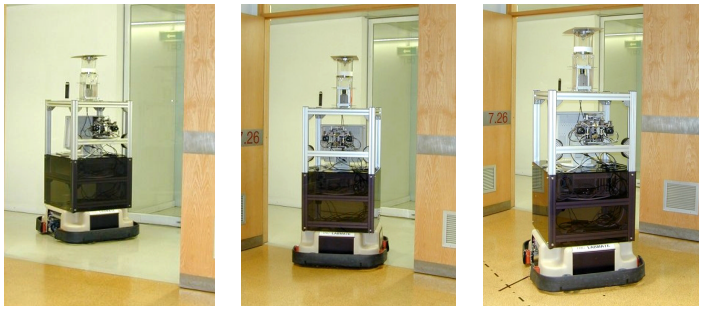
UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO



Omnidirectional Vision for Mobile Robot Navigation

José António da Cruz Pinto Gaspar
(Mestre)

Dissertação para a obtenção do Grau de Doutor em
Engenharia Electrotécnica e de Computadores

Orientador: Doutor José Alberto Rosado dos Santos Victor

Júri:

Presidente:	Reitor da Universidade Técnica de Lisboa
Vogais:	Doutor João José dos Santos Sentieiro
	Doutor Helder de Jesus Araújo
	Doutora Maria Isabel Lobato de Faria Ribeiro
	Doutor José Alberto Rosado dos Santos Victor
	Doutor Kostas Daniilidis
	Doutor Pedro Manuel Urbano de Almeida Lima

Dezembro de 2002

Aos meus pais
e irmã.

Agradecimentos

Um doutoramento envolve certamente numerosos apoios e valiosas contribuições de muitas pessoas, sendo portanto impossível fazer uma secção de agradecimentos completa. Em muitos casos não tenho sequer palavras para expressar a minha gratidão e confio simplesmente que as palavras que ficam por dizer, surgirão naturalmente em futuros encontros.

Começo por agradecer a José Santos-Victor, orientador desta tese, as inúmeras sugestões e o apoio constante durante todo o trabalho. Encontrámo-nos pela primeira vez quando eu estava ainda a realizar a minha licenciatura. Na altura visão por computador era somente uma área promissora. Foi uma experiência gratificante estar envolvido desde o princípio e crescer cientificamente no nosso laboratório de visão. A aposta na visão foi definitivamente uma aposta ganha.

Ao Professor João Sentieiro quero agradecer a criação das condições e de um ambiente excelente de investigação. Quero também agradecer as sugestões e indicações nos meus primeiros passos como assistente. Ao Professor Eduardo Morgado quero agradecer o espelhamento excelente do nosso espelho esférico: depois de todos estes anos de operação o espelho continua perfeito para realizar mais experiências.

Agradeço de forma especial ao Niall, um amigo de muitas viagens, a colaboração próxima e o entusiasmo na investigação em visão omnidireccional (*and, long live to U2!*). Etienne, obrigado pelas numerosas argumentações, debates e apoio na obtenção de reconstruções de modelos tridimensionais (*et vive le D'Hilbert!*). Alexandre, colega de gabinete e amigo desde longa data, obrigado pelo continuo debate, pelas inúmeras sugestões e pelos exemplos de vida! Nuno Gracias, obrigado pela discussão de ideias, por todas as organizações de eventos que realizaste e pelo companheirismo na *empresa das danças de salão*. A vós quero agradecer também a leitura cuidadosa de vários capítulos da tese. Muitas melhorias resultaram das vossas sugestões.

Ao João Paulo Costeira agradeço as várias discussões animadas e energéticas. Maciel e César, obrigado pelos pontos de vista positivos e divertidos do *universo e tudo o resto*. António, és uma referência de boa disposição e de infinita sensibilidade para problemas de computador! Carlos Carreira, obrigado pelas várias ideias para a construção do primeiro aparato experimental. Sjoerd, obrigado pelo optimismo bem informado. Vítor Costa, obrigado pela amizade e pela ajuda continua nas actualizações informáticas. À Cláudia, à Raquel e à Klaudia, tenho de agradecer não só o apoio e as contribuições na visão omnidireccional e na modelação 3D, mas também os cursos concentrados de *samba*, *forró*

e *cultura polaca*. Roger, Plínio, Manuel e Ricardo, apesar de serem os membros mais novos da equipa, são também desde já fontes de discussões interessantes e de novas experiências.

Agradeço aos meus colegas da secção de Sistemas e Controlo, Francisco Garcia, Pedro Aguiar e João Sanches, o esforço que fizeram para me ajudar a organizar o tempo para a escrita da tese. Foi sem dúvida importante.

O diálogo com colegas de áreas aparentemente distantes ou vocacionadas para sistemas muito diferentes, proporciona muitas vezes direcções de trabalho novas, interessantes e profícuas. Neste caso, devo agradecer as numerosas e valiosas conversas com António P. Aguiar, Pedro Encarnação, João Mota, Artur Arsénio e em geral aos meus colegas do ISR, João Pedro, João Xavier, Sebastião, Alves, Paulo Oliveira, Rodrigo, Carlos Marques,... E claro, não esqueço a ajuda da Filomena, do Nuno, do Rafael, do Loïc, da Maria, da Anastácia, do Sr. Serralha e da D. Eduarda nos procedimentos administrativos e de montagens materiais. Obrigado por todos aqueles trabalhos urgentes. Não é possível agradecer a todos amigos e colegas do IST e do ISR, mas aqui eu postulo um novo *axioma* e agradeço-vos a todos.

Devo também agradecimentos especiais aos meus amigos de longa data Paulo Rogério e Luís Filipe. Cruzámo-nos pela primeira vez há 18 anos, quando começámos na mesma escola os estudos em electrónica. Depois de todos estes anos continuamos a partilhar experiências e soluções construtivas. Ao Constantino agradeço as novidades sempre interessantes que trás dos Estados Unidos. Francisco M. C. S. Moreira, sim é verdade que os bons princípios e as boas causas merecem os maiores esforços! Aos meus colegas de ténis de mesa, quero agradecer a ajuda na manutenção de uma combinação saudável de vida à volta de computadores/robots vs *mais vida* (e também a afinação do meu *top-spin*).

Ao Instituto Superior Técnico e ao Instituto de Sistemas e Robótica, agradeço a dispensa de serviço docente e o apoio material para a realização do trabalho de investigação. Este apoio é certamente crucial para o desenvolvimento de programas de doutoramento, e sem dúvida a minha experiência confirma-o.

Finalmente, quero expressar os meus mais profundos agradecimentos aos meus pais pela fonte constante de apoio ao longo de todos estes anos. À minha irmã agradeço a lembrança continuada de que existe sempre um ponto de vista positivo e um caminho construtivo para os problemas mais complicados.

J. A. Gaspar

Acknowledgements

The work of a PhD is definitely an impossible mission without lots of help and therefore it is almost unfeasible to thank everyone. In many cases I do not even have the words to express all my gratitude. Even so, let me start and try, knowing that this section is finite but the future occasional meetings shall have almost infinite power for completing the task.

I would like to thank to José Santos-Victor, my supervisor, for his many suggestions and constant support during this research. We first met many years ago, while I was still working for my graduation. At that time vision was just a promising research area. It has been a grateful experience being involved from the beginning and see the extraordinary growth of our vision laboratory. Constantly betting on vision was definitely a rewarding direction.

To Professor João Sentieiro I want to thank for creating such an excellent research environment and the help in my first steps as a teaching assistant. To Professor Eduardo Morgado I thank the perfect (surface) mirror coating of our spherical mirror: it is still effective for many more experiments.

Special thanks are due to Niall, a friend on many trips and my closest colleague in omnidirectional vision research (and, long live to U2!). Etienne, thanks for all the valuable discussion and support for obtaining the reconstruction results (et vive le D'Hilbert!). Alexandre, office colleague and a long time friend, thanks for the continuous debate, the numerous suggestions and the examples of life! Nuno Gracias, the best *event manager* I know and also a companion of the ballroom dance classes, thanks for the discussion of ideas. I must thank you all also the careful reading of the chapters of the thesis. Many improvements resulted from your suggestions.

To João Paulo Costeira, thanks for the animated and energized discussions. To Maciel and César, thanks for the positive and humorous views of the *universe and everything*. António, you are a reference on the best mood and infinite sensibility for computer problems! Carlos Carreira, thanks for the many ideas for building the first setup. Sjoerd, thanks for the *well informed optimism*. Vítor Costa, thanks for the friendship and the continuous support in the computer hardware updates. To Cláudia, Raquel and Klaudia, I have to thank not only advances on omnidirectional vision and 3D modelling, but also the *crash* courses on *samba*, *forró* and *polish culture*. Roger, Plínio, Manuel and Ricardo are the most recent team members, and already a source of discussion and new experiences.

I thank to my teaching colleagues, Francisco Garcia, Pedro Aguiar and João Sanches, the efforts made to help me organise the time for the writing of the thesis. It was definitely helpful.

Talking with colleagues involved in research areas that seem at a first sight unrelated or directed towards too different setups, turns out many times to motivate very helpful working directions. In this line, I must thank António P. Aguiar, Pedro Encarnação, João Mota, Artur Arsénio and in general to my ISR colleagues, João Pedro, João Xavier, Sebastião, Alves, Paulo Oliveira, Rodrigo, Carlos Marques, ... And of course, I do not forget the help of Filomena, Nuno, Rafael, Loïc, Maria, Anastácia, Sr. Serralha and D. Eduarda in the paper and hardware work. Many thanks for helping on those always urgent works. It is not possible to thank to all my IST and ISR friends and colleagues, but here I postulate a new *axiom* and I thank to you all.

Special thanks are due also to my old friends Paulo Rogério and Luís Filipe. We met 18 years ago when we were starting our studies on electronics. After all these years we still discuss our experiences and share constructive solutions. To Constantino I thank the hot news directly from US. Francisco Moreira, yes it is true that good principles and causes deserve good fights! To my table-tenis fellows I thank for helping to maintain a healthy combination of computer/robots based life vs other life (and tuning up my *top-spin*).

To Instituto Superior Técnico and to the Institute for Systems and Robotics, I want to thank the leave from teaching and the resources given to pursue my research. This support is certainly crucial for the continuous development of doctoral programs, and my own experience confirms it.

Finally, I wish to express my deepest gratitude to my parents for the constant source of support all over these years. My sister I thank for reminding me that there is always a positive view and a constructive approach to the hardest problems.

J. A. Gaspar

Resumo

A investigação realizada no âmbito da tese, versa a navegação visual de robots móveis em ambientes interiores, com ênfase especial nos aspectos de desenho do sensor, de representação do ambiente, de auto-localização e da interacção com pessoas. O ponto principal é que da exploração dos aspectos individuais, de uma forma combinada, resulta um sistema de navegação eficiente.

O desenho do sensor permite criar representações do ambiente úteis para a auto-localização. A partir de uma única câmara omnidireccional, estimamos de forma precisa / qualitativa a auto-localização e construímos um sistema de acordo com as tarefas específicas de navegação.

Do estudo da geometria de câmaras omnidireccionais baseadas em espelhos esféricos, resulta que em muitas aplicações é negligenciável o erro associado à inexistência de centro de projecção único. Resultam também transformações da imagem que representam em ortografia o plano do pavimento. Estas representações são úteis para a navegação.

A interface com pessoas é desenhada com o objectivo de permitir a selecção intuitiva de localizações a atingir pelo robot. Os modelos gerados descrevem de forma rica a cena, e podem ser observados em poses arbitrárias. A reconstrução a partir de uma única imagem é possível com a colaboração do utilizador que fornece algumas propriedades de co-linearidade e co-planaridade presentes na cena.

Os algoritmos apresentados na tese são validados experimentalmente, mostrando-se em várias situações que é obtida navegação precisa. A visão omnidireccional revela-se útil por exemplo em tarefas de parqueamento e de passagem de portas. São também apresentadas experiências de navegação em percursos longos realizadas pela combinação das metodologias propostas. As metodologias desenvolvidas facilitam a operação autónoma ou teleguiada e a interacção com robots móveis.

Palavras Chave

Visão omnidireccional, Navegação de robots, Interfaces homem máquina, Reconstrução interactiva.

Abstract

The research, described in the thesis, concerns the visual navigation of indoor robots, emphasising the aspects of sensor design, environmental representations, accurate self-localisation and interaction with humans. The main point is that by exploring these different aspects, in a combined manner, an effective navigation system is obtained.

Sensor design is an enabling key for creating environmental representations adequate for accurate localisation. We present a system capable of self-localising using only a single omnidirectional camera, explicitly taking into account the nature of navigation task at hand in the design process.

We detail the geometry associated with omnidirectional cameras using spherical mirrors. We show that minimal error is induced by not having a single centre of projection. Methods used to obtain the bird's eye (orthographic) view of the ground are presented. This representation significantly simplifies the solution to navigation problems, by eliminating any perspective effects.

In order to achieve effective interaction with humans, we provide an intuitive user interface for target selection, built from an omnidirectional image. The models generated provide a rich scene description, which the user is free to rotate and translate. Reconstruction from a single image is possible with limited user input in the form of co-linearity or co-planarity properties.

We provided real world experimental results showing that our algorithms achieve highly precise navigation in several situations. Omnidirectional vision is shown to be beneficial for such tasks as docking and door traversal. Combined experiments, involving long distance navigation are also detailed.

The developed methodologies facilitate autonomous or guided exploration (tele-operation) and human-robot interaction.

Keywords

Omnidirectional Vision, Navigation, Human-robot interfaces, Interactive Reconstruction.

Contents

Chapter 1

Introduction

> *"... affixed to the wall on the left are printed posters : 'The Cheapest Labor: Rossum's Robots.' 'Tropical Robots - A New Invention - $150 a Head.' 'Buy Your Very Own Robot.' 'Looking To Cut Production Costs? Order Rossum's Robots.' "*
> *From the prolog of the play* R.U.R. (Rossum's Universal Robots) *by Karel Capek, 1920.*

In the last few decades many robots have been developed, produced and installed mainly for industrial applications. Despite original concerns that robots would replace human labor, and in the process create large unemployment problems, the reverse was that robots become human extensions. In the car industry, for example robots hold and rotate the cars for humans to inspect or to operate upon some localised regions.

Industrial robots are essentially robot arms, teleoperated or programmed for repetitive tasks. Mobile robots appeared more recently. The scientific and technical challenges are larger, mainly requiring autonomy without posing a risk to people or to the robots themselves. One very exciting application of mobile robots is in planetary exploration. Planetary exploration is much too expensive and potentially hazardous for human intervention. There are also environments in which it is possible for humans to intervene, but a robotic intervention could be less costly. For example, service robots have been introduced in hospitals for transporting medicines and therefore saving time to nursing staff [60].

The technical developments of the last decade make it possible to manufacture service robots for everyday indoor applications. However, the sensing modalities are still too limited, expensive or difficult to install, while requiring autonomy for operation in large environments.

In order to introduce robots in the human society, it is necessary to consider the (social) interactions of the robots to other robots and to humans. This implies building autonomous robots that share the sensed data and models. Sharing sensed data and models is important for establishing communication languages, where the robot behaviour is simple to understand and to specify. Autonomy is relevant for keeping the level of user intervention to a minimum. The operator should only be concerned with high-level

planning, such as "dock here" or specifying intermediate goals such as "go to this door". The robot must take care of all the lower-level control, such as staying in the center of a corridor or crossing a door.

In this thesis we address the problem of mobile robots navigation using visual information. Overall, the challenges we tackle for building autonomous robots are mainly threefold: (i) designing sensors adequate for the tasks at hand (ii) endowing the robot with the environmental representations (world models) and navigation modules able to solve navigation tasks; and (iii) designing adequate visual interfaces whereby the remote user can interact with the system in a simple and flexible manner.

In the sensorial aspect, vision is of particular interest as it enables not only self-localisation grounded to the world structure (or alternatives for navigation without explicit localisation) but, simultaneously, other applications such as vigilance or environment understanding by an human operator. The versatility of vision motivated and continues to motivate much research with many initial promising results [4, 66, 117, 5, 11, 58]. However, current vision-based navigation requires large computational resources, but still lacks the robustness required for many real-world applications.

In contrast, examples of efficiency can be drawn from biology. Insects, for instance, can solve very large and complex navigation problems in real-time [108], in spite of having limited sensory and computational resources.

One striking observation is the diversity of "ocular" geometries (see figure 1.1). Many animals eyes point laterally, which may be more suitable for navigation purposes. The majority of insects and arthropods benefit from a wide field of view and their eyes have a space-variant resolution. To some extent, the performance of these animals can be explained by their specially adapted eye-geometries. Similarly, in this thesis we explore the advantages of having large fields of view by using an *omnidirectional camera* with a 360° horizontal field of view.

Most of the research on vision-based navigation has been centered on the problem of building full or partial 3D representations of the environment [120], which are then used to drive an autonomous robot. We argue that shifting the emphasis from the actual navigation problem to the process of building these 3D maps, also contributes to explain the large computational resources and the lack of robustness in many applications.

By contrast, navigation in the animal world exhibits remarkable levels of robustness and flexibility when compared to today's robots. This is possibly due [13, 92] to the usage of a very parsimonious combination of perceptual, action and representational strategies. A related aspect, also intimately connected with the performance of a navigation system, is the nature of the navigation requirements when covering long distances, as compared to those for short paths. Many animals, for instance, make alternate use of landmark-based navigation and (approximate) route integration methods [108]. For example, to walk along a city avenue, it is sufficient to know our position to within an accuracy of one block. However, entering our hall door would require much more precise movements.

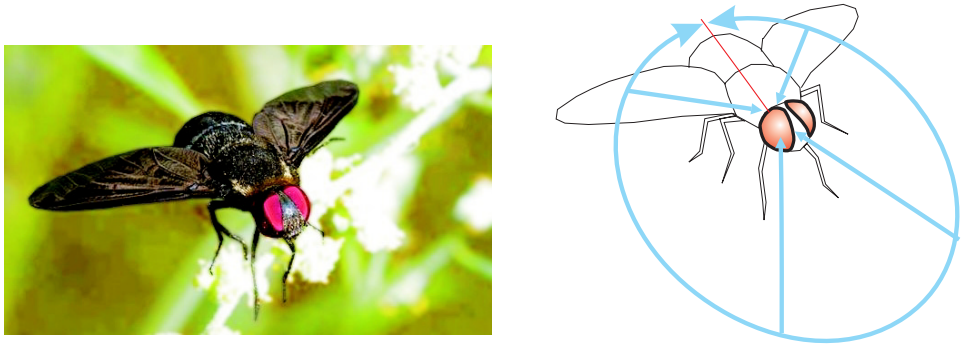


Figure 1.1: Photograph of a *true fly*. The azimuthal field of view of a *true fly* is about 360^o. *Photograph courtesy of Armando Frazão (http://photo.digitalg.net/).*

This *path distance/accuracy* tradeoff between long-distance/low-precision and short-distance/high-accuracy mission segments plays an important role in finding efficient solutions to the robot navigation problem. We will refer to these navigation modes as *Topological Navigation* versus *Visual Path Following*.

A *Visual Interface* should provide the user with a rich perception of the robot's environment and task status and, at the same time, offer an intuitive way to give commands or mission specifications to the system. We present a simple "point-and-go" visual interface based on omnidirectional images which, depending on the task at hand, provides one of three different scene representations to the operator. Each representation is an environmental model suited to a given task. For example, the robot heading and direction are easily specified by clicking on a panoramic image representation rather than having to type in degree headings.

In summary, in this thesis we propose a new methodology for vision-based robot-navigation comprising three main aspects: design of the omnidirectional vision sensors, world modelling for navigation and visual interfaces for human-robot interaction. In the methodology, omnidirectional imaging allows building powerful environmental representations useful both for acting as world maps for navigation and for developing visual interfaces for the user. The combined design of the sensor, navigation tasks and human-robot interface contributes for obtaining an effective navigation system.

In the following sections we introduce more precisely the main aspects of our work. These aspects will be developed in detail in the following chapters of the dissertation.

1.1 Omnidirectional Vision Sensors

Many arthropods and insects with ingenious ocular geometries, despite having very limited resources, are very efficient in navigation. The diversity of ocular geometries therefore suggest that in biology the vision sensors evolve also as part of the process of solving the

navigation tasks at hand.

We find therefore important and challenging the question of which would be the vision sensor mostly adequate for robot navigation. Similarly to many ocular geometries found in biology, in this thesis we explore the advantages of wide fields of view by using omnidirectional cameras.

Omnidirectional cameras provide a 360° view of the robot's environment, in a single image, and have been applied to autonomous navigation, video conferencing and surveillance [117, 21, 68, 58, 85, 77], among others. Omnidirectional cameras have several advantages over conventional cameras. For example, visual landmarks are easier to find with omnidirectional images, since they remain in the field of view much longer than with a conventional camera. There is also improved robustness to occlusion due to a different scaling in the view-field of an omnidirectional camera: a person at a distance of 2m occludes about 50% of a traditional camera equipped with a 12mm focal length lens as compared to 5% of an omnidirectional camera.

Advantages can be found also in egomotion estimation [79]. As Nelson and Aloimonos pointed out in [79], egomotion estimation algorithms are simpler when using omnidirectional images, since the rotation and orientation components of a movement can be decoupled. Madsen and Andersen show that the accuracy of self-localisation is largely influenced by the locations of the landmarks relative to the robot [71]. Only omnidirectional vision allows selection of landmarks all around the robot.

In order to use omnidirectional cameras first it is necessary to study their geometry and design. Omnidirectional cameras can be built using fish-eye lenses or by rotating cameras, but most frequently are obtained with catadioptric panoramic cameras [78], which combine conventional cameras (lenses) and convex mirrors. Mirror shapes can be conic, spherical, parabolic or hyperbolic [2, 99, 109]. The various mirror shapes imply diverse imaging geometries. As Geyer and Daniilidis show in [38], hyperbolic and parabolic mirrors can be dealt with through a unified model. The other cases, still require specific models to be developed for each application.

Given the imaging geometry, computer vision methods can be transported to omnidirectional sensors most of the time without requiring transformations. In this vein we can find works in stereo [81, 11], or approaches with optical flow for structure from motion [40, 116]. An example where the change is significant is the epipolar constraint. For omnidirectional cameras based on hyperbolic mirrors, the epipoles (in this case two) are both in the image, and Svoboda et al in [99] show that the epipolar constraint is a conic function.

Svoboda also shows that the camera motion can be estimated using a method based on the factorisation of the essential matrix, introduced by Longuet and Higgins [69], adapted for omnidirectional cameras [100]. This process is however non-linear and, despite less sensitive than with conventional cameras motion estimation methods, it lacks adequate robustness.

We follow an alternative approach, based on image dewarpings. We dewarp omnidirectional images to obtain (orthographic) *Bird's eye views* of the ground plane, where perspective effects have been removed. This is useful as tracking ground plane features is transformed to the linear problem of estimating a 2D rigid transformation. Localisation is simply the inverse of these 2D transformations.

We dewarp also omnidirectional images to obtain *Panoramic views*, where the imaging of vertical lines of the scene is transformed from radial to vertical lines. Tracking the vertical lines consists therefore in estimating translations. These lines in conjunction with ground lines, define ground features which are used to further improve the localisation accuracy.

In order to obtain the dewarpings, we detail the image formation model for omnidirectional cameras with spherical and hyperbolic mirrors. Although our sensors do not have a single projection center (for a complete list of camera-mirror pairs and respective mounting distances see Baker and Nayar [2]) we found that this is not a severe limitation to our approach. The accurate modelling of the imaging geometry allows to derive the desired dewarpings.

The image dewarpings can also be obtained in a more efficient manner by directly using specialised mirror shapes. Chahl and Srinivasan [12] propose a family of constant gain mirrors that result in better mappings from vertical distances to the image plane. Hicks and Bajcsy show how to obtain a mirror profile from the numerical solution of a differential equation, which provides a bird's eye view of the ground plane [48]. This is an example of a constant resolution camera in the sense that there is a linear relation of metric distances from the ground plane to the image pixels. Other linear relations are useful, such as constant vertical or angular resolution. In our work we derive a general methodology for the design of these cameras.

Our methodology considers in addition the use of variable resolution cameras, more precisely log-polar cameras [64]. These cameras concentrate the finest detail in the centre of the sensor as inspired by the fovea of the human eye. The polar arrangement of the sensor when combined with a curved mirror, permits to obtain direct panoramic dewarpings, once more saving computational resources.

Concluding, it is important to note that Panoramic and Bird's Eye Views provide not only simple sensor models for navigation, but also simple environmental representations. For example they are means for humans to specify goals for the robot to reach and to assess (display) the progression of navigation tasks.

In summary, omnidirectional cameras have interesting advantages over conventional cameras for navigation applications. The projection model of an omnidirectional camera depends on the mirror type, but many of them are represented by a unified model. Cameras with a single projection centre have simplified models. In our applications we use Panoramic and Bird's Eye Views as environmental representations. Those images are obtained from the original omnidirectional image through dewarpings or custom mirror

shapes / camera types. In the following sections we detail the use of the sensor and environmental representations for navigation.

1.2 Navigation

Traditionally, localisation has been identified as a principal component of the navigation system of a mobile robot [65]. This drove continuous research and development of sensors providing direct localisation measurements.

There is a large variety of self-localisation solutions available [6] in the literature. However they are in general characterised by a hard and limiting tradeoff between robustness and cost. As paradigmatic and extreme examples we can refer solutions based on artificial landmarks (beacons) and those based on odometry. Solutions based on beacons are robust but expensive in terms of the materials, installation, maintenance or configuration to fit a specific new purpose. The solutions based on odometry are inexpensive, but since they rely on the integration of the robot's internal measurements, i.e. not grounded to the world, errors accumulate over time.

As referred, we use vision as it provides world structure information. In particular, omnidirectional vision has been noted to be beneficial for the navigation of mobile robots, making sense to explore it as part of the solution of the navigation problem. In addition, the studies of animal navigation indicating a very parsimonious use of resources, suggest considering navigation modalities for the tasks at hand.

Our robot combines two main navigation modalities: Visual Path Following and Topological Navigation. In Visual Path Following, the short-distance / high-accuracy navigation modality, the orthographic view of the ground plane is a convenient world model as it makes simple representing / tracking ground plane features and computing the pose of the robot. Panoramic views, i.e. images as obtained from cylindrical sensors, are a complementary representation that is useful for vertical line features. These types of views are easily obtained from omnidirectional cameras using image dewarpings.

In Topological Navigation, the large-distance low-precision navigation modality, omnidirectional images are used in its raw format to characterise the environment by its appearance. Omnidirectional images are advantageous as they are more robust to occlusions created e.g. by humans. Visual servoing is included in topological navigation as the means providing the robot local control, thus saving environment representation detail and computational (memory) resources.

In the following sections, the navigation modalities, Visual Path Following and Topological Navigation, are introduced in more detail.

1.2.1 Visual Path Following

Visual Path Following can be described in a simple manner as a trajectory following problem, without having the trajectory explicitly represented in the scene. The trajectory

is only a data structure learnt from example / experience or specified through a visual interface. Progression is assessed based on the tracking of visual features (landmarks) within the environment. Visual Path Following is typically used for precision tasks such as docking or door traversal, for example.

As referred, this problem encompasses two main parts: determining self-localisation and computing the control signals for moving the robot. Given that the control can be computed using e.g. the controller proposed by de Wit et al [18], we concentrate on the localisation problem.

Deriving a robot location from a set of image features is a general camera exterior orientation / pose estimation problem [101]. Given the back-projection model [45] of an omnidirectional sensor allows to use the factorisation method of Longuet and Higgins [69] for determining the pose, as detailed by Svoboda in [100]. Geyer and Daniilidis [39] go further by using an unified projection model for the omnidirectional camera, derive an absolute conic based calibration method and then apply the factorisation method arriving to an approach that encompasses calibration, structure and motion.

We follow an alternative approach by using scene representations, provided by the omnidirectional camera, adequate to the navigation task. These representations make possible to directly extract robot positions. As scene representations we use the Panoramic and the Bird's Eye Views, which are adequate for indoor environments where there are many vertical and horizontal straight lines. The robot pose can be computed from the bearings to vertical lines and the map of those features, using the Betke and Gurvits' method [5]. Alternatively the robot pose can be retrieved from the intersection of ground and vertical lines (corner points) or more generally as the result of a matching merit maximisation process. These methods are used concurrently and the current location is chosen by evaluating the image evidence, represented by a matching merit function.

The matching merit function is further useful for a fine pose adjustment. It ameliorates the pose estimate and reduces also the probability of losing tracking. Finally, the analysis of the merit function provides a detection mechanism for tracking losses.

The image features, straight lines, despite being tracked and identified using the RANSAC [28] robust detection procedure, still carry some noise. The estimated robot localisation is therefore also contaminated with noise that, depending on the particular scene (landmarks structure) and desired trajectory, can be very significant constantly saturating the control signals. These signals are filtered with an Extended Kalman Filter (EKF). It is interesting to note that filtering the robot localisation along time is an efficient computation [114] as compared to work on image features [17], which involves a larger and variable state size EKF.

The appropriate choice of the sensor and the environmental representations, together with taking into account the task at hand, results in an efficient methodology that hardwires some tasks requiring precise navigation.

1.2.2 Topological Navigation

When entering one's hall door, humans need to navigate with more precision than when walking along a city avenue. This observation of a *path distance/accuracy trade-off* allows for a parsimonious use of the available resources for navigation [35, 111].

For local (precise) navigation we use *Visual Path Following*, as detailed in the preceding section. For global navigation tasks we need another navigation modality. Precision may be lesser since the main objective is a global representation of the environment that allows global (qualitative) self-localisation measurements.

The problem of global representations have been considered in several works. Zheng and Tsuji [121] represent the environment by its appearance, collecting side by side vertical image strips taken along a trajectory. The current robot location is found by the maximum correlation of a local image with the global mosaic. As the global data is large and the computation time significant at the registration process, Li and Tsuji [66], propose an alternative more compact (iconic) representation encompassing salient regions of interest that are used later as landmarks for qualitative localisation.

Basri and Rivlin [4], represent the environment also by its appearance, but using only lines, more precisely 2D views of 3D lines of the scene. Generic views are defined as linear combinations of a small number of views. The weights of the linear combination define the current camera location. The localisation can be very precise, but it is required a registration of the current image points against the database one.

Murase and Nayar augment the detail at describing the appearance of objects for an application of object recognition [74]. Appearance is defined as an image based representation of an object, which is a function of the combined effects of the object's shape, surface reflectance properties, pose and illumination conditions.

Appearance based techniques can be applied straightforwardly to omnidirectional cameras. Hong et al [49], using an omnidirectional camera based on a spherical mirror, assign to each robot-location a 1D signature consisting of the average of the intensity values for the various elevations at each azimuthal angle. The environment is therefore represented by a set of those signatures and the current location found by the correlation of the current view with the database. The robot is controlled to follow a path by homing to a sequence of database views.

Yagi et al [117], use an omnidirectional camera based on a conic mirror, also obtain a 1D representation at each robot-location, but now describing the existence at each azimuth of a vertical edge line. The tracking of bearings to the vertical edge lines combined with the knowledge of the robot motion allows defining a 2D map of the location of the vertical lines (landmarks). This map is used later for retrieving the robot location during normal operation.

More recently, there has been used directly the appearance as defined by Murase and Nayar. Ishiguro and Tsuji [54] represent the environment by omnidirectional images taken at a regular 2D grid. The self localisation is computed by comparing the power spectrums

of current and reference images so as to overcome mismatches due to different robot headings. Aihara et al [1], take a similar approach by computing autocorrelation values, but focusing their work on the problem of representing a large working space.

In these cases, the representation of the environment for localisation and control computation requires densely sampling the environment, resulting in large databases that in some cases need to be partitioned [1].

The approach closest to ours is that of Santos-Victor et al [91], that combines appearance based localisation with visual servoing for navigation the robot. However the imaging geometry, the data representation and matching procedure are different. Our solution for determining the global (qualitative) position of the robot is therefore *appearance-based* combined with visual servoing [35, 113]. Visual servoing allows to derive control signals for the robot without having to densely sample the environment for obtaining reference images.

Appearance, as defined by Murase and Nayar, implies extremely large amounts of data (images). The data is however largely redundant either, and thus they propose to compress it constructing a low-dimensional eigenspace [74], obtained via Principal Component Analysis (PCA). Appearance is therefore approximated by a manifold on the reduced order eigenspace. Most of the works cited in this section apply this compression technique [74, 54, 1, 91].

Illumination is of particular importance in appearance based localisation when considering navigation tasks involving trajectories close to windows at different times of the day. There may occur large non-uniform illumination changes which often fail image comparisons based on conventional L_2 norms or normalised correlation methods. We incorporate this analysis in our work and propose modified environmental representations and image comparison techniques, for locations known to be affected by the illumination changes.

To conclude, we use topological navigation for global tasks, such as going to a distant location. The representation used is a topological map of the environment based on the appearances of the various locations. Advantageously, Visual Path Following complements Topological Navigation, providing the robot with the ability to undertake tasks requiring different levels of knowledge of the world.

In the following section we introduce visual interfaces to help the user set high-level tasks for the robot to accomplish. These are supported by the navigation modalities just presented that do not require the user to accurately control the robot along a path.

1.3 Visual Interfaces

Introducing robots in the human society implies designing the (social) interactions with humans. This encompasses robots communicating to humans their sense of the scene and accepting tasks to accomplish.

Our objective is to design a user interface that is intuitive, easy to use and allows the operator to achieve tasks with maximum efficiency. We place a large emphasis on simplicity

with the result that our interface is completely vision-based. The interaction with the robot is based on the navigation modalities introduced in the preceding sections, so as to give to the user only high level decisions, such as target locations, with the advantage of being feasible even over low bandwidth communication channels. At the simplest level, there are three modes at which the operator can control the robot: heading, position and pose.

The modes for controlling the robot's *heading* or (x, y) *location* are naturally based on respectively *panoramic* or *bird's eye views*. An immediate benefit of using these views is that every heading direction or target location, in a region surrounding the robot, can be specified with a single command. This gives the operator a great deal of flexibility when deciding in what direction, or towards which location, the robot should travel while simultaneously allowing a speedy decision to be made. When the target locations are within the region covered by the topological map, the robot uses the Topological Navigation tasks already available to move to the target point. Otherwise, the operator adds a new Visual Path Following task by specifying landmarks and trajectories in the bird's eye views. The target location is then reached following the path relying on the self-localisation relative to the landmarks. Thus, there is a natural correspondence between the design of the user interface and the action required from the robot.

The final mode the operator can use to control the robot consists of a 3D model of the world. The operator has the option of viewing the remote scene by taking a virtual walk through it and the robot will attempt to follow the specified trajectory. To increase the usability of the robot, it must be simple to integrate new working areas (e.g. rooms), and therefore to build new 3D models for interaction. To build the 3D models we propose *Interactive Scene Reconstruction*, where the 3D models are built combining the data of one or more scene images with some limited user provided input.

Interactive scene reconstruction has recently drawn lots of attention. Debevec et al in [19], propose an interactive scene reconstruction approach for modelling and rendering architectural scenes. It is an hybrid approach combining geometry-based and image-based methods traditionally found in the computer-graphics and computer-vision communities. They derive a geometric model combining edge lines observed on the images with geometrical properties known a priori. This approach is advantageous relative to building a CAD model from scratch, as some information comes directly from the images. In addition, it is simpler than a conventional structure from motion problem because, instead of reconstructing points, it deals with reconstructing scene parameters, which is a much lower dimension and better conditioned problem. The single image case is not considered, as feature correspondences still play an important role.

Criminisi et al in [15] show how to take measurements from a single view obtained by an uncalibrated camera. The measurements are obtained up to a scale factor that may be found from a known length of an object present in the scene. The method is based on choosing a reference-plane and a reference-direction not parallel to the plane, from which projection equations can be written and the desired measurements extracted. The

calibration of the camera is extracted also from scene geometrical properties (vanishing points) [10]. The measurements are taken on the reference plane or on the reference direction. Hence, the geometry constraints are again found to be useful for reconstruction.

In [96] Sturm uses an omnidirectional camera based on a parabolic mirror and a telecentric lens for reconstructing the 3D scene. The user specifies relevant points and planes grouping those points. The directions of the planes are computed e.g. from vanishing points, and the image points are back-projected to obtain parametric representations where the points move on the 3D projection rays. The points and the planes, i.e. their distances to the viewer, are simultaneously reconstructed by minimizing a cost functional based on the distances from the points to the planes. This work shows that, as in many cases, algorithms designed for conventional cameras can be transported in a straightforward manner to the omnidirectional cameras.

We build 3D models using omnidirectional images and some limited user input, as in Sturm's work. However our approach is based on a different reconstruction method and the omnidirectional camera is a generalised single projection centre camera modelled by the Unified Projection Model [38]. The reconstruction method is that proposed by Grossmann for conventional cameras [42], applied to single projection centre omnidirectional cameras for which a back-projection model was obtained.

The back-projection transforms the omnidirectional camera to a (very wide field of view) pin-hole camera. The user input is of geometrical nature, namely alignment and coplanarity properties of points and lines. After back-projection, the data is arranged according to the geometrical constraints, resulting in a linear problem whose solution can be found in a single step.

3D models are perhaps the most intuitive representations of the world scenes as they allow the operator to specify commands in a "walk-through" manner. The operator, unlike when using the panoramic mode, has an intuitive idea of distances the robot may be required to travel. Naturally, the operator is not only constrained to viewing the scene from a "natural viewpoint" but can manipulate the model to view the scene from any 3D world point. A unique feature of this interface is that the user can tell the robot to arrive to a given destination at a certain orientation simply by rotating the 3D model.

1.4 Structure of the Dissertation

This thesis addresses the problem of Mobile Robot Navigation based on Omnidirectional Vision. By exploring the combined design of (i) the sensor, (ii) the navigation modalities and (iii) the human-robot interfaces, an effective navigation system is obtained.

For clarity of exposition the three main aspects are described in different chapters, despite the fact that naturally each topic is closely related to the others. The navigation modalities are further subdivided into two chapters, where the first one details a local and precise navigation modality and then the following chapter proposes a global qualitative modality that, at specific locations, instantiates the local navigation.

In **Chapter 2**, we present the geometric (projection) models and the design criteria for omnidirectional cameras. The Panoramic and Bird's Eye Views, dewarpings of the omnidirectional images, are introduced as environmental representations yielding priors (hypotheses) that are useful for the navigation tasks. Also presented is a unified design for constant resolution cameras. These cameras provide the dewarped images directly.

In **Chapter 3** we propose Visual Path Following for local and precise navigation. It is performed based on landmarks composed by ground and vertical lines, which are typical in indoor scenarios, such as corridor guidelines and door or window frames. Self-localisation is identified as a main component and therefore we present a number of methods for computing the robot's pose. Visual path following is tested in docking and door crossing experiments.

Chapter 4, vision-based navigation with an omnidirectional camera, addresses navigation tasks covering large areas. Topological Navigation is used for traversing long paths. To represent the global environment we use appearance based methods. For the regions where large non-uniform illumination changes may occur, there are proposed representations based on image-edges, instead of intensities. At locations requiring precise navigation it is used Visual Path Following. The resulting navigation system is tested in an extended experiment consisting of travelling in a room from the docking position to the door, crossing the door to the corridor, travelling along the corridor and coming back to the starting room and docking position.

Chapter 5 describes interactive scene modelling. The scene is represented by a 3D model, which is reconstructed from single or multiple omnidirectional images and some geometrical properties provided by the user. The model is applied in an human-robot visual interface.

Finally, in **Chapter 6** we draw some conclusions and establish further directions of research.

1.5 Original Contributions

In this thesis we address the problem of mobile robot navigation in several aspects. We have contributions in the design of the complete system and its individual components, namely sensor design, navigation control, perception algorithm and user interaction:

- We propose a novel uniform formalism for designing constant resolution (omnidirectional) cameras [32]. This is a result of designing a number of omnidirectional vision setups in which we have identified the convenience of constant resolution for navigation applications.

- We propose Visual Path Following for local and precise navigation. It is a simple method due to the combined sensor-navigation design [34].

- We describe methods based on image-edges for improving robustness of Topological Navigation to large, non-linear illumination changes [111].
- We describe a method for interactive reconstruction based on omnidirectional images. It combines a method designed for conventional cameras with the unified back-projection model that we propose for single projection centre omnidirectional cameras [33].

Our approach to navigation resemble biological examples. For instance, the way humans perform their travellings, suggest a tradeoff between *path distance / accuracy* in the navigation modalities. We use Topological Navigation for travelling long distances, requiring qualitative positioning of the robot, while Visual Path Following is used for local and precise navigation tasks. By clearly separating the nature of the navigation tasks a simple and yet powerful navigation system is obtained [35].

Chapter 2

Omnidirectional Vision Sensors

The design of omnidirectional vision sensors is a central and fundamental part of our work. Of particular importance are the design criteria, namely the required field of view and the spatial arrangement of the output images. We detail the geometric aspects concerning image formation in order to implement methodologically the design criteria. The Panoramic and the Bird's Eye Views are fixed transformations applied upon the omnidirectional image, as opposed to the geometric models reconstructed from image and user provided data (which we shall detail in a latter chapter). These fixed transformations yield some priors very helpful for the task at hand. For example the assumption of working in a ground plane is reasonably general while providing direct cues for mobile robot navigation.

2.1 Introduction

Omnidirectional cameras have fields of view comprehending all directions around the viewpoint. This is an enhancement relative to *perspective* cameras and bring great promises for developing, simplifying or improving several applications in Computer Vision.

In the recent past some traditional problems like scene representation, surveillance or mobile robot navigation, were found to be conveniently approached using different sensors. There has been undertaken significant effort in research and development of omnidirectional vision systems [38, 77, 2, 119, 99, 12, 47, 14, 35, 20]. However, significant research and development continues to be done.

There are many types of omnidirectional cameras, being mostly common the ones based on rotating cameras, fish-eye lenses or mirrors. The work described in this dissertation is mainly concerned with the omnidirectional vision systems combining cameras with mirrors, normally referred as catadioptric systems in the optics domain, specifically in what concerns the mirror profile design.

Omnidirectional cameras are sometimes termed *panoramic* as it is normally the case that the view field is limited in the vertical direction being complete just horizontally. We will however use the term omnidirectional as it is the most common in the literature.

In the following we present a brief history of omnidirectional vision sensors and review the state of the art.

The history

As pointed out by Christopher Geyer in [38], the history on catadioptric systems may be traced back to the ancient Greece, with the work by Diocles on *Burning Mirrors* [105]. This work shows that a parabolic mirror converges parallel rays to a single point. It is interesting to note that the geometrical analysis there performed is applicable to recent catadioptric systems.

However, there were the inventions and breakthroughs on the automatic acquisition of images such as in photography, films or video, together with the advent of computers and generic image processing tools, that really motivated the most recent research and development.

Omnidirectional images were firstly obtained from rotating cameras. After the research by Bill McBride and Steven Morton [73], it turned out that those cameras appeared more than 150 years ago. The first referenced patent dates back to 1843. It was issued by Joseph Puchberger of Retz, Austria, and consisted on a rotating lens that formed an image over a film placed on a cylindrical surface. The horizontal field of view was limited to about 150^o but that was a start since a few years later, more precisely in 1857, M. Garella of England introduced a new rotating photographic mechanism and was already obtaining the 360^o field of view.

Rotating one camera, implies a delay on the acquisition of the image. Therefore, applications requiring instantaneous acquisition of omnidirectional images cannot be realised with moving cameras. The original idea of the (static camera) omnidirectional vision sensor has been initially proposed by Rees in 1970, in the US patent [86]. Rees proposed to use a hyperbolic mirror to capture the omnidirectional image which can be transformed to normal perspective images.

In 1987, Oh and Hall published experiments on robot navigation based on an omnidirectional sensor based on a fisheye lens [82]. In 1990 Yagi and Kawato [115] made an omnidirectional vision sensor using a conic mirror. In 1991 and 1993, Hong and others [49] and Yamazawa and others [118] published their designs based respectively on spherical and hyperbolic mirrors. In all these cases the application was robot's navigation.

Some mirrors when appropriately combined with cameras yield simple projection models. In order to obtain the systems conforming to simple models it is necessary in general to do a precise placement of the mirror relative to the camera. Nayar and Baker [76], in 1997, developed and patented a system combining a parabolic mirror and a telecentric lens, that is well described by a simple model and simultaneously overcomes the requirement of precise assembly. Further their system is superior in the acquisition of non-blurred images. It is interesting to note that this successfully designed system has a geometry similar to the ancient greek one, referenced in the beginning of this section, just following the light

rays in the opposite direction.

Many other systems have been and continue to be designed, as new applications, technological opportunities or research results appear. We overview the state of the art in the following section.

State of the art

Nowadays, there are many commercial omnidirectional vision sensors, mainly for the real estate markets. These systems offer well designed solutions, but still lack the flexibility required in many applications. For example when considering omnidirectional vision for robots, it is important to have fast data acquisition and to select the appropriate field of view. Small robots require larger fields of view as the relevant information is at a different scale relative to their sizes. Therefore, when concerning mobile robots, omnidirectional vision sensors are still manufactured according to custom designs, e.g. Chahl and Srinivasan in [11], Yagi et al in [117], etc.

Some designs aim at proving linear projection properties between world and image points. There are several recent reports on the construction of these cameras, for example: Chahl and Srinivasan [12], Conroy and Moore [14], Hicks and Bajcsy [48], Ollis et al [83], Gaechter and Padjla [30], Gaspar et al [32], etc.

Along with the sensor development, modelling evolves also for efficient representations, trading between model detail and convenience for mathematical derivations. Projection equations considering a single projection centre, as the ones of Baker and Nayar [2] and Geyer and Daniilidis [38], are mostly convenient.

Besides custom designs generally driven by the applications, there are research topics motivated by technical opportunities. Hardware optimisation evolves currently to achieve faster data acquisition, system size minimization while looking for resolution maximization. For example, the folded cameras as provided by Nayar et al [75], i.e. cameras where the light rays are reflected by a number of mirrors before reaching the imaging sensor, are currently targeting the market of small omnidirectional cameras.

Currently the knowledge on computer vision is still being migrated and incorporated into the omnidirectional vision research area. Given the base geometry, computer vision methods may be transported to omnidirectional sensors, most of the time without requiring transformations. In this direction there has been work in stereo reconstruction [81, 11], optical flow for structure from motion [40, 116]. An example where changes are significant are the epipolar constraints. Svoboda et al in [99] show for an omnidirectional camera with an hyperbolic mirror that the epipolar constraint is a conic function and that the two epipoles are usually visible in the omnidirectional image. Geyer and Daniilidis, [38], show that the epipolar constraint is a circle for cameras based on a parabolic mirror and a telecentric lens.

Chapter Organisation

In this chapter we detail the geometry of *catadioptric omnidirectional cameras*. In particular we describe a projection model for *Single Projection Centre* systems, the *Unified Projection Model* proposed by Geyer and Daniilidis [38], and present a projection model for non-single projection centre systems.

We describe the image formation model for a catadioptric omnidirectional camera with a general mirror profile and detail two cases of importance: the spherical and the hyperbolic mirrors.

We describe a method of image dewarping to obtain bird's eye views of the ground plane. This representation allows algorithmic simplifications because perspective effects of the ground floor are eliminated. For example, position information for ground points is available without reconstruction [25] or uncalibrated reconstruction [43], and ground features move rigidly in these images, thus being easier to track.

Finally, we introduce the so called *Constant Resolution Cameras* which are cameras combined with customised mirrors to yield linear properties in the projection model.

2.2 Catadioptric Omnidirectional Cameras

Catadioptric Omnidirectional Cameras combine conventional cameras and mirrors primarily to obtain specific fields of view. Omnidirectional sensing is possible with convex mirrors such as conic mirrors, spherical mirrors or hyperbolic mirrors [2, 99, 109]. An omnidirectional sensor allows one to track a single feature from varying viewpoints, when it would otherwise be out of the field of view of a fixed classical camera. Another advantage, over a standard pan and tilt camera, lies in its simplicity: the system has no moving parts and given that the robot-sensor relative position remains fixed, the orientation of the sensor is related to that of the robot by a rigid transformation.

The most common way to model a single camera is to use the well known projective camera model [25, 45]:

$$\tilde{m} = \tilde{P}\ \tilde{M} \tag{2.1}$$

where $\tilde{m}$ is the projection of the world point $\tilde{M}$, $\tilde{P}$ is the projection matrix which comprises the intrinsic parameters and a transformation from world coordinates to camera coordinates (see Fig.2.1).

With catadioptric cameras there is an additional transformation function due to the mirror standing between the world scene and the camera:

$$\tilde{m} = \tilde{P}\ \mathcal{F}\left(\tilde{M}\right) \tag{2.2}$$

where $\tilde{P}$ has the same meaning as described for the projective camera model, and $\mathcal{F}$ is the function introduced by the reflection of the light rays at the mirror. This function depends on the mirror shape and is in general non-linear. Stated simply it determines the

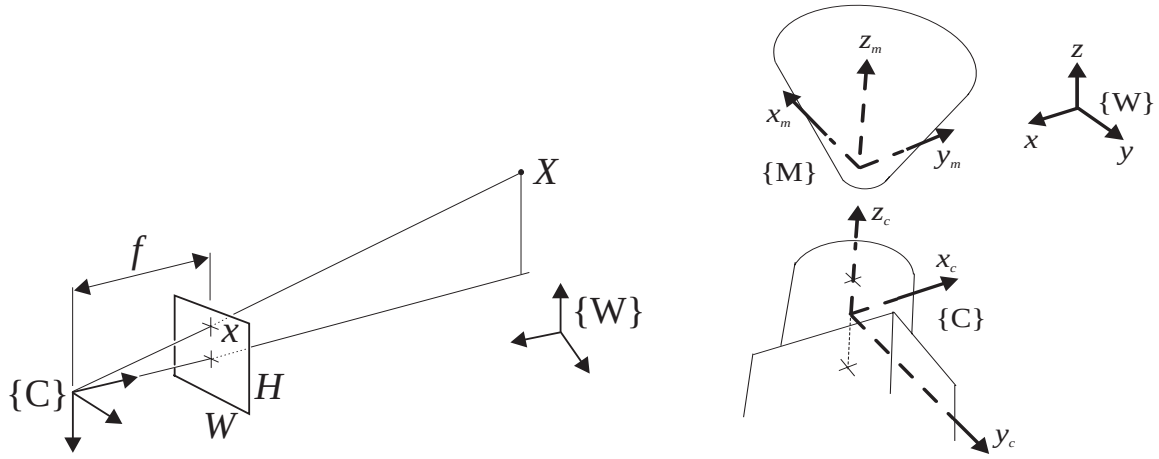


Figure 2.1: Pin-hole (left) and catadioptric (right) camera geometry.

point of the mirror where the ray coming from the 3D point $\tilde{M}$ is reflected and directed towards the camera's optical center.

In summary, catadioptric omnidirectional cameras are described by the projective camera model complemented by a non linear part given by function $\mathcal{F}$. Figure 2.1 illustrates one catadioptric omnidirectional camera. Notice that since the system is rotationally symmetric relatively to the z-axis, 3D geometry x, y, z is simplified to 2D geometry r, z ($r = \sqrt{x^2 + y^2}$).

Depending on the particular camera and mirror setup, the light rays incident to the mirror surface may all intersect at a virtual point. In this case the system is referred as having a *Single Projection Centre*. This is a very convenient property that allows simple projection models such as the *Unified Projection Model* that we describe in the following section.

2.2.1 Unified Projection Model

The *Unified Projection Model*, defined by Geyer and Daniilidis in [38], represents in an unified manner several single projection centre systems, such as pin-hole cameras and, most importantly, the recent omnidirectional (catadioptric) cameras based on hyperbolic, elliptical or parabolic mirrors.

The unified projection model combines a mapping to a sphere followed by a projection to a plane. The centre of the sphere lies on the optical axis of the projection to the plane. This allows a reduced representation with two parameters, l and m, representing the distances from the sphere centre to the projection centre, O and to the plane (see Fig.2.2). The projection of a point in space (x, y, z) to an image point (u, v) can be written as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{l+m}{l \cdot r - z} \begin{bmatrix} x \\ y \end{bmatrix} = \mathcal{P}(x, y, z; l, m) \tag{2.3}$$

$$r = \sqrt{x^2 + y^2 + z^2}$$

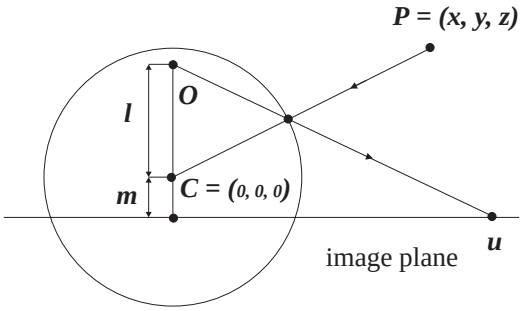


Figure 2.2: Unified Projection Model.

Each catadioptric camera with a single projection centre can be represented by a set of values l, m. For example, for pin-hole cameras, we have $l = 0$ and $m = 1$, while for cameras with hyperbolic mirrors, l and m are defined by the mirror parameters eccentricity and inter-focal length. The camera intrinsic parameters, image centre and focal length, combine naturally with the model as a two dimensional affine transformation of $[u\ v]^T$.

The unified projection model does not suit all omnidirectional vision sensors. In particular the non-single projection centre sensors cannot be exactly represented. However, in some applications it is possible to find an approximation to the unified case and thus take again the advantages of a simple and general model. We follow this approach in reconstruction tasks where it is known that the relevant 3D data is much farther way than the size of the mirror.

In order to find a good approximation to the unified model, it is convenient to have a precise modelling because it makes evident not only the quality of the approximation but also how the approximation quality varies with the parameters and the scene. This is one reason to introduce and use more precise and robust models. Another reason is the design of the sensor itself, where it is important to find the physical sizes of the individual components. This is detailed in the next section.

2.2.2 Model for Non-Single Projection Centre Catadioptric Cameras

As referred in the previous section, non-single projection centre systems cannot be represented exactly by the unified projection model. One such case is an omnidirectional camera based on an spherical mirror. The intersections of the projection rays incident to the mirror surface, define a continuous set of points distributed in a volume[3], unlike the unified projection model where they all converge to a single point. In the following, we derive a projection model for non-single projection centre systems.

Reflection Point

The image formation process is determined by the trajectory of rays that start from a 3D point, reflect on the mirror surface and finally intersect with the image plane. Considering first order optics [46], the process is simplified to the trajectory of the principal ray.

When there is a single projection centre it immediately defines the direction of the principal ray starting at the 3D point. Alternatively, we need to resort to more general laws still holding in our system. The local reflection law states that are the same the angles of incidence and reflection of a light ray. This angles are easily obtained if the point at which the ray is reflected is known. Therefore, the first goal to achieve is to find the reflection point.

One way to find the reflection point is through a numerical minimization process based on a intuitive geometric reasoning. The geometric reasoning looks for the error made by choosing an incorrect mirror reflection point. To detail this method we start, without loss of generality, reducing the problem to a 2D plane.

Due to the rotational symmetry of the system we only need to consider the design of the mirror profile. In the following we will use function $F(t)$ to represent this profile along the radial coordinate t. $F'(t)$ denotes the local slope of the mirror profile.

Given an arbitrary mirror point $(t, F(t))$ an incident light ray is defined. The distance from that ray to the 3D point to project will be zero only when the reflection point is the correct one[1]. Then we may find the reflection point by minimizing the just introduced error distance.

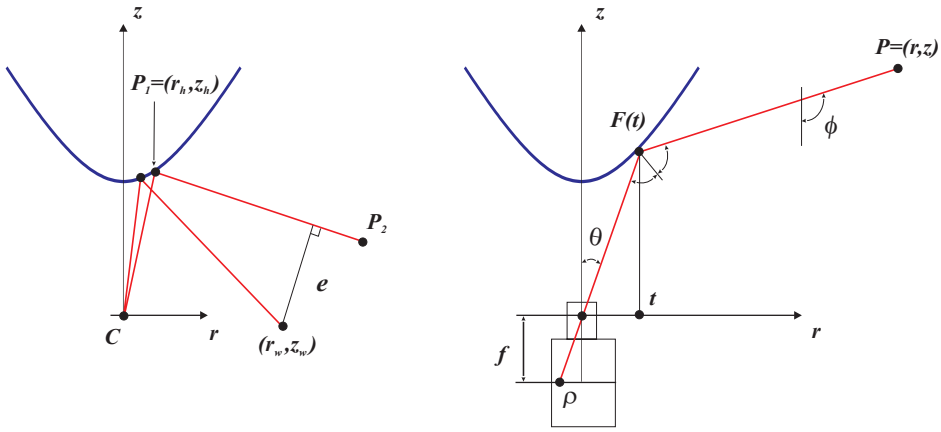


Figure 2.3: Left: defining the error distance for an approximate reflection point. Right: Main variables defining the projection equation.

Given the mirror point $P_1 = [t\ F(t)]^T$ the reflected light ray has the direction of $v = P_1$ since the camera centre (not to confuse to the world system's projection centre) is at the origin. A point on the incident ray may then be defined as $P_2 = 2v^T v_t v_t$ where v_t is the vector tangent to the mirror, $[1\ F'(t)]^T$ normalised to have unit norm. The error distance is then the distance from the 3D point $[r\ z]^T$ to the incident light ray defined by P_1 and P_2 (see Fig.2.3):

$$e(t) = dist\left([r\ z]^T,\ l\left(P_1(t), P_2(t)\right)\right) \tag{2.4}$$

Finally the radial coordinate of point of reflection is obtained minimizing the error,

[1]We are assuming that the 3D point to project does not lye inside the cone defined by the mirror volume and the camera centre.

$\hat{t} = \arg_t \min e(t)$, and the vertical coordinate as $F(\hat{t})$.

In case of being critical the computation time or desiring to do mathematical derivations on the computation of the reflection point, it is necessary to have a direct solution. The goal now is to obtain a system of equations whose solution directly gives the reflection point and consequently the trajectory of the light ray.

The geometry of the image formation of the omnidirectional catadioptric camera is shown in Figure (2.3).

Based on first order optics [46], and in particular on the reflection law, the following equation is obtained:

$$\phi = \theta + 2.atan(F') \tag{2.5}$$

where θ is the camera's vertical view angle, ϕ is the system's vertical view angle, F denotes the mirror shape (it is a function of the radial coordinate, t) and F' represents the slope of the mirror shape, as before.

Equation (2.5) is valid both for single [38, 77, 2, 119, 99], and non-single projection centre systems [12, 47, 14, 35, 20]. When the mirror shape is known, it provides the projection function. For example, consider the single projection centre system combining a parabolic mirror, $F(t) = t^2/2h$ with an orthographic camera [77], one obtains the projection equation, $\phi = 2atan(t/h)$ relating the (angle to the) 3D point, ϕ and an image point, t.

In order to make the relation between world and image points explicit it is only necessary to replace the angular variables by cartesian coordinates. We do this assuming the pin-hole camera model and calculating the slope of the light ray starting at a generic 3D point (r, z) and hitting the mirror:

$$\theta = atan\left(\frac{t}{F}\right) \quad , \quad \phi = atan\left(-\frac{r-t}{z-F}\right).$$

Then, expanding the trigonometric functions, one obtains an equation on the variables t, r, z encompassing the mirror shape, F and slope, F':

$$\frac{\frac{t}{F} + 2\,\frac{F'}{1-F'^2}}{1 - 2\,\frac{tF'}{F\left(1-F'^2\right)}} = -\frac{r-t}{z-F} \tag{2.6}$$

This is Hicks and Bajcsy's differential equation relating 3D points, (r, z) to the reflection points, $(t, F(t))$ which directly imply the image points, $(t/F, 1)$ [47]. We assume without loss of generality that the focal length, f is 1, since it is easy to account for a different (desired) value at a later stage.

Projection of a 3D Point

Given the method to calculate the reflection point at the mirror surface, it is now simple to derive the projection equations of a 3D point. It is only necessary to extend the simplified plane geometry to the 3D case and to include the camera's intrinsic parameters.

Let $P = [x\ y\ z]^T$ denote the coordinates of a general 3D point. We want to find the image projection, $p = [u\ v]^T$, of P using a catadioptric omnidirectional camera. The coordinates of P can be expressed in cylindrical coordinates as

$$P = [\varphi\ r\ z]^T = \left[\arctan(y/x)\ \ \sqrt{x^2+y^2}\ z\right]^T \tag{2.7}$$

Knowing the mirror profile, F and slope, F' the reflection point $P_m = [\varphi\ t\ F(t)]^T$ on the mirror surface is the solution of Eq.(2.6) for the given 3D point (r, z).

All that remains to be done is to project the 3D point P_m onto the image plane $p = (u, v)$. Using the perspective projection model and taking into account the camera intrinsic parameters, we get:

$$\begin{bmatrix} u^* \\ v^* \end{bmatrix} = f\ \frac{t}{F} \begin{bmatrix} \cos\varphi \\ \\ \sin\varphi \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \end{bmatrix} \begin{bmatrix} u^* \\ v^* \\ 1 \end{bmatrix}. \tag{2.8}$$

where $\alpha_u, \alpha_v, u_0, v_0$ denote the vertical and horizontal image scale factors and position of the principal point on the image coordinate system.

Maximum vertical view angle

Property 1 *Consider a catadioptric camera with a pin-hole at* $(0, 0)$ *and a mirror profile* $F(t)$*, which is a strictly positive* C_1 *function, with domain* $[0,\ t_M]$ *that has a monotonically increasing derivative. If the slope of the light ray from the mirror to the camera,* t/F *is monotonically increasing then the* maximum vertical view angle *is obtained at the mirror rim,* $t = t_M$*.*

Proof: from Eq.(2.5) we see that the maximum of the vertical view angle, ϕ is obtained when t/F and F' are maximums. Since both of these values are monotonically increasing, then the maximum of ϕ is obtained at the maximal t, i.e. $t = t_M$.

□

Notice that if the slope of the light ray from the mirror to the camera, t/F is not monotonically increasing then there is a mirror region occluding at least one other mirror region. When t/F changes from increasing to decreasing it is possible to define a line tangent to the mirror passing through the camera pin-hole. Larger t values motivate inner mirror points relative to the tangent line, and are therefore occluded by points on t values happening before the tangent. Thus, the tangent line ultimately defines the maximum vertical view angle.

In practice, the mirror profile is truncated much earlier than the bound indicated by the tangent line. The reason is that the reflections close to the bound are almost tangencial and the projection geometry becomes too sensitive to the uncertainties inevitably introduced by the assembly of the system or the manufacturing of its components.

Since the mirror has a limited size, in some cases it does not fill the complete field of view provided by the lens, leading to an image region showing 3D structure not reflected

by the mirror. For omnidirectional cameras placed upwards with the mirror over the pin-hole camera, this can cause direct ceiling light exposition. In this case, it is typical to include a pale occluding the view area around the mirror.

Scaling Property

Let us define the scaling of the mirror profile (and distance to camera) $F(t)$ by $(t_2, F_2) \doteq \alpha.(t, F)$, where t denotes the mirror radial coordinate. More precisely we are defining a new mirror shape F_2 function of a new mirror radius coordinate t_2 as:

$$t_2 \doteq \alpha t \quad \wedge \quad F_2(t_2) \doteq \alpha F(t). \tag{2.9}$$

This scaling preserves the geometrical property that we state in the following.

Property 2 *Given a catadioptric camera with a pin-hole at* $(0, 0)$ *and a mirror profile* $F(t)$*, which is a* C_1 *function, the* vertical view angle is invariant *to the system scaling defined by eq.(2.9).*

Proof: we want to show that are equal the vertical view angles at corresponding image points,

$$\phi_2(t_2/F_2) = \phi(t/F)$$

which, from Eq.(2.5), is the same as comparing the corresponding derivatives:

$$F_2'(t_2) = F'(t)$$

and is demonstrated using the definition of the derivative:

$$F_2'(t_2) = \lim_{\tau_2 \to t_2} \frac{F_2(\tau_2) - F_2(t_2)}{\tau_2 - t_2} = \lim_{\tau \to t} \frac{F_2(\alpha\tau) - F_2(\alpha t)}{\alpha\tau - \alpha t} = \lim_{\tau \to t} \frac{\alpha F(\tau) - \alpha F(t)}{\alpha\tau - \alpha t} = F'(t)$$

□

Stated simply, the scaling of the system geometry does not change the local slope at mirror points defined by fixed image points. In particular, the mirror slope at the mirror rim does not change and therefore the vertical view angle of the system does not change.

Notice that despite the vertical view angle remaining constant the observed 3D region actually changes but usually in a negligible manner. As an example, if the system sees an object 1 metre tall and the mirror rim is raised 5 cm due to a scaling, then only those 5 cm become visible on top of the object.

Now we have all the tools to introduce and detail the steps of catadioptric camera design based on spherical or hyperbolic mirrors. These steps comprise defining the design specifications and respective methods for computing mirror parameters. The scaling property simplifies significantly the design process.

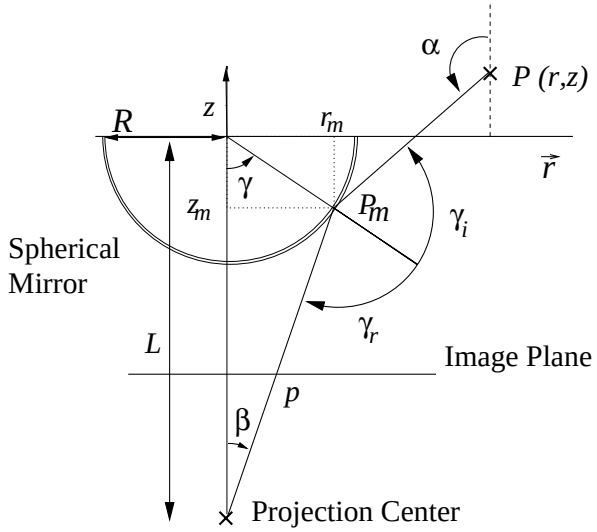


Figure 2.4: Catadioptric Omnidirectional Camera (spherical mirror) projection geometry. Symmetry about the z-axis simplifies the geometry to 2D (r,z).

2.2.3 Omnidirectional camera based on a spherical mirror

In this section we describe the model for image formation with a spherical mirror, using the general projection model for catadioptric cameras introduced in the previous section. Catadioptric vision sensors based on a spherical mirror are modelled essentially by the equation of reflection at the mirror surface, Eq.(2.5), stating that are equal the incidence and reflected angles of the ray of light.

The mirror profile is simply represented by a semi-circle equation:

$$F(t) = L - \sqrt{R^2 - t^2} \tag{2.10}$$

where R is the radius of the spherical mirror and L is the camera to sphere centre distance (see figure 2.4).

The geometry of image formation is obtained by relating the coordinates of a 3D point, P to the coordinates of its projection on the mirror surface, P_m and finally to its image projection p, as shown in Figure 2.4. This is done by the projection equations deduced in the previous section for a general mirror profile.

Replacing F in Eq.(2.6) by the semi-circle expression, we obtain an equation on the mirror radial coordinate, t. The parameters of the equation are the 3D point (r, z) and the system constants R, L, respectively the mirror radius and the camera to mirror centre distance. The solution of the equation gives the radial coordinate of the reflection point on the mirror surface. We represent this solution by the operator $\mathcal{P}_0$:

$$t = \mathcal{P}_0(r, z; R, L)$$

The vertical coordinate of the mirror point comes then as $F(t)$. Equations 2.7 and 2.8 complete the projection model, preforming the transformation into cylindrical coordinates

and accounting for the intrinsic parameters. Therefore we obtain a projection operator, $\mathcal{P}$ that given the 3D coordinates of a point $P = [X\ Y\ Z]^T$ allows us to obtain its image projection $p = [u\ v]^T$:

$$p = \mathcal{P}(P, \vartheta) \tag{2.11}$$

where ϑ contains all the intrinsic and extrinsic parameters of the catadioptric omnidirectional vision sensor:

$$\vartheta = [L\ R\ \alpha_u\ \alpha_v\ u_o\ v_o]^T.$$

Designing the system

The design of an omnidirectional camera system is achieved by exploiting the degrees of freedom of the model parameters. In the particular case of the camera based on a spherical mirror there are three main degrees of freedom, namely the mirror radius R, the camera to mirror distance L and the focal length of the camera lens f (α_u and α_v in the model).

Depending on the application at hand, there are different design goals. Controlling camera resolution towards a certain target is an example. In our case we are mainly interested in specifying the vertical view field of the camera. In particular we want to specify vertical view field going above the horizon line a number of degrees.

We start by fixing the focal length of the camera, which directly determines the view field θ. Then the maximum vertical view field of the system, ϕ, is imposed with the reflection law Eq.(2.5). This gives the slope of the mirror profile at the mirror rim, F' or the necessary sector of the spherical surface, γ:

$$\gamma = atanF' = \phi - \theta.$$

Assuming that the mirror radius is unitary, the distance from the camera to the mirror centre is then:

$$L = cos\gamma + sin\gamma/tan\theta$$

which shows the distance between the camera and the mirror surface, $D = L - 1$. Since there are minimal focusing distances, D_{min} which depend on the particular lens, we have to guarantee that $D \geq D_{min}$. We do this applying the scaling property

$$(R, L) \leftarrow k.(1, L)$$

with $k = D_{min}/D$. If the mirror is still too small to be manufactured then an additional scaling up may be applied. The camera self-occlusion becomes progressively less important when scaling up.

Figure 2.5 shows an omnidirectional camera based on a spherical mirror, built in house for the purpose of conducting navigation experiments. The mirror was designed to have a view field of 10^o above the horizon line. The lens has $f = 8mm$ (vertical view field of about $\pm 15^o$ on a $6.4mm \times 4.8mm$ CCD). The minimal distance from the lens to the mirror surface was set to $25cm$. The calculations indicate a spherical mirror radius of $8.9cm$.

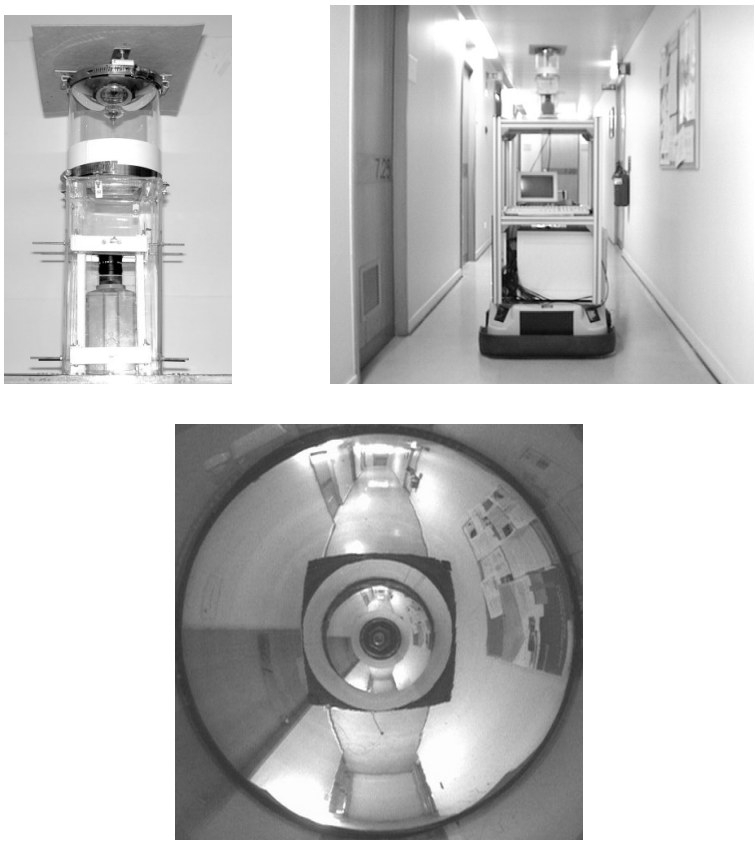


Figure 2.5: (Top-left) Omnidirectional camera based on a spherical mirror, (top-right) Camera mounted on a Labmate mobile robot and (bottom) Omnidirectional image.

Model parameters estimation (Calibration)

Due to uncertainties in the manufacturing and assembling of an omnidirectional camera, the resulting parameters certainly deviate from the desired values. It is therefore important to estimate the real parameters either for a precise modelling or simply to control the quality of the manufacturing process.

In the previous section we introduced the projection operator $\mathcal{P}$ that, given the 3D coordinates of a point, allows us to obtain its image projections. Now our goal is to estimate its parameters, ϑ, for a real system.

The mirror radius can be measured easily and we assume that it is known, $R = 8.35\,cm$. We further assume that the pixel aspect ratio is known. However, the camera-mirror distance, L, the overall image scale factor α and the principal point, (u_0, v_0), can only be measured up to some measurement error:

$$\begin{cases} L = 27 + \delta L\ [cm] \\ \alpha = \hat{\alpha} + \delta\alpha\ [pix/cm] \\ u_0 = \hat{u_0} + \delta u_0\ [pix] \\ v_0 = \hat{v_0} + \delta v_0\ [pix] \end{cases} \tag{2.12}$$

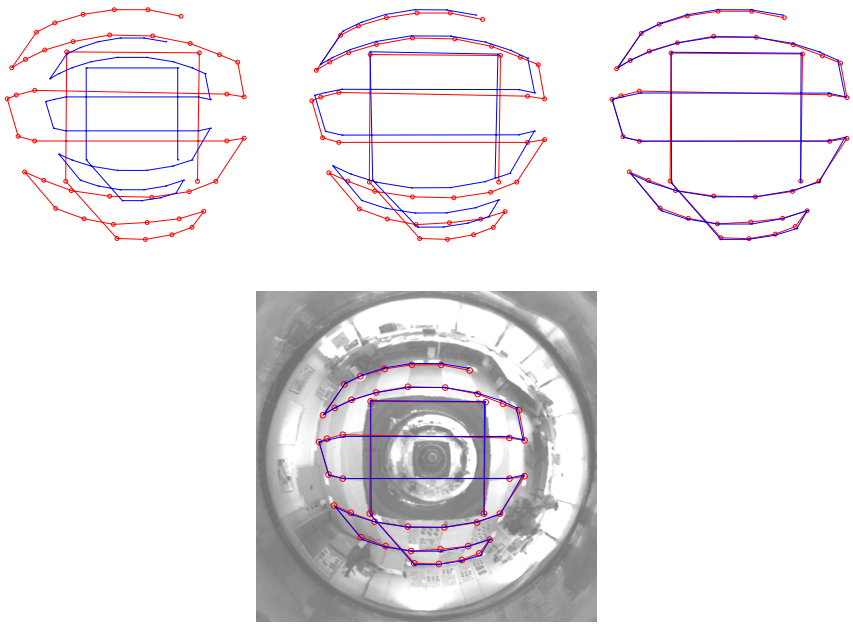


Figure 2.6: Iterations of the calibration procedure, from the initial values (top-left) to the final result superimposed on the image (bottom).

Hence, we define the adjustment $\delta\vartheta$ required to correct the nominal parameter vector ϑ:

$$\delta\vartheta = [\delta L \ \delta\alpha \ \delta u_0 \ \delta v_0]^T$$

To estimate $\delta\vartheta$ we use a set of known 3D points, P^i, and the corresponding image projections p^i, and minimize the following cost function:

$$\delta\vartheta = \arg\min_{\delta\vartheta} \sum_i \| \ p^i - \mathcal{P}(P^i, \delta\vartheta) \ \|^2 \tag{2.13}$$

Figure 2.6 illustrates the calibration procedure. The figure shows the observed and computed images of known 3D points, for parameter vectors obtained at different iterations of the minimization process. The final parameter vector gives a good approximation of the projection model to the real world camera.

At this point we have defined the projection operator for catadioptric omnidirectional images with spherical mirrors, and described a procedure to estimate the model parameters, starting from initial nominal settings.

2.2.4 Omnidirectional camera based on an hyperbolic mirror

In this section we detail the model for image formation of an omnidirectional camera based on a hyperbolic mirror. As in the case of the spherical mirror the modelling is based on the equation of reflection at the mirror surface, Eq.(2.5).

The profile of an hyperbolic mirror has the general form:

$$F(t) = L + \frac{a}{b}\sqrt{b^2 + t^2} \tag{2.14}$$

where a, b are the major and minor axis of the hyperboloid and L controls the camera to the mirror distance. As an example, $L = 0$ in the omnidirectional camera proposed by Chahl and Srinivasan's [12]. Their design yields a constant gain mirror that linearly maps 3D vertical angles into image radial distances.

Chahl and Srinivasan's design does not have the single projection centre property, which is obtained placing the camera at one hyperboloid focus, i.e. $L = \sqrt{a^2 + b^2}$, as Baker and Nayar show in [2] (see Fig.2.7). This solution was actually proposed in the 70s by Rees and there are commercial systems based on it (e.g. the one provided by the company Accowle). The contribution of the work by Baker and Nayar is the constructive search for all single projection centre systems, and the finding that there are only three setups satisfying that goal.

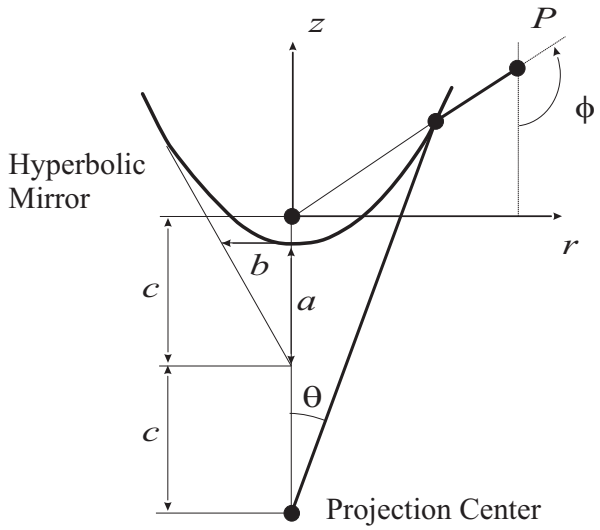


Figure 2.7: Catadioptric Omnidirectional Camera (hyperbolic mirror) projection geometry for the single projection centre case. Symmetry around z-axis simplifies the geometry to 2D (r,z).

We are interested in creating an image formation model able to cope with the general case of single or non-single projection center as given by Eq.(2.14). As in the case of the spherical mirror, this is done applying once more the projection equations previously deduced for a general mirror profile. The reflection point on the mirror surface is found solving Eq.(2.6) using the hyperbolic mirror profile of function F we just presented. Similar to the case of the spherical mirror, we represent the solution of the equation with an operator $\mathcal{P}_0$ but now in the parameters of the hyperboloid:

$$t = \mathcal{P}_0(r, z; a, b)$$

As before, to relate 3D and image points in cartesian coordinates, equations 2.7 and 2.8 do the transformation into cylindrical coordinates and account for the intrinsic parameters.

This steps are finally combined into a single projection operator $\mathcal{P}$:

$$p = \mathcal{P}(P, \vartheta)$$

where P is a 3D point and ϑ contains the intrinsic and extrinsic parameters of the omnidirectional vision sensor:

$$\vartheta = [a\ b\ \alpha_u\ \alpha_v\ u_o\ v_o]^T.$$

Notice that the projection equations for the hyperbolic and spherical mirrors are similar, except when the hyperbolic mirror configures the single projection centre case. Then the reflection point on the mirror surface is linearly related to the 3D point and is found with a simple closed formula.

Design Steps

Let the most marginal point seen by a pin-hole camera with the field of view θ be, without loss of generality, at $(1, F(1))$:

$$F(1) = 1/tan\theta$$

and set at that point the vertical view field using Eq.(2.5):

$$F'(1) = tan(\phi - \theta)/2.$$

Then replacing F and F' by their expressions we obtain the system of equations:

$$\frac{a}{b}\frac{1}{\sqrt{b^2+1}} = \frac{\tan(\phi-\theta)}{2} \quad and \quad \sqrt{a^2+b^2} + \frac{a}{b}\sqrt{b^2+1} = \frac{1}{\tan\theta}$$

whose solution gives the hyperboloid parameters a, b. Notice that we took the single projection centre case, $L = \sqrt{a^2+b^2}$, but it similarly could have taken other cases such as the constant gain mirror, $L = 0$.

Now say that the minimal focusing distance, as given by the lens, is D_{min}. Then scale the system by $k = D_{min}/F(0)$ using the scaling property $t_2 = kt,\ F_2(t_2) = kF(t)$, which is the same as scaling both the hyperboloid parameters, a, b by k:

$$(a, b) \leftarrow (ka, kb).$$

As in the case of the spherical mirror, additional scaling up may be useful to obtain other mirror sizes and to decrease the pin-hole camera self-occlusion. The calibration of the parameters is expressed using also an optimisation method.

We designed an omnidirectional camera based on an hyperbolic mirror, once more for the navigation application. There are constraints on the system size which must be kept as small as possible to be carried by a small robot. Due to the small size of the robot it is convenient to place sensor resolution further away. The hyperbolic mirror is known to have this advantage as compared to the spherical one.

To chose the system vertical view angle, ϕ we took into consideration that for a small

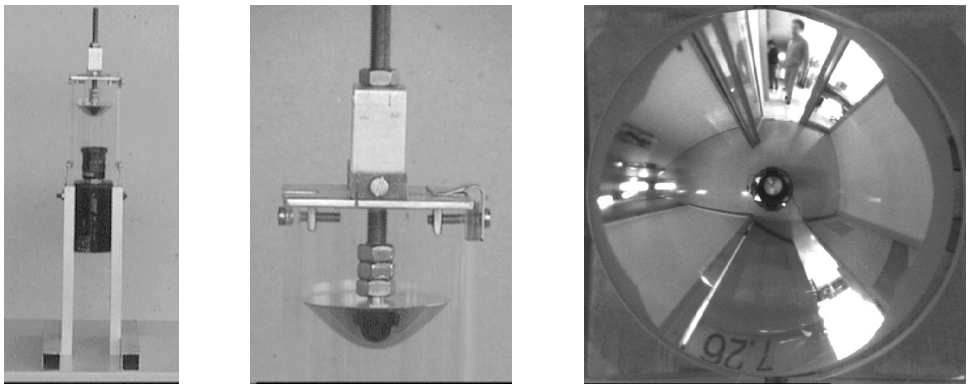


Figure 2.8: Omnidirectional system using a hyperbolic mirror (left), close up of the hyperbolic mirror (middle) and omnidirectional image (right).

robot it is desirable to look higher, i.e. to have a large field of view . Since the sensor resolution is finite, the larger field of view implies a coarser resolution for the world structure seen by the system. Also as the view angle raises, and looks higher than the horizon, the probability of blinding the sensor by direct illumination also increases. So a compromise was taken and the vertical view angle was chosen to be around $+20^o$ above the horizon line.

The pin-hole camera view angle, θ depends essentially on the chosen lens. In this case common off-the-shelf lenses were used, $f = 8mm$ and $f = 16mm$.

The hyperbolic mirror rim radius, r is controlled to minimize the size of the setup. As stated previously, the metric values of the setup can be scaled to the desired size. Smaller mirror sizes are interesting because weight is expensive in mobile platforms. On the other hand smaller mirrors require better manufacturing quality in order to preserve the projection quality for the various mirror sizes. Also smaller mirrors imply more care when assembling the camera and mirror since the tolerance to position the camera projection center over the hyperbolic mirror focus is smaller. Special care must be taken too with camera (lens) self occlusion: scaling down the mirror size makes self-occlusion more significant and so there is a compromise between system size and closest viewable ground point. The chosen value for the mirror rim radius was 25mm. Figure 2.8 shows the designed camera.

2.2.5 Comments on *no single projection centre*

Designing a system to have the single projection centre property is convenient as the sensor will have a simplified model. However, there are only a few cases of mirrors and cameras conforming to this property.

As previously stated, catadioptric cameras based on a spherical mirror do not have a single projection centre. The same happens in other cases such as when using conic mirrors, where the locus of projection centres is a 3D circle, or with hyperbolic mirrors whenever the pin-hole is not exactly placed at the mirror focus. Therefore, the situation

of non-single projection centre is quite common in practice.

When a catadioptric system has no single projection centre it is true that in general radial disparities vary non-linearly with the distance to the image centre and the correction cannot be automatically done as it depends on the observed scene. However there are other measurements that may still be taken. For example, when the mirror axis is aligned with the camera axis, azimuthal angular measurements may be taken.

It happens also that in many applications there is no significant difference between the single projection centre and the non-single projection centre cases. Usually when the observed 3D data is much far away than the size of the mirror, good approximations can be found [33]. Therefore, we may say that designing and building catadioptric omnidirectional cameras with no single projection centre, e.g. using spherical mirrors, is worthwhile since they allow for the same measurements to be taken.

Cost could be an important factor. For example, a spherical mirror is certainly less expensive than an hyperboloid. At the same time, due to the convenience of the single projection centre property on geometric modelling, it makes sense also to investigate and test whether the property (approximately) holds for the non-single projection centre systems.

This allows for design flexibility together with simplified modelling. In this way models based on the single projection centre property may become the most common, in the same way as the pin-hole model is used for standard cameras even when it is just an approximation valid for the tasks at hand.

Up to this point we presented geometric models and methods for designing / calibrating omnidirectional cameras. Now we can use the models for processing omnidirectional images in order to build representations of the world convenient for navigation tasks or human-robot interfaces. This is described in the following sections.

2.3 Image Dewarpings for Scene Modelling

Images acquired with an omni-directional camera, e.g. based on a spherical or hyperbolic mirror, are naturally distorted. For instance, a corridor appears as an image band of variable width and vertical lines are imaged radially. Knowing the image formation model, we can correct some distortions to obtain Panoramic images or Bird's Eye Views.

In a panoramic image, each scan line contains the projections of all visible points at a constant angle of elevation. Hence, the dewarping consists of mapping concentric circles to lines [12]. For example, the horizon line is actually transformed to a scan line.

Bird's eye views are obtained by radial correction around the image center. The bird's eye view is a scaled orthographic projection of the ground plane, and significantly simplifies the navigation system. In these images, straight lines of the ground plane are imaged with no deformation, and are therefore simpler to track.

Since the image centre is required for both the panoramic and Bird's Eye View dewarpings, we start with its estimation process.

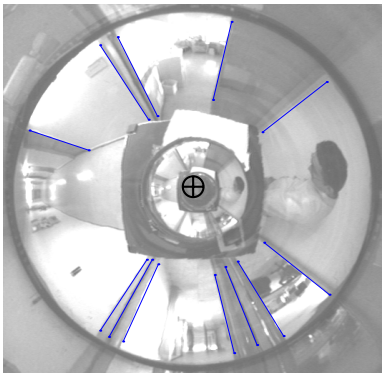


Figure 2.9: Image centre, marked with the sign $\bigoplus$, estimated from the intersection of radial lines corresponding to 3D vertical lines.

2.3.1 Image centre

To estimate the image centre we select image lines known to be projections of 3D vertical lines. Those lines are defined by pairs of image points and are known to be radial relative to the image centre, thus allowing image centre estimation (fig.2.9).

Defining a line by a point, (u_l, v_l) and a vector $(\Delta u_l, \Delta v_l)$, the image centre, (u_0, v_0) is found scaling the vector by the appropriate factor k_l:

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} u_l \\ v_l \end{bmatrix} + k_l \begin{bmatrix} \Delta u_l \\ \Delta v_l \end{bmatrix}. \tag{2.15}$$

Given two or more lines, allows us to find the image centre. With more than two lines a least squares solution is found on the vector scaling factors and on the desired image centre. Figure 2.9 shows an example of finding the image centre.

2.3.2 Panoramic View

In the case where the camera axis is vertical and aligned with the mirror axis[2] passing through the mirror centre, 3D points at same height and at same distance from the catadioptric omnidirectional vision sensor, project to a 2D circle in the image. 3D points at different heights at same distance, produce different concentric circles. Higher 3D points project to outer 2D circles and lower 3D points project to inner circles. Outer and inner circles map respectively to top and bottom lines of the dewarped image.

The image dewarping is defined simply as:

$$I(u, v) = I_0\left(R\cos(\alpha) + u_0, R\sin(\alpha) + v_0\right)$$

where (u_0, v_0) is the image centre, α is a linear function of u taking values in $[0,\ 2\pi]$ and R is a linear function of v sweeping the radial coordinate to cover all the effective omnidirectional image area. The number of columns of the resulting panoramic image

[2]In the case of the spherical mirror there are infinite mirror axis so there is no alignment requirement.

dependes on the minimum step of α. This step is chosen according to the number of pixels of the middle circle. Therefore, inner circles are over-sampled and outer circles are sub-sampled.

Figure 2.10 shows two examples of the remapping described in this section, to obtain panoramic views.

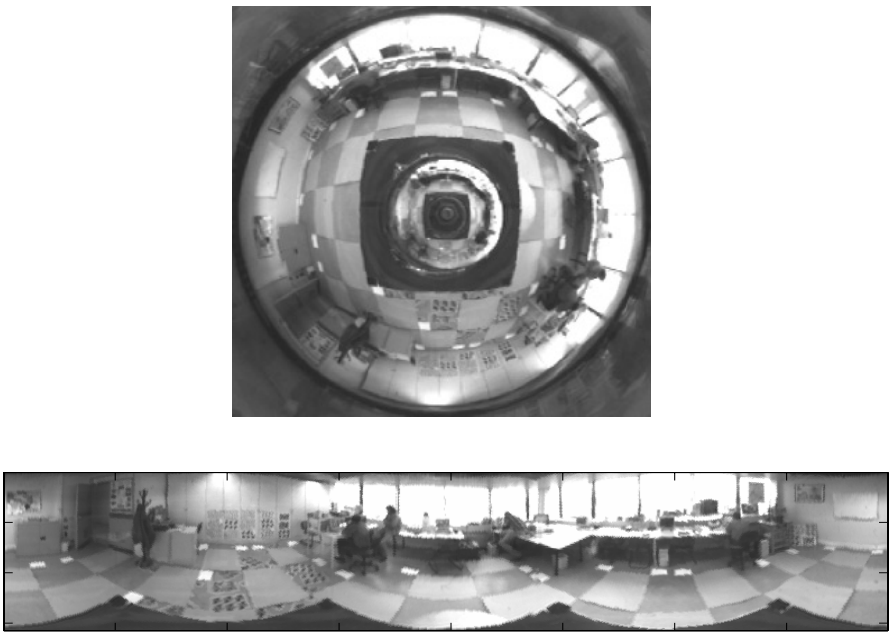


Figure 2.10: Illustration of the remapping for dewarping catadioptric image to panoramic view. (Top) original omnidirectional image, (bottom) dewarping result.

In summary:

- Panoramic images obtained with the described dewarping procedure only approximate perspective images. Usually the vertical sampling varies non-linearly relative to the 3D structure. Horizontal disparities observed in panoramic images, contrarily to perspective cameras, are measured on an non-planar surface. This must be taken in consideration for reconstruction tasks. One advantage of these images, is that horizontal angles may be measured on panoramic images without knowing the focal length of the lens.

- It is possible in the single projection centre case to make, for a chosen region of interest, a second dewarp which exactly recovers a perspective image. If there is no single projection centre then only approximate perspective images can be obtained. In practice this is not to restrictive since regions of interest of panoramic images are already good enough approximations to perspective images for many applications.

- As long as mirror, camera and mobile platform remain fixed to each other, the panoramic view dewarping is time invariant and can be programmed with a 2D lookup table.

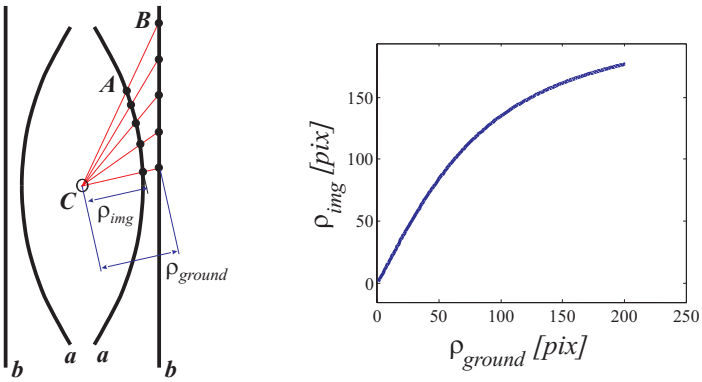


Figure 2.11: Image dewarping for bird's eye view. (Left) corridor guidelines, a as they are imaged in the omnidirectional image and, b the desired imaging in the bird's eye view. (Right) radial look up table. ρ_{img} and ρ_{ground} are distances measured from the image centre respectively in the original and dewarped images.

2.3.3 Bird's Eye View

The images acquired by our omnidirectional sensors are naturally distorted due to the geometry of the mirror and the perspective camera projection. Different world areas are mapped with different image resolutions. In general, 3D straight lines are projected as curves in the image. For instance, the horizon line is projected as an image circle. Only 3D lines that belong to vertical planes containing camera and mirror axis project as straight (radial) lines. In this section we present a method to dewarp a catadioptric omnidirectional image to a (orthographic) bird's eye view of the ground plane.

The azimuthal coordinate of a 3D point is not changed by the imaging geometry of an omnidirectional camera. Therefore, the dewarping of an omnidirectional image to a bird's eye view is a radial transformation. Figure 2.11-left shows that in order to dewarp the imaged corridor guidelines, a to the desired straight lines, b each point A needs to be transformed to B. Hence, we can build a 1D look up table relating a number of points at different radial distances on the omnidirectional image and the respective real distances. The 1D look up table is the radial transformation to be performed for all directions on an omnidirectional image in order to obtain the bird's eye view.

However, the data for building the look up table is usually too sparse. In order to obtain a dense look up table we use the projection model of the omnidirectional camera. Firstly, we rewrite the projection operator, $\mathcal{P}_\rho$ in order to map radial distances, ρ_{ground} measured on the ground plane, to radial distances, ρ_{img}, measured in the image:

$$\rho_{img} = \mathcal{P}_\rho(\rho_{ground}, \vartheta) \tag{2.16}$$

Using this information, we build a look up table that maps densely sampled radial distances from the ground plane to the image coordinates. Since the inverse function cannot be expressed analytically, once we have an image point, we search the look up table to determine the corresponding radial distance on the ground plane (see an example

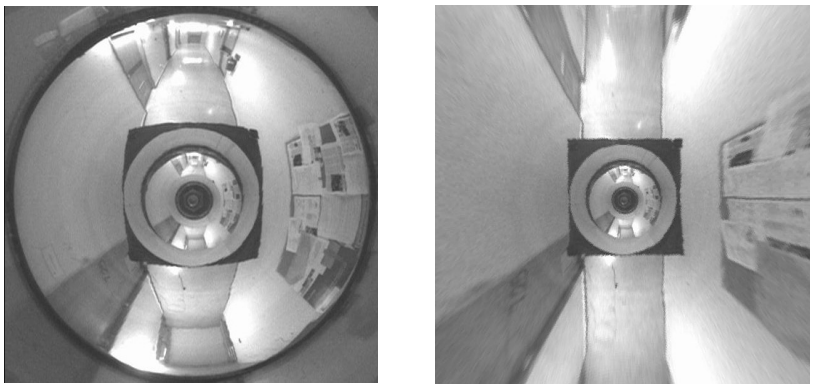


Figure 2.12: Image dewarping for bird's eye view. Left to right: original and dewarped images.

of a look up table in figure 2.11). The image dewarping to a bird's eye view is done efficiently in this way.

Figure 2.12 shows an example of the bird's eye view dewarping. The corridor guidelines are curved lines in the original omnidirectional image and become straight lines on the ground dewarped image, as desired.

As a final remark, notice that our process to obtain the look up table encoding the Bird's Eye View, is equivalent to perform calibration. However, for our purposes a good dewarping is simply the one that makes ground plane straight lines appear straight in the Bird's Eye View. This is much less than traditional calibration and therefore is easier to reach and check for.

2.3.4 Concluding Notes

In this section we presented image dewarpings for obtaining Panoramic and Bird's Eye Views from omnidirectional images.

Doing fixed image dewarpings is actually a way to do (or help) *Scene Modelling*. The image dewarpings make evident geometrical properties of the scene, such as vertical and ground plane straight lines, and thus simplify scene modelling to collecting a number of features.

In the following, we generalise the image dewarpings and detail how to obtain them directly by designing custom shaped mirrors.

2.4 Constant Resolution Cameras

The image formation model of a catadioptric omnidirectional camera is determined by the shape of the mirror. In some cases, one can design the shape of the mirror in such a way that certain world-to-image geometric properties are preserved - which we will refer to as *linear projection properties*.

The choice of the properties that should be preserved by the catadioptric imaging

system is naturally related to the specific application at hand: e.g. tracking and visual navigation. Specific details on the applications using these omnidirectional cameras can be found in [110, 20, 35]. The desired linear projection properties, can be categorised into three main types:

- **Constant vertical resolution** - This design constraint aims to produce images, where objects at a (pre-specified) fixed distance from the camera's optical axis will always be the same size in the image, independent of its vertical coordinates. As a practical example this viewing geometry would allow for reading signs or text on the surfaces of objects with minimal distortion. As another example, tracking is facilitated by reducing the amount of distortion that an image target undergoes when an object is moving in 3D. Finally, in visual navigation it helps by providing a larger degree of invariance of image landmarks w.r.t the viewing geometry.

- **Constant horizontal resolution** - The constant horizontal resolution ensures that the ground plane is imaged under a scaled Euclidean transformation. As such, it greatly facilitates the measurement of distances and angles directly from the image as well as easier tracking of points lying on the pavement thus having a large impact on robot navigation algorithms.

- **Constant angular resolution** - Here we wish to ensure uniform angular resolution simulating a camera with a spherical geometry. This sensor has interesting properties e.g. for ego-motion estimation [79].

Some of these designs have been presented in the literature, [12, 47, 14, 38] with a *different derivation* for each case. In this section, we present a *unified* approach that encompasses all the previous designs and allows for new ones. The key idea is that of separating the equations for the reflection of light rays at the mirror surface and the mirror *Shaping Function*, that explicits the linear projection properties to meet.

In some applications, one may be interested in having one type of projection property in a certain area of the visual field and other projection property in other areas of the visual field. We present a so-called *Combined Mirror* where the outer part of the image sensor is used to obtain a constant vertical resolution image, while the inner part is devoted to yield a constant horizontal resolution image. In this case, both constraints on the mirror shape resulting from the two design goals are combined together in a single profile.

Our general mirror design methodology is firstly developed for standard (cartesian) image sensors. However, a different choice of the sensor pixel layout may bring additional geometric and computational benefits for the resulting image/processing. Due to the rotational symmetry of the omnidirectional images, a natural choice is to use an image sensor with a polar structure. In this work we use the SVAVISCA [64] log-polar sensor developed at DIST, University of Genova. As a result of this mirror sensor combination, panoramic images can be directly read out from the sensor with uniform angular resolution, without

requiring any additional processing or image warping. Our general design methodology is applicable in exactly the same terms to this sensor layout.

This section is organised as follows: first we introduce the mirror shaping function and detail shaping functions which allows us to set specific constant resolution properties of the sensor. Then we detail the design of a combined mirror for a log-polar camera, instantiating some of the presented constant resolution properties. Finally we analyse the resulting mirror. Images obtained with the designed sensor are shown.

2.4.1 The Mirror Shaping Function

As detailed in section 2.2.2, the imaging by a normalised pin-hole camera of a 3D point reflected on an arbitrary mirror shape, is represented by a differential equation [47].

The differential equation, Eq.(2.6) here rewritten for the convenience of the reader, relates the mirror shape, F and its slope, F' which are functions of the mirror radius variable t:

$$\frac{\frac{t}{F} + 2\,\frac{F'}{1-F'^2}}{1 - 2\,\frac{tF'}{F\left(1-F'^2\right)}} = -\frac{r-t}{z-F} \tag{2.17}$$

where (r, z) is a generic 3D point. In order to numerically compute the solution of the differential equation, it is convenient to have the equation in the form of an explicit expression for F' [3].

Re-arranging Eq.(2.17) results in the following second order polynomial equation:

$$F'^2 + 2\alpha\,F' - 1 = 0 \tag{2.18}$$

where α is a function of the mirror shape, (t, F) and of an arbitrary 3D point, (r, z):

$$\alpha = \frac{-\left(z-F\right)F + \left(r-t\right)t}{\left(z-F\right)t + \left(r-t\right)F} \tag{2.19}$$

We call α the mirror *Shaping Function*, since it ultimately determines the mirror shape by expressing the relationship that should be observed between 3D coordinates, (r, z) and those on the image plane, determined by t/F. In the next section we will show that the mirror shaping functions allow us to bring the desired linear projection properties into the design procedure.

Concluding, to obtain the mirror profile first we specify the shaping function, Eq.(2.19) and then solve Eq.(2.18), or simply integrate:

$$F' = -\alpha \pm \sqrt{\alpha^2 + 1} \tag{2.20}$$

where we choose the + in order to have positive slopes for the mirror shape, F.

[3] Having an explicit formula for F' allows to directly use matlab's ode45 function

2.4.2 Setting Constant Resolution Properties

Our goal is to design a mirror profile to match the sensor's resolution in order to meet, in terms of desired image properties, the application constraints. As shown in the previous section, the shaping function defines the mirror profile, and here we show how to set it accordingly to the design goal.

For constant resolution mirrors, we want some world distances, D, to be *linearly* mapped to (pixel) distances, p, measured in the image sensor:

$$D = a_0.p + b_0. \tag{2.21}$$

for some values of a_0 and b_0 which mainly determine the visual field.

When considering conventional cameras, pixel distances are obtained by scaling metric distances in the image plane, ρ. In addition, knowing that those distances relate to the slope t/F of the ray of light intersecting the image plane as:

$$\rho = f.\frac{t}{F}, \tag{2.22}$$

the linear constraint may be conveniently rewritten in terms of the mirror shape as:

$$D = a.t/F + b \tag{2.23}$$

Notice that the parameters a and b can easily be scaled to account for a desired focal length, thus justifying the choice $f = 1$.

In the following sections, we will specify which 3D distances, $D(t/F)$, should be mapped linearly to pixel coordinates, in order to preserve different image invariants (e.g. ratios of distances or angles in certain directions).

Constant Vertical Resolution

The first design procedure aims to preserve relative vertical distances of points placed at a fixed distance, C, from the camera's optical axis. In other words, if we consider a cylinder of radius, C, around the camera optical axis, we want to ensure that ratios of distances, measured in the vertical direction along the surface of the cylinder, remain unchanged when measured in the image. Such invariance should be obtained by adequately designing the mirror profile - yielding a constant vertical resolution mirror.

The derivation described here follows closely that presented by Gaechter and Pajdla in [31]. The main differences consist of (i) a simpler setting for the equations describing the mirror profile and (ii) the analysis of numerical effects when computing the derivatives of the mirror-profile to build a quality index (section 2.4.4). We start by specialising the linear constraint in Eq.(2.23) to relate 3D points of a vertical line l with pixel coordinates (see Fig.2.13):

$$z = a.t/F + b,\ r = C.$$

Inserting this constraint into Eq.(2.19) we obtain the specialised shaping function:

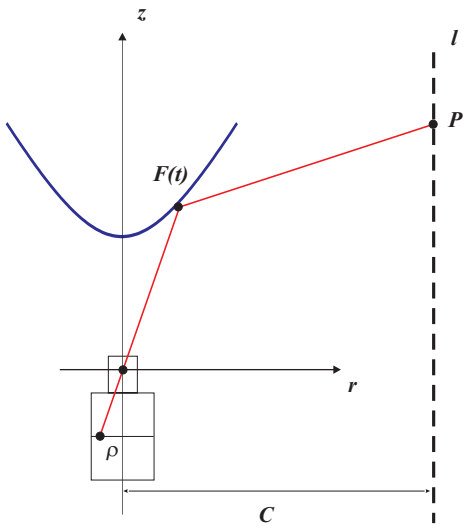


Figure 2.13: Constant vertical resolution: points on the vertical line l are linearly related to their projections in pixel coordinates, p.

$$\alpha = \frac{-\left(a\frac{t}{F} + b - F\right)F + (C-t)\,t}{\left(a\frac{t}{F} + b - F\right)t + (C-t)\,F}. \tag{2.24}$$

Hence, the procedure to determine the mirror profile consists of integrating Eq.(2.20) using the shaping function of Eq.(2.24), while t varies from 0 to the mirror radius.

The initialization of the integration process is done by computing the value of $F(0)$ that would allow the mirror rim to occupy the entire field of view of the sensor.

Constant Horizontal Resolution (*Bird's Eye View*)

Another interesting design possibility for some applications is that of preserving ratios of distances measured on the ground plane. In such a case, one can directly use image measurements to obtain ratios of distances or angles on the pavement (which can greatly facilitate navigation problems or visual tracking). Such images are also termed *Bird's eye views*.

Figure (2.14) shows how the ground plane, l, is projected onto the image plane. The camera-to-ground distance is represented by $-C$ (C is negative because the ground plane is lower than the camera centre) and r represents radial distances on the ground plane. The linear relation to image pixels is therefore expressed as:

$$r = a.t/F + b,\ z = C;$$

The linear constraint inserted into Eq.(2.19) yields a new shaping function:

$$\alpha = \frac{-(C-F)\,F + \left(a\frac{t}{F} + b - t\right)t}{(C-F)\,t + \left(a\frac{t}{F} + b - t\right)F} \tag{2.25}$$

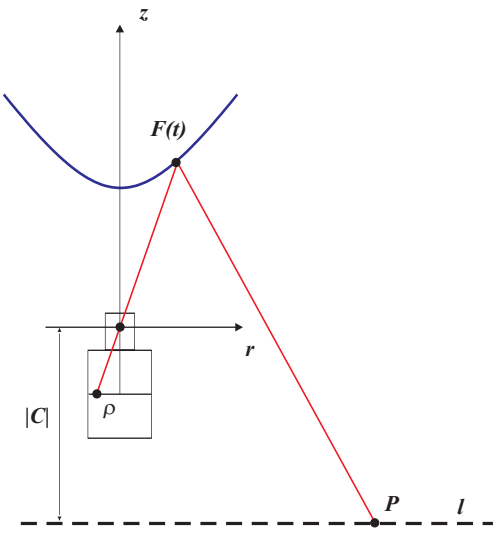


Figure 2.14: Constant horizontal resolution: points on the horizontal (radial) line l are linearly related to their projections in pixel coordinates, p.

that after integrating Eq.(2.20) would result in the mirror profile proposed by Hicks and Bajcsy [47].

Constant Angular Resolution

One last case of practical interest is that of obtaining a linear mapping from 3D points spaced by equal angles to equally distant image pixels, i.e. designing a constant angular resolution mirror.

Figure 2.15 shows how the spherical surface with radius C surrounding the sensor is projected onto the image plane. In this case the desired linear property relates angles with

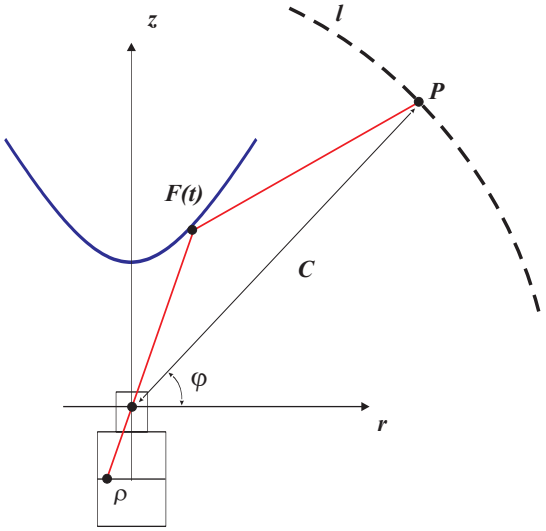


Figure 2.15: Constant angular resolution: points of the line l on the surface of a sphere of radius C, are linearly related to their projections in pixel coordinates, p.

image points:

$$\varphi = a.t/F + b$$

and the spherical surface may be described in terms of r, z, the variables of interest in Eq.(2.19), simply as:

$$r = C.cos(\varphi),\ z = C.sin(\varphi).$$

Then, replacing into Eq.(2.19) we finally obtain:

$$\alpha = \frac{-\left(C\sin(a\frac{t}{F}+b)-F\right)F + \left(C\cos(a\frac{t}{F}+b)-t\right)t}{\left(C\sin(a\frac{t}{F}+b)-F\right)t + \left(C\cos(a\frac{t}{F}+b)-t\right)F} \tag{2.26}$$

Integrating Eq.(2.20), using the shaping function just obtained (Eq.(2.26)), would result in a mirror shape such as the one of Chahl and Srinivasan [12]. The difference is that in our case we are imposing the linear relationship from 3D vertical angles, φ directly to image points, $(t/F, 1)$ instead of angles relative to the camera axis, $atan(t/F)$.

Shaping functions for Log-polar Sensors

Log-polar cameras are imaging devices that have a spatial resolution inspired in the human-retina. Contrarily to standard cameras, the resolution is not constant on the sensing area. More precisely, the density of the pixels is higher in the centre and decays logarithmically towards the image periphery. The organisation of the pixels also differs from the standard cameras, as a log-polar camera consists of a set of concentric circular rings, each one with a constant number of pixels (see Appendix A).

Advantageously, combining a log-polar camera with a convex mirror results in an omnidirectional imaging device where the panoramic views are extracted directly due to the polar arrangement of the sensor. Moreover, the improved central resolution of log-polar cameras allows to obtain panoramic images with better quality in the lines extracted close to the centre of the sensor. These reasons motivated our study on constant resolution cameras based on log-polar cameras.

In the following, we approach the design of constant resolution omnidirectional cameras based on log-polar cameras, starting from the derivations for standard cameras. Firstly we derive the desired linear relationships, and as expected the main difference is introduced by the log-polar cameras' pixel distribution. The rule observed for the linear relationships can be transported directly to the shaping functions and thus we obtain straightforward all the constant resolution designs considered for standard cameras.

In a log-polar camera, the relation of the linear distance, ρ, measured on the sensor's surface and the corresponding pixel coordinate, p, is specified by:

$$p = \log_k(\rho/\rho_0) \tag{2.27}$$

where ρ_0 and k stand for the fovea radius and the rate of increase of pixel size towards the periphery.

As previously stated, our goal consists of setting a linear relationship between world distances (or angles), D and corresponding (pixel) distances, p (see Eq.(2.21)). Combining into the linear relationship the pin-hole model, Eq.(2.22) and the logarithmic law defining pixel coordinates, Eq.(2.27) results in the following constraint:

$$D = a.\log(t/F) + b \tag{2.28}$$

It is interesting to note that the only difference in the form of the linear constraint when using conventional or log-polar cameras, equations (2.23) and (2.28), is that the slope t/F is replaced by its logarithm. Replacing the slope by its log directly in the shaping functions, results in the desired expressions for the log-polar camera.

Hence, when using a log-polar camera the shaping function becomes, for the case of constant vertical resolution:

$$\alpha = \frac{-\left(a.log\frac{t}{F} + b - F\right)F + (C-t)\,t}{\left(a.log\frac{t}{F} + b - F\right)t + (C-t)\,F}, \tag{2.29}$$

for the case of constant horizontal resolution:

$$\alpha = \frac{-(C-F)\,F + \left(a.log\frac{t}{F} + b - t\right)t}{(C-F)\,t + \left(a.log\frac{t}{F} + b - t\right)F}, \tag{2.30}$$

and finally, for the case of constant angular resolution:

$$\alpha = \frac{-\left(Csin(a.log\frac{t}{F} + b) - F\right)F + \left(Ccos(a.log\frac{t}{F} + b) - t\right)t}{\left(Csin(a.log\frac{t}{F} + b) - F\right)t + \left(Ccos(a.log\frac{t}{F} + b) - t\right)F}. \tag{2.31}$$

As for the case of standard cameras, the mirror shape results from the integration of Eq.(2.20), only now using the above shaping functions.

Concluding, we obtained a design methodology of constant resolution omnidirectional cameras, that is based on a shaping function whose specification allows to choose the particular linear property. This methodology generalises a number of published design methods for specific linear properties. For example the constant vertical resolution design results in a sensor equivalent to the one of Gaechter et al [31]. It is of particular interest the constant angular resolution sensor, as it would be an implementation of a spherical sensor giving a constant number of pixels per solid angle. This is similar to the case of Conroy and Moore [14] with the difference that due to the nature of the log-polar camera we do not need to compensate for lesser pixels when going closer to the camera axis.

2.4.3 Combining Constant Resolution Properties

In some applications, one may be interested in having different types of mappings for distinct areas of the visual field. This is typically the case for navigation where the constant vertical resolution mirror would facilitate the tracking of vertical landmarks, while the *Bird's eye view* would make localization with respect to ground plane features easier.

For this reason, we have designed and manufactured a combined mirror design for the SVAVISCA log-polar camera (see Appendix A for details), where the central and inner parts of the sensor (fovea and inner part of the retina) are used for mapping the ground plane while the external part of the retina is used for representing the vertical structure. These three regions are represented respectively by R_1, R_2 and R_3 in Figure 2.16 and the corresponding mirror sections as M_1, M_2 and M_3. As detailed in the Appendix, the central

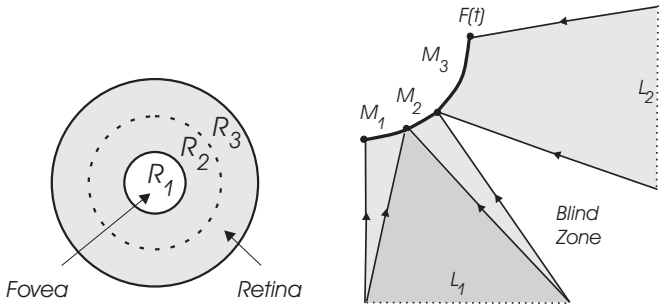


Figure 2.16: Combined mirror specifications. The horizontal line L_1 is reflected by the mirror sections M_1 and M_2, and observed by the sensor areas R_1 and R_2. The vertical line L_2 is reflected by M_3 and observed in R_3.

part of the SVAVISCA camera, the fovea, has equally sized pixels while the external part, R_2 and R_3 in the figure, has an exponential growth of the pixel size in the radial direction.

In the process of computing the mirror profile each of the parts is obtained individually using the shaping function encompassing the desired constant resolution property and the local pixels distribution. Therefore, for (R_3, M_3) we used the shaping function as given in Eq.(2.29) to impose constant vertical resolution in the case of a log-polar camera, while for (R_1, M_1) and (R_2, M_2) the expressions were respectively Eq.(2.25) and Eq.(2.30) to impose constant horizontal resolution for equally sized and exponentially growing pixels.

The field of view for each part of the sensor is determined by the corresponding parameters, a and b, which determine the vertical/horizontal segments that must be mapped onto the image. Conversely minimum and maximum distances on the ground, heights on the vertical direction or angles to points on a sphere, determine the a, b parameters.

Figure 2.17 shows the obtained mirror profile comprising the three sections, the first two designed to observe the ground plane within distances $48cm$ to $117cm$ from the sensor axis, and the third one to observe -10^0 to $+25^0$ around the horizon line. The camera height above the floor was defined as $70cm$, the radius of the cylinder as $200cm$ and the focal length used was $25mm$. The mirror radius was set to $3cm$.

2.4.4 Analysis of the Mirrors and Results

The combined mirror described in the preceding section is composed as three parts. Here we analyze the quality of the outer part, designed to have constant vertical resolution. The other two parts would have similar analysis.

There are two main factors of main influence:

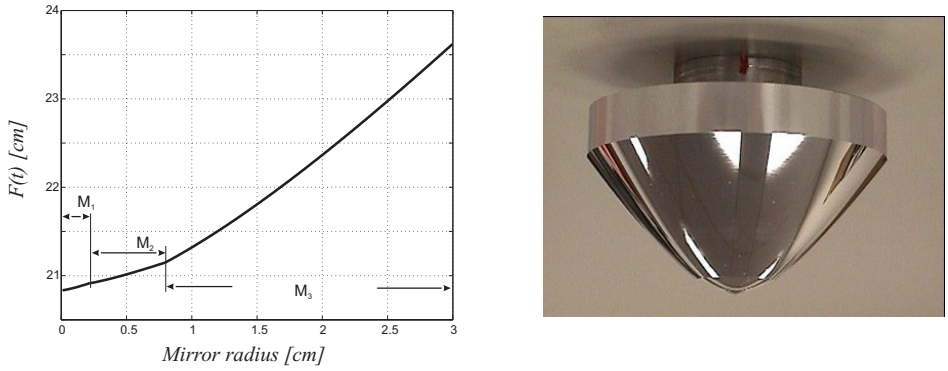


Figure 2.17: Combined mirror for the Svavisca log-polar image sensor. Computed profile (left) and manufactured mirror (right).

- Numerical Errors - As we do not have an analytic description of the mirror shape and as the actual profile is obtained through numerical integration it is important to verify the influence of numerical integration errors in the overall process.
- Sensitivity - As the designed sensor does not have a single center of projection, the linear mappings obtained between pixel distances and world distances are only valid for specific world surfaces (e.g. specific vertical cylinders or horizontal planes in our case). How do the linear projection properties degrade for objects laying at distinct distances other than those considered for the design ?

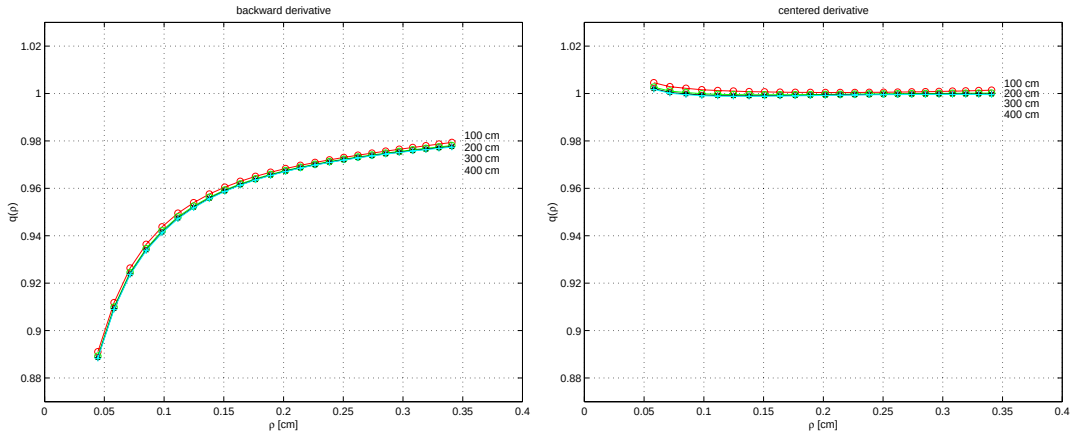


Figure 2.18: Analysis of the design criterion for different distances with a log-polar image sensor using different numeric approximations to the derivative: backward (left) and centered (right) differences.

As proposed in [31], the analysis of the mirror profile is done by calculating a quality index, $q(\rho)$. This quality index is defined as the ratio between the numerical estimate of the rate of variation of the 3D distances, D, with respect to distances in the image plane,

$[\partial D/\partial\rho]_n$, and the corresponding theoretical value, $[\partial D/\partial\rho]_t$:

$$q(\rho) = \frac{[\partial D(\rho)/\partial\rho]_n}{[\partial D(\rho)/\partial\rho]_t} \tag{2.32}$$

In the case under analysis, the theoretical value is obtained from Eq.(2.28) (noting that $t/F = \rho/f$) and the numerical value from the back-projection [45], which results directly from Eq.(2.17) given that the mirror shape is known.

For the perfect design process we should have $q(\rho) = 1$. Computing $q(\rho)$ involves numerically differentiating the profile $F(t)$. Figure (2.18) shows some results obtained with different discrete approximations to derivatives.

These results show two main points. Firstly, the influence of varying distance with respect to the desired mapping properties does not seem to be too important, which suggests that we are close to the situation of a single projection centre. Secondly, the way derivatives are computed is very important in terms of quality analysis. The simplest form of numerical differentiation leads to an error of about 10%, while a better approximation shows that the computed profile meets the design specifications up to an error of about 1%. Variations when the distance changes from the nominal $d = 200cm$ to $1m$ or $4m$ are not noticeable.

Figure 2.19 shows the combined mirror assembled with the camera and mounted on top of a mobile robot. The world scene contains vertical and ground patterns to test for the linear properties. Figure 2.20 shows an image as returned by the camera.

Figure 2.19: Svavisca camera equipped with the combined mirror (left) and world scene with regular patterns distributed vertically and over the floor (right).

Figure (2.21) shows resulting images. The panoramic image results as a direct read out from the sensor (see Fig.2.21, top) and the bird's eye views are obtained after a change from cartesian to polar coordinates (Fig.2.21, bottom left and right). In the panoramic image the vertical sizes of black squares are equal to those of the white squares, thus showing linearity from 3D measures to image pixel coordinates. In the bird's eye views the rectilinear pattern of the ground was successfully recovered (the longer side is about twice the size of the shorter one).

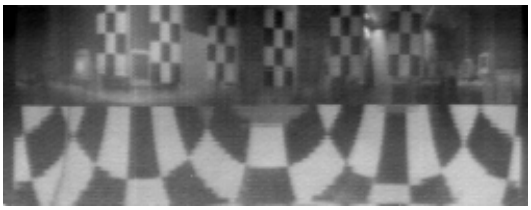


Figure 2.20: Image acquired with the Svavisca camera equipped with the Combined Mirror.

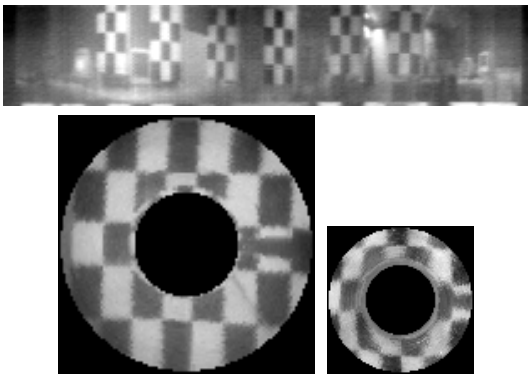


Figure 2.21: Images obtained from the original image in Fig.2.20: (top) panoramic and (bottom) bird's eye views. The bird's eye views have a transformation from cartesian to polar coordinates. The bird's eye view at right originated from the fovea area.

Figure (2.22) shows another panoramic image, where some of the vertical chess patterns of the world scene were placed closer to the sensor. Placing the patterns closer did not make significant changes to the constant resolution property, as predicted in the numerical analysis.

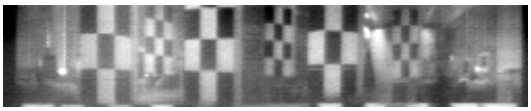


Figure 2.22: Chess patterns placed closer to the robot.

2.4.5 Concluding Notes

In this section we have described a general methodology for the design and evaluation of mirrors profiles encompassing desired constant resolution properties. A function defining the mirror shape, the *Shaping Function*, was introduced and it was shown how to derive formulas to achieve the following specifications:

- Constant *vertical* resolution mirror - distances measured in a vertical line at a fixed distance from the camera axis, are mapped linearly to the image, hence eliminating any geometric distortion.

- Constant *horizontal* resolution mirror - where radial lines on the floor are mapped linearly in the image.
- Constant *angular* resolution mirror - equally spaced points on a meridian of a sphere are mapped linearly in the image plane.

The methodology also considers the case of log-polar cameras. The difference from the standard camera case is an additional logarithmic relation that appears in the shaping function.

A prototype of a combined mirror was built. Here the outer part of the sensor's retina was used for constant vertical resolution design; the inner part of the sensor's retina was used to generate the constant horizontal resolution criterion and the log-polar sensor's fovea has also been used, in spite of the reduced number of pixels, to generate a bird's eye view of the ground floor. Resulting images show that the design was successful as the desired linear properties were obtained.

2.5 Approximating the Unified Projection Model

It is well known that the geometry of 3D projection light-rays can be derived independently of the scene-structure depth, provided that all the projection rays intersect at a single projection centre. This is convenient for instance for obtaining perspective images from omnidirectional images, or applying standard reconstruction methods. However, as noted in section 2.2.5, only a few cameras conform to the single projection centre property [2].

In this section we use an omnidirectional camera based on a spherical mirror and show that it can be approximated by a pin-hole camera. We described already the unified projection model [38], which is known to be equivalent to a pin-hole camera, and now we shall show that it can approximate the projection when using a spherical mirror [34].

Given the modelling of the camera with the unified projection model, we derive the geometry of projection light-rays, termed *back-projection* after Sturm in [97, 96], and then show how to obtain perspective images. These will be useful in a later chapter for reconstructing and texture mapping 3D scene models.

2.5.1 Unified projection model parameters

A camera with a spherical mirror cannot be exactly represented by the unified projection model. In order to find an approximate representation we focus upon the image projection error, instead of analysing the projection centre itself.

In section 2.2.3 we estimated the parameters of the projection model of the omnidirectional camera based on a spherical mirror. Therefore, now we can find the unified projection model parameters simply by minimizing the differences in the imaging of 3D test points by both models.

Let $\mathcal{P}(x_i, y_i, z_i; \vartheta)$ denote the unified projection model defined in Eq.(2.3) and $\mathcal{P}_c$ be the projection with a spherical mirror defined in Eqs.(2.6, 2.7, 2.10). Grouping into ϑ and

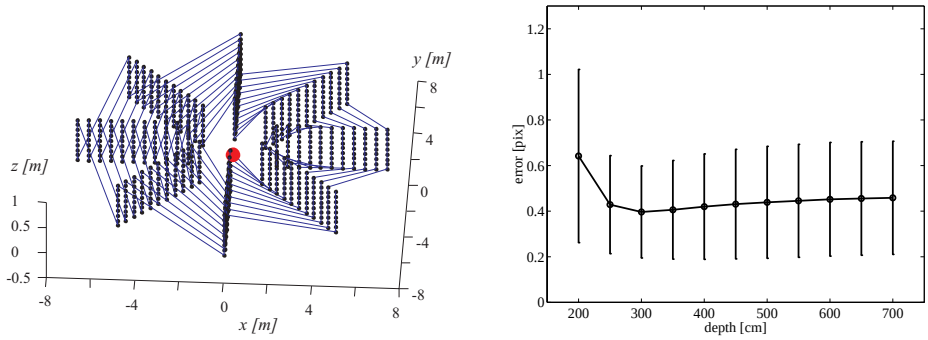


Figure 2.23: (Left) 3D test points. The large dot in the centre represents the omnidirectional camera. (Right) Mean absolute error between the unified projection model and the projection with the spherical mirror. Vertical bars indicate the standard deviation. Our omnidirectional images have 500x500 pixels.

ϑ_c the geometric and intrinsic parameters for the former and latter projections, we want to minimize a cost functional associated to the image projection error:

$$\hat{\vartheta} = \arg_{\vartheta} \min \sum_i \|\mathcal{P}(x_i, y_i, z_i; \vartheta) - \mathcal{P}_c(x_i, y_i, z_i; \vartheta_c)\|^2 \tag{2.33}$$

The minimization of the functional gives the desired parameters, ϑ, for the unified projection model, $\mathcal{P}$, that will approximate the real sensor characterised by $\mathcal{P}_c$ and ϑ_c.

Figure 2.23 shows that the approximation errors measured in the image plane are small by considering 3D points distributed around the sensor at several heights, in a range of 2 to 7m from the camera's optical axis.

2.5.2 Using back-projection to form perspective images

The acquisition of correct perspective images, independent of the scenario, requires that the vision sensor be characterised by a single projection centre. The unified projection model has, by definition, this property but, due to the intermediate mapping over the sphere, the obtained images are in general not perspective.

In order to obtain correct perspective images, the spherical projection must be first reversed from the image plane to the sphere surface and then, re-projected to the desired plane from the sphere centre. We term this reverse projection *back-projection*.

The back-projection of an image pixel (u, v), obtained through spherical projection, yields a 3D direction $k \cdot (x, y, z)$ given by the next equations derived from Eq.(2.3):

$$\begin{gathered} a = (l + m), b = (u^2 + v^2) \\ \begin{bmatrix} x \\ y \end{bmatrix} = \frac{la - sign(a)\sqrt{a^2 + (1 - l^2)b}}{a^2 + b} \begin{bmatrix} u \\ v \end{bmatrix} \\ z = \pm\sqrt{1 - x^2 - y^2} \end{gathered} \tag{2.34}$$

where z is negative if $|a|/l > \sqrt{b}$, and positive otherwise. It is assumed, without loss of generality, that (x, y, z) is lying on the surface of the unit sphere.

Figure 2.24 illustrates the back-projection. Given an omnidirectional image we use back-projection to map image points to the surface of a sphere centred at the camera viewpoint.

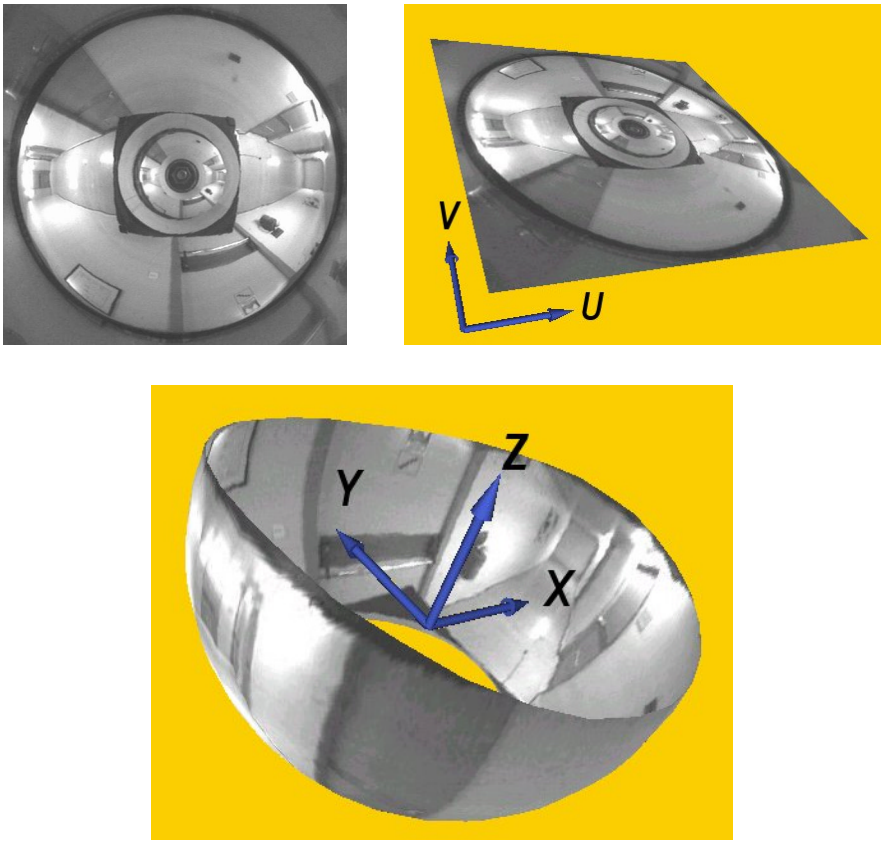


Figure 2.24: Top: original omnidirectional image and an alternative display where the axes are placed so that the first row appears close to the bottom. Bottom: back-projection to a spherical surface centred at the camera viewpoint.

At this point, it is worth noting that the set $\{(x, y, z)\}$ interpreted as points of the projective plane, already define a perspective image. However for the purpose of displaying or to obtain specific viewing directions further development is needed.

Let R denote the orientation of the desired (pin-hole) camera relative to the frame associated to the results of back-projection, the new perspective image $\{(\lambda u, \lambda v, \lambda)\}$ becomes:

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = K \cdot R^{-1} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.35}$$

where K contains intrinsic parameters and λ is a scaling factor. This is the pin-hole camera projection model [25], when the origin of the coordinates is the camera centre.

Figure 2.25 shows perspective images obtained from the omnidirectional image shown

in figure 2.24. The perspective images illustrate the selection of the viewing direction and the focal length.

Figure 2.25: Examples of perspective images obtained from the omnidirectional image shown in Fig.2.24. On the top row the perspectives have differing directions while on the bottom row we simulated three typical focal lengths, 4.5mm, 8mm and 16mm for a 6.4x4.8mm ccd.

2.6 Concluding Notes

In this chapter we presented geometric models for catadioptric omnidirectional cameras, together with design methodologies for the most frequent cameras based on spherical and hyperbolical mirrors.

We built one omnidirectional camera based on a spherical mirror with a view field to see above the horizon line. A camera based on an hyperbolic mirror was built with the additional goal of size minimization.

Images acquired using non-planar mirrors are naturally distorted. Some of the distortions may be predicted and corrected. Dewarping to obtain panoramic and bird's eye views are two useful examples. They allow e.g. simple tracking of vertical and ground plane lines. We presented our estimation methods for the two dewarpings and for the image centre which is required in both cases.

We also presented the constant resolution cameras. Here we proposed a unifying design methodology for the most typical cases of constant vertical, horizontal and angular resolution. The unifying approach is based on a shaping function whose specialization allows for the selection of the various constant resolution properties.

A prototype of a combined mirror has been built where the outer part of the sensor's retina is used for constant vertical resolution design; the inner part of the sensor's retina is used to generate the constant horizontal resolution criterion and the log-polar sensor's fovea has also been used, in spite of the reduced number of pixels, to generate a bird's eye view of the ground floor. Resulting images show that the design was successful as the desired linear properties were obtained.

Finally, we presented an approximation of the unified projection model to the omni-

directional camera based on a spherical mirror. Despite not having the single projection centre property, the geometry of the omnidirectional camera is well described by the unified model in a large 3D region of interest. We used this modelling for obtaining the back-projection equations of image points, independent of the 3D world structure.

The omnidirectional cameras detailed in this chapter are the sole sensors we use for developing our mobile robot navigation methodologies. The geometrical modelling of the cameras and the dewarpings of the omnidirectional images, are useful when extracting world-structure information for navigation. More specifically, the panoramic and bird's eye views will be used latter in the design of the navigation modalities (chapters 3 and 4) because of the convenient and simple ways they represent the world structure. The back-projection equations, which are required for instance for obtaining perspective images from omnidirectional images or applying standard reconstruction methods, will be used in the interactive reconstruction methods detailed in chapter 5.

Chapter 3

Visual Path Following

Visual Path Following is described in simple terms as a trajectory following behaviour, without having the trajectory explicitly identified in the scene. The trajectory exists only as a computer data structure.
In this chapter we show that Self-Localisation *is a major component of visual path following and that the ground dewarp representation significantly simplifies the solution to self-localisation problems, since the image coordinates differ from ground coordinates by a simple scale factor, thus eliminating any perspective effects. Of particular importance is the use of carefully designed low-level image processing processes.*
Experiments with a mobile robot equipped with an omnidirectional vision sensor are detailed showing the validity of the proposed path following method.

3.1 Introduction

Many real world mobile robot applications, such as home assistance or office mail delivery require high levels of autonomy, but unlike industrial applications, the environment should not be altered to suit the given task. Typically, robot autonomy should cover large areas, for example several offices or an entire house. Building a complete metric description of such large areas, is expensive in computing power and in sensor allocation.

Alternative solutions [59, 91] use qualitative and topological properties of the environment for navigation, mainly when the robot has to travel rather large distances. A different approach is necessary when in regions with precise guidance or localisation requirements, e.g. a door crossing. The work described in this chapter addresses such precise navigational problems, based on a method that we call *Visual Path Following*, to complement those systems based on topological maps.

Visual Path Following is a method whereby once a robot has arrived at the start of a previously specified path, it can perform path following to a given location relying on the visual tracking of features (landmarks).

In the visual servoing literature [51], when a robot is controlled through set-points (pose

values) it is said to be performing *Dynamic look and move*, while when directly using torque signals it is performing *Direct visual servoing* [24, 98]. At the image processing level, there is a clear distinction between *Position-based* and *Image-based control*, i.e. whether or not computing the pose.

As pointed out by Wunsch and Hirzinger [114], there are pros and cons for each of the visual servoing approaches: *Image based visual servoing* requires little computation and guarantees robustness against errors in sensor modelling and camera calibration; *Position based visual servoing* has the distinct advantage that both geometric and dynamic models can be included in a straightforward fashion to increase the robustness of the vision task.

Commanding robot torques directly from image measurement errors, i.e. integrating in a single step image processing and control signals computation, results in more robust systems. However, it is necessary to adhere to strict processing rates. It is important to note that usually the sampling frequency at the robot-motors level is one or two orders of magnitude larger than the image processing one. While image processing is typically limited to about 25Hz, robot control may range the 100Hz to 10KHz.

We design *Visual Path Following* based on a feedback control system encompassing a self-localisation module and a control module (see Fig.3.1). The control module computes the commanding signals for the robot, namely linear and angular velocities, based on the error distance obtained by comparing the robot's current pose, estimated by the self-localisation module, with the desired pose.

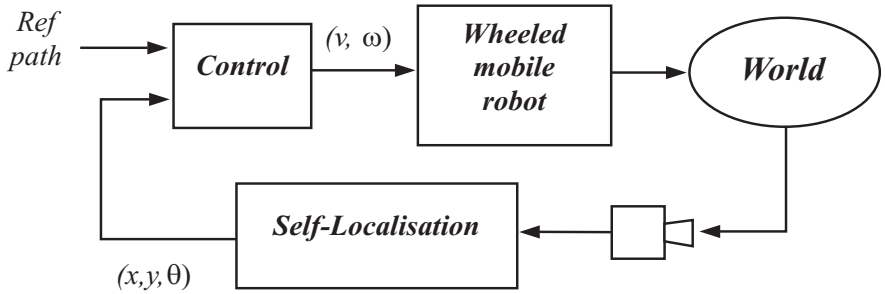


Figure 3.1: Visual Path Following flowchart.

Most of the current research in *localisation* relies upon the integration of complimentary sensors or concentrates on solving general problems like *Structure From Motion*, *Simultaneous Localisation And Map building*, calibration, etc. Our localisation approach is fully supported by use of monocular vision, and is aimed at increasing the contribution of vision to the sensor fusion paradigm.

There is ample and extensive work on vision-based self-localisation i.e. camera-pose computation (for example Gennery [37], Lowe [70], Koller et al [61], Taylor [103]). The main difference of our approach is that we combine sensor-design and localisation-algorithm development in order to solve the localisation problem.

In our work visual tracking is achieved using omnidirectional images acquired by the

Figure 3.2: Omnidirectional vision helps in preserving features in the field of view. Even after crossing the door, some of the features of the corridor continue to be visible.

omnidirectional cameras described in chapter 2. Omnidirectional Vision presents well known advantages for Self-localisation not only in algorithmic terms but also in scale terms. For example, algorithmic advantages are found in better landmark stability, as the landmarks stay for longer periods in the field of view (see Fig.3.2). Scale advantages appear e.g. in cases of people occluding the images: one person at a distance of one metre to the camera occludes about 20^o of the field of view, which represents 60% of the image of a standard camera equipped with a $12mm$ lens, as compared to a much smaller value of 6% of an omnidirectional camera.

Gluckman and Nayar in [40] note that traditional cameras, unlike the omnidirectional ones, often suffer from the problem that the direction of translation falls out of the field of view, which makes egomotion extremely sensitive to noise. In other words, we can say that omnidirectional cameras are beneficial for uncertainty reduction in egomotion estimation.

This idea has been computationally explained by Fermüller and Aloimonous [26]. Previous simulated experiments by Tian, Tomasi and Heeger [104] were already showing experimentally the advantage of very wide field of views for egomotion computation.

Given the close relationship of egomotion and self-localisation, it is also expected that the same property, i.e. uncertainty reduction when using omnidirectional cameras, holds for self-localisation. Madsen and Andersen show in [71] that there are sets of landmarks allowing for better accuracy at self-localisations. Our own simulations show that it is beneficial to select widely separated landmarks thus demonstrating an advantage of omnidirectional cameras on self-localisation (see appendix B).

In summary, we implement *Visual Path Following* as a feedback control system combining two major components, namely mobile robot *Control* and *Self-localisation*. Self-localisation is based on the tracking of straight line segments identified on omnidirectional images. The controller computes the robot's linear and angular velocities based on the estimated location and the desired trajectory. The trajectory is conveniently defined in images coordinates relative to a model of the scene defined upon edge line segments extracted from the images.

Chapter Organisation

Firstly we describe localisation. We introduce geometric scene models based on ground plane and vertical line segments, and detail our method for tracking the model line segments. Then we describe three pose computation methods that are used concurrently, and present a photometric criterium that allows tracking failure detection and defining the best / optimising the pose estimate.

After describing the localisation, we detail the robot control law which closes the loop that forms the Visual Path Following navigation modality.

Finally, we present localisation and control experiments. We test these modules both individually and combined to form the navigation modality.

3.2 Vision-based Self-localisation

Vision-based self-localisation derives robot poses from images. It encompasses two mostly relevant parts: image processing and pose-computation. Image processing provides the tracking of the features of the scene. Pose-computation is the geometrical computation that derives the robot pose from the observations of the scene features given the scene model.

Designing the image processing level involves modelling the environment. One way to inform a robot of an environment is to give it a CAD model, as in the work of Kosaka and Kak [62], recently reviewed in [22]. The CAD model usually comprises metric values that need to be scaled to match the images acquired by the robot. In our case, we overcome this need by defining geometric models composed of features of the environment directly extracted from images.

Pose-computation, as the robot moves in a plane, consists in estimating a 2D pose and an orientation. Assuming that the robot knows fixed points of the environment (landmarks) then there are two main methods of self-localisation relative to the environment: trilateration and triangulation [6]. Trilateration is the determination of a vehicle's position based on distance measurements to the landmarks. Triangulation has a similar purpose but is based on bearing measurements.

In general one single image taken by a calibrated camera provides only bearing measures. Thus, triangulation is the natural way to calculate self-localisation. However, there are some camera poses / geometries that provide more information. For example, a bird's eye view (detailed in section 2.3.3) provides an orthographic view of the ground plane, and allows to observe not only bearings but simultaneously distances to the landmarks laying over the floor. Given distances and bearings, the pose-computation is simplified to the calculation of a 2D rigid transformation. We estimate pose both using bearings only or bearings and distances.

Another point worth discussing, is the uncertainty in localisation introduced by the various processing steps from the images to the estimated pose. The fact that the pose-

computation is based on the feature locations, implies that it contains errors propagated from the feature tracking process. We propose then a complimentary pose-computation optimisation step, based on a photometric criterium. We term this optimisation fine pose adjustment, as opposed to the pose-computation based on the features which is termed coarse pose computation. It is important to note that the pose-estimation based on features is important for providing an initial guess for the optimisation step.

In the following sections firstly we detail the geometric models of the environment and the tracking of their features. Then we present the pose-computation methods. Pose-computation is further divided into coarse estimation and fine adjustment steps.

3.2.1 Scene Geometric Modelling and Tracking

Despite the fact that localisation can be based on tracked image corners [93], more robust and stable results are obtained with line segments as noted for example by Spetsakis and Aloimonos in [94]. Besides that, two concurrent lines define a corner, and so from the lines we can use corners based algorithms (i.e. using edge segments is not a hard-decision).

Geometric Scene-Model

Geometric models of the scene are collections of segments identified on Bird's Eye and Panoramic views. Ground segments are rigidly interconnected in the Bird's Eye views while vertical segments will vary their locations according to the viewer location. Considering both types of segments, the models are "wire-frames" whose links change according to the viewpoint.

Each scene model must have a minimal number of features (line segments) in order to allow self-localisation. One line of the ground plane permits finding only the orientation of the robot and gives a single constraint on its localisation. Two concurrent ground lines, or one ground and one vertical, already allow finding the robot position and orientation. Given three lines either all vertical, one on the ground, two on the ground (not parallel) or three on the ground (not all parallel), always permit computing the pose and therefore form valid models [1].

Our first experiments are based on a simple pattern, composed of two black rectangles on the ground plane. This pattern defines eight ground segments (see fig. 3.3) thus including redundancy to improve robustness. The landmark in the original omnidirectional image (left) is deformed according to the position on the image plane while in the bird's eye view (middle) it has a constant size and thus it is simpler to track. (right) The geometric model is extracted directly from the bird's eye view.

Based on the same idea of modelling the scene with line segments, it is possible to create scene-models for self-localisation but using natural features, i.e. features already present in the scene. Figure 3.4 shows one such example. The model is composed of three ground lines, two of which are corridor guidelines, and eight vertical segments essentially

[1] Assuming known the xy coordinates of the intersection of the vertical line(s) with the ground plane.

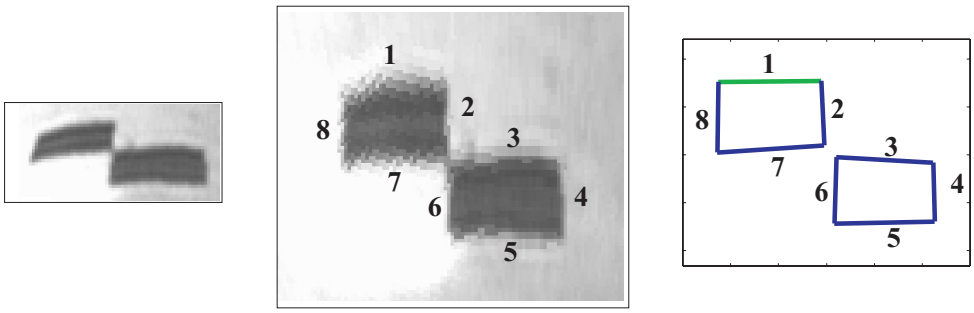


Figure 3.3: Geometric model of a ground plane landmark, composed by two black rectangles. The landmark in (left) the original omnidirectional image and in (middle) the bird's eye view. (right) The geometric model is extracted directly from the bird's eye view.

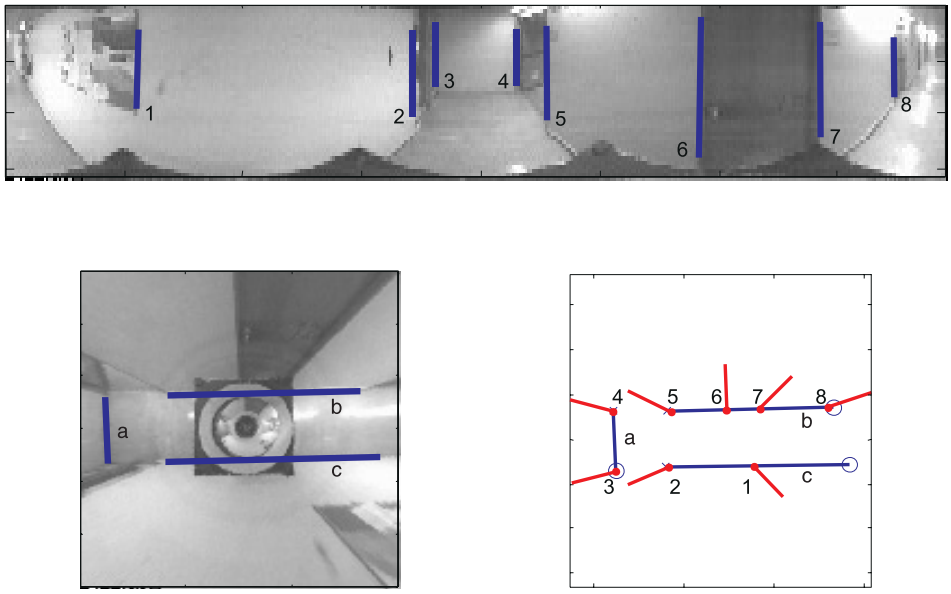


Figure 3.4: Geometric models for a door crossing experiment. In the panoramic and bird's eye view images, respectively top and bottom-left, are illustrated the segments composing the model shown in the bottom-right.

defined by the door frames. A single door frame (i.e. two vertical lines) and one corridor guideline would suffice but it is beneficial to take more lines than the minimum once more to improve robustness of the self-localisation.

In order to represent a certain scene area, a minimal number of segments will be necessary according to visibility and quality issues (see Talluri and Aggarwal in [102] for a geometrical definition of visibility regions).

Models characterising different world regions are related by rigid 2D transformations. These transformations are firstly defined between every two neighbour models at locations where both models are (partially but with enough relevance) visible. Navigation is therefore possible in the area composed as the union of the individual areas provided by the individual models.

Feature Tracking

Assuming that the robot pose evolves smoothly along time, the model segments need to be detected only once at the initialisation stage and from then on, it is only necessary to track them which is much more efficient in computational terms.

We track both edges lying on the ground plane and vertical edge segments. Notice that vertical lines project as radial (or vertical) lines, in the bird's eye view (or panoramic) images. Since the robot position and orientation are estimated relative to a pre-defined coordinate system, the process of tracking is simplified by utilizing bird's eye (orthographic) views of the ground plane, thus preserving angular measurements and uniformly scaling distances.

Edge segments are represented by 15 to 30 sampled points, that are tracked by searching the image perpendicularly to the edge segments (see fig.3.5(left)). Defining a coordinate system attached to a segment, where l and x are respectively the directions along and orthogonal to the segment, then the search is simplified to find the x_l representing the new x-locations for the segment points l:

$$\{\,(l,0)\,\} \quad \rightarrow \quad \{\,(l,x_l)\,\}\,.$$

The search criterion is based upon the evaluation of the image gradient and the distance to the original edge position. The image gradient absolute value, $|I_x(l,x)|$ should be maximum at the edge point and the sign, $sign(I_x(l,x))$ must be preserved along the tracking. Assuming small displacements for the segments, then it is expected that the new edge positions are still the closest ones to the originals. The preference for small distances of segment displacements is introduced by a triangle function, $\Lambda(x)$ with the maximum at zero and evaluating to zero at the extremes of the search region (see fig.3.5(right)). Combining all these constraints and priors the search problem is then:

$$x_l = \arg_x \max \left[h\left(|I_x(l,x)|\right) \;.\; \left(sign\left(I_x(l,x)\right) == s_0\right) \;.\; \Lambda(x) \right]$$

where h is a *non-maximum suppression* filter as the one of Canny's edge detector [9, 106]

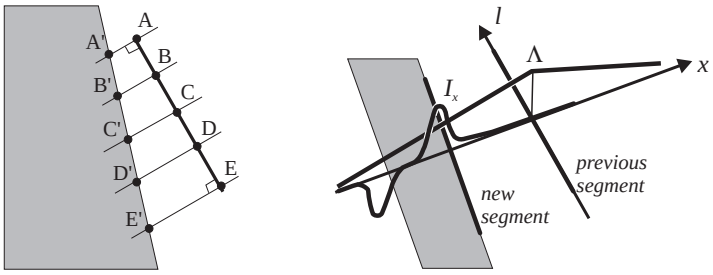


Figure 3.5: (Left) The segment is tracked at control points. (Right) The criterium to find the new location of an edge point is based on the local gradient and on the distance to the original point.

and s_0 is the sign of the gradient along the orthogonal direction that should be preserved during the tracking.

The just determined distances, x_l are corrupted with noise due e.g. to the image formation process or to eventual local occlusions. For this reason, we finally estimate the new segment location through the robust fitting procedure RANSAC [28].

As a concluding remark, notice that our procedure ignores segment lengths. They are controlled at a higher level through the consistency of the model, as it is too much error prune trusting on the image brightness for this purpose.

At the current stage of implementation, the relevant features to track and the feature co-ordinate system are initialised by the user.

Up to now, we have tracked individual segments. In the next section we show how to compute robot's motion, i.e. translation and rotation, from these data.

3.2.2 Pose computation

We utilise bird's-eye view and panoramic images to track environmental features so as to estimate the robot's pose or to drive the robot along a given trajectory. The self-localisation procedure is based on the tracking of the geometric models. The tracking of the models requires rigidity of the world structure (but naturally not rigidity of the observed model segments itself).

Self-localisation computations depend on the nature of the input data. An important distinction arrives for the availability (or not) of distance measurements. In the absence of distance measures, i.e. using bearings only, the calculations and map requirements are different. Therefore, a description of the pose computation methods given the combinations of distance and / or bearing observations is given. Associated to the pose computations there are also constraints on the number and location of features that need to be observed to avoid singularities in run-time.

When the models consist only of vertical lines and their projections onto the ground plane are known, then self-localisation may be based on bearing readings to those landmarks using the computations described by Betke and Gurvits [5].

Another simple method of calculating pose from the models arises when the segments

of the model intersect at ground points. In this case, the model, despite encompassing ground and vertical segments, is simplified to the case of a set of ground points. This set of points moves rigidly in the Bird's Eye View, and therefore self-localisation is in essence the computation of the 2D transformation tracking the movement of the points.

This method requires intersecting segments, which is similar to tracking corners but in a much more stable manner. This is specially true when dealing with long segments, as the noise in the orientation of small segments may become significant, affecting the computation of the intersections and the quality of corner estimates.

Alternatively, localisation is achieved through an optimisation procedure, namely minimizing the distance between model and observed line segments, directly at the pose parameters. This is computationally more expensive, but more robust to direction errors on the observed line segments.

In summary, we estimate self-localisation using three different methods, namely bearings based localisation, estimation of the rigid transformation of the ground points, or an optimisation procedure comparing the distance between the model and the observations. In the following sections we detail each of these methods.

Bearings based pose computation

The objective is to estimate the robot pose, i.e. its position and heading, w.r.t. the world system of coordinates, based on a map of known landmarks. The observations consist of bearings, that is angles to the landmarks w.r.t. the robot heading (see fig. 3.6).

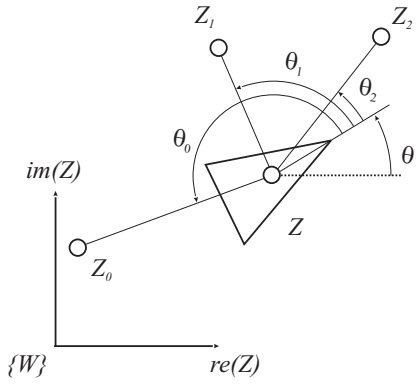


Figure 3.6: Triangulation: given the map $Z_0...Z_n$ calculate the robot pose, (Z, θ) from the observed bearings $\theta_0...\theta_n$.

The algorithm of Betke and Gurvits [5], addresses precisely this objective. We summarise here their algorithm for the sake of completeness.

Landmarks are represented efficiently using complex numbers, $Z_0...Z_n$. From that representation is possible to derive a system of linear equations on the n unknown distance ratios defined w.r.t. the robot, $r_k = |(Z_k)^r| \,/\, |(Z_0)^r|$:

$$b_k r_k - b_i r_i = c_k - c_i$$

where k, i range from 1 to n, $c_k = 1/(Z_k - Z_0)$, $b_k = c_k.exp(j\theta_k)$ (expressions c_i, b_i are equal to those on the index k) and θ_k is the bearing observation to landmark k. Notice that by combining each index k with an index i results in n^2 equations, of which only $n-1$ equations are independent. However those are complex equations expressed on real valued variables. Therefore, there are actually $2*(n-1)$ equations, the system is over-determined and so there is a *least squares* solution for the unknowns $[r_1...r_n]$. Manipulating the expressions, the *least squares* solution can be computed with linear complexity in the number of the landmarks.

Finally the robot position, Z and orientation, θ become:

$$Z = Z_0 - \frac{1}{n}\sum_k \frac{Z_k - Z_0}{r_k.e^{j(\theta_k - \theta_0)}} \quad , \quad \theta = \arctan(Z_0 - Z) - \theta_0$$

This procedure does not work, when all the landmarks and the robot are aligned or lay on a circle. This is confirmed by some simple geometric reasoning, but it rarely occurs in practice.

In our application, the landmarks are vertical lines pertaining to the model of the scene. The model contains the locations of those landmarks, i.e. their projections onto the ground plane. Those locations are defined essentially by the intersection of ground and vertical segments at the time of building the model.

Corners based pose computation

The features selected for tracking are image corners defined by the intersection of edge segments [44], which can usually be found in indoor environments. The detection process benefits from a larger spatial support, as opposed to local corner detection filters, thus leading to increased accuracy and stability. Figure 3.7(a) shows a corner point E defined as the intersection of lines $\overline{AB}$ and $\overline{CD}$. In this way, corners do not necessarily have to correspond to image points of extreme changes in brightness. This approach can deal with information loss due to occlusion or filtering (e.g. the "roundness" of corners due to image smoothing).

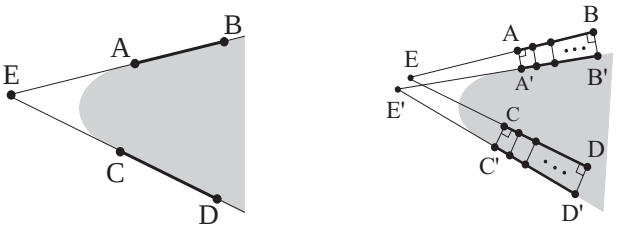


Figure 3.7: (Left) Corner point E is defined as the intersection of lines $\overline{AB}$ and $\overline{CD}$; (Right) The segments $\overline{AB}$ and $\overline{CD}$ are adjusted to $\overline{A'B'}$ and $\overline{C'D'}$, and thus the corner point is tracked from E to E'.

We track corner points by tracking the corresponding support edge lines. Each edge segment is tracked by searching along its perpendicular direction as described in the pre-

vious section. Figure 3.7(b) demonstrates how the segments $\overline{AB}$ and $\overline{CD}$ at time t, are tracked to the segments $\overline{A'B'}$ and $\overline{C'D'}$, respectively, in the image at time $t+1$.

After determining the new corner positions, we estimate a 2D rigid transformation between two successive bird's eye views, yielding the robot position and orientation relative to some pre-defined co-ordinate system.

Pose computation as a distance minimization problem

The corners based method is interesting as it results in a very simple computation algorithm. It is noted however that the basic features are the segments, and therefore the pose computation problem can be casted directly on those features.

Intuitively, the best pose estimate should align the scene model and the observed lines as well as possible. Therefore we can formulate an optimisation problem to determine the pose that minimizes the distances between model and observed lines.

Figure 3.8 shows and example of a model and the evolution of its segments in the next image. For each segment two distance errors are measured. The best pose should minimize the sum of the distance errors.

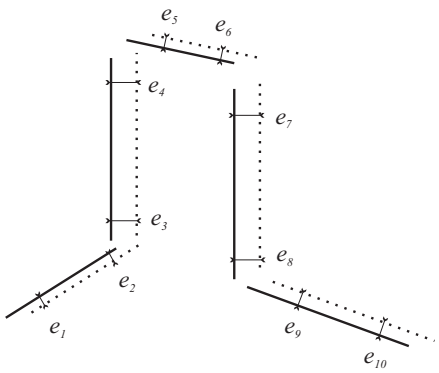


Figure 3.8: A model (dotted lines) and the evolution of its segments in the next image (thick lines). For each segment there are defined two distance errors ($e_1 ... e_{10}$).

Defining pose as $\mathbf{x} = [x\ y\ \theta]$ and the distance between the segments ab and cd as:

$$d\left(cd, ab\right) = f\left(c-a, b-a\right) + f\left(d-a, b-a\right)$$

where a, b, c, d are the segment extremal points and f is the normalised internal product:

$$f(\mathbf{v}, \mathbf{v_0}) = \frac{\left|\mathbf{v}^T.\mathbf{v_0^\perp}\right|}{\left\|\mathbf{v_0^\perp}\right\|}$$

then the problem of pose estimation based on the distance between model and observed segments can be expressed by the minimization of a cost functional:

$$\mathbf{x}^* = \arg_{\mathbf{x}} \min \sum_i d\left(s_i, s_{0i}(\mathbf{x})\right)$$

where s_i stands for observed vertical and ground line segments, and s_{0i} indicates the model segments (known a priori).

There are two main issues in this approach: obtaining an initial guess of the pose; and optimising the geometric distance between model and observed segments. The optimisation is performed with a generic gradient descent algorithm provided that the initialisation is close enough. For the initial guess of the pose there are also simple solutions such as using the pose at the previous time instant or, when available, an estimate provided by ground points 2D rigid transformation as in the previous method.

In summary, we have presented three different methods for computing the robot's pose. Each of the methods has its pros and cons considering the nature of the data (scene model), the noise and the computational complexity. Often the best method can be found only at run-time. Hence we run the three methods concurrently and at each time instant choose the best pose estimate according to a matching merit functional, which we detail in the following section.

3.2.3 Choosing the best pose-computation

Despite the optimizations that were performed for pose-computation, there are residual errors that result from the low-level image processing, segment tracking, and from the methods itself.

Our pose-computation algorithms are based on local gradients. In particular we consider image points corresponding to local maximums of the absolute value of the gradient. These points correspond to 3D scene points that are known by the robot. Due to noisy observations, in general the robot will not find its exact localisation, and therefore the matching of its knowledge of the scene with the observed image will not be perfect too.

Hence, it is interesting to define a *matching merit function*. This function allows us to assert about the quality of matching and through it we can describe precisely the self-localisation problem.

The merit function evaluates local image gradients at model edges, ∇I. In order to find the model edges in the image plane, the merit function, μ needs to take into account the model, $\{P_i\}$, its current pose, $\mathbf{x}$, and the projection function $\mathcal{P}$:

$$\mu(\mathbf{x}) = \sum_i |\nabla I\left(\mathcal{P}(P_i; \mathbf{x})\right)|. \tag{3.1}$$

Whenever the pose is precisely known, the merit function is expected to be maximum. Small deviations about the correct location imply merit losses.

In order to decide which is the best pose estimate derived from the three methods, we use therefore the matching merit function. The method yielding the best matching merit, i.e. the one returning the model pose that collects most of the local gradients, is the one that is chosen as being returning the most accurate pose calculation.

Choosing the best pose-computation is a simple evaluation of the matching merit. As

there is no optimisation of the pose estimate it is not guaranteed to be optimal. For this reason we have an additional step of fine pose adjustment that we shall detail in the following section.

3.2.4 Fine pose adjustment and detecting tracking losses

The coarse self-localisation process relies exclusively on the observed segments, and looks for the best robot pose justifying those observations on the image plane. The image brightness was considered only to track the segments individually due to the computational cost required to use it directly for pose estimation.

Having a good initial estimation of the self-localisation, allows us to use directly the image brightness to tune the estimated robot pose. Some errors at the segment's tracking stage may be recovered through the global interpretation of the current image with the a priori geometric model. Since the model is composed of segments associated to image edges, we want to maximize the summation of gradients at every point of the model wire-frame.

Denoting pose by $\mathbf{x}$ then we want to maximize the gradients at the image points corresponding to the projections of the model, i.e. we want to maximize the matching merit function defined in section 3.2.3 Eq.(3.1):

$$\mathbf{x}^* = \arg_{\mathbf{x}} \max \mu(\mathbf{x}). \tag{3.2}$$

Usually, there are model points that are non-visible during some time intervals while the robot moves. This is due e.g. to camera (platform) self-occlusion or to the finite dimensions of the image. In these cases the introduced matching merit function does not evolve smoothly with pose changes and is maximised by considering the maximum number of points possible, instead of the true segments pose. It is therefore interesting to include a smoothness prior at the function.

The solution we found is based on preserving the gradient values at control points of the segments of the model. This makes possible to indicate realistic values of gradient for the non-visible points and the optimisation of the merit function in this way resists to some occlusion. The gradient values are updated along time with a first order auto-regressive filter.

Let V denote the set o fall visible points. We re-define the matching merit function as:

$$\mu(\mathbf{x}, t) = \sum_{i \in V} |\nabla I\left(\mathcal{P}(P_i; \mathbf{x})\right)| + \sum_{i \notin V} m_i(t-1) \tag{3.3}$$

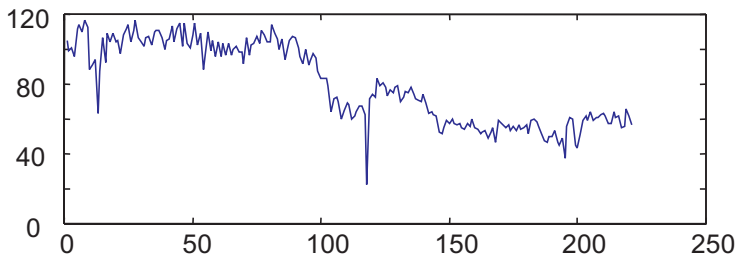
where m_i is the state of the gradient at the model point index i and is updated whenever the point is visible:

$$m_i(t) = \begin{cases} \lambda . |\nabla I\left(\mathcal{P}(P_i; \mathbf{x})\right)| + (\lambda - 1) . m_i\left(t-1\right), \ i \in V \\ m_i(t-1), \ i \notin V \end{cases}$$

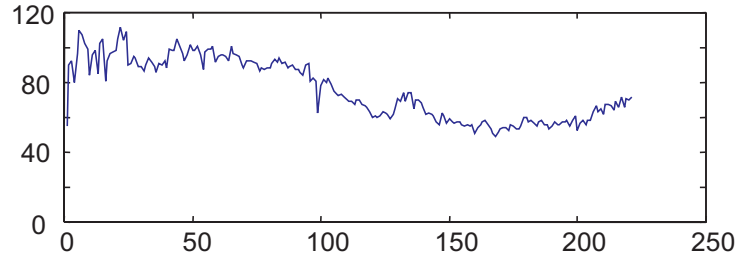
The smoothing factor, λ, takes the value 0.3 corresponding to limiting to 30% the participation of each observation to the gradient state. Finally, the pose is computed as before, Eq.(3.2), but now maximizing the new merit matching function $\mu(\mathbf{x}, t)$ at the current time instant t. We perform this optimisation by exhaustive search for some narrow bands around the current pose estimate.

In order to detect the loss of tracking during operation, the tracking process is continuously self-evaluated by the robot. This evaluation is based on the matching merit function, i.e. gradient intensities obtained within specified areas around the landmark edges. If the merit decreases significantly compared to the expected values, a recovery mechanism is immediately launched.

Figure 3.9 shows that the fine pose adjustment improves the merit score evaluating the quality of the tracking. Comparing both plots, we can see that is successfully avoided one sudden negative peak of the merit value, bellow less than 50% of the local values, corresponding to a tracking loss and its recovery.



Merit vs t



Merit vs t (using fine pose adjustment)

Figure 3.9: Merit score vs time, before (top) and after (bottom) fine pose adjustment. Each Y value is an average of absolute gradient at edge points of the model. After adjustment the scores are higher and evolve smoother along time.

In summary we may say that the matching merit solves the two issues raised in this section: the definition of the optimal pose estimate and the performance monitoring / failure detection.

3.2.5 Summary of the Self-localisation Module

Summarising the self-localisation module, we present the complete algorithm as a sequence of operations that the robot is continuously performing.

1. Feature tracking: acquire a new image and find the new segment locations given the previous ones.
2. Coarse pose computation: compute the pose of the robot knowing the location of the segments in the current image and the geometric model of the scene; choose from the presented methods (bearings based, corners based or segments distance minimization) the one yielding the best matching merit.
3. Fine pose adjustment: ameliorate the pose estimate by directly optimizing the matching merit function.
4. Model re-projection: compute current segment locations given the current pose.
5. goto 1.

Given the robot pose and a reference trajectory, we designed a control scheme that drives the distance and orientation errors to zero, while maintaining a forward velocity. This is detailed in the next section.

3.3 Control of the Mobile Robot

Providing good localisation estimation is an important part of the path following problem. The remaining part consists of using this information for controlling the robot.

The robot state consists of a pose vector, $\mathbf{x} = (x, y, \theta)$, describing its position (in *pixels*) and orientation. The navigation system can modify the robot's linear and angular velocities denoted by (v, ω). The robot dynamic model is that of a wheeled unicycle mobile robot, with two degrees of freedom (linear and angular velocities):

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \omega \end{cases} \tag{3.4}$$

The path to follow, Ψ, is defined as a set of points $\mathbf{x}_\Psi = (x_\Psi, y_\Psi, \theta_\Psi)$, expressed in the same coordinate system and units as the robot state vector, $\mathbf{x}$.

At each time instant, the motion planning module must determine a reference point on the trajectory, $(x_\Psi^{ref}, y_\Psi^{ref})$ which is then used to determine the position and orientation errors so as to correct the robot's motion:

$$(x_\Psi^{ref}, y_\Psi^{ref}) = \underset{(x_\Psi^{ref}, y_\Psi^{ref})}{\arg\min} \left\{ \| \, (x_\Psi^{ref}, y_\Psi^{ref}) - (x, y) \, \|^2 \right\}$$

To avoid multiple solutions, we use a regularization term that selects the path point, $\mathbf{x}_\Psi^{ref}(k)$ closest to that at the previous time instant, $\mathbf{x}_\Psi^{ref}(k-1)$. A signed distance-to-path error, d and an orientation error, $\tilde{\theta}$ are defined as:

$$d = [x - x_\Psi^{ref} \quad y - y_\Psi^{ref}][n_x \quad n_y]^T, \qquad \tilde{\theta} = \theta - \theta_\Psi^{ref}$$

where $[n_x \quad n_y]$ is the normal to the path at the chosen reference point. The geometry of this kinematic motion planner is shown in Figure 3.10.

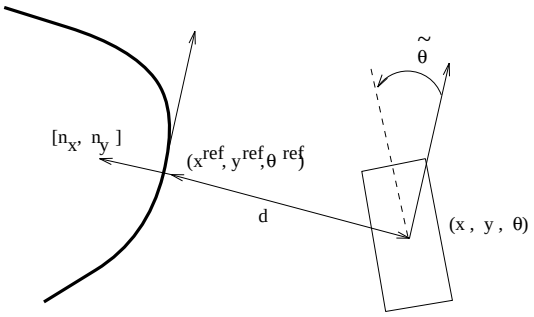


Figure 3.10: Kinematic motion planner used to reference points and to define the control error for the visual path following system.

The dynamic controller used to generate the robot's angular velocity was proposed de Wit et al in [18] for path following, and shown to be stable:

$$\omega = -k_3|v|\tilde{\theta} - k_2 v \; d \; \frac{\sin\tilde{\theta}}{\tilde{\theta}} + \frac{v\cos\tilde{\theta}c(s)}{1 - c(s)\; d} \tag{3.5}$$

where k_2, k_3 are constants to be tuned, s designates the path length, and $c(s)$ is the local path curvature.

In order to tune the controller, it is important to note that the choice of Frenet coordinates and the control law, results in a linear second order system representing the robot-path error distance, d. Second order systems are intuitively characterised by the natural frequency, α and damping coefficient, ξ which are directly related with the controller parameters as:

$$k_2 = \alpha^2 \quad , \quad k_3 = 2\xi\alpha.$$

Usually ξ is set to $1/\sqrt{2}$ meaning a small overshoot, and α is left free to specify faster or slower systems, i.e. shorter or longer settling times.

To describe more intuitively the control law of Eq.(3.5) it is interesting to analyse two particular cases of importance, namely the cases where the reference path is a straight line or a circle:

- When the reference path is a straight line, $c(s) = 0$, control reduces to the simple case of encompassing only two terms. One is proportional to the heading error and the other to the distance to the trajectory.
- When the path is a circle, the curvature is a constant. If the robot is close to the

steady state, i.e. the heading and distance errors are close to zero, then the third term of the control law is constant which is according to the intuition that to describe a circle the robot must have a constant angular velocity (the linear velocity is set to constant from the beginning).

Mostly, the forward velocity, v, is equal to the maximum, $V_{\max}$ but for safety reasons, we impose a maximum value on the angular velocity, $|\omega| < W_{\max}$. When this value is achieved, we saturate ω and reduce v to $V_{\max}.W_{\max}/|\omega|$, in order to avoid large lags in narrow turns.

The curvature, $c(s)$ is defined as the derivative of the heading relative to the path length s, i.e. $c(s) = d\theta/ds$. It is therefore important to have smooth trajectories as otherwise there will appear saturations on the control signal and very little ability of the controller to react to heading and distance errors.

There are very interesting works on selecting the best trajectories to follow, guaranteeing continuous curvature properties based essentially on polynomials [80] or more recently on clothoids / spirals [29, 56]. In our case, as many of the paths are taught by example, it is not possible to plan the trajectories. We take therefore the simple approach of filtering the path. The filter we use is a Hamming window, a technique borrowed from speech processing domain.

Our trajectories are defined as arrays of discrete points and therefore the curvature is estimated using finite differences. The finite differences are centered for unbiased curvature estimates. In order to have accurate curvature values we interpolate our trajectories to have a reference path about an order of magnitude more dense than the typical jumps on pose estimates.

For the current control law, noise in self-localisation measurements (x, y, θ) directly implies noise in control outputs (v, ω). To prevent this direct noise transmission we include temporal integration of the measurements with an Extended Kalman Filter (EKF).

The inputs and outputs for the EKF are then poses (x, y, θ), and the dynamics are that of a wheeled mobile robot - unicycle type, Eq.(3.4) with state vector augmented with velocities $\mathbf{x} = (x, y, \theta, v, \omega)$. Velocities are assumed constant and driven by white noise. Forward velocity noise covariance is assumed low due to the control characteristics.

3.4 Experiments and results

In this section we present experiments and results following the organisation of the chapter. First we show results of simulated experiments carried on the modules of self-localisation and control separately. Then, we describe experiments and show results of Visual Path Following, combining the modules tested but now conducted on a mobile robot in a real world scenario.

3.4.1 Relative usage of coarse pose computation methods

The coarse self-localisation is performed in parallel by the three methods presented previously. The computed pose is selected from the three available ones according to the photometric criterium detailed in section 3.2.3 Eq.(3.1).

The experiment documented here is composed by three path segments differing essentially by their shapes. The first path segment is a straight line, while the other two are arcs of circumference of about 90^o.

Figure 3.11 shows the percentage of utilisation of each method at each time interval. The three path-segments are concatenated in the figure.

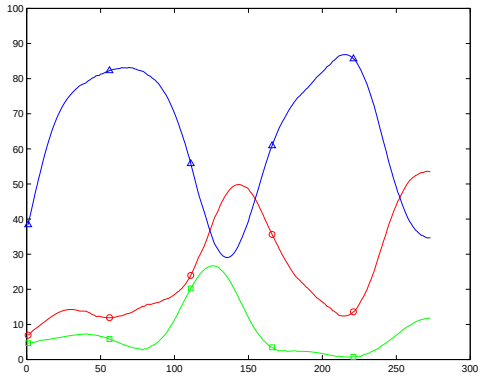


Figure 3.11: Self-localisation method usage (percentage) vs time (image number). The lines marked with ∘ (red), □ (green) and △ (blue) represent respectively: bearings based localisation (Betke & Gurvits), corners based localisation and localisation by optimising line-segment distances.

This experiment shows that the method based on the minimization of the distance between the model and the observed segments are chosen most of the time and clearly dominates in the first path segment (see the next table).

Method	Path segment 1	Path segment 2	Path segment 3
Betke & Gurvits	12.8%	31.1%	44.4%
Corners transf.	5.5%	10.9%	8.9%
Optimis. segms. dist.	81.7%	58.0%	46.7%

At curvilinear path-segments the method proposed by Betke and Gurvits, compares favorably relative to the minimization of the distance between segments due to the minimization process itself. We impose a maximum number of iterations to guarantee that the total computation time is limited and approximately constant for each processed image. In the case of the curvilinear path-segments, the movement of the image segments is larger and therefore the optimization would require more iterations. Since they are not performed the method is superseded in quality by others.

An additional justification is found on the used model. This model is based mainly in

vertical segments whose essential information is in fact angular (bearings) being therefore captured almost integrally by the Betke and Gurvits' method.

3.4.2 Robot Control

The next experiments are simulations of a wheeled mobile robot performing visual path following. The self-positioning is simulated with mobile robot model integration. The vision procedure is noiseless for the purpose of testing only the control. In all the experiments the sampling period is 0.7 sec.

Tuning the gain

In a first experiment, we set the robot to follow a trajectory with two controllers characterised by different gains (see Figure 3.12). The range of gains to achieve successful path following is large, however to obtain specific error bounds there are minimum gain values. Note that the second controller, middle plot of the figure and line b of the rightmost plot, exhibits a smaller error-distance to the trajectory. This is expected as the gain is larger.

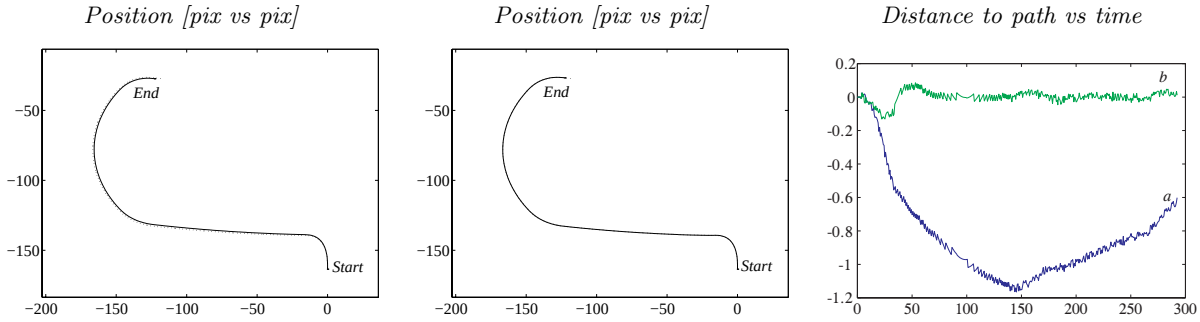


Figure 3.12: Control tests: different gains result in apparently equal path following results (Left and Middle), but actually the error is smaller in the case of the larger gain (Right curve b).

In a new experiment, the initial location of the platform is not exactly coincident with the reference trajectory. Due to filtering, the curvature is under-estimated at the start of the reference trajectory. Therefore the controller has to compensate both errors, i.e. the errors in position and initial angular velocity (almost zero due to the filtering). In these examples, unlike the previous ones, the platform should start turning to its left side from the beginning of the experiment.

Figure 3.13, left column, shows an example where the controller does not have enough gain to compensate both errors, and thus shows that a wrong initial curvature may be enough for inaccurate trajectory following. In the right column of the figure, the curvature error creates a lag in trajectory following, but the controller reacts and compensates the error due to its larger gain.

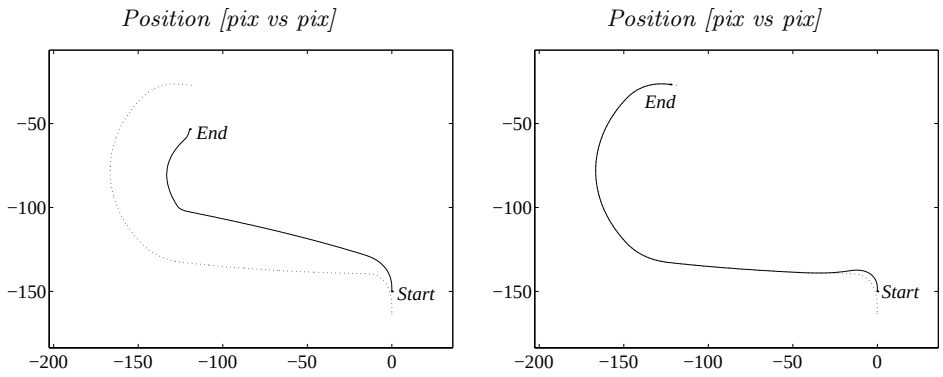


Figure 3.13: Control tests: initial curvature. The dotted and solid lines show the reference and the followed trajectories respectively.

Distant initial condition

Figure 3.14 shows that the controller is capable of dealing with a very distant initial position relative to the start point of the trajectory. The robot starts at [0 0 0] with an angular velocity as specified by the controller given the closest (desired) trajectory point. Since the robot is too far from the reference trajectory the term proportional to the distance is clearly dominant, thus justifying the constant sign of the angular velocity till closer to the reference path.

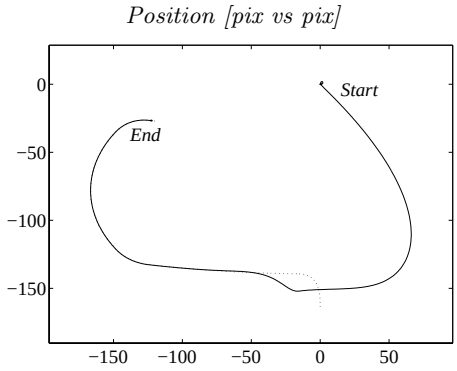


Figure 3.14: Control tests: far initial location. The dotted and solid lines show the reference and the followed trajectories respectively.

Forward velocity reduction at sharp turns

Figure 3.15 shows saturation of the angular speed at the various turns. Due to the look-ahead characteristics of the trajectory planning module, forward velocity reduction is not so significant at the turns after the first one. The behavior in the first turn is different because there are no prior path reference points that can be used in the kinematic planning.

Figure 3.16 shows the error signals and control actions for the same input and using the kinematic planner and dynamic controller described before. This example illustrates forward velocity reduction to complement angular velocity saturation.

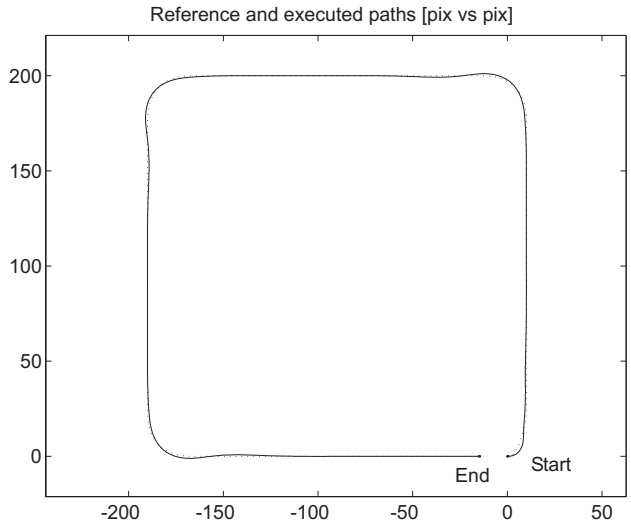


Figure 3.15: Visual path tracking shown in a simulated trajectory, with the robot moving anti-clockwise. Dotted and solid lines correspond to reference and robot trajectories respectively.

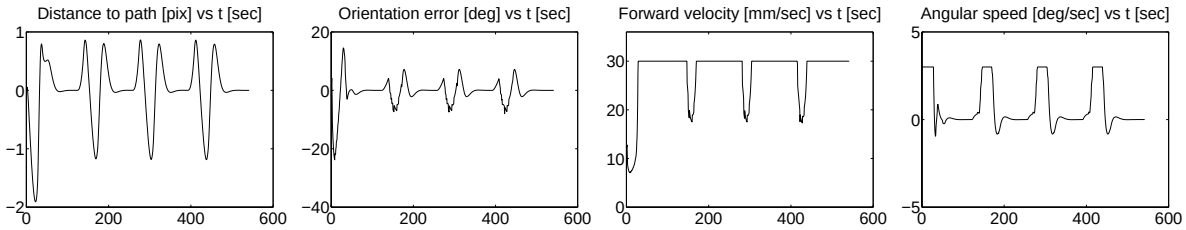


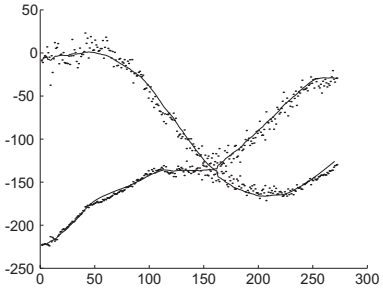
Figure 3.16: Visual path tracking in a simulated trajectory. Left to right: distance to path and orientation error (controller inputs), forward and angular velocities (controller outputs).

3.4.3 Vision and Control

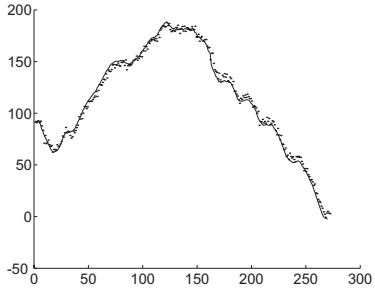
Experiments were conducted using catadioptric panoramic vision system built in our institute, mounted on a TRC labmate mobile robot. Processing was carried on with an on-board PC PII-350MHz equipped with a TekRam image acquisition board.

Docking experiment

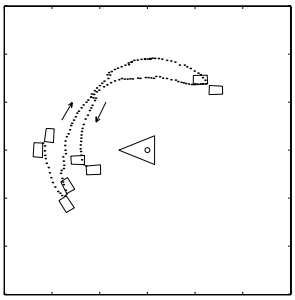
For *Visual Path Following*, we specified a reference trajectory in image coordinates, relative to a single landmark composed of two rectangles. The mobile robot uses the input of the omni-directional camera to move under closed loop control, as described in this chapter.



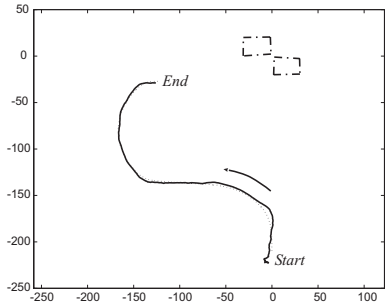
(a) XY observations [pix] vs time [image number]



(b) Heading observations [deg] vs time [image number]



(c) Landmark position [pix vs pix]



(d) Robot position [pix vs pix]

(e)

Figure 3.17: Visual Path Following, with the trajectory specified in image coordinates. (a) x, y positions before (dotted line) and after filtering (solid line). (b) Orientation before (dotted line) and after filtering (solid line). (c) Tracking of the landmark in the robot frame (d) Dash-dotted line shows the landmark that defines the origin. The dotted line is the specified trajectory and the solid line shows the filtered position estimates. (e) Image of mobile robot at the end of path following.

Figures 3.17(a,b) show estimates of self-localization. Noise is primarily due to the small size of the chosen landmark and poor image resolution. The Kalman filter can effectively reduce noise mainly along smooth paths. Figure 3.17(c) shows tracking results to illustrate the convenient use of omni-directional vision for landmark tracking, in spite of its large azimuthal movement relative to the robot. Figure 3.17(d) shows that the errors between the reference trajectory (dotted) and that resulting from visual self-localization

(solid line) are very small. Figure 3.17(e) shows the mobile robot at the final position after completion of the desired navigation task.

The processing time was approximately 0.8sec/image, where 50% was used on image processing and the remaining 50% for displaying debugging information, image acquisition and serial communication with the mobile robot.

Door traversal

Figure 3.18 illustrates tracking and self-localization while traversing a door from the corridor into a room. The tracked features (shown as black circles) are defined by vertical and ground-plane segments, tracked in bird's eye view images.

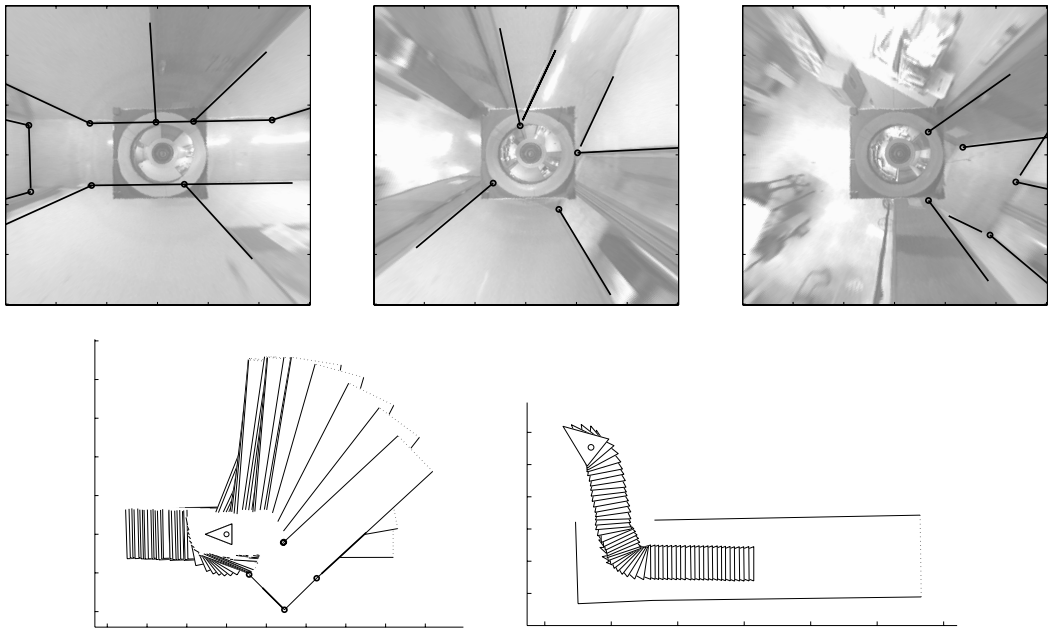


Figure 3.18: Top: Feature tracking at three instants (black circles); Bottom: estimated scene model and self-localization results.

Currently, the user initializes the relevant features to track. To detect the loss of tracking during operation, the process is continuously self-evaluated by the robot, based on gradient intensities obtained within specified areas around the landmark edges. If these gradients decrease significantly compared to those expected, a recovery mechanism is launched.

3.5 Concluding Notes

We described the use of an omnidirectional vision system for navigation tasks, namely visually path following. One of the main advantages lies in acquiring panoramic images of the environment without any moving parts on the physical setup.

In chapter 2 we have described the image formation model and the method adopted for estimating the model parameters. Using this projection model, we presented a method

for obtaining ground dewarped images or a bird's eye view of the ground floor. In this chapter we showed that this representation significantly simplifies navigation problems, since the image coordinates differ from the ground plane coordinates by a simple scale factor, thus eliminating any perspective effects.

Using this ground dewarped images we have presented a method to track image corner points defined by their support edge segments. Tracking this features is done in a robust manner and allows the estimation of the robot position and orientation relative to a known navigation landmark.

We further showed how this framework can be used to perform what we call *Visual path following*. A trajectory to follow is simply specified in the image plane and a suitable controller is used to drive the robot along that desired path. We described both the kinematic motion planner and the dynamic controller. Experiments were shown both in a simulated environment and with a real robot.

Chapter 4

Vision-based Navigation with an Omnidirectional Camera

This chapter proposes a method for the visual-based navigation of a mobile robot in indoor environments, using a single omnidirectional camera. It is of particular importance to fulfill global tasks, namely travelling long distances and therefore representing large environmental areas.
We approach the global tasks with Topological Navigation. It does not require knowledge of the exact position of the robot but rather, a qualitative *position on the topological map. The navigation process combines appearance based methods and visual servoing upon some environmental features.*
By combining Topological Navigation with Visual Path Following (detailed in chapter 3), a simple and yet powerful navigation system is obtained.

4.1 Introduction

Both robustness and an efficient usage of computational and sensory resources can be achieved by using visual information in closed loop to accomplish specific navigation tasks or behaviors [90, 89]. However, this approach cannot deal with global tasks or coordinate systems (e.g. going to a distant goal), because it lacks adequate representations of the environment. Hence, a challenging problem is that of extending these local behaviors, without having to build complex 3D representations of the environment.

We use the *Topological Navigation* approach for travelling long distances in the environment, without demanding accurate control of the robot position along a path. The environment is represented by a *Topological Map* [63, 59, 112], described by a graph. *Nodes* correspond to recognizable *landmarks*, where specific actions may be elicited, such as entering a door or turning left. *Links* are associated with regions where some environmental structure can be used to control the robot.

In our approach, landmarks are directly represented by *omni-directional images*. Links

are represented by sequences of images that correspond to trajectories which the robot can follow by servoing upon some environmental features.

We use omni-directional images as an implicit topological representation of the environment, and rely on appearance based methods [49, 55, 1, 107] to provide a *qualitative* measurement of the robot's *global* position. Progression is assessed by comparing the current view with images acquired *a priori* [72]. Images of the topological map are encoded as a manifold in a low dimensional eigenspace obtained from Principal Components Analysis.

Ishiguro and Tsuji, in [54], propose to find the robot's localisation by matching images in the frequency domain, whereas Horswill [50] used the actual views as landmarks. The work most closely related to ours is that described in [91], which combined appearance based methods and visual servoing, although the image geometry, matching scheme and method of servoing were different from those detailed in this chapter.

Visual servoing is applied to control *locally* the pose of the robot relative to image features, and to navigate between nodes. In this work, we control the robot heading and position by servoing upon the corridor guidelines, extracted from bird's eye views of the ground plane. Thus, the combination of appearance based methods and visual servoing, at the global level, means that we can maintain a causality constraint while traversing longer distances and sampling the environment less frequently than with previous approaches [1, 72, 54].

When the robot moves in environments containing wide windows, large non-uniform illumination changes may occur close to the windows at different times of the day. These changes are often sufficient for failing the comparison of images taken at the same place when using the L_2 norm, and due to the non-linearity cannot be made robust even by using zero mean normalised cross correlations.

Image edges are known to be robust against illumination changes and thus constitute more effective data for self-localisation. However the direct comparison of edges is still not the solution as they are sensitive to noise and small changes of the camera pose.

To overcome this problem, in object recognition problems there are compared distances between shapes, i.e. sets of edge points, instead of directly comparing the edge points. In [36] Gavrila et al propose chamfer distances to detect pedestrian and traffic signs. Detecting pedestrians is interesting as they are characterised by variable shapes therefore requiring graceful degrading detection methods. The chamfer distance is in essence an approximation to the average of distances between corresponding points of two shapes, which is computed efficiently by mask based image operations.

Huttenlocher et al [52] take an alternative approach, of using Hausdorff distances to compare the shapes. Qualitatively an Hausdorff distance measures the maximal distance between corresponding points using a robust technique.

Unlike object recognition applications where one needs to detect shapes in unknown locations of an image, in robot self-localisation, the important variable is an image index identifying images in a database. Despite this difference, the problems are similar and

thus the techniques of object recognition can be applied to the self-localisation.

During normal operation, the robot does not change instantaneously to an arbitrarily different position. There is therefore in self-localisation a causality property, that is useful for designing more efficient search algorithms.

Concluding, global tasks in our system, such as going to a distant location, are performed using topological navigation. The representation used is a topological map of the environment based on the appearances of the various locations. The map is described by a graph structure: nodes correspond to recognizable landmarks where different actions can be elicited, and links are associated with regions where some environmental structure can be used by the robot to control its pose (visual servoing).

Recognizable landmarks are reference images (i.e. specific appearances) associated with a qualitative position. In regions where large non-uniform illumination changes may occur, image edges are used instead of the intensity images, and the matching techniques are shape based. Appearances imply large storage capacities, and therefore are represented using approximations by manifolds on reduced order eigenspaces. As we do not require complex systems to capture precise (metric) information, problems of drift and slippage are easily overcome. Furthermore, topological maps deal only with proximity and order and so global errors do not accumulate.

Advantageously, Visual Path Following (detailed in chapter 3) complements Topological Navigation, providing the robot with the ability to undertake tasks requiring different levels of knowledge of the world.

Relying upon the two environmental representations / navigation methodologies, our robot is able to perform enlarged navigation tasks encompassing e.g. navigation in corridors, doors crossing and docking.

Chapter Organisation

We start by describing the Topological Navigation modality. Of particular importance are the appearance-based environmental representations for mobile robot self-localisation. Firstly, we describe representations based on image eigenspaces. Then we present two methods based on image-edges to represent the environment at regions of large non-uniform illumination changes.

Topological navigation is combined with Visual Path Following (detailed in chapter 3), for building a simultaneously global/qualitative and local/precise navigation system.

Finally, we present topological localisation and combined navigation experiments.

4.2 Navigating using Topological Maps

We use a *topological map* to describe the robot's *global* environment. This map is used to reference the qualitative position of the robot when traveling long distances. A mission could be specified as: "*go to the third office on the left-hand side of the second corridor*".

The robot must be able to travel along a corridor, recognize the ends of a corridor, make turns, identify and count door frames. These behaviors are implemented through an appearance based system and a visual servoing strategy.

The appearance of an object, defined by Murase and Nayar in [74], is a set of images resulting of the combined effects of the object's shape, reflectance properties, pose in the scene and illumination conditions. Hence, the matching of a run-time image with one of the images of the appearance set therefore indicates a particular combination of the properties.

As the shape and reflectance are constant properties of the object, one image matching identifies one pose, assuming as a first step that the matching procedure is illumination independent and that differing poses imply distinct images. For example preserving an ordering of the poses on the appearance set, the retrieval of a particular image indicates therefore a location within a path. If in addition some of the appearance poses are marked as landmarks, then there is also a recognition of distinct places. For example in a corridor scene, the appearance based system provides qualitative estimates of the robot position along a corridor, and recognizes distinctive places such as corners or door entrances.

A map is thus a collection of inter-connected images, as in the example of Figure 4.1, representing the floor map of our institute. To go from one particular locale to another, we do not have to think in precise metric terms. For example, to move the robot from one corner to the opposite one we may indicate the robot to follow one corridor till the first corner and then to follow the next corridor again till the corner, therefore reaching the destination. The navigation problem is decomposed into a succession of sub-goals,

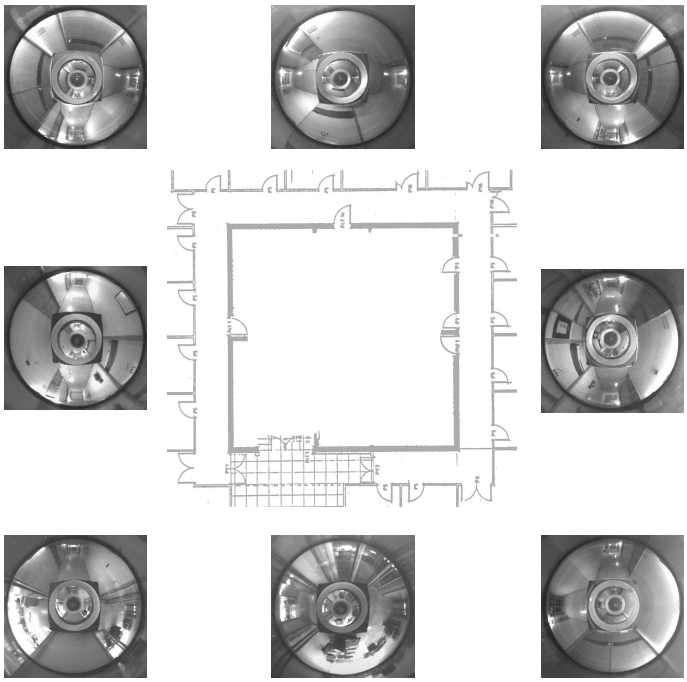


Figure 4.1: A topological map of the floor map of our institute.

identified by recognizable landmarks. The required navigation skills are the ability to follow roads, make turns and recognize that we have reached a landmark.

To control the robot's trajectory along a corridor, we detect the corridor guidelines and generate adequate control signals to keep the robot on the desired trajectory. This processing is performed on bird's eye views of the ground plane, computed in real-time.

Topological maps scale easily by connecting graphs at multiple resolutions to map different regions of the environment. All the robot needs is a specialized behavior to navigate along the *links* by using a vision-based control process and a procedure to recognize the locations/*nodes* where further actions may be undertaken.

In the following we present three different methods to build the topological map and describe the respective localisation techniques. The first method uses directly the grey-level images. As finding the closest image from a large database is computationally expensive, a compression technique based on principal component analysis is applied to the database.

The next two methods work upon edge-images, in order to be more robust to non-uniform illumination changes that occur for example close to windows. One of the methods compares views measuring the average distance of the corresponding features. The algorithm for computing the average distance is designed to be time efficient by using mask-based image processing. The third (last) method is based on a computationally costly distance measure but includes intrinsically the advantage of increasing the robustness to partial occlusions. Again it is used an eigenspace approach to achieve compact storage and fast indexing.

4.2.1 Image Eigenspaces as Topological Maps

The topological map consists of a large set of reference images, acquired at pre-determined positions (landmarks), connected by links on a graph. Since the robot perceives the world through omnidirectional images, these images are a natural way of represent landmarks.

During operation, the reference image that best matches the current view indicates the robot's *qualitative* position in the topological map. Hence, the reference images can be seen as a large-dimensional space where each point indicates a possible reference position of the robot.

In general, the number of images required to represent the environment is very large, and one needs to find a method to compress this information. We build a reduced-order manifold to approximate the reference images, using Principal Component Analysis (PCA), as described by Murase and Nayar in [74], and detailed by Winters in [113] or Gaspar, Winters and Santos-Victor in [35].

Each reference image is associated with a *qualitative* robot position (e.g. half way along the corridor). To find the robot position in the topological map, we have to determine the reference image that best matches the current view. The distance between the current view and the reference images can be computed directly using their projections (vectors) on the lower dimensional eigenspace. The distance is computed between M-dimensional

coefficient vectors (typically 10 to 12), as opposed to image size vectors (128 × 128). The position of the robot is that associated with the reference image having the lowest distance.

When using intensity images to build a topological representation of the environment the robot is prone to miscalculating its location where *large non-uniform* deviations in illumination occur (see Fig.4.2). This is due to the comparison of images being still a sum of squared differences of brightness (radiance) values, therefore directly influenced by the illumination changes. However, it can be overcome by using edge images to represent the environment.

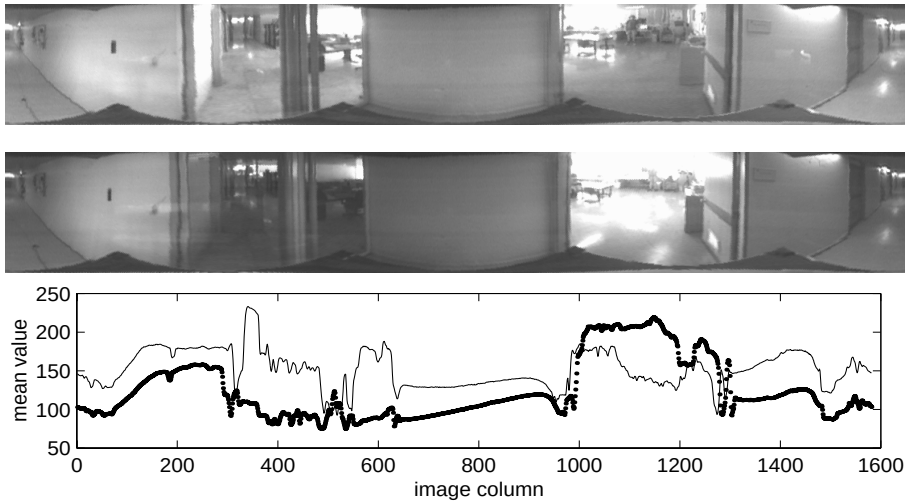


Figure 4.2: Top: images acquired at 5pm and 11am. Bottom: image intensity shows large non-uniform deviation in brightness (the thin line relates to the first image).

The direct comparison (correlation) of database and run-time edge images is still not the desired solution, as it is not robust to the edge-point position errors created by the natural small differences of the robot locations. The corresponding edge-points of matching images are found usually at near but different positions. We can only say that the shapes observed in the images are similar. The solution is therefore to compare shapes instead of edge-points. In particular when evaluating the matching of two images we are interested in computing distances between shapes present in both images.

There are several possible definitions of the distance between shapes. Two very well known are the Hausdorff distance and the chamfer distance. The Hausdorff distance is the maximum of the distances between all the points of one shape to the corresponding points of the other shape [52]. The chamfer distance is, in essence, the average distance between the points of the two shapes [36, 7]. In the following sections we shall detail these two distances and present their application to the localisation problem.

4.2.2 Localisation Based on the Chamfer Distance

Topological localisation consists of identifying the current run-time image in a set of database (template) images. As referred, one way to compare images, in a manner robust

against non-uniform illumination changes, is to use edge-images and a distance measure that compares shapes defined by edges. The chamfer distance is one such measure that can be computed efficiently. In this section we review the chamfer distance and then its application to robot self-localisation.

The chamfer distance is based on the correlation of a template edge-image with a *distance transformed image*. The distance transform of an edge-image is an image of the same size of the original, that indicates at each point the distance to the closest edge point [7, 36, 16]. In other words, the distance transform is an augmented representation of the edge-image as it contains the edge points, represented by the zero values, and at non-edge points the distances to the closest edge points.

There are several metrics for computing a distance transform which have been extensively reviewed by Cuisenaire in [16]. Of particular interest are the distance transforms where the value at each pixel can be computed from its neighbours, since they result in fast algorithms. Unfortunately the Euclidean metric does conform to this rule, but there are good approximations such as the *chamfer distance transform*[1].

The chamfer distance transform is computed from an edge-image using the forward and backward masks shown in figure 4.3 [7, 36]. There are various possible values for the constants in the masks. We use the values according to Montanari's metric [16]. Alternatively, Borgefors in [7] proposes optimal values to minimize the difference to the Euclidean metric and proposes sub-optimal integer approximations to save computations, but this is beyond the scope of the present review.

$+\sqrt{2}$	+1	$+\sqrt{2}$
+1	**+0**	

	+0	+1
$+\sqrt{2}$	+1	$+\sqrt{2}$

Figure 4.3: Forward and backward masks for computing the distance transform. The element in bold face indicates the centre of the mask.

The constants shown in the masks are added to each of the local values and the resulting value of the mask computation is the minimum of the set. Both masks are applied along the rows of the initialised image. More precisely, the forward and backward masks are applied starting respectively at the top-left and bottom-right corners of the image. The computation power is therefore similar to a linear filtering by a FIR filter with a 3×3 support mask.

Figure 4.4 shows the distance transform of the edges of an omnidirectional image. We remove the inner and outer parts of the omnidirectional image as they contain artifact edges, i.e. edges not related to the scene itself, created by the mirror rim and the robot plus camera self-occlusion.

[1] Not to confuse with the chamfer distance between two shapes. The chamfer distance transform is an image processing operation useful for computing the chamfer distance of two shapes.

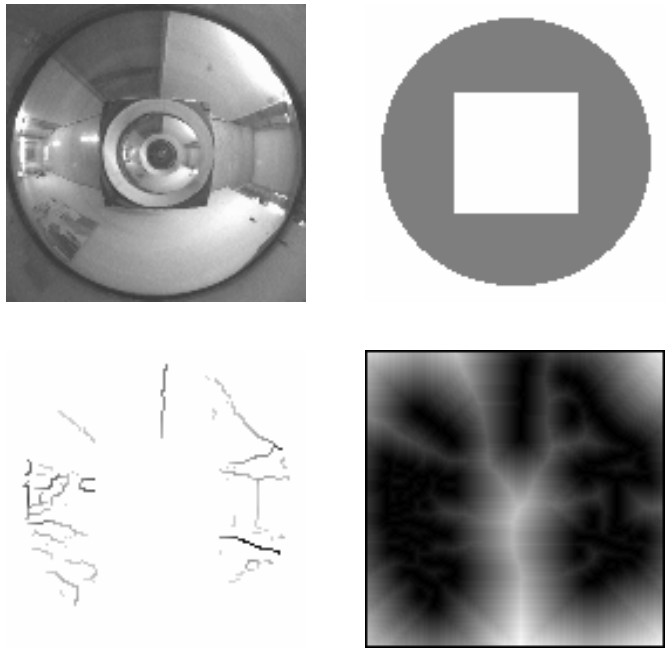


Figure 4.4: Distance transform: (top) original omnidirectional image and mask showing in grey the region of interest, (bottom) edges found in the region of interest and the distance transform of the edge-image.

Finally, given the distance transform, the chamfer distance of two shapes is then computed as the correlation:

$$d(D,T) = \frac{\sum_{i,j} D_{ij} T_{ij}}{\sum_{i,j} T_{ij}} \tag{4.1}$$

where T is a template edge-image and D is the distance transform of the edges of the current run-time image. As weaker edges (small gradient magnitudes) are more susceptible to noise, we set T_{ij} to the gradient magnitudes of the template images, instead of binary edges. Hence, we give more weight to the strongest edges.

Equation 4.1 says that the *chamfer distance* is an average of the distances between corresponding points of the shapes. In a strict sense, it is an approximation as the underlying *chamfer distance transform* is itself an approximation to the Euclidean distance. In practice this difference is not relevant as typically the shapes to compare are at similar poses and the distances between the points are small enough to make negligible the difference of the chamfer and the Euclidean distances.

In the topological localisation application, we want to find the database image corresponding to the current run-time image. In order to find the best matching we search the database using the chamfer distance as the comparison measure. The comparison of images is done from an edge-image to a distance transformed edge-image. The distance transformation may be applied either to the run time or to the database images [36]. We apply the distance transform to the run time edge-images and leave to the template edge-

images the role of selecting the relevant edge locations. The reason is that run time images may have occlusions caused by humans, which create non-scene edges possibly far away of the template edges. These large (erroneous) distances would be accounted if exchanging the roles of the template and run-time images, thus considering all run-time edges.

The distance as defined by Eq.(4.1) is zero for perfectly matching images. Therefore we search for the image matching the current image I_m in a set $T_1 \dots T_n$ by minimizing Eq.(4.1),

$$\hat{n} = \arg_n \min d\left(D(I_m), T_n\right). \tag{4.2}$$

Notice that, unlike recognition applications such as pedestrian and sign detection in an image [36], in the localisation application the template and run-time images have equal sizes. The search parameter is an image index instead of translation, rotation and scaling. The range of the index is the size of the database.

Usually there is a large number of database images, and thus finding the localisation as in Eq.(4.2) is computationally expensive. However it only needs to be performed at the first moment, when the robot is dropped-in-scene. During normal operation there is a causality constraint along the consecutive locations. We reduce the search range to a window around the last location, typically sized of ± 5 images.

Concluding, the *chamfer distance* of two shapes is an approximation to the average distance between the points of the shapes computed in an efficient manner using the *chamfer distance transform*. Efficiency in computation is important as the localisation procedure is continuously searching the database and therefore constantly preforming distance evaluations. In the next section we will use the Hausdorff distance that, although computationally more expensive than the chamfer distance, improves the robustness properties against occlusions.

4.2.3 Eigenspace approximation to the Hausdorff fraction

In this section we present the method of topological localisation based on the *Hausdorff fraction*. It is appearance based as the preceding methods and, as the chamfer distance based method, it uses edge-images in order to be more robust to non-uniform illumination changes. The *Hausdorff distance* [88] (of which the Hausdorff fraction is a subset) is a technique whereby one can measure the distance between two sets of points, in our case edge images. In the following, first we review the Hausdorff fraction, then we discuss its application to localisation and finally show an example.

The Hausdorff distance of two shapes is defined as the minimum of the distances from the points of one shape to the corresponding points of the other shape. As this value normally depends on the shape chosen as the template, the Hausdorff distance is the maximum of the two distances obtained taking each of the shapes as the template. The distance from a shape chosen as template to the other shape is termed therefore a *Directed Hausdorff distance*. Using the directed distance is a normal choice for critical time dependent systems, and hereafter when referring to distances we shall be considering

directed distances.

The Hausdorff distance is very sensitive to even a single outlying point of one of the shapes. The Generalised Hausdorff distance, defined by Huttenlocher et al in [52], is thus proposed as a similar measure but robust to partial occlusions. The generalised Hausdorff distance is an f^{th} quantile of the distances between all the points of one shape to the corresponding points of the other shape. For example the $\frac{1}{2}th$ quantile is the median and the $1st$ quantile is the maximum reverting therefore the generalised distance to the original definition. The quantile is chosen according to the expected noise and occlusion levels.

In recognition applications, the generalised Hausdorff distance is further specialised for saving computational power. The Hausdorff fraction, the measure we are interested, instead of measuring a distance between shapes evaluates the percentage of superposition considering one of the shapes dilated. Still for computational efficiency, the principal components analysis is included resulting in an eigenspace approximation to the Hausdorff fraction [53].

The eigenspace approximation is built as follows: Let I_m be an observed edge image and I_n^d be an edge image from the topological map, arranged as column vectors. The Hausdorff fraction, $\hbar(I_m, I_n^d)$, which measures the similarity between these images, can be written as:

$$\hbar(I_m, I_n^d) = \frac{I_m^T I_n^d}{\|I_m\|^2} \tag{4.3}$$

An image, I_k can be represented in a low dimensional eigenspace [74, 112] by a coefficient vector, $\mathcal{C}_k = [c_1^k, \cdots, c_M^k]^T$, as follows:

$$c_j^k = e_j^T.(I_k - \bar{I}).$$

Here, $\bar{I}$ represents the average of all the intensity images and can be also used with edge images. Thus, the eigenspace approximation to the Hausdorff fraction can be efficiently computed as:

$$\widehat{\hbar}(I_m, I_n^d) = \frac{C_m^T C_n^d + I_m^T \bar{I} + I_n^{dT} \bar{I} - \|\bar{I}\|^2}{\|I_m\|^2}. \tag{4.4}$$

To find the matching location m for a current run-time image I_n^d, it is necessary to search the maximum [2] of the Hausdorff fraction comparing the run-time image with all the database images. Using Eq.(4.4) for retrieving a database image is more efficient than using Eq.(4.3) as the term that is computed for every m, respectively $C_m^T C_n^d$ and $I_m^T I_n^d$, is a comparison performed on a low-dimension space against a full-image size correlation. The computation time for the remaining terms of Eq.(4.4) becomes negligible for large databases, as some of the terms are computed only once per run time image and the others are pre-computed. This results in an image retrieval process significantly faster than the direct comparison of the images, as in traditional eigenspace matching.

One important issue with approximating the Hausdorff fraction is to include some

[2] Contrasting to the methods presented in the preceding sections, the Hausdorff fraction is maximised because it is, in essence, a correlation.

tolerance at the matching step. Huttenlocher et al. [53] build the eigenspace using both dilated and undilated model views and pre-process the run time edge images to dilate the edges. In our pre-processing we use low pass filtering instead of edge dilation. The purpose is to maintain the local maxima of gradient magnitude at edge points while enlarging the matching area. We found this to be a good tradeoff between matching robustness and accuracy.

To test this view-based approximation we collected a sequence of images, acquired at different times, 11am and 5pm, near a large window. Figure 4.5 shows the significant changes in illumination, especially near the large window at the bottom left hand side of each omni-directional image. Even so, the view based approximation can correctly

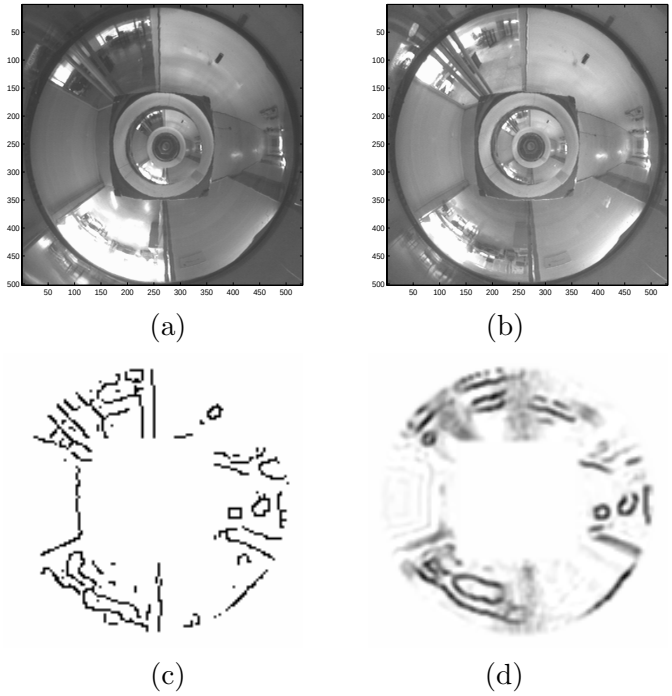


Figure 4.5: (a) An omni-directional image obtained at 11:00, (b) one obtained at 17:00; (c) An edge-detected image and (d) its retrieved image.

determine that the unknown image shown in Figure 4.5(a) was closest to the database image shown in Figure 4.5(b), while PCA based on brightness distributions would fail. For completeness, Figure 4.5 (c) and (d) shows a run-time edge image and its corresponding retrieved image using the eigenspace approximation to the Hausdorff fraction.

4.2.4 Integration of Topological Navigation and Visual Path Following

The mobile robot in continuous operation is most of the time performing topological navigation. At some points of the mission the navigation is required to change to the visual path following modality. Then the robot needs to retrieve the scene features (straight lines in our case) chosen at the time of learning that specific visual path following task.

The search for the features can be approached as a general pattern matching problem

using e.g. a generalised Hough transform as in [114, 27]. We approach the problem by coordinating the two navigation modalities. To find the features, the uncertainty of the location of the robot is controlled by using more detailed topological maps and by increasing the searching regions of the features otherwise bounded according to the maximum speed of the robot.

At the initialisation of the system, the robot will start normally at a known docking place, but if a failure occurs during the operation the robot may have to restart at an unknown (within the topological map) position. When starting at the docking place the undocking visual path following task may be immediately elicited. If starting at an unknown place, i.e. a drop-in-scene case, then the self-localisation is found using the topological localisation module.

The combination of omni-directional images and the Topological and Visual Path Following navigation strategies are illustrated by the complete experiments described in this chapter. We believe that the complementary nature of these approaches and the use of omni-directional imaging geometries result in a very powerful solution to build efficient and robust navigation systems.

4.3 Experimental Results

The experiments described in this chapter were undertaken at the Instituto de Sistemas e Robótica (ISR), in Lisbon, Portugal. It consists of a typical indoor environment, with corridors, offices and laboratories.

We used a TRC Labmate from HelpMate Robotics Inc., equipped with an omni-directional vision system built in-house (see figure 2.5 in chapter 2). This system contains a Cohu CCD camera pointed upwards, looking at a spherical mirror. Grayscale images were captured with a full resolution of 768x576 pixels, and sub-sampled to 128x128 images for PCA and 600x600 for visual servoing and Visual Path Following. All the processing was carried out on-board the mobile platform by a Pentium II 350MHz PC.

The results obtained illustrate the potential of our approach in a variety of different tests. First, we show Topological Localisation results obtained on pre-acquired sequences of images. Finally, we present integrated results of real-world experiments combining the Topological and Visual Path Following navigation modalities.

4.3.1 Topological Localisation Results

We perform two experiments to test the three presented topological localisation methods. In the first experiment we test that the images after compression by the various methods are still sufficiently different to yield correct localisation results, and in the second experiment we test the robustness of the methods against illumination changes.

The experiments are based on three sequences of images: one database sequence describing the environment and two run-time sequences acquired along a fraction of the

represented environment. One of the run time sequences was acquired at a time of the day different to the database set, resulting therefore in very different lighting conditions.

Experiment 1: the run time sequence, as compared to the database, is acquired under similar illumination conditions, the length of the traversed path is about 50% of the original and the images are acquired at a different sampling frequency (distance between consecutive images). Figure 4.6 shows that the three methods give similar localisation results, as desired. The small differences among the methods are due to the distinct image database (appearance set) construction techniques. The figure shows that in the current experiment the three methods despite compressing the information, preserve enough detail to distinguish each image relatively to all the others.

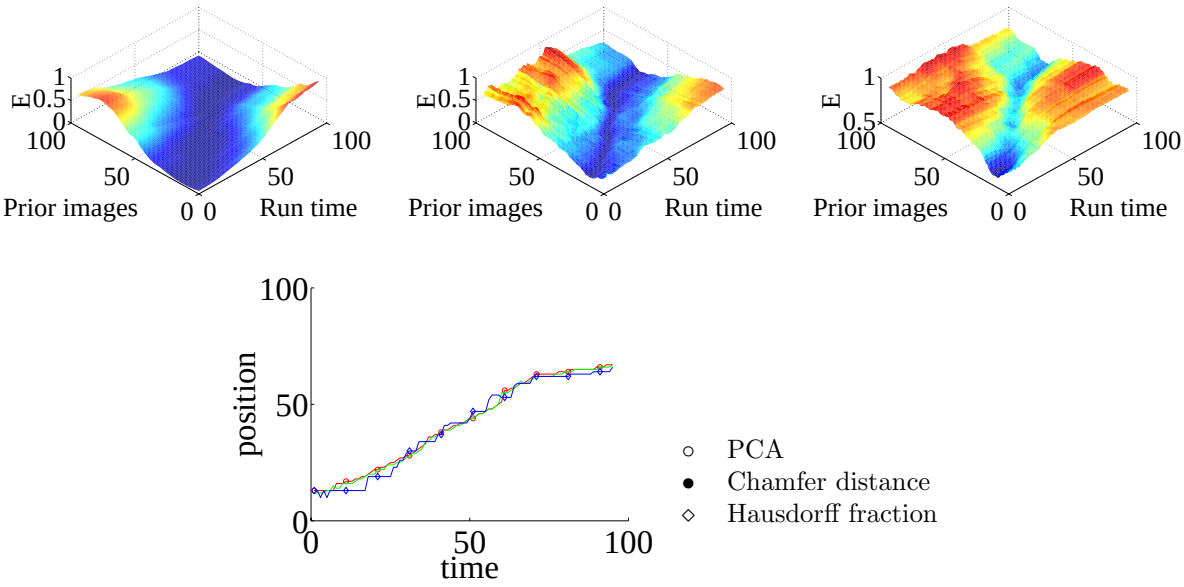


Figure 4.6: Three methods of topological localisation, (top, from left to right): localisation based on PCA, Chamfer distance and Hausdorff distance. The clear valleys show that there is enough information to distinguish the robot locations. (Bottom) localisation as found by each of the methods i.e. ordinates corresponding to minimum values found at each time instant on the 3D plots.

Experiment 2: figure 4.7 shows topological localisation as found by each of the methods for two sequences taken in the same path but at different times of the day, resulting in very different lighting conditions. We can see that the method based on PCA, i.e. the one using directly brightness values, fails to obtain correct locations particularly at the last part of the test, while the other two methods, which are based on edges, obtain good results.

As expected the edges based methods are more suited to dealing with very different illuminations. In our navigation experiments we use mainly the PCA over brightness values, as most of our scenario is not subject to large illumination changes, and using brightness values is more informative than using only edges. For the parts of the scene where illumination can change significantly we use the Hausdorff based method. The

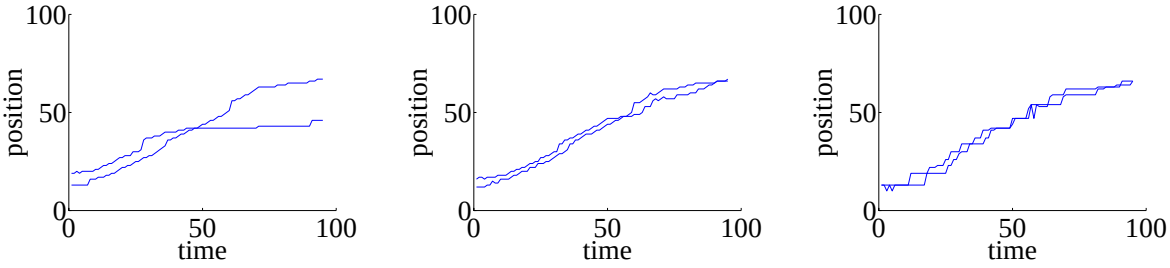


Figure 4.7: Topological localisation experiments using the three methods over two sequences acquired under different lighting conditions. From left to right: localisation based on PCA, Distance transform and Hausdorff distance.

reason of its choice when compared to the Distance transform, is that it is faster for the first localisation at dropped-in-scene situations.

4.3.2 Combined Navigation Experiments

The concluding experiment integrates global and local navigational tasks, by combining the *Topological Navigation* and *Visual Path Following* paradigms.

To navigate along the topological graph, we still have to define a suitable vision-based behavior for corridor following (*links* in the map). In different environments, one can always use simple knowledge about the scene geometry to define other behaviors. We exploit the fact that most corridors have parallel guidelines to control the robot heading direction, aiming to keep the robot centered in the corridor.

The visual feedback is provided by the omni-directional camera. We use *bird's eye views* of the floor, which simplifies the servoing task, as these images are a scaled orthographic projection of the ground plane (i.e. no perspective effects). Figure 4.8 shows a top view of the corridor guidelines, the robot and the trajectory to follow in the center of the corridor.

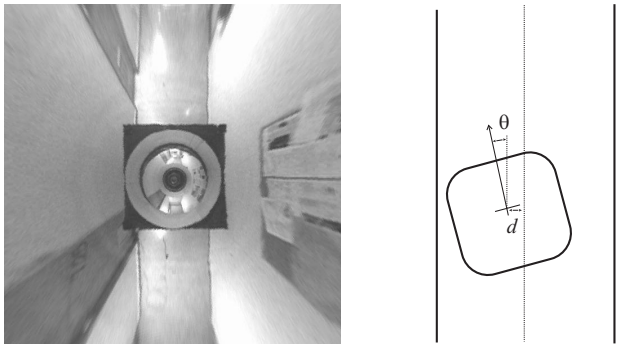


Figure 4.8: (Left) Bird's eye view of the corridor. (Right) Measurements used in the control law: the robot heading θ and distance d relative to the corridor centre. The controller is designed to regulate to zero the (error) measurements actuating on the angular and linear speeds of the robot.

From the images we can measure the robot heading with respect to the corridor guidelines and the distance to the central reference trajectory. We use a simple kinematic

planner to control the robot's position and orientation in the corridor, using the angular velocity as the single degree of freedom.

Notice that the use of bird's eye views of the ground plane simplifies both the extraction of the corridor guidelines (e.g. the corridor has a constant width) and the computation of the robot position and orientation errors, with respect to the corridor's central path.

Hence, the robot is equipped to perform Topological Navigation relying on appearance based methods and on the behavior for corridor following. This is a methodology for traversing long paths. For local and precise navigation the robot uses Visual Path Following as detailed in chapter 3. Combining these behaviours the robot can perform missions covering extensive areas while achieving local accurate goals. In the following we describe one such mission.

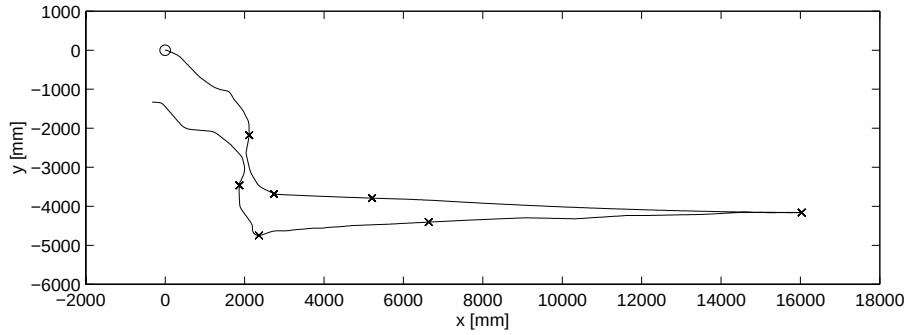
The mission starts in the Computer Vision Lab. Visual Path Following is used to navigate inside the Lab, traverse the Lab's door and drive the robot out into the corridor. Once in the corridor, control is transferred to the Topological Navigation module, which drives the robot all the way to the end of the corridor. At this position a new behaviour is launched, consisting of the robot executing a 180 degree turn, after which the topological navigation mode drives the robot back to the Lab entry point.

Figure 4.9: Experiment combining visual path following for door traversal and topological navigation for corridor following.

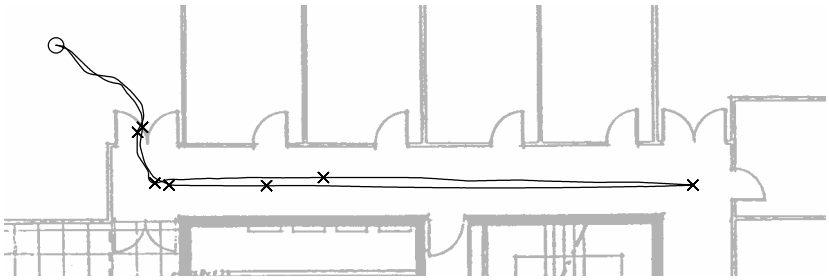
During this backward trajectory we use the same image eigenspaces as were utilised during the forward motion by simply rotating, in real-time, the acquired omni-directional images by 180 degrees. Alternatively, we could use the image's power spectrum or the Zero Phase Representation [84]. Finally, once the robot is approximately located at the lab entrance, control is passed to the Visual Path Following module. Immediately it locates the visual landmarks and drives the robot through the door. It follows a pre-specified path until the final goal position, well inside the lab, is reached. Figure 4.9 shows an image sequence to relate the robot's motion during this experiment.

In Figure 4.10(a) we used odometric readings from the best experiment to plot the

robot trajectory. When returning to the laboratory, the uncertainty in odometry was approximately 0.5m. Thus, door traversal would not be possible without the use of visual control. Figure 4.10(b), shows the actual robot trajectory, after using ground truth measurements to correct the odometric estimates. The mission was successfully accomplished.



(a)



(b)

Figure 4.10: A real world experiment combining Visual Path Following for door traversal and Topological Navigation for long-distance goals. Odometry results before (a) and after (b) the addition of ground truth measurements.

This integrated experiment shows that omni-directional images are advantageous for navigation and support different representations suitable both for Topological Maps, when navigating between distant environmental points, and Visual Path Following for accurate path traversal. Additionally, we have described how they can help in coping with occlusions, and with methods of achieving robustness against illumination changes.

4.4 Concluding Notes

We presented a method for the visual-based navigation of a mobile robot in indoor environments, using an omni-directional camera as the sole sensor.

Our key observation is that different navigation methods and environmental representations should be used for different problems, with distinct requirements in terms of processing, accuracy, goals, etc.

We distinguish between missions that involve traveling long distances, where the exact trajectory is unimportant (e.g corridor following), as opposed to other cases where the robot must accurately follow a pre-specified trajectory (e.g. door traversal). For these two types of missions we presented two distinct paradigms: *Topological Navigation* and *Visual Path Following*.

Topological Navigation relies on graphs that describe the topology of the environment. The qualitative position of the robot on the graph is determined efficiently by comparing the robot's current view with previously learned images, using a low-dimensional subspace representation of the input image set. At each node (landmark), a different navigation behavior can be launched, such as entering a door or turning left.

Whenever the robot needs to move in cluttered environments or follow an exact path, it resorts to *Visual Path Following*. In this case, tracked features are used in a closed loop visual controller to ensure that the robot moves according to the desired trajectory.

Omni-directional images are used in these two navigation modes to build the necessary environmental representations. For example, the *Bird's Eye Views* of the ground floor substantially simplify navigation problems by removing perspective effects.

Combining *Topological Navigation* and *Visual Path Following* is a powerful approach that leads to an overall system which exhibits improved robustness, scalability and simplicity.

Chapter 5

Interactive Scene Modelling

We start by proposing a first method for 3D reconstruction of structured environments from an omnidirectional image, which is based on a ground plane map identified by the user.
Then we use a generalised method, that reconstructs the scene from single or multiple images, and permits a generic world relative reconstruction frame. It is based on a reduced amount of user information, in the form of 2D pixel coordinates, alignment and coplanarity properties amongst subsets of the corresponding 3D points.
Just a few panoramic images are sufficient for building the 3D model, as opposed to a larger number of "normal" images that would be required to reconstruct the same scene [57, 96].

5.1 Introduction

The construction of scene models is a well known problem in the computer graphics and in the computer vision communities. While in the former there is traditionally a strong emphasis in using precise user-defined geometric and texture data, in the latter the emphasis is more on the direct use of images to automatically correspond and generate depth or shape maps. Recently, many works started to combine with success both approaches in a way well tuned for each purpose [19, 57, 15, 67, 23, 87, 97, 96].

Our motivation comes from the tele-operation of mobile robots. Given an image of a structured environment and some user input regarding the geometry of the environment, one can reconstruct the 3D scene for visualisation or for specifying actions of the robot. The robot is equipped with an omnidirectional camera that provides a 360° view of the environment in a single image. The wide field of view of omnidirectional vision sensors makes them particularly well suited for fast environmental modelling.

The modelling we are interested in is concerned with obtaining 3D textured structures representing the world around the robot. Hence, we divide the modelling in two steps: (i) obtaining the structure and (ii) texture mapping.

The 3D Structure is represented as a set of 3D points forming the basis of a wireframe

model of the scene. It is determined based on image observations and some limited user input. As there is direct user intervention the process of obtaining the structure is termed *Interactive Reconstruction* [96, 95, 97, 33].

The user input consists of 2D points localised in the image and of part of his knowledge of parallelism or perpendicularity of lines and planes of the scene. Typically, the user will identify some points in the omnidirectional images and indicate that some subsets of points are collinear or coplanar.

The texture mapping step of the modelling process is performed after reconstruction. Texture mapping takes as input the reconstructed structure and uses the projection model for assigning, to each point of the 3D model, brightness (radiance) values retrieved from the image.

Chapter organisation

Firstly we introduce a modelling method based on the ground-plane map. Then, we present the general reconstruction method based on co-linearity and co-planarity properties, designed for the case of omnidirectional cameras. Finally we present our results on interactive scene modelling based on single and multiple images. We present also an application of interactive models on building human-robot interfaces.

5.2 A Map Based Modelling Method

In the case of structures described by floor-plans, 3D models can be extracted from omnidirectional images using the Bird's Eye View. For example at a corridor-corner, see Fig.5.1, the Bird's Eye View shows directly the floor-plan and the user just needs to select the relevant ground lines. In order to complete the model it is only necessary to find the heights of the walls and the appropriate texture mapping.

However we are interested in using directly the omnidirectional images as their field of view is larger and therefore the reconstructed models are enlarged too. A floor map consisting of a set of points can be obtained from an omnidirectional image. After *back-projecting* (detailed in chapter 2) and scaling the points to have constant height, the resulting map is the same as extracted from a Bird's Eye View. Once the floor-plan is obtained, the reconstruction consists just in lifting the model to 3D. This requires knowing the height of the walls in the same scale as that of the floor-plan.

The height of a wall can be computed from the direction of light-rays corresponding to the imaged points of that wall, i.e. using again back-projection. Given the back-projection vectors of two points on a vertical line of the wall, one point on the floor and the other on the top, the wall top is found by scaling the respective vector to make equal the ground-projections of both rays (note that the vector pointing to the ground point is already correctly scaled as indicated by the floor plan).

To conclude, we obtain a reconstruction method where all image input comes directly from the omnidirectional images. The user specifies points on the ground plane and on

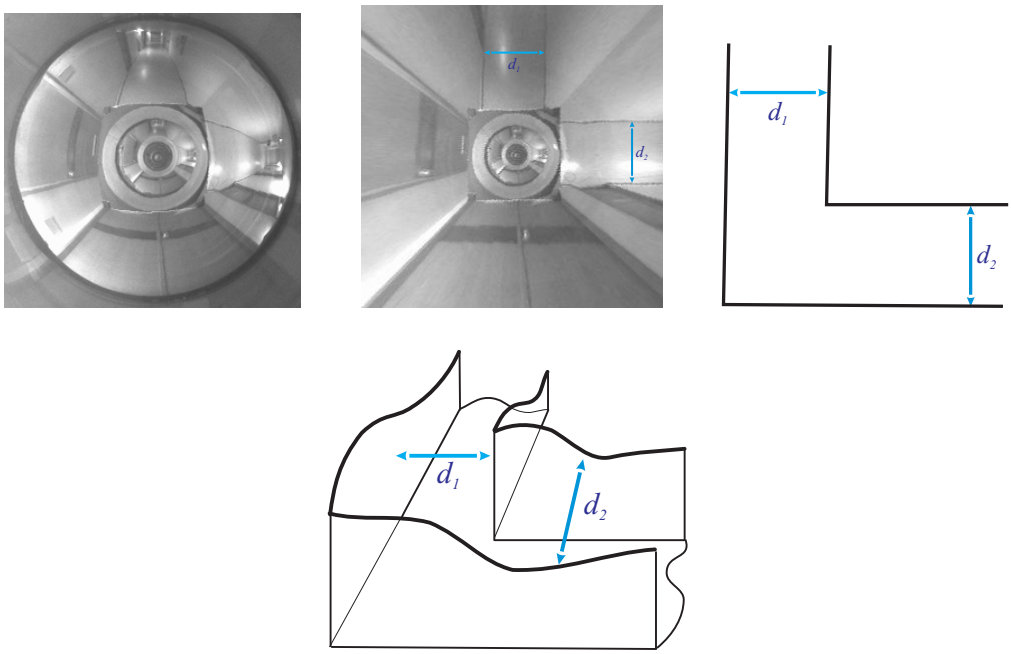


Figure 5.1: Map based reconstruction method. (Top left) Original omnidirectional image, shown upside-down for simpler comparison with the next images. (Top right) The Bird's Eye View gives the floor-plan of the a corridors corner which is already a geometric model of the scene. (Bottom) The 3D model is completed by specifying the wall heights.

the walls. Each wall encompasses points on the ground and points defining the wall boundaries.

Summary of the proposed reconstruction algorithm:

1. User input: the user indicates planar surfaces representing walls and floor. Typically the user chooses a set of points and then defines polygons upon those points, specifying which ones are on the ground plane.

2. Back-project all image points to be reconstructed. The result is a set of unit length 3D vectors.

3. Scale all vectors resulting from ground points to have a constant height. The camera axis, z is assumed vertical and therefore ground points must have a constant z value.

4. For each wall determine, from its ground points, the distance to the camera and a normal vector. Scale all wall points to be in a plane characterised by the distance and the normal vector just determined.

Figure 5.2 shows a result of reconstruction. The input are ground points indicating the corridor and corridor-end walls. A number of extra points indicate the wall-tops.

Even though we have not explicitly imposed the geometric properties of the scene, namely orthogonality and parallelism among the walls, they are approximately retrieved in the final reconstruction.

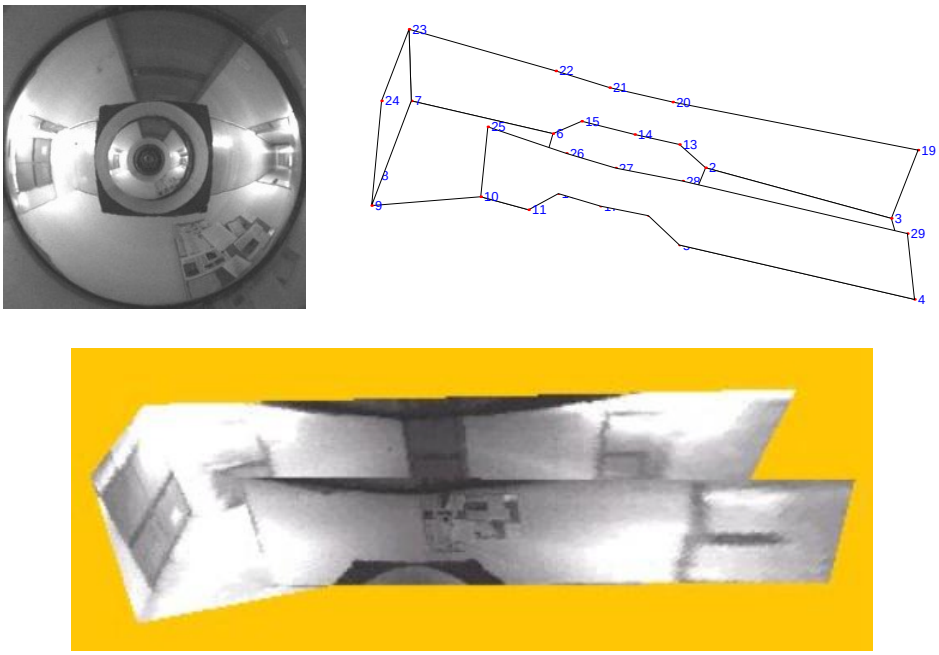


Figure 5.2: Interactive 3D Model from an Omnidirectional Image.

One important observation about the modelling method just introduced is that back-projection provides the same type of information as the Bird's Eye View dewarping. After back-projecting and scaling the ground points the resulting map is the same as extracted from the Bird's Eye View. It is interesting to note that the map based modelling method was firstly envisaged for perspective images and then was transported to omnidirectional images through the use of back-projection.

This approach shall be followed again in the next section, where we describe a modelling method generalising the one just presented.

In the generalised method the user is free to set co-linearity and co-planarity properties along any of the directions of the world coordinate system. The generalised method will be the main subject of the remaining of the chapter.

5.3 Modelling Based on Alignment and Coplanarity Properties

The enlarged field of view of omnidirectional cameras makes them particularly well suited for scene modelling. Just a few panoramic images are sufficient for obtaining a good perception of the entire scene, as opposed to a larger number of "normal" images that would be required to display the same scene [57, 96].

We present now a generalised reconstruction method based on co-linearity and co-planarity properties. The user is free to those properties along any of the directions of the world coordinate system. The 3D relationships are then combined at the same time for a global solution weighting all data equally and therefore gaining extra robustness against

user input noise. These are advantages relative to the map based method.

A world coordinate system is the most natural choice for observing geometric properties such as parallelism or perpendicularity present in the scene structure. To integrate in a straightforward manner those geometrical properties it is therefore convenient to formulate the reconstruction problem in the desired world coordinate system, instead of using the camera coordinate system.

We start defining the reconstruction reference frame and preparing the data for reconstruction. The image points become as acquired by a pin-hole camera. Then we can use the reconstruction algorithm of Grossmann et al. [41] designed for perspective images. To complete the modelling we perform the texture mapping.

5.3.1 Back-projection and the Reference Frame

Prior to the reconstruction process, omnidirectional images are back-projected to a spherical surface, from which perspective images are simple to extract. This is an automatic process for most of the omnidirectional camera types [38], as it depends only on the omnidirectional camera parameters [33]. The goal of this section is to derive an image formation model that is a pin-hole camera whose orientation is conveniently related with the world structure.

Vanishing points, i.e. image points representing scene points at an infinite distance to the camera [10], represent scene directions through which the reconstruction can be conveniently done. A vanishing point is the intersection in the image of the projection of parallel 3D lines. If one has two image lines parallel in 3D, defined by two points, AB and CD, then the corresponding vanishing point $\mathbf{r}$ is:

$$\mathbf{r} = (A \times B) \times (C \times D) \tag{5.1}$$

where the points A, B, C and D are in homogeneous coordinates obtained according to the back-projection Eq.(2.34). If more points are available in each line or more lines in each set, least squares estimates replace the external products in Eq.(5.1) and more accurate estimates are obtained for the vanishing point coordinates [96].

Given three (unit-norm) vanishing points, $\mathbf{r_1}$, $\mathbf{r_2}$ and $\mathbf{r_3}$, representing three world orthogonal directions, it is possible to obtain perspective images in a reference frame built on those directions using Eq.(2.35) with $R = [\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{r_3}]$.

The optical axis of our sensor, z, is in the vertical direction and is therefore aligned with one of the most frequent (and representative) directions of lines in the scene. The corresponding vanishing point in the reference frame defined by back-projection Eq.(2.34) is simply $[0\ 0\ 1]^T$, and the new reference frame, associated to the rotation matrix of

Eq.(2.35), takes the form of a rotation about the z axis:

$$R = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In what follows, we rotate the image so that the x and y axes of the camera frame coincide with that of the world reference frame. One thus has, in Eq.(2.35),

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [\mathbf{e}_1\, \mathbf{e}_2\, \mathbf{e}_3] \tag{5.2}$$

where the $\mathbf{e}_i$ form the canonical basis of $\Re^3$.

5.3.2 Reconstruction Algorithm

The reconstruction process aims at obtaining 3D data from image points and limited user input. The user provides co-linearity and co-planarity properties of the scene. The user should be able to specify the geometric properties in any of the most relevant directions of the scene.

The algorithm of E. Grossmann et al [41, 42], targets and solves precisely this objective. Their formulation works upon single or multiple images (viewpoints) and brings in addition an algebraic procedure to test whether there is a single solution (up to a scale factor) to the reconstruction problem. 3D relationships are combined at the same time for a global solution weighting all data equally. We summarise here this algorithm in order to maintain this chapter self-contained.

We have just shown that, for all practical effects, we can consider that the input image is obtained by a pinhole camera aligned with the world reference frame. Chosen the coordinate system, then the 3D points to estimate are precisely defined and their relation with the image points may be written. Next it will be integrated the auxiliary geometric information, which serves to determine what distinct coordinates are that will be estimated.

Let $p = [u\ v\ 1]^T$ be the projection, in homogeneous coordinates, of a 3D point, $P = [P_x\ P_y\ P_z]^T$ that we want to reconstruct. Then, if we consider a normalized camera [25], whose orientation is represented by a 3×3 rotation matrix R, we have the following:

$$p = \lambda RP \tag{5.3}$$

where λ is a scaling factor. As is usual, we choose 0 as the origin of the coordinates for reconstruction.

We can rewrite Eq.(5.3) as $p \times RP = 0$. Representing the external product as a matrix

product, denoting S_p as the Rodriguez matrix of p, we obtain:

$$S_p R P = 0_3 \tag{5.4}$$

which is a linear system in the coordinates of the 3D point. Generalising this system to N points we again obtain a linear system:

$$A.\mathcal{P} = 0_{3N} \tag{5.5}$$

where A is block diagonal and $\mathcal{P}$ contains the $3N$ tridimensional coordinates that we wish to locate:

$$A = \begin{bmatrix} S_{p_1} R & & & \\ & S_{p_2} R & & \\ & & \ddots & \\ & & & S_{p_N} R \end{bmatrix} \ , \ \mathcal{P} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_N \end{bmatrix} \tag{5.6}$$

Thus, A is of size $3N \times 3N$. Since only two equations from the set defined by equation (5.4) are independent, the co-rank of A is equal to the number of points N. As expected, this shows that there is an unknown scale factor for each point.

Now, adding some limited user input, in the form of co-planarity or co-linearity point constraints, a number of 3D coordinates become equal and thus the number of columns of A may be reduced. As a simple example, if we have 5 points we have 5×3 distinct coordinates, i.e. the number of columns of A. Now, if we impose the constraint that points P_1, P_2, P_3 are co-planar, with constant z value and points P_4, P_5 are co-linear, with a constant (x, y) value, then coordinates P_{2z}, P_{3z}, P_{5x}, P_{5y} are dropped from the linear system defined by equation (5.5). Thus, the total number of free coordinates is reduced from the initial 15 to 11.

Given sufficient user input, the co-rank of A becomes 1 and thus it has a single (up to scale) null vector $\mathcal{P}^*$. The general form of the solution for Eq.(5.5) is thus :

$$\mathcal{P} = \lambda \mathcal{P}^* \tag{5.7}$$

where λ is an arbitrary scale factor.

Equation (5.7) says that, even with a single view, there is no ambiguity in the reconstruction, other than that –well-known– of scale. Hence, an algebraic criterium for detecting non single-block structures, i.e. structures with separated parts scaled differently, can be drawn directly on the co-rank of A: the user defined a single-block structure if the co-rank of A is one. In practice A is full-rank due to the noise in the image points indicated by the user. However, it is possible to construct a twin matrix of A, based only on the (noiseless) geometrical properties input, also provided by the user, where the co-rank criterium can still be applied [41].

The reconstruction algorithm is easily extended to the case of multiple cameras. Assuming that there are additional cameras whose orientations are known relative to the

first one (this is not a limitation since as usual the world frame is selected from vanishing points), then the only novel data is the translation, t of each new camera relatively to the first one:

$$p = \lambda(RP - Rt) \tag{5.8}$$

where t is chosen to be zero for the first camera, as in the case of single images, and t is represented by $t_1 \dots t_j$ for j additional cameras.

Considering as an example two additional cameras and doing the same derivation done for a single image then similar A and $\mathcal{P}$ are defined for each camera using Eq.(5.6) and the problem has six new degrees of freedom corresponding to the two unknown translations t_1 and t_2:

$$\left[\begin{array}{ccc|c|c} A_1 & & & & \\ & A_2 & & -A_2.\mathbf{1}_2 & \\ & & A_3 & & -A_3.\mathbf{1}_3 \end{array}\right] \left[\begin{array}{c} \mathcal{P}_1 \\ \mathcal{P}_2 \\ \mathcal{P}_3 \\ t_1 \\ t_2 \end{array}\right] = 0 \tag{5.9}$$

where $\mathbf{1}_2$ and $\mathbf{1}_3$ are matrices to stack the blocks of A_2 and A_3.

As before co-linearity and co-planarity properties reduce the co-rank of the matrix. In addition, there will be frequent to observe the same point in different images. Each one decreases three unknown point coordinates. When the co-rank is one, then there is a single block structure and reconstruction is performed again up to a scale factor.

A concluding important point is that the reconstruction of the whole scene is obtained in a single step.

Using the reconstruction method just presented, we build 3D geometric descriptions of the scene. In order to complete the scene modelling we complement the geometric model with texture mapping. This is described in the next section.

5.3.3 Texture Mapping

Texture mapping is the process of finding the brightness (radiance) value for each 3D point of the world model. The radiance values are found in the omnidirectional image, given the projection function applied to the 3D points. The projection function depends on the type and specific parameters of each omnidirectional camera, as seen in the first chapter.

Since the world coordinate system, taken from the vanishing points, is not necessarily coincident with the one of the camera, then the 3D points of the world need to be transformed to the camera frame. This is done using the inverse of the rotation matrix built from the vanishing directions (introduced in the beginning of the section).

In the case of multiple cameras, the reconstruction is carried out in the coordinate system of the first camera. In order to find the projections of the reconstructed points in every image it is necessary to transform the points to the local coordinate system. This is achieved using the rotation matrices found a priori for each of the cameras and the translation vectors vectors estimated in the reconstruction process.

The resolution of the 3D model texture is chosen according to the image resolution. Each 3D face projects approximately as a sector on the omnidirectional image and therefore its arc-length is an estimate of the available data. Note that the arc-length varies in accordance to the radius, being largest at the rim of the omnidirectional image. Usually we take the middle radius which is a good compromise between resulting quality and over-sampling.

In the case of reconstruction based on multiple images, where each wall comprises a number of overlapping faces indicated by the user at different images, then the resolution can be chosen from the closest image or as a weighted value among the values provided by the various images.

5.4 Results

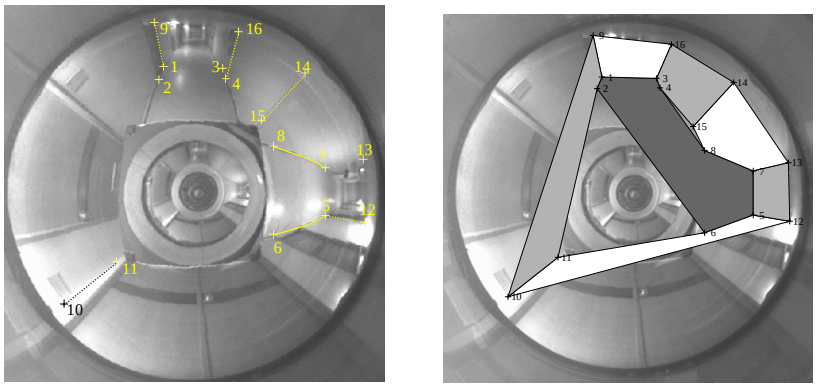
The scenario we are modelling is our own Institute. It consists of a typical indoor environment, with corridors, offices and laboratories. We used the omnidirectional camera based on the spherical mirror (which is described in detail in section 2.2.3) mounted on top of a mobile robot. We show results of the modelling method based on co-linearity and co-planarity properties, based on single and multiple images.

Figure 5.3 shows an omnidirectional image and superposed user input. This input consists of the 16 points shown, knowledge that sets of points belong to constant x, y or z planes and that other sets belong to lines parallel to the x, y or z axes. The table bellow the images details all the user-defined data. Planes orthogonal to the x and y axes are in light gray and white respectively, and one horizontal plane is shown in dark gray (the topmost horizontal plane is not shown as it would occlude the other planes). The coordinates in the original image were transformed to the equivalent pin-hole model coordinates and used for reconstruction.

Figure 5.4 shows the resulting texture-mapped reconstruction. This result is interesting given that it required only a single image and limited user input to reconstruct the surroundings of the sensor.

Figure 5.5 shows a reconstruction based on multiple images. A set of eight omnidirectional images was used for user input and texture mapping. The images were taken at the corridors corners and at the middle of the corridors-length. The images are shown in appendix C. This is an example as the combination of multiple images permits the modelling of large scenes.

In this section we presented results of interactive reconstruction based on omnidirectional images. The omnidirectional images allowed to obtain global scene representations using a reduced number of images, as compared to the number of standard limited field of view images that would be necessary to cover the same areas. Interactive reconstruction proved effective on obtaining 3D models of low textured scenes, which would be difficult to obtain with conventional methods based on stereo or motion data. In the following section, we present an application of the reconstructed scene models, namely building



Axis	Planes	Lines
x	$(1,2,9,10,11)$, $(3,4,14,15,16)$, $(5,7,12,13)$	
y	$(5,6,10,11,12)$, $(7,8,13,14,15)$, $(1,3,9,16)$	$(1,2)$
z	$(1,2,3,4,5,6,7,8)$, $(9,12,13,16)$	

Figure 5.3: User-defined planes and lines. (Top-left) Original image with superposed points and lines localised by the user. (Top-right) Planes orthogonal to the x, y and z axis are shown in light gray, white, and dark gray respectively. Table: the numbers are the indexes shown on the image. The first column indicates the axis to which the planes are orthogonal and the lines are parallel.

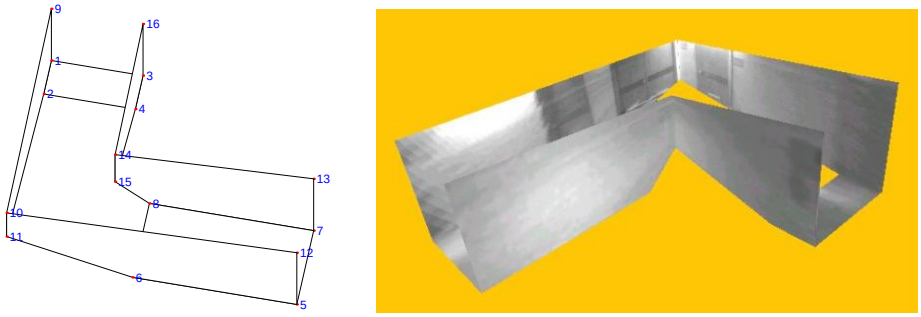


Figure 5.4: Interactive modelling based on co-planarity and co-linearity properties using a single omnidirectional image. (Left) Reconstruction result, (right) view of textured mapped 3D model.

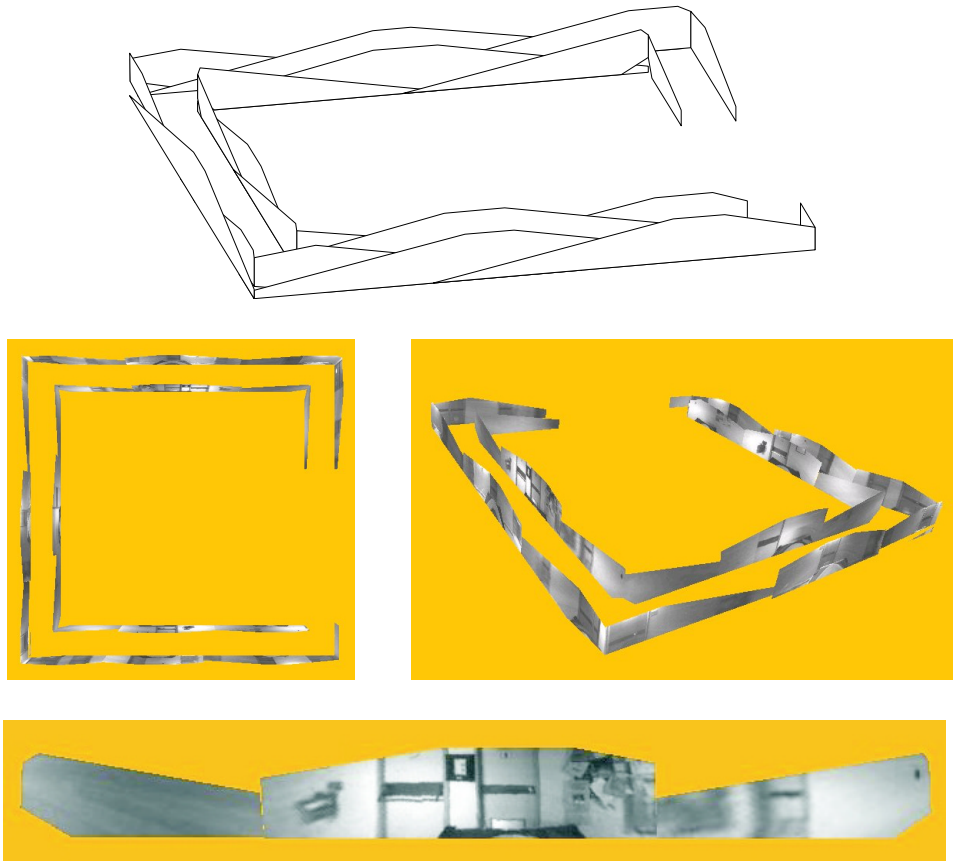


Figure 5.5: Interactive Reconstruction based on multiple Omnidirectional Images. (Top) Reconstruction result, (middle) two views of the textured map model, (bottom) enlarged view of one of the walls. A simple heuristic has been applied to remove the overlapping among co-planar surfaces.

intuitive visual human-robot interfaces.

5.5 Application: Human Robot Interface

Once we had developed effective methods for precise navigation on pre-defined paths and autonomous qualitative robot navigation along a topological map, we turned our attention to developing an intuitive user interface from which to select subtasks. While final experiments have yet to be undertaken, in this section we show how to construct this interface.

Our setup is based on a mobile platform equipped with an omnidirectional camera, as described for the navigation experiments, and in addition a radio modem for wireless communications with a base station (see Figure 5.6). The radio link has low bandwidth, close to one order of magnitude less than the necessary for live video. Considering in addition exceptional peak loads of the network causing unpredicted delays, it is not possible to guarantee a live-interaction with the robot. This is a reason for considering having scene

models close to the user to help bridging communication delays.

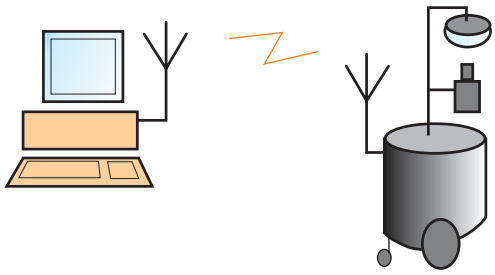


Figure 5.6: The human robot interface is implemented over a wireless network. The robot has the navigation skills described on previous chapters. The base station is dedicated to the implementation of the interface.

Each omnidirectional image provides a rich scene description, which the user is free to rotate. This provides a good understanding of the scene and a way to specify simple targets. At the simplest level, there are three modes at which the operator can control the robot:

1. Heading: Move the robot in a certain direction, e.g "go forward", "turn left".
2. Position: Go to a specified position, e.g. docking/parking.
3. Pose: Control position and orientation using a 3D model, e.g. "go to the third office in the second corridor".

The robot heading is easily specified by the user by simply clicking on the desired direction of travel in a panoramic image. An immediate benefit of using omnidirectional images is that every heading direction can be specified with a single command (see Figure 5.7 - left). This gives the operator a great deal of flexibility when deciding in what direction the robot should travel while simultaneously allowing a speedy decision to be made. The robot then follows the desired direction until it receives a new command from the user. We note here that the operator does not specify (x, y) coordinates for the robot to follow. The only goal of the interface is to provide enough information to allow the operator to make a decision on the robot's heading. When the operator is using this interface, the Topological Navigation modality is used. Thus, there is a natural correspondence between the design of the user interface and the action required of the robot.

In order to specify more complex missions, encompassing (x, y) locations, we chose a new representation based on *bird's-eye views* of the robot's surroundings. In this representation the user clicks a point which is a target location for the robot (see Figure 5.7 - right). If the target location is within the region covered by the topological map, then the robot uses the navigation tasks already available to move to the target point. Otherwise, the user can add a new Visual Path Following task by choosing landmarks and via points in the bird's-eye view image. The way points are linearly interpolated by the path planner

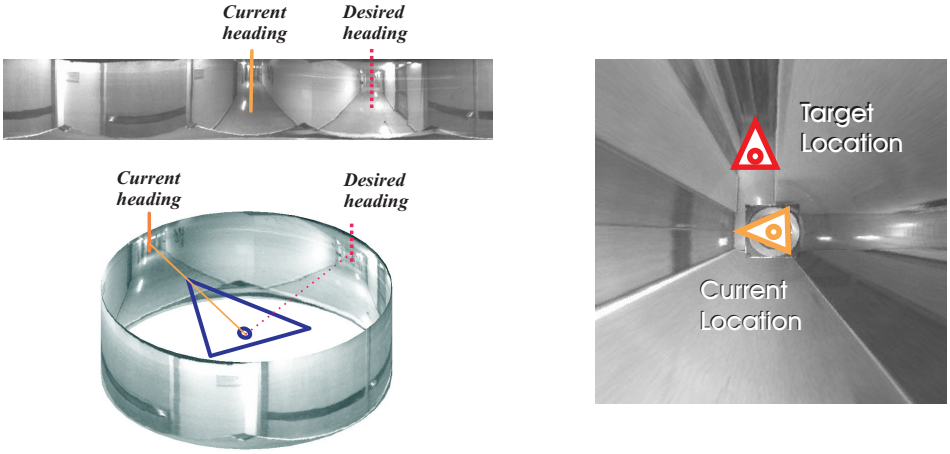


Figure 5.7: Tele-operation interfaces based on dewarped omnidirectional images. (Left-top) To specify the target heading the robot is to follow, the operator simply clicks on the desired direction in a panoramic image. (Left-bottom) The current and desired headings from an outside point of view. (Right) A bird's eye view of the robot's surroundings allows specifying xy locations.

module running on the robot. After definition, the path is then followed, by relying upon self-localisation measurements relative to the landmarks.

Interactive models bring another degree of freedom. They give the observer in addition the freedom in the pose. The user gains extra feeling and experience of local immersion as interaction with the world-scene is generated on the base station and therefore is not delayed by the communications, happens as a live event. See figure 5.8.

Given that the targets are specified on interactive models, i.e. models built and used on the user side, they need to be translated as tasks that the robot understands. The translation depends on the local world models and navigation sequences the robot has in its database.

Most of the world that the robot knows is in the form of a topological map. In this case the targets are images that the robot has in its database (topological map). The images used to build the interactive model are nodes of the topological map. Thus, a fraction of a distance on an interactive model is translated as the same fraction on a link of the topological map.

At some points there are precise navigation requirements. Many of these points are identified in the topological map and will be invoked automatically when travelling between nodes of the topological map. Therefore, many of the Visual Path Following tasks performed are not asked explicitly by the user. However, if the user desires he may add new Visual Path Following tasks. In that case the user chooses landmarks, navigates in the interactive model and then asks the robot to follow the same trajectory.

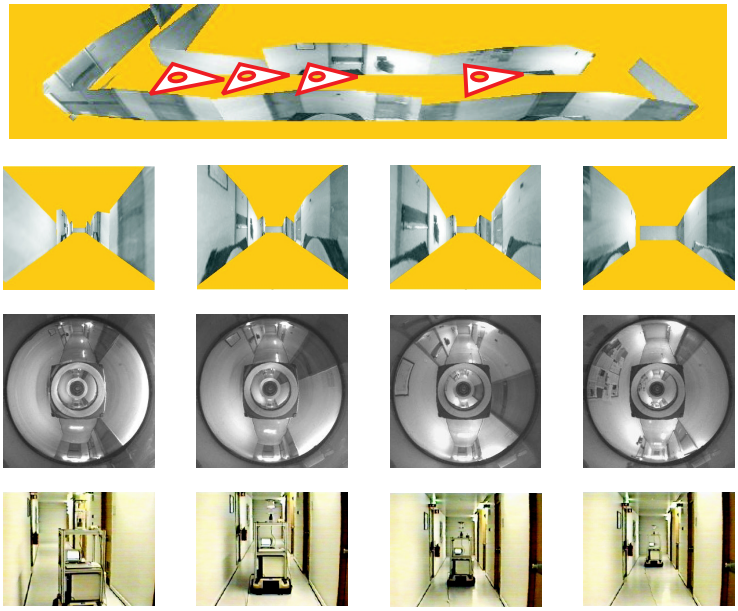


Figure 5.8: Tele-operation interface based on 3D models: (top) tele-operator view, (middle) robot view and (bottom) outside view.

5.6 Concluding Notes

Using a single-view reconstruction technique requiring limited user input, one obtains a model of the environment (surroundings) of the robot that carries the sensor. The main point in transporting the method for omnidirectional cameras, was the development of a novel back-projection model for cameras modeled by the Unified Projection Model (detailed in chapter 2). We have shown that our sensor equipped with a spherical mirror can be approximately modelled by this projection model and consequently back-projection can be performed with the developed generalised model.

The reconstruction method handles also multiple viewpoints to create large scene models. The reconstruction of a model encompassing four corridors has been shown. An application of the interactive models has been devised. It is the specification of planar surfaces where mosaicking can be performed to improve the quality of the data (e.g. texture resolution). Preliminary results were shown.

Interactive modelling offers a simple procedure for building a 3D model of the scene where a vehicle may operate. Even though the models do not contain very fine details, they can provide the remote user of the robot with a sufficiently rich description of the environment. The user can instruct the robot to move to desired position, simply by manipulating the model to reach the desired view point. Such simple scene models can be transmitted even with low bandwidth connections.

There are a number of ways to further apply and extend this work. We plan on carrying out extended closed-loop control experiments verifying the applicability of our navigation framework. Another research direction is that of automatically estimating

geometric constraints that can be used for 3D reconstruction, hence keeping the user intervention to a minimum.

Chapter 6

Conclusions and Future Work

This chapter concludes the dissertation. We review the principal components of our work, namely sensor, navigation modalities and human-robot interface design. Then we discuss the individual and the combined design of those components. Finally we establish a number of directions for future work.

6.1 Summary

In this thesis we addressed the problem of mobile robot navigation based on omnidirectional vision. **Chapter 1** introduced the problem and presented motivations for exploring a number of directions. In particular, motivations from the biological and robotics research fields, shaped our approach to encompass three principal aspects: sensor, navigation and human-robot interface design.

Chapter 2 detailed the design of omnidirectional vision sensors. Namely, we presented projection models and design criteria for catadioptric omnidirectional cameras. We also described how to obtain Panoramic and Bird's Eye Views, which are dewarpings of omnidirectional images, that are useful for navigation tasks. The dewarpings can be obtained directly by constant resolution cameras, for which we proposed an unified design methodology. We applied our design methodologies to build omnidirectional cameras based on spherical and hyperbolic mirrors, and a system combining multiple constant resolution properties that is based on a log-polar camera.

Then, in **chapters 3** and **4**, we introduced the navigation tasks. For local and precise navigation we proposed Visual Path Following. Self-localisation was identified as a principal component and a number of methods were presented for computing the robot pose. Visual Path Following was tested in real world settings with both docking and door-crossing experiments being undertaken.

Topological Navigation was used for global and qualitative navigation. It relied on appearance based methods to represent the environment. Scenes encompassing regions of large non-uniform illumination change, may not be correctly represented by appearances comprising only brightness values obtained at specific illuminations, and therefore the

robot may fail to self-localise. We propose the use of edge-based appearance representations on those regions. In order to have graceful localisation degradation, we have used tolerant edge-shape comparison methods based on chamfer or Hausdorff distances.

An extended real world experiment was carried out, combining Topological Navigation and Visual Path Following. The experiment started in a laboratory by undocking the robot and crossing the door using Visual Path Following, then navigating along corridors using Topological Navigation and finally coming-back to the docking position again using Visual Path Following.

Finally, in **chapter 5**, we described interactive scene reconstruction based on omnidirectional images. 3D models were obtained from omnidirectional images and limited user input of geometrical nature. We proposed the use of a method designed for conventional cameras with omnidirectional cameras for which a back-projection model was derived. Back-projection allows us to transform an omnidirectional camera onto a very wide field-of-view pin-hole camera. The reconstruction was tested in corridor environments using single or multiple images. We also presented an application of the 3D models, namely a visual human-robot interface.

The topics of sensor, navigation and human-robot interface design, summarised in the last paragraphs, are naturally interrelated. In the following section we discuss how those interrelations affected on the design options.

6.2 Discussion

Vision is becoming the sensor of choice for the navigation of mobile robots, mainly because it can provide information on the world structure. This is convenient as compared to the integration of internal information (odometry) which accumulates errors over time. Vision also compares favorably with other sensors providing measurements of the world structure, due to the rich amount of information provided.

Omnidirectional vision brings some additional advantages, such as longer feature trackings and simultaneous tracking at multiple disparate viewing directions. For example, in the docking experiment, the mobile robot (chapter 3), tracking the landmark over the entire trajectory, requires omnidirectional vision as the landmark moves relative to the robot in a wide range of azimuthal angles. As another example, successful completion of the door crossing experiment, relies on the tracking of features surrounding the sensor. These types of experiments are not possible with limited field of view (conventional) cameras. Even cameras equipped with pan-and-tilt mounting would be unable to perform the many separate landmark trackings of our experiments.

Omnidirectional vision sensors have non-linear effects on the geometry that hinder the application of conventional robust self-localisation methods. Therefore we chose to augment, or design, our omnidirectional vision sensors to be capable of providing output views (images) with simple geometries. Our sensors output Panoramic and Bird's Eye views that are images as obtained by cylindrical retinas or pin-hole cameras imaging the

ground plane. Panoramic and Bird's Eye views are useful for navigation, namely for servoing tasks, as they make localisation a simple 2D rigid transformation estimation problem.

Designing navigation modalities for the distinct tasks at hand is easier and more effective as compared to designing a single complex navigation mode [8]. Sensor design, as referred, contributes additionally to simpler design of the navigation modalities. Our combined navigation modalities, Visual Path Following and Topological Navigation, constitute an effective approach to tasks containing both short paths to follow with high precision and long paths to follow qualitatively. The experiment encompassing undocking / docking, door crossing and corridors navigation (chapter 4), is an illustration of a complex task handled by our approach.

The navigation modalities we implemented are helpful for building human-robot interfaces by providing navigation autonomy and therefore allowing high level commands. The human operator does not need to continuously drive the robot with the joystick, and is therefore free to concentrate his work on selecting target locations for the robot to reach. Using 3D scene models in human-robot interfaces for commanding the robot is advantageous, comparing e.g. to teleoperation based on live-video, because of the obtained independence to the turn-around communication delays.

Interactive reconstruction is an effective method to obtain the 3D scene models as compared to conventional reconstruction methods, based on stereo or motion data. For example the model of the corridor corner, in chapter 5, was built from a single image. This constitutes a very difficult task for automatic reconstruction due to the low texture. Omnidirectional images are interesting for reconstruction as they provide very wide views of the scene which would otherwise involve many (conventional) images. Thus, the bandwidth of the communication channel to the robot may be reduced. Hence, interactive reconstruction based on omnidirectional images is a good solution for a human-robot interface, as the communication channel to the robot is typically characterised to have a very narrow bandwidth.

Concluding, when considering the whole system, (i) our vision sensing approach was found useful and convenient because it provided world-structure information for navigation, (ii) the navigation modalities fulfill the purpose of semi-autonomous navigation by providing autonomy while combining naturally with the human-robot interfaces, (iii) the human-robot interfaces provide intuitive ways to set high level tasks, by combining limited user input with the simple output of the sensor (images).

In summary, the synergetic design of the sensor, navigation and human-robot interfaces, contributes to an effective system making parsimonious use of both the sensor and the computational resources.

6.3 Directions for Future Work

The technical developments of the last few years allow the foreseeing of interesting research and development directions for Omnidirectional Vision cameras. For example micro-mirrors are a new form of obtaining variable shape mirrors and consequently to obtain variable optics adequate for distinct tasks. In addition the introduction of digital cameras is contributing not only to the enlargement of sensor resolution but also to the freedom of selecting the resolution used. This makes possible variable sized regions of interest with no compromise of resolution and advantageously directly provided by the hardware. Therefore research on how to link sensing with the tasks at hand will be even more relevant in the future.

In the Visual Path Following framework, we plan to research on automatic feature selection to complement the manual initialisation methods and increase redundancy in the feature space. This improves robustness to tracking failures or occlusions, and provides scalable alternative trajectories.

Appearance based scene representations provide a good solution for qualitative navigation. As the amount of data involved is large, an important issue is to find solutions for representing growing scenarios while maintaining the representation continuity. We plan to research on establishing new criteria for automatic control of the resolution of the appearance representations along the world scene, i.e. robot working space.

On Interactive Scene Modelling, another research direction is that of automatically estimating geometric constraints that can be used for 3D reconstruction, hence keeping the user intervention to a minimum. In terms of our future work, we plan on using large scene models obtained from the fusion of different models in order to extend the information available to the user when using the human-robot interface.

Learning methodologies are promising tools for improving robot's navigation capabilities. In the navigation experiments described in the dissertation, the robot have been instructed for new tasks using trajectories in the Bird's Eye View images or images of the topological maps, that is using the reference signals of the proposed navigation modalities. Hence, the reference signals are currently the most natural way for learning new tasks. Alternatively, we can involve once more the sensor and control design and thus augment the learning space by using new navigation modalities. As a long term research, we plan to study learning by instruction and by experience, for example by incorporating in the robot imitation and exploring behaviours.

There are many challenges for building mobile robots. Of particular importance are the navigation modules able to solve simple navigation tasks, the respective environmental representations and the visual interfaces for simple and flexible human-robot interaction.

In order to approach the challenges, in this thesis we explored several aspects and potential advantages of omnidirectional cameras over conventional cameras. We also designed navigation modalities and visual interfaces based on the sensors and on the tasks at hand, therefore considering parsimonious use of the available resources. The results

obtained provide solid reasons as to our choice of using omnidirectional vision to sense the environment, and on the approach to navigation and human-robot interface modalities.

We believe that in the future robots will be equipped with very general visual perception systems. Every new mobile robot will then cope easily with novel environments and, as it happened with computers, every person will have their *very own robot* or what we may term the *personal service robot*.

Appendix A

SVAVISCA log-polar camera

In this section we briefly revise the characteristics of the SVAVISCA log-polar sensor developed by DIST, University of Genova [64]. The log polar sensor is shown in Figure A.1.

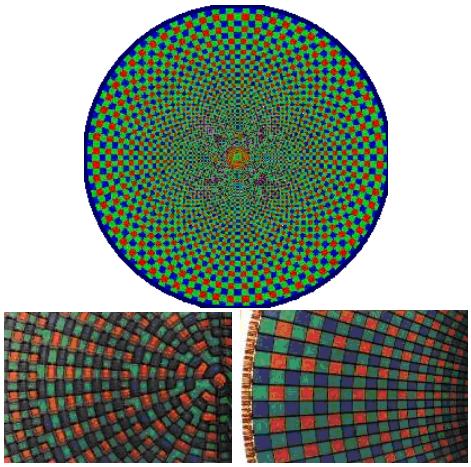


Figure A.1: General view of the SVAVISCA Log Polar Sensor (top). Detailed views of the foveal (bottom-left) and retinal (bottom-right) regions.

Inspired by the resolution of the human retina, the log-polar sensor is divided into two parts: the fovea and the retina. The fovea is the inner part of the sensor, with uniform pixel density and a radius of $\rho_0 = 0.273mm$. The retina is the outer part of the sensor, consisting of a set of concentric circular rings, with a constant number of pixels, whose resolution decays logarithmically towards the image periphery. The sensor main specifications are summarized below:

- The fovea has 42 rings. The fovea has a constant pixel-size; the distribution of pixels is such that we have 1 pixel in the first ring, 6 in the second, an then 12, 18, 24, etc, until reaching the number of 252 pixels in the 42^{nd} ring; The fovea radius is $\rho_0 = 272.73\mu m$;
- The retina has 110 rings, with 252 pixels each; The increase rate of pixel in the retina is $k = 1.02337$;

- The total number of pixels is 33.193 with 5.166 in the fovea; The minimum size of pixel is $6.8 \times 6.45 \mu m^2$; In the fovea the minimum pixel height is 6.52 μm; The diameter of sensor is $\rho_{max} = 7,135.44 \mu m$.

Appendix B

Uncertainty at pose computation

In order to analyse the sensitivity of pose computations to the noise of the measurements, we built a simulated setup where the robot and the landmark locations are fixed, but the map of the landmarks, known a priori by the robot, is perturbed with additive noise. Therefore, in our setup the observed bearings are indirectly corrupted by the additive noise present in the map.

A set of disturbances to the map motivates a set of pose estimates, whose variance finally indicates the pose computation sensitivity for each base landmark configuration.

We considered three landmark configurations:

$$map_k = \left\{ \begin{bmatrix} 0 \\ -5 \end{bmatrix}, 5 \begin{bmatrix} \cos(-90^\circ + \alpha_k) \\ \sin(-90^\circ + \alpha_k) \end{bmatrix}, 5 \begin{bmatrix} \cos(-90^\circ - \alpha_k) \\ \sin(-90^\circ - \alpha_k) \end{bmatrix} \right\}$$

where α_k is 30°, 174° or 120° respectively for $k = 1...3$.

All maps are disturbed using the same noise samples. The noise generated for the maps, η depends on the distances to the landmarks[1] and on a constant factor, α to introduce intrinsic parameters of the visual system (α=0.01):

$$\eta_{i_x}, \eta_{i_y} \sim unif(-1/2, 1/2) \cdot \left\| [x_i \ y_i]^T \right\|^2 .\alpha \quad , i = 1...3.$$

The pose is computed using the method of Betke and Gurvits [5].

Figure B.1 shows the results of pose-computation for the three examples of maps disturbed with noise. The last map, map_3, characterised to be observer centred and to have uniform angular distribution of the landmarks, motivates the smallest uncertainty region in the pose-computation. Therefore it is experimentally verified that (i) pose-computation depends on the structure of the landmarks and (ii) a uniform arrangement of the landmarks is a good choice. The distribution of the landmarks around the observer also indicates the need of an horizontal omnidirectional field of view, contrary to the one provided with conventional cameras by itself.

Given that the configuration of the landmarks affects pose-computation uncertainty, a

[1] This is not relevant for the next example, since all landmarks are at the same distance to the observer, but will for the later ones.

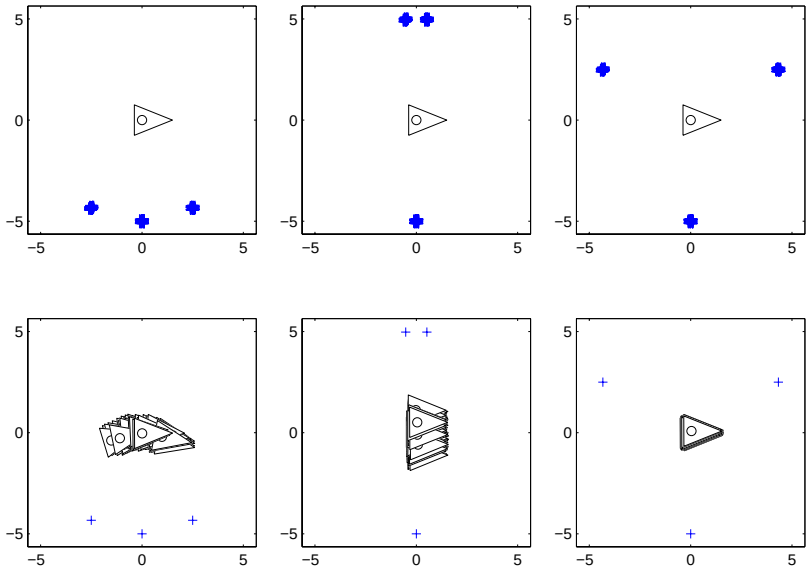


Figure B.1: Localisation uncertainty for three configurations of landmarks. The smallest uncertainty results from the equally-spaced azimuthal-angle configuration.

natural question arises whether it is possible to find an optimal configuration.

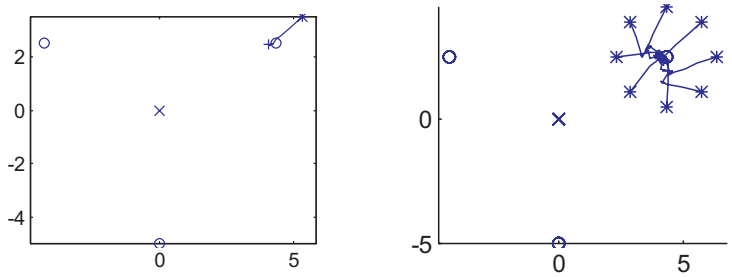
Since the observations comprise only bearings, the pose-computation problem remains the same as long as the configuration of the landmarks is rotated rigidly around the observer. Hence the uncertainty analysis of the pose computation is invariant to rotations of the configuration of the landmarks. Therefore the bearing to the first landmark can be defined arbitrary and thus one of the landmarks may be fixed.

Fixing one landmark, i.e. overcoming the invariance to rotation of the configuration, several criteria of uncertainty minimization can still be drawn. The first criterium we considered is formulated just in two degrees of freedom, namely the angles of two landmarks (LM2 and LM3) relative to a fixed one (LM1). It works on the uncertainty ellipse which we desired to be as small and close to a circle as possible. This is achieved looking for the eigenvalues of the ellipse.

The minimization criterium just described is built over two degrees of freedom, namely one for each of the two landmarks. Alternatively the two degrees of freedom may be used in a single landmark, i.e. two of the landmarks become fixed while the third one is allowed to move freely in the XY plane.

Fixed landmarks:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0 \\ -5 \end{bmatrix}, \quad \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = 5 \begin{bmatrix} \cos(-\pi/6 - \pi/2) \\ \sin(-\pi/6 - \pi/2) \end{bmatrix}.$$



(a) (c)

Figure B.2: Optimal landmark configuration. (a) Through the optimisation procedure the top-left landmark starts at the $*$ and ends at the $+$ very close to the expected o location. (b) Several starting positions of the top-left landmark yield the same resulting final location.

Performed minimization:

$$(x_3^*, y_3^*) = \underset{(x_3, y_3)}{\arg\min} \sqrt{\lambda_1^2 - \lambda_2^2} \quad , \quad \lambda_1, \lambda_2 = eigenvalues\,(R)\,.$$
$$s.t.\ |\lambda_1 - \lambda_2| = 0$$

With the possibility of varying the distance between the landmark and the observer the factor $\left\|[x_i\ y_i]^T\right\|^2$ becomes relevant. This factor directly affects the disturbances / noise in the map (see previously shown expression for the η_{ij}.).

Figure B.2 shows several map optimisation results for the case of two fixed landmarks and one free in the plane. It shows the convergence of the optimisation to the situation where the landmarks become angularly equi-spaced around the observer and at a constant distance: the third landmark goes to the circle centred in the observer with the radius given by the common distance of the other landmarks to the observer. This observation further motivates the use of omnidirectional images for navigation and localisation.

Appendix C

Set of Images for the All-corridors Reconstruction

Figure C.1 shows the base set of images used for the four corridors modelling. The images were taken at half-lengths and corners of the corridors.

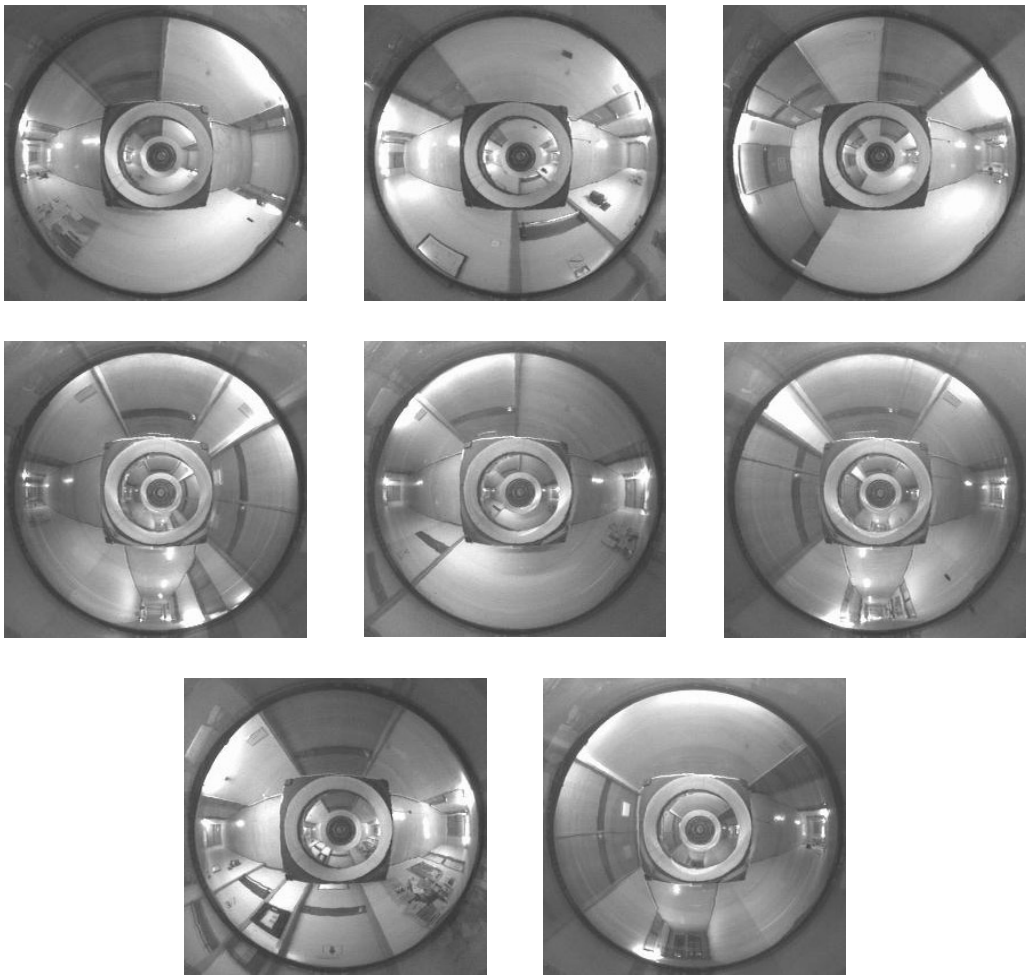


Figure C.1: Set of images used for the reconstruction of the four corridors model.

Bibliography

[1] N. Aihara, H. Iwasa, N. Yokoya, and H. Takemura, *Memory-based self-localization using omnidirectional images*, Proc. Int. Conf. Pattern Recognition, 1998, pp. 1799–1803.

[2] S. Baker and S. K. Nayar, *A theory of catadioptric image formation*, Proc. Int. Conf. Computer Vision, January 1998, pp. 35–42.

[3] ———, *A theory of single-viewpoint catadioptric image formation*, International Journal of Computer Vision **35** (1999), no. 2, 175–196.

[4] R. Basri and E. Rivlin, *Localization and homing using combinations of model views*, Artificial Intelligence **78** (1995), 327–354.

[5] M. Betke and L. Gurvits, *Mobile robot localization using landmarks*, IEEE Trans. on Robotics and Automation **13** (1997), no. 2, 251–263.

[6] J. Borenstein, H. R. Everett, and Liqiang Feng, *Navigating mobile robots: Sensors and techniques*, A. K. Peters, Ltd., Wellesley, MA, 1996
(also: Where am I? Systems and Methods for Mobile Robot Positioning, ftp.eecs.umich.edu/people/johannb/pos96rep.pdf, 1996).

[7] G. Borgefors, *Hierarchical chamfer matching: A parametric edge matching algorithm*, IEEE Transactions on Pattern Analysis and Machine Intelligence **10** (1988), no. 6, 849–865.

[8] R. A. Brooks, *A robust layered control system for a mobile robot*, IEEE Transactions on Robotics and Automation **2** (1986), 14–23.

[9] J. Canny, *A computational approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence **8** (1986), no. 6, 679–698.

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

[11] J. S. Chahl and M. V. Srinivasan, *Range estimation with a panoramic visual sensor*, J. Opt. Soc. Am. A **14** (1997), no. 9, 2144–2151.

[12] ———, *Reflective surfaces for panoramic imaging*, Applied Optics **36** (1997), no. 31, 8275–8285.

[13] T. S. Collett, E. Dillmann, A. Giger, and R. Wehner, *Visual landmarks and route following in the desert ant*, J. Comp. Physiology A **170** (1992), 435–442.

[14] T. Conroy and J. Moore, *Resolution invariant surfaces for panoramic vision systems*, IEEE ICCV'99, 1999, pp. 392–397.

[15] A. Criminisi, I. Reid, and A. Zisserman, *Single view metrology*, ICCV, 1999, pp. 434–441.

[16] Olivier Cuisenaire, *Distance transformations: Fast algorithms and applications to medical image processing*, Ph.D. thesis, U. Catholique de Louvain, October 1999.

[17] A. Davison and D. Murray, *Mobile robot localisation using active vision*, European Conference on Computer Vision (Freiburg, Germany), vol. 2, June 1998, pp. 809–825.

[18] C. Canudas de Wit, H. Khennouf, C. Samson, and O. J. Sordalen, *Chap.5: Nonlinear control design for mobile robots*, Nonlinear control for mobile robots (Yuan F. Zheng, ed.), World Scientific series in Robotics and Intelligent Systems, 1993.

[19] P. E. Debevec, C. J. Taylor, and J. Malik, *Modeling and rendering architecture from photographs: a hybrid geometry and image-based approach*, SIGGRAPH, 1996.

[20] C. Deccó, J. Gaspar, N. Winters, and J. Santos-Victor, *Omniviews mirror design and software tools*, Tech. report, Omniviews deliverable DI-3, available at http://www.isr.ist.utl.pt/labs/vislab/, September 2001.

[21] L. Delahoche, C. Pégard, B. Marhic, and P. Vasseur, *A navigation system based on an omnidirectional vision sensor*, Proc. Int. Conf. Intelligent Robotics and Systems (Grenoble, France), 1997, pp. 718–724.

[22] G. DeSouza and A. Kak, *Vision for mobile robot navigation: A survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002), no. 2, 237–267.

[23] A. Dick, P. Torr, and R. Cipolla, *Automatic 3d modelling of architecture*, BMVC (vol1), 2000, pp. 372–381.

[24] B. Espiau, F. Chaumette, and P. Rives, *A new approach to visual servoing in robotics*, IEEE Trans. on Robotics and Automation **8** (1992), no. 3, 313–326.

[25] O. Faugeras, *Three-dimensional computer vision - a geometric viewpoint*, MIT Press, 1993.

[26] C. Fermüller and Y. Aloimonos, *Ambiguity in structure from motion: Sphere versus plane*, International Journal of Computer Vision **28** (1998), no. 2, 137–154.

[27] Mark Fiala, *Panoramic computer vision*, Ph.D. thesis, University of Alberta, 2002.

[28] M. A. Fischler and R. C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Communications of ACM **24** (1981), no. 6, 381–395.

[29] S. Fleury, P. Souères, J. Laumond, and R. Chatila, *Primitives for smoothing mobile robot trajectories*, IEEE Transactions on Robotics and Automation **11** (1995), no. 3, 441–448.

[30] S. Gaechter and T. Pajdla, *Mirror design for an omnidirectional camera with a uniform cylindrical projection when using svavisca sensor*, Tech. report, Czech Tech. Univ. - Faculty of Electrical Eng. ftp://cmp.felk.cvut.cz/pub/cmp/articles/pajdla/ Gaechter-TR-2001-03.pdf, March 2001.

[31] S. Gaechter, T. Pajdla, and B. Micusik, *Mirror design for an omnidirectional camera with a space variant imager*, IEEE Workshop on Omnidirectional Vision Applied to Robotic Orientation and Nondestructive Testing, August 2001, pp. 99–105.

[32] J. Gaspar, C. Deccó, J. Okamoto Jr, and J. Santos-Victor, *Constant resolution omnidirectional cameras*, 3rd International IEEE Workshop on Omni-directional Vision at ECCV, 2002, pp. 27–34.

[33] J. Gaspar, E. Grossmann, and J. Santos-Victor, *Interactive reconstruction from an omnidirectional image*, 9th International Symposium on Intelligent Robotic Systems (SIRS'01) (Toulouse, France), July 2001.

[34] J. Gaspar and J. Santos-Victor, *Visual path following with a catadioptric panoramic camera*, Int. Symp. Intelligent Robotic Systems (Coimbra, Portugal), July 1999, pp. 139–147.

[35] J. Gaspar, N. Winters, and J. Santos-Victor, *Vision-based navigation and environmental representations with an omni-directional camera*, IEEE Transactions on Robotics and Automation **16** (2000), no. 6, 890–898.

[36] D. Gavrila and V. Philomin, *Real-time object detection for smart vehicles*, IEEE, Int. Conf. on Computer Vision (ICCV), 1999, pp. 87–93.

[37] Donald B. Gennery, *Visual tracking of known three-dimensional objects*, International Journal of Computer Vision **7** (1992), no. 3, 243–270.

[38] C. Geyer and K. Daniilidis, *A unifying theory for central panoramic systems and practical applications*, ECCV 2000 (Dublin, Ireland), June 2000, pp. 445–461.

[39] ———, *Structure and motion from uncalibrated catadioptric views*, Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition, December 2001.

[40] J. Gluckman and S. K. Nayar, *Ego-motion and omnidirectional cameras*, IEEE International Conference on Computer Vision, 1997, pp. 999–1005.

[41] E. Grossmann, D. Ortin, and J. Santos-Victor, *Algebraic aspects of reconstruction of structured scenes from one or more views*, British Machine Vision Conference, BMVC2001 (Manchester), September 2001, pp. 633–642.

[42] Etienne Grossmann, *Maximum likelihood 3d reconstruction from one or more uncalibrated views under geometric constraints*, Ph.D. thesis, Instituto Superior Técnico, Dept. Electrical Engineering, Lisbon - Portugal, 2002.

[43] G. Hager, D. J. Kriegman, A. S. Georghiades, and O. Ben-Shahar, *Toward domain-independent navigation: Dynamic vision and control*, IEEE CDC'98, Dec. 1998.

[44] R. M. Haralick and L. G. Shapiro, *Computer and robot vision (vol. 1)*, Addison-Wesley, 1992.

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

[46] E. Hecht and A. Zajac, *Optics*, Addison Wesley, 1974.

[47] A. Hicks and R. Bajcsy, *Catadioptric sensors that approximate wide-angle perspective projections*, IEEE Workshop on Omnidirectional Vision - OMNIVIS'00, June 2000, pp. 97–103.

[48] R. Andrew Hicks and Ruzena Bajcsy, *Reflective surfaces as computational sensors*, CVPR'99, Workshop on Perception for Mobile Agents, 1999.

[49] J. Hong, X. Tan, B. Pinette, R. Weiss, and E. M. Riseman, *Image-based homing*, IEEE Int. Conf. Robotics and Automation, 1991, pp. 620–625.

[50] I. Horswill, *Polly: A vision-based artifical agent*, Proc. Nat. Conf. Artifical Intelligence (Washington, DC, USA), 1993, pp. 824 – 829.

[51] S. Hutchinson, G. Hager, and P. Corke, *A tutorial on visual servo control*, IEEE Transactions on Robotics and Automation **12** (1996), no. 6, 651–670.

[52] D. Huttenlocher, G. Klanderman, and W. Rucklidge, *Comparing images using the hausdorff distance*, IEEE Transactions on Pattern Analysis and Machine Intelligence **15** (1993), no. 9, 850–863.

[53] D. Huttenlocher, R. Lilien, and C. Olsen, *View-based recognition using an eigenspace approximation to the hausdorff measure*, IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999), no. 9, 951–956.

[54] H. Ishiguro and S. Tsuji, *Image-based memory of environment*, Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 1996, pp. 634–639.

[55] S. D. Jones, C. Andersen, and J. L. Crowley, *Appearance based processes for visual navigation*, Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, J. Wiley, July 1997, pp. 551–557.

[56] Y. Kanayama and B. Hartman, *Smooth local path planning for autonomous vehicles*, Proc. Int. Conf. on Robotics and Automation, vol.3, 1989, pp. 1265–1270.

[57] S. B. Kang and R. Szeliski, *3d scene data recovery using omnidirectional multibaseline stereo*, CVPR, 1996, pp. 364–370.

[58] K. Kato, S. Tsuji, and H. Ishiguro, *Representing environment through target-guided navigation*, Proc. Int. Conf. Pattern Recognition, 1998, pp. 1794–1798.

[59] J. Košecká, *Visually guided navigation*, Proc. Int. Symp. Intelligent Robotic Systems (Lisbon Portugal), July 1996, pp. 301–308.

[60] A. Kochan, *Helpmate to ease hospital delivery and collection tasks, and assist with security*, Industrial Robot: An International Journal **24** (1997), no. 3, 226–228.

[61] D. Koller, K. Daniilidis, and H.-H. Nagel, *Model-based object tracking in monocular image sequences of road traffic scenes*, International Journal of Computer Vision **10** (1993), no. 3, 257–281.

[62] A. Kosaka and A. Kak, *Fast vision-guided mobile robot navigation using modelbased reasoning and prediction of uncertainties*, CVGIP: Image Understanding **56** (1992), no. 3, 271– 329.

[63] B. Kuipers, *Modeling spatial knowledge*, Cognitive Science **2** (1978), 129–153.

[64] LIRA Lab, *Document on specification*, Tech. report, Esprit Project n. 31951 - SVAVISCA - available at http://www.lira.dist.unige.it - SVAVISCA - GIOTTO Home Page, May 1999.

[65] J. J. Leonard and H. F. Durrant-Whyte, *Mobile robot localization by tracking geometric beacons*, IEEE Trans. on Robotics and Automation **7** (1991), no. 3, 376–382.

[66] S. Li, S. Tsuji, and A. Hayashi, *Qualitative representation of outdoor environment along route*, Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition, 1996, pp. 177–180.

[67] D. Liebowitz, A. Criminisi, and A. Zisserman, *Creating architectural models from images*, Proceedings EuroGraphics (vol.18), 1999, pp. 39–50.

[68] L. J. Lin, T. R. Hancock, and J. S. Judd, *A robust landmark-based system for vehicle location using low-bandwidth vision*, Robotics and Autonomous Systems **25** (1998), 19–32.

[69] H. C. Longuet-Higgins, *A computer algorithm for reconstructing a scene from two projections*, Nature **293** (1981), 133–135.

[70] David G. Lowe, *Robust model-based motion tracking through the integration of search and estimation*, International Journal of Computer Vision **8** (1992), no. 2, 113–122.

[71] C. Madsen and C. Andersen, *Optimal landmark selection for triangulation of robot position*, J. Robotics and Autonomous Systems **13** (1998), no. 4, 277–292.

[72] Y. Matsumoto, M. Inaba, and H. Inoue, *Visual navigation using view-sequenced route representation*, Proc. IEEE Int. Conf. Robotics and Automation, 1996, pp. 83–88.

[73] B. McBride, *Panoramic cameras time line*, www page, http://panphoto.com/TimeLine.html.

[74] H. Murase and S. K. Nayar, *Visual learning and recognition of 3d objects from appearance*, International Journal of Computer Vision **14** (1995), no. 1, 5–24.

[75] S. Nayar and V. Peri, *Folded catadioptric camera*, IEEE Int. Conf. Computer Vision and Pattern Recognition (Fort Collins, CO), June 1999, pp. 23–25.

[76] S. K. Nayar, *Catadioptric image formation*, Proc. of the DARPA Image Understanding Workshop (New Orleans, LA, USA), May 1997, pp. 1431–1437.

[77] ———, *Catadioptric omnidirectional camera*, Proc. IEEE Conf. Computer Vision and Pattern Recognition (Puerto Rico), June 1997, pp. 482–488.

[78] ———, *Omnidirectional video camera*, Proc. of the DARPA Image Understanding Workshop (New Orleans, LA, USA), May 1997.

[79] R. Nelson and J. Aloimonos, *Finding motion parameters from spherical motion fields (or the advantage of having eyes in the back of your head)*, Biological Cybernetics **58** (1988), 261–273.

[80] W. Nelson, *Continuous-curvature paths for autonomous vehicles*, Proc. Int. Conf. on Robotics and Automation, vol.3, 1989, pp. 1260–1264.

[81] S. A. Nene and S. K. Nayar, *Stereo with mirrors*, ICCV'97, 1997.

[82] S. Oh and E. Hall, *Guidance of a mobile robot using an omnidirectional vision navigation system*, Proc. of the Society of Photo-Optical Instrumentation Engineers, SPIE (1987), no. 852, 288–300.

[83] M. Ollis, H. Herman, and S. Singh, *Analysis and design of panoramic stereo using equi-angular pixel cameras*, Tech. report, Carnegie Mellon University Robotics Institute, TR CMU-RI-TR-99-04, 1999, comes from web.

[84] T. Pajdla and V. Hlavac, *Zero phase representation of panoramic images for image based localization*, 8th Inter. Conf. on Computer Analysis of Images and Patterns CAIP'99, 1999.

[85] V. Peri and S. K. Nayar, *Generation of perspective and panoramic video from omnidirectional video*, Proc. DARPA Image Understanding Workshop, 1997, pp. 243–246.

[86] D. Rees, *Panoramic television viewing system, us patent 3 505 465*, postscript file, April 1970.

[87] D. Robertson and R. Cipolla, *An interactive system for constraint-based modelling*, BMVC (vol2), 2000, pp. 536–545.

[88] W. Rucklidge, *Efficient visual recognition using the hausdorff distance*, Lecture Notes in Computer Science, vol. 1173, Springer-Verlag, 1996.

[89] J. Santos-Victor and G. Sandini, *Visual behaviors for docking*, Computer Vision and Image Understanding **67** (1997), no. 3, 223–238.

[90] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi, *Divergent stereo in autonomous navigation : From bees to robots*, Int. J. Computer Vision **14** (1995), no. 2, 159–177.

[91] J. Santos-Victor, R. Vassallo, and H. J. Schneebeli, *Topological maps for visual navigation*, Proc. Int. Conf. Computer Vision Systems, 1999, pp. 21–36.

[92] B. Schatz, S. Chameron, G. Beugnon, and T. S. Collett, *The use of path integration to guide route learning ants*, Nature **399** (1999), 769–772.

[93] J. Shi and C. Tomasi, *Good features to track*, Proc. of the IEEE Int. Conference on Computer Vision and Pattern Recognition, June 1994, pp. 593–600.

[94] M. Spetsakis and J. Aloimonos, *Structure from motion using line correspondences*, International Journal of Computer Vision **4** (1990), no. 3, 171–183.

[95] P. Sturm, *Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction*, IEEE Conference on Computer Vision and Patttern Recognition (Puerto Rico), June 1997.

[96] ———, *A method for 3d reconstruction of piecewise planar objects from single panoramic images*, 1st International IEEE Workshop on Omnidirectional Vision at CVPR, 2000, pp. 119–126.

[97] P. Sturm and S. Maybank, *A method for interactive 3d reconstruction of piecewise planar objects from single images*, British Machine Vision Conference, 1999, pp. 265–274.

[98] V. Sundareswaran, P. Bouthemy, and F. Chaumette, *Active camera self-orientation using dynamic image parameters.*, Proc. of the 3rd. European Conference on Computer Vision (Stockholm, Sweden), 1994.

[99] T. Svoboda, T. Pajdla, and V. Hlaváč, *Epipolar geometry for panoramic cameras*, Proc. European Conf. Computer Vision (Freiburg Germany), July 1998, pp. 218–231.

[100] Tomáš Svoboda, *Central panoramic cameras design, geometry, egomotion*, Ph.D. thesis, Czech Technical University, Faculty of Electrical Engineering, September 1999.

[101] R. Talluri and J. K. Aggarwal, *Chap.4.4 - position estimation techniques for an autonomous mobile robot - a review*, Handbook of pattern recognition and computer vision (C. H. Chen, L. F. Pau, and P. S. P. Wang, eds.), World Scientific Publishing company, 1993.

[102] _____, *Mobile robot self-location using model-image feature correspondence*, IEEE Transactions on Robotics and Automation **12** (1996), no. 1, 63–77.

[103] C. J. Taylor and D. J. Kriegman, *Structure and motion from line segments in multiple images*, IEEE Transactions on Pattern Analysis and Machine Intelligence **17** (1995), no. 11, 1021–1032.

[104] T. Y. Tian, C. Tomasi, and D. J. Heeger, *Comparision of approaches to egomotion computation*, IEEE Proc. Int. Conf. Pattern Recognition, 1996, pp. 315–320.

[105] G. Toomer, *Diocles on burning mirrors : the arabic translation of the lost greek original*, Springer-Verlag, 1976.

[106] E. Trucco and A. Verri, *Introductory techniques for 3-d computer vision*, Prentice Hall, 1998.

[107] M. A. Turk and A. P. Pentland, *Face recognition using eigenfaces*, Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1991, pp. 586–591.

[108] R. Wehner and S. Wehner, *Insect navigation: use of maps or ariadne's thread?*, Ethology, Ecology, Evolution **2** (1990), 27–48.

[109] S. C. Wei, Y. Yagi, and M. Yachida, *Building local floor map by use of ultrasonic and omni-directional vision sensor*, Proc. of the IEEE Int. Conf. on Robotics and Automation, 1998, pp. 2548–2553.

[110] N. Winters, J. Gaspar, A. Bernardino, and J. Santos-Victor, *Vision algorithms for omniviews cameras*, Tech. report, Omniviews deliverable DI-3, available at http://www.isr.ist.utl.pt/labs/vislab/, September 2001.

[111] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor, *Omni-directional vision for robot navigation*, 1st International IEEE Workshop on Omni-directional Vision at CVPR, 2000, pp. 21–28.

[112] N. Winters and J. Santos-Victor, *Omni-directional visual navigation*, Proc. Int. Symp. on Intelligent Robotic Systems (Coimbra - Portugal), July 1999, pp. 109–118.

[113] Niall Winters, *A holistic approach to topological navigation using omnidirectional vision*, Ph.D. thesis, University of Dublin, Trinity College, 2001.

[114] P. Wunsch and G. Hirzinger, *Real-time visual tracking of 3-d objects with dynamic handling of occlusion*, IEEE Int. Conf. on Robotics and Automation, April 1997, pp. 2868–2873.

[115] Y. Yagi and S. Kawato, *Panoramic scene analysis with conic projection*, IEEE Int. Conf. on Robotics and Automation, 1990.

[116] Y. Yagi, W. Nishii, K. Yamazawa, and M. Yachida, *Stabilization for mobile robot by using omnidirectional optical flow*, Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, November 1996, pp. 618–625.

[117] Y. Yagi, Y. Nishizawa, and M. Yachida, *Map-based navigation for mobile robot with omnidirectional image sensor COPIS*, IEEE Trans. Robotics and Automation **11** (1995), no. 5, 634–648.

[118] K. Yamazawa, Y. Yagi, and M. Yachida, *Omnidirectional imaging with hyperbolic projection*, IEEE Int. Conf. on Robotics and Automation, 1993.

[119] ———, *Obstacle detection with omnidirectional image sensor hyperomni vision*, IEEE ICRA, 1995, pp. 1062–1067.

[120] Z. Zhang and O.D. Faugeras, *Building a 3D world model with a mobile robot: 3D line segment representation and integration"*, Proc. Int. Conf. Pattern Recognition, 1990, pp. 38–42.

[121] J. Zheng and S. Tsuji, *Panoramic representation for route recognition by a mobile robot*, International Journal of Computer Vision **9** (1992), no. 1, 55–76.