Perception and interpretation of dynamic scenarios using lidar data and images

Institut de Robòtica
i Informàtica Industrial

Agustin Alberto Ortega Jimenez

Institut de Robótica i Informática Industrial

Universitat Politécnica de Catalunya

Consejo Superior de Investigaciones Científicas

A thesis submitted for the degree of

Doctor of Philosophy

Barcelona, September 2015

Universitat Politècnica de Catalunya
BarcelonaTech (UPC)

PhD program:
Automatic Control, Robotics and Computer Vision

The work presented in this thesis has been carried out at:

Institut de Robòtica i Informàtica Industrial, CSIC-UPC

Thesis supervisor:
Juan Andrade-Cetto

Thesis committee:
José Gaspar, President
Francesc Moreno Noguer, Secretary
Simon Lacroix

Dedication

To my parents Angel and Margarita.

Acknowledgements

I would like to express all my gratitude to all the people that supported me during all this time. First I would like to thank my advisor Juan Andrade-Cetto, for the support all these years for concluding my thesis, as well as for his support in difficult times that I have lived along my Phd studies. Also I would like to thank the former director at IRI, Alberto Sanfeliu for giving me the opportunity to work in the Institut de Robótica i Informática Industrial, and to the staff at the Institut for their support over the years.

During these years I have carried out internships in two different institutions that have contributed to the development of this thesis. First, I would like to thank specially J. Gaspar and A. Bernardino for a fruitful stay at the Instituto Superior Técnico (IST) in Lisbon, Portugal; and to S. Lacroix and A. Maligo for their assistance during the experiments made at LAAS, and in general to all the personnel at the Laboratoire d'Analyse et d'Architecture des Systémes (LAAS) - CNRS, Toulouse, France.

I would also like to thank my friends Ernesto Teniente, Monika Mac, Leonel Rozo, Magda Kosno, María Sarmiento, Kasia Pajeska,and Rafael Valencia for sharing great moments in Barcelona during my studies. To my officemates Anaís Garrel and Michael Villamizar that shared the workplace with me during all those years.

I would like to say thanks also to Alexia Denisot that has been my support and motivation during the last years of my Phd studies. For the good moments together and specially for supporting me during the difficult times, always looking ahead and making us stronger. To her parents Alain Denisot, Lynda Collaou, and her sister Marina Denisot. Also to their grand parents Andree Angels, and Jean-Claude Vallet, that have supported me and opened their arms unconditionally making me a part of their family.

During 2014 I continued my research at UMR-CNRS 6134 SPE - University of Corsica. During this time Lucile Rossi and Tom Toulouse also have helped me and supported me with

comments of encouragement to finish my thesis. I thank them for that.

I would like to thank an important person that supported me during my stay in Corsica, my best friend all this time helping me to keep going, giving me motivation and unconditional friendship when I needed it more. Thanks Marina Beshara. You know that always you will have an important place in my heart.

I am so grateful with my parents Angel and Margarita that all the time were supporting me during my professional life. They have helped me to reach my professional goals and dreams. To my brothers Ángel and Víctor that have been my example to continue always ahead still in difficult times. My nieces Paola and Ixtchel that are my inspiration for those happy moments together. To my sister in law Sarahi and my beautiful niece Aztli for all those great moments in San Francisco. Thanks to the whole Ortega family always encouraging me to go forward.

Finally, I would like to acknowledge all sources of financial support that in one way or another contributed for the development of this thesis. These include, a PhD scholarship from the Mexican Council of Science and Technology (CONACYT) with the scholarship number 181412, a Research Stay Grant from the Agencia de Gestio d'Ajuts Universitaris i de Recerca of The Generalitat of Catalonia (CTP 2013), the Consolidated Research Group VIS (SGR2009-2013), a Technology Transfer Contract with the Asociación de la Industria Navarra, the National Research Projects PAU and PAU+ (DPI2008-06022 and DPI2011-2751) funded by the Spanish Ministry of Economy and Competitivenes, and the EU URUS project (IST-FP6-STREP-045062).

Reconocimientos

Quisiera expresar mi gratitud a todas las personas que me han apoyado durante todo este tiempo. Primero me gustaría agradecer a mi tutor de tesis Juan Andrade-Cetto por todo el apoyo estos años para concluir la tesis, tambien por apoyarme en momentos difíciles que he vivido a lo largo de mi doctorado. Tambien quisiera agradecer al exdirector del IRI, Alberto Sanfeliu por darme la oportunidad de trabajar en el Institut de Robótica i Informática Industrial, y al personal del Instituto por su apoyo todos estos años.

Durante estos años he llevado a cabo estancias de investigación en dos diferentes instituciones que han contribuido para el desarrollo de esta tesis. Primero, quisiera agradecer especialmente a J. Gaspar y A. Bernardino por la estancia fructifera en el Instituto Superior Técnico (IST) en Lisboa, Portugal; y a S. Lacroix y A. Maligo por su asistencia durante los experimentos hechos en LAAS y en general a todo el personal en el Laboratorio d'Analyse et d'Architecture des Systémes (LAAS) - CNRS, Toulouse, France.

Quisera agradecer a mis amigos Ernesto Teniente, Monika Mac, Leonel Rozo, Magda Kosno, María Sarmiento, Kasia Pajeska, y Rafael Valencia por compartir grandes momentos conmigo en Barcelona durante mis estudios. A mis compañeros de despacho Anaís Garrel y Michael Villamizar en el cual compartimos lugar de trabajo todos estos estos años.

Quisiera decir gracias a Alexia Denisot que ha sido un gran apoyo y motivación durante los últimos años de mi doctorado. Por todos esos buenos momentos juntos y especialmente por el apoyo en momentos difíciles siempre mirando hacia delante y haciéndonos mas fuertes. A sus padres Alain Denisot, Lynda Collaou, y su hermana Marina Denisot. Tambien a sus abuelos Andree Angels, and Jean-Claude Vallet, que me han apoyado y han abierto sus manos incondicionalmente haciéndome parte de su familia.

Durante el 2014 continué mi investigación en la Universidad de Córsega. Durante este

tiempo Lucile Rossi y Tom Touluse tambien han ayudado y apoyado con comentarios de motivación para finalizar esta tesis. Gracias a ellos por eso.

Quisiera agradecer a una persona muy importante que me apoyó durante mi estancia en Córsega, mi mejor amiga que durante todo este tiempo que me ayudo a continuar adelante, motivándome y con una amistad incondicional cuando mas la necesitaba. Gracias Marina Beshara. Sabes que siempre tendrás un lugar importante en mi corazón.

Estoy tan agradecido con mis padres Angel y Margarita que todo el tiempo han sido mi apoyo durante mi vida profesional. Ellos me han ayudado para alcanzar mis metas profesionales y sueños. A mis hermanos Ángel y Víctor que han sido mi ejemplo para continuar siempre adelante aún en momentos difíciles. A mis sobrinas Paola e Ixtchel que son mi inspiración por esos momentos juntos. A mi cuñada Sarahi y mi guapa sobrina Aztli for esos grandes momentos en San Francisco. Gracias a la completa familai Ortega animándome siempre a ir adelante.

Finalmente, quisiera agradecer todas fuentes de finaciamiento que de una manera u otra han contribuido al desarrollo de esta tesis. Esto incluye la beca de doctorado del Consejo Nacional de Ciencia y Tecnología (CONACyT) con número de beca 181412, una beca de movilidad de la Agencia de Gestio d'Ajuts Universitaris i de Recerca de la Generalitat de Catalunya (CTP 2013), el grupo consolidado de Investigación VIS (SGR2009-2013), un contrato de transferencia de la Asociación de la Industria Navarra, y los proyectos nacionales de investigación PAU y PAU+ (DPI2008-06022 y DPI2011-2751) financiados por el Ministerio Español de Economía y Competividad, y el proyecto europeo URUS (IST-FP6-STREP-045062).

Remerciements

Je tiens à exprimer toute ma gratitude auprès des personnes qui m'ont soutenues pendant toute cette période. Tout d'abord, je voudrais remercier mon directeur de thèse, Monsieur Juan Andrade-Cetto pour son soutien durant toutes ces années qui m'a encouragé à terminer ma thèse et soutenu dans les moments difficiles. De plus, je tiens à remercier l'ancien directeur, Monsieur Alberto Sanfeliu de l'Institut Robótica i Informática Industrial qui m'a accordé l'opportunité de travailler dans son institut ainsi que le personnel de l'institut pour son soutien durant ces années.

Durant les dernières années de ma thèse, j'ai effectué des stages dans deux instituts de recherches qui m'ont aidé à l'élaboration de cette thèse. Dans un premier temps, je veux spécialement remercier J. Gaspar et A. Bernardino pour le séjour enrichissant à l'Instituto Superior Tecnico (IST) de Lisbonne au Portugal; S. Lacroix et A. Maligo pour leur assistance pendant les experiences menées au LAAS et de manière plus générale, tout le personnel du Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) CNRS de Toulouse, France.

Je tiens également à remercier mes amis Ernesto Teniente, Monika Mac, Leonel Rozo, Magda Kosno, María Sarmiento, Kasia Pajeska, et Rafael Valencia avec qui j'ai partagé de grands moments à Barcelone pendant mes études. A mes collègues de bureau Anaís Garrel et Michael Villamizar avec qui j'ai collaboré pendant toutes ces années.

Je voudrais aussi dire merci à Alexia Denisot qui a été mon soutien et ma motivation pendant les dernières années de mes études de doctorat. Pour les bons moments passés ensemble et spécialement pour m'avoir soutenu dans les moments difficiles, elle m'a poussé à toujours aller de l'avant afin de nous rendre plus forts. A ses parents, également Alain Denisot, Lynda Collaou et sa sœur Marina Denisot. Aussi à leurs grands-parents Andrée Angels et Jean-Claude Vallet, qui m'ont soutenu et m'ont ouvert leurs bras inconditionnellement. Ils m'ont montré que je faisais parti de leur famille.

Au cours de l'année 2014, je poursuivais mes recherches à l'UMR-CNRS 6134 SPE - Université de Corse. Pendant ce temps, Lucile Rossi et Tom Toulouse m'ont également aidé et soutenu avec des commentaires encourageants pour terminer ma thèse. Je les remercie pour cela.

Je tiens à remercier une personne importante qui m'a soutenue pendant mon séjour en Corse, ma meilleure amie durant tout ce temps, me poussant à continuer, me motivant et me donnant son amitié inconditionnelle quand je n'en pouvais plus. Merci Marina Beshara. Tu sais que toujours tu auras une place importante dans mon cœur.

Enfin, je tiens à remercier toutes les sources de soutien financier qui, d'une manière ou d'une autre, ont contribué à l'élaboration de cette thèse. Ceux-ci comprennent, un doctorat de bourse du Conseil mexicain de la science et de la technologie (CONACYT) octroyant un nombre de bourses d'études de 181412, un séjour de recherche Grant de l'Agencia de Gestio d'Ajuts Universitaris i de Recerca de la Generalitat de Catalogne (CTP 2013), le Groupe de recherche consolidé VIS (SGR2009-2013), un Transfert de technologie, Contrat avec l'Asociación de la Industria Navarra, le national de recherches Projets PAU et PAU+ (DPI2008-06022 et DPI2011-2751) financés par le ministère espagnol de l'Economie et COMPETITIVITE, et le projet de l'UE URUS (IST-FP6-STREP-045062).

Abstract

This thesis reports research on the fusion of data coming from laser range scanners and cameras for scene interpretation. These devices are complementary in that one provides information about the distance at which objects are located, whereas the second one provides information about their appearance. We provide solutions that show how one can be used to help in the calibration of the other one, and in such case, how the noise of the first propagates to the estimates computed by the second. Moreover, to provide a tight integration of the two we develop solutions not only for the accurate geometric calibration between them, but also for their correct synchornization.

We also studied how the combination of the two sensors can be exploited to identify dynamic scene events. Once the two sensors are geometrically calibrated, we can reliable associate low level features extracted in each of them. We exploit such tight correspondence for the accurate annotation of dynamic events occurring in the scene, and are able to segment out those moving elements (people) from an otherwise static scene combining the data from laser range finders and cameras. In the quest for an adequate data fusion algorithm we encountered another often overlooked sensor calibration problem, that of sensor synchronization. We provide solutions to the synchronization between a camera and a low-rate high-density laser range scanner, and also with a high-rate low-density range scanner. In the end, we provide alternatives for the synchronization and also for the data fusion between camera images and each of the two range sensors using Gaussian mixture models as the core fusion methodology.

The thesis was developed in the context of the national projects PAU (DPI2008-06022) and PAU+ (DPI2011-2751), and of the EU project URUS (IST-FP6-STREP-045062).

Resumen

Esta tesis aborda el problema de fusión de datos a partir de distintas modalidades sensoriales, sensores láser de distancia y cámaras, para la interpretación de escenas. Ambos dispositivos son complementarios el uno del otro. El primero proporciona información sobre la distancia a la que se encuentran los objetos mientras que el segundo proporciona información acerca de su apariencia. En esta tesis proveemos de soluciones que muestran como el primero puede utilizarse para la calibración del segundo, y en dado caso, como se propaga la incerteza en las lecturas del primero a la estimación de los parámetros del segundo. Además para poder obtener una integración a bajo nivel de ambos, desarrollamos soluciones que permiten relacionar espacialmente el uno con el otro además de conseguir su adecuada sincronización temporal.

La tesis aborda también el uso de ambos sensores para la identificación de eventos de una escena en movimiento. Una vez que los sensores han sido calibrados geométricamente, podemos asociar las características de bajo nivel calculadas de cada uno de ellos y explotamos esta asociación de características para anotar los eventos que ocurren en la escena y segmentar los elementos que se mueven (personas). Durante la búsqueda de una técnica adecuada para la fusión de la información de ambos sensores, encontramos necesario abordar un problema que habitualmente no es estudiado con rigurosidad, el de la sincronización de los mismos. Esta tesis proporciona dos soluciones distintas para la sincronización entre una cámara y dos tipos de sensores láser. Por un lado, una solución que permite sincronizar la cámara con un sensor láser de frecuencia de adquisición de datos baja pero alta resolución, y un segundo método para sincronizar un sensor láser de frecuencia elevada de adquisición de datos pero de baja resolución espacial. Además de proponer estas alternativas de sincronización, la tesis presenta resultados de fusión de datos usando mezclas de Gaussianas para la detección de eventos dinámicos.

La tesis se ha desarrollado en el contexto de los proyectos del Plan Nacional de I+D PAU

(DPI2008-06022) y PAU+ (DPI2011-2751), y del proyecto europeo URUS (IST-FP6-STREP-045062).

Contents

List of Figures

Chapter 1

Introduction

In robotics and computer vision, scene interpretation is referred to the task of analyzing sensory data to come up with hypotheses of the events occurring in the real world. Broadly speaking, scene interpretation generalizes many perception problems such as traffic monitoring, generic object recognition [91], detection of moving objects in crowded urban areas [95], or mobile robot localization [48, 69].

All these sample applications of scene interpretation require the fusion of data coming from multiple complementary sources. For instance, global positioning systems (GPS) have the capability of measuring the location of a moving system in open space, and inertial measurement units (IMUs) provide the rotational velocity and the linear acceleration of such moving system. By fusing the data coming from these two propiceptive sensors, one can compute very accurate estimates about its position and velocity. The same applies for exteroceptive sensing modalities such as sonars, lasers or cameras. The first two measure the distance from the moving system to other objects, whereas the last one can measure their appearance, and when several images from different viewpoints are analyzed, they also provide estimates about the distances to those other objects. The complementarity of the various sensing modalities allows for a better interpretation of the environment.

In this thesis we concentrate on the fusion of two of these sensors, namely laser range scanners and cameras. We study how one can be used to help in the calibration of the other one, and in such case, how the noise of the first propagates to the estimates computed by the second. Moreover, to provide a tight integration of the two we develop solutions not only for the accurate geometric calibration between them, but also for their correct synchornization.

Laser range scanners have become ubiquitous sensors in robotics applications, mainly because their steep reduction in cost in recent years. A decade ago a real-time 3D range scanner of the ones used for the DARPA Grand Challenge would cost about 100,000 US dollars. These systems have dropped an order of magnitude, and we can find sensors with similar capabilities now for a tenth of their original price. Conventional laser range scanners of the kind used in robotics applications provide distance measurements to surrounding objects a hundred of meters away with an accuracy in the centimeter range, depending on the illuminated object reflectance properties, and find applications for instance in moving object detection, recognition [44], or tracking [72].

Another advantage of laser range scanners is that their distance readings are not affected substantially from illumination conditions, a situation that other distance measurement devices such as time of flight (ToF) cameras do not share. In addition lasers provide less echo artifacts than other distance measurement devices such as sonars and radars.

The main limitation of laser range scanners is that although reflectance information can be read from the sensor signal, it is not as rich as that of cameras, and properties such as color or texture cannot be reliable computed from them.

Cameras are also ubiquitous not only in robotics applications [10], but in a much larger scope. As of today, a common household might have tens of cameras attached to many devices, from mobile phones, to computers, video games, tablets, surveillance and security systems, cars, etc. Hence computer vision applications for scene interpretation have multiplied in recent years [22, 68, 93]. Camera prices are even lower than that of lasers, and their size has also reduced to the point in which they can be embedded almost everywhere. Cameras provide information about the appearance of objects but alone do not provide estimates of the distance at which those objects are located. It is through the triangulation of multiple images from different vantage points that one can infer the distance to an object, but perspective geometry forces for a large baseline to exist between the different views for a distance measurement to be reliable. However, when two images of the same object are taken from distant points, the object appearance can change substantially and make distance computation a difficult task. Hence, lasers and cameras are naturally complementary devices for the measurement of appearance and distance properties of objects in a scene.

In this thesis we exploit such complementarity. First, in Chapter 3, we use a laser-made 3D map of a large area to aid in the calibration of a network of non-overlapping cameras.

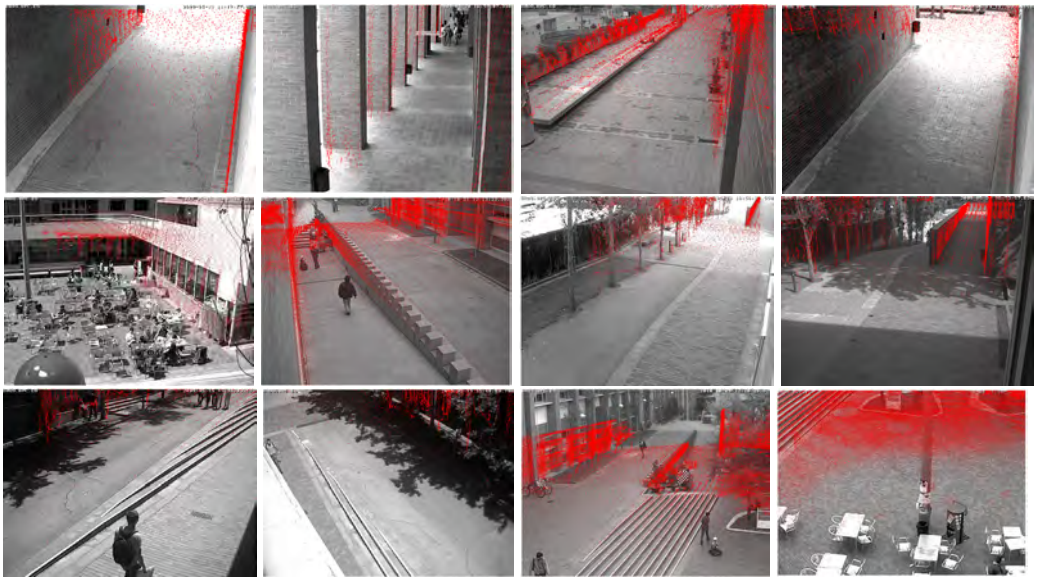


Figure 1.1: Projection of a laser-computed 3D map on images from a camera network.

Environmental conditions such as wind or drastic temperature changes often call for automatic methods to be able to recalibrate such large camera networks. Our laser-made map helps recover not only very accurate estimates about the position and orientation of each camera in the network, but also helps refine their intrinsic properties. Figure 1.1 shows such application, in which the 3D map is reprojected to each of the cameras in the network. Notice how it would be nearly impossible to accurately relate geometrically one camera to the other in the network should the laser map not be available. The overlapping between the cameras is significantly non existent.

Using data from a second device, in this case the laser range finder, to calibrate a first device or set of devices, in this case the camera network, poses a relevant question: what is the amount of noise introduced by the first sensor into the calibration estimate of the second one. In Chapter 4 we address this question rigorously, propagating the noise model of the laser range finder through the 3D reconstruction and the intrinsic and extrinsic parameter estimation functions. This uncertainty propagation is approximated with first order models, and Monte Carlo simulations are used to demonstrate that such first order models provide consistent estimates of the seeked parameters for normal variable ranges. Once the simulations were finalized, we applied the method to real experimental conditions, and showed how the uncertainty in one sensor propagates to the calibration of the other.

The solutions reported in these two Chapters relate to static imagery. We are also interested in analyzing how the combination of the two sensors can be exploited to identify dynamic scene events. Dynamic event recognition is of utmost importance in many computer vision and robotics applications, such as surveillance, otlier removal, etc. [28, 31, 95]. Once the two sensors are geometrically calibrated, we can reliable associate low level features extracted in each of them, i.e., the projection of each point in the range map to its corresponding camera image coordinates. In Chapters 5 and 6 we exploit such tight correspondence for the accurate annotation of dynamic events occurring in a scene. We are able to segment out those moving elements (people) from an otherwise static scene combining the data from laser range finders and cameras. In the quest for an adequate data fusion algorithm we encountered another often overlooked sensor calibration problem, that of sensor synchronization. We provide solutions to the synchronization between a camera and a low-rate high-density laser range scanner, and also with a high-rate low-density range scanner. The significantly different acquisition rates of the two sensors called for different synchronization solutions. In the end, we provide alternatives for the synchronization and also for the data fusion between camera images and each of the two range sensors using Gaussian mixture models as the core fusion methodology.

Figure 1.2 shows how the two sensing modalities act complementarily for the detection of motion events. The top frame shows in green the motion estimates computed purely from the camera images. Notice how reflectances and sensor saturation produce false positive detections. The middle frame shows in blue those estimates computed in the laser range image. Notice how the low resolution in the laser range image leaves undetected some elements such as person arms and bottom parts of the legs. An adaptive mixture of local experts algorithm helps reconcile the estimates from the two sensors to come up with a better interpretation of the dynamic elements occurring in the scene, as shown in the bottom frame.

1.0.1 Summary of contributions

Figure 1.3 presents an outline of this thesis. We divide this document in 3 principal topics: 3D segmentation for planar structures, camera calibration and uncertainty analysis, and segmentation of dynamic objects. The thesis contributions, chapter by chapter are:

In **Chapter 2**, we present an efficient graph-based algorithm for the segmentation of planar regions out of 3D range maps of urban areas. The algorithm is motivated by Felzenszwalb's algorithm for 2D image segmentation [18], and is extended to deal with non-uniformly sampled

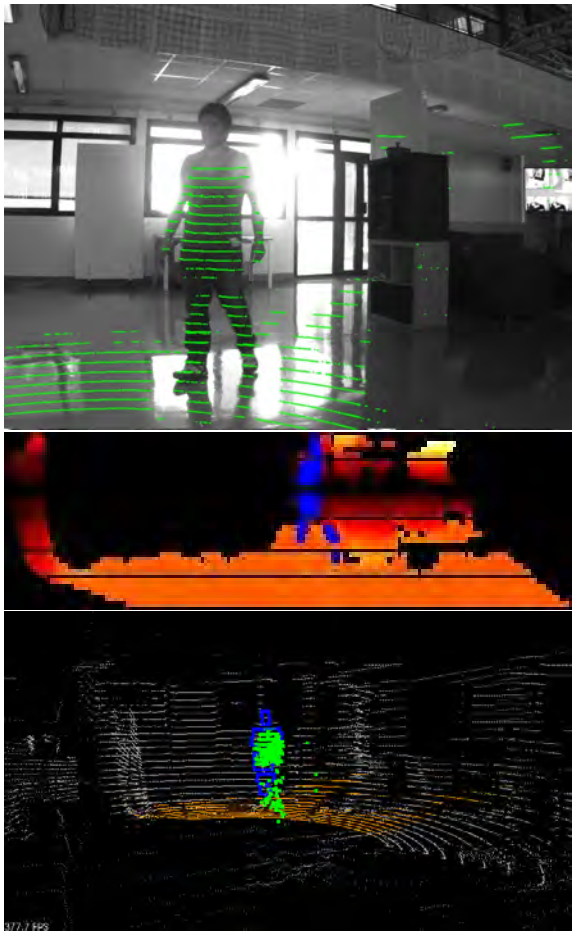


Figure 1.2: Recognition of motion events using a high-rate-low-resolution scanner

3D range data using an approximate nearest neighbor (ANN) search. Inter-point distances are sorted in increasing order and this list of distances is traversed by growing planar regions that satisfy both local and global variation of distance and curvature. We compare in our experiments the proposed method vs. a method that uses expectation maximization (EM). In contrast to EM, no prior knwoledge about the number of segments in the scene is needed, and the algorithm runs in $O(n \log n)$, with n the number of points in the point cloud. We present our results of the segmentation algorithm at ECMR 2009 [55].

In **Chapter** 3 the segmentation algorithm is used to extract planes of a range map, and 3D lines are recovered from the intersection of such planes. These 3D lines are the used for the calibration of a large network of cameras. We developed a semi-automatic procedure for the calibration of the camera network with non-overlapping fields of view. The method is de-

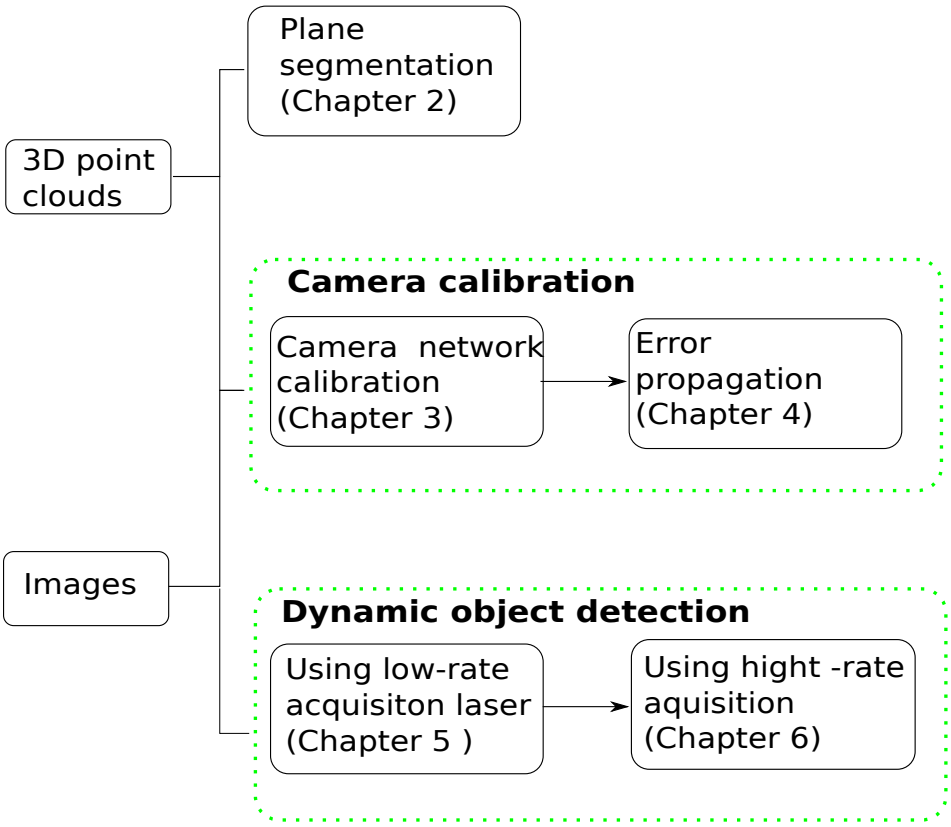


Figure 1.3: Thesis outline

vised as a two-step proces in which first camera calibration is achieved through the DLT-Lines algorithm, matching image lines to 3D lines, and the results are further refined using Lambert-Marquardt optimization over a reprojection error cost function. The method is validated for the calibration of the Barcelona RobotLab camera network [84], and in the geottaging of imagery taken at UPC's Facultat de Matemàtiques i Estadística. A direct result of the proposed calibration procedure is the ability to create direct mappings (homographies) between image coordinates and world points on the ground plane (walking areas) that support person and robot detection and localization algorithms. The technique was presented at IROS 2009 [54], and its application for robot mapping at the IROS'09 Workshop on Network Robot Systems [4]. An extended version of the method was published at the journal Sensors [56]. The method was a direct contribution and a result of the European Union-funded project "Ubiquitous networking robotics in urban settings" (URUS). This part of the thesis was carried in collaboration and also during a research stay at Instituto Superior Técnico at Lisbon (IST), Portugal.

Chapter 4 addresses the problem of estimating the amount of error introduced by one sensing modality when it is used to aid in the calibration of another. In particular, to estimate how noise in the measurement of range data propagates to the estimation of intrinsic and extrinsic parameters of the cameras. We present a noise propagation uncertainty analysis for the specific case of the DLT-Lines algorithm [74]. Once the projection matrix is computed for each camera, the error sources are propagated through the projection matrix towards the position of the camera. We validated the consistency of the uncertainty analysis with Monte Carlo simulations, and applied the technique in a real camera network. This allowed us to evaluate the accuracy of DLT-Lines algorithm in real settings. We presented the results for the uncertainty analysis at ECMR 2013 [58] and CVIU 2015 [23]. This part of the thesis emerged also from the collaboration with researchers at Instituto Superior Técnico at Lisbon (IST), Portugal.

In **Chapter** 5 we present a method to segment dynamic objects from a high-resolution and low-rate acquisition 3D scanner. Data points are tagged as static or dynamic based on the classification of pixel data from registered imagery. Per-pixel background classes are adapted online as Gaussian mixtures, and their matching 3D points are classified accordingly. We analyzed the correct calibration and synchronization of the scanner with the the accessory camera. The presented results of the method are shown for a small indoor sequence with several people following arbitrarily different trajectories. We published our results at ECMR 2011 [52] and in a poster session during the summer school at Ecole Normal Superior in Paris 2011 [57].

Finally, **Chapter** 6 presents a method to segment dynamic objects using a high-rate sensor. We fuse Gaussians mixtures of detection hypotheses from range and visible images by using the method of adaptive mixture of local experts, which adaptively learns weights for the contribution of each sensor from incoming data. In this case, the laser range scanner used was a Velodyne $32E$, which acquires data in real time. In this chapter and also in chapter 5, we paid special attention to sensor synchronization, providing solutions for laser range to image registration for both low rate and high rate laser sensors. The results of this chapter were developed during a research stay at the Laboratoire d'Analyse et d'Architecture des Systémes (LAAS-CNRS), Toulouse, France, in 2013. We published the results at the CIMAT-X Workshop on Image processing, Guanajuato Mexico, 2014 [53].

1.0.2 Publications derived from this thesis

The publications derived from this thesis are:

- R. Galego, **A. Ortega**, R. Ferreira, A. Bernardino, J. Andrade-Cetto, and J. Gaspar Uncertainty analysis of the DLT-lines calibration algorithm for cameras with radial distortion. Computer Vision and Image Understanding, 2015, In press, http://dx.doi.org/10.1016/j.cviu.2015.05.015 [23].

- **A. Ortega**, M. Silva, E.H. Teniente, R. Ferreira, A. Bernardino, J. Gaspar, and J. Andrade-Cetto. Calibration of an outdoor distributed camera network with a 3D point cloud. Sensors, 14(8):13708-13729, 2014. http://dx.doi.org/10.3390/s140813708 [56].

- **A. Ortega** and J. Andrade-Cetto. Dynamic object detection fusing LIDAR data and images, X Workshop on Image Processing, Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Mexico, Oct. 2014. http://www.iri.upc.edu/download/scidoc/1563 [53].

- **A. Ortega**, R. Galego, R. Ferreira, A. Bernardino, J. Gaspar and J. Andrade-Cetto. Estimation of camera calibration uncertainty using LIDAR data. In Proc. European Conference on Mobile Robots, Barcelona, Spain, Sept. 2013. http://dx.doi.org/10.1109/ECMR.2013.6698868 [58].

- **A. Ortega** and J. Andrade-Cetto. Segmentation of dynamic objects from laser data. In Proc. European Conference On Mobile Robots, Orebro, Sweden, Sept. 2011. http://www.iri.upc.edu/download/scidoc/1259 [52].

- **A. Ortega**, J. Andrade-Cetto, Segmentation of dynamic objects from low acquisition rate range data (poster) ENS/INRIA Visual Recognition and Machine Learning Summer School, Paris France, 2011. http://www.iri.upc.edu/download/scidoc/1659 [57].

- J. Andrade-Cetto, **A. Ortega**, E. Teniente, E. Trulls, R. Valencia, and A. Sanfeliu. Combination of distributed camera network and laser-based 3D mapping for urban service robotics. In Workshop on Network Robots Systems IEEE/RSJ Conf. Intell. Robots Syst., St. Louis, MO, USA, Oct. 2009. http://www.iri.upc.edu/download/scidoc/1039 [4].

- **A. Ortega**, B. Dias, E. Teniente, A. Bernardino, J. Gaspar, and J. Andrade-Cetto. Calibrating an outdoor distributed camera network using laser range finder data. In Proc. IEEE/RSJ Conf. Intell. Robots Syst., St. Louis, MO, Oct. 2009. http://dx.doi.org/10.1109/IROS.2009.5354294 [54].

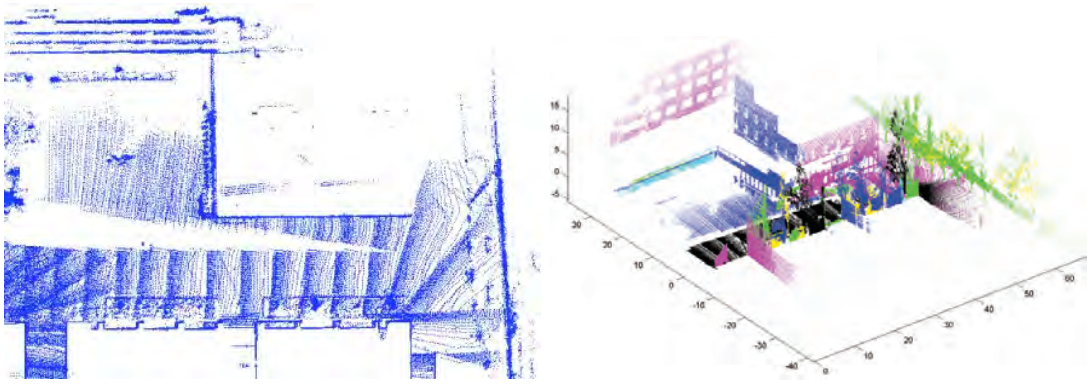
- **A. Ortega**, I. Haddad, and J. Andrade-Cetto. Graph-based segmentation of range data with applications to 3D urban mapping. In Proc. European Conference on Mobile Robots, Mlini/Dubrovnik, Croatia, Sept. 2009. http://www.iri.upc.edu/download/scidoc/1022 [55].

Chapter 2

3D planar segmentation

In this chapter we present one technique to segment planar surfaces of outdoor urban areas. This method works on point clouds with 3D sparse information and segments out planes to be later used to produce traversability maps, to aid in the calibration of a camera network, or to generate VR models of the scene. The method is motivated by a graph-based image segmentation algorithm [18], that has been modified to deal with non-uniformly distributed 3D range data. Figure 2.1 shows an aerial view of a section of our application scenario, the Barcelona Robot Lab (BRL) located at the Campus Nord of the Universitat Politècnica de Catalunya (UPC). The BRL is a 10,000 m^2 facility for research in outdoor service mobile robotics, and is equipped with a camera network together. A 3D map of the area was produced with the method reported in Valencia et al., [90].

The segmentation of 3D range maps into planar surfaces is usually addressed by region growing algorithms. The system presented by Poppinga et al., [59] for instance, contains a number of heuristics to obtain incremental plane fitting with the assumption that nearest neighbors are taken directly from the indexes in the range image. Moreover, its secondary polygonalization step is viewpoint dependent, relying also on the neighboring associations given by the indexes of the range data. In contrast, in our method, nearest neighbors are obtained using an efficient approximate nearest neighbor search over the entire 3D point map. If the number of planes to detect is known a priori, EM can be used to assign points to planes in terms of normal similarity, density of points and curvature [36]. The technique is shown for indoor scenes in which planar patches are usually orthogonal to each other. For larger, sparser point distributions, such as the ones found in our outdoor range data, the assumption of an a priori knowledge of the number of planes is unrealistic. To this end, hierarchical EM can be used



(a) (b)

Figure 2.1: Partial view of the Barcelona Robot Lab. The segmentation results shown correspond to a search for 30 nearest neighbors per point, 0.5 m distance threshold, and 0.5 curvature threshold. a) Unsegmented map (top view). b) Segmented planes.

[86], incrementally reducing the number of planes with a Bayesian information criterion, at the expense of higher computational cost.

Contrary to region growing, one could search for region boundaries instead. A good exemplar of this technique is presented in an architectural modeling application [12], in which polyhedral models are generated from range data by clustering points according to their normal directions plotted on a Gaussian sphere. This mechanism helps overcome the sparsity of the point distribution. The assumption that the scene is made of planar regions is exploited to detect plane intersections and corners to compute plausible segmentations of building structures made of polyhedrons of low complexity. The method presented in this chapter is motivated on a graph-based image segmentation algorithm that grows regions according to local and global region similarity in linear time [18]. Our similarity measures rely on closeness of points and normal curvature. Moreover, neighbor candidates for region growing are searched for with an Approximate Nearest Neighbor (ANN) technique [6] that runs in logarithmic time.

The segmentation algorithm we present is capable of segmenting maps with over 8 million points and with accuracies that range from 5 to 20 cm, and is very flexible with only three parameters to tune: a nearest-neighbor bound, and thresholds for maximum distance between points and maximum curvature for a region.

This chapter is structured as follows. First in Section 2.1 or proposed alternative is described in detail and compared with a state of the art approach that uses Expectation Maxi-

mization (EM) to fit a probabilistic model of flat surfaces of the range map [36]. The comparison takes into account both quality of results and execution time. Finally, experiments on simulated and real data are presented, followed by some concluding remarks in Sections 2.2 and 2.3, respectively.

2.1 Graph-based 3D segmentation

Our method builds upon Felzenszwalb's algorithm for 2D image segmentation [18], and extends it to deal with non-uniformly sampled 3D range data. The algorithm proceeds as follows. First, the entire data set is preprocessed to compute local normal orientations of fitted planar patches for each point with respect to its k-Nearest Neighbors (kNNs). Then, the distances between nearest neighbors are computed. These distances are then sorted in increasing order and the resulting list is processed to create a forest of trees by merging neighboring points according to point distances and to the angle between their normals. These two measures, the distance between neighboring points and the angle between their normals, account for local segment variation. Global segment variation is also considered by computing the angle between a point normal and the aggregated normal for the current segment, i.e., the current tree in the forest. Local and global variation are both taken into account during tree merging hypotheses.

2.1.1 Fitting normals to local planar patches

Consider each 3D point in the dataset with coordinates $\mathbf{p} = (x, y, z)^\top$. The error between a fitted planar patch and the range map values for the kNNs to $\mathbf{p}$ is given by

$$\varepsilon = \sum_{i \in K} (\mathbf{p}_i^\top \mathbf{n} - d)^2 , \tag{2.1}$$

where $\mathbf{n} = (n_x, n_y, n_z)^\top$ is the local surface normal at $\mathbf{p}$, K is the set of kNNs to $\mathbf{p}$, and d the distance from $\mathbf{p}$ to the plane. This error can be re-expressed in the following form

$$\varepsilon = \mathbf{n}^\top \underbrace{\left(\sum_{i \in K} \mathbf{p}_i \mathbf{p}_i^\top \right)}_{\mathbf{Q}} \mathbf{n} - 2d \underbrace{\left(\sum_{i \in K} \mathbf{p}_i^\top \right)}_{\mathbf{q}} \mathbf{n} + |K| d^2 .$$

Combining the above error metric with the orthonormality property for each local surface normal into a Lagrangian of the form

$$l\left(\mathbf{n}^{\top}, d, \lambda\right) = \varepsilon + \lambda(1 - \mathbf{n}^{\top}\mathbf{n}),$$

the local surface normal that best fits the patch K is the one that minimizes the above expression [3]. Deriving l with respect to $\mathbf{n}$ and d, and setting the derivatives to zero, it turns out that the solution is the eigenvector associated to the smallest eigenvalue of

$$\left(\mathbf{Q} - \frac{\mathbf{q}\,\mathbf{q}^{T}}{|K|^{2}}\right)\mathbf{n} = \lambda\mathbf{n}.$$

2.1.2 Segmentation criteria

In the segmentation algorithm, local surface normals $\mathbf{n}$ are computed for each point in the point cloud, fitting local planar patches. To account for global variation, planar patches are merged, growing a forest of trees based on curvature and mean distance. The curvature between two candidate regions is computed from the angle between their two normals, which must be below a user-selected threshold t_c,

$$|\arccos(\mathbf{n}_i^{T}\mathbf{n}_j)| < t_c \tag{2.2}$$

Two segments passing the curvature criteria are merged if they also pass a distance constraint. That is, if the sum of distances between their centers along weighted orthogonal directions is below a user-selected threshold t_d,

$$\frac{k_i \mid (\mathbf{c}_i - \mathbf{c}_j)^{T}\mathbf{n}_j \mid + k_j \mid (\mathbf{c}_j - \mathbf{c}_i)^{T}\mathbf{n}_i \mid}{k_i + k_j} < t_d \tag{2.3}$$

with k_i and k_j the number of points each segment holds and $\mathbf{c}_i$ and $\mathbf{c}_j$ the segment centers. See Figure 2.2.

2.1.3 Implementation details

The input parameters to the segmentation algorithm are $|K|$ the number of local neighbors to consider for the fitting of planar patches, a distance threshold t_d, and a curvature threshold t_c. Each planar patch is stored in a tree structure. The tree contains in each node a 3D point belonging to the segment. The parent node contains also the surface normal. The entire scene is thus represented as a forest of disjoint trees. At each iteration over the list of ordered distances, the merging of neighboring planar patches is hypothesized. If the local and global variation criteria are satisfied, both in terms of neighboring distance and curvature, the segments are

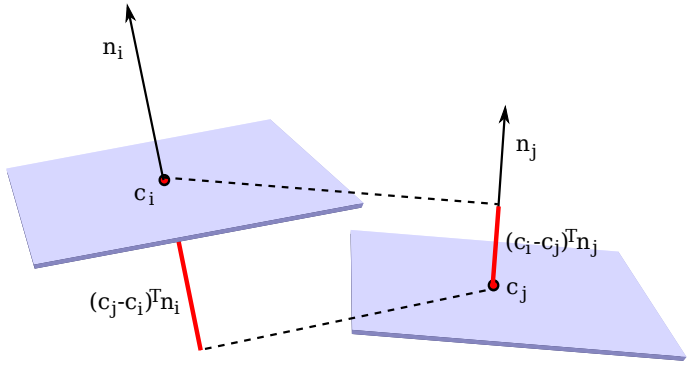


Figure 2.2: Projection of the region centers $\mathbf{c}_i$ and $\mathbf{c}_j$ onto neighboring planar patches.

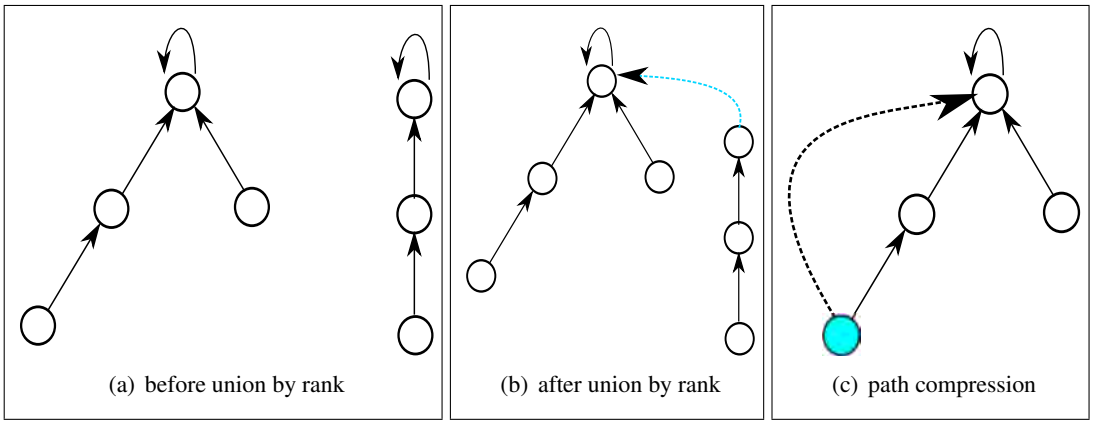


Figure 2.3: Operations used to maintain the height of trees minimal during the merge of planar patches.

joined using union by rank and path compression. Union by rank means choosing as tree root the one with larger cardinality when merging two trees, thus minimizing the depth of the tree. Path compression makes all nodes on a tree point to its parent, thus effectively reducing the tree depth to 1 [13]. See Figure 2.3.

2.1.4 Computational complexity analysis

The ANN library we use to search for approximate nearest neighbors has expected computational complexity $O(\log n)$ [6], and worst case complexity $O(n)$. Moreover, the complexity of union by rank and path compression is worst case $O(\alpha(n))$, where $\alpha(n)$ is the very slowly growing inverse of Ackermann's function [13], which for any conceivable application

is $\alpha(n) < 4$. Therefore, our region growing algorithm takes linear time in the number of points in the dataset, and the bottleneck of the algorithm is the nearest neighbor search. The overall expected computational complexity of our range data segmentation algorithm is $O(n \log n)$, with worst case computational complexity $O(n^2)$ for ill posed distributions of the 3D points. This complexity is in contrast to the much more expensive iterative algorithms that relay on EM.

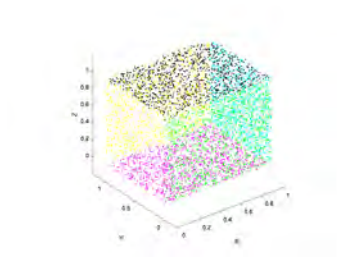
2.2 Comparison with EM

The proposed algorithm was tested using synthetic and real data. In the first experiment we built a synthetic model of an open 3D box consisting of five equally sized faces with varying noise parameters and also with various levels of outliers to account for lack of structure in the scene. For each plane, N 2D points are drawn from a uniform distribution in 3D. Then, each point is corrupted with zero mean Gaussian noise with independent variance σ^2 on each axis. Finally, a small percentage of these points is further normally corrupted with three times variance σ^2 to simulate the presence of outliers. We used the following values in the simulation: $\sigma^2 = \{0.0001, 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5\}$, percentage of outliers equal to $5\%, 10\%$, and $N = 1000$ points per plane, i.e., 5000 points per open cube.

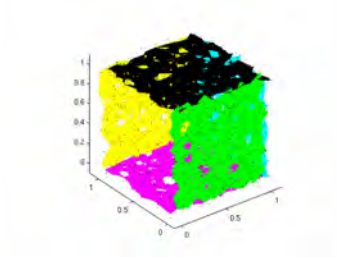
The proposed algorithm was compared with Liu's EM algorithm [36]. In our implementation of Liu's method, the following parameters were used: $J = 5$ planes; points are considered outliers when $x_{\texttt{max}} > 2$; and the density of each plane is smaller than 70% of the simulated points. The terminating condition in the standard EM algorithm is reached when $J = 5$ planes are found and the E and M steps have iterated over 25 cycles. Figure 2.4 shows comparison of Liu's method to ours for cubes generated with 5% and 10% of outliers and noise parameters $\sigma^2 = 0.0001$ and $\sigma^2 = 0.001$.

Figure 2.5 shows the mean square reprojection error for each plane ε/N, computed from Equation 2.1, and averaged for all planes in the open cube. For the selected operating parameters, both methods have comparable segmentation results.

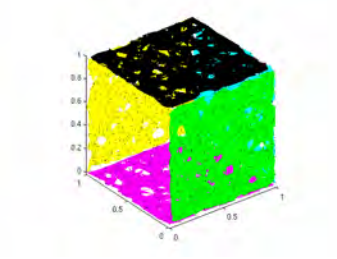
The clear advantage of the proposed algorithm is its computational cost. To compare algorithm speed, the open cube is sampled with $N = \{100, 500, 1000, 5000\}$, a fixed 1% amount of outliers, variance $\sigma^2 = 0.01$, and maximum iteration to 25 cycles for the EM algorithm. Figure 2.6 reports execution times for both the EM-based and our graph-based segmentation



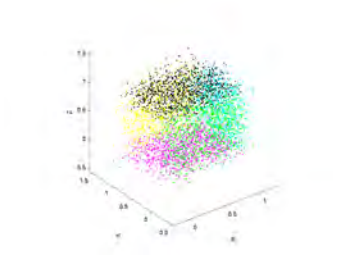
(a) Synthetic data with $\sigma^2 = 0.0001$ and 5% of outliers.



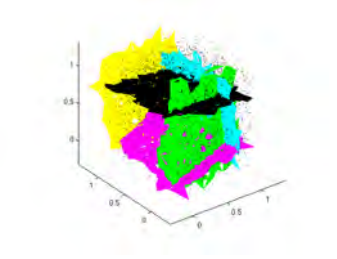
(b) EM-based segmentation results.



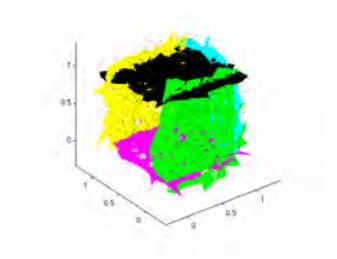
(c) Graph-based segmentation results.



(d) Synthetic data with $\sigma^2 = 0.001$ and 10% of outliers.



(e) EM-based segmentation results.

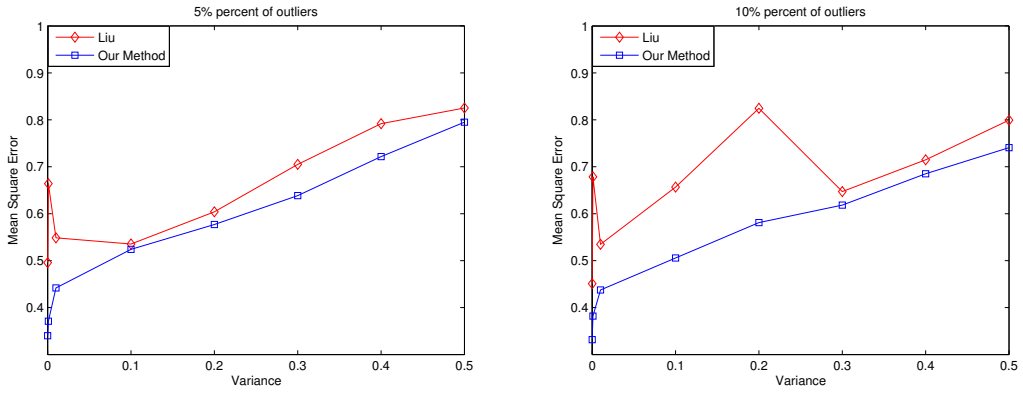


(f) Graph-based segmentation results.

Figure 2.4: Synthetically generated data for an open cube with five faces. Expectation-maximization-based segmentation is computed with our implementation of the method reported in [36]. The last column shows segmentation results over the same data with the proposed graph-based segmentation algorithm.

approaches. In spite that the expected computational complexity of our algorithm is $O(n \log n)$, its constant factor is significantly smaller than that of the EM-method. At 5000 data points, our method takes only about 3 seconds, whereas the EM-based approach is over 150 times slower, taking more than 7 minutes to compute the segmentation, in our implementation. All reported times are for experiments performed in a Pentium 4 PC with 2GB RAM running Matlab under Linux.

The method is applied to our real data set of the Barcelona Robot Lab, acquired during an outdoor 3D laser-based SLAM session [90]. The set contains over 8 million points and maps the environment with accuracies that vary from 5 cm to 20 cm approximately (see Figure 2.7). The input parameters for our segmentation algorithm applied to this set are $K = 30$ nearest neighbors, $t_d = 0.5$ for distance threshold, and $t_c = 0.5$ for curvature threshold. Segmentation results are shown in Figure 2.8. The proposed algorithm takes approximately 20 minutes to complete the plane segmentation. To show the applicability of the algorithm to robotics tasks,



(a) (b)

Figure 2.5: Mean square reprojection error with varying noise parameters and percentage of outliers for the two segmentation algorithms. a) Reprojection error with 5% of outliers. b) Reprojection error with 10% of outliers.

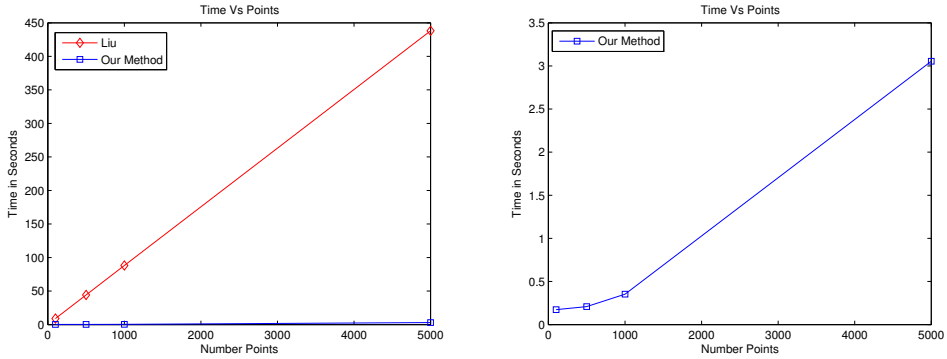
segments are labeled according to their normal orientation to indicate traversable regions versus walls and obstacles.

2.3 Remarks

The presented technique for range data segmentation has several advantages when compared to region merging EM-based algorithms. On the one hand, the computational cost of the presented approach is very appealing to handle large point clouds. Moreover, no a priori knowledge on the number of planes in the scene is needed.

One possible refinement to this segmentation method is to further build polygons from the set of points in each segmented plane with the aim of producing realistic virtual reality models. This however is not needed for our laser-to-image calibration purposes and is left out of the scope of the thesis.

In the following chapter we will show how this segmentation mechanism can aid the calibration of a non-overlapping large outdoor camera network.



(a) (b)

Figure 2.6: Time comparison between EM-based segmentation and the proposed graph-based approach. a) Execution time for both algorithms. b) Execution time for the proposed approach.

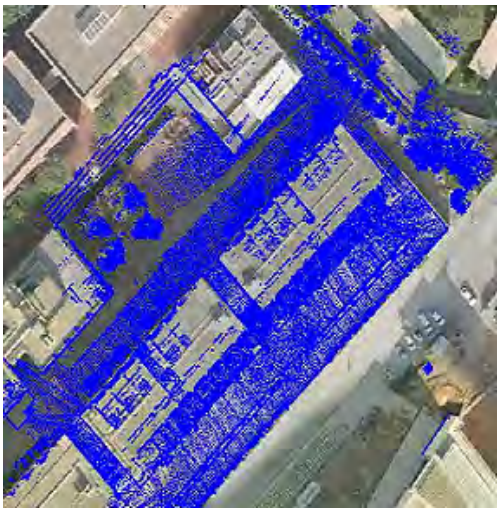
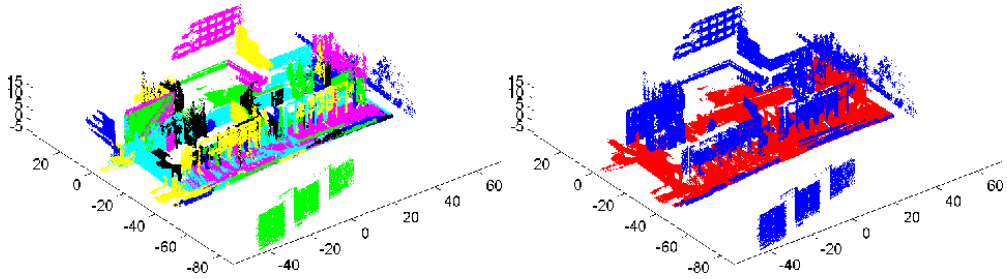


Figure 2.7: Aerial view of the Barcelona Robot Lab, and its 3D point map.



(a) (b)

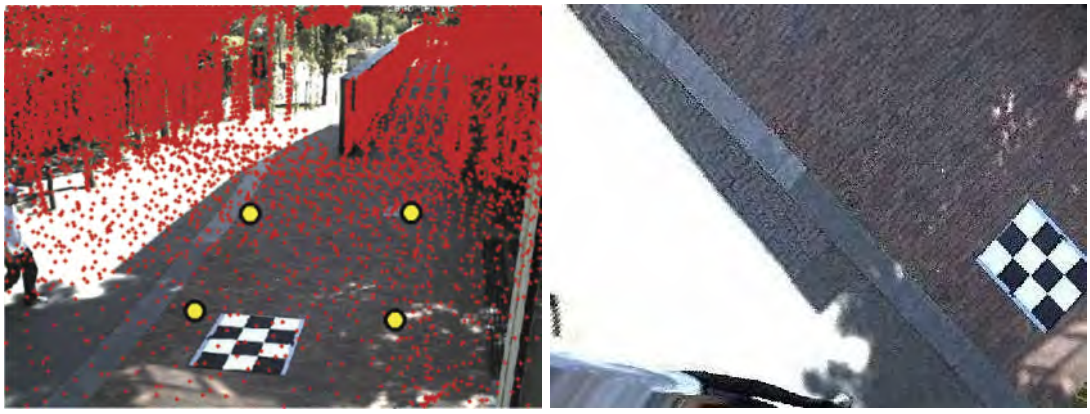
Figure 2.8: Barcelona Robot Lab. a) Planes extracted from the map of the Barcelona Robot Lab with the proposed graph-based segmentation approach. b) A possible application of the algorithm is to label segments according to traversability conditions. The segmentation results help differentiate horizontal planes for traversability (in red) from walls and obstacles (in blue).

Chapter 3

Camera network calibration using 3D information

The development of powerful laser sensors combined with simultaneous location and mapping (SLAM) methodologies [85] allows the possibility to have available high precision 3D maps registered over large areas [43]. These maps come in the form of large point clouds and are typically used to support robot navigation systems [90], and are usually acquired with laser range finders. In this chapter we present a methodology to calibrate an outdoor, non-overlapped, distributed camera network using such range data. See Figure 3.1. To that end, we acquired a 3D map covering the complete area of the network and, in particular, containing those areas corresponding to the fields of view of the cameras.

Traditional techniques for camera calibration require the use of non-planar [87] or planar [99] patterns, usually made of points, lines or checkerboards [14, 67], conics [11] or, even, augmented reality tags (ARTag) [19]. Unfortunately, for large outdoor camera networks, calibration patterns of reasonable sizes often project on images with very small resolution, mainly because cameras are usually located at a considerable height with respect to the floor, consequently making pattern segmentation difficult. In addition, a pattern-based independent calibration of each camera would require a secondary process to relate all camera coordinate systems to a global reference frame. But, establishing this relation with small to null overlapping fields of view is nearly impossible. For planar scenarios, a direct linear transformation (DLT) [30] suffices to estimate image to plane homographies [50]. Unfortunately, the planar scenario assumption is too restrictive, especially in situations with nonparallel, locally planar surfaces, such as ramps and plazas, which often occur in real urban environments, as in our case.



(a) (b)

Figure 3.1: Results of the proposed calibration system. (a) Plane selection in a graphical user interface and registration of the laser range data with a view from one of the cameras in the network; (b) recovered orthographic view of the ground plane. The chess pattern shown is not used for calibration; it serves just to visually evaluate the quality of the ground-plane rectifying homography.

An interesting technique to calibrate the camera network without the need of a pattern is with the aid of a bright moving spot [82]. The technique assumes overlapping fields of view to estimate the epipolar geometry to extract homographies, estimate depth and, finally, compute the overall calibration of the camera network. In our case, the cameras' fields of view seldom overlap, and the visibility of the bright spot does not always hold in sunlight. Another alternative is to place an LED light on a moving robot and to track it with a secondary robot equipped with a laser sensor, relating their position estimates to the camera network [97]. Yet another system that relies on tracking a moving object to estimate the extrinsic parameters is [64], which assumes a constant velocity model for the target. Tracking a moving target each time the system needs recalibration might be prohibitive. The estimation of the camera location purely by analyzing cast shadows is also mathematically possible, but with very low position accuracy in practice [32], and if one is interested only in the topology of the network configuration and not in a metric calibration, multi-target tracking of people could also be an alternative [88]. In contrast to these approaches, we opt for a system that does not rely explicitly on a moving pattern or shadow to calibrate the network and that produces an accurate metric calibration.

For camera network systems that incorporate camera orientation control (pan and tilt) and

motorized zoom, it is possible to use such motion capabilities to first estimate the intrinsic parameters rotating and zooming, fitting parametric models to the optical flow, and then to estimate extrinsic parameters aligning landmarks to image features [8]. Unfortunately, in our case, the cameras are not active. Another option is to use stereo pairs instead of monocular cameras at each node in the network. In this way, local 3D reconstruction can be obtained directly within each node and registered globally using graph optimization techniques [42]. Overlapping fields of view are still necessary in that case. A third option to calibrate the camera network, albeit relative translation, is to use a vertical vanishing point and the knowledge of a line in a plane orthogonal to the vertical direction on each camera image [33]. A different, but related, problem is the relative positioning of one or more cameras with respect to a range sensor. To that end, calibration can be achieved using a checkerboard pattern, as in [25]. A related methodology to calibrate extrinsically an omnidirectional camera using point correspondences between a laser scan an the camera image is proposed in [70]. In contrast to our approach, the method assumes known intrinsic camera parameters. For a method to calibrate the laser intrinsic parameters instead, the reader is referred to [45].

We benefit from the availability of a dense point cloud acquired during a 3D laser-based SLAM session with our mapping mobile robot [90]. The set contains over eight million points and maps the environment with accuracies that vary from 5 cm to 20 cm, approximately. These data replace the need for a checkerboard pattern, a tracked beam, a robot or active capabilities of the cameras and are used as external information to calibrate the network.

Our work is largely related in spirit to the method described in [35], in which a set of images are registered into a urban point cloud. One major difference of the approach is on the assumptions made with respect to the characteristics of the scene during 3D feature extraction. In particular, the above-mentioned technique exploits the fact that buildings have strong parallel edges and that these cluster with similar orientation. On the contrary, we exploit the fact that in urban scenes, large planar regions also meet at long straight edges. In contrast with [75], in which edge parallelism is used to calibrate only the attitude and focal length of cameras for traffic monitoring, we use edge information to calibrate also the camera location.

The rest of the chapter is organized as follows. We explain first how nominal calibration is executed and then how this calibration is refined, first by extracting 3D features from the point clouds and optimizing over the reprojection error of their matching to those found in images.

When needed, a final refinement step is computed by means of the DLT-lines algorithm. Experiments on a pair of scenarios are presented to show the feasibility of the proposed solution. The chapter is concluded with some remarks and possible enhancements of the method.

3.1 Nominal calibration

The proposed calibration procedure is illustrated in Figure 3.2. It consists of two main steps. In the first step, the internal camera parameters are initialized to the manufacturer specifications (image pixel width and depth and focal length), and a nominal calibration of the camera external parameters is obtained by manually registering the point cloud to an aerial image of the experimental site with the aid of a graphical user interface, prompting the user to coarsely specify the camera location, orientation, height and field of view. These initial parameters allow the cropping of the full point cloud into smaller regions of interest compatible with the field of view of each camera. The user can then adjust the registration by manually modifying each of the parameters (see the video associated with [54], available in the IEEE Xplore digital library).

In the second step, an automatic refinement of the camera calibration parameters is obtained by matching 2D image lines to their corresponding 3D edges in the point cloud. To this end, the point cloud is segmented into a set of best fitting planes with large support using local variation as discussed in the previous chapter, and also in [55], and straight edges are computed from the intersection of perpendicular planes in the set. The extracted 3D lines are associated with 2D image lines, and this information is fed to a non-linear optimization procedure that improves both the intrinsic and the extrinsic parameters. Finally, the homographies of the walking areas are computed for the planar regions in the range data. The final output of the whole calibration procedure consists in: (1) the extrinsic camera parameters, *i.e.*, the position and orientation of each camera in the world frame; (2) the intrinsic camera parameters (focal distance, image center aspect ratio and distortion terms); and (3) the homographies of each walking area.

The first step of the calibration procedure needs to be performed only once, during the camera network installation or when the network topology changes, *i.e.*, cameras are added/moved, and takes only a couple of minutes. The second step, which does not require user intervention, can be executed as frequently as needed to keep the system calibrated, despite small modifications in camera orientation due to weather conditions and maintenance operations. In the

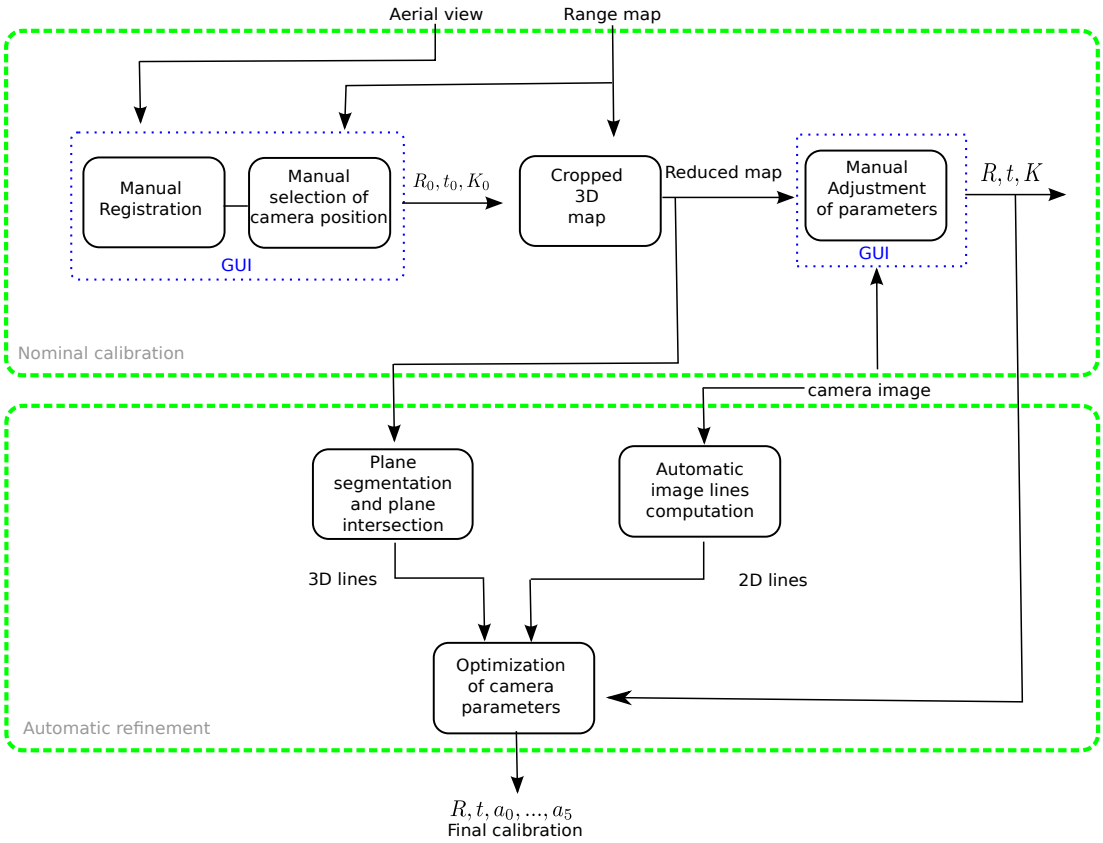
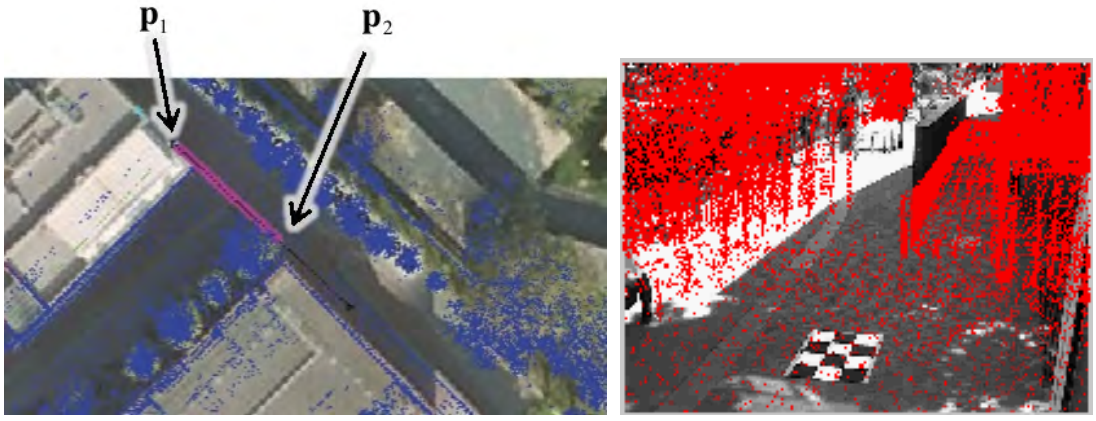


Figure 3.2: Two-step calibration methodology. In the first step, a graphical user interface is used to assist in an initial manual registration of the point cloud. The second step refines this registration, matching 2D image lines with 3D edges in the point cloud.

following paragraphs, we detail the nominal calibration.

For the nominal calibration, the user interacts with a graphical user interface to coarsely locate each camera with respect to the reference frame of the point cloud and the viewing direction in the ground plane. For an easier interaction, the point cloud appears registered with an aerial view of the environment. The registration of the point cloud with the aerial view is computed manually using the DLT [30]. The camera parameters are initialized with default intrinsic parameters.

We then compute initial values for the camera pose: the center of projection $\mathbf{t}$ is simply the user selected point $\mathbf{p}_1$, located at a user defined height (*i.e.*, 6 m in our application). See Figure 3.3. The two selected points $\mathbf{p}_1$ and $\mathbf{p}_2$ set the azimuth direction ψ. The elevation ϕ



(a) (b)

Figure 3.3: Graphical user interface. (a) The point cloud is shown to the user overlaid on top of an aerial view of the environment. The user is prompted to select (1) a coarse camera location $\mathbf{p}_1$; and (2) the viewing direction $\mathbf{p}_2$ indicated by the magenta line; (b) During the initialization process, the user can manually adjust intrinsic and extrinsic parameters on a projected view of the point cloud.

gives a user-defined inclination to the ground (17° in the shown example), and the roll ρ is set to π to account for the proper axes changes. These parameters suffice to compute the initial rotation matrix $\mathbf{R} = \mathbf{R}_\rho \mathbf{R}_\psi \mathbf{R}_\phi$ with:

$$\mathbf{R}_\psi = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R}_\phi = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix}, \text{and } \mathbf{R}_\rho = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

Assuming that the principal point (u_0, v_0) is located at the image center, we can compute an initial estimate for the camera intrinsic parameters using as input the focal length f and the CCD pixel size in millimeters k_u and k_v, *i.e.*, $\alpha_v = fk_v$, $\alpha_u = fk_u$, and:

$$\mathbf{K} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

The initial estimate of the perspective projection matrix for each camera is:

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \tag{3.2}$$

Once this initial estimate is obtained for a particular camera, the user can further adjust each parameter manually, whilst a projection of a cropped section of the point cloud that falls

within the viewing frustum is visualized in the image. Note, however, that this initial estimate does not take into account radial distortion parameters. These are refined along with the rest of parameters in the second step of the method.

3.2 Calibration refinement

To improve camera calibration from the nominal parameters, we propose an automatic method, where relevant 3D edges are extracted from the point cloud and matched to corresponding image lines. In practice, the method works well with about a half-dozen lines selected from each camera image.

The procedure uses the nominal calibration as an initial rough approximation and can be executed anytime to recover from small perturbations in camera orientation or any other parameter, due to weather (wind, rain, *etc*.) or maintenance operations (repair, cleaning).

3.2.1 3D edge computation

The computation of straight lines from the point cloud relies on identifying and intersecting planes. The method to segment planar regions was described in the previous chapter.

Once a set of segments is obtained, their intersecting edges are computed, and the ones with sufficient support from their generating planes and with good orthogonality conditions are selected for projection onto the images. The two steps, plane segmentation and edge extraction are summarized in Algorithm 1.

Algorithm 1 The algorithm to find edge lines at orthogonal plane intersections within the point cloud.

EDGEEXTRACTION(M)
INPUT:
M: Point cloud.
OUTPUT:
E: 3D lines.

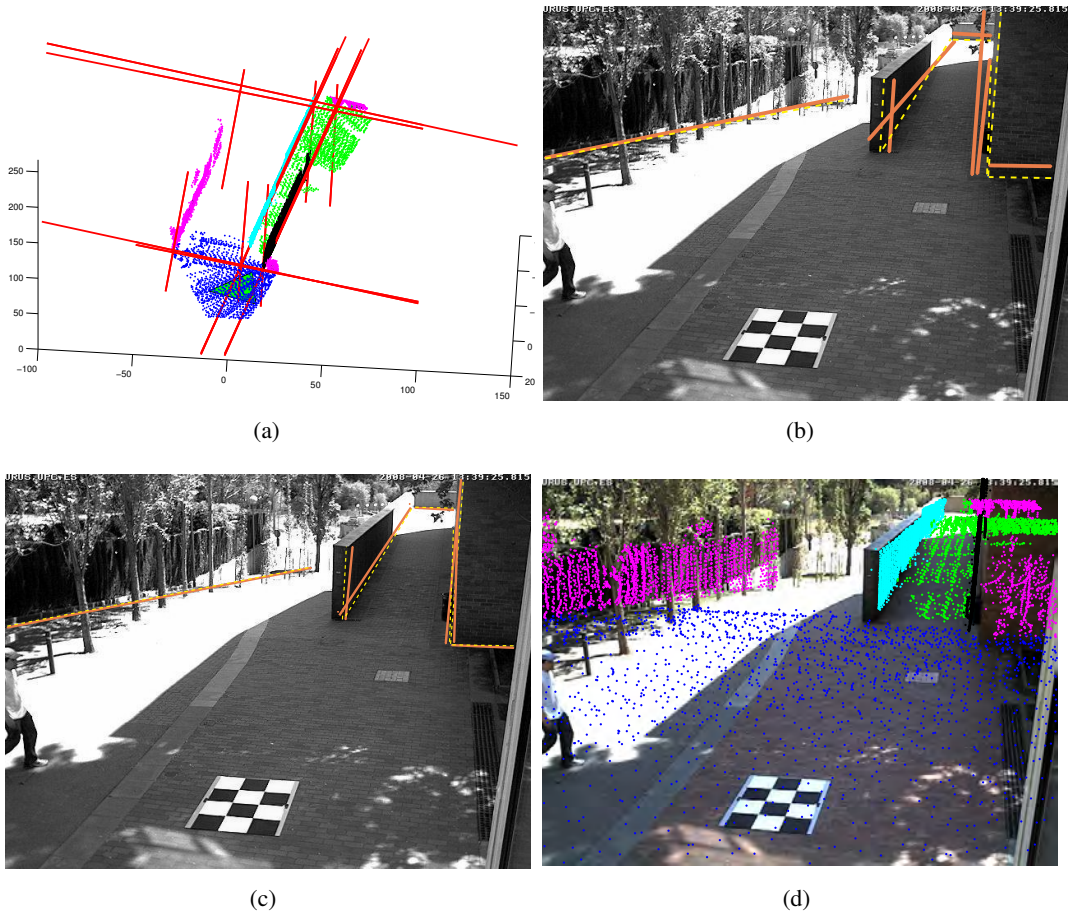
$D \leftarrow \emptyset$
for each $\mathbf{p}_i \in M$ **do**
  $N_i \leftarrow$ FINDNEIGHBORS($\mathbf{p}_i, M$)
  $\mathbf{n}_i \leftarrow$ COMPUTENORMAL($\mathbf{p}_i, N_i$)
  LABEL($\mathbf{p}_i$) $\leftarrow i$
  $D \leftarrow D \cup$ FINDDISTANCES($\mathbf{p}_i, N_i$)
end for
$D \leftarrow$ SORTDISTANCES(D)
for each $d_k \in D$ **do**
  $\mathbf{c}_i \leftarrow$ START(d_k)
  $\mathbf{c}_j \leftarrow$ END(d_k)
  if LABEL($\mathbf{c}_i$) $\neq$ LABEL($\mathbf{c}_j$) **then**
    if $|\cos^{-1}(\mathbf{n}_i^T \mathbf{n}_j)| < t_c$ **then**
      if $\frac{k_i|(\mathbf{c}_i-\mathbf{c}_j)^T\mathbf{n}_j|+k_j|(\mathbf{c}_j-\mathbf{c}_i)^T\mathbf{n}_i|}{k_i+k_j} < t_d$ **then**
        MERGETREES($\mathbf{c}_i, \mathbf{c}_j$)
      end if
    end if
  end if
end for
$E \leftarrow \emptyset$
$L \leftarrow$ LABELS(M)
for each pair of segments $(\mathbf{s}_i, \mathbf{s}_j) \in L$ with respective $(\mathbf{n}_i, \mathbf{n}_j)$ **do**
  if ORTHOGONAL($\mathbf{n}_i, \mathbf{n}_j$) **then**
    $E \leftarrow E \cup$ PLANEINTERSECTION($\mathbf{n}_i, \mathbf{n}_j$)
  end if
end for

3.2.2 Optimization

Straight image lines are extracted from the camera images using the method proposed in [94]. The line set is pruned to those lines larger than a predefined threshold.

3D model lines are projected onto the image plane using the projection matrix computed during the nominal calibration step. 2D-3D line association is attained by matching such projections to the closest 2D image line.

Once the 3D-2D association is established, nonlinear optimization is performed to improve



(a) (b)

(c) (d)

Figure 3.4: Optimization. (a) computation of plane intersections in the range data; (b) projection of lines onto the image plane using the nominal calibration parameters; (c) line matching optimized in the image plane; (d) segmented point cloud reprojected onto the calibrated image.

the nominal calibration by minimizing the squared sum of the line endpoint projection errors:

$$\underset{\vartheta}{\text{minimize}} \sum_i \|\mathbf{u}_i - \mathbf{u}_{id}(\vartheta)\|^2 \tag{3.3}$$

where $\mathbf{u}_{id}(\vartheta)$ is the distorted projection of the 3D endpoint $\mathbf{p}_i$, $\vartheta = (\mathbf{K}, \mathbf{R}, \mathbf{t}, a_1, a_2)$ are the set of parameters being optimized and $\mathbf{u}_i$ is the image point. The optimization is solved using Levenberg-Marquardt nonlinear optimization. See Figure 3.4.

In this step of the method, image distortion is modeled based on even powers of the radial distance in the image plane:

$$\mathbf{u}_{id} = \mathbf{u}_{in} + (1 + \sum_{j=1}^{2} a_j r^{2j})(\mathbf{u}_{in} - \mathbf{u}_0) \tag{3.4}$$

where a_j are the distortion parameters, $r^2 = \|\mathbf{u}_{in} - \mathbf{u}_0\|^2$ is the radial distortion factorization, $\mathbf{u}_0$ is the computed principal point, $\mathbf{u}_{in}$ is the normalized (pinhole) image projection of point $\mathbf{p}_i$:

$$\begin{bmatrix} \mathbf{u}_{in} \\ 1 \end{bmatrix} \sim \mathbf{K}[\mathbf{R}|\mathbf{t}] \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} \tag{3.5}$$

and $\sim$ denotes equality up to a scale factor.

3.2.2.1 Initialization

The nominal calibration introduced in Section 3.1 is, in general, sufficient to initialize the calibration optimization formalized in Equation 4.3. However, one finds that while it is usually simple and very effective to observe precisely some parameters as the camera horizontal position in the coordinates of a laser range finder map; other parameters, such as camera height, 3D rotation, focal length or principal point, are more challenging. The possibility of automating the initialization of the optimization procedure is a convenient feature for calibrating cameras in a network.

The initialization of the calibration optimization process may be setup to find from none to all of the calibration parameters. 'None' means using all of the nominal calibration results to reinitialize the optimization. Whereas 'all' means refining all parameters from image and laser range finder data. In between there are several cases of interest, many of which have solutions published. For example, if a camera has its intrinsic parameters calibrated before being mounted in place, then one just has to estimate the extrinsic parameters by solving the well-known Perspective-n-Point (PnP) problem [20]. In the following, we detail two cases. In the first case, we show how to estimate all of the parameters from 3D lines imaged by the camera to calibrate. In the second case, we consider that the camera position is precisely known and detail how to estimate the intrinsic parameters and the camera orientation only.

As proposed in [74], the use of image lines instead of isolated points in the camera calibration process brings an advantage. Image processing can be used to fine tune the location of the lines in the image and therefore automatically improve the calibration data input. In this section, *DLT-Lines* is presented as a method to initialize the optimization step, allowing one to estimate simultaneously the camera projection matrix and radial distortion, from the 3D point cloud and 2D lines.

Considering the shorthand notation for image points $\mathbf{m}_i = [\mathbf{u}_i^\top \ 1]^\top$ and 3D points $\mathbf{M}_i = [\mathbf{p}_i^\top \ 1]^\top$ the perspective camera model, Equation 3.5, becomes $\mathbf{m}_i \sim \mathbf{P}\mathbf{M}_i$.

The projection of a 3D line $\mathbf{L}_i$ to the camera image plane can be represented by the cross product of two image points in projective coordinates:

$$\mathbf{l}_i = \mathbf{m}_{1i} \times \mathbf{m}_{2i} \tag{3.6}$$

Any point $\mathbf{m}_{ki}$ lying in the image line $\mathbf{l}_i$ implies that $\mathbf{l}_i^\top \mathbf{m}_{ki} = 0$. Hence, applying the multiplication of $\mathbf{l}_i^\top$ to both sides of the perspective camera model, *i.e.*, $\mathbf{l}_i^\top \mathbf{m}_{ki} = \mathbf{l}_i^\top \mathbf{P}\,\mathbf{M}_{ki}$, leads to:

$$\mathbf{l}_i^\top \mathbf{P}\,\mathbf{M}_{ki} = 0 \tag{3.7}$$

where $\mathbf{M}_{ki}$ is a 3D point in projective coordinates lying in $\mathbf{L}_i$. The properties of the Kronecker product [39] allow one to obtain a form factorizing the vectorized projection matrix:

$$(\mathbf{M}_{ki}^\top \otimes \mathbf{l}_i^\top)\,\mathcal{P} = 0\,, \tag{3.8}$$

where the notation $\mathcal{P} = \mathrm{vec}(\mathbf{P})$ is used to indicate that the elements of the matrix $\mathbf{P}$ are stacked columnwise in vector form.

Considering $N \geq 12$ pairs $(\mathbf{M}_{ki}, \mathbf{l}_i)$, one forms a matrix $\mathbf{B}$, $N \times 12$, by stacking the N matrices $\mathbf{M}_{ki}^\top \otimes \mathbf{l}_i^\top$. An example of $N = 12$ arises when one observes six 2D lines imaging six 3D lines, $\mathbf{L}_i$ $(i = 1,\ ...,\ 6)$, each one represented by two end points, $\mathbf{L}_i \leftrightarrow (\mathbf{M}_{i1}, \mathbf{M}_{i2})$. Alternatively, given a 3D line $\mathbf{L}_i$ and its projection represented by the image line $\mathbf{l}_i$, any 3D point lying on the 3D line $\mathbf{L}_i$ can be paired with the 2D line $\mathbf{l}_i$. On the other hand, any image line $\mathbf{l}_i$ can be paired with any 3D point lying on $\mathbf{L}_i$, *i.e.*, more than one image line can be paired with a 3D point.

The least squares solution, more precisely the minimizer of $\|\mathbf{B}\,\mathcal{P}\|^2$ subjected to $\|\mathcal{P}\| = 1$, is the right singular vector corresponding to the least singular value of $\mathbf{B}$.

Note that the perspective camera model, as presented in Equation 3.5, does not contain yet the radial distortion. To include radial distortion, we use Fitzgibbon's division model [21]. An undistorted image point, $\mathbf{u} = [u\ v]^\top$, is computed from a radially distorted image point $\mathbf{u}_d = [u_d\ v_d]^\top$ as $\mathbf{u} = \mathbf{u}_d/(1 + \lambda\,\|\mathbf{u}_d\|^2)$, where λ represents the radial distortion parameter. The division model allows one to define a line $\mathbf{l}_{12}$ as the cross product of two points:

$$\mathbf{l}_{12} = \begin{bmatrix} u_{1d} \\ v_{1d} \\ 1 + \lambda s_1^2 \end{bmatrix} \times \begin{bmatrix} u_{2d} \\ v_{2d} \\ 1 + \lambda s_2^2 \end{bmatrix} = \hat{\mathbf{l}}_{12} + \lambda \mathbf{e}_{12} \tag{3.9}$$

where s_i is the norm of distorted image point i, $s_i^2 = u_{id}^2 + v_{id}^2$, the distorted image line is denoted as $\hat{\mathbf{l}}_{12} = [u_{1d}\ v_{1d}\ 1]^\top \times [u_{2d}\ v_{2d}\ 1]^\top$ and the distortion correction term $e_{12} = [v_{1d}s_2^2 - v_{2d}s_1^2,\ u_{2d}s_1^2 - u_{1d}s_2^2,\ 0]^\top$.

Applying Equation 3.9 on Equation 3.8 leads to the following equation:

$$\left(\mathbf{M}_{k12}^\top \otimes (\hat{\mathbf{l}}_{12} + \lambda \mathbf{e}_{12})^\top\right) \mathcal{P} = 0 \tag{3.10}$$

which can be rewritten as:

$$(\mathbf{B}_{ki1} + \lambda \mathbf{B}_{ki2})\ \mathcal{P} = 0 \tag{3.11}$$

where $\mathbf{B}_{ki1} = \mathbf{M}_{k12}^\top \otimes \hat{\mathbf{l}}_{12}^\top$, $\mathbf{B}_{ki2} = \mathbf{M}_{k12}^\top \otimes \mathbf{e}_{12}^\top$ and $\mathbf{M}_{k12}$ denotes the $k-th$ 3D point projecting to the distorted line $\hat{\mathbf{l}}_{12}$.

To solve Equation 3.11 instead of Equation 3.8, we still need to consider $N \geq 12$ pairs $(\mathbf{M}_{ki}, \hat{\mathbf{l}}_i)$, where $N = k_{max} i_{max}$, and form now two $N \times 12$ matrices, $\mathbf{B}_1$ and $\mathbf{B}_2$, instead of just $\mathbf{N}$, by stacking matrices $\mathbf{B}_{ki1}$ and $\mathbf{B}_{ki2}$. Left-multiplying the stacked matrices by $\mathbf{B}_1^\top$ results in a polynomial eigenvalue problem (PEP), which can be solved, for example, in MATLAB using the `polyeig` function. It gives, simultaneously, the projection matrix, in the form of $\mathcal{P}$, and the radial distortion parameter λ.

Having estimated the projection matrix, $\mathbf{P}$, the camera intrinsic and extrinsic parameters can be obtained by QR-decomposition [30]. More precisely, given the sub-matrix $\mathbf{P}_{3\times3}$ containing the first three columns of $\mathbf{P}$ and $\mathbf{S}$ an anti-diagonal matrix:

$$\mathbf{S} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{3.12}$$

the QR-decomposition allows factorizing

$$\mathbf{P}_{3\times3}{}^\top \mathbf{S} = \mathbf{Q}\mathbf{U}\,, \tag{3.13}$$

where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{U}$ is an upper triangular matrix. Then, the intrinsic parameters and the rotation matrices are computed as $\mathbf{K} = -\mathbf{S}\mathbf{U}^\top\mathbf{S}$ and $\mathbf{R} = \mathbf{Q}^\top\mathbf{S}$. Finally, the camera position is obtained with $\mathbf{t} = \mathbf{K}\mathbf{P}_4$, where $\mathbf{P}_4$ is a 3×1 vector containing the fourth column of $\mathbf{P}$. If the diagonal of $\mathbf{K}$ contains negative values, then it is corrected by post-multiplying by a diagonal matrix. In MATLAB/Octave `D= diag(sign(diag(K))); K= K*D; R= D*R; t= D*t;`. In addition, since $\pm\mathbf{P}$ are

both solutions of Equation 3.8, the factorization of $\mathbf{P}$ may imply $det(\mathbf{R}) = -1$. If $det(\mathbf{R}) = -1$, then the factorization of $\mathbf{P}$ is repeated using $-\mathbf{P}$.

To convert the obtained distortion parameter λ to the distortion parameters in Equation 3.4, we sample the region of interest and find a least squares solution for the best parameter fit in this region. Starting from a set of camera points evenly sampled at pixel granularity covering the image dimensions $\{\mathbf{u}_{id}\}$, we apply the Fitzgibbon distortion model to obtain an uneven set of undistorted pixel coordinates $\{\mathbf{u}_i\}$. We next solve the optimization problem:

$$\underset{a_j}{\text{minimize}} \sum_i \left\| \mathbf{u}_0 + \left(1 + \sum_{j=1}^{2} a_j r^{2j}\right)(\mathbf{u}_i - \mathbf{u}_0) - \mathbf{u}_{id} \right\|^2 \tag{3.14}$$

which is a linear least squares problem in the variables a_j, where a closed form solution is available. For small distortions, we empirically find that $a_1 = \lambda$ and $a_2 = 0$ provide a good fit to initialize the main optimization algorithm.

3.2.2.2 Known camera location

In the case one knows accurately the camera location, e.g., the camera has been imaged by the 3D data acquisition system, then the number of degrees of freedom of the calibration problem is decreased. The DLT methodology presented in the previous section can be further simplified.

Subtracting the camera center to all points of the point cloud results in a coordinate system where the camera is at the origin, and thus, the projection matrix, $\mathbf{P} = \mathbf{K}[\mathbf{R}|\ \mathbf{t}]$ is equivalent to a simple homography, $\hat{\mathbf{P}} = \mathbf{KR}$. Considering image lines $\mathbf{l}_i$ and 3D points, $\mathbf{p}_{ki} = [x_{ki}\ y_{ki}\ z_{ki}]^\top$, imaged as points of the lines, recalling Equation 3.7, one has:

$$\mathbf{l}_i^\top \mathbf{KR}(\mathbf{p}_{ki} - {}^W\mathbf{t}_C) = 0 \tag{3.15}$$

where ${}^W\mathbf{t}_C$ denotes the camera projection center in world coordinates. As such, one obtains linear constraints similar to the ones already derived for *DLT-Lines*:

$$\left((\mathbf{p}_{ki} - {}^W\mathbf{t}_C)^\top \otimes \mathbf{l}_i^\top\right) \mathcal{KR} = 0\,, \tag{3.16}$$

with $\mathcal{KR} = \text{vec}(\mathbf{KR})$

The length of $\mathcal{KR}$ is just nine, *i.e.*, the knowledge of the camera location saves three variables to estimate, and thus, the estimation process is intrinsically simplified. Finally, the projection matrix, $\mathbf{P}$, can be obtained decomposing $\hat{\mathbf{P}} = \mathbf{KR}$ and adding the camera location as $\mathbf{P} = [\hat{\mathbf{P}} \mid \hat{\mathbf{P}}\,{}^W\mathbf{t}_C]$.

The calibration problem has been reduced to the estimation of a homography, represented by $\mathcal{KR}$ and, therefore, not including radial distortion. A similar form based on Equation 3.16 can be obtained for the radial distortion case represented in Equation 3.10:

$$\left(\tilde{\mathbf{p}}_{k12}^{\top} \otimes (\hat{\mathbf{l}}_{12} + \lambda \mathbf{e}_{12})^{\top}\right) \mathcal{KR} = 0 \tag{3.17}$$

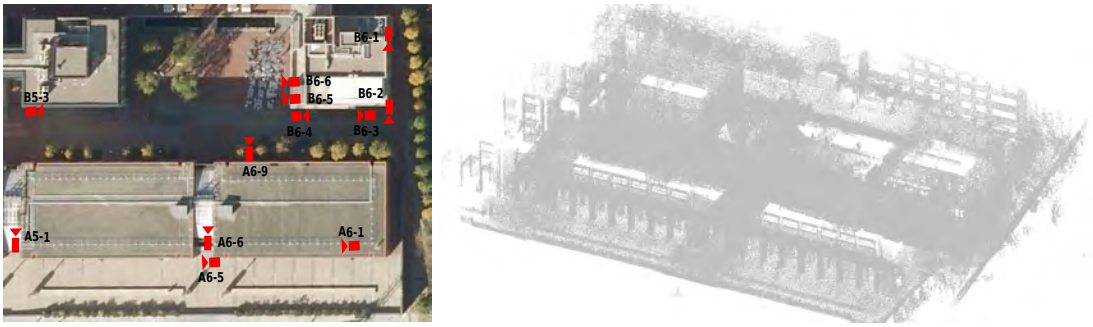
where $\tilde{\mathbf{p}}_{k12} = (\mathbf{p}_{k12} - {}^{W}\mathbf{t}_{C})$. Equation 3.17 can be re-written in the form of Equation 3.11, which can be used to estimate the camera projection matrix $\mathbf{P}$ and radial distortion parameter λ, as shown before.

3.3 Experiments

To demonstrate the performance of the proposed calibration method, we show calibration results of two different outdoor scenarios. In both cases, the range data was gathered using a custom-built 3D laser with a Hokuyo UTM-30LX scanner mounted in a slip-ring. Each scan has 194,580 points with a resolution of 0.5° azimuth and 0.25° elevation. The first dataset was acquired in the Barcelona Robot Lab (BRL). We only use 12 of the 21 cameras. They are shown in Figure 3.5. For this dataset, our 3D laser scanner was mounted on Helena, a Pioneer 3AT mobile robot, acquiring a total of 400 scans; however, only 30 of them were necessary to cover the area of the selected cameras. The complete dataset is available online [84].

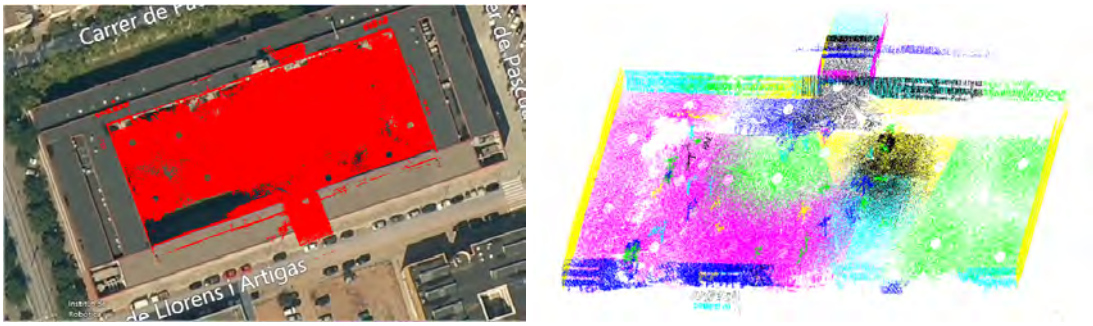
The second dataset was gathered in the inner courtyard of the Facultat de Matemàtiques i Estadística (FME), located at the Campus Sud of the UPC. For this dataset, the range sensor was mounted atop our robot Teo, a rough outdoor terrain Segway RPM400 mobile robot. In this case, only 39 scans were collected. Figure 3.6 shows the point cloud registered onto an aerial view of the scene and the segmented planes. A mobile phone camera was used to acquire geo-tagged images from different position in this scenario.

In both cases, the point clouds generated from the aggregation of multiple scans are pre-processed to remove outliers, to smooth out the planar regions and to provide a uniform point distribution through subsampling. The details of the filtering scheme used follow [90]. The parameters used to segment the range data in both scenarios were $n = 25$ neighbors to fit planar patches, a distance threshold of $t_d = 0.5$ and a curvature threshold $t_c = 0.8$. Furthermore, we only consider lines intersecting orthogonal planes with a deviation of $\pm 3^{\circ}$ from orthogonality. For those cases when there were less than six independent lines detected for the calibration of



(a) (b)

Figure 3.5: The Barcelona Robot Lab. (a) aerial view of the camera distribution; and (b) the point cloud.



(a) (b)

Figure 3.6: The Facultat de Matemàtiques i Estadística (FME) scenario. (a) The point cloud registered onto an aerial view of the scene; and (b) the segmented point cloud.

a camera, each detected line was broken into three segments, and all of these were used for the optimization step. This allowed us to have a sufficiently large number of lines to find a least squares solution to Equations 3.16 and 3.17. In most of those cases, however, the DLT-lines algorithm did not contribute to the improvement of the solution, and the first optimization sufficed to find acceptable results.

For the BRL dataset, the initial elevation angle was set to 17°. We initialized on this value, because most of the cameras are located about 6 m above the ground, and objects in the images for which lines can be detected reliably are closer than 20 m. The horizontal field of view is initialized to 40°, which corresponds to an 8-mm lens in a 0.25-in CCD.

The final calibration of the internal camera parameters for the BRL set are given in Ta-

Camera	Focal length (pixels)	Principal point (pixels)	Mean reprojection error (pixels)	SD (pixels)
A5-1	926.0, 926.0	499.2, 348.6	1.4	4.5
A6-1	857.8, 857.8	255.4, 338.6	1.6	5.3
A6-5	920.5, 920.5	314.0, 240.9	1.3	4.7
A6-6	842.3, 842.3	296.6, 145.4	0.7	4.2
A6-9	747.3, 747.3	366.6, 226.6	8.8	12.4
B5-3	801.2, 801.2	364.1, 202.1	5.3	6.1
B6-1	597.9, 597.9	388.8, 219.8	0.9	1.6
B6-2	817.9, 817.9	295.1, 262.1	2.6	4.4
B6-3	804.2, 804.2	374.1, 246.1	4.7	7.5
B6-4	824.1, 824.1	336.6, 237.2	3.6	6.5
B6-5	840.0, 840.0	317.9, 229.6	2.2	4.3
B6-6	862.5, 862.5	320.1, 247.1	4.6	8.5

Table 3.1: Estimated internal camera parameters of the BRL scenario.

Camera	Position (m)	Orientation (radians)	Ground truth position (m)	Elapsed time (s)
A5-1	−37.93, −25.37, 3.88	−0.63, 1.85, 2.03	−37.91, −28.2, 3.75	95.2
A6-1	42.84, −25.47, 4.06	1.57, −0.73, −1.37	42.66, −24.95, 3.99	86.5
A6-5	12.76, −28.10, 2.58	1.61, −0.90, −1.18	12.39, −28.95, 3.10	45.2
A6-6	9.83, −22.91, 3.07	0.70, −2.21, −2.10	9.44, −23.5, 2.95	79.6
A6-9	19.32, −4.02, 3.49	0.10, 1.85, 1.94	19.03, −6.65, 3.11	56.8
B5-3	−39.93, −1.16, 2.23	1.20, 1.25, 0.88	−38.94, −2.63, 1.95	69.5
B6-1	52.22, 19.14, 3.53	1.43, 0.60, −0.09	51.98, 18.55, 3.11	66.0
B6-2	51.66, 7.50, 3.66	1.52, 0.45, −0.09	51.55, 7.20, 3.45	107.3
B6-3	50.48, 2.53, 3.10	1.50, −0.60, −1.33	50.18, 4.36, 2.90	98.6
B6-4	34.45, 1.21, 3.13	1.15, 1.17, 0.42	32.85, 1.63, 2.90	76.8
B6-5	31.80, 6.34, 2.93	1.50, −0.94, −1.31	31.88, 7.48, 5.10	123.9
B6-6	31.81, 14.74, 5.82	1.50, −0.82, −1.32	31.64, 15.59, 5.10	103.2

Table 3.2: Estimated external camera parameters of the BRL scenario.

ble 3.1, along with the mean reprojection error and standard deviation. Note that a comparison of these estimates to those obtained with a checkerboard is unrealistic due to the actual positioning of the cameras. An unfeasibly large checkerboard would be needed and moved along the whole camera workspace to achieve significant results. Table 3.2 gives the obtained camera locations for this experiment, together with the camera ground truth locations manually extracted from the 3D point cloud. These values should only be used as a reference, since small variations of focal length might have incidence in the final positioning of the camera along the principal axis, without detriment to the camera reprojection for a limited range of depth values. The elapsed computation time for the calibration of each camera is also reported in the table. Figure 3.7 shows the reprojection of each matched line onto the corresponding camera images after the optimization is computed.

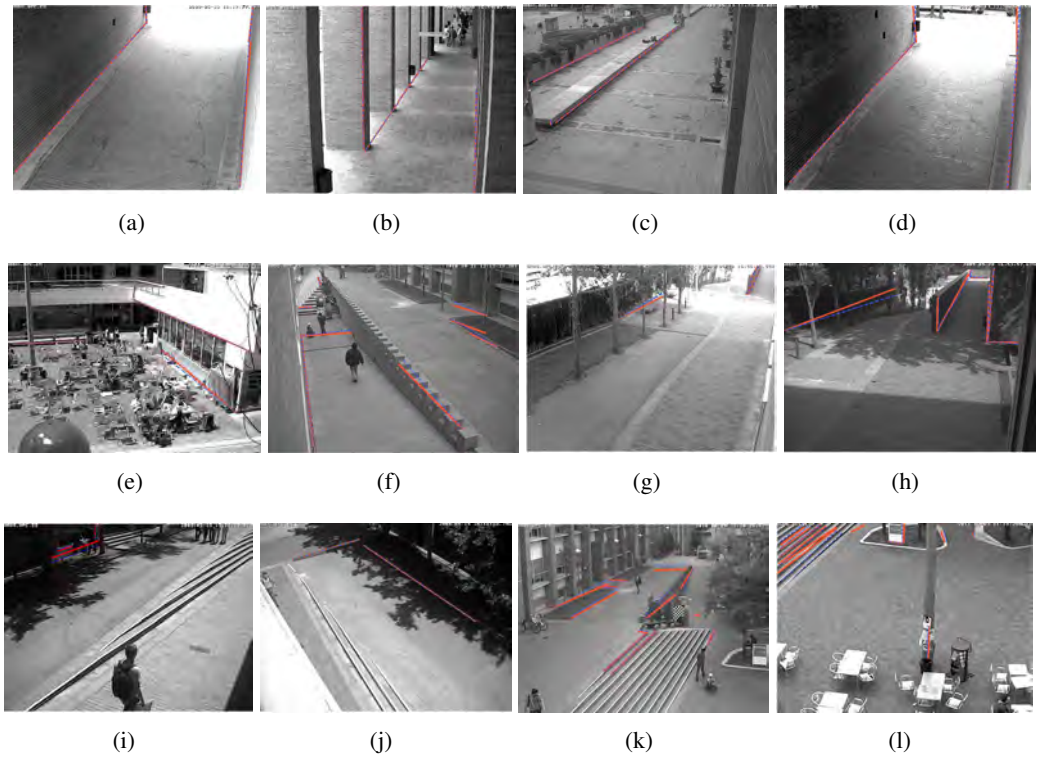


Figure 3.7: Results of the final calibration of the camera network for the BRL scenario. Optimization approximates the projected laser lines (blue) to the image lines (red). (a) A5-1; (b) A6-1; (c) A6-5; (d) A6-6; (e) A6-9; (f) B5-3; (g) B6-1; (h) B6-2; (i) B6-3; (j) B6-4; (k) B6-5; (l) B6-6.

The estimated camera poses are plotted in Figure 3.8. Camera viewpoints are represented with triangular pyramids that suggest the viewing direction. To empirically judge the quality of the calibration results, we can also project all points from the map that fall in each camera viewing frustum onto the image plane. This is shown in Figure 3.9 for the BRL.

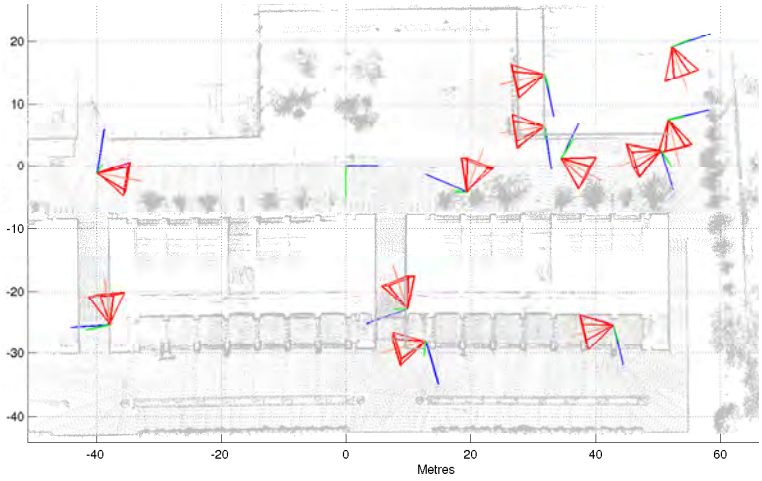


Figure 3.8: Calibrated camera locations for the Barcelona Robot Lab (BRL) dataset.

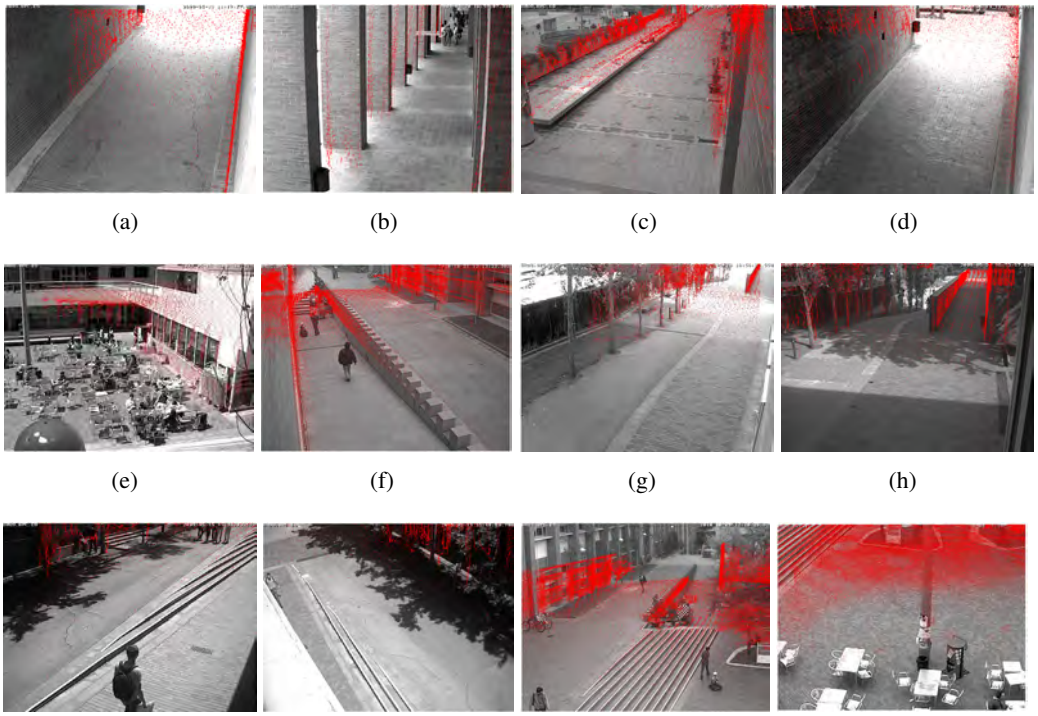


Figure 3.9: Visualization of range data on each of the camera views of the BRL scenario. (**a**) A5-1; (**b**) A6-1; (**c**) A6-5; (**d**) A6-6; (**e**) A6-9; (**f**) B5-3; (**g**) B6-1; (**h**) B6-2; (**i**) B6-3; (**j**) B6-4; (**k**) B6-5; (**l**) B6-6.

Camera	Focal length (pixels)	Principal point (pixels)	Mean reprojection error (pixels)	SD (pixels)
Cam-mean	2989.6, 2998.9	499.2, 348.6	9.8	15.3
Cam-1	2985.2, 2986.7	493.2, 343.2	15.3	26.8
Cam-2	2985.5, 2980.2	490.2, 346.1	1.2	3.6
Cam-3	2985.7, 2983.2	560.2, 350.8	2.4	6.0
Cam-4	2984.3, 2987.4	510.2, 340.3	3.6	9.0
Cam-5	2989.4, 2987.2	493.2, 341.6	6.5	8.6

Table 3.3: Estimated internal camera parameters for the FME scenario.

Camera	Position (m)	Orientation (radians)	Ground truth position (m)	Elapsed time (s)
Cam-1	−21.29, 3.39	−1.21, −0.90, −1.76	−21.73, 4.04	80.9
Cam-2	0.04, −13.42	0.45, −2.23, 0.37	−0.02, −12.75	106.1
Cam-3	43.83, 15.27	2.27, 0.18, 2.13	42.70, 15.80	182.1
Cam-4	−8.61, −4.81	1.40, −1.27, 1.25	−9.38, −5.73	95.6
Cam-5	−13.97, −0.44	−0.81, −0.80, −1.77	−14.14, −0.56	112.7

Table 3.4: Estimated external camera parameters for the FME scenario.

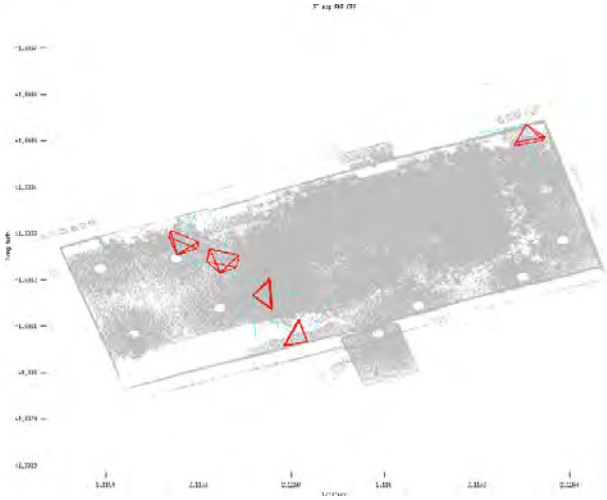
In the case of the FME scenario, all images were taken with a mobile phone. GPS readings on the phone were used as initial position estimates. The local world model was considered planar, so that a homography can be used to translate from GPS coordinates to the metric representation used in the point cloud.

Table 3.3 shows the results of the calibration of internal camera parameters. Since all images were computed using the same camera, the mean values obtained for the internal parameters can be used as a reference. These mean values are shown in the first line in the table.

Table 3.4 reports the different camera positions estimated with our algorithm and contrasted to the GPS readings on the phone. GPS coordinates are transformed to metric coordinates with the WGS84 standard and affine transformed with the DLT algorithm to align them with the FME building. Height values are not reported, as their readings from the phone GPS unit are unreliable.

This comparison is shown schematically in Figure 3.10. Figure 3.10a shows the estimated camera viewpoints, whereas Figure 3.10b shows a comparison between GPS readings (blue squares) and the camera poses computed with our method (red dots).

Figure 3.11 shows results of the optimization results for the FME dataset. In blue projected 3D lines and in red selected image lines. The heavy presence of unstructured data made it



(a)

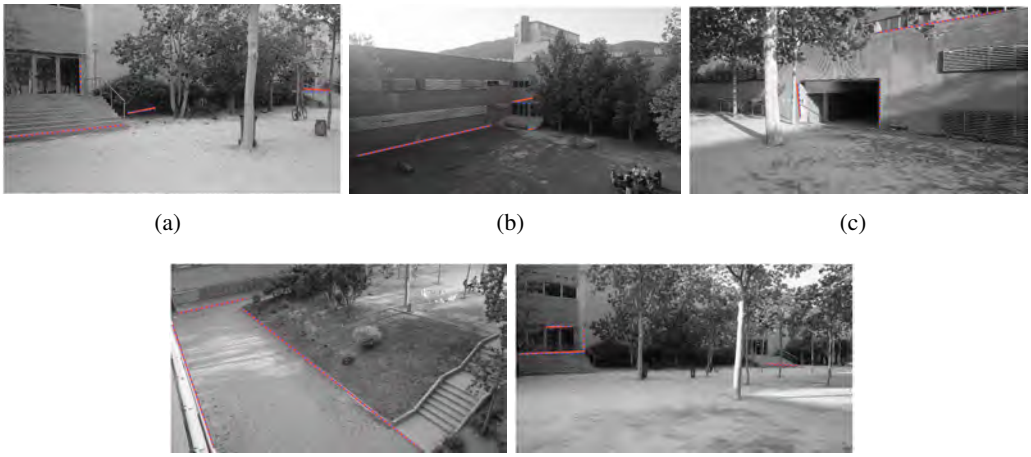
(b)

Figure 3.10: Camera localization for the FME scenario. (a) Camera viewpoint estimates; and (b) a comparison between GPS measures (blue squares) and our method (red points).

difficult to find a large number of support lines for calibration in this scenario, suggesting the need for calibration also with point/appearance features, together with lines. We leave this hybrid scheme as an open alternative for further development of the method.

Computing Homographies

To measure events occurring on the scene, such as path lengths or areas of crowdedness, it would be necessary to obtain direct mappings from images to planar regions in the floor. The idea is to have a practical way to transfer 2D images to the world coordinates of the targets detected. To this end, we compute the homographies of user-selected planes with the end of a graphical user interface. The user selects polygonal regions in the images, and the 3D points



(a) (b) (c)

(d) (e)

Figure 3.11: The results of the final calibration camera network. Optimization approximates the projected laser lines (blue) to the image lines (red) using the FME dataset. (**a**) Cam-1; (**b**) Cam-2; (**c**) Cam-3; (**d**) Cam-4; (**e**) Cam-5.

that project inside these polygons are used to approximate the 3D planes. The algorithm to compute the homographies is the standard direct linear transform [30]. Figure 3.12 shows the result of this computation for the two scenarios. Camera images, each one of a different color mask, are projected onto their corresponding planar regions in the map.

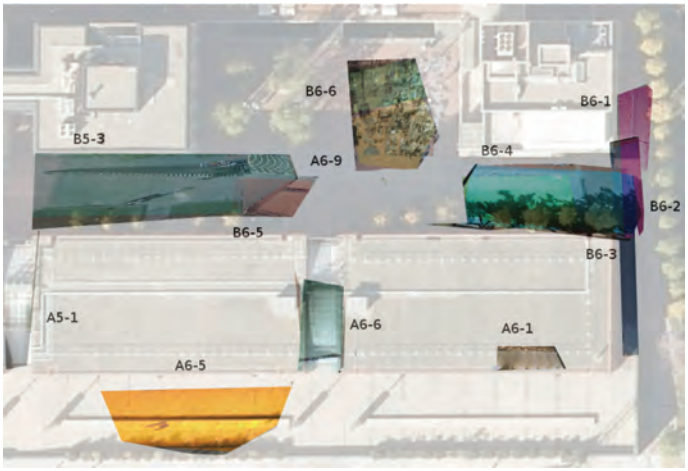
3.4 Remarks

In this chapter, we have proposed a methodology to calibrate outdoor distributed camera networks having small or inexistent overlapping fields of view between the cameras. The methodology is based on the matching of line image features with 3D lines computed from dense 3D point clouds of the scene.

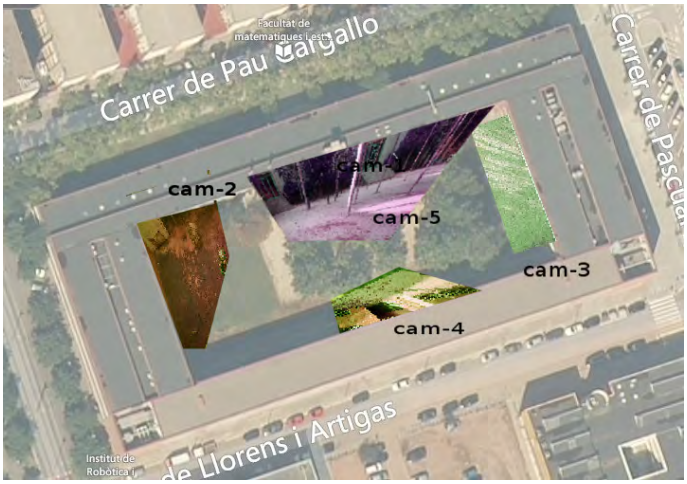
In the first stage, the user obtains the nominal calibration by using default intrinsic parameters for the cameras and indicating their positions and orientations on an aerial view aligned with the range map. Next, the calibration of each camera is improved by an automatic optimization procedure detecting lines in the 3D map and matching them with image lines. The lines are detected in the point cloud by automatically segmenting out planar regions and finding such plane intersections. The optimization procedure then minimizes the distance between points in the lines found in the map and their corresponding points in the image lines. The

method has been used to calibrate the Barcelona Robot Lab, a 10,000 m^2 area for mobile robot experimentation, and for camera localization at the FME patio, both located at the UPC campus in Barcelona.

Future work will include an analysis of uncertainty for the external calibration using the DLT-lines algorithm and the combination of feature points together with lines for scenes with a limited structure.



(a)



(b)

Figure 3.12: Computed homographies for the two scenarios. (a) BRL scenario; (b) FME scenario.

Chapter 4

Error propagation analysis of camera calibration

One way to evaluate the quality of our camera calibration system is to analyize the level of reconstruction in metric scale that can be attained with it. This chapter presents an uncertainty analysis of the camera calibration procedure presented in the previous chapter. The analysis propagates errors in pixel coordinates to the actual error in metric reconstruction.

To that end, we compute a first order linearization of the calibration method around the provided solution and perform first order error propagation [29] of this model. We validate the obtained uncertainty models with synthetic data using Monte Carlo simulations, and with real data from our experiments with the BRL camera network.

This analysis of the camera calibration is needed since the error inflicted by the bad estimation of intrinsic parameters affects the extrinsic parameter estimation [34]. We propose hence a first order error estimation analysis of the full extrinsic calibration, including the effects of the *DLT-Lines* algorithm [74] in the propagation of uncertainty.

Some precedents related to our uncertainty analysis are presented in [81], where the calibration is preformed using essential matrices, and in which first order error propagation is also proposed between the calibration and the motion parameters; or in [73], where the analysis is made measuring parameter correlation.

In our case, we take into account the uncertainty of feature extraction occuring in the two sensor modalities. Namely, we propagate image feature detection noise in pixels, and also, uncertainty in the 3D parameters of the line features extracted from the range map. For our

simulation studies, the range map is generated with a syntethic model of an RGBD camera. In this model, there exists a one to one matching between the pixels in the image and their corresponding 3D values. For the real experiments however, we resort back to our original setting, the camera network and the 3D range map computed using a laser range finder and a prior a SLAM session.

Our results compare the estimated uncertainty bounds with those values obtained using Monte Carlo simulations. We then use this result to identify bounds for the metric reconstruction level of our real world camera network.

This chapter of the thesis emerged from the collaboration at Instituto Superior Técnico at Lisbon (IST), Portugal, and we present more elaborated experiments published in CVIU 2015 [23].

4.1 Error propagation

In this section we derive first order error propagation formulas for the *DLT-Lines* calibration process. In a first step we derive the expression propagating error variance in the calibration data to error variance in the projection matrix entries. In a second step we derive error propagation from the matrix entries to the camera projection center.

Uncertainty in the 3D data is due to a wide variety of reasons. Assuming that 3D data is acquired by a sensor such as a LIDAR or an RGBD camera, we can experiment uncertainty in 3D point reconstruction from different sources. 3D point estimation error can be caused by error in the estimated camera pose, error in the estimated intrinsic parameter values of the calibration, or even due to discretization at the pixel level.

For example, an offset in the real location of a sensor implies also an offset in the position of the calibrated camera; or error in the intrinsic parameters of the 3D sensor may induce artificial zooming in the calibrated camera.

Having identified these common sources of uncertainty, we detail now how this uncertainty propagates through the *DLT-Lines* calibration methodology.

In a first step, ①, we derive an expression to propagate uncertainty in the image points $\mathbf{m}$ to uncertainty in the estimated 2D line parameters $\mathbf{l}$. In a second step, ②, we propagate such line parameter uncertainty estimates, together with that of the 3D endpoints $\mathbf{M}$, to uncertainty

estimates of the elements in the camera projection matrix $\mathbf{P}$. In a third step, ③, we propagate the uncertainty in the elements of the projection matrix to specific estimates for the uncertainty in the camera pose with respect to a calibration pattern, and the uncertainty of the intrinsic parameters, $\mathbf{R}, \mathbf{t}$, and $\mathbf{K}$, respectively. The whole process is summarized in the next diagram:

$$\begin{array}{ccccccc} \mathbf{m} & ① & \mathbf{l}, \mathbf{M} & ② & \mathbf{P} & ③ & \mathbf{R}, \mathbf{t}, \mathbf{K} \\ \Sigma_{\mathbf{m}} & \rightarrow & \Sigma_{\mathbf{l}}, \Sigma_{\mathbf{M}} & \rightarrow & \Sigma_{\mathbf{P}} & \rightarrow & \Sigma_{\mathbf{R}}, \Sigma_{\mathbf{t}}, \Sigma_{\mathbf{K}} \end{array}$$

Chapter 3 provides expressions for the transformations ①, ② and ③. In this chapter, we develop the first order approximation needed to propagate variances through the same transformations. The general rule used is that given a differentiable function $\mathbf{y} = \mathbf{f}(\mathbf{x})$, a first order covariance propagation $\Sigma_{\mathbf{y}}$ is given by

$$\Sigma_{\mathbf{y}} = \mathbf{J} \Sigma_{\mathbf{x}} \mathbf{J}^{\top} , \tag{4.1}$$

where $\mathbf{J}$ is the Jacobian of $\mathbf{f}$.

Transformation ① is explicit, but the same cannot be said of ② and ③, where no closed form is given, and such transformations come as a result of some optimization process. In these cases, the transformation is not said to be explicit, but rather implicit. The seeked Jacobian is computed using the implicit function theorem (Appendix A.2), which in general words states that the Jacobian $\mathbf{J}$ of the unknown function $\mathbf{y} = \mathbf{f}(\mathbf{x})$ satisfying the system of Equations $\mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ can be expressed in terms of the partial derivatives of $\mathbf{g}$ with respect to $\mathbf{x}$ and $\mathbf{y}$ in the form

$$\mathbf{J} = -\left(\frac{\partial \mathbf{g}}{\partial \mathbf{y}} \right)^{-1} \frac{\partial \mathbf{g}}{\partial \mathbf{x}} . \tag{4.2}$$

The result is used in the following way. Assume that a solution for $\mathbf{y}$ is given by the optimization of the cost function $C(\mathbf{x}, \mathbf{y})$. We can define the system of Equations $\mathbf{g}(\mathbf{x}, \mathbf{y})$ as the vector

$$\mathbf{g}(\mathbf{x}, \mathbf{y}) = \left(\frac{\partial C(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}} \right)^{\top} \tag{4.3}$$

that when evaluated at the minimizer $\mathbf{y}^*$ is equal to $\mathbf{0}^{\top}$.

Plugging Equation 4.3 in 4.2, the Jacbian that linearly transforms $\mathbf{x}$ to $\mathbf{y}$ becomes

$$\mathbf{J} = -\left(\frac{\partial^2 C}{\partial \mathbf{y}^2} \right)^{-1} \left(\frac{\partial^2 C}{\partial \mathbf{y} \partial \mathbf{x}} \right)^{\top} . \tag{4.4}$$

Notice that the above cost functional C correspons to an unconstrained optimization problem. If on the contrary, our optimization is subject to a set of h constraints on the optimized parameters, say for instance $\mathbf{h}(\mathbf{y}) = \mathbf{0}$, we can use a Lagrange multiplier [17] to define a new criterion

$$L(\mathbf{x}, \mathbf{y}, \lambda) = C(\mathbf{x}, \mathbf{y}) + \lambda^\top \mathbf{h}(\mathbf{y}) \,. \tag{4.5}$$

Minimizing L by setting its derivative with respect to $\mathbf{y}$ equal to zero yields

$$\left(\frac{\partial C(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}}\right)^\top + \left(\frac{\partial \mathbf{h}(\mathbf{y})}{\partial \mathbf{y}}\right)^\top \lambda = \mathbf{E} + \mathbf{K}\lambda = \mathbf{0} \,. \tag{4.6}$$

Assuming, without loss of generality that the first h rows of $\mathbf{K}$ are linearly independent, let $\mathbf{K}_1$ be the $h \times h$ top matrix of $\mathbf{K}$, and $\mathbf{E}_1$ a vector with the top h elements of $\mathbf{E}$. We can then use this subsystem to solve for the Lagrange multipliers with

$$\lambda = -\mathbf{K}_1^{-1}\mathbf{E}_1 \,. \tag{4.7}$$

We can then replace λ in the remaining equations

$$\mathbf{K}_2\lambda + \mathbf{E}_2 = \mathbf{0} \,, \tag{4.8}$$

and define the function $\mathbf{g}(\mathbf{x}, \mathbf{y})$ as follows. Its first components are equal to

$$-\mathbf{K}_2(\mathbf{y})\mathbf{K}_1^{-1}(\mathbf{y})\mathbf{E}_1(\mathbf{x}, \mathbf{y}) + \mathbf{E}_2(\mathbf{x}, \mathbf{y}) \,, \tag{4.9}$$

and its last h components are gven by $\mathbf{h}(\mathbf{y})$. This satisfies the condition $\mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, and the seeked derivative of f is hence equal to

$$\mathbf{J} = -\left(\frac{\partial \mathbf{g}}{\partial \mathbf{y}}\right)^{-1} \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \,. \tag{4.10}$$

4.1.1 From image points to homogeneous line coordinates

A closed form expression for the Jacobian $\mathbf{J_l}$ that maps image points to homogeneous lin coordinates can easily be obtained differentiating Equation 3.6 with respect to the image coordinates of the two points $\mathbf{m}_1$ and $\mathbf{m}_2$ in the image line

$$\mathbf{J_l} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ -v_2 & -u_2 & -v_1 & u_1 \end{bmatrix} \,, \tag{4.11}$$

Hence, the linear propagation of the independent point variances $\Sigma_{\mathbf{m}_1}$ and $\Sigma_{\mathbf{m}_2}$ can be computed with

$$\Sigma_{\mathbf{l}} = \mathbf{J}_{\mathbf{l}} \begin{bmatrix} \Sigma_{\mathbf{m}_1} & \mathbf{0} \\ \mathbf{0} & \Sigma_{\mathbf{m}_2} \end{bmatrix} \mathbf{J}_{\mathbf{l}}^{\top} . \tag{4.12}$$

4.1.2 From 3D point coordinates and homogeneous line coordinates to camera calibration entries

To analyze how the image line and 3D point variances $\Sigma_{\mathbf{l}}$ and $\Sigma_{\mathbf{M}}$ propagate to the calibration parameter variances in $\Sigma_{\mathbf{P}}$, we need to set as implicit function the minimizer of the set of Equations 3.8 for all the points in the image lines, in the least squares sense.

The optimal calibration matrix elements, expressed in vector form, as in Equation 3.8, correspond to the minimizer of the problem

$$\begin{aligned} \mathcal{P}^* = \arg\min_{\mathcal{P}} \; & \mathcal{P}^{\top}\mathbf{B}^{\top}\mathbf{B}\mathcal{P} \\ \text{s.t. } & \mathcal{P}^{\top}\mathcal{P} = 1 \end{aligned} \tag{4.13}$$

Our Karush-Khun-Tucker condition (Equation 4.6) for this problem becomes

$$\mathbf{g}_1 = 2\mathbf{B}\mathcal{P} + 2\lambda\mathcal{P} = \mathbf{0} \tag{4.14}$$

and the constraint is

$$g_2 = h(\mathcal{P}) = \mathcal{P}^{\top}\mathcal{P} - 1 = 0 . \tag{4.15}$$

Solving for λ in the first line of Equation 4.14, and substituting in the rest, and augmenting the system with the constraint g_2, we form $\mathbf{g}((\mathbf{l},\mathbf{M}),\mathcal{P})$ whose derivatives give a closed form expression for the seeked Jacobian

$$\mathbf{J}_{\mathcal{P}} = -\left(\frac{\partial \mathbf{g}((\mathbf{l},\mathbf{M}),\mathcal{P})}{\partial \mathcal{P}}\right)^{-1} \left(\frac{\partial \mathbf{g}((\mathbf{l},\mathbf{M}),\mathcal{P})}{\partial (\mathbf{l},\mathbf{M})}\right) , \tag{4.16}$$

and

$$\Sigma_{\mathcal{P}} = \mathbf{J}_{\mathcal{P}} \begin{bmatrix} \Sigma_{\mathbf{l}} & \mathbf{0} \\ \mathbf{0} & \Sigma_{\mathbf{M}} \end{bmatrix} \mathbf{J}_{\mathcal{P}}^{\top} . \tag{4.17}$$

4.1.3 From camera calibration matrix entries to camera pose

Propagating uncertainty from the estimated $\mathbf{P}$ into its decomposition $\mathbf{K}[\mathbf{R}|\mathbf{t}]$, or the separate intrinsic parameters (scaling, shear or principal point), involves computing the Jacobian of the transformation starting from the QR decomposition from Equation 3.13, and ending with the calibration parameters extracted from $\mathbf{P}$.

Despite straightforward, this process involves lengthy expressions for each parameter, which can be used as indicators of the precision of the calibration. The uncertainty of the camera projection center is one such indicator that can be computed using concise expressions.

Denoting the projection center, referred to the world coordinate system, as $\mathbf{c} = [c_x\ c_y\ c_z]^\top$, one has that $\mathbf{c}$ projects to a point at infinity, $\mathbf{m}_c = [0\ 0\ 0]^\top = \mathbf{0}$, i.e.,

$$\mathbf{P}\,[\mathbf{c}^\top 1]^\top = \mathbf{0}. \tag{4.18}$$

Representing the projection matrix as a collection of columns, $\mathbf{P} = [\mathcal{P}_1\ \mathcal{P}_2\ \mathcal{P}_3\ \mathcal{P}_4]$, the projection center can be computed as the solution of the linear system of three equation in three unknowns, $[\mathcal{P}_1\ \mathcal{P}_2\ \mathcal{P}_3]\mathbf{c} = -\mathcal{P}_4$.

Considering that one wants to apply operations (derivatives) on the transformation from $\mathbf{P}$ to $\mathbf{c}$, it is convenient to derive a closed form expression for $\mathbf{c}$. Using the Cramer's rule to solve the system, one has

$$t_x = \det([-\mathcal{P}_4,\ \mathcal{P}_2,\ \mathcal{P}_3])/w \tag{4.19}$$

$$t_y = \det([\mathcal{P}_1,\ -\mathcal{P}_4,\ \mathcal{P}_3])/w \tag{4.20}$$

$$t_z = \det([\mathcal{P}_1,\ \mathcal{P}_2,\ -\mathcal{P}_4])/w \tag{4.21}$$

where $w = \det([\mathcal{P}_1\ \mathcal{P}_2\ \mathcal{P}_3])$.

It is interesting to note that the choice of the world coordinate system is key to obtain concise expressions. Instead of $\mathbf{t} = \mathbf{K}^{-1}\mathcal{P}_4$, one has $\mathbf{c} = \mathbf{P}_{3\times3}^{-1}\mathcal{P}_4$ which differs from $\mathbf{t}$ just by a rotation. Noting that $\mathbf{P}_{3\times3} = \mathbf{KR}$ and $\mathcal{P}_4 = \mathbf{Kt}$, one has $\mathbf{c} = (\mathbf{KR})^{-1}\mathbf{Kt} = \mathbf{R}^{-1}\mathbf{t}$, and hence avoiding decomposing $\mathbf{P} = \mathbf{K}[\mathbf{R}\ \mathbf{t}]$.

Hence, computing the derivatives of Equations 4.19- 4.21, with respect to the terms in $\mathbf{P}$, one propagates explicitly the error variance of the projection matrix, $\Sigma_\mathbf{P}$, to the error variance of the projection center, $\Sigma_\mathbf{c}$

$$\Sigma_\mathbf{c} = \mathbf{J}_\mathbf{c}\Sigma_\mathcal{P}\mathbf{J}_\mathbf{c}^\top\ , \tag{4.22}$$

where

$$\mathbf{J_c} = \frac{1}{w^2}\begin{bmatrix} ad & -mw^2-ae & nw^2+af & -dw & -ag & pw^2+ah & -qw^2-ai & gw & aj & -sw^2-ak & tw^2+al & -jw \\ mw^2-bd & be & -ow^2-bf & ew & bg-pw^2 & -bh & rw^2+bi & -hw & sw^2-bj & bk & -uw^2-bl & kw \\ cd-nw^2 & ow^2-ce & cf & -fw & qw^2-cg & ch-rw^2 & -ci & iw & cj-tw^2 & uw^2-ck & cl & -lw \end{bmatrix}$$

and

$$\begin{aligned} a &= -\det([-\mathcal{P}_4,\ \mathcal{P}_2,\ \mathcal{P}_3]) \\ b &= \det([\mathcal{P}_1,\ -\mathcal{P}_4,\ \mathcal{P}_3]) \\ c &= -\det([\mathcal{P}_1,\ \mathcal{P}_2,\ -\mathcal{P}_4]) \\ d &= \mathbf{P}_{22}\mathbf{P}_{33} - \mathbf{P}_{23}\mathbf{P}_{32} \\ e &= \mathbf{P}_{21}\mathbf{P}_{33} - \mathbf{P}_{23}\mathbf{P}_{31} \\ f &= \mathbf{P}_{21}\mathbf{P}_{32} - \mathbf{P}_{22}\mathbf{P}_{31} \\ g &= \mathbf{P}_{12}\mathbf{P}_{33} - \mathbf{P}_{13}\mathbf{P}_{32} \\ h &= \mathbf{P}_{11}\mathbf{P}_{33} - \mathbf{P}_{13}\mathbf{P}_{31} \\ i &= \mathbf{P}_{11}\mathbf{P}_{32} - \mathbf{P}_{12}\mathbf{P}_{31} \\ j &= \mathbf{P}_{12}\mathbf{P}_{23} - \mathbf{P}_{13}\mathbf{P}_{22} \\ k &= \mathbf{P}_{11}\mathbf{P}_{23} - \mathbf{P}_{13}\mathbf{P}_{21} \\ l &= \mathbf{P}_{11}\mathbf{P}_{22} - \mathbf{P}_{12}\mathbf{P}_{21} \\ m &= \frac{\mathbf{P}_{23}\mathbf{P}_{34} - \mathbf{P}_{24}\mathbf{P}_{33}}{w} \\ n &= \frac{\mathbf{P}_{22}\mathbf{P}_{34} - \mathbf{P}_{24}\mathbf{P}_{32}}{w} \\ o &= \frac{\mathbf{P}_{21}\mathbf{P}_{34} - \mathbf{P}_{24}\mathbf{P}_{31}}{w} \\ p &= \frac{\mathbf{P}_{13}\mathbf{P}_{34} - \mathbf{P}_{14}\mathbf{P}_{33}}{w} \\ q &= \frac{\mathbf{P}_{12}\mathbf{P}_{34} - \mathbf{P}_{14}\mathbf{P}_{32}}{w} \\ r &= \frac{\mathbf{P}_{11}\mathbf{P}_{34} - \mathbf{P}_{14}\mathbf{P}_{31}}{w} \\ s &= \frac{\mathbf{P}_{13}\mathbf{P}_{24} - \mathbf{P}_{14}\mathbf{P}_{23}}{w} \\ t &= \frac{\mathbf{P}_{12}\mathbf{P}_{24} - \mathbf{P}_{14}\mathbf{P}_{22}}{w} \\ u &= \frac{\mathbf{P}_{11}\mathbf{P}_{24} - \mathbf{P}_{14}\mathbf{P}_{21}}{w} \end{aligned}$$

and w as defined above. Similar expressions can be computed for the rest of the pose parameters.

4.2 Experiments

In order to validate the proposed uncertainty analysis formulas we conduct some experiments in a synthetic environment for which one has available precise and accurate ground truth. In addition, we apply the proposed uncertainty analysis to a real setup based on an outdoor scene, the Barcelona RobotLab, which has been reconstructed in 3D using lidar data, thus providing directly the required 3D information for the *DLT-Lines* method.

4.2.1 Synthetic experiments

In this section the variance of the entries of the projection matrix, $\Sigma_{\mathbf{P}}$, predicted using the proposed uncertainty analysis (Equation 4.17) is compared against a synthetic a Monte Carlo simulation. The simulation allows us to vary the internal camera parameter accuracies as well as the precision of the camera pose, and to set a fixed number of image points and their 3D references as calibration pattern without the need to resort to the actual image processing routines for feature extraction and matching at the various noise levels We consider various levels of white Gaussian noise in the localization of the image points, the localization of the $3D$ points, or both.

Two simulation setups were generated, the first synthetic setup is formed by two cameras, namely a mobile color-depth (RGBD) camera which collects 3D data and a fixed RGB camera. See Figure 4.1. Frame (b) in that figure shows a synthetic image simulated for the RGB camera, while frame (d) and frame (e) show synthetic intensity and range images simulated for the RGBD camera.

In this setup, we analyze what happens when just the RGB image has noise. In other words, the noise in 3D points is set to null ($\sigma_{\mathbf{M}} = 0$). To improve readability, variance is written using upper case, Σ, and standard deviation is written using lower case, σ. The uncertainty analysis was performed using both the proposed propagation methodology and Monte Carlo simulations. The Monte Carlo simulation was configured for 300 runs at each level of noise. The standard deviation of the noise in the 2D points varies from 0 to 6 pixels, and the standard deviation of the 3D location of feature points varies from 0 to 2 cm. For all these runs, the variance of every entry in $\mathbf{P}$, i.e. $\Sigma_{\mathbf{P}_{ij}}$ was computed.

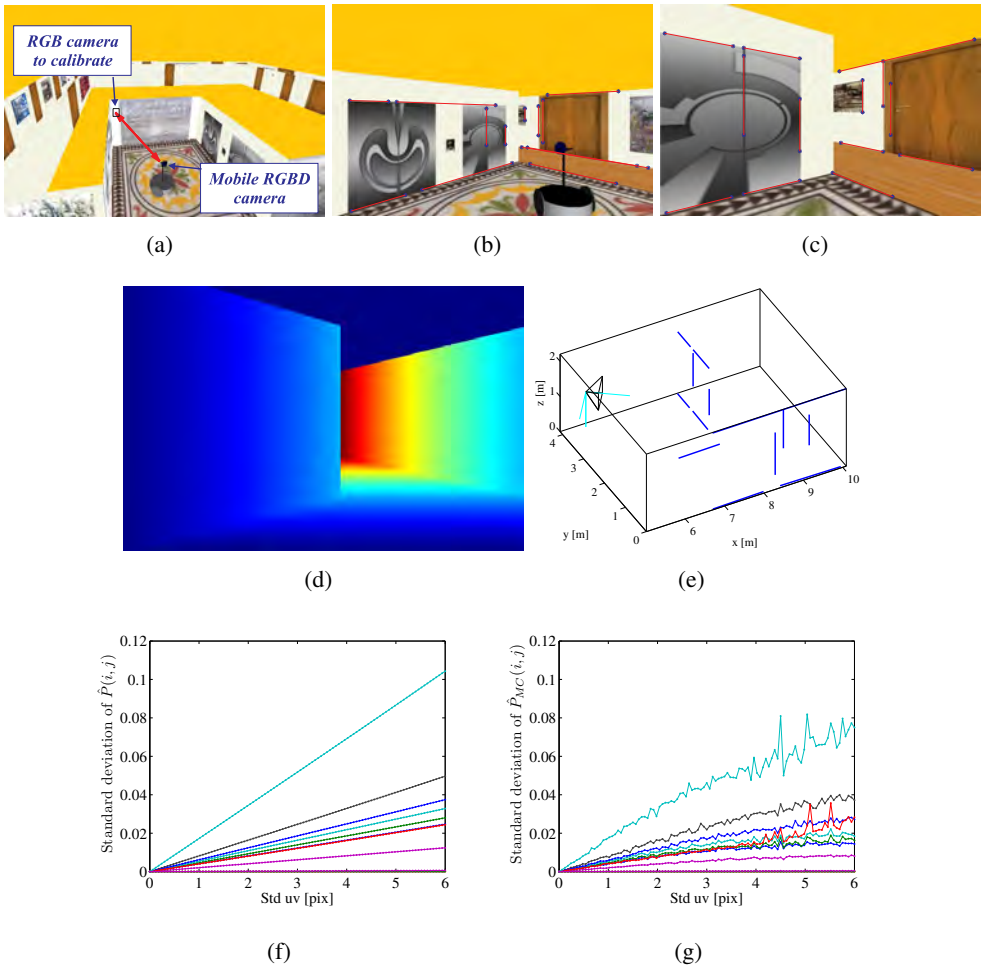
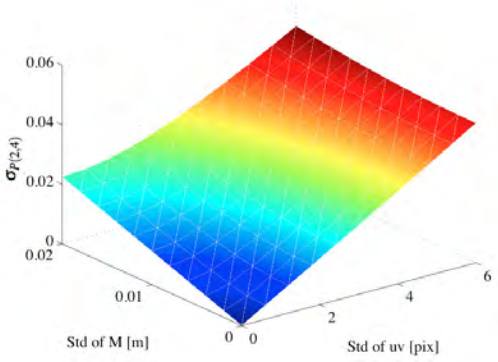
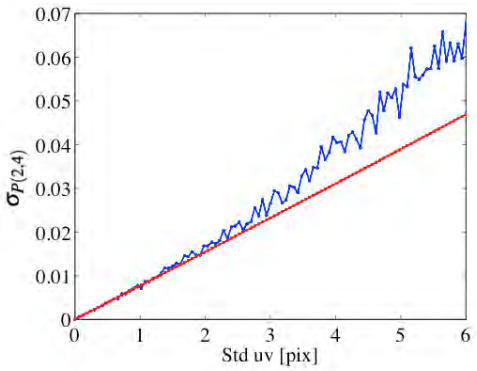


Figure 4.1: Analysis of camera calibration uncertainty. (a) VRML setup. (b) RGB image. (c) RGBD intensity image. (d) RGBD range image. Each line defined in the RGBD image corresponds to a line in the RGB image, and leads to a 3D line in the world/RGBD coordinate system. (e) 3D lines form the required input data for *DLT-Lines* calibration. (f) Relation between the error in the RGB image coordinates and the projection matrix parameters. (g) Monte Carlo simulations of the same relation between image error standard deviation and the standard deviation of the projection matrix elements.

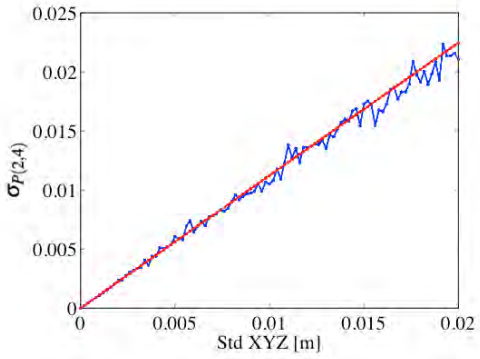
(a) Setup



(b) Noise in *uv* and *xyz*



(c) Noise in *uv*



(d) Noise in *xyz*

Figure 4.2: Single camera setup. (a) 3D information is known for the image lines shown. (b) Analytic computation of the propagation of the combined noise in pixels for the *uv* coordinates of calibration points abd the *xyz* coordinates of the calibration pattern, to the element $\mathbf{P}_{24}$ in the calibration matrix. (c) Propagation of noise in pixels for *uv* coordinates of calibration points to noise in the entry $\mathbf{P}_{24}$ of P. In red the analytic result, and in blue the Monte Carlo simulations. (d) Propagation of noise in meters for the *xyz* coordinates of the calibration pattern to noise in the entry $\mathbf{P}_{24}$ of $\mathbf{P}$. In red the analytic result, and in blue the Monte Carlo simulations.

The linear propagation of the standard deviation of each of the entries of $\mathbf{P}$, computed with the proposed methodology, is shown in frame (f) in the same figure. As expected, some entries of $\mathbf{P}$ are more robust to noise than others. Frame (g) shows the Monte Carlo simulation results for varying levels of pixel noise, again for all the entries in $\mathbf{P}$. Plots (f) and (g) indicate that the analytical values obtained using the linear propagation analysis match those of Monte Carlo results for values of $\sigma_{\mathbf{m}}$ lower than approximately 3 pixels. Hence, validating out linear approximation for the propagation of uncertainties in the camera calibration process.

Nonlinearities have more incidence for larger levels of image noise, making our first order approximation unreliable. Nonetheless, pixel value noises in ranges below 3 pixels are acceptable for most imaging sensors.

The second setup is based on a single RGBD camera. The setup can be seen in Figure 4.2(a), which corresponds to a typical 'L' shaped corridor. Camera calibration ground truth is known and is used to assess the validity of the noise propagation estimation method. Frame (b) shows the theoretical value for $\sigma_{\mathbf{P}_{24}}$ in the presence of noise, simultaneously in both the image and the range values. The plot shows the correlated effects between the image and range noise values.

Monte Carlo simulations were also run for this setup. Plots (c) and (d) show both the analytic and estimated value of σ for $\mathbf{P}_{24}$ as a function of variations in image and depth noise.

Plot (c) shows once more that the first order approximation is only valid up to around $\sigma_{\mathbf{m}} = 3$ pixels. The theoretical prediction is nevertheless accurate for lower levels of noise showing that the proposed uncertainty analysis takes correctly into account the scene structure. Nonlinear effects have less influence for variations of range as shown in plot (d).

4.2.2 Experiments in real scenarios

For our real scenario experiments we use a mobile Pioneer 3AT robot equipped a 3D range sensing device consisting of a Hokuyo UTM-30LX laser mounted on a slip-ring. The laser resolution is set to 0.5 degrees in azimuth with 360 degree omnidirectional field of view, and 0.5 degrees resolution in elevation for a range of 270 degrees. Each point cloud contains 194,580 range measurements of up to 30 meters with noise varying from 30mm for distances closer to 10m, and up to 50mm for objects as far as 30m. Our robot includes also two Flea2 cameras [52]. The dataset used for the experiments is the Barcelona RobotLab dataset [84].

We analyze pose error in different indoor and outdoor scenarios. The first experiment involves an indoor scenario. In this scenario, a camera calibration pattern is used to compare classical image based calibration with our 3D-2D line-based calibration scheme. We show the result in Figure 4.3. In this setup we used 12 lines, the noise in 3D points is ($\sigma_{\mathbf{M}} = 0$) meters, and image points ($\sigma_{\mathbf{m}} = 2$) pixels. The Ellipsoid representing the position uncertainty can be seen in the zoomed part (blue ellipse).

The next experiment consists in the estimation of pose uncertainty for the calibration of the camera network explained in [56]. Calibration results for a subset of cameras are shown in Figure 4.4. The top frames contain the camera images, the reprojected 3D point cloud, and the 3D lines used for calibration. The bottom frame shows again the 3D lines used for calibration and the estimated 3D poses and their associated position covariances. These covariances are magnified 10 times to ease visualization.

4.2.3 Experiments discussion

Using only vertical and horizontal lines at the roof level, $z = 0$, results in a rank deficient problem, more precisely, $rank(\mathbf{B}) = 10$ for matrix $\mathbf{B}$ in Equation 4.13.

Rewriting $\mathbf{B}$ as $\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^T$. And, letting $\mathbf{V} = [\mathbf{v}_1 \ \ldots \ \mathbf{v}_{12}]$ be the 12 singular vectors of $\mathbf{B}$. In the case of $rank(\mathbf{B}) = 10$, the solution is a linear combination of the last two singular vectors, $\mathbf{v}_{11}$ and $\mathbf{v}_{12}$ corresponding to the smallest singular values of $\mathbf{B}$

$$\mathcal{P}^* = w_{11}\mathbf{v}_{11} + w_{12}\mathbf{v}_{12} \tag{4.23}$$

This solution has an ambiguity between the camera height and the vertical focal length. The null space is a set of camera configurations where the camera has fixed x and y coordinates while z varies. As z gets higher, the camera is rotated downwards and the vertical focal length is augmented so that the imaging does not change.

In order to constrain the solution, we use the square pixels constraint to reformulate the calibration problem as a 1D nonlinear optimization problem

$$\begin{aligned} \mathcal{P}^* &= w^*\mathbf{v}_{11} + \sqrt{1-(w^*)^2}\mathbf{v}_{12} \\ w^* &= arg\ _w\ \min \|\mathbf{K}_{11} - \mathbf{K}_{12}\| \end{aligned} \tag{4.24}$$

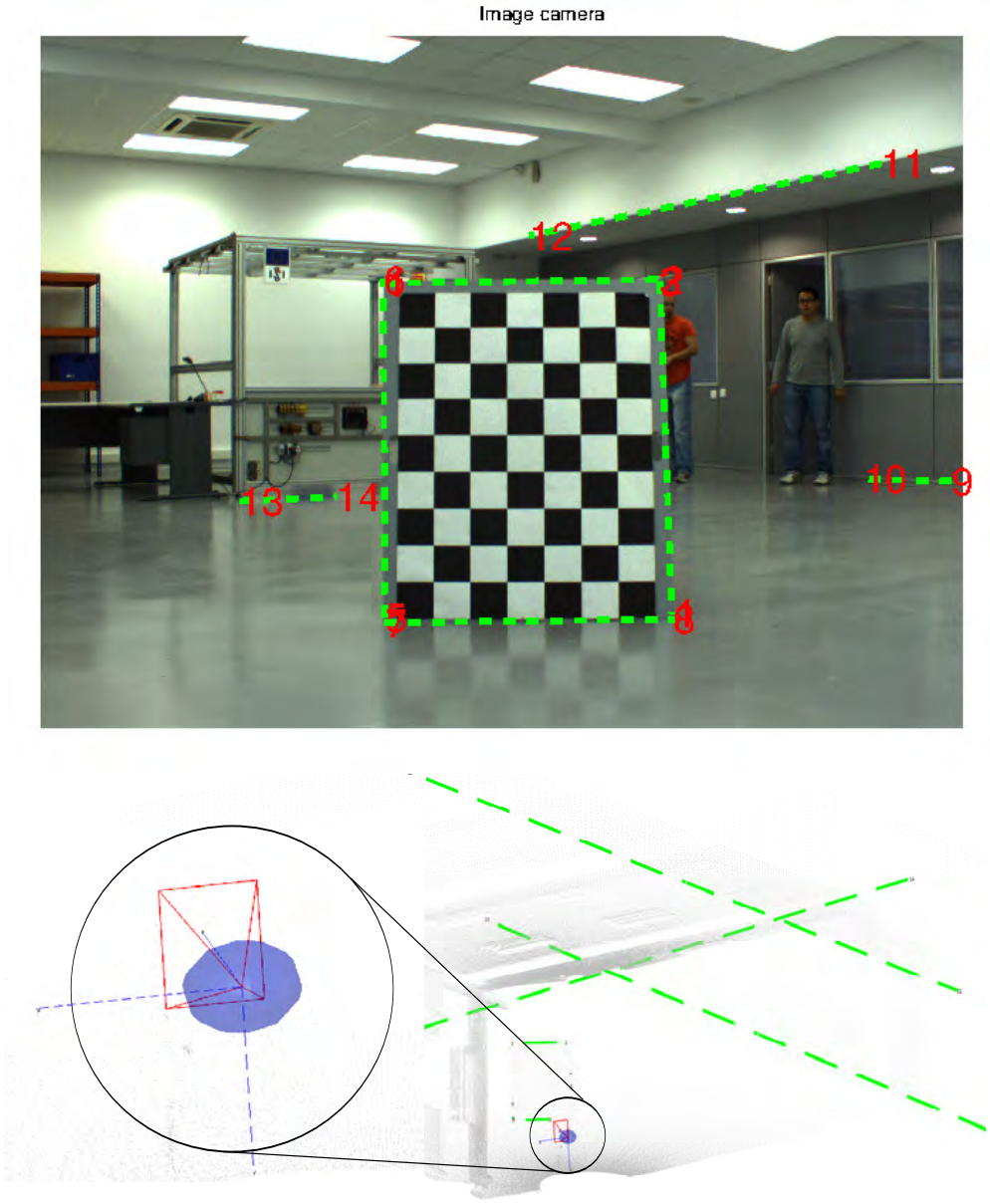
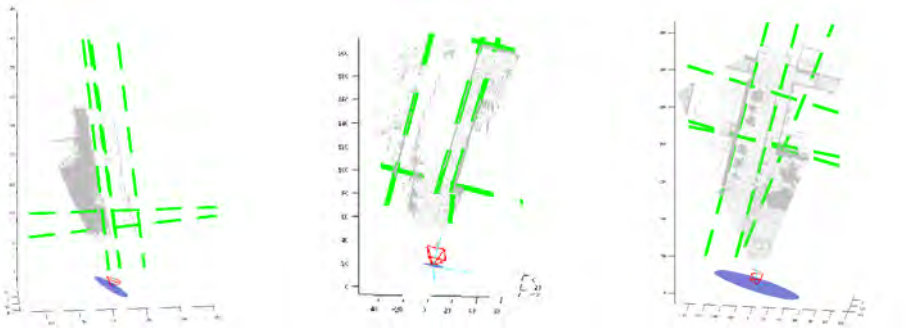


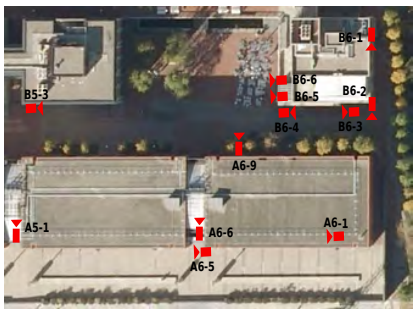
Figure 4.3: Propagation of pose uncertainty for an indoor experiment. The top frame shows the lines used, and the inset in the bottom frame shows the computed pose covariance.

(a) Camera B52 (b) Camera B62 (c) Camera B65



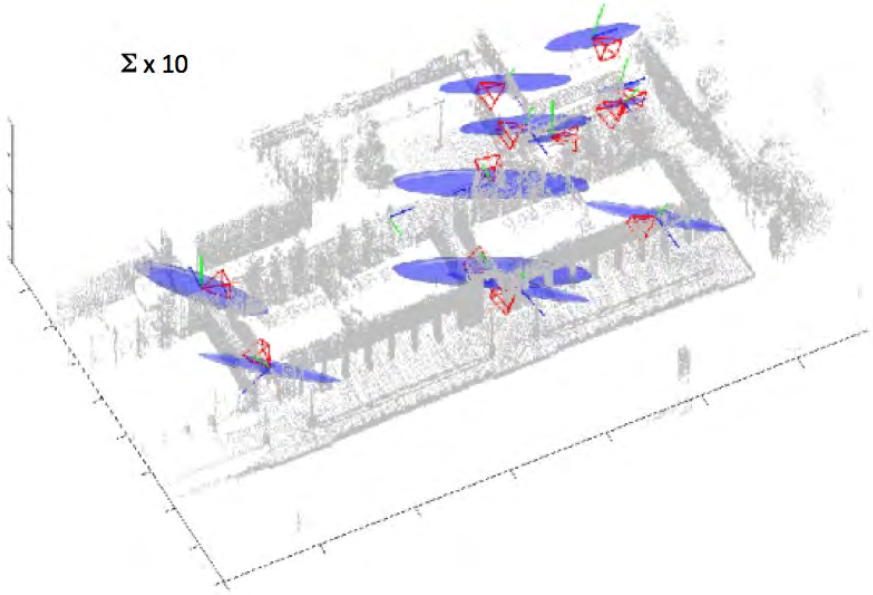
(d) Camera B52 (e) Camera B62 (f) Camera B65

Figure 4.4: Barcelona RobotLab. Top frames: Reprojected 3D point clouds and 2D image lines used for calibration for the cameras with labels B52, B62 and B65. Bottom frames: 3D lines and estimated robot locations and robot location covariances. Covariance hyper-ellipsoids have been magnified 10 times to ease visualization.



(a)

(b)



(c)

Figure 4.5: (a) Barcelona RobotLab camera network used in our calibration experiments; (b) platform used to collect the 3D map; (c) propagation of the image error onto the camera pose. The results have been enlarged by a factor 10 to ease visualization.

where $w^* \in [0,\ 1]$ and the intrinsic parameters matrix, $\mathbf{K}$, is computed through QR decomposition of the vector $\mathcal{P}^*$ reshaped to a 3×4 matrix. Note that by construction $\|\mathcal{P}^*\| = 1$, since $\mathbf{v}_{11}$ and $\mathbf{v}_{12}$ are orthogonal and have unit norm. The interval $w^* \in [-1,\ 0]$ is not considered since $-\mathcal{P}^*$ and $\mathcal{P}^*$ lead to the same solution.

The solution obtained is displayed graphically in Figure 4.5(c), and allows retrieving the camera pose (localization and rotation) and its intrinsic parameters.

In the above mentioned analysis we are using a simplified perspective projection model that does not consider radial distortion of the image. However, since the results depend on the line extraction method, and thus on the quality of line estimates on the images, the results on pose estimation will nonlinearly degrade to a point in which the estimate will be inconsistent with our linear propagation of variances shown in Figure 4.1. As reported in [74], the error of the computed horizontal focal lenght was 4.9×10^{-5}, the reprojection error $0.4707[pix^2]$, the rotation error 0.01 radians, and the pose error 0.0092 meters, still within proper uncertainty bounds.

4.3 Remarks

In this chapter we presented a methodology to analyze how uncertainty is propagated form image pixels and 3D point coordinates to the calibration parameter estimates of a particular line-based camera calibration methodology. The experiments shown include combination of image data (2D line extraction) and 3D data in the form of ranges acquired from a ToF sensor, a lidar, or an RGBD camera.

Our methodology starts by estimating the camera projection matrix using the coordinates of 3D lines and their image correspondances using the *DLT-Lines* algorithm. In other words, the projection matrix is estimated by minimizing a quadratic cost-function of the reprojection error (MLS). The fact that calibration corresponds to the minimization of a cost function allows propagating the covariance of the calibration data to estimate the covariance of the projection matrix parameters using the implicit function theorem. Given the estimate of the covariance of the projection matrix parameters we can finally propagate the uncertainty to the camera location. In this case, explicit expressions are derived in closed form. We have demonstrated that our uncertainty analysis is consistent by testing it with Monte Carlo simulations. And we also show and application of this propagation of uncertainty to a real outdoors scenario.

Figure 4.5 shows the result of that error propagation in the form of uncertainty hyperellipsoid bounds (magnified 10 times) for the metric reconstruction of the camera locations in the Barcelona Robot Lab.

Chapter 5

Segmentation of dynamic objects using a low-rate data acquisition 3D sensor

In the previous chapters we presented a method to calibrate a camera network fusing range data and images, as well as means to evaluate to what extent noise in the extraction of image features or 3D features influences the calibration parameters, and eventually the camera pose estimate. All the methods seen thus far work on the premise that the scene is static, and that all imaged elements are solidary to a common reference frame. In reality, this is seldom the case, and robotics applications must be able to handle scenes with moving elements, also possibly during calibration.

For this reason it is desirable to develop subsystems that can detect what is static and what is dynamic on the scene, and in our case again, we can benefit from the possibility of fusing 3D range data with that of the cameras.

In the following two chapters we present two techniques to segment out dynamic objects from a scene, so that the static data can eventually be used for calibration, or the dynamic data be analyzed with other purposes. This chapter presents a method to segment out dynamic objects using a low rate scanning device, and Chapter 6 develops a similar methodology but for the case of a fast real-time 3D scanning device.

2D and 3D lidar scanning are popular sensors often used in mobile robotics. They are used for robot navigation [41], trajectory planning [80], scene reconstruction [77], and even object recognition [5]. Aside from pricey devices such as the Velodyne $HDL-64E$, high resolution 3D lidar scanning is only possible at low-frame rates. As an example, we have built

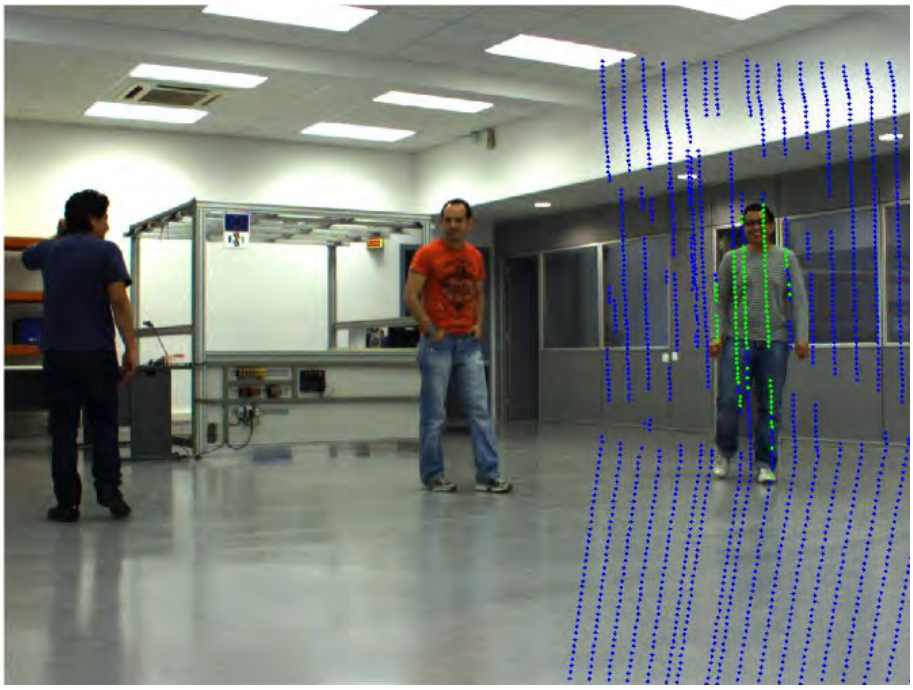


Figure 5.1: Several laser scans of a dynamic object reprojected on their corresponding image frame.

an omnidirectional lidar sensing device for outdoor mobile robotics applications that scans with resolutions and acquisition times that range from 0.5 degrees at 9 seconds per revolution to finer point clouds sampled at 0.1 degrees resolution at a more demanding processing time of 45 seconds per revolution. This sensor has been devised for low-cost, dense $3d$ mapping. The removal of dynamic and spurious data from the laser scan is a prerequisite to dense 3d mapping.

We address the problem of dynamic object segmentation by synchronizing such laser range sensor with a color camera, and using the high frame-rate image data to segment out dynamic objects from the low-rate acquired points clouds. Per-pixel class properties of image data are adapted online using Gaussian mixture models (GMM). The result is a synchronized labeling of foreground/background corresponding laser points and image data as shown in Figure 5.1.

As we have seen in previous chapters, methods that study the segmentation of 3D laser data usually focus on the extraction of valuable geometric primitives such as planes or cylinders [55] with applications that vary from map building, to object classification [15], road clas-

sification [47], or camera network calibration [54]. All these methods however are designed to work on static data and do not consider temporal information. For outdoor map building applications, the removal of dynamic objects from the laser data is desirable. Furthermore, for low-rate scanning devices such as ours, moving items in the scene would appear as spurious 3D data; hence the need to segment them out.

Background segmentation is a mature topic in computer vision, and is applied specially to track objects in scenarios that change illumination over time but keep the camera fixed to a given reference frame. The most popular methods adapt the probability of each image pixel to be of background class using the variation of intensity values over time. Such adaptation can be tracked with the aid of a Kalman filter [65] taking into account illumination changes and cast shadows. These methods can be extended to use multimodal density functions [78, 79] in the form of Gaussian mixture models, whose parameters are updated depending on the membership degree to the background class.

Range data only may not be sufficient for a proper classification, and appearance information might be also useful. The classification of objects fusing 3D range data and appearance information has been addressed in the past, again for the analysis of static scenes only. Posner et al. [60–62] proposed an unsupervised method that combines 3D laser data and monocular images to classify image patches to belong to a set of 8 different object classes. The technique oversegments images based on texture and appearance properties, and assigns geometric attributes to these patches using the reprojected 3D point correspondences. Each patch is then described by a bag of words and classified using a Markov random field to model the expected relationship between patch labels.

These methods (and ours) have as a prerequisite the accurate calibration of both sensors, the laser and the camera. The computation of the rigid body transformation between 2D and 3D laser scanners and a camera are common procedures in mobile robotics and are usually solved with the aid of a calibration pattern. The techniques vary depending on the type of sensor to calibrate, and on the geometric motion constraints between the two sensor reference frames [49, 54, 89, 98]. Sensor synchronization on the other hand has received less attention. Sensor synchronization and occlusions are studied in [71] for the case of the Velodyne HDL-64 sensor. A more general method to synchronize sensors with varying latency is proposed in [51].

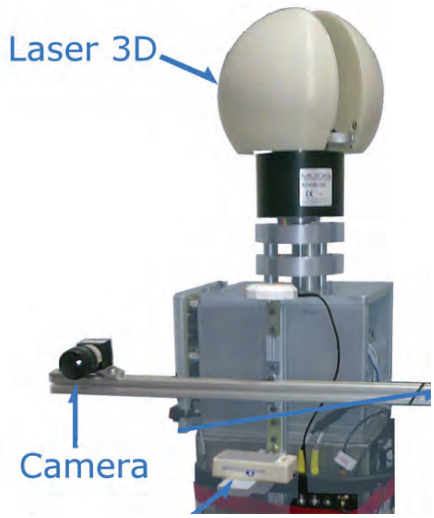


Figure 5.2: Our custom built 3D range sensing device and a rigidly attached color camera.

The chapter is organized as follows. Section 5.1 gives our custom built sensor specifications, and details the methods developed for sensor synchronization and sensor calibration. Section 5.2 details the background segmentation algorithm. Results of the method are shown in Section 5.3 on a real indoor scenario with several people moving with random patterns. Conclusions and future work are detailed in Section in 5.4.

5.1 Sensor synchronization and calibration

5.1.1 Sensor specifications and data acquisition

Our 3D range sensing device consists of a Hokuyo UTM-30LX laser mounted on a slip-ring, with computer-controlled angular position via a DC brushless motor and a controller. For the experiments reported in this chapter, laser resolution has been set to 0.5 degrees in azimuth with 360 degree omnidirectional field of view, and 0.5 degrees resolution in elevation for a range of 270 degrees. Each point cloud contains 194,580 range measurements of up to 30 meters with noises varying from 30mm for distances closer to 10m, and up to 50mm for objects as far as 30m. The color camera used is a Pointgray Flea camera with M1214-MP optics and 40.4 degree field of view. Figure 5.2 shows a picture of the entire unit.

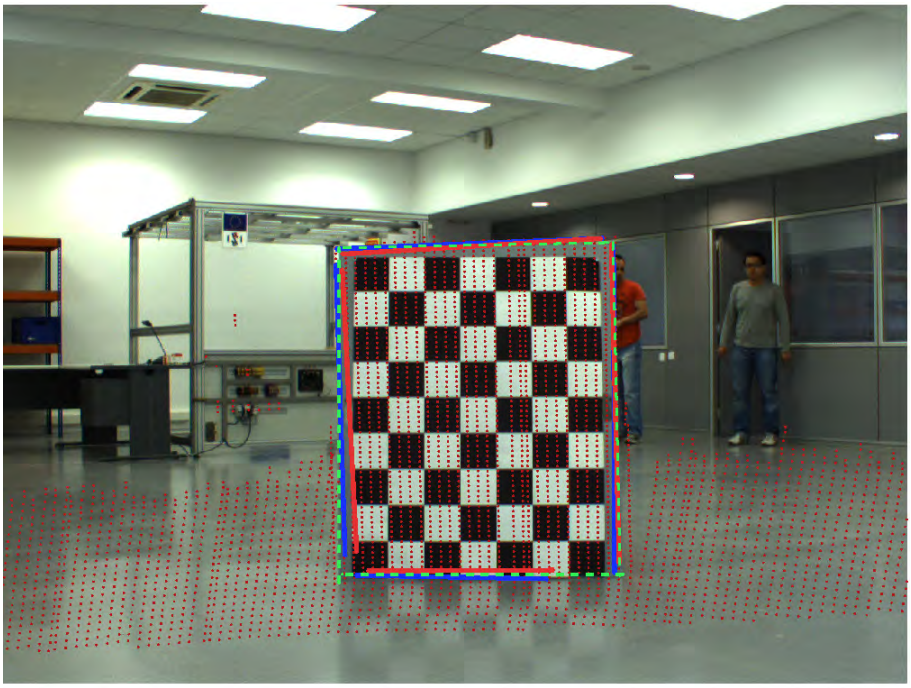


Figure 5.3: Laser-camera pose refinement using line primitives. The green dotted lines show the image features. Red lines show reprojection prior to pose refinement, and blue lines correspond to refined reprojected estimates.

5.1.2 Sensor calibration

We are interested in the accurate registration of laser range data with intensity images. Registration can be possible by first calibrating the intrinsic camera parameters and then, finding the relative transformation between the camera and laser reference frames. Intrinsic camera calibration is computed using the method described in Chapter 3, although other methods could also be used [30, 87, 99]. Extrinsic calibration between the laser and camera is initialized by selecting correspondences of the calibration plane corners on both sensing modalities with the aid of a graphical user interface, and using Hagger's method for pose estimation [37], as shown in Figure 5.3.

The method is subject to the resolution of the laser scanner for the selection of the four 3D to 2D corner matches in the pattern. Pose estimation is further refined by minimizing the reprojection error of line primitives. Lines in the 3D point cloud are obtained growing and intersecting planar patches as in [55]. Their corresponding matches in the images are manually

selected using the graphical user interface.

Line reprojection error is computed as the weighted sum of angular and midpoint location reprojection errors,

$$\varepsilon = \sum (\theta_i - \theta_p)^2 + w(\mathbf{m}_i - \mathbf{m}_p)^T (\mathbf{m}_i - \mathbf{m}_p) \tag{5.1}$$

where θ indicates the line orientation in the image in radians, and $\mathbf{m}$ are the line center image coordinates. The subscript i corresponds to measured image features, and the subscript p indicates projected model features. The weight w is a free tuning parameter to account for the difference between angular and Cartesian coordinates. Figure 5.3 shows in green the measured image lines, and in red the initial estimates of the reprojected lines. Once Eq. 5.1 is optimized for, the resulting reprojected lines are those shown in blue.

5.1.3 Synchronization

At 0.5 degree resolution, our 3D scanner takes about 9 seconds to complete a scan, which is made of a 180 degree turn of the sensor. The camera frame rate is set to 17 fps, thus we have roughly 153 images per full 3D image.

The timestamps between consecutive laser slices $t_{\texttt{slice}_i}$, and grabbed images $t_{\texttt{frame}_j}$ are compared and set to lie within a reasonable threshold T_s in milliseconds.

$$|t_{\texttt{slice}_i} - t_{\texttt{frame}_j}| \leq T_s \tag{5.2}$$

With $T_s = 1/17$, each laser scan is uniquely assigned to its corresponding image frame, roughly two to three per image. Increasing this threshold, allows to increase the number of laser slices that can be assigned to one image as shown in Figure 5.4.

5.2 Background substraction

Once we have time correspondences between 3D laser slices and image frames, we can use background segmentation results on the image sequence to classify the corresponding 3D points in each time slice as belonging to a dynamic or static object. The method we implemented, inspired in [79], is explained next. It models the two classes, object and background, as Gaussian mixtures.

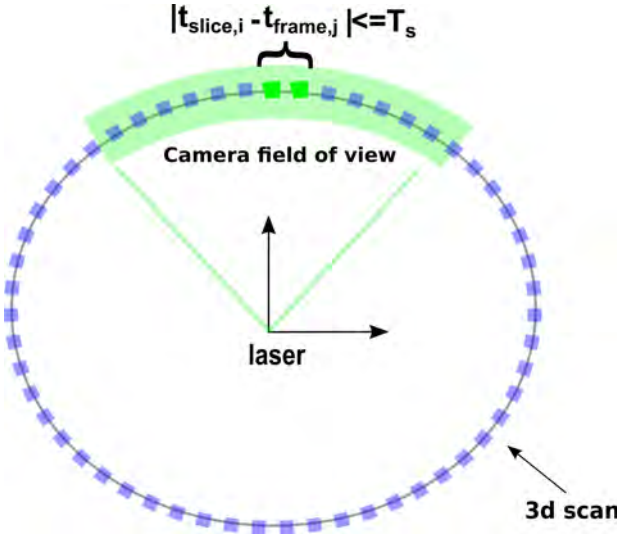


Figure 5.4: Camera and laser synchronization.

5.2.1 Mixture model

For each pixel in the image, the probability of its RGB coordinates $\mathbf{x}$ to be of the background class is modeled as a mixture of K Gaussian distributions.

$$p(\mathbf{x}) = \sum_{k=0}^{K} \omega_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) \tag{5.3}$$

with ω_k the weight of the k-th Gaussian, and K a user selected number of distributions.

This classification scheme assumes that the RGB values for neighboring pixels are independent. During the training session, when a pixel RGB value $\mathbf{x}$ falls within 2.5 standard deviations of any of the distributions in the sum (in the Mahalanobis sense), evidence in the matching distributions is stored by recursively updating their sample weight, mean, and variance with

$$\omega_k(t+1) = (1-\alpha)\omega_k(t) + \alpha \tag{5.4}$$

$$\mu_k(t+1) = (1-\rho)\mu_k(t) + \rho\mathbf{x} \tag{5.5}$$

$$\Sigma_k(t+1) = (1-\rho)\Sigma_k(t) + \rho(\mathbf{x}-\mu(t))^T(\mathbf{x}-\mu(t)) \tag{5.6}$$

and

$$\rho = \alpha\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) \tag{5.7}$$

Note once ω is updated in Equation 5.4, the weights need to be renormalized. And, just as in [79] we also consider during the training session, that when a pixel value $\mathbf{x}$ falls below a 2.5 standard deviation of the distribution, the least probable distribution of the Gaussian sum is replaced by the current RGB pixel value as the current mean, with an initially high variance, and a low prior weight.

5.2.2 Background class

The mixture model on each pixel encodes the distribution of colors for the full image sequence set per full 3D scan (about 153 images). The static portion of the data, i.e., the background, is expected to have large frequency and low variance. By ordering the Gaussians on each sum by the value $\frac{\omega}{\det\Sigma}$, the distributions with larger probability to be of the background class will be aggregated in the top of the list. Static items might however be multimodal in their color. For instance, a flickering screen or a blinking light. As a result we choose as background class the first $B < K$ ordered distributions which add up to a factored weight ω_B, where

$$B = \texttt{argmin}_b(\sum_{i=1}^{b} \omega_i \geq \omega_B) \, . \tag{5.8}$$

5.2.3 Point classification

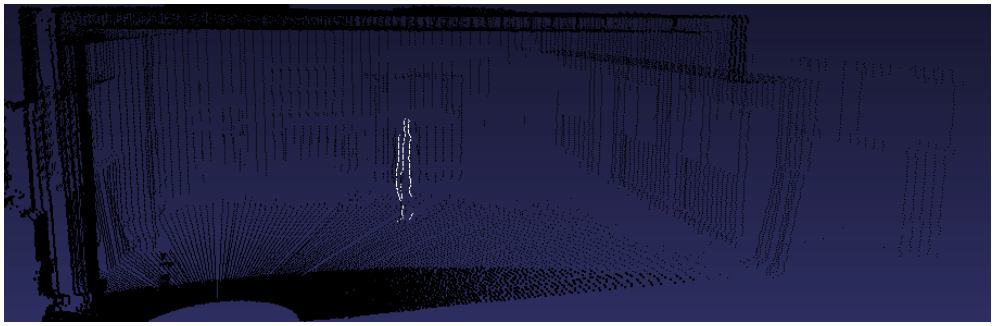
Each point on each scan slice is reprojected to its matching image frames. 3D points are reprojected to the image and classifying according to Equation 5.8. Ideally, for tight bounds on T_s, only one image will be assigned to each scan slice. Robustness to noise is possible however, if this bound is relaxed and we allow for larger values of T_s, so that more than one image can be matched to the same scan slice. We call this set of images I.

Thus for each point in a slice, the corresponding pixel values $\mathbf{x}$ from the whole set I is visited, and checked for inclusion in the set B of distributions. Class assignment is made if $\mathbf{x}$ belongs to B for all the images in the set I.

5.3 Experiments

Results are shown for a series of indoor sequences with moderate dynamic content. For background segmentation, the multimodal distribution is set to contain 4 Guassians, the learning

(a) $T_s = 1/\text{fps}$. (b) $T_s = 0.5\text{sec}$ (c) Foreground segmentation



(d) Segmented dynamic object

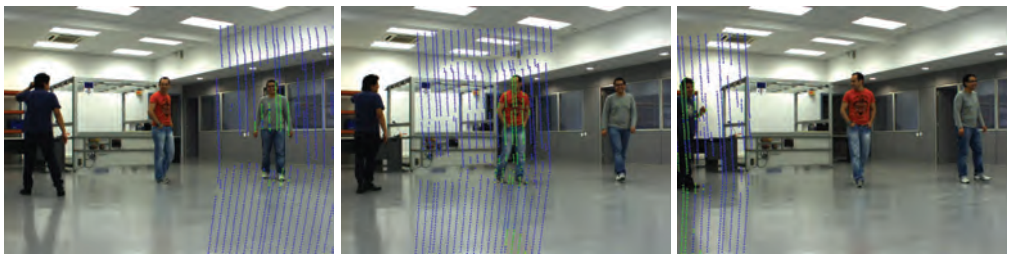
Figure 5.5: Segmentation results for a sequence with one moving person and varying values of the synchronization threshold.

rate is set at $\alpha = 0.3$, and the background class is set to one third of the frequency in the distributions, i.e., $\omega_B = 0.3$. The synchronization threshold T_s is varied from the minimal $1/17$ to a more conservative value of 0.5 seconds.

The first analyzed sequence corresponds to a single person moving in front of the laser and camera. Frames (a) and (b) in Figure 5.5 show final results of point classification for different values of T_s; frame (c) shows the image pixel classification results; and frame (d) shows the 3D reconstruction of both, the segmented dynamic object, and the entire 3D scene.

The second sequence contains a more challenging scenario with three people with slow random walking trajectories. Given the slow motion rate of the people, laser range readings hitting on them are difficult to categorize as being dynamic. The background segmentation algorithm proposed in this chapter helps to alleviate this issue. Figure 5.6 shows results of background segmentation in this new sequence for varying values of the synchronization parameter. Setting this parameter slightly above the camera acquisition rate accounts for synchronization errors and produces better segmentation results. Frames (a-c) in the image show the segmentation results for $T_s = 1/\text{fps}$, whereas frames (d-f) show segmentation results for $T_s = 0.5\text{sec}$.

(a) (b) (c)



(d) (e) (f)

Figure 5.6: Segmentation results for a sequence with three people moving randomly and varying values of the synchronization threshold. Frames (a-c) show three sequence instances segmented at $T_s = 1/\text{fps}$. Frames (d-f) show the same sequence instances segmented at $T_s = 0.5\text{sec}$.

Figure 5.7 shows 3D reconstruction results of the segmented data and of the full 3D scene. The results shown are for a synchronization threshold of 0.5 sec.

We appreciate the suggestion during the peer review phase of the conference version of this work to compare our method with other approaches. Unfortunately, as far as we know, the system presented is unique, and there are no other methods in the literature that take low-rate 3D scans and remove dynamic content from them using high-rate imagery. To validate the approach, we can report however an empirical comparison with ground truth image difference. Assuming a clean background scan is available (without people), image difference to a full dynamic cloud was computed with the Point Cloud Library [66] using a distance threshold of 3mm. Figure 5.8 shows results of such image difference computation. The results of our method are visually comparable to such ground truth experiment. We apply the difference between the computed points cloud with our method (Figure 5.5) and the PCL (Figure 5.8). points cloud the percent of classification is 93.4%.

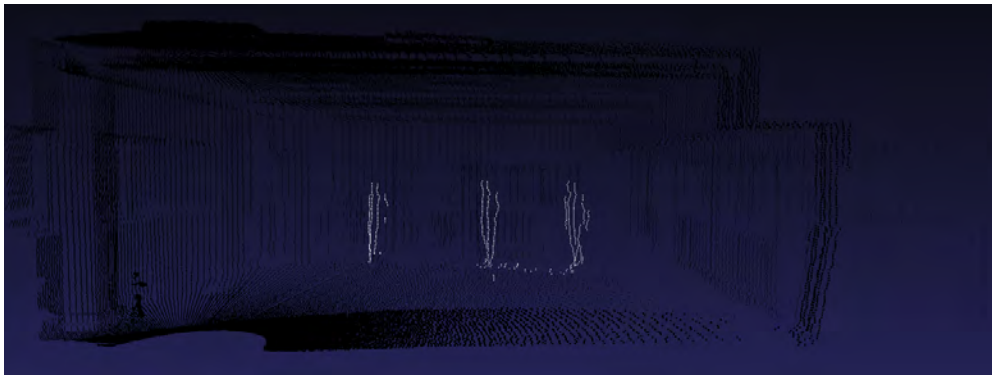


Figure 5.7: Segmentation results for a sequence with three slowly moving people with random walking trajectories.

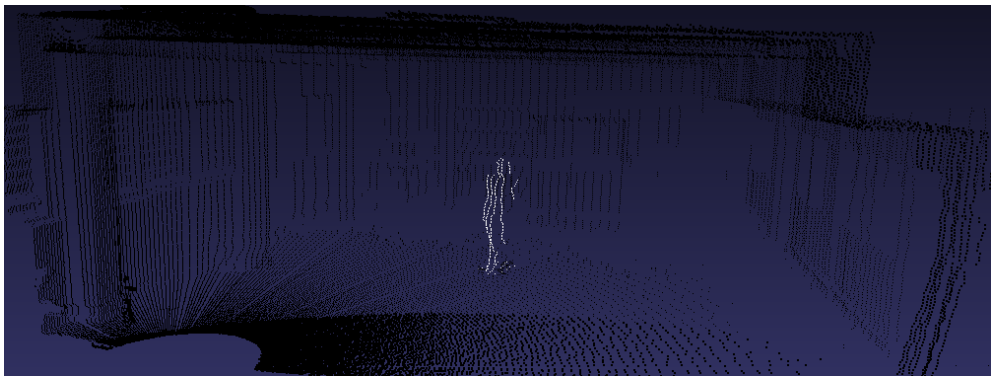


Figure 5.8: Result of applying point cloud difference using PCL.

5.4 Remarks

In this chapter we presented a method to segment low-rate 3D range data as static or dynamic using multimodal classification. The technique classifies fast-rate image data from an accessory camera as background/foreground adapting at frame rate a per-pixel Gaussian mixture distribution. The results of image classification are used to tag reprojected laser data.

Special attention is paid to the synchronization and metric calibration of the two sensing devices. Sensor synchronization is of paramount importance as it allows to match high frame rate imagery with their corresponding low rate laser scans. The method is tested for indoor sequences with moderate dynamics.

The proposed method was designed to remove spurious data or dynamic objects from low acquisition rate lidar sensors. The result is a cleaner 3D picture of static data points. These

point clouds could then be aggregated into larger datasets with the guarantee that dynamic data and noise will not jeopardize point cloud registration. One possible application of the technique is robotic 3D mapping.

Chapter 6

Segmentation of dynamic objects using a high-rate 3D sensor

This chapter presents an extension to the method presented in the previous chapter for the case of high-rate acquisition lasers. The difference between both methods is that in the previous chapter, low acquisition-rate was compensated with the fact that we had a very large number of points per scan, in the order of millions. In this chapter instead, we are dealing with real time range scanning of much lower density levels The segmentation method from the previous chapter has the disadvantage of not being able to recover the true object motion due to the large difference between sensor timestamps. The high-rate of the sensor used in the method described in this chapter allows a significantly better estimate of the object motion.

Dynamic object detection, such as that of humans, is of major importance in many robotics, computer vision, and intelligent vehicle applications [2, 83, 92]. Techniques such as SLAM, require the identification of sensor signatures coming from moving objects. These readings should be pruned out before mapping, otherwise the maps would turn out corrupted with spurious data [28]. Moreover, dynamic object detection allows to have safe navigation avoiding collision [7, 96]. Also, the detection and segmentation of human motion is important for higher level applications such as human-robot interaction [26].

We propose again, the detection of dynamic objects using Gaussian mixtures models (GMM) on intensity and range images, that in this case are generated from a Velodyne $32E-HLD$ laser, a sensor which produces more than one million points per second.

The learned classes are fused to label pixels/voxels as dynamic or static. Once more, we

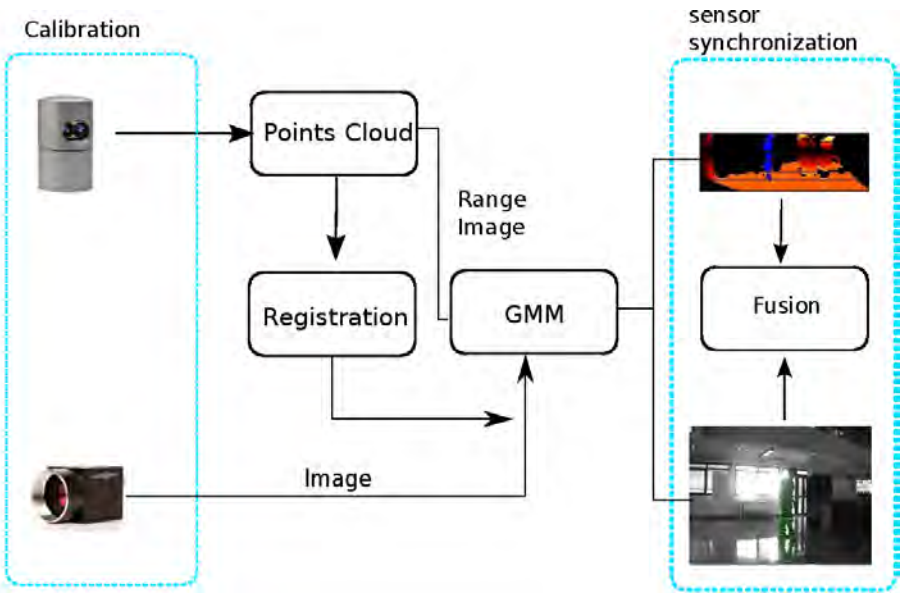


Figure 6.1: Proposed method for dynamic object detection fusing range data and intensity images

pay important attention in sensor synchronization to ensure accurate 3D point cloud-image correspondence. Figure 6.1 shows a block diagram of the proposed method including the modules that are explained in the coming sections.

Other methods that recognize moving people in range date include those implemented for the case of RGB-D sensors [38] that combine the data coming from the time of flight camera together with the image intensity data to create a 3D point cloud annotated with RGB values. In these images, people tracking or gesture recognition can be achieved by analyzing the motion flow vectors resulting from the analysis of a temporal sequence of such point clouds [16, 27], known as scene flow. Scene flow is computed in the abovecited reference using a particle filter that supports multiple hypothesis and that contrary to proir methods, does not oversmooth the motion field during regularization. In [24] fusing the range data coming from a Velodyne sensor with camera imagery via fuzzy rules. The attributes that are related through the fuzzy rules are obstacle size, object class, spatial context, temporal context, and height. The result is an estimate of the class candidate for each dynamic object in the scene. And in [46] object detection estimates are computed from the combination of Velodyne's range data and images by segmenting the images using the GraphCuts method using this segmentation to drive an object classifier on vector quantized binary features of the range data.

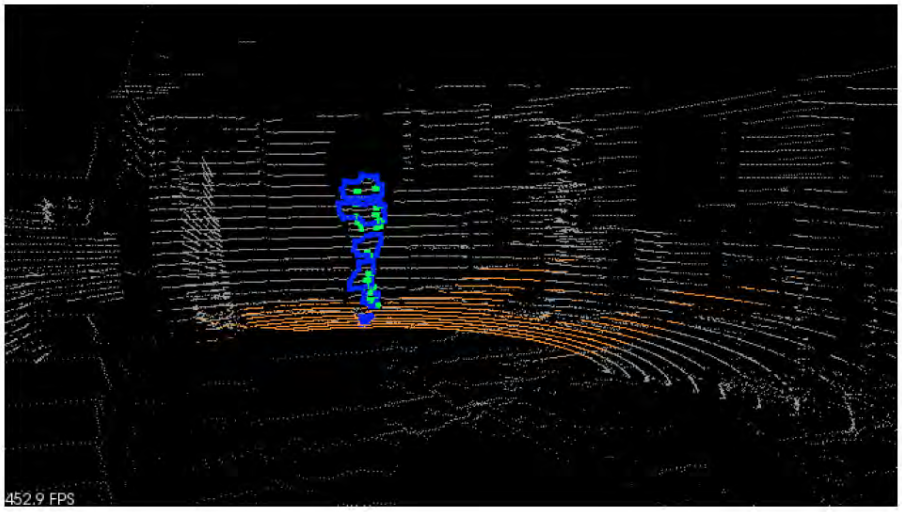


Figure 6.2: Tagged dynamic object on a 3D point cloud.

In our case, Gaussian mixture models are used to encode the moving object classes. The system has two sensors: a Velodyne $32E-HLD$ and a camera. Given that it is necessary to have a very accurate camera-laser synchronization to accurately assign intensity values to the laser 3D points, first, we compute the extrinsic parameters that relate the coordinate frames of the Velodyne and the camera. Then, we create range images from the 3D point clouds to be segmented using GMMs. after that, we apply a similar procedure to segment out objects on the intensity images, also using GMM. To fuse these two estimates for the moving objects, those coming from the range date, and those from the images, we resort to an adaptive mixture of local experts (MLE) architecture [76]. The result is that each fused pixel/voxel is labeled as dynamic or static.

The following sections explain the method in detail. First Section 6.1 provides the sensor specifications used for our experiments, and the methods used to compute the extrinsic laser-camera calibration parameters, and sensor synchronization. Section 6.2 presents the dynamic segmentation scheme for each sensor modality using GMM, and the fusion using MLE. Then, we present our results over an indoor scenario with people moving in random directions in Section 6.3. Finally, in the last Section, we discuss conclusions and future work.

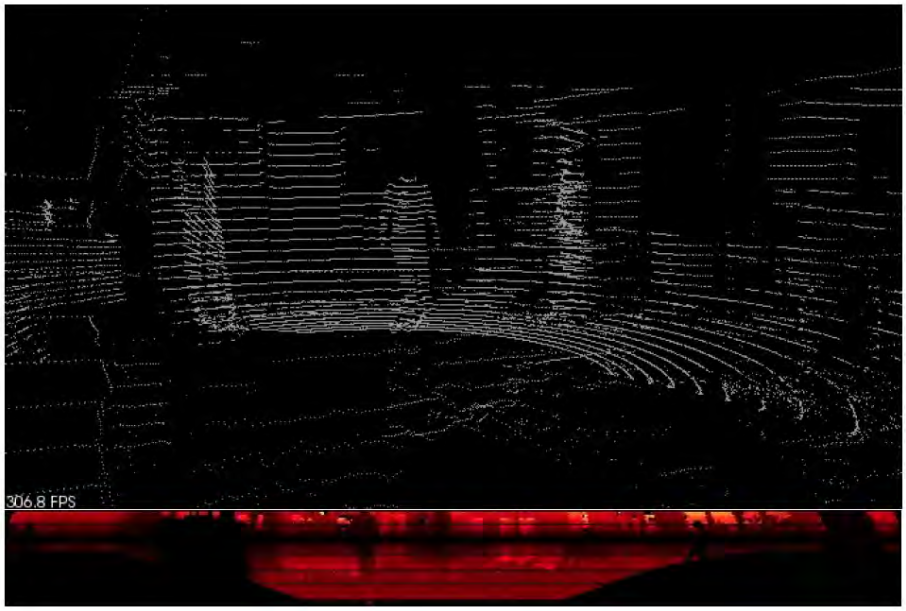


Figure 6.3: (top) Threedimensional view of the scan of a Velodyne laser scanner with approximately 33260 3D points. (Bottom) Range image generated with 0.3 radians of resolution

6.1 Sensor specifications and calibration

The Velodyne $HDL-32E$ laser scanner is composed by a rotating array of 32 range lasers. The lasers are aligned from +10 to -30 degrees allowing a 360 degrees horizontal Field of View (FOV), and with 41.3 degrees of vertical FOV view. The sensor bursts 700,000 points per second with a range of 70 meters and typical accuracy of +/- $2cm$ at a $10Hz$ rate. Figure 6.3 shows a range image created with this sensor with image resolution of 0.3 radians. To gather intensity data, we use a Flea-camera with a frame rate of 30 fps.

We need to calibrate our sensor as was seen in the previous chapter. For the case of the Velodyne, we implemented our calibration routines in ROS and used OpenCV and PCL [1, 9]. The calibration routines are fully automated in this case, linking image coordinates to 3D points in the point cloud. As with Zhang's calibration method [99], the algorithm finds a chess pattern on the 3D point cloud , set in our case to be the closest perpendicular plane in the floor. Then we search for the 3D corners of the selected planar region by estimating the convex hull of the point-annotated plane [17]. We perform a similar procedure at the image to find the chess board corners. This allows us to automatically associate image points in

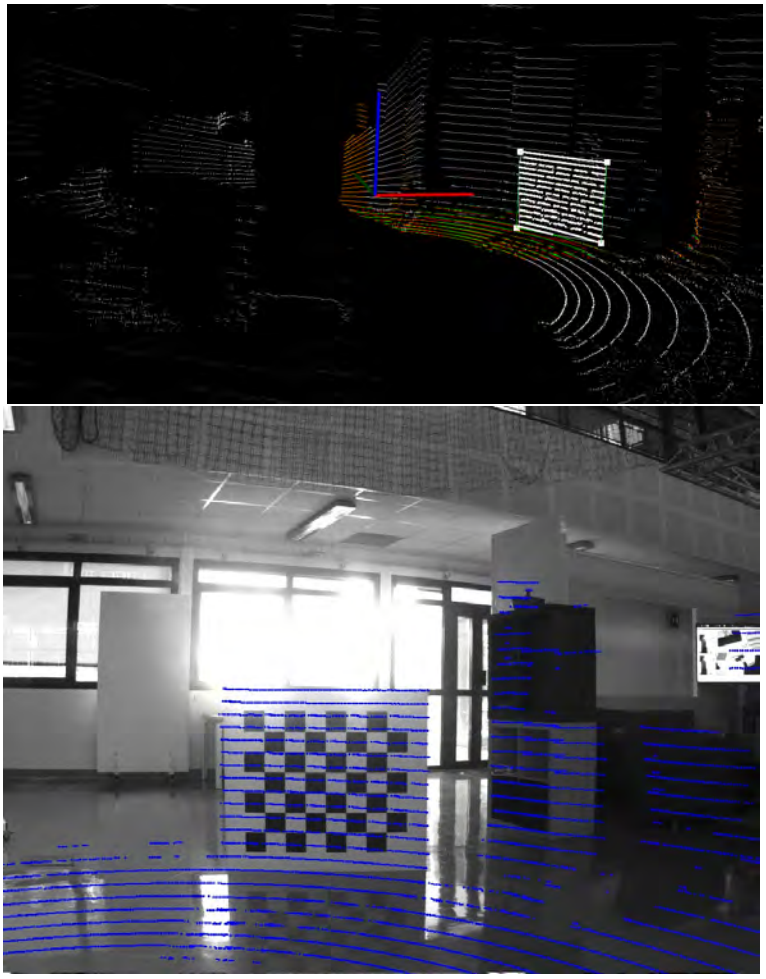


Figure 6.4: Extrinsic laser-camera calibration. (Top) Automatic chess pattern detection, (Bottom) 3D point cloud registration on the image plane.

the chess pattern to their 3D correspondences. Once this is solved, we use Lu's method for pose estimation [37] that assumes an internally calibrated camera and known 3D-2D point correspondences. Extrinsic laser-camera calibration results are shown in Figure 6.4. The image shows 3D points reprojected to the image plane.

We apply a synchronization method named approximate time policy, which is readily available in ROS. This allows to associate data information from different sensors with unsynchronized timestamps. The method simply clusters messages from the various sensors using the last timestamp in the cluster as pivot. Its only user-specified parameter is the largest sensor frame-rate T_s, in this case, that of the laser.

6.2 Data fusion

Once we have computed time correspondences between 3D point clouds and image frames we use background segmentation on the image intensity sequence as explained in the previous chapter, and inspired in [79] to find hypothetical classes of moving objects.

To find hypotheses for the moving objects in the range data, first the whole range image from the laser is cropped to match the camera field of view. Then, the same background substraction algorithm is applied on the cropped range image.

The result is a couple of distributions for the existence of a moving object at each pixel in the whole intensity and range images, $p_I(y|\mathbf{x})$ and $p_R(y|\mathbf{x})$.

To fuse the outputs from the intensity and range object detection modules we develop a method inspired in the adaptive mixture of local experts (MLE) architecture [76] as follows. The fused estimate is a weighted sum of the two beliefs

$$p(y|\mathbf{x},\alpha) = \sum_{i=0}^{i} g_j(\mathbf{x}) p(y|\mathbf{x},\alpha_i) \tag{6.1}$$

where y indicates the object class, $\mathbf{x}$ are corresponding pixel coordinates in both the intensity and range images, g are user selected weighting functions, and α is the learning rate for each of the two sensing modalities.

Figure 6.5 depicts the methods graphically. Note that the pixel indexes may not correspond exactly in both the range and intensity images, but instead relate to each other through the extrinsic calibration of the sensors.

6.3 Experiments

We present the results for two indoor data sequences with dynamic content. We implement our algorithms using ROS [63] for the dynamic segmentation and the Point Cloud Library (PCL) [66] for the laser-camera extrinsic calibration. Multi-modal distribution is set to 4 Gaussians, the learning rate is set at $\alpha = 0.3$, and the background class is set to contain initially one third of the frequency in the distributions, i.e., $\omega_B = 0.3$. We apply similar settings for the range images. The synchronization threshold T_s is varied to the minimal $1/10$ Velodyne frame rate. Finally, range images are generated with horizontal and vertical resolutions of 0.3 radians.

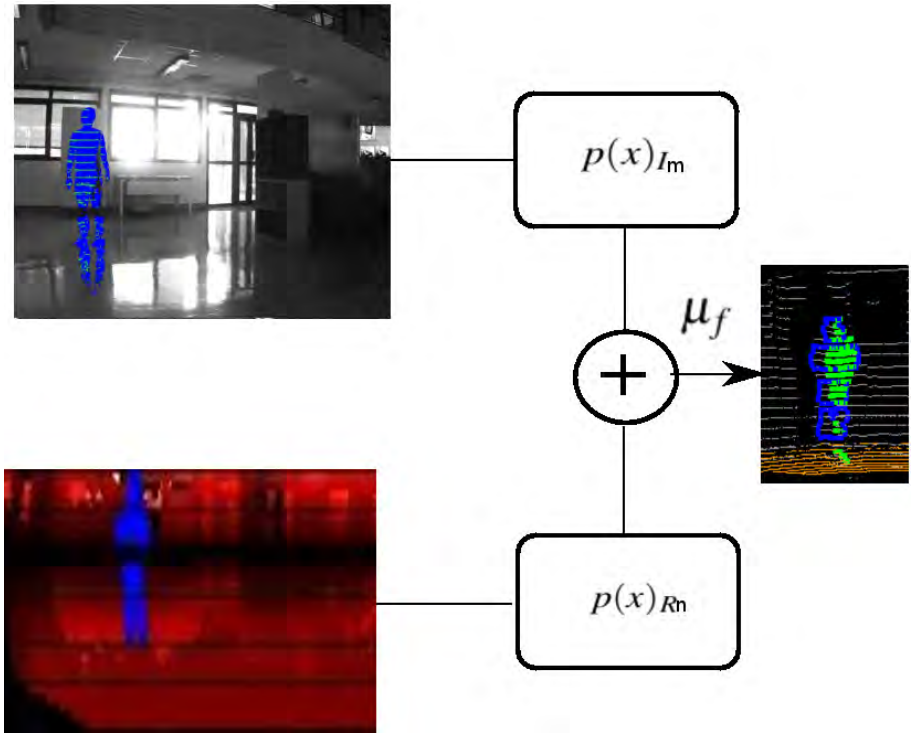
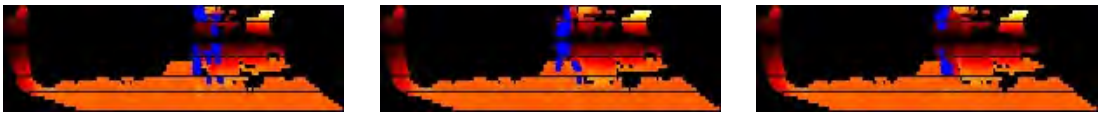


Figure 6.5: Data fusion of range and intensity data using adaptive mixture of local experts.

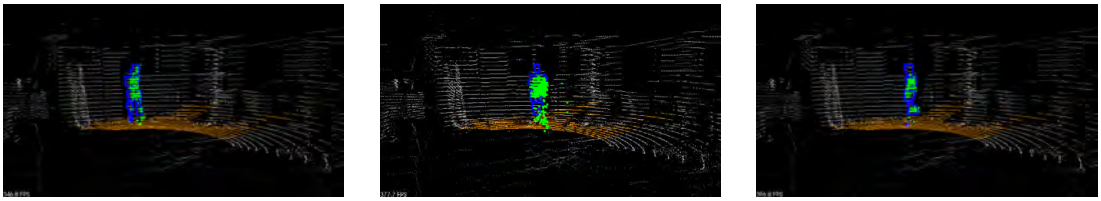
We analyzed first a sequence that corresponds to a single person moving in front of the laser and camera. The results are shown in Figure 6.6. The dynamic regions detected independently in each of the two sensor modalities are reprojected to each image and shown as green dots on the intensity images, and as blue dots on the range images. The adaptive MLE fused results, each with their own contribution, are shown as both green and blue dots in the 3D scene in the last row. Note that 3D false positive classification of points falling on the floor plane due to cast shadows and object reflections can be easily removed thanks to the accurate extrinsic calibration of the sensors.

The second sequence contains people moving with random walking trajectories. We can see that some spurious points are included in the segmentation process, but if needed, these could be easily eliminated by region clustering or with 3D morphological operations.

Point cloud projected to the GMM region



GM over range images



Results of the image and range fusion

Figure 6.6: Result with one object moving in the scene.

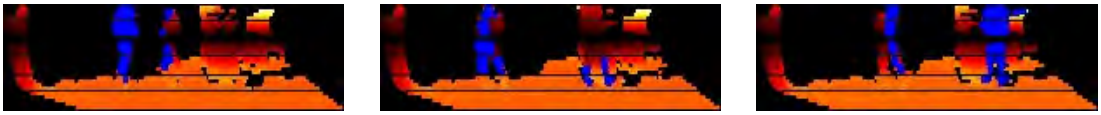
6.4 Remarks

In this chapter we present a method to segment dynamic object on 3D points clouds using a Velodyne scanner and an intensity camera. We segment individually the data from each sensing modality using a Gaussian mixture classifier, and fuse the data of the different classifiers using an adaptive mixture of local experts scheme.

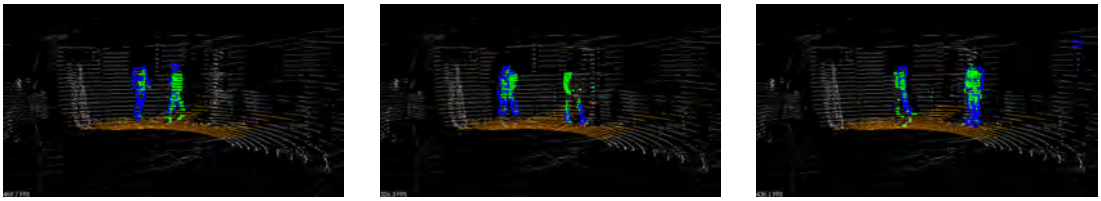
The method pays attention to the importance of extrinsic laser-camera calibration in order to have accurate 3D point registratio. To that end we propose an automatic laser-camera calibration mechanism that finds and matches the corners of a planar convex hull in both the image and the range data. Sensor synchronization also plays an important role in the method to guarantee that the annotation of moving range data matches that of the intensity images.

This method to segment out dynamic objects in a scene may be used to further enhance SLAM or 3D reconstruction methods, by easily eliminating possible outliers.

Point cloud projected to the GMM region



GM over range images



Results of the image and range fusion

Figure 6.7: Result with two objects moving in the scene.

Chapter 7

Conclusions

Multimodal sensing in computer vision and robotics is commonplace. It can be found in many systems such as mobile manipulators, unmanned aerial vehicles, or autonomous cars. When the sensors are chosen appropriately, their complementary properties provide significant added value to what can be achieved with each of the sensors individually. Often however, the fusion of the different sensor modalities occurs at high levels of abstraction, once their outputs have been processed and hypotheses for the detected events or objects need to be reconciled. Throughout the development of this thesis we have realized that in order to fuse multimodal sensor data early on in the detection/recognition pipeline, proper calibration between the various sensors needs to be accomplished.

This calibration of utmost importance for the fusion algorithms to work well, even at the lowest pixel level. Apart from the intrinsic parameters of each sensor, which are often calibrated and provided in the specifications sheets by the sensor manufacturer, there are two more important aspects that need to be considered when calibrating multiple sensors in a holistic system, spatial calibration, and sensor synchronization.

By spatial calibration we mean the geometric relation between two sensors that relates readings from the two into a common reference frame. By synchronization we mean that the one can guarantee that the dynamic events observed by one sensor correspond to the very same events observed by the other.

In Chapter 3 of this thesis we make use of the data coming from one sensor type to aid in the geometric calibration of another one. Specifically, we calibrate the location of a set of cameras in a camera network having small or non existent overlapping fields of view using as a

common reference a 3D map previously built with a range scanning device. In a first stage the user indicates a nominal calibration of the camera network providing intrinsic parameter values from specification sheets, and roughly locating the cameras on an aerial view aligned with the 3D map. Then, the system partially automated matches low level features (lines) in both the images and the 3D map. To aid in the detection of the lines in the 3D map, the segmentation method developed in Chapter 2 is used. Once lines are matched in the two sensor modalities, a cost function is computed measuring the reprojection error of the 3D lines with respect to those on the images. An optimization of this cost function is computed, revising not only the location and orientation of the cameras, but also their intrinsic parameters. The final result is a 3D reconstruction of the camera locations with an accuracy of a few centimeters for an area of about 10000 sqm. This level of accuracy is sufficient for the type of scene analysis applications to which such camera networks are designed for.

To better understand the implications of using the output of one sensor modality for the calibration of another one we must take into account how the sensor noise from the first one propagates to the calibration estimates of the other. This is what we analyzed in Chapter 4, in which we computed such noise propagation using first order models of uncertainty. The analysis was made for the case with null radial distortion. This assumption can be made for low distortion lenses or when images are first rectified, a common procedure to eliminate nonlinear effects of the camera lens. The consistency of the analysis was studied using Monte Carlo simulations, and it showed, for our particular case, that the first order noise propagation model adequately addressed the characteristics of the nonlinear effects introduced by the 2D to 3D reconstruction, the least squares optimization for the calibration parameters and the SVD used to recover the 3D camera pose; all within normal sensor noise ranges.

Chapters 5 and 6 make use of our calibration results for the interpretation of fused data from laser scanners and cameras in the task of dynamic event identification in a scene. The metric calibration between the two sensors developed in the previous chapters proved to be not sufficient for the task at hand, and temporal synchronization between the two sensors played a very important role. In Chapter 5, the laser used was a low-rate high-resolution device, and the temporal synchronization between it and the camera allowed to have each laser stripe to be univocally associated with its corresponding image. Reducing the sensor speed we were able to annotate each image with more than once laser stripe when needed, and increasing the sensor speed we were also able to detect those images in the camera buffer that did not

have an associated laser scan. All in all, the simple synchronization method developed in this chapter was accurate enough for the adequate annotation of events occurring in the scene. In Chapter 6 however, we utilized a high-frequency low-resolution scanner. The higher frame rate of the Velodyne scanner called for a different synchronization solution. In that case, temporal calibration was achieved using an approximate time policy algorithm, a technique that clusters messages using pivoting timestamps. Thanks to these two temporal calibration methods, we were now able to adequately classify the dynamic events occurring in the scene in both cases. In the former case, dynamic event recognition was achieved through background subtraction using a Gaussian mixture model. In the latter case, this did not suffice and we had to approach the data fusion using a mixture of local experts architecture.

In the end, the spatial and temporal calibration methods proposed in this thesis to fuse data from range scanners and cameras allow the adequate treatment of the sensor fusion problem at the lowest level of data processing, that is, at the pixel and point levels, something that is often not possible or overlooked in poorly calibrated systems. The methods developed in this thesis proved useful to solve two difficult tasks in computer vision, the adequate calibration of a large network of cameras with non-overlapping fields of view, and the recognition of dynamic events using either low-rate-high-resolution and high-rate-low-resolution laser scanners.

Appendix A

This Appendix presents two techniques to propagate noise through a possibly nonlinear function. The first one is Monte Carlo simulation in which samples are drawn from the input distribution estimate and are passed through the nonlinear function to compute the parameters of the output distribution. The second one is linear propagation of uncertainties [17], in which instead of sampling the input distribution, one operates directly with its parameters, which in the case of Gaussian distributions are its mean and covariance, to come up with a representation of the output distribution.

A.1 Monte Carlo simulation

Given a function

$$\mathbf{y} = f(\mathbf{x}) \tag{A.1}$$

that relates an input random vector $\mathbf{x}$ to the output random vector $\mathbf{y}$, and given the parameters of the input distribution $\mu_{\mathbf{x}}$ and $\Sigma_{\mathbf{x}}$, we want to find the parameters of the output distribution $\mu_{\mathbf{y}}$ and $\Sigma_{\mathbf{y}}$.

To do so, we generate N samples $\mathbf{x}_i$ from the input distribution, which for the specific case of Gaussian distributions is $\mathcal{N}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}})$. Each sample is passed through the nonlinear function to compute an output data set, $\mathbf{y}_i = f(\mathbf{x}_i)$.

The mean of the output distribution is simply computed with

$$\mu_{\mathbf{y}} = \sum_i \mathbf{y_i}, \tag{A.2}$$

and its second moment or covariance is given by

$$\Sigma_{\mathbf{y}} = \frac{1}{n} \sum_i (\mathbf{y}_i - \mu_{\mathbf{y}})(\mathbf{y}_i - \mu_{\mathbf{y}})^T \tag{A.3}$$

Monte Carlo methods are a reliable technique for the estimation of the first and second order moments of the distribution, however they have the disadvantage of being slow and costly to compute because we have to generate a set of samples from the input distribution, and depending on the size of $\mathbf{x}$, the number N of samples needed to richly represent the whole distribution might be too large to handle.

A.2 First order error propagation

An alternative to Monte Carlo noise propagation is to compute a first order estimation of the output distribution parameters through linearization.

By computing a first order Taylor expansion of Equation A.1, and disregarding the higher order terms, the mean and covariance of the output distribution become

$$\mu_{\mathbf{y}} = f(\mu_{\mathbf{y}}), \tag{A.4}$$

$$\Sigma_{\mathbf{y}} = \mathbf{J}\Sigma_{\mathbf{y}}\mathbf{J}^T \tag{A.5}$$

where $\mathbf{J}$ is the Jacobian of f with respect to $\mathbf{x}$.

There are cases however in which the relation between the input and output variables is unknown, i.e., there is not an explicit form as in Equation A.1 that relates the two random vectors. But often there is an implicit relation between them, of the form $\Phi(\mathbf{x}, f(\mathbf{x})) = \mathbf{0}$.

In that case, we can take advantage of the *Implicit function theorem*:

Theorem 1 "Let $S \subset R^n \times R^m$ be an open set and let $\Phi : S \to R^m$ be a differentiable function. Suppose that $(\mathbf{x}_0, \mathbf{y}_0) \in S$, that $\Phi(\mathbf{x}_0, \mathbf{y}_0) = \mathbf{0}$, and that $\left|\frac{\partial \Phi}{\partial \mathbf{y}}\right|_{(\mathbf{x}_0, \mathbf{y}_0)} \neq \mathbf{0}$, then there is an open neighborhood $X \subset R^n$ of $\mathbf{x}_0$, a neighborhood $\mathbf{Y} \subset R^m$ of $\mathbf{y}_0$, and a unique differentiable function $f : X \to Y$ such that $\Phi(\mathbf{x}, f(\mathbf{x})) = 0$ for all $\mathbf{x} \in X$".

The proof of this theorem can be found in [40]. Taking the derivative by parts of $\Phi(\mathbf{x}, f(\mathbf{x})) = \mathbf{0}$ we have that

$$\frac{\partial \Phi}{\partial\, \mathbf{x}} + \frac{\partial \Phi}{\partial f(\mathbf{x})}\frac{df(\mathbf{x})}{d\mathbf{x}} = \mathbf{0} \quad . \tag{A.6}$$

Solving for the seeked Jacobian $df(\mathbf{x})/d\mathbf{x}$ and substituting $\mathbf{y}$ for $f(\mathbf{x})$ we get

$$\mathbf{J} = -\left(\frac{\partial \Phi}{\partial\, \mathbf{y}}\right)^{-1}\frac{\partial \Phi}{\partial\, \mathbf{x}} \quad . \tag{A.7}$$

Bibliography

[1] *Proceedings of the IEEE ICRA Workshop on Open Source Software in Robotics*, Kobe, 2009.

[2] P. F. Alcantarilla, J. J. Yebes Torres, J. Almazan, and L. M. Bergasa. On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1290–1297, Saint Paul, May 2012.

[3] J. Andrade-Cetto and A. C. Kak. Object recognition. In J. G. Webster, editor, *Wiley Encyclopedia of Electrical and Electronics Engineering*, supplement 1, pages 449–470. John Wiley & Sons, New York, 2000.

[4] J. Andrade-Cetto, A. Ortega, E. Teniente, E. Trulls, R. Valencia, and A. Sanfeliu. Combination of distributed camera network and laser-based 3D mapping for urban service robotics. In *Proceedings of the IEEE/RSJ IROS Workshop on Network Robot Systems*, pages 69–80, Saint Louis, October 2009.

[5] K.O. Arras, O.M. Mozos, and W. Burgard. Using boosted features for the detection of people in 2D range data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3402–3407, Rome, April 2007.

[6] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6):891–923, November 1998.

[7] A. Azim and O. Aycard. Detection, classification and tracking of moving objects in a 3D environment. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 802–807, Alcalá de Henares, June 2012.

[8] A. Del Bimbo, F. Dini, G. Lisanti, and F. Pernici. Exploiting distinctive visual landmark maps in pan-tilt-zoom camera networks. *Computer Vision and Image Understanding*, 114 (6):611–623, June 2010.

[9] G.R. Bradski and A. Kaehler. *Learning OpenCV*. O'Reilly Media, Inc., 1st edition, 2008.

[10] R. A. Brooks. Visual map making for a mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 824–829, St. Louis, March 1985.

[11] S. Cai, Z. Zhao, L. Huang, and Y. Liu. Camera calibration with enclosing ellipses by an extended application of generalized eigenvalue decomposition. *Machine Vision Applications*, 24:513–520, 2013.

[12] J. Chen and B. Chen. Architectural modeling from sparsely scanned range data. *International Journal of Computer Vision*, 78(2-3), 2008.

[13] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, 1992.

[14] A. De la Escalera and J.M. Armingol. Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration. *Sensors*, 10(3):2027–2044, 2010.

[15] F. Endres, C. Plagemann, C. Stachniss, and W. Burgard. Unsupervised discovery of object classes from range data using latent Dirichlet allocation. In *Robotics: Science and Systems V*, Seattle, June 2009.

[16] H. Evan, X. Ren, and D. Fox. RGB-D flow: Dense 3-D motion estimation using color and depth. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2276–2282, Karlsruhe, May 2013.

[17] O. Faugeras. *Three-Dimensional Computer Vision. A Geometric Viewpoint*. The MIT Press, Cambridge, 1993.

[18] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, September 2004.

[19] M. Fiala and C. Shu. Self-identifying patterns for plane-based camera calibration. *Machine Vision Applications*, 19:209–216, 2008.

[20] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–385, 1981.

[21] A. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proceedings of the 15th IEEE Conference on Computer Vision and Pattern Recognition*, pages 125–132, Kauai, December 2001.

[22] D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice-Hall, 2003.

[23] R. Galego, A. Ortega, R. Ferreira, A. Bernardino, J. Andrade-Cetto, and J. Gaspar. Uncertainty analysis of the DLT-lines calibration algorithm for cameras with radial distortion. *Computer Vision and Image Understanding*, 2105. To appear.

[24] Z. Gangqiang, X. Xuhong, and Y. Junsong. Fusion of Velodyne and camera data for scene parsing. In *in Proc. Int. Conf. on Information Fusion*, pages 1172–1179, Singapore, July 2012.

[25] A. Geiger, F. Moosmann, O. Car, and B.Schuster. Automatic camera and range sensor calibration using a single shot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3936–3943, Saint Paul, May 2012.

[26] JM. Gottfried, J. Fehr, and C. Garbe. Computing range flow from multi-modal Kinect data. In *Proceedings of the 6th International Symposium on Visual Computing*, volume 5875 of *Lecture Notes in Computer Science*, pages 758–767, Las Vegas, September 2011.

[27] S. Hadfield and R. Bowden. Kinecting the dots: particle based scene flow from depth sensors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2290–2295, Barcelona, November 2011.

[28] D. Hahnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1557–1563, Taipei, September 2003.

[29] R.M. Haralick. Propagating covariance in computer vision. In *Proceedings of the 13th IAPR International Conference on Pattern Recognition*, volume 1, pages 493–498, Vienna, August 1996.

[30] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2nd edition, 2004.

[31] V. Ila, J. Andrade-Cetto, R. Valencia, and A. Sanfeliu. Vision-based loop closing for delayed state robot mapping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3892–3897, San Diego, November 2007.

[32] I.N. Junejo and H. Foroosh. GPS coordinates estimation and camera calibration from solar shadows. *Computer Vision and Image Understanding*, 114(9):991–1003, September 2010.

[33] I.N. Junejo, C. Xiaochun, and H. Foroosh. Autoconfiguration of a dynamic nonoverlapping camera network. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 37(4):803–816, August 2007.

[34] R. Kumar and A. R. Hanson. Sensitivity of the pose refinement problem to accurate estimation of camera parameters. In *Proceedings of the IEEE International Conference on Computer Vision*, Osaka, Japan, December 2000.

[35] L. Liu and I. Stamos. A systematic approach for 2D-image to 3D-range registration in urban environments. *Computer Vision and Image Understanding*, 116(1):25–37, 2012.

[36] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun. Using EM to learn 3D models of indoor environments with mobile robots. In *Proceedings of the 18th International Conference on Machine Learning*, pages 329–336, Williamstown, July 2001.

[37] C.P. Lu, G.D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:610–622, 2000.

[38] M. Luber, L. Spinello, and K. O. Arras. Learning to detect and track people in RGB-D data. In *Proc. of The Workshop on RGB-D Cameras (RSS)*, 2011.

[39] H. Lütkepohl. *Handbook of Matrices*. Wiley and Sons, 1996.

[40] J.E. Marsden. *Elementary Classical Analysis*. W.H. Freeman and Company, 1974.

[41] F. Maurelli, D. Droeschel, T. Wisspeintner, S. May, and H. Surmann. A 3D laser scanner system for autonomous vehicle navigation. In *Proceedings of the 14th International Conference on Advanced Robotics*, Munich, June 2009.

[42] A. Mavrinac, X. Chen, and K. Tepe. An automatic calibration method for stereo-based 3D distributed smart camera networks. *Computer Vision and Image Understanding*, 114 (8):952–962, August 2010.

[43] P. Merchán, A. Adán, S. Salamanca, V. Domínguez, and R. Chacón. Geometric and colour data fusion for outdoor 3D models. *Sensors*, 12(6):6893–6919, 2012.

[44] C. Mertz, L. E. Navarro-Serment, R. MacLachlan, P. Rybski, A. Steinfeld, A. Suppé, C. Urmson, N. Vandapel, M. Hebert, C. Thorpe, D. Duggins, and J. Gowdy. Moving object detection with laser scanners. *Journal of Field Robotics*, 30(1):17–43, 2013.

[45] F.M. Mirzaei, D.G. Kottas, and S.I. Roumeliotis. 3D LIDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization. *International Journal of Robotics Research*, 31(4):452–467, 2012.

[46] S. Mohottala, S. Ono, M. Kagesawa, and K. Ikeuchi. Fusion of a camera and a laser range sensor for vehicle recognition. In *Proceedings of the IEEE CVPR Workshops*, pages 16–23, Florida, June 2009.

[47] F. Moosmann, O. Pink, and C. Stiller. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 215–220, Xian, June 2009.

[48] E. M. Mouaddib and B. Marhic. Geometrical matching for mobile robot localization. *IEEE Transactions on Robotics and Automation*, 16(5):542–552, October 2000.

[49] P. Nú nez, P. Drews Jr, R. Rocha, and J. Dias. Data fusion calibration for a 3D laser range finder and a camera using inertial data. In *Proceedings of the European Conference on Mobile Robotics*, Dubrovnik, September 2009.

[50] K. Okuma, J.J. Little, and D. G. Lowe. Automatic rectification of long image sequences. In *Proceedings of the Asian Conference on Computer Vision*, Jeju Island, January 2004.

[51] E. Olson. A passive solution to the sensor synchronization problem. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1059–1064, Taipei, October 2010.

[52] A. Ortega and J. Andrade-Cetto. Segmentation of dynamic objects from laser data. In *Proceedings of the European Conference on Mobile Robotics*, pages 115–121, Orebro, September 2011.

[53] A. Ortega and J. Andrade-Cetto. Dynamic object detection fusing LIDAR data and images. In *Proceedings of the X Taller de Procesamiento de Imágenes*, 2014.

[54] A. Ortega, B. Dias, E. Teniente, A. Bernardino, J. Gaspar, and Juan Andrade-Cetto. Calibrating an outdoor distributed camera network using laser range finder data. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 303–308, Saint Louis, October 2009.

[55] A. Ortega, I. Haddad, and J. Andrade-Cetto. Graph-based segmentation of range data with applications to 3D urban mapping. In *Proceedings of the European Conference on Mobile Robotics*, pages 193–198, Dubrovnik, September 2009.

[56] A. Ortega, M. Silva, E.H. Teniente, R. Ferreira, A. Bernardino, J. Gaspar, and J. Andrade-Cetto. Calibration of an outdoor distributed camera network with a 3D point cloud. *Sensors*, 14(8):13708–13729, 2014.

[57] Agustin Ortega and Juan Andrade-Cetto. Segmentation of dynamic object from low acquisition rate range data. In *2nd ENS/INRA Visual Recognition and Machine Learning Summer School*, Paris, July 2011.

[58] A. Ortega-Jimenez, R. Galego, R. Ferreira, A. Bernardino, J. Gaspar, and J. Andrade Cetto. Estimation of camera calibration uncertainty using lidar data. In *Proceedings of the European Conference on Mobile Robotics*, pages 361–366, Barcelona, September 2013.

[59] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak. Fast plane detection and polygonalization in noisy 3D range images. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3378–3383, Nice, Sep. 2008.

[60] I. Posner, D. Schroeter, and P. Newman. Describing composite urban workspaces. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4962–4968, Rome, April 2007.

[61] I. Posner, M. Cummins, and P. Newman. Fast probabilistic labeling of city maps. In *Robotics: Science and Systems IV*, Zurich, June 2008.

[62] I. Posner, D. Schroeter, and P. Newman. Online generation of scene descriptions in urban environments. *Robotics and Autonomous Systems*, 56(11):901–914, 2008.

[63] M. Quigley, B. Gerkey, K. Conley, T. Foote J. Faust, J. Leibs, E. Berger, R. Wheeler, and A.Y. Ng. ROS: An open-source robot operating system. In *Proceedings of the IEEE ICRA Workshop on Open Source Software in Robotics* icr [1].

[64] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Proceedings of the 18th IEEE Conference on Computer Vision and Pattern Recognition*, pages 187–194, Washington, July 2004.

[65] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. In *Proceedings of the IASTED International Conference on Robotics and Manufacturing*, pages 193–199, Istanbul, August 1995.

[66] R.B Rusu and S. Cousins. 3D is here: Point cloud library (PCL). In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1–4, Shanghai, May 2011.

[67] D. Samper, J. Santolaria, F. J. Brosed, A. C. Majarena, and J. J. Aguilar. Analysis of Tsai calibration method using two- and three-dimensional calibration objects. *Machine Vision Applications*, 24:127–131, 2013.

[68] A. Sanfeliu, A. Grau, J. Climent, R. Alquézar, F. Serratosa, J. Aranda, J. Vergés, and J. Andrade. Pattern recognition research at the IRI-CSIC/ESAII group. In A. Grau, editor, *Proceedings of the 1st Foro Iberoamericano de Reconocmiento de Formas y Análisis de Imágenes*, pages 345–354, Barcelona, September 2000. International Association for Pattern Recognition, UPC Information, Image and Publication Service.

[69] A. Sanfeliu, J. Andrade-Cetto, M. Barbosa, R. Bowden, J. Capitán, A. Corominas Murtra, A. Gilbert, J. Illingworth, L. Merino, Josep M. Mirats Tur, P. Moreno, A. Ollero, J. Sequeira, and M.T. Spaan. Decentralized sensor fusion for ubiquitous networking robotics in urban areas. *Sensors*, 10(3):2274–2314, March 2010.

[70] D. Scaramuzza, A. Harati, and R. Siegwart. Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4164–4169, San Diego, November 2007.

[71] S. Schneider, M. Himmelsbach, T. Luettel, and H.J. Wuensche. Fusing vision and lidar - synchronization, correction and occlusion reasoning. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 388–393, San Diego, June 2010.

[72] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. Tracking multiple moving targetswith a mobile robot using particle filters and statistical data association. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1665–1670, Seoul, May 2001.

[73] S-W. Shih, Y-P. Hung, and W-S. Lin. Accuracy analysis on the estimation of camera parameters for active vision systems. In *Proceedings of the 13th IAPR International Conference on Pattern Recognition*, volume 1, pages 930–935, Vienna, August 1996.

[74] M. Silva, R. Ferreira, and J. Gaspar. Camera calibration using a color-depth camera: Points and lines based DLT including radial distortion. In *Proceedings of the IEEE/RSJ IROS Workshop on Color-Depth Camera Fusion in Robotics*, Vilamoura, October 2012.

[75] K-T Song and J-Ch Tai. Dynamic calibration of pan-tilt-zoom cameras for traffic monitoring. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 36 (5):1091–1103, 2006.

[76] L. Spinello and K. O. Arras. Leveraging RGB-D data: Adaptive fusion and domain adaptation for object detection. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4469–4474, Saint Paul, May 2012.

[77] I. Stamos and P.K. Allen. 3D model construction using range and image data. In *Proceedings of the 14th IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 1531, Hilton Head, SC, June 2000.

[78] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the 13th IEEE Conference on Computer Vision and Pattern Recognition*, pages 246–252, Fort Collins, June 1999.

[79] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[80] H. Surmann, A. Nuchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45(3-4):181–198, 2003.

[81] T. Svoboda and P. Sturm. A badly calibrated camera in ego-motion estimation, propagation of uncertainty. In *Proceedings of the International Conference on Computer Analysis of Images and Patterns*, volume 1296 of *Lecture Notes in Computer Science*, pages 183–190, 1997.

[82] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence:Teleoperation and Virtual Environments*, 14(4):407–422, 2005.

[83] A. Teichman and S. Thrun. Practical object recognition in autonomous driving and beyond. In *Proceedings of the IEEE Workshop on Advanced Robotics and its Social Impacts*, pages 35–38, Half-Moon Bay, CA, October 2011.

[84] E.H. Teniente, M. Morta, A. Ortega, E. Trulls, and J. Andrade-Cetto. Barcelona Robot Lab data set, 2011.

[85] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, 2005.

[86] R. Triebel, W. Burgard, and F. Dellaert. Using hierarchical EM to extract planes from 3D range scans. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4437–4442, Barcelona, April 2005.

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

[88] N. Ukita. Probabilistic-topological calibration of widely distributed camera networks. *Machine Vision Applications*, 18:249–260, 2007.

[89] R. Unnikrishnan and M. Hebert. Fast extrinsic calibration of a laser rangefinder to a camera. Technical Report CMU-RI-TR-05-09, Robotics Institute, Pittsburgh, July 2005.

[90] R. Valencia, E.H. Teniente, E. Trulls, and J. Andrade-Cetto. 3D mapping for urban service robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3076–3081, Saint Louis, October 2009.

[91] A. J. Vayda and A. C. Kak. A robot vision system for recognition of generic shaped objects. *Computer Vision and Image Understanding*, 54(1):1–46, July 1991.

[92] M. Villamizar, J. Andrade-Cetto, A. Sanfeliu, and F. Moreno-Noguer. Bootstrapping boosted random ferns for discriminative and efficient object classification. *Pattern Recognition*, 45(9):3141–3153, 2012.

[93] Michael Villamizar, Helmut Grabner, Francesc Moreno-Noguer, Juan Andrade-Cetto, Luc Van Gool, and Alberto Sanfeliu. Efficient 3D object detection using multiple pose-specific classifiers. In *Proceedings of the British Machine Vision Conference*, pages 20.1–20.10, Dundee, August 2011.

[94] R.G. von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010.

[95] C-C. Wang, C. Thorpe, and S. Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 842–849, Taipei, September 2003.

[96] D.Z. Wang, I. Posner, and P. Newman. What could move? finding cars, pedestrians and bicyclists in 3D laser data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4038–4044, Saint Paul, May 2012.

[97] T. Yokoya, T. Hasegawa, and R. Kurazume. Calibration of distributed vision network in unified coordinate system by mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1412–1417, Pasadena, May 2008.

[98] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2301–2306, Sendai, September 2004.

[99] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.