact in IS orientation for estimating translation among virtual cameras</td>
<td>87</td>
</tr>
<tr>
<td>4.8</td>
<td>Analysis of noise impact in measurement of the heights of two 3D points for estimating translation among virtual cameras</td>
<td>88</td>
</tr>
<tr>
<td>4.9</td>
<td>Analysis of noise impact in extraction of the image coordinates for estimating translation among virtual cameras</td>
<td>89</td>
</tr>
<tr>
<td>4.10</td>
<td>Analysis of the relation between the distances of 3D points and the accuracy of the result in the proposed algorithm to estimate the translation among virtual cameras</td>
<td>90</td>
</tr>
<tr>
<td>5.1</td>
<td>Investigation of the criteria for visibility of a general convex polygon</td>
<td>93</td>
</tr>
<tr>
<td>5.2</td>
<td>Involved vectors in registration of plane corresponding to Fig. 5.1</td>
<td>93</td>
</tr>
<tr>
<td>5.3</td>
<td>Structure of a chromosome string.</td>
<td>96</td>
</tr>
<tr>
<td>5.4</td>
<td>Cost function between a camera and a polygon edge</td>
<td>97</td>
</tr>
<tr>
<td>Number</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.5</td>
<td>Local minima problem for a triangular polygon and three cameras</td>
<td>100</td>
</tr>
<tr>
<td>5.6</td>
<td>Results for Camera placement optimization using the proposed GA algorithm</td>
<td>103</td>
</tr>
<tr>
<td>5.7</td>
<td>Results for Camera placement optimization using the proposed GA algorithm</td>
<td>104</td>
</tr>
<tr>
<td>5.8</td>
<td>Results for Camera placement optimization using the proposed GA algorithm</td>
<td>105</td>
</tr>
<tr>
<td>5.9</td>
<td>Schematic of a smart-room including a mobile robot</td>
<td>107</td>
</tr>
<tr>
<td>5.10</td>
<td>Schematic of the problem of calibration among a LRF and a stereo camera</td>
<td>109</td>
</tr>
<tr>
<td>5.11</td>
<td>Geometric concepts in LRF and stereo camera calibration</td>
<td>111</td>
</tr>
<tr>
<td>5.12</td>
<td>Setup used in the experiments</td>
<td>114</td>
</tr>
<tr>
<td>5.13</td>
<td>Illustration of corresponding points collection between LRF and stereo camera</td>
<td>115</td>
</tr>
<tr>
<td>5.14</td>
<td>Evaluation of LRF-stereo camera calibration method</td>
<td>116</td>
</tr>
<tr>
<td>5.15</td>
<td>Data acquired by the laser range finder in three different planes</td>
<td>116</td>
</tr>
<tr>
<td>5.16</td>
<td>Reprojection of 3D range data on the image using the proposed calibration method between LRF and camera.</td>
<td>117</td>
</tr>
<tr>
<td>5.17</td>
<td>Two stages in BOF to estimate occupancy and velocity distribution</td>
<td>119</td>
</tr>
<tr>
<td>5.18</td>
<td>Applying Bayesian Occupancy Filtering to deal with the dynamic of scene in the proposed registration framework.</td>
<td>122</td>
</tr>
<tr>
<td>5.19</td>
<td>Bayesian Occupancy Filtering and Tracking</td>
<td>123</td>
</tr>
<tr>
<td>A.1</td>
<td>2D-LRF and a camera</td>
<td>131</td>
</tr>
<tr>
<td>A.2</td>
<td>2D-LRF and two cameras</td>
<td>132</td>
</tr>
</tbody>
</table>
A.3 Scheme of a camera network and a LRF equipped robot agent . . . . 133
List of Tables
# List of Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3D reconstruction algorithm using a set of inertial based horizontal planes.</td>
<td>36</td>
</tr>
<tr>
<td>2</td>
<td>Algorithm of real-time 3D reconstruction</td>
<td>53</td>
</tr>
<tr>
<td>3</td>
<td>3D data registration using inertial-planes in a recursive form</td>
<td>79</td>
</tr>
<tr>
<td>4</td>
<td>Simulation to evaluate affect of IS noise on translation estimation</td>
<td>85</td>
</tr>
<tr>
<td>5</td>
<td>Criteria to check the edges visibility for a given polygon</td>
<td>92</td>
</tr>
<tr>
<td>6</td>
<td>Algorithm to generate a gene</td>
<td>95</td>
</tr>
<tr>
<td>7</td>
<td>Algorithm to generate a chromosome</td>
<td>96</td>
</tr>
<tr>
<td>8</td>
<td>Algorithm to compute the cost for the genes and chromosome</td>
<td>99</td>
</tr>
<tr>
<td>9</td>
<td>Genetic algorithm to search for an optimal solution for camera placement</td>
<td>101</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction
1.1 Motivation

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 such that nowadays even many smart phones can be found equipped in both IS and camera sensors. 3D earth cardinal orientation is one of the outputs of an IS. We take the advantage of having a camera already coupled with an IS and investigate the problem of multi-sensor data registration. There is a variety of applications from different areas which can benefit from 3D reconstruction. These areas include surveillance, human motion and behaviour modelling, virtual-reality, smart-room, health-care, games, teleconferencing, human-robot interaction, medical industries and scene and object understanding.

1.2 Contributions

We explore the use of IS for 3D volumetric data registration by using a network of cameras and inertial sensors with no planar ground assumption. The synergy among the image and inertial data (3D orientation from IS) is explored for the purpose of multi-layer data registration. The axis of the work is multi-sensor multi-layer data registration nevertheless we take the advantage and have some related contributions. As an overview we have the following contributions in this thesis:

- Proposing a homographic framework for 3D data registration using a distributed network of cameras and inertial sensors. [AD12] [AD11c] [AD10b] [AD11b] [AD10a] [AFKD10] [AFQ+11].

- A real-time prototype to fully reconstruct human body (and objects as well) by using a parallel processing architecture on GP-GPU [AAMD11].
• Proposing a two-point-based method to estimate translations among virtual cameras in the framework [AD12] [AD11a] [AD10a] [AFQ+11].

• Geometric relations among different projective image planes and Euclidean inertial planes involved in the framework. Moreover proposing an algorithm to have an appropriate coverage of the camera network to a polygonal object or scene, by using genetic algorithm [AD11d].

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

1.3 Related works

Previous works related to the scope of this thesis fall into four 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, and eventually some other relevant issues related to the topic of dissertation such as data filtering and tracking, and extrinsic parameters estimation.

1.3.1 Multi-view 3D reconstruction

There have been many works in the area of 3D reconstruction. Khan in [KYS07] proposed a homographic framework for the fusion of multi-view silhouettes. A markerless 3D human motion capturing approach is introduced in [MGB07] using multiple views. Zhang in [ZWW03] introduced an algorithm for 3D projective reconstruction based on infinite homography. Homography-based mapping is used to implement a 3D reconstruction algorithm by Zhang and Hanson in [ZH96]. Wada et al. [WWTM00] studied a 3D reconstruction method using homography transformation. Sorman et al. in [SZB+07] presented a multi-view reconstruction method based on volumetric graph-cuts. Lai and Yilmaz in [LY08] used images from uncalibrated cameras for perform-

*SLaRF; available to download at http://paloma.isr.uc.pt/~hadi
ing projective reconstruction of buildings based on Shape From Silhouette (SFS) approach where buildings structure is used to compute vanishing points. Feldmann et al. [FMS+10] utilized the volumetric 3D reconstruction for the aim of online body motion tracking system. 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]. 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]. In [LW08] SFS is combined with stereo imaging for the sake of 3D reconstruction by Lin. Michoud in [MSEH08] proposed a method to eliminate appearing ghost object in SFS-based 3D reconstructions. 3D reconstruction of a dynamic scene is investigated in [CRM06] by Calibi. Franco in [FB05] used a Bayesian occupancy grid to represent the silhouette cues of objects. 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]. Khan in [Kha08] proposed some algorithms to track, reconstruct and object classification by using a homographic occupancy constrain. A method to extract silhouettes is proposed by Hwang et al. in [HKK05] which is based on a proposed change detection algorithm and graph cuts based optimization. A survey on motion-parallax-based 3-D reconstruction techniques is provided by Lu et al. in [LZWN04]. Uriol [Uri05] used a camera network to reconstruct human and synthesis an avatar. Brice et al. [BES05] investigated the use of multi-view geometry to human model and pose reconstruction. Jethwa in [Jet04] proposed a method to perform efficient voxel-based reconstruction of urban environments using a large set of images. Luo et al. [LBTVD10] introduced a method to estimate human pose for multiple persons based on volume reconstruction. A framework for 3D reconstruction is proposed by Bardsley [Bar08] in which multiple facets of 3D reconstruction has been analysed. 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. Inter-frame redundancy is exploited for real-time volumetric reconstruction of arbitrary shapes by Ruiz et al. [RM08] where a cluster of computers is employed. A distributed 3D reconstruction algorithm is proposed in [UHRM09] by Uson et al. to monitor a moving train coach by using a set of fish-eye cameras. 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. 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. Different multi-view stereo reconstruction algorithms are compared and evaluated on a common ground truth by Seitz et al. [SCD+06]. Lai and Yilmaz in [LY08] proposed a reconstruction method using slicing planes. Capturing of complex human movements from multiple views is studied in [Keh05] by Kehl. Liu and Cooper [LC10] developed a method called Ray Markov Random Fields for image-based 3D modeling. Ruwwe et al. in [RKR+08] proposed an approach for image registration based on reconstructed 3D octrees by voxel carving. A method for reconstruction of human head using multicamera stream is proposed in [KD09] by Kim and Dayhot. 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]. A system for real-time 3D human visual hull reconstruction and skeleton voxels extraction is proposed by Yang et al. [YZL+09]. 3D reconstruction of natural underwater scenes using an stereo-vision is studied by Brandou et al. [BAP+07]. Hu and Tan [HT06] investigated the problem of depth recovery and affine reconstruction under camera pure translation. A human body posture estimation method based on back projection of human silhouette images is proposed by Takahashi et al. [KT07]. Metric 3D reconstruction for large structures from uncalibrated images and using homography techniques is investigated by Tang et al. [TWH+07]. 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 [KTD09] 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 reconstruction 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.

1.3.2 Using IS to accompany vision

The use of inertial sensors to accompany computer 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 monocular 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.

*Micro-Electro-Mechanical-Systems
1.3. Related works

[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 accuracy of IS is improved less that 0.001°.

1.3.3 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 et al. in [FLD10] proposed a real-time implementation of Bayesian models for perception through multi-modal sensors by using CUDA. Almeida 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 resoulation and real-time reconstruction using visual hulls. A GPU-based shape from silhouette (SFS) algorithm is implemented in
Chapter 1. Introduction

[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 implementation 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.

1.3.4 Other relevant topics

Multi-view observation and data fusion for tracking

Currently, tracking, particularly applied for surveillance systems, is one of the most interesting research area, because of using more and more security in many public and private spaces. Nowadays, if you go in a bank, airport, etc., you can see some cameras around the places which can provide a network named Closed Circuit Television (CCTV). Ainsworth in [Ain04] discussed that CCTV is one of the easiest system which can use as a surveillance manager, and he believed that a CCTV needs a team operators which is trained and equipped for fast response. By growing the number of surveillance cameras, it feels that we have to use some real-time methods which can obtain human and other objects behaviour and activities by the cameras, because using human sources for that are going to be impossible and useless.

There are many techniques and methods which are for solving different problems which appear in different situations for a surveillance system. Stauffer and Grimson in [CS00, Sta99] have some discussions on real-time object tracking which is based on adaptive background mixture models. Haritaoglu et al. in [IH00] discussed real-time surveillance which comprised background modelling, people detection, multi-people
1.3. Related works

tracking, dynamic appearance model, detecting and tracking some main human body parts (head, hand, feet and torso), detecting and tracking an object which human is carrying and recognizing events which happen between human and objects. Cucchiara et al. in [RC01] presented a system which named Sakbot for detecting and tracking a moving object in a video surveillance system. The background of system update statistically and knowledge-based way and also for shadow problem they used the HSV color information. Buzan et al. in [DB04, BUZ03] showed a method for tracking of human motion (trajectory). They estimated object trajectories in 3D space by an Extended Kalman Filter (EKF) which was used to provide the 3D tracked motion to a constant. Hall et al. in [DNR+05] presented a comparison between different detection methods which are based on adaptive background. Boult et al. in [TEB01] discussed a special surveillance application concerning to non-cooperative and concealed objects in outdoor scenarios. They used omni-directional camera to obtain wide field of view from the scene and used LOTS approach to deal with the high sensitive applications. They believed that the LOTS which is based on multiple background models, is better to deal with complex background noises such as trees and grass motion which isn’t considered significant. Jorge et al. in [PMJ04] described a method for on-line object tracking including two parts; the first part was object trajectory detection which obtained by simple low level operations and the second one was trajectory labelling which were performed by an Bayesian network model. Abrantes et al. in [Abr02] presented a long-term multiple object tracking system by a Bayesian network Ayers and Shah in [DA01] discussed a system to recognize some human actions such as entering a room, using a PC, pickup a phone, opening a cabinet, etc. with the assumption that the scene’s structure is known. Dee and Hogg in [Dee04] presented an approach for detecting of abnormal or interesting events in movies which included some specific types of intentional behaviour. Eshel in [EM08] proposed a homography-based multiple camera people detection and tracking in a crowd dense and a similar topic is investigated also in [Tur08]. Aliakbarpour et al. in [AKQ+11] used a distributed heterogeneous sensor network to detect human abnormal behaviour in an outdoor scene using Hidden Markov Model (HMM). In [RAQ+11, KAQ+10], Roudposhti and Aliakbarpour et al. applied Laban Movement Analysis (LMA) descriptor in order to analyse human be-
haviour by using both frequency and spatial features. Roudposhti and Aliakbarpour et al. in [KAM+11] investigated the problem of human behaviour analysis using LMA and HMM. In [QRA+11], Quintas, Aliakbarpour et al. used concurrent HMM to analyse human behaviour model in a smart-room through a sensor network.

Fusion of the sensors outputs is one of the big challenges related to the earlier applications. A data fusion model, called Joint Directors of Laboratories (JDL), has been proposed by the US Joint Directors of Laboratories Data Fusion Sub-Group [SS06, SHB04, GFSP04]. Smith et al. in [SS06] provided a survey about existing methods for solving multi sensor data fusion problems by Smith et al. They categorized related papers by the JDL levels and discussed ability and inability of different methods in the different level of JDL with the context of different application such as autonomous robotics, military applications, and mobile systems. Finally they presented other papers related to multi sensor tracking problems which they divided in two parts which are out of sequence measurements and the effect of one sensor on another or data correlation. Armesto et al. in [AT07] presented an approach to model and fuse of non-linear data from multi-rate systems which have visual and inertial sensors. Their application is on egocentric systems such as robot navigation. Asoh et al. in [AAY+04] used particle filters to reduce computational cost of using only Bayesian inference for fusion of sound and vision data to recognizing multiple moving human sound sources. Hofmann and Aliakbarpour et al. in [HKAR11] performed multi-modal data fusion in a voxel occupancy grid for the sake of human tracking and behaviour analysis. Baltzakis et al. in [BAT03] presented a method to use 2D range data by Laser range finder (LRF) and vision data to give more accuracy to avoid collision in robot motion planing applications. Bellotto and Hu also presented an approach in [BH06] to fuse the LRF data and vision data by PTZ camera using Unscented Kalman Filter (UKF) for people tracking by egocentric way. In their method, they used LRF data for detecting legs and vision system for faces and then they presented a method to estimate human situations. Chakravarty and Jarvis in [CJ07] discussed a fusion method for LRF and panoramic vision data to track people from a stationary robot simultaneously, where particle filter was used for tracking and a mixture of Gaussian background subtraction algorithm was applied to modelling colour of each person.
Martin et al. in [CMG05] described a method to fuse three different sensors such as LRF, sonar and omnidirectional camera, to cover their weakness for people detection and tracking applications by a mobile robot. Jin and Mokhtarian in [JM05] presented a data fusion based on particle filter for head detection and tracking applications applicable for video surveillance systems. Zou and Bhanu in [ZB05] discussed a comparison between Time-Delay Neural Network (TDNN) and Bayesian Network (BN) approaches for human motion detection by fusion of audio and video data. Braillon et al. in [BUP06] described a method to detect obstacles by fusion of stereo camera data and optical flow field given by one of the stereo pair’s camera.

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. 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
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. 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.
1.4 Publications

1.4.1 Peer-reviewed journal articles


1.4.2 International awards

- Best 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 Vi-
- 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.


1.4. Publications

Khoshhal, K. and Dias, J., 14th International Conference on Advanced Robotics (ICAR 2009), 2009.

As a co-author


- Probabilistic LMA-based Human Motion Analysis by Conjugating Frequency and Spatial based Features, Kamrad Khoshhal Roudposhti, Hadi Aliakbarpour, Joao 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.


- Using Concurrent Hidden Markov Models to Analyse Human Behaviours in a Smart Home Environment, Joao Quintas, Kamrad Khoshhal Roudposhti, Hadi
16 Chapter 1. Introduction

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, JosÃ© 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 discusses about 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.

In chapter 4 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.

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 an appendix (Appen. A) 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 geometric method through 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

A framework to register the 3D data of a scene is proposed and explained in this section. In this thesis, we use the following convention for mathematical symbols: Vectors and matrices are all in bold, except for rotation matrices, camera calibration matrices and homography transforms which appear in normal capital. 3D points appear in capital bold and 2D points in small bold. Superscript in a variable indicates the reference frame in which the variable is expressed. For transformation matrices, subscript indicate the origin system and superscript means the destination system.

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
2.2. Three dimensional data registration using inertial planes

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.

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 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.
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 $\mathbf{X} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ in the scene and its corresponding projection $\mathbf{x} = [x \ y \ 1]^T$ on the image plane are related via a $3 \times 4$ matrix $A$, called camera projection matrix, through the following equations (assuming the camera’s coordinate frame as the world’s coordinate frame):

$$\mathbf{x} = A \mathbf{X}$$ \hspace{1cm} (2.1)

$A$ defined as

$$A = K \left[ \begin{array}{c} I_{3\times3} \ 0_{3\times1} \end{array} \right]$$ \hspace{1cm} (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:

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

(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 $f_x = f m_x$ and $f_y = f m_y$ 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:

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

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 $t$ among the two coordinate frames.
Figure 2.4: Homography among two image planes induced by a plane. π is a plane in the scene. x is a 3D point on π observed by two cameras C1 and C2. x1 and x2 are the images of 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 x1 to x2. 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.

frames (Fig. 2.3). Considering this, the general camera projection matrix A is expressed [HZ03] as

\[ A = K [R|t] \]  \hspace{1cm} (2.5)

**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 \( A' = K' [R|t] \) (concerning first camera center as world reference frame) where \( K \) and \( K' \) are the calibration matrices of the cameras. Also assume that \( x_1 \) and \( x_2 \) are the imaged points of a 3D point x lying on the plane π. Then \( x_1 \) and \( x_2 \) are called a pair of corresponding points and the
2.2. Three dimensional data registration using inertial planes

relation between them can be expressed as \( x_2 = H x_1 \) in which \( H \) is a \( 3 \times 3 \) transformation matrix called planar homography induced by the plane \( \pi \) [YMS04] and is equal to (up to scale)

\[
H = K' \left( R + \frac{1}{d} \mathbf{n}^T \right) K^{-1}
\]  
(2.6)

where \( R \) and \( t \) 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 point like \( x \) lying on the plane \( \pi \), its imaged point \( x_2 \) on the second camera is uniquely obtained via \( x_1 \) as \( x_2 = H x_1 \). However for a point like \( x' \) not lying on the plane \( \pi \) (or off-the-plane), then using the homography matrix \( H \) as \( x_2 = H x_1 \) yields to have the point \( x'_2 \) which is just a point lying on \( l_2 \) (\( l_2 \) being the epipolar line passing through the actual corresponding point \( x'_2 \) and the epipole \( e_2 \)). Whereas the actual imaged point corresponding to \( x' \) on the second camera is \( x'_2 \).

Network of cameras and inertial sensors

Fig. 2.5 shows a sensor network setup with a number of cameras. \( \pi_{ref} \) is an Euclidean inertial plane *, defined by the 3D orientation of IS, and is common for all cameras. Here \( \{W\} \) is the world reference frame (a detailed specification of this reference frame shall be introduced in Sec. 2.2.2). 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 \( \pi_{ref} \) as \( \pi \times \) (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 sensor, \( f \) 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

*It might appear just as \( \pi \) in the equations
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. \( \pi_{ref} \) is a virtual reference plane (Euclidean) which is defined by using 3D orientation of IS and is common for all virtual cameras.

The matrix among these two reference frames is

\[
^vT_c = \begin{bmatrix}
^vR_c & ^v_t_c \\
0_{3 \times 3} & 1
\end{bmatrix}
\]

where \(^vR_c\) is the rotation matrix and \(^v_t_c\) is equal to \(0_{3 \times 1}\).

In order to register a 3D point \(X\) onto the \(\pi_{ref}\) as \(\pi\), three steps can be taken:

- First, the 3D point \(X\) is projected on the camera image plane by \(\pi = AX\) (\(A\) is the projection matrix of the camera \(C\)).

- Second, \(\pi\) (the imaged point on the camera image plane) is projected to its corresponding point on the virtual camera’s image plane as \(\pi\). Indeed this operation is
2.2. Three dimensional data registration using inertial planes

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.

A homography transformation among two projective planes and can be expressed as \( y = H_x x \), \( H_x \) being a \( 3 \times 3 \) homography matrix [HZ03].

- Third, the projected point on the virtual image plane, \( y \), is reprojected to the world virtual plane, \( \pi_{\text{ref}} \). This operation is among a projective plane and an Euclidean plane and again can be expressed as \( \pi_x = H_v y \), where \( H_v \) is a \( 3 \times 3 \) homography matrix [HZ03].

The first step is done by the camera based on the pinhole camera model (previously introduced). The second and third steps are described in the following two sub-sections. Assuming to already have \( H_c \) and \( H_v \), the final equation for registering a 3D point \( X \) onto the reference plane \( \pi_{\text{ref}} \) will be (see Fig. 2.9):

\[
\pi_{\text{ref}} x = \pi H_v y H_c A X
\] 

(2.8)
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 \( ^VR_C \) and a translation equal to 0\( \times \)1.

The way of obtaining \( ^vH_C \) (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 \( \pi_{ref} \)) is discussed in the next sub-sections by starting to describe the conventional coordinate systems.

### 2.2.2 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. 2.7, there are five coordinate systems involved in this approach to be explained here:
2.2. Three dimensional data registration using inertial planes

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, $\pi_{ref}$ is defined as a virtual world plane which is horizontal and parallel to the image plane of virtual camera.

- **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 $\{E\}$.

- **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 $\pi_{\text{ref}}$ defined in $\{W\}$ (the world reference frame of this approach). The reference 3D plane $\pi_{\text{ref}}$ 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. 2.7). In this proposed method the idea is to not using any real 3D plane inside the scene for 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 $\pi_{\text{ref}}$ 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 $\pi_{\text{ref}}$. With no loss of generality we place $O_W$, the center of $\{W\}$, in the 3D space such a way that $O_W$ 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

$$
\begin{bmatrix}
W_{R_V} & t \\
0_{1 \times 3} & 1
\end{bmatrix}
$$

(2.9)

where $W_{R_V}$ is a rotation matrix defined as (see Fig. 2.7):

$$
W_{R_V} = \begin{bmatrix}
\hat{i} & -\hat{j} & -\hat{k}
\end{bmatrix}
$$

(2.10)

$\hat{i}$, $\hat{j}$ and $\hat{k}$ being the unit vectors of the $X$, $Y$ and $Z$ axes, respectively. Also $t$ 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 $t = [0 \ 0 \ d]^T$.

We continue the discussion to obtain a $3 \times 3$ homography matrix $^vH_c$ 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 $^v\mathbf{x} = ^vH_c^c\mathbf{x}$ (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 $^vH_c$ is called infinite
2.2. Three dimensional data registration using inertial planes

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

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]):

$$ VH_C = \lim_{d \to \infty} K (V R_C + \frac{1}{d} Tn^T) K^{-1} = K V R_C K^{-1} $$  \hspace{1cm} (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. 2.7) as following:

$$ V R_C = V R_E E R_{IS} f S R_C $$  \hspace{1cm} (2.12)
Chapter 2. 3D Data registration using inertial planes

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\}:

\[ ISR_C \] can be obtained through a 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:

\[ VR_E = \begin{bmatrix} \hat{i} & -\hat{j} & -\hat{k} \end{bmatrix} \quad (2.13) \]

### 2.2.3 Projection of 3D data onto a world inertial plane

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

\[ x = K \begin{bmatrix} r1 & r2 & r3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K \begin{bmatrix} r1 & r2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2.14) \]

where \( r1, r2 \) and \( r3 \) are the columns of the \( 3 \times 3 \) rotation matrix, \( K \) is the camera calibration matrix (defined in Eq. 2.4) and \( t \) is the translation vector between \( \pi_{ref} \) and camera center [HZ03]. As can be seen Eq. (2.14) indicates a plane to plane projective
2.2. Three dimensional data registration using inertial planes

transformation and therefore can be expressed like a planar homography:

\[ x = V_H \pi x \] (2.15)

where

\[ V_H \pi = K [ r_1 \ r_2 \ t ] \] (2.16)

\( \pi^H \) denoting a 3 × 3 homography matrix and \( \pi x = [ X \ Y \ 1]^T \). 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. \( W_{RV} \) or the rotation matrix between a virtual camera and \( \{ W \} \) was described through Eq. (2.10). Then considering \( W_{RV} \) from Eq. (2.10), \( \pi_{ref} \) as the interesting world plane and \( t = [ t_1 \ t_2 \ t_3 ]^T \) 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:

\[ \pi_H^{-1} = K [ \hat{i} \ -\hat{j} \ t ] = \begin{bmatrix} f_x & 0 & f_x t_1 + u_0 t_3 \\ 0 & -f_y & f_y t_2 + v_0 t_3 \\ 0 & 0 & t_3 \end{bmatrix} \] (2.17)

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

2.2.4 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 ho-
Chapter 2. 3D Data registration using inertial planes

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.

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.

There is a geometrical explanation to support this interpretation. Fig. 2.12 demonstrates a person being observed by two virtual cameras $V_1$ and $V_2$. $\pi'$ 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 $\pi'$ and $\mathbf{Y}$ is off the plane. The 3D points $\mathbf{X}$ and $\mathbf{Y}$ are imaged as $x_1, y_1, x_2$ and $y_2$ on the image planes of $V_1$ and $V_2$, respectively (using the proposed homograpy transformation).
2.2. Three dimensional data registration using inertial planes

Figure 2.11: Illustration of the registration using homography concept. (a): A scene including a human is depicted. $\pi_k$ is an inertial-based virtual world plane. Three cameras are observing the scene. (b): The registration layer (top view of the plane $\pi_k$ 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 $\hat{I}^c$ from $c$ to $v$
    obtain $t$ for each $v$ // translation vector
end

for $h = h_{\text{min}}$ to $h_{\text{max}}$ step $\Delta h$ begin
    for each $v$ involved in \{virtual camera\} begin
        obtain projection $\hat{I}^v$ from $v$ to $\pi^h$
    end
    for each $i \in \{1..\text{height}(\hat{I}^v)\}$ begin
        for each $j \in \{1..\text{width}(\hat{I}^v)\}$ begin
            $n_c = \text{card}($\{virtual camera\}) //cardinality
            $R(h, i, j) = \prod_{v=1}^{n_c} \hat{I}^v(i, j)$
        end
    end
end

return $R$ // as volumetric 3D reconstruction of the object

phy methods). Suppose $\pi' x_1$, $\pi' y_1$, $\pi' x_2$ and $\pi' y_2$ are respectively the projections of the imaged points $x_1$, $y_1$, $x_2$ and $y_2$ onto $\pi'$. As seen in Fig. 2.12, for an on the plane point such as $X$, all three points $X$, $\pi' x_1$, $\pi' x_2$ are coincident and meet on $\pi'$. In contrary, for the point $Y$ which is off the plane, the three points $Y$, $\pi' y_1$, $\pi' y_2$ are distinct. $y'_2$ denotes the image of $\pi' y_1$ on the image plane of $V_2$. The vector between $y'_2$ and $y_2$ is called parallax. Indeed the line through $y'_2$ and $y_2$ is the image of the ray passing through the center of $V_1$ and $Y$ (which is also an epipolar line). For all points off the inertial plane $\pi'$, the norm of their parallax is bigger than zero and for those points which are on $\pi'$, 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
Figure 2.12: Geometric interpretation of the intersection among a person and an inertial plane $\pi'$: $X$ and $Y$ are two exemplary 3D points belonging to the person’s body, being observed by two virtual cameras $V_1$ and $V_2$. $X$ lies on $\pi'$ and $Y$ is off the plane $\pi'$. The 3D points get projected on $\pi'$ using the proposed homographic method. The homographic projections of the point which are on $\pi'$ such as $X$ are coincident ($\pi'_x$, $\pi'_y$) whereas for 3D points off the plane (such as $Y$) their projections on $\pi'$ are distinct ($\pi'_y$ and $\pi'_y'$). In other words, for the points off $\pi'$ there is a parallax (like the vector through $y_2$ and $y'_2$).

can be used to perform volumetric 3D scene reconstruction. Such a reconstruction approach is encapsulated and described as an algorithm in Alg. 1. Here $\{\text{camera}\}$ and $\{\text{virtual camera}\}$ 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 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. $\Delta h$ can be interpreted as the horizontal resolution for the algorithm.
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. 2.2.2, 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.
2.2. Three dimensional data registration using inertial planes

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.
2.3. Experiments

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.

2.3 Experiments

A set of experiments have 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 experiments are used for 3D volumetric reconstruction of a cat statue and a manikin. 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 Cali-

*http://www.unibrain.com
Chapter 2. 3D Data registration using inertial planes

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

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 by applying an appropriate homography transformation, described in Sec. 2.2.2, 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 ?? (Sec. 4.3). 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,
2.3. Experiments

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

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 2.2.2. Fig. 2.16-c. shows the mentioned virtual image planes. Using the proposed 2-point-height method (shall be introduced in chapter ??) 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
250\text{mm}. The interval distance between the inertial-based virtual planes is considered as \(5\text{mm}\). As an example, Fig. 2.16-d indicates the reprojection of the three virtual camera images onto a virtual world plane at height=\(380\text{mm}\). 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
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 have been carried out for the proposed volumetric reconstruction algorithm where 3D reconstructions for a cat statue and a manikin 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

Real-time implementation using GPU-CUDA
Chapter 3. Real-time implementation using GPU-CUDA

3.1 Introduction

This chapter presents a full-body volumetric reconstruction of a person in a scene using a sensor network, where some of them can be mobile. The concept used in chapter is based on the previous chapter, however a brief summarization is presented here. The sensor network is comprised of couples of camera and inertial sensor (IS), as seen in Fig. 3.1. 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. 3.2). 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 gestures and some objects in the scene are reconstructed and demonstrated.
3.2 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. 3.4). A CUDA device consists of a set of multi-core processors. Each multicore processor is simply referred to as a multiprocessor. Cores of a multiprocessor work in
Figure 3.3: 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.
3.2. Parallel processing using GPU

Figure 3.4: CUDA architecture [AMD11].

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. 2. 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 $N_\pi$ 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 $V_i$. $\Delta h$ 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 fellow-chart of Fig. 3.6.
Algorithm 2: 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 \( N_{\pi} \) indicate the number of cameras and number of inertial-planes, respectively. \( I_{ci} \) and \( I_{vi} \) respectively are the image planes of camera \( C_i \) and its corresponding virtual camera \( V_i \). \( \Delta h \) 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. 3.6).

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
  \( v_i H_{ci} \leftarrow K_i v_i R_{ci} K_i^{-1} \)
  \( I_{vi} \leftarrow v_i H_{ci} I_{ci} \)

/* projecting virtual images onto inertial-planes */
for \( j \leftarrow 0 \) to \( N_{\pi} - 1 \) do
  for \( i \leftarrow 1 \) to \( N_c \) do
    \( \pi_{h} H_{vi} \leftarrow inv(\pi_{h} H_{\pi ref} + h P_i k^T) \) /* using Eq. (4.1) */
    \( \pi_{h} (v_i) \leftarrow \pi_{h} H_{vi} I_{vi} \)
    \( h \leftarrow h + \Delta h \)

/* obtaining intersection of the projected virtual images for each inertial plane */
for \( j \leftarrow 0 \) to \( N_{\pi} - 1 \) do
  /* cell-wise binary AND. Please see Fig. 3.5 */
  \( \pi_j \leftarrow \prod_{i=1}^{N_c} \pi_{h} (v_i) \)

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 \( \pi_{h} \) (the inertial plane at the height \( h \)) a set of temporary planes \( \{ \pi_{h} (v_1), \pi_{h} (v_2), \ldots, \pi_{h} (v_{N_c}) \} \) is defined. \( \pi_{h} (v_i) \) indicates the temporary inertial-plane corresponding to the virtual camera \( V_i \). The images planes of each vir-
Figure 3.5: Cell-wise intersection of the projections of the virtual images onto an exemplary inertial-plane $\pi_h$: Firstly the images of all virtual cameras get projected onto a temporary inertial plane. $\pi_h^{(vi)}$ 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 $\pi_h$. ($m$ and $n$ indicate the indices of a cell). Note that the silhouettes are considered as binary.

Fig. 3.3 depicts an architectural view corresponding to the algorithm. The parts colored in yellow are implemented on CUDA. Fig. 3.6 demonstrates the 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
3.2. Parallel processing using GPU

Figure 3.6: 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. 2.
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. The processes labelled by VirImgGen, GPU_Project2VirtualPlane and Gpu_Plane_Intersection are the part which are done on CUDA using a parallel implementation.

3.3 Experiments

3.3.1 Infrastructure

Fig. 3.7 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
3.3. Experiments

Each camera is rigidly coupled with an IS (we used Xsens MTx [xse]). Fig. 3.8 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+).

3.3.2 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. 3.10 and 3.11 where an acting person is fully reconstructed in 3D. One of the samples is separately shown in Fig. 3.9 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 \(384 \times 384 \text{ cm}^2\) which is relatively large. Using a parallel implementation of the algorithm (using GPU), we managed to have a frequency close to \(2.5 \text{ Hz}\) for reconstruction of the mentioned area (using the hardware stated in sub-section 3.3.1). The number of layers and their intervals can be adjusted depending to the need of an application and available hardware.

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. 3.12). In Fig. 3.12-(a) a semi rectangular blue box object is reconstructed. Fig. 3.12-(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. 3.12-(c). Fig. 3.13-(a) and Fig. 3.13-(b) show the result for a scene including a person and a manikin. The person seated on a chair is reconstructed and shown in Fig. 3.13-(c).
3.3. Experiments

Figure 3.9: 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.

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. 3.6, 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 $384 \times 384 \text{cm}^2$, have been used. The infrastructure and hardware are as stated in sub-section 3.3.1. The total processing time for a full 3D reconstruction is 405 ms which leads to have a frequency close to 2.5 Hz.

Fig. 3.14 depicts the average processing time in ms for different size of inertial-planes (the scale is $10^4 \text{cm}^2$). The number of inertial-planes in this experiments is a
Figure 3.10: 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 $50\, mm$. 
3.3. Experiments

Figure 3.11: 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 50 millimeters.
Figure 3.12: 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.
3.3. Experiments

Figure 3.13: 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 manikin. In the second one, (b), the person is seated in front of the manikin. (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.
Figure 3.14: Average processing times in ms for different size of inertial-planes. The notations are related to the flowchart shown in Fig. 3.6. Number of 2D inertial-planes used in this statistic is 48.

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. 3.15. The size of inertial-planes (they are equal in the sizes) is considered as a constant equal to $384 \times 384 \text{ cm}^2$. Similar to Fig. 3.14, the colors 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.
3.3. Experiments

Figure 3.15: Average processing times in ms for different number of inertial-planes. The notations are related to the flowchart shown in Fig. 3.6. The size of each 2D inertial-planes used in this statistic is $384 \times 384 \text{cm}^2$.

Figure 3.16: 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.
3.3.3 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.
3.4 Conclusion

In order to localize the mobile sensor, the method proposed in [AD11a] is used. Fig. 3.16 shows a case where just two cameras from the infrastructure is used for the 3D reconstruction of a manikin (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 manikin. 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. 3.17. This figure shows the advantage of having a mobile sensor which could cooperate with the infrastructure.

3.4 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 manikin and some objects. The presented results are quite promising.
Chapter 4

Parametric homography and translation estimation
4.1 Introduction

In this chapter we discuss about two topics, *parametric homography among different virtual planes* and *estimation of translation vectors among cameras*. 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 sensor 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.

4.2 Parametric homographies among different planes in the framework

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 $\pi_{\text{ref}}$ was obtained as $\pi H_V$ (see Eq. (2.17)). It is also desired to obtain the homography matrix from a virtual image to another world 3D plane parallel to $\pi_{\text{ref}}$ once we already have $\pi H_V$. Lets consider $\pi'$ as a 3D plane which is parallel to $\pi_{\text{ref}}$ and has a height $\Delta h$ w.r.t it (see Fig. 4.1). $\pi' H_V$ denotes the homography transformation which maps points of the image plane of $V$ onto $\pi'$. 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 $\Delta h$ as follows:

$$\pi' H_V^{-1}(\Delta h) = \pi H_V^{-1} + \Delta h P \hat{k}^T$$ (4.1)
4.2. Parametric homographies among different planes in the framework

Figure 4.1: Extending homography for planes parallel to $\pi_{\text{ref}}$. $\pi' \mathbf{H}_V$ is the already available homography matrix among virtual image plane $I'$ and the reference plane $\pi_{\text{ref}}$. $\pi'$ is another Euclidean virtual plane, parallel to $\pi_{\text{ref}}$. $\Delta h$ is the distance among $\pi$ and $\pi'$. The idea is to obtain $\pi' \mathbf{H}_V$, the homography between the image plane and $\pi'$ as a function of $\pi \mathbf{H}_V$ and $\Delta h$ (see Eq. (4.1)).

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

**Parametric homography relation among Euclidean inertial planes**

Suppose $\pi'$ is an inertial-plane with an Euclidean distance $\Delta h$ to the reference inertial plane $\pi_{\text{ref}}$. $\pi' \mathbf{H}_\pi$ 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. 4.2). Such a homography transformation can be expressed by the following equation:

$$\pi' \mathbf{H}_\pi = \pi' \mathbf{H}_V \pi \mathbf{H}_V^{-1}$$

(4.2)

where $\pi' \mathbf{H}_V$ is the homography transformation among the image plane of a virtual camera $V$ and the inertial-plane $\pi'$, and $\pi \mathbf{H}_V$ is the homography transformation between the
Figure 4.2: Parametric homography among an inertial-plane $\pi'$ and the reference inertial-plane $\pi_{ref}$. The homography transformation $\pi' H_{\pi_{ref}}$, induced by image plane of virtual camera $V$, which maps points from $\pi_{ref}$ onto $\pi'$ can be expressed as a function of $\Delta h$, $\Delta h$ being the Euclidean distance among two inertial-planes.

By substituting $\pi' H_{\pi}$ with Eq. (4.1), Eq. (4.2) becomes:

$$\pi' H_{\pi} = \left(\pi H_V^{-1} + \Delta h P \hat{k}^T\right)^{-1}\pi H_V^{-1}$$  \hspace{1cm} (4.3)

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

---

*Considering $A$ as a square matrix and $U$ and $V$ as two column vectors, the Sherman-Morrison-Woodbury formula gives $(A + U V^T)^{-1} = A^{-1} - A^{-1} U (I + V^T A^{-1} U)^{-1} V A^{-1}$
4.2. Parametric homographies among different planes in the framework

Figure 4.3: 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 $\Delta h_0$. In this case, the homography transformation among any two consecutive inertial-planes can be expressed as a function of $\Delta h_0$ and the index of the plane (see Eq. (4.12)).

\[
(\pi H^{-1} + \Delta h P \hat{k}^T)^{-1} \equiv \pi H - \frac{\pi H V P \hat{k}^T \pi H V}{\alpha + \hat{k}^T \pi H V P}
\]  

(4.4)

where $\alpha = \frac{1}{\Delta h}$. Eventually, Eq. (4.3) after simplifications can be expressed as a function of the distance between two inertial-planes:

\[
\pi' H_{\pi}(\alpha) = I_{3 \times 3} - \frac{1}{\alpha + \beta} \Gamma
\]

(4.5)

where $\beta$ is an scalar equal to
\[ \beta = \hat{k}^T \pi H_V P \]  \hspace{1cm} (4.6)

and \( \Gamma \) is a 3 \times 3 matrix equal to

\[ \Gamma = \pi H_V P \hat{k}^T \]  \hspace{1cm} (4.7)

Note that both \( \beta \) and \( \Gamma \) are constant for all inertial planes induced by the camera \( V \) (assuming that the calibration parameters will not change). Eq. (4.5) is interesting in the sense that once a basic homography \( \pi H_V \) to project an image to the reference inertial plane \( \pi_{ref} \) is obtained, a direct projection can be performed from \( \pi_{ref} \) to any arbitrary inertial plane namely \( \pi' \) with just knowing the Euclidean distance (\( \Delta h \)) among them for the purpose of 3D data registration.

While Eq. (4.5) expresses the projective relation among the reference plane \( \pi_{ref} \) 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. (4.3) depicts a set of inertial planes where the Euclidean distance among any two consecutive planes is equal to \( \Delta h_0 \). Suppose \( \pi_k H_\pi \) expresses the homography projection, induced by a virtual image, from the reference plane \( \pi_{ref} \) to \( k \)th inertial plane. Such a transformation can be written as:

\[ \pi_k H_\pi = \pi_k H_{\pi_{k-1}} \pi_{k-1} H_\pi \]  \hspace{1cm} (4.8)

and then

\[ \pi_k H_{\pi_{k-1}} = \pi_k H_\pi \pi_{k-1} H_\pi^{-1} \]  \hspace{1cm} (4.9)

In Eq. (4.9), by substituting the terms \( \pi_k H_\pi \) with its equivalence from Eq. (4.5) and \( \pi_{k-1} H_\pi \) with Eq.(4.3) we have (considering \( \alpha_0 = 1/\Delta h_0 \) :
4.2. Parametric homographies among different planes in the framework

Figure 4.4: Homography between image planes of two virtual cameras.

\[
\pi_k H_{\pi_{k-1}} = (I_{3\times3} - \frac{k}{\alpha_0 + k\beta} \Gamma) \left[ (\pi_{H_{\pi}}^{-1} + \frac{k-1}{\alpha_0} P\hat{k}^T)^{-1} \pi_{H_{\pi}}^{-1} \right]^{-1}
\]  (4.10)

after simplification:

\[
\pi_k H_{\pi_{k-1}} = (I_{3\times3} - \frac{k}{\alpha_0 + k\beta} \Gamma)(I_{3\times3} + \frac{k-1}{\alpha_0} \Gamma)
\]  (4.11)

The above equation can be written in a quadratic form as following:

\[
\pi_k H_{\pi_{k-1}} (k, \alpha_0) = I_{3\times3} + f(k, \alpha_0) \Gamma + g(k, \alpha_0) \Gamma^2
\]  (4.12)

where \( f : \mathbb{R}^2 \rightarrow \mathbb{R} \) and \( g : \mathbb{R}^2 \rightarrow \mathbb{R} \) (two coefficients of the quadratic function) are two scalar functions whose inputs parameters are the index of inertial plane \( \pi_k \) and the vertical resolution factor \( \alpha_0 \) as following:
Chapter 4. Parametric homography and translation estimation

\[ f(k, \alpha_0) = \frac{k\beta(k-1) - \alpha_0}{\alpha_0(k\beta + \alpha_0)} \]  \hspace{1cm} (4.13)

and

\[ g(k, \alpha_0) = \frac{k(1-k)}{\alpha_0(\alpha_0 + k\beta)} \]  \hspace{1cm} (4.14)

Note that \( \Gamma \) is a \( 3 \times 3 \) matrix which is constant provided that the calibration parameters do not change. Therefore \( \pi_k H_{\pi_{k-1}} \), the homography matrix (induced by a virtual camera’s image plane) which transforms the 2D points from inertial plane \( \pi_{k-1} \) to its consecutive inertial plane \( \pi_k \) is obtained as a function of \( k \) and \( \alpha_0 \) which is presented by Eq. (4.12).

**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 (\( \pi \)) 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. 4.4 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 \( j H^\pi_i \), the homography matrix among \( V_i \) and \( V_j \), induced by an inertial plane such as \( \pi \). Based on equation (2.6), \( j H^\pi_i \) can be expressed as:

\[ j H^\pi_i = K_j (R + \frac{1}{d} \Delta t n^T) K_i^{-1} \]  \hspace{1cm} (4.15)

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 \( (I_{3 \times 3}) \). \( \Delta t \) is a 3-elements vector describing the translation from \( V_i \) to \( V_j \). \( n = [ 0 \ 0 \ -1 ]^T \) is the normal of plane \( \pi \) and \( d \) is the distance between \( \pi \) and \{\( V_i \)\} along the Z axis of \{\( V_i \)\}. Therefore, after substitutions and simplifications, Eq. (4.15) can be
4.2. Parametric homographies among different planes in the framework

Figure 4.5: Homography between the image planes of two virtual cameras, induced by an inertial plane $\pi'$ parallel to the reference inertial-plane $\pi_{ref}$.

expressed as:

$$jH^\pi_i = K_j\left[ \hat{i} \quad \hat{j} \quad (\hat{k} - \frac{\Delta h}{d}) \right] K_i^{-1}$$  \hspace{1cm} (4.16)

where $\hat{i}$, $\hat{j}$ and $\hat{k}$ are the unit vectors for X, Y and Z axes, respectively. Assuming no changes in camera parameters, Eq. (4.16) for any inertial-plane just depends to $d$, the Euclidean distance between the inertial-plane and $\{V_i\}$.

Eq. (4.16) expresses the homography relation among the image planes of two cameras, induced by the reference inertial plane $\pi_{ref}$. It is interesting to obtain the homography among two image planes induced by another inertial plane (\(\pi'\)) using the basic relation from Eq. (4.16). Such a homography matrix can be notated as $v_2H^\pi_{v_1}$ and is depicted in Fig. (4.5). One can write this homography as:
The terms $\pi'H_{v_2}^{-1}$ and $\pi'H_{v_1}$ can be replaced by their equivalences using from Eq. (4.1) and (4.4), respectively:

$$v_2 H_{v_1}^{\pi'} = (v_2 H_{\pi'}) (\pi'H_{v_1}) = (\pi'H_{v_2}^{-1}) (\pi'H_{v_1})$$  \hspace{1cm} (4.17)

$$v_2 H_{v_1}^{\pi'} = (\pi'H_{v_2}^{-1} + \frac{1}{\alpha} P_2 \hat{k}^T) (\pi'H_{v_1} - \frac{\pi'H_{v_1} P_1 \hat{k}^T \pi'H_{v_1}}{\alpha + \beta_1})$$  \hspace{1cm} (4.18)

where $P_1$ and $P_2$ are respectively the principal vectors of the virtual cameras $V_1$ and $V_2$. $\alpha$ is equal to the reverse of $\Delta h$ (the distance among the two inertial planes $\pi_{ref}$ and $\pi'$) and moreover, $\beta_1$ is an scalar value related to $v_1$ defined by Eq. (4.6). After simplification and replacing $\pi'H_{v_2}^{-1} \pi'H_{v_1}$ with $v_2 H_{v_1}^{\pi'}$ we will have:

$$v_2 H_{v_1}^{\pi'} = v_2 H_{v_1}^{\pi} - f(\alpha) v_2 H_{v_1}^{\pi} \Phi_1 + \frac{1}{\alpha} \Phi_2 - \frac{f(\alpha)}{\alpha} \Phi_1 \Phi_2$$  \hspace{1cm} (4.19)

$\Phi_1$ and $\Phi_2$ are two $3 \times 3$ matrices which are constant for each virtual camera and defined as following:

$$\Phi_1 = P_1 \hat{k}^T \pi'H_{v_1}$$  \hspace{1cm} (4.20)

$$\Phi_2 = P_2 \hat{k}^T \pi'H_{v_1}$$  \hspace{1cm} (4.21)

and $f(\alpha)$ is a scalar function of the distance among two virtual planes:

$$f(\alpha) = \frac{1}{\alpha + \beta_1}$$  \hspace{1cm} (4.22)

Eq. (4.19) can be even more simplified and reformulated as following:
Algorithm 3: 3D data registration using inertial-planes in a recursive form

Function ThreeDimRegistration()
begin
/* Initialization */
for \( i \leftarrow 1 \) to \( N \) do
\[ v_i H_c \leftarrow K_i \, v_i R_i K_i^{-1} \]
\[ I_c \leftarrow v_i H_c \, I_c \]
\[ \pi_{ref} H_v \leftarrow inv(K_c \begin{bmatrix} \mathbf{i} & -\mathbf{j} & \mathbf{t}_c \end{bmatrix}) \quad \text{Eq. (2.17)} \]
\[ \pi_{ref}^v \leftarrow \pi_{ref} H_v I_v \]
\[ \Gamma_v \leftarrow \pi_{ref} H_v P_v \, \mathbf{k}^T \quad \text{Eq. (4.7)} \]
\[ \beta_v \leftarrow \mathbf{k}^T \pi_{ref} H_v P_v \quad \text{Eq. (4.6)} \]
/* Performing recursive registration for each camera */
for \( i \leftarrow 1 \) to \( N \) do
\[ \pi_{\{v\}} \leftarrow \text{RegisterRecursive} \left( \pi_{N-1}, \pi_{\{v\}} \right) \]
/* Performing cell-wise intersection, see Fig. 3.5 */
for \( i \leftarrow 0 \) to \( N-1 \) do
\[ \pi_i \leftarrow \prod_{j=1}^{N} \pi_{j|} \]
end

Function RegisterRecursive \( (k, \pi_{\{v\}}) \)
begin
if \( k = 0 \) then
return \( \pi_{\{v\}} \)
else
\[ \pi_{\{v\}} \leftarrow \text{GenerateNextHomography} \left( k-1 \right) \]
\[ \pi_k \leftarrow \text{Warp} \left( H, \text{RegisterRecursive} \left( k-1, \pi_{\{v\}} \right) \right) \]
return \( \pi_k \)
end

Function GenerateNextHomography \( (k) \)
begin
\[ H = I_{3 \times 3} + f(k, \alpha_0) \Gamma + g(k, \alpha_0) \Gamma^2 \quad \text{Eq. (4.12)} \]
return \( H \)
end

inline Function \( f(k, \alpha_0) \)
return \( \frac{k \beta(k-1) - \alpha_0}{\alpha_0 (k \beta + \alpha_0)} \quad \text{Eq. (4.13)} \)

inline Function \( g(k, \alpha_0) \)
return \( \frac{k(1-k)}{\alpha_0 (k \beta + \alpha_0)} \quad \text{Eq. (4.14)} \)
\[ v_2 H'_{v_1} = v_2 H_{v_1} \Psi(\alpha) + \frac{1}{\alpha} \Psi(\alpha) \Phi_2 \]  

(4.23)

where \( \Psi(\alpha) \) is a \( 3 \times 3 \) matrix being a function of \( \alpha \):

\[ \Psi(\alpha) = I - f(\alpha)\Phi_1 \]  

(4.24)

As one can see, the Eq. 4.19 (or Eq. 4.23) expresses the homography among two virtual cameras induced by an inertial plane \( \pi' \) parallel to \( \pi_{ref} \), by using a linear equation of the homography among the same virtual cameras but induced through the reference inertial plane \( \pi_{ref} \).

4.2.1 Volumetric reconstruction: a recursive form

In previous chapters some algorithms to perform 3D data registration of object or human were 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. 3 which is capable of performing the 3D reconstruction task in a recursive manner. In this algorithm, \( \text{ThreeDimRegistration()} \) is the main function in which firstly the variables are initialized. After initialization, for each camera, \( \text{RegisterRecursive()} \) function is called. This function recursively projects and registers the image data onto the consecutive inertial planes in the scene. In this function, \( \text{Warp()} \) is a function which performs the operation of usual homography warping.

4.3 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
4.3. Translation estimation among two virtual cameras

Figure 4.6: Translation between two virtual cameras. $X_1$ and $X_2$ are two arbitrary 3D point in the scene. $Z_1$ and $Z_2$ are the relative heights of $X_1$ and $X_2$ w.r.t. first camera, $V_0$. $t$ is the translation vector among two virtual cameras which can be estimated using the proposed method.

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 $t$ among virtual cameras.

Our approach is based on having the heights of two arbitrary 3D points in the scene such $X_1 = [X_1 \ Y_1 \ Z_1]^T$ and $X_2 = [X_2 \ Y_2 \ Z_2]^T$ (see Fig. 4.6) 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 $0X_1 = [0X_1 \ 0Y_1 \ 0Z_1]^T$ and $0X_2 = [0X_2 \ 0Y_2 \ 0Z_2]^T$ are coordinates of the two 3D points $X_1$ and $X_2$ expressed in the first virtual camera center, respectively. Based on the assumption, the parameters $0Z_1$ and $0Z_2$ which indicate the heights of $X_1$ and $X_2$ in ${V_0}$ are known. Recalling that $V_0$ is downward-looking and has its optical axis parallel to the gravity. Therefore the term \textit{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 \( ^0X_1 \) and \( ^0X_2 \) numerically obtained in a metric scale:

\[
\begin{align*}
^0X_1 &= ^0Z_1 \left( K^{-1}^1x_1 \right) \\
^0X_2 &= ^0Z_2 \left( K^{-1}^1x_2 \right)
\end{align*}
\]  

(4.25)

where \( ^0x_1 \) and \( ^0x_2 \) are respectively the imaged points of \( X_1 \) and \( X_2 \) in the first virtual camera image plane. The same can be considered for the second virtual camera. Suppose \( ^1X_1 = [ \begin{smallmatrix} ^1X_1 \end{smallmatrix} \begin{smallmatrix} ^1Y_1 \end{smallmatrix} \begin{smallmatrix} ^1Z_1 \end{smallmatrix} ]^T \) and \( ^1X_2 = [ \begin{smallmatrix} ^1X_2 \end{smallmatrix} \begin{smallmatrix} ^1Y_2 \end{smallmatrix} \begin{smallmatrix} ^1Z_2 \end{smallmatrix} ]^T \) are respectively coordinations of the 3D points \( X_1 \) and \( X_2 \) expressed in the second virtual camera center (\( \{ V_1 \} \)). Then likewise using projective property of a camera we can have the following equation:

\[
\begin{align*}
^1X_1 &= ^1Z_1 \left( K^{-1}^2x_1 \right) \\
^1X_2 &= ^1Z_2 \left( K^{-1}^2x_2 \right)
\end{align*}
\]

(4.26)

In contrary to the Eq. (4.25), Eq. (4.26) 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 \( (K^{-1}^2x_1) \) and \( (K^{-1}^2x_2) \) in Eq. (4.26) as well express the 3D position of the points \( ^1X_1 \) and \( ^1X_2 \) however up to scale factors \( ^1Z_1 \) and \( ^1Z_2 \). Here it is desirable to rewrite the Eq. (4.26) as the following:

\[
\begin{align*}
^1X_1 &= ^1Z_1 ^1\hat{X}_1 \\
^1X_2 &= ^1Z_2 ^1\hat{X}_2
\end{align*}
\]

(4.27)

where \( ^1\hat{X}_1 = (K^{-1}^2x_1) \) and \( ^1\hat{X}_2 = (K^{-1}^2x_2) \). Then the Eq. (4.25) and Eq. (4.27) can be related through the translation vector between \( \{ V_0 \} \) and \( \{ V_1 \} \) as:

\[
\begin{align*}
^0X_1 = R^1X_1 + t &= R^1Z_1 ^1\hat{X}_1 + t \\
^0X_2 = R^1X_2 + t &= R^1Z_2 ^1\hat{X}_2 + t
\end{align*}
\]

(4.28)
where $R$ is the rotation matrix between two cameras and $t = [ t_1 \ t_2 \ t_3 ]^T$. Since we are considering the virtual cameras and there is no rotation among them then we can simply consider $R$ as an $3 \times 3$ identity matrix. In Eq. (4.28) 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. (4.28) can be arranged in the form of

$$Ax = B \quad (4.29)$$

where

$$A = \begin{bmatrix} ^1\hat{X}_1 & 0_{3\times1} & I_{3\times3} \\ 0_{3\times1} & ^1\hat{X}_2 & I_{3\times3} \end{bmatrix} \quad (4.30)$$

$$x = \begin{bmatrix} ^1Z_1 & ^1Z_2 & t_1 & t_2 & t_3 \end{bmatrix}^T \quad (4.31)$$

$$B = \begin{bmatrix} 0X_1 \\ 0X_2 \end{bmatrix} \quad (4.32)$$

Therefore $x$ in Eq. (4.29) can be estimated using the least square approach as follows:

$$x = (A^TA)^{-1}A^TB \quad (4.33)$$

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.
4.3.1 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 $500 \times 500 \times 1000 \text{ cm}^3$. 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.

**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 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 $0.5^\circ$ on the roll and pitch angles, and $1.0^\circ$ 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 $0.001^\circ$. 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. 4.7-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 ($\delta$) for each one of the angles (roll, pitch

*Microelectromechanical systems
4.3. Translation estimation among two virtual cameras

Algorithm 4: 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 \( t \). The distributions for the input noise (IS observation) and the estimation are shown in Fig. 4.7.

\[
\begin{bmatrix}
500 & 0 & 300 \\
0 & 505 & 280 \\
0 & 0 & 1
\end{bmatrix}
/* Extrinsic parameters of two cameras */
\]

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) */

\( R \leftarrow Euler2RotationMatrix(\varepsilon_{roll}, \varepsilon_{pitch}, \varepsilon_{yaw}) 

/* translation between two cameras */

\[
\begin{bmatrix}
100 + 400 \times \text{RND} \\
100 + 400 \times \text{RND} \\
200 \times \text{RND}
\end{bmatrix}; /* 0 <= RND <= 1 (random function)
\]

/* two random 3D points (ground truth as well) */

\[
\begin{bmatrix}
-500 + 1000 \times \text{RND} \\
-500 + 1000 \times \text{RND} \\
50 + 600 \times \text{RND}
\end{bmatrix}
; \begin{bmatrix}
-500 + 1000 \times \text{RND} \\
50 + 600 \times \text{RND}
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 \times \text{X} \leftarrow 0 \times \text{X} + \text{t} \\
1 \times \text{X} \leftarrow 1 \times \text{X} + \text{t}
\end{bmatrix}
/* heights of 3D points w.r.t. the first camera reference frame */

\[
0 \times \text{h} \leftarrow 0 \times \text{X} (3) \\
1 \times \text{h} \leftarrow 1 \times \text{X} (3)
\]

/* applying the IS’ noise on two generated 3D points */

\[
\begin{bmatrix}
1 \times \text{X} \leftarrow \text{R} \times 0 \times \text{X} + \text{t} \\
1 \times \text{X} \leftarrow \text{R} \times 1 \times \text{X} + \text{t}
\end{bmatrix}
/* points on image planes */

\[
0 \times \text{x} \leftarrow \text{K1} \times 0 \times \text{X} \\
0 \times \text{x} \leftarrow \text{K1} \times 1 \times \text{X} \\
1 \times \text{x} \leftarrow \text{K2} \times 0 \times \text{X} \\
1 \times \text{x} \leftarrow \text{K2} \times 1 \times \text{X}
\]
/* estimating \( t \) */

\[
\begin{bmatrix}
0 \times \text{X} \leftarrow 0 \times \text{h} \times \text{K1}^{-1} \times 0 \times \text{x} \\
0 \times \text{X} \leftarrow 0 \times \text{h} \times \text{K1}^{-1} \times 1 \times \text{x}
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 \times \text{X} \leftarrow \text{K2}^{-1} \times 0 \times \text{x} \\
0 \times \text{X} \leftarrow \text{K2}^{-1} \times 1 \times \text{x}
\end{bmatrix}
\]

\[
A = \begin{bmatrix}
\tilde{\text{X}}_1 & 0_{3 \times 1} & I_{3 \times 3} \\
0_{3 \times 1} & \tilde{\text{X}}_2 & I_{3 \times 3}
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
0_{3 \times 1} & 0_{3 \times 1} \\
0_{3 \times 1} & 0_{3 \times 1}
\end{bmatrix}
\]

\( \text{X} \leftarrow \text{lsrvec}(A, B) */ Least-squares solution for \( AX = B */

\( \tilde{\text{t}} (1) \leftarrow \text{X} (3); \tilde{\text{t}} (2) \leftarrow \text{X} (4); \tilde{\text{t}} (3) \leftarrow \text{X} (5)
\]

\( \text{E}_{\text{XYZ}} \leftarrow \tilde{\text{t}} - \text{t} /* error in X, Y and Z axes in estimated translation */
\]

hist(\( \text{E} \)) /* plot the distribution histogram of \( \text{E} */
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 \( \delta_{\text{roll}} = 0.17°, \delta_{\text{pitch}} = 0.17° \) and \( \delta_{\text{yaw}} = 0.33° \).

These IS measurement noise are applied to the generated data in the simulation and the translation vector \( t = [X \ Y \ Z]^T \) for each sample is estimated (see Alg. 4)). Fig. 4.7-bottom depicts the error distributions for the three components of \( t \). One can see that the error distributions along three axes have Gaussian shapes as well.

**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. 4.8 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 10 cm, 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 \( t \) are plotted in blue, red and green.

**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. 4.9 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 \( N(\mu = 0, \delta = \frac{5}{2} = 1.67) \) applied to image coordinates. The error distributions for three elements of the estimated translation vector \( t \) is plotted in blue, red and green (in cm).
4.3. Translation estimation among two virtual cameras

Figure 4.7: 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 $\delta$ for each angle is considered $\frac{1}{3}$ of its corresponding maximum error value, which yields to have $\delta_{\text{roll}} = 0.17$, $\delta_{\text{pitch}} = 0.17$ and $\delta_{\text{yaw}} = 0.33$. The bottom image indicates the error distributions in cm for the three elements ($X$, $Y$ and $Z$) of estimated $t$ using simulated data (with the noise distributions shown in the top figure).
Chapter 4. Parametric homography and translation estimation

Figure 4.8: 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 10 cm, a Gaussian white noise $N(\mu = 0, \delta = \frac{10}{\sqrt{3}} = 3.33)$ is applied (plotted in purple). The error distributions for three elements of the estimated translation vector $t$ is plotted in blue, red and green.

**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. 4.10 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. 4.7-top. The diagrams of standard deviation and average for the three elements ($X$, $Y$ and $Z$) of the estimated translation vector $t$ are presented in Fig. 4.10 (in cm). As one can see, the distance of the 3D points w.r.t. the cameras has no significant effect on the accuracy of $Z$ component of the estimated $t$. However there is an almost linear relation between the distance and the error values for
4.4 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 were obtained which are able to generate the homography matrix among two planes for different cases. A new 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
Figure 4.10: 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. 4.7-left. Standard deviation and average for the three elements of the estimated translation vector $t$ are plotted (in cm).

evaluate the quality of the translation estimation method was performed.
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

\begin{algorithm}[h]
\caption{Criteria to check the edges visibility for a given polygon. $k$ is number of polygon’s edges. 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.}
\begin{algorithmic}
\For {$j = 1$ to $k$}
\If {$\exists b_i, i \in 1..n_c$ where $\angle(n_j, b_i) > \frac{\pi}{2}$}
\State consider the edge $e_j$ as <visible>
\Else
\State consider the edge $e_j$ as <invisible>
\EndIf
\EndFor
\end{algorithmic}
\end{algorithm}

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 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
5.2. Edge visibility criteria and camera configuration

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 is shown from the top view of the inertial reference plane $\pi_{ref}$. 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, $l_i$ and $r_i$, 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 $l_i$ and $r_i$ is shown in red ($b_i$). $n_i$ 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.
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 $N = \{n_1, n_2, ..., n_k\}$. 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) $(l_i, r_i)$ to the polygon and for each tangents pair a bisecting vector $b_i$ 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 $b_i$ where $\angle(n_j, b_i) > \frac{\pi}{2}$ (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.

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
Algorithm 6: Algorithm to generate a valid gene. \( V \) is the matrix of vertices of the polygon. \( \text{max}_\text{fov} \) is the maximum possible FOV for each gene(camera) and \( \text{space} \) 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 \( \text{getTangentsToPolygon}(V,p) \) 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 \( \text{createGene}() \)

Function \text{createGene}()

**Input:** \( \{V, \text{max}_\text{fov}, \text{space}\} \)

**Output:** \( \{\text{gene}\} \)

begin
    repeat
        \( p \leftarrow \text{random 2D position in space} \)
        \( [l,r] \leftarrow \text{getTangentsToPolygon}(V,p) \)
        \( \text{fov} \leftarrow \arccos\left(\frac{l.r}{|l||r|}\right) \)
        until \( \text{fov} \leq \text{max}_\text{fov} \)
        gene \( \leftarrow \text{generate_an_empty_gene()} \)
        gene.p \( \leftarrow p \)
        gene.fov \( \leftarrow \text{fov} \)
        gene.b \( \leftarrow |l|r + |r|l \)
    end
    return gene
end
Chapter 5. Contribution on sensor configuration and tracking

Figure 5.3: Structure of a chromosome string.

<table>
<thead>
<tr>
<th>chromosome</th>
</tr>
</thead>
<tbody>
<tr>
<td>gene(1)</td>
</tr>
<tr>
<td>p</td>
</tr>
</tbody>
</table>

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
5.2. Edge visibility criteria and camera configuration

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

problem, a cost function $f(\alpha = \angle(b, n)) : [0, \pi] \to [0, \lambda]$ is defined (see Fig. 5.4) for a bisector $b$ and the normal of an edge $n$ as following:

$$f(\alpha = \angle(b, n)) = \begin{cases} \frac{1}{\pi}(\alpha - \pi) & ; \frac{\pi}{2} < \alpha \leq \pi \\ \lambda & ; \text{others} \end{cases}$$

where $1 < \lambda < 2$ and $\alpha$ 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 12-19). 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 12-19). 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

$^*e_{12}$ denotes the edge connecting vertex $v_1$ to vertex $v_2$
angle among the bisector vector of the cameras and $n_1$ (normal of $e_{12}$) is straight ($\pi$).

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 12-19 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 ($b_c$ and $b_p$ 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 using the following equation:

$$\text{aug}(b_c, b_p) = |1 - \frac{\angle(b_c, b_p)}{\pi}|$$

(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; $\text{max.fov}$, 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 $\text{space}$, 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. The rest of the population (80%) is 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
Algorithm 8: Algorithm to compute the cost for the genes and chromosome. The inputs are \( V \), 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 who are visiting an edge which was previously visited by an antecedent gene of the chromosome (lines 12-19). The penalty value is obtained using Eq. (5.2).

**Function getChromosomeCost() Input: \{V, chromosome\}**

**Output:** \{cost of chromosome\}

begin
\[
\lambda \leftarrow 1.2
\]
\[
l \leftarrow \text{length(chromosome)} \quad // \text{number of genes in chromosome}
\]
for \( i \leftarrow 1 \) to \( l \) do
   gene(\( i \)).cost \leftarrow 0
   for \( j \leftarrow 1 \) to \( k \) do
      \( h \leftarrow \text{mod}(j,k) + 1 \)
      \( b \leftarrow \text{gene}(i).b \)
      \( \alpha \leftarrow \arccos\left( \frac{b.e_{jh}}{|b||e_{jh}|} \right) \)
      if \( \alpha > \frac{\pi}{2} \) and \( \alpha \leq \pi \) then
         //check if \( e_{jh} \) was previously visited by an antecedent gene
         for \( \text{prev}_j \leftarrow 1 \) to \( i - 1 \) do
            if \( \text{gene}(\text{prev}_j).e_{jh} \leq 1 \) then
               //yes visited, so penalize it!
               \( b_i \leftarrow \text{gene}(i).b \)
               \( b_p \leftarrow \text{gene}(\text{prev}_j).b \)
               \( \alpha \leftarrow \arccos\left( \frac{b_p.b_i}{|b_p||b_i|} \right) \)
               \( \text{augmented}_\text{cost} \leftarrow |1 - \frac{\alpha}{\pi}| \)
               \( \text{gene}(i).e_{jh} \leftarrow \text{gene}(i).e_{jh} + \text{augmented}_\text{cost} \)
         end
      else
         gene(\( i \)).e_{jh}.cost \leftarrow \lambda
      end
   end
   gene(\( i \)).cost \leftarrow \text{gene}(i).cost + \text{gene}(i).e_{jh}.cost
end
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 lead to having 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 that are zero since $n_1$ is opposite to their bisector vectors ($b_1$, $b_2$, and $b_3$), 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).

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
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\}

begin
  /* Initialization */
  chrom\_length \leftarrow N_c  // number of genes in a chromosome
  n_{elites} \leftarrow 20  // number of elites to be selected
  n_{pop} \leftarrow 100  // number of population
  n_{stop} \leftarrow 150  // stop after no change in a consecutive n_{stop} iterations
  search\_space \leftarrow \text{space}
  fov_{max} \leftarrow \pi / 4  // maximum possible FOV for a camera.
  n_{trial\_max} \leftarrow 2000  // maximum number of iterations

  /* First generation */
  for i \leftarrow 1 to n_{pop} do
    pop(i) \leftarrow \text{generate\_chromosome}(V, fov_{max}, chrom\_length) // Alg. 7
    evaluate\_fitness()  // using Alg. 8
    elites \leftarrow pop(1..n_{elites})
    cost\_history(1) \leftarrow \text{getChromosomeCost}(V, elites(1))  // Alg. 8
    t \leftarrow 0
    n_{repeated} \leftarrow 0

  /* Iterations */
  while t < n_{trial\_max} and n_{repeated} < n_{stop} do
    t \leftarrow t + 1
    pop(1..n_{elites}) \leftarrow elites
    pop(n_{elites} + 1..n_{pop}) \leftarrow \text{crossover and mutation on elites}
    evaluate\_fitness()  // using Alg. 8
    elites \leftarrow pop(1..n_{elites})
    cost\_history(t) \leftarrow \text{getChromosomeCost}(V, elites(1))  // Alg. 8
    if t > 1 and cost\_history(t) == cost\_history(t - 1) then
      n_{repeated} \leftarrow n_{repeated} + 1
    else
      n_{repeated} \leftarrow 0

    optimal\_chromosome \leftarrow elites(1)
  return optimal\_chromosome
end
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 section (d). The vertical axes in 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.

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

(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 \( 1200 \times 1200 \text{cm}^2 \).
Chapter 5. Contribution on sensor configuration and tracking

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 \(cm^2\).
5.3. Integration of mobile vision and laser sensor within a camera network - the estimation of extrinsic parameters

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 \( 1200 \times 1200 \text{ cm}^2 \).
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 use the proposed data registration framework), having a mobile sensor can grant some profits, as briefly discussed in Sec. 3.3.3. 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 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 ($2D^1$) 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.

**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, $\alpha$, 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:
\[
\begin{bmatrix}
  x \\
y \\
z
\end{bmatrix} =
\begin{bmatrix}
  c_i c_j & c_i d_x + s_i d_z \\
  s_j & 0 \\
-s_i c_j & -s_i d_x + c_i d_z
\end{bmatrix}
\begin{bmatrix}
  \rho_{ij} \\
  1
\end{bmatrix}
\]  
(5.3)

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

where \(\rho_{ij}\) is the \(j\)-th measured distance with corresponding orientation \(\theta_j\) in the \(i\)-th scan plane, which makes the angle \(\alpha_j\) 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.

**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 of the stereo pair. Furthermore, the IS is strongly coupled to the laser range finder and is used in order to measure the angle \(\alpha_i\). 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 \(\alpha\), which is described as

\[
C_{T_{L(\alpha)}} = \begin{bmatrix}
C R_{L(\alpha)} & C t_{L(\alpha)} \\
0_{1\times3} & 1
\end{bmatrix}
\]  
(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
5.3. Integration of mobile vision and laser sensor within a camera network - the estimation of extrinsic parameters

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)

\[ cP = CT_L^jP \]  \hspace{1cm} (5.5)

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

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 \( CT_{L(\alpha_0)} \), \( CT_{L(\alpha_1)} \) and \( CT_{L(\alpha_2)} \), which stand for the transformation ma-
tices from the \{L\} to \{C\} when the LRF is placed in three different orientations around its rotation axis, \(\alpha_0\), \(\alpha_1\) and \(\alpha_2\), respectively.

2. Obtaining \(^L_T L(\alpha)\). 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 \(\alpha\).

Firstly the LRF is placed in three different angles \(\alpha_0\), \(\alpha_1\) and \(\alpha_2\) 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 \(^c p^{\alpha_j} = \{^c p_i^{\alpha_j} | i = 1...N, j = 0...2\}\) and \(^l p^{\alpha_j} = \{^l p_i^{\alpha_j} | i = 1...N, j = 0...2\}\) 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:

\[
^c p^{\alpha_j} = ^C R_{L(\alpha_j)} ^l p^{\alpha_j} + ^C t_{L(\alpha_j)} \tag{5.6}
\]

being \(^C R_{L(\alpha_j)}\) the rotation matrix, and \(^C t_{L(\alpha_j)}\) 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 method 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:

\[
E = \sum_{j=1}^{N} \left| ^c p^{\alpha_j} - (^C R_{L(\alpha_j)} ^l p^{\alpha_j} + ^C t_{L(\alpha_j)}) \right|^2 \tag{5.7}
\]
5.3. Integration of mobile vision and laser sensor within a camera network - the estimation of extrinsic parameters

Figure 5.11: a) The LRF is placed (tilted) in three different angles $\alpha_0$, $\alpha_1$ and $\alpha_2$. 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 $\{I\}$.

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 a 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 $\alpha_j$ angle:

1. Laser Range Finder data acquisition and pre-processing. A simple method is used to distinguish the bright spot from the background. Let $l^{\alpha_j}_{i\text{back}} = \{P^{\alpha_j}_{i\text{back}}(\theta, \rho) \mid i = 1 \ldots n_l\}$ be the range data of the background (before putting the bright spot inside the LRF view field) and $l^{\alpha_j}_i = \{P^{\alpha_j}_i(\theta, \rho) \mid i = 1 \ldots n_l\}$ 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 object we can use

$$\left|l^{\alpha_j}_i - l^{\alpha_j}_{i\text{back}}\right| \geq U_{th}$$ (5.8)
where $U_{th}$ is a threshold. Thus, in order to obtain the $^{l}P_{i}^{α_j}$ 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 $^{l}P_{i}^{α_j}$ and $^{c}P_{i}^{α_j}$. The valid $^{l}P_{i}^{α_j}$ and $^{c}P_{i}^{α_j}$ are used in the Eq. 5.7 which is a least-squares solution to find $^{C}R_{L(α_j)}$ and $^{C}t_{L(α_j)}$ based on singular value decomposition (SVD) as is described in [KSAB87].

B) Obtaining $^{l}T_{L(α)}$:

In order to obtain the transformation between $\{L(α)\}$ 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 $\{L(α_0)\}$, $\{L(α_1)\}$ and $\{L(α_2)\}$, 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 $\{L(α)\}$ can be obtained by

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

(5.9)

where $\triangle$ denotes the area of the triangle. Finally, the transformation matrix $^{l}T_{L(α)}$ can be described as
5.3. Integration of mobile vision and laser sensor within a camera network - the estimation of extrinsic parameters

$$I_T L(\alpha) = \begin{bmatrix}
\cos(\alpha) & 0 & \sin(\alpha) & d \sin(\alpha) \\
0 & 1 & 0 & 0 \\
-\sin(\alpha) & 0 & \cos(\alpha) & d \cos(\alpha) \\
0 & 0 & 0 & 1 \\
\end{bmatrix}$$

(5.10)

C) Calculating $^L(\alpha) T_C$:

Let’s consider the transformation matrix from $\{L\}$ to $\{C\}$ as

$$C_T L(\alpha) = C_T I I_T L(\alpha)$$

(5.11)

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 $\alpha$ 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:

$$C_T I = C_T L(\alpha_0) I_T L^{-1}(\alpha_0)$$

(5.12)

Once $C_T I$ has been obtained, the desired transformation $C_T L(\alpha)$ can be estimated according to Eq. (5.11).

Experiments

The proposed approach is tested using the sensor platform shown in Fig. 5.12. The stereo head is the STH-MDCS from Videre Design, a compact, low-power colour digital stereo head with an IEEE 1394 digital interface. It consists of two 1.3 mega pixel, progressive scan CMOS images mounted in a rigid body, and a 1394 peripheral interface.
module, joined in an integral unit. Images obtained were restricted to $320 \times 240$ pixels. The LRF mounted on the tilt unit is an Hokuyo URG-04LX [hok], a compact laser sensor which has a resolution of $0.36^\circ$ and the field of view of $240^\circ$. Furthermore, an Xsens-MTi[xse] inertial sensor is strongly coupled to the laser range finder.

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 $\alpha = 12.1^\circ$, $\alpha = 23.2^\circ$ and $\alpha = 50^\circ$ 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 ($\alpha = 23.2^\circ$). 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 $\alpha_0 = 2^\circ$, $\alpha_1 = 12^\circ$ and $\alpha_2 = 23.2^\circ$). In another experiment, a full 3D range data and also an image are taken from a manikin 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.
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
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.

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

$$p(state_i|sensory\ input, I) = \frac{p(sensory\ input|state_i, I) \cdot p(state_i|I)}{p(sensory\ input|I)}$$  \hspace{1cm} (5.13)
in which $p(\text{sensory input}|\text{state}_i, I)$ stands for the likelihood, $p(\text{state}_i|I)$ comes from the previous knowledge of the $\text{state}_i$ (without implying new sensors observations) and $p(\text{sensory input}|I)$ is known as the evidence and actually is a normalizing constant which can be written as an integral and then:

$$p(\text{state}_i|\text{sensory input}, I) = \frac{p(\text{sensory input}|\text{state}_i, I).p(\text{state}_i|I)}{\int_\theta p(\text{sensory input}|\theta, I).p(\theta|I).d\theta} \quad (5.14)$$

As can be seen in the equation 5.14, which is known as the Bayesian equation, one can calculate the probability of a hypothesis ($\text{state}_i$) 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:

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

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

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

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:

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

then:

$$p(x_t|z_{0:t}) = \alpha p(z_t|x_t)p(x_t|z_{0:t-1}) \quad (5.18)$$
5.4. Low-level data filtering and tracking using Bayesian approach

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

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:

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

Here again using the Markov assumption we would have:

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

and consequently:

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

and finally using that to rewrite the equation 5.18:

\[
p(x_t|z_{0..t}) = \alpha p(z_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|z_{0..t-1}) dx_{t-1}
\] (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 Bayesian Occupancy Filter (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, $^l c \in y$, we have the following variables:

- $A_{c^l} \subset y$: represents antecedent set for the cell $c^l$.
- $A_{c^l}^l \in A_{c^l} \subset y$: indicates the antecedent of the cell $^l c$ at the current step.
5.4. Low-level data filtering and tracking using Bayesian approach

- \( A_{cl}^{t-1} \in A_c \subset y \): indicates the antecedent of the cell \( cl \) at the previous step.

- \( O_{cl}^t \): a boolean variable to indicate whether the cell \( cl \) at the time \( t \) is occupied \( (O_{cl}^t = 1) \) or not \( (O_{cl}^t = 0) \).

- \( Z_1, ..., Z_S \): represents measurements taken by \( S \) sensors.

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

\[
P(lA_{cl}^{t-1} A_{cl}^t O_{cl} Z_1 ... Z_S) = P(lA_{cl}^{t-1}) P(A_{cl}^t | A_{cl}^{t-1}) P(O_{cl}^t | A_{cl}^{t-1}) \prod_{i=1}^S P(Z_i^t | A_{cl}^t O_{cl}^t)
\]

(5.23)

where

- \( P(A_{cl}^{t-1}) \) is the probability for a given neighbouring cell \( A_{cl} \) to be antecedent of the cell \( cl \), belonging to the layer \( l \), at the time \( t - 1 \).

- \( P(A_{cl}^t | A_{cl}^{t-1}) \) : for a cell \( cl \), it is the distribution over antecedents at time \( t \).

- \( P(O_{cl}^t | A_{cl}^{t-1}) \): for a cell \( cl \), it is the distribution over over occupancy given the antecedents of \( cl \).

- \( P(Z_i^t | A_{cl}^t O_{cl}^t) \) 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 \( P(O_{cl}^t A_{cl}^t) \) of the system state’s probability distribution is made as \textit{a priori}. Then the predicted distribution is updated by using the current observation \( \prod_{i=1}^S P(Z_i^t | A_{cl}^t O_{cl}^t) \) which leads to have a estimation \( P(O_{cl}^t A_{cl}^t | Z^t) \) of the system state’s probability distribution.
Figure 5.18: Applying Bayesian Occupancy Filtering to deal with the dynamic of scene in the proposed registration framework.
5.4. Low-level data filtering and tracking using Bayesian approach

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.
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 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 laser range finder was proposed.
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.

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 were made to demonstrate the performance of the system for different configurations. A large set of human gestures and objects were reconstructed in 3D using this prototype.

In the proposed framework, thanks to IS, the rotations among all virtual cameras are equal to the $3 \times 3$ identity matrix. 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 $0.01^\circ$. 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 using 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 generating thousands of data in simulation. The proposed method to estimation 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 essential modality in data register scene perception, hav-
ing the further advantages of being passive sensors and of yielding additional informa-
tion, 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 sen-
sor 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 esti-
mation 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.

As one more contribution, 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.
Appendix A

Extrinsic parameter estimation among a 2D-LRF and a mono-camera
Chapter A. 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. A.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:

\[
C_X = C_R_L L_X + C_t_L
\]  

(A.1)

where \( C_R_L = [ r_1 \ r_2 \ r_3 ]^t \) and \( t = t_x \ t_y \ t_z \) 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. A.1 where \( C_R_L \) and \( C_t_L \) can be solved by applying a least-squares approach \[KSAB87\] to minimize the following function

\[
\min_{R,I} \sum_{i=1}^{n} \left\| C_R_L L_X_i + C_t_L \right\|^2
\]  

(A.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 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 \( p_i = (u_i, v_i, 1)^t \). Using collinearity
of the center of projection, $p_i$ and $X_i$ we will have the following equations [LHM00]:

$$u_i = \frac{r_1^LT_x + t_x}{r_3^LT_x + t_z}$$ \hspace{1cm} (A.3)

$$v_i = \frac{r_2^LT_x + t_y}{r_3^LT_x + t_z}$$ \hspace{1cm} (A.4)

and the collinearity equation [LHM00] can be written as

$$v_i = \frac{1}{r_3^LT_x + t_z} (R^LX_i + t)$$ \hspace{1cm} (A.5)

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

$$V_i = \frac{v_i v_i^T}{v_i^Tv_i}$$ \hspace{1cm} (A.6)

and an error vector can be defined as following [LHM00]
\[ e_i = (I - V_i)(R^L X + t) \]  \hspace{1cm} (A.7)

After, \( R \) and \( t \) can be estimated by minimizing the sum of the squared error over them, based on Lu’s [LHM00]’s method:

\[ E(R, t) = \sum_{i=1}^{n} \|e_i\|^2 = \sum_{i=1}^{n} \|(I - V_i)(R^L X + t)\|^2 \]  \hspace{1cm} (A.8)

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

\[ C_{T_L} = \begin{bmatrix} CR_L & C_{t_L} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \]  \hspace{1cm} (A.9)

**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
Figure A.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.

cameras laser range finder. There is even an advantage that the cameras can have no overlap in their FOV. Fig. A.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 A.3 shows a exemplary scenario in 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. A.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,
Chapter A. Extrinsic parameter estimation among a 2D-LRF and a mono-camera

namely $C_2^T L$. Then the transformation between C2 and C1 can be easily expressed as:

$$C_1^T C_2 = C_1^T L C_2^T L^{-1}$$  \hspace{1cm} (A.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.
Bibliography
Chapter A. Extrinsic parameter estimation among a 2D-LRF and a mono-camera

Bibliography


BIBLIOGRAPHY


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


[BUZ03] DAN CALIN BUZAN. Robust tracking of human motion. Master’s thesis, Computer Science Department, Graduate School of Arts and Sciences, Boston University, 2003.


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


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


[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.


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


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


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


[NNT07] Christian Nitschke, Atsushi Nakazawa, and Haruo Takemura. Real-time


[OD02] T. Okatani and K. Deguchi. Robust estimation of camera translation be-
tween two images using a camera with a 3d orientation sensor. In *Pattern


[PD03] F. Porikli and A. Divakaran. Multi-camera calibration, object tracking
and query generation. In *Multimedia and Expo, 2003. ICME ’03. Pro-

[PMJ04] Jorge S. Marques Pedro M. Jorge, Arnaldo J. Abrantes. On-line ob-
ject tracking with bayesian networks. In *Proceeding of Inetrnational
Workshop on Image Analysis for Multimedia Inetractive Systems, Lisboa

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


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

[QRA+11] Joao Quintas, Kamrad Khoshhal Roudposhti, Hadi Aliakbarpour, Martin
Hofmann, and Jorge Dias. Using concurrent hidden markov models to ana-
lyze human behaviours in a smart home environment. In *Proceeding of International Workshop on Image Analysis for Multimedia Interactive
Services (WIAMIS 2011), University of Delft, Netherlands.*, 2011.

[RAQ+11] Kamrad Khoshhal Roudposhti, Hadi Aliakbarpour, Joao Quintas, Martin
Hofmann, and Jorge Dias. Probabilistic lma-based human motion ana-
sis by conjugating frequency and spatial based features. In *12th interna-
tional Workshop on Image Analysis for Multimedia Interactive Services
(WIAMIS’11), 2011.*
Chapter A. Extrinsic parameter estimation among a 2D-LRF and a mono-camera


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


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

