FONDO SOCIALE EUROPEO
Programma Operativo Nazionale 2000/2006
"Ricerca Scientifica, Sviluppo Tecnologico, Alta Formazione"
Regioni dell'Obiettivo 1 – Misura III.4
"Formazione superiore ed universitaria"

Facoltà di Ingegneria
Dipartimento di Ingegneria dell'Informazione e Ingegneria Elettrica

Ecole Doctorale
Informatique et Information pour la Société
EDA 335

Dottorato di Ricerca in Ingegneria dell'Informazione
IV Ciclo – Nuova Serie

TESI DI DOTTORATO

Real-time Video Analysis from a Mobile Platform: Moving Object and Obstacle Detection

CANDIDATO: **ALESSANDRO LIMONGIELLO**

COORDINATORE: **PROF. MAURIZIO LONGO**

TUTORS: **PROF. MARIO VENTO**
PROF. JEAN-MICHEL JOLION

Anno Accademico 2005 – 2006

Al mondo che soffre,
ma non vuole amare ...

Table of Contents

Sommario

"E continuamente penso alla vita nel suo essere profondo;
vago di arcani mondi e lontane genti, inquieto. ..."
A.L.

Negli ultimi anni, la comunità scientifica mondiale ha mostrato un interesse crescente nella navigazione autonoma di piattaforme mobili. Esistono, infatti, innumerevoli contesti in cui la navigazione autonoma è necessaria per l'esecuzione di differenti compiti. Tali sistemi sono comunemente noti con il nome di *Automated Guided Vehicles* (AVG) e *Autonomous Mobile Robot* (AMR). La navigazione autonoma di una piattaforma mobile è un argomento vasto, che abbraccia differenti tecnologie e discipline. Indipendentemente dal campo di applicazione, è necessaria una percezione dell'ambiente in cui la piattaforma mobile opera, in modo da garantire la navigazione. L'interazione tra il robot e l'ambiente è, inoltre, da rapportare allo specifico compito da eseguire.

Esistono diversi metodi che consentono di acquisire informazioni sull'ambiente, una buona strategia è di utilizzare diversi tipi di sensori, quindi integrare i diversi dati prima di prendere delle decisioni. La visione, in particolare, è l'unico strumento non invasivo per percepire l'ambiente circostante. Per questo motivo, anche se un laser potrebbe essere più efficace di una telecamera, in alcuni contesti potrebbero non essere tollerate interferenze e quindi essere richiesti necessariamente approcci basati sulla visione. Inoltre, la visione, fornisce una quantità di informazioni maggiore; per esempio, un laser può individuare solo la posizione di un ostacolo, mentre la visione può

determinare la sagoma dell'oggetto, riconoscerlo, e poi inseguirlo nel tempo.

In questa tesi presentiamo un sistema di visione per la navigazione autonoma di una piattaforma mobile. Il sistema è in grado di interagire con lo spazio immediatamente circostante, riconoscendo gli ostacoli e gli oggetti in movimento, costruendo una visione stabile del mondo esterno. Un sistema di visione per la navigazione autonoma, infatti, deve essere in grado di identificare gli oggetti dell'ambiente e classificarli come "ostacoli", in modo da evitarli, oppure come oggetti "target", così da poterli inseguire.

Ci occupiamo del complesso problema chiamato "Obstacle detection and avoidance". Consiste nell'individuazione degli ostacoli in modo da trovare un cammino sicuro da seguire durante la navigazione autonoma della piattaforma mobile. Un ostacolo può essere un oggetto in movimento o un oggetto immobile, potrebbe appartenere ad un preciso insieme di oggetti (per esempio altri robot o veicoli, persone), o potrebbe essere un oggetto generico e inatteso; quindi, un sistema per l'obstacle detection dipende fortemente da cosa si intende per ostacolo.

Affrontiamo il problema dell'obstacle detection nel senso più generale, cioè nel caso di *ambiente non strutturato*. L'obiettivo è complicato dall'assenza di informazioni sull'ambiente e sugli oggetti presenti nella scena. Inoltre, il movimento della telecamera, solidale al robot, rende l'analisi video molto difficile e fa fallire la maggior parte degli algoritmi presenti in letteratura. Infine, l'elaborazione deve

essere real-time per poter guidare velocemente la piattaforma mobile lungo un percorso sicuro.

L'intero sistema verrà descritto in accordo ad una precisa metodologia per lo sviluppo di un sistema di visione, il così detto *systemic approach*. In questo modo, il sistema presenterà precise specifiche di utilizzo e garantirà buone prestazioni in uno specifico dominio di applicazione. L'intero sistema può essere suddiviso nei tre compiti fondamentali: adeguata *rappresentazione dell'ambiente*; analisi dell'immagine per l'*individuazione degli ostacoli*; interpretazione dell'immagine per il *superamento degli ostacoli* e l'*analisi comportamentale*.

Il contributo maggiore del presente lavoro riguarda una "percettiva" rappresentazione dell'ambiente, cioè non una rappresentazione "passiva", ma rapportata all'obiettivo finale della navigazione autonoma. Tale rappresentazione è basata sul paradigma della stereo vision e permette di individuare nella scena gli ostacoli e gli oggetti in movimento proprio in relazione alla navigazione, infatti il risultato che perseguiamo ha una "risoluzione adeguata" ai nostri obiettivi. Definiamo, quindi, un sistema scalabile che opera a partire da una risoluzione richiesta in uno specifico contesto.

Molti autori hanno espresso il convincimento che un sistema di visione robotica dovrebbe essere in grado di riprodurre il sistema di visione umano, quindi dovrebbe essere basato sulla visione stereoscopica. Il vantaggio maggiore della visione stereoscopica rispetto ad altre tecniche (per esempio optical flow, o tecniche basate su modelli) è che viene prodotta una descrizione completa della scena,

possono essere individuati sia ostacoli fermi che in movimento (senza definire un modello complesso per l'ostacolo), e tale tecnica è meno sensibile ai cambiamenti ambientali (l'inconveniente maggiore delle tecniche basate sull'optical-flow). La stereo vision fornisce una rappresentazione 3D (o almeno una approssimazione 2D ½) della scena. Una coppia di immagini acquisite da una telecamera stereo contiene implicitamente informazioni di profondità della scena: questa è la principale assunzione della stereo vision. La difficoltà più grande sta nello stabilire una corrispondenza nelle due immagini tra i punti rappresentanti lo stesso punto della scena; questo processo è chiamato *disparity matching*. In letteratura, tutti gli approcci sono basati su questa corrispondenza puntuale.

Noi proponiamo una estensione di tale concetto, più precisamente definiamo un valore di disparità per un'intera regione della scena a partire dalle due viste omologhe della regione nella coppia stereo. La ragione principale per tale estensione è che un approccio basato sulla corrispondenza puntuale è ridondante in applicazioni AMR e AVG. In questo contesto, infatti, non è molto importante avere una buona ricostruzione delle superfici, ma è più importante identificare adeguatamente lo spazio occupato da ogni oggetto nella scena, anche assegnandogli un'unica informazione di disparità. Inoltre gli approcci basati sul pixel sono poco robusti in alcuni contesti reali, specialmente nel caso di filmati acquisiti da una piattaforma mobile. Il nostro metodo fornisce la profondità media di una intera regione facendo un calcolo integrale sulla regione stessa, così da avere minori problemi in aree uniformi rispetto ad altri metodi. La stima della posizione delle

regioni risulta sufficientemente accurata per la navigazione e il sistema è sufficientemente veloce per applicazioni real-time.

I risultati del nostro metodo di corrispondenza stereo sono stati confrontati con i migliori algoritmi in letteratura. Tali algoritmi sono, tradizionalmente, testati su database standard composti da immagini statiche (non provenienti da telecamere in movimento), ben calibrate e acquisite in condizioni controllate di illuminazione. In questa tesi riportiamo alcuni risultati ottenuti su filmati più realistici acquisiti dalla nostra piattaforma mobile, in modo da sottolineare i limiti degli algoritmi pixel-based presenti ad oggi in letteratura. E' stata proposta, anche, una metrica quantitativa di confronto sperimentale con riferimento allo specifico obiettivo dell'obstacle detection nel contesto della navigazione autonoma.

L'organizzazione complessiva della tesi è descritta di seguito. Nel capitolo 1 viene illustrata l'importanza applicativa e scientifica dei sistemi di analisi video in tempo reale per la navigazione autonoma di piattaforme mobili, mostrando i vantaggi delle tecniche vision-based rispetto ad altre alternative. Il capitolo 2 è dedicato allo stato dell'arte e ad una possibile classificazione dei diversi approcci di visione robotica. Dopo una breve rassegna sulle metodologie di visione computazionale, nel capitolo 3 è presentata l'architettura del nostro sistema. Il capitolo 4 contiene un survey sulla stereo vision e presenta alcuni dei più importanti algoritmi presentati in letteratura. Il capitolo 5 è dedicato al nostro approccio: dopo il nostro metodo innovativo per la rappresentazione dell'ambiente, presentiamo il sottosistema per la detection degli oggetti in movimento e degli ostacoli. In fine, nel

capitolo 6 è presente una discussione dei risultati sperimentali su un database stereo standard e sulla nostra sequenza video. Le conclusioni della tesi sono riportate nel capitolo 7.

Summary

"Continuously thinking about the Life and its Deepness;
wandering thoughts of arcane worlds and far people, restless. ..."
A.L.

During the last years, the World Scientific community has shown an increasing interest in autonomous navigation of mobile platforms. The reason is that a lot of contexts need an autonomous mobile platform for different aims. These systems are usually known as *Automated Guided Vehicles* (AVG) and *Autonomous Mobile Robot* (AMR) systems. The autonomous navigation of a mobile platform is a broad topic, covering a large spectrum of different technologies and disciplines. Independently from the several applying fields, we need a perception of the environment in which the mobile platform moves, in order to guarantee an autonomous navigation. The interaction between the mobile robot and the environment is, then, related to the particular goal.

Several methods can be used by a robot to acquire information on the environment in which it is moving, a good strategy is to use different kind of sensors and then integrate the different data before deciding what to do. Vision is the only way that makes a non invasive perception of external environment possible. For this reason, even if a laser could perform better than cameras, in given problem we can not always tolerate signals interferences and are so obliged to use the vision approach. Moreover, Vision can provide a much larger set of information; for example, a laser can only locate an obstacle, vision

can identify the shape of the object, recognize it, and then follow it in the time.

In this thesis we present a vision system for autonomous navigation of a mobile platform. The system is enable to interact with its immediate surroundings, recognizing obstacles and other moving objects, and obtaining a stable view of the world. In fact, a vision system for autonomous navigation has to detect the objects in the environment and classify them as “obstacles”, in order to avoid them, or as “target” objects, in order to follow them.

We face the challenging problem of the “Obstacle detection and avoidance”. It consists of obstacle detection in order to find the safety path to follow during the autonomous navigation of a mobile platform. An obstacle could be a moving or a motionless object, it could belong to a precise set of objects (for example other robots or vehicles, people), or it could be a generic and unexpected object. Afterwards, a system for obstacle detection depends on what an obstacle means.

We face the obstacle detection problem in the most general framework, in fact we consider an *unstructured environment*. This task is very hard to solve, we do not have a large knowledge of the environment and of the objects in the scene. Moreover, the motion of the camera, mounted on the robot, makes the video analysis very difficult and the most algorithms, in the literature, fail. Finally, an autonomous navigation needs a real-time elaboration to guide quickly the mobile platform through the safety path.

The entire system will be described according to a precise methodology for vision system development, called *systemic*

approach. In this way, our system will have some using specifications and will guarantee a good performance for the specified application domain. We can divide the whole system into three most significant tasks: a suitable *representation of the environment*; image analysis in order to *detect obstacles*; image understanding for *obstacle avoidance* and *behavioral analysis*.

The major contribution of this work concerns a "perceptive" representation of the environment, that it is not a "passive" representation, but related to the final goal of autonomous navigation. It is based on the stereo vision paradigm and detect obstacles and moving objects in the scene right according to the autonomous navigation goal, that is obtaining a result as fine as it is enough for our aims. Therefore, we define a scalable system that works with a required resolution in a specific context.

Many authors have expressed their conviction that a robotic vision system should aim at reproducing the human vision system, and so should be based on stereo vision. The greatest advantage of stereo vision with respect to other techniques (e.g. optical flow, or model-based) is that it produces a full description of the scene, can detect motionless and moving obstacles (without defining a complex obstacle model), and is less sensitive to the environmental changes (the major disadvantage of optical-flow techniques). The stereo vision provides a 3D representation (or at least an approximation like a 2D ½ representation) of the scene. A pair of images acquired from a stereo camera implicitly contains depth information about the scene: this is the main assumption of stereo vision. The main difficulty is to

establish a correspondence between points of the two images representing the same point of the scene; this process is called *disparity matching*. All the approaches, in the literature, are based on this pixel correspondence.

We propose an extension of that concept, namely we define a disparity value for a whole region of the scene starting from the two homologous views of it in the stereo pair. The main reason of this extension is that a pixel-matching approach is redundant for AMR and AVG applications. In fact, in this framework, it is not very important to have a good reconstruction of the surfaces, but it is more important to identify adequately the space occupied by each object in the scene, even by just assigning to it a single disparity information. Moreover the pixel-based approaches are lacking in robustness in some realistic frameworks, especially for video acquired from a mobile platform. Our method estimates the average depth of the whole region by an integral measure, and so has fewer problems with uniform regions than other methods have. The estimate of the position of the regions is sufficiently accurate for navigation and it is fast enough for real time processing.

The results of our method for stereo matching are shown in a comparison with the best algorithms in the literature. The tests for stereo matching algorithms are usually performed with standard databases composed of static images (i.e. acquired from a static camera), well-calibrated and acquired in uniform lighting. We report some results obtained on a more realistic video acquired from our mobile platform, in order to underline the limits of the algorithms

present by now in the literature that are pixel-based. It is also proposed a quantitative measurement for performance evaluation, with a reference to our specific goal of the obstacle detection in autonomous navigation framework.

The plan of the thesis is described in the following. In Chapter 1 the significance of real-time video analysis systems for autonomous navigation of mobile platforms is shown and compared with other alternatives techniques. Chapter 2 is devoted to the state of art of the robotic vision and a possible classification of the different approaches has been defined. After a brief survey on computer vision methodologies, in the Chapter 3 our system architecture is shown. Chapter 4 presents a survey on Stereo Vision and some of the most relevant algorithms in the literature are shown. Chapter 5 is devoted to our approach. After our novel method for the representation of the environment, we present the subsystem for Moving Object and Obstacle Detection. Finally, in Chapter 6 there is a discussion of experimental results on standard stereo database and also on our stereo video sequence. Conclusions are drawn in Chapter 7.

Résumé

"Et tout le temps je pense à la vie dans son être profond;
je vague de mondes mystérieux et de gens lointaines, inquiet. ..."
A.L.

Dans les dernières années, la communauté scientifique mondiale a montré un intérêt croissant dans la navigation autonome de plateformes mobiles. En effet ils existent de nombreux contextes dans lesquels la navigation autonome est nécessaire pour l'exécution de différentes tâches. Des tels systèmes sont communément connus sous le nom *Automated Guided Vehicles* (AVG) et *Autonomous Mobile Robot* (AMR). La navigation autonome d'une plateforme mobile est un vaste sujet, qui concerne différentes technologies et disciplines. A part le domaine d'application, une perception de l'espace dans lequel la plateforme mobile agit est nécessaire, de façon à garantir la navigation. L'interaction entre le robot et l'espace environnant, en outre, dépend de la tâche spécifique.

Il existe différentes méthodes qui permettent d'acquérir des informations sur l'espace environnant. Une bonne stratégie est celle d'utiliser différents types de capteurs et d'intégrer différentes données avant de prendre des décisions. La vision, en particulier, c'est le moyen unique qui n'est pas envahissant pour percevoir l'espace environnant. Pour cette raison, même si un laser pourrait être plus efficace qu'une caméra, dans quelques contextes où des interférences ne sont pas tolérées, il faut nécessairement utiliser des approches basées sur la vision. En outre, la vision, fournit une quantité

d'informations majeure; par exemple un laser peut déterminer seulement la position d'un obstacle, alors que la vision peut déterminer la silhouette de l'objet, le reconnaître, et ensuite le poursuivre dans le temps.

Dans cette thèse nous présentons un système de vision pour la navigation autonome d'une plateforme mobile. Le système est en mesure d'interagir avec l'espace immédiatement environnant, en reconnaissant les obstacles et les objets en mouvement et en construisant une vision stable du monde extérieur. Un système de vision pour la navigation autonome, en effet, doit être en mesure d'identifier des objets, dans l'espace contrôlé, et les classifier comme « obstacles », de façon à les éviter, ou bien comme des objets « target », ainsi qu'il puisse les poursuivre.

Nous nous occupons du problème appelé « Obstacle detection and avoidance ». Il s'agit de la détermination des obstacles pour chercher un chemin sûr à suivre pendant la navigation autonome de la plateforme mobile. Un obstacle peut être un objet en mouvement ou un objet immobile, ou pourrait appartenir à un ensemble précis d'objets (par exemple d'autres robots ou véhicules, ou alors des personnes), ou bien il pourrait être un objet générique et inattendu ; donc, un système pour la détection d'obstacles dépend fortement de ce qu'on entend pour obstacle.

Nous affrontons le problème de la détection d'obstacles dans le sens plus général, c'est-à-dire dans le case d'*espaces non structurés*. L'objectif est compliqué pour l'absence d'informations sur l'espace environnant et sur les objets présents dans la scène. En outre, le

mouvement de la caméra, solidaire au robot, rend l'analyse très difficile et la plupart des algorithmes présents en littérature échouent dans la tâche. Finalement, l'élaboration doit être en temps réel pour pouvoir conduire rapidement la plateforme mobile le long d'un parcours sûre.

Le système sera décrit en accord à une méthodologie précise (appelé «systemic approach ») pour le développement d'un système de vision. De cette manière, le système présentera des précis détails d'utilisation et il garantira des bonnes prestations dans un domaine d'application spécifique. Le système est partagé dans trois composants fondamentales : un composant pour la *représentation de l'espace environnant* ; un système d'analyse des images pour la *détection des obstacles* ; un composants pour l'interprétation des images pour le *dépassement des obstacles* et pour l'*analyse comportemental*.

La contribution majeure de ce travail concerne la représentation « perceptive » de l'espace, c'est-à-dire une représentation qui n'est pas « passive » mais qui est comparé à l'objectif final de la navigation autonome. Telle représentation est basée sur le paradigme de la « stereo vision » et elle permet de déterminer dans la scène les obstacles et les objets en mouvement par rapport à la navigation. En effet les résultats que nous poursuivons sont adaptés aux objectifs. Nous définissons, donc, un système scalable qui agit à partir d'une solution demandée dans un spécifique contexte.

Beaucoup d'auteurs ont exprimé la conviction qu'un système de vision robotique devrait être en mesure de reproduire le système de vision humaine, donc devrait être basé sur la vision stéréoscopique.

L'avantage majeur de la vision stéréoscopique par rapport aux autres techniques (par exemple « optical flow », ou des techniques basées sur des modèles), est qu'elle produit une description complète de la scène ; telle technique est moins sensible aux changements de l'espace environnant (l'inconvénient majeur des techniques basées sur l' « optical-flow »). La « stereo vision » fournit une représentation 3D (ou au moins une approximation 2D ½) de la scène. Un couple d'images acquises d'une caméra stéréo contient implicitement toutes les informations de profondeur de la scène : celle-ci est la thèse principale de la vision stéréo. La difficulté majeure réside en l'établissement d'une correspondance, dans les deux images, entre les points représentants le même point de la scène ; ce procédé est appelé « disparity matching ». Dans la littérature, toutes les approches sont basées sur cette correspondance ponctuelle.

Nous proposons une étendue de tel concept. Plus précisément nous définissons une valeur de disparité pour une région de la scène à partir des deux vues homologues de la région dans le couple d'images stéréo. La raison principale pour telle étendue est qu'une approche basée sur la correspondance ponctuelle est redondante en applications AMR et AVG. Dans ce contexte, en effet, ce n'est pas très important d'avoir une bonne reconstruction des surfaces, mais c'est plus important d'identifier adéquatement l'espace occupé de chaque objet dans la scène, même en lui assignant une unique information de disparité. En outre les approches basées sur le pixel sont peu robustes dans des contextes réels, spécialement dans le case de séquence vidéo acquises d'une plateforme mobile. Notre méthode fournit la

profondeur moyenne d'une région entière en faisant un calcule intégral sur la région même, de façon à avoir des mineurs problèmes dans les surfaces uniformes par rapport aux autres méthodes. L'estimation de la position des régions résulte suffisamment soignée pour la navigation et le système est suffisamment rapide pour les applications en temps réel.

Les résultats de notre méthode de correspondance stéréo ont été confrontés avec les meilleures algorithmes de la littérature. Ces algorithmes sont, traditionnellement, testée sur des bases de donnée standards composés d'images statiques (acquises par une caméra qui n'était pas en mouvement), bien calibrées et avec des conditions d'éclairage contrôlées. Dans cette thèse nous montrons quelques résultats obtenus sur des séquences vidéo plus réalistes acquises par la notre plateforme mobile, de façon à souligner les limites des algorithmes basées sur les pixels présentes aujourd'hui dans la littérature scientifique. Une métrique quantitative de comparaison expérimentale a été aussi proposée. Cette métrique fait référence au spécifique objectif de la détection d'obstacles dans le contexte de la navigation autonome.

L'organisation globale de la thèse est ici décrite. Dans le premier chapitre est illustrée l'importance applicative et scientifique des systèmes d'analyse vidéo en temps réel pour la navigation autonome de plateformes mobiles, en montrant les avantages des techniques basée sur la vision par rapport aux autres alternatives. Le deuxième chapitre est dédié à l'état de l'art et à un classement des différentes approches de la vision robotique. Après une revue sur les

méthodologies de vision, dans le chapitre 3 est présentée l'architecture du système. Le quatrième chapitre contient une revue sur la vision stéréo et présente quelqu'un des plus importantes algorithmes de la littérature scientifique. Le chapitre 5 présente notre approche : nous présentons une méthode innovatrice pour la représentation de l'espace e un sous-système pour la détection des objets en mouvement et des obstacles. Finalement, dans le sixième chapitre est présent une discussion des résultats expérimentaux sur des bases de données stéréo standard et sur le nôtres séquences vidéo. Les conclusions de la thèse sont rapportées dans le chapitre 7.

Chapter 1

Autonomous navigation of mobile platforms: a visual approach

"... Mislaid, lost in a black moonless night, I fall into
A painful despondency of being a man. ..."
"... Smarrito, perduto in una notte buia, senza luna, cado
nella disperazione angosciosa di essere uomo. ..."
A.L.

1.1 Introduction

During the last years, the World Scientific community has shown an increasing interest in autonomous navigation of mobile platforms. The reason is that a lot of contexts need an autonomous mobile platform for different aims. For example, a mobile robot can be engaged in general fields, for mining, cleaning, maintenance and supervision aims; or for searching operations in hard environment, as collapsed or radiation areas, space or underwater explorations. In the industrial areas a mobile platform is used for material transportation, as like intelligent driver helper systems have been developed in open roads to increase security. Finally, automatic vehicles have been used in numerous military operations. These systems are usually known as *Automated Guided Vehicles* (AVG) [1,2] and *Autonomous Mobile Robot* (AMR) [3] systems. The autonomous navigation of a mobile

platform is a broad topic, covering a large spectrum of different technologies and disciplines, as robotics, A.I., informatics, computer architecture, telecommunications, control theory and automation, etc. It draws on some very ancient techniques, as well as some of the most advanced space science and engineering. Independently from the several applying fields, we need a perception of the environment in which the mobile platform moves, in order to guarantee an autonomous navigation.

1.2 Autonomous navigation systems nowadays

In 1997 the robot *Sojourner Rover* [4] made its first "semi-autonomous" step over the Mars ground. The mobile robot Sojourner Rover was used during the *Pathfinder mission (*formerly known as the Mars Environmental Survey, or MESUR, Pathfinder). It was a six-wheeled vehicle which was controlled by an Earth-based operator. The communication time delay was between 6 and 41 minutes depending on the relative position of Earth and Mars, requiring some *autonomous* control. The on-board control system was capable of compressing and storing a single image on-board. The rover was equipped with a black and white imaging system which was used to image the surrounding terrain to study size and distribution of soils and rocks, as well as locations of larger features.

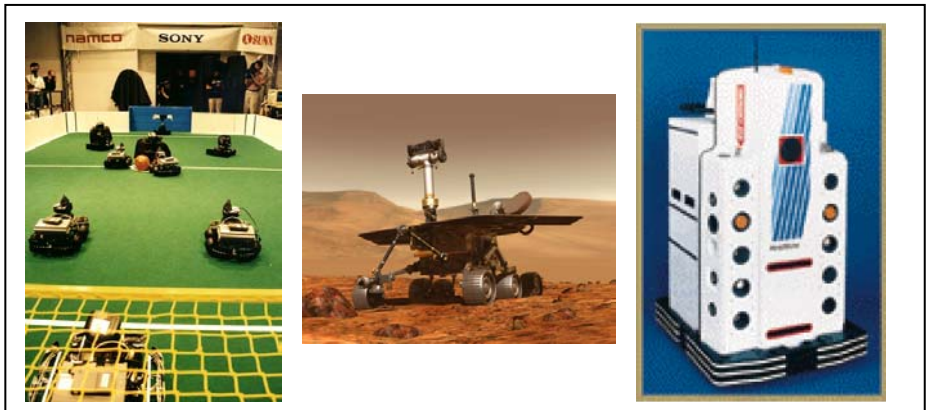


Figure 1: Some examples of autonomous mobile robot: on the left side, some Robots playing football during the Robocup Competition; on the middle side, the Sojourner rover; on the right side, the HelpMate mobile robot.

Since then a lot of robots, which gained information about the extern environment with the help of different type of sensors, have been produced. For example, Robocup [5] is an International Competition of robots playing football. In this case, a robot has a lot of sensors to understand the environment and the location of the other robots and some cooperation mechanisms to plan the game. HELPMATE [6] is a mobile robot used in hospitals for transportation tasks. It has various on board sensors for autonomous navigation in the corridors. The main sensor for localization is a camera looking to the ceiling. It can detect the lamps on the ceiling as reference (landmark).

An example of industrial mobile platform is the newest generation of *Automated Guided Vehicles* (AGV) by VOLVO [7] used to transport motor blocks from on assembly station to an other. It is guided by an electrical wire installed in the floor but it is also able to leave the wire to avoid obstacles. There are over 4000 AGV only at

VOLVO's plants. An other example of Automated Guided Vehicles is the DARPA Challenge [8]. It is a cross country race of autonomous vehicles across the California and Nevada desert that the Defense Advanced Research Projects Agency (DARPA) made to any non-governmental group interested in competing for the million dollar prize.

Figure 2: Some examples of Automated Guided Vehicles: on the left side, a vehicle during the D.A.R.P.A. Challenge ; on the right side, newest generation of Automated Guided Vehicle by VOLVO.

1.3 Why this research is still of interest

About twenty years ago, the growing of interest towards Intelligent Transport Systems (ITS) led to the birth of different governmental foundations, in United States, with the goal of exploiting the opportunities given by this field of study. The military, mainly through the Defense Advanced Research Projects Agency (DARPA) and the research offices of the Navy, Army and Air Force (ONR, ARO, AFOSR), is a major supporter of basic and applied research in robotics. The military funds work in all the major subject areas of robotics (manipulation, vision, planning, locomotion, sensing and

computing) by heavily supporting robotics in industry and university research centers (such as Carnegie-Mellon University, MIT and Stanford). In 1982, the National Research Council Commission on Engineering and Technical Systems produced a report [9] for the United States Army, in order to underline how reduce risk and improve effectiveness in applications of Robotics and Artificial Intelligence. In Europe the PROMETHEUS (PROgraM for a European Traffic with Highest Efficiency and Unprecedented Safety) program was born in 1986. The government in the U.S. founded NAHSC (National Automated Highway System Consortium) in 1995.

Nowadays we are living the ITS second generation characterized by a mature technology. Different approaches were used to face the problem: robotics, A.I., informatics, computer architecture, telecommunications, control theory and automation, etc. Researchers, today, are trying to improve the intelligence in the vehicle (or robot) rather than in the infrastructures. For example, a highway equipped so that modified vehicles can move in an autonomous way would be very expensive. For this reason the research is oriented towards the improvement of the sensors on the vehicles. A more and more detailed understanding of the environment from sensory data is the main issue to have an autonomous navigation. Sometimes an information fusion is the main topic of the research, other times a new method to analyze the sensory information is defined. Anyway, the autonomous navigation is a so hard and useful objective that its interest can not come to an end.

1.4 Computer Vision for autonomous navigation

Several methods can be used by a robot to acquire information on the environment in which it is moving, a good strategy is to use different kind of sensors and then integrate the different data before deciding what to do.

Vision is the only way that makes a non invasive perception of external environment possible. A camera is a so-called passive sensor, and through it we can get information without "polluting" the environment with signals, this is what happens when we use infrared or lasers (active sensor). For this reason, even if a laser could perform better than cameras, in given problem we can not always tolerate signals interferences and are so obliged to use the vision approach. Such a problem could be faced, for example, every time there is more than one robot on the scene. Moreover, Vision can provide a much larger set of information than passive sensors [10,11]. For example, a passive sensor can only locate an obstacle, vision can identify the shape of the object, recognize it, and then follow it in the time. A vision system is closer to the human vision system (HVS), so the first result is that the system we are going to produce will fail at least in the same circumstances in which the HVS fails (rain, fog, etc.). Furthermore video signals are well known and very general. In fact such systems can use all the well known techniques in image and video processing. As consequence a video-based navigation system is

more general than an active-sensory system. An architecture for a video navigation system can be used in a wide range of situations and environments. On the other hand, only in recent years the computing power needed for real-time video processing has become sufficiently available and affordable for dealing with this kind of applications. As already said, the solution is not "vision or not vision", but a mixture of different sensors, and the visual sensor has become more and more suitable for all the mentioned reasons.

In the last years, a wide discussion about computer vision methodologies [12,13,14,15,16] has been faced. In general, there are many ways to manage the development of a computer vision architecture. Vision is an under constrained problem and the used methodologies are different depending on the different sources of constraints they consider and on the different goals to achieve. A brief survey on computer vision methodologies will be presented in the chapter 3. Right now, we just like to introduce the main dichotomy between "passive" and "active" vision. The passive vision, firstly defined by Marr [13], suggests to investigate the visual information in order to obtain a reconstruction of the environment, then use it in several recognition and understanding aims. The visual information is the unique input of a visual system. Aloimonos *et al.* [14], instead, consider that observer (the robot, for example) actively affects the visual system. The perception of the environment can not leave the observer out of consideration. As it will be clear afterwards, we design a visual system according to the so-called *systemic approach* [12], in which even the goal is an integrate part of the understanding process.

The video analysis is faced thinking about the nature of the environment, the nature of the mobile robot and the kind of goal we suppose to reach. In this way an ill-posed problem becomes tractable.

1.5 Open problems in visual autonomous navigation

If the goal is to move a mobile robot from one coordinate location to another coordinate location, we believe there is sufficient accumulated expertise in the research community today to design a mobile robot that could do that in a typical building. A vehicle can follow a line on the ground if a well structured environment is available. But, if the goal is to carry out behavior-based navigation — an example being to find a particular object on the scene, a person in a searching operation, or an interesting object in a planetarium exploration — we are still far away. Useful navigation where a robot must be aware of the meaning of the objects encountered in the environment is beset with a harder-to-solve version of the central problem of general computer vision — automatic scene interpretation. On the other side, an intelligent vehicle should be able to move in several kind of environment, and not only in structured one. For this reason, scene interpretation for mobile platforms is a harder problem than for stationary platforms because, in the mobile context, there is much less control over illumination and background scene clutter. Progress will also surely be made in more efficient ways of representing the metrical and topological properties of the environment and, on the other, in more efficient ways of

representing the uncertainties in a robot's knowledge of its environment and its own position relative to the environment. These uncertainties have to take into account in the visual system, in fact the best autonomous platform is the one which has awareness of its limits and stops to be sure (or asks human operator an help).

The results acquired today in intelligent vehicles are a hope for tomorrow; other problems above all legal ones are arising today: who is responsible in an intelligent vehicle accident? Nowadays, technology is not so mature to have intelligent autonomous cars moving on our highways, in the next future, however, we could see autonomous vehicle used in industrial environments (indoor environments).

References

[1] M. Bertozzi, A. Broggi and A. Fascicoli, "Vision-based intelligent vehicles: State of art and perspectives", Robotics and Autonomous Systems, vol. 32, pp. 1-16, 2000.

[2] V. Kastrinaki, M. Zervakis and K. Kalaitzakis, "A survey of video processing techniques for traffic applications", Image and Vision Computing, vol. 21, pp. 359–381, 2003.

[3] G. N. DeSouza and A. C. Kak, "Vision for Mobile Robot Navigation: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pp. 237-267, 2002.

[4] Path Finder Rover, Jet Propulsion Laboratory, California Institute of Technology, National Aeronautics and Space Administration, Pasadena. Link: http://www.nasa.gov/

[5] M. Veloso, E. Pagello and H. Kitano (Eds.), "Robocup-99: Robot Soccer World Cup III", Lecture Notes in Computer Science, vol. 1856, pp. 802, 2000.

[6] http://www.ntplx.net/~helpmate/

[7] http://www.danahermotion.com/industry_solutions/agv/

[8] http://www.darpa.mil/grandchallenge/index.asp

[9] National Research Council Commission on Engineering and Technical Systems, "Applications of Robotics and Artificial Intelligence to Reduce Risk and Improve Effectiveness, A Study for the United States Army", Manufacturing Studies Board, National Academy of Sciences, pp. 91, 1983.

[10] R. Jain, R Kasturi and B.G. Schunch, Machine Vision , *McGraw-Hill*, 1991.

[11] A. Low, Introductory Computer Vision and Image Processing, *McGraw-Hill*, 1991.

[12] J.M. Jolion, "Computer Vision Methodologies", CVGIP: Image Understanding, vol. 59, no. 1, pp. 53 – 71, January 1994.

[13] D. Marr, "Vision", Freeman, New York, 1982.

[14] Y. Aloimonos, I. Weiss and A. Bandopadhay, "Active Vision", International Journal on Computer Vision, vol. 2, no. 1, pp. 333 – 356, 1988.

[15] H. Freeman, "Machine vision: Algorithms, Architectures, and Systems", Perspectives in Computing, vol. 20, Academic Press, 1988.

[16] M. A. Lavin, "Industrial machine vision – Where are we? What do we need? How do we get it?", Machine Vision (H. Freeman, Ed.) Perspective in Computing, vol. 20, pp. 161 – 185, Academic Press, 1988.

Chapter 2

The Robotic Vision: State of Art

"... But a sweet and sudden light excites my weak heart,
barren by this Icy Life. It's so easy to be happy? ..."
"...Ma una luce in-aspettata e dolce riscalda il mio fragile cuore,
dal ghiaccio della vita inaridito. E' così facile essere felici? ..."
A.L.

2.1 Introduction

Experimental robotics is a research field in which the efforts of different disciplines and scientists convey: engineering, informatics, A.I., physiology, etc. The aim of robotics is to produce human-independent and efficient automatons. Nowadays robots are produced with the hardware that is sold in the consumer-market, so we can overcome budget problems maintaining a good level of sophistication. Modern architectures based on Intel x86, ARM and PowerPC, both in the general purpose and embedded versions, have a calculus power that can not face the computational load of vision algorithms. This mismatch has not been a problem, since it forced researchers to produce even more efficient and smart algorithms.

The aim of artificial vision is to give to a machine some visual capacities so that it can get information from the external environment. Artificial vision comes from the evolution of Image

Processing and Pattern Recognition [1] [2]. In a vision system we have two basic steps: *image acquisition* and *image processing*.

Two or more cameras are needed in order to acquire images used, subsequently, for navigation. The choice of the camera, that is going to be used, is of utmost importance for the following elaboration. There are a lot of camera models, from the simple pin-hole[1] one to the digital one. The image quality and the next elaboration step will be dramatically influenced by the choice of the cameras. For example, since with a pin-hole camera we acquire an image with a great radial distortion, the very first thing that the vision system has to do is filter the image in order to reduce distortion. If we use a camera with enough focal length the filtering stage is unnecessary. Modern cameras are so sophisticated that they can solve (i.e. at a sensor level) problems like auto-tuning to light variation or contrast. The usually CCD cameras are employed. In vision applications, they have a refresh frequency in the 25-35 HZ range and a contrast intensity on the same image of about 10.000:1 (while common cameras have about 500:1). CMOS technology based sensors improve, greatly, contrast intensity and max resolution.

The image processing system must be a real-time one in order to produce results in a time that can be useful for navigation applications. This is why we need a great calculus power. It is easy to understand

---

[1] **Pin hole**: A pinhole camera is a camera without a conventional glass lens. An extremely small hole in a very thin material can focus light by confining all rays from a scene through a single point.

how the elaboration time is an upper bound (together with the actuators speed) to the robot speed. Some years ago this calculus power was only available on parallel (even embedded) systems, while nowadays we can think of using general purpose computers in the image processing.

There are two main approaches to vision; the natural one and the artificial one. The former aims reproduce in a laboratory a natural vision system, such as Human Vision System (HVS) or insects vision system, the latter uses information coming from the images in a way that is "intelligible" by the machine that has to process them. With such a classification the approach used in this thesis is definable as an artificial one.

In this chapter we show a possible classification of visual systems for Robot Navigation, and underline our focus that is the obstacle avoidance.

2.2 Visual navigation: a possible classification

A possible schema to classify all the approaches for autonomous navigation is shown in the following **Figure 1**.

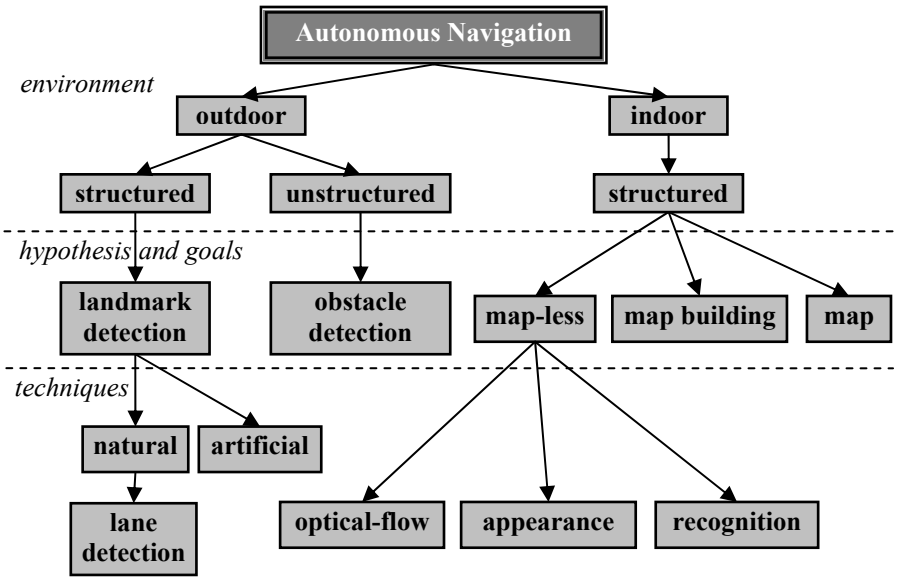


Figure 1: A classification schema for autonomous navigation. A visual system for autonomous navigation of a mobile platform can be classified according to: the kind of *environment*, the *hypothesis and goals*, the *techniques.*

Therefore, a visual system for autonomous navigation of a mobile platform can be classified according to: the kind of *environment*, in order to obtain useful information to take into account in the following analysis (the degree of structured information); the *hypothesis* on the environment, i.e. the kind of scene interpretation we want to face, and the intermediate *goals* we can define to reach the final navigation goal (we can guide a robot or a vehicle using landmarks or detecting obstacles, knowing a map of the environment or without a map); finally, different *techniques* are defined depending on the specific environment and the intermediate goals.

The environment in which the robot is going to move itself has a great influence on the navigation approach. So we have a first

important classification between *indoor navigation* and *outdoor navigation* [3]. Indoor navigation can use techniques unavailable in an outdoor context, such as a model of the external environment.

In indoor navigation we have three main approaches:

1. *Map-Based Navigation* [4,5,6,7,8,9,10,11,12,13,14]. These are systems that depend on user created geometric models or topological map of the environment. Those models may contain different degrees of detail, varying from a complete CAD model of the environment to a simple graph interconnections or interrelationships between the elements in the environment. Since the central idea in any map-based navigation is to provide to the robot, directly or indirectly, a list containing a sequence of landmarks expected to be found during navigation, the task of the vision system is then to search and identify the landmarks observed in an image. Once they are identified, the robot can use the provided map to estimate its position (self-localization) by matching the observation (image) against the expectation (landmark description in the database). The computations involved in vision–based localization can be divided into the following four steps [14]:
    a. *Acquire sensory information* – for vision-based navigation, this means acquiring and digitizing camera images.
    b. *Detect landmarks* – usually this entails extracting edges, smoothing, filtering, and segmenting regions in

the basis of differences in grey levels, colors, depths or motions.

c. *Establish matches between observation and expectation* – within this step, the systems tries to identify the observed landmarks by searching in the database for possible matches according to some measurement criteria.

d. *Calculate position* - once a match (or a set of matches) is obtained, the system needs to calculate its position as a function of the observed landmarks and their positions in the database.

2. *Map Building based Navigation* [15,16,17,18]. These are systems that use sensors to construct their own geometric or topological models of the environment and then use these models for navigation. This approach does not have good performances, since a map is built according to information coming from different kind of sensors, lasers for example.

3. *Map-less Navigation* [19,20,21,23]. These are systems that use no explicit representation at all about the space in which the navigation is going to take place, but rather resort to recognizing objects found in the environment or to tracking those objects by generating motions based on visual observations. It is, of course true that in the approaches that build maps automatically there is no prior description of the environment either: but before navigation can be carried out the system must create a map. In this category we include:

optical flow based navigation, *appearance-based matching* navigation and *object recognition* based navigation.

a. *Optical flow based navigation* [23,24,25]. Santos-Victor et al. [23] have developed an optical-flow based system that mimics the visual behavior of bees. It is believed that the predominantly lateral position of the eyes in insects favors a navigation mechanism using motion-derived features rather than using depth information. In insects the depth information that can be extracted is minimal due to extremely narrow binocular field they process. On the other hand, motion parallax can be much more useful especially when the insect is in relative motion with respect to the environment. Also, the accuracy and the range of operation can be altered by changing the relative speed. For example, features such as "time-to-crash" (which is dependent on the speed) are more relevant than distance; when it is necessary to say, jump over an obstacle. In *robee*, as the robot in [23] is called, a divergent stereo approach was employed to mimic the *centering reflex* of a bee. If the bee is in the centre of a corridor, the difference between the velocity of the range seen with the left and the velocity of the image seen with the right eye is approximately zero, and the bee stays in the middle of the corridor. However, if the velocities are different, the bee moves toward the side

whose image change with smaller velocity. With regard to the robotic implementation, the basic idea is to measure the difference between image velocities computed over a lateral portion if the left and right images and use this information to guide the robot. For that the authors computed the average of the optical flows on each side.

b. *Appearance-based matching* navigation [26,27]. Another way of achieving autonomous navigation in a map-less environment is to "memorize" this environment. The idea is to store images or templates of the environment and associate those images with the commands or controls that will lead the robot to its final destination. An images database is created that the system uses to verify if the robot is in a situation already faced, if yes the navigation goes on "*remembering*" the results acquired during past elaborations.

c. *Object recognition* [7,28]. Object recognition is the basic idea to another approach to navigation. With such an approach commands as "move to the desk in front of you" are given to the robot. In this case the command is very important for the system as it carries information within itself. For example "move to the desk in front of you" says to the robot that the landmark is a desk and that it is situated in front of it.

As with indoor navigation, outdoor navigation usually involves obstacle-avoidance, landmark detection, map building/updating, and position estimation. However, as in the research reported so far in outdoor navigation, a complete map of the environment is hardly ever known a priori and the system has to cope with the objects as they appear in the scene, without prior information about their expected position. Nevertheless, outdoor navigation can still be divided in two classes according to the level of structure of the environment: outdoor navigation in *structured environments* and in *unstructured environments*.

1. In general, outdoor navigation in structured environments requires some sort of road following [29,30]. Road following means an ability to recognize the lines that separate the lanes or separate the road from the berm; the texture of the road surface and the adjoining surfaces; etc. In systems that carry out road following, the models of the environment are usually simple, containing only information such as vanishing points, road and lane widths (*lane detection*), etc. Road-following for outdoor robots can be like hallway-following for indoor robots, except for the problems caused by shadows, changing illumination conditions, changing colours, etc.
2. With regard to unstructured outdoor navigation [32,33,34,35,36,37], we define an outdoor environment with no regular properties that could be perceived and tracked for navigation as an unstructured environment. In such cases, the vision system can make use of at most a generic

characterization of the possible obstacles in the environment. Unstructured environment arise in cross-country navigation, as for example in planetary (lunar/Martian-like) terrain navigation [35,36,37]. Sometimes in these sorts of applications, the robot is supposed to just wander around, exploring the vicinity of the robot without a clean-cut goal.

We can also design a relation between the degree of autonomy of a robot and the environment knowledge as shown in **Figure 2**.

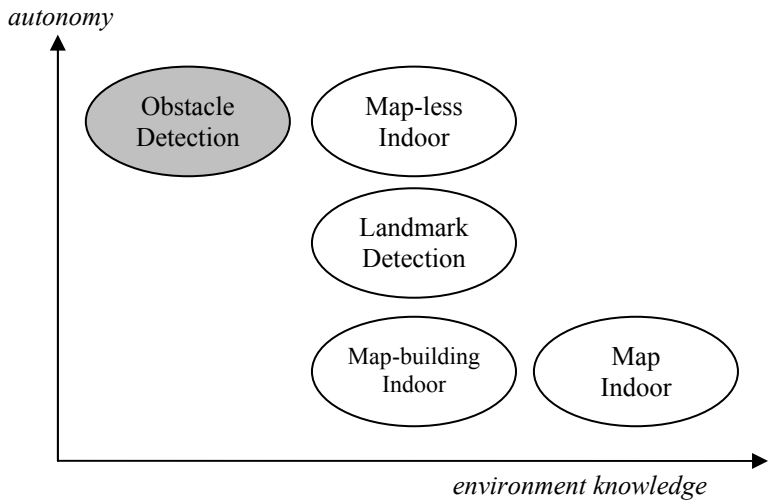


Figure 2: A relation between the degree of autonomy of a robot and the environment knowledge

As it is clear from **Figure 2**, the degree of autonomy of a robot is in inverse relation to the knowledge of the environment. Moreover, it is heavily dependent on the *hypothesis and goals* section, shown in **Figure 1**. Therefore the previous relation can also be read in a 3D space as shown in **Figure 3**, where the *degree of autonomy* of a robot

is in relation with the *environment knowledge* and the *hypothesis and goals*.

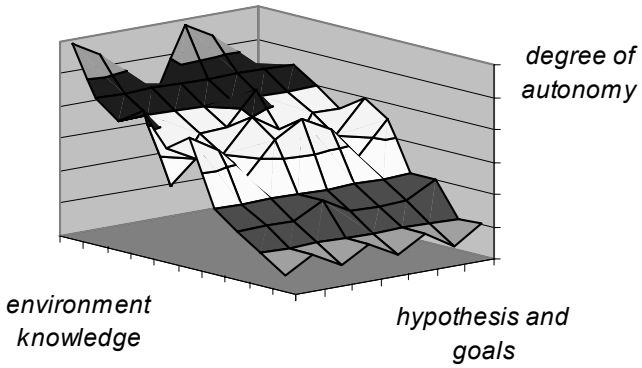


Figure 3: A relation between the degree of autonomy of a robot and the environment knowledge and the hypothesis and goals.

We are particularly interesting into the Obstacle Detection and Map-less Indoor. In both the contexts, we do not have a large knowledge of the environment, so that a robot has to understand what happens around itself, which objects are to avoid and which ones to follow. In the next section, we will explain several approaches for Obstacle Detection, that are also good for Map-less Indoor context. The only difference between the two aims is that in case of indoor applications we can use a low environment information related to the memorization or the recognition of some objects, highly present in the environment. Finally, in the outdoor context, the lighting changes, shadows and the strongly interaction with other objects require a higher level of attention to the robustness.

2.3 Obstacle detection and avoidance

Obstacle detection and avoidance consists of obstacle detection in order to find the safety path to follow during the autonomous navigation of a mobile platform.

An obstacle could be a moving or a motionless object, it could belong to a precise set of objects (for example other robots or vehicles, people), or it could be a generic and unexpected object. Afterwards, a system for obstacle detection depends on what an obstacle means. For example, if we are interested in automatic guidance, we could assume only moving obstacles (no one stops on the road) and vehicles and people being the only kind of obstacle. Moving from these hypothesis, the obstacle detection is easy and can be resolved with a simple *template matching* [38,39]. In cases of exploration, instead, a robot works in an open environment (without a road or a line to follow) and different kinds of obstacle can be present.

We can address the problem related to obstacle detection and avoidance, in general, using two different paradigms (see **Figure 4**):

- 3-D motion estimation
- 3-D space reconstruction

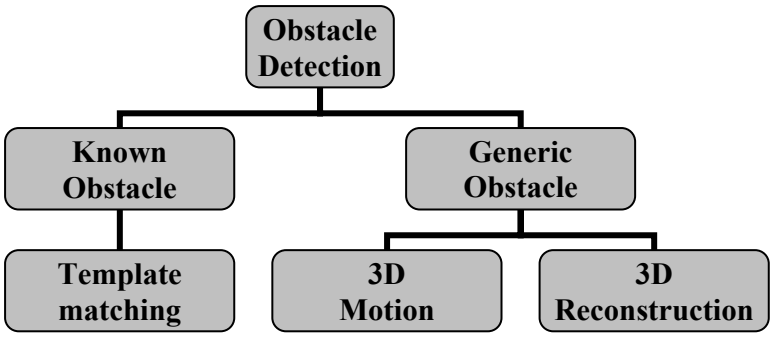


Figure 4: Obstacle detection and avoidance

The first one is the traditional problem of "structure from motion" [40,41,42], that is to find methods of recovering the 3-D motion parameters and the structure of the objects from dynamic images. The way the problem has been addressed was first to compute the exact position to which each point in the image has moved. In cases of small motion the vector field that represents the change of every point in the image, the so-called *optical flow field*, is computed from spatio-temporal derivatives of the image intensity function. This requires the employment of additional constraints, such as smoothness. In cases where the motion is large, features such as points, line or contours in images taken at different time instants are corresponded. From the derived optical flow field or the correspondences between features the 3-D motion is then determined. From measurements on the image we can only compute the relative motion between the observer and any point in the 3-D scene. Consequently, we can recover the so-called *egomotion*, that is the 3-D motion vector of the observer, looking at the static part of the scene, and a *3-D object motion* relative to the observer. There exist many reasons for the limitations of optical flow approach in real applications. To begin, the computation of optical

flow is an ill-posed problem, i.e., unless we impose additional constraints, we cannot estimate it. Thus, the 3-D motion estimation is highly sensitive to small changes in the data. Finally, because of the computation of a relative motion between objects and observer, the objects with a motion similar to the observer could be ignored as obstacles.

The 3-D space reconstruction approaches [43,44] face the problem of obstacle detection looking at a certain representation of the environment. Instead of a comparison between images in a time sequence in order to have motion vectors, it is addressed as a spatial relationship between more images of the same scene in the same instant from different points of view. The most common approach is the *stereo vision*, where two cameras are used to have depth information of the environment. The advantage of this technique is that a quasi full description of the scene can be done, so that an obstacle is precisely identified and also recovered as a solid object, maybe useful for recognition aims. Unfortunately, even this technique presents some problems in real contexts. It is usually time-consuming, then it can be lacking in robustness in cases of small changes between the different view point images, and it requires strong calibration between images.

In this thesis, we address the obstacle detection and avoidance problem as a 3-D reconstruction problem. We use stereo vision to detect motion-less objects, in the 3-D space. If these objects have some dangerous characteristics, i.e. they are big objects and/or close to the robot, we label them as obstacles. We choose to face the

problem using a 3-D space reconstruction approach, instead of the structure from motion approach, because we prefer a more detailed description of the scene, we want to detect also obstacles with the motion similar to the robot, and finally because the spatial coherence is less sensitive to the changing of the acquiring condition of the environment.

Both optical flow and stereo vision are used to analyze the motion of obstacles in relation to the mobile platform. In this way we can distinguish motionless from moving obstacles, and perform a tracking of the moving ones.

2.4 A schema for a Robotic Vision System

In this section we are going to analyze a possible structure for a vision system. Even if each project has ad hoc solutions in order to face different necessities, we can focus on the common features shared by the different system obtaining a scheme useful in many vision applications. In **Figure 5** this schema is given. This is a *bottom-up* model in which the elaboration is from the image to the command sent to the robot. The very first level, always present in a vision system, is the image acquisition level.

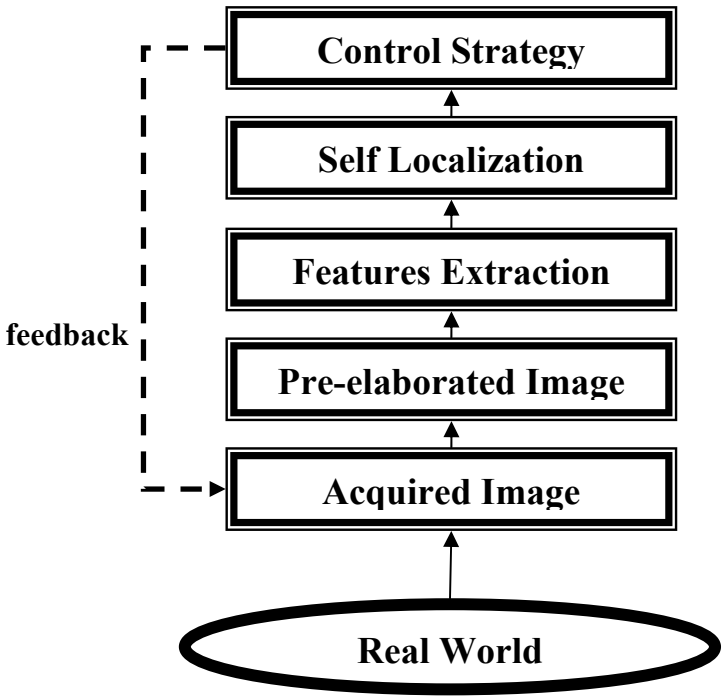


Figure 5: A vision system model

Pre-elaboration level can include different techniques depending from the application such as: thresholding, edge detection, distortion removing, IPM. These techniques can or cannot be applied to the acquired image according to the extraction features algorithm adopted. For example, in grey values images usually a filter with threshold is used. The result of the entire process is critically dependent from these stages. Low definition cameras, unnecessary filtering algorithms, can produce a failure of the entire system. In the feature extraction we elaborate the images after the filters application. The elaboration in this step can be very different. Usually, we search for particular region of the image easily recognizable by its shape; this is the case of the

signs on the floor based navigation.

Other approaches are also followed. For example, in ALVINN (Autonomous Land Vehicle In a Neural Net) [45,46,47] the images are processed by a neural network. Moreover, signs on the wall can be extracted not only based on their shape but also on their color or contrast among the different elements. Sometimes, the elements that make an unstructured environment a structured one are artificially inserted; in other cases they are "naturally" present in the scene as, for example, the line between the wall and the floor in a room.

After the features extraction, we evaluate the position of the vehicle in the scene (self localization). In this stage we calculate the difference between the optimal and the real position of the robot. Different errors can be taken in account: *offset error*, i.e. the distance between the barycentre of the vehicle and the point in which it was supposed to be, *orientation error*, i.e. the angle between the vehicle axis and the tangent to the ideal trajectory, *curvature error* that is the difference between the ideal and real bending radius.

Once errors are calculated, we can have a control strategy to cancel or minimize them. In this step we take in account not only the errors just evaluated but also the interaction floor-vehicle, the delay in the actuators "steering" capacity, etc., so we have different strategies according to the context. As it is shown in **Figure 5** the control strategy step directly influences the first step of image acquisition (as a feedback). In fact we refer to an active vision system [48], considering that the observer (the robot, for example) actively affects the visual system.

References

[1] W.K.Pratt, "Digital Image Processing", John Wiley & Sons, New York, 1978.

[2] A. Watt and F. Policarpo, "The Computer Image", Addison-Wesley, 1998.

[3] G. N. DeSouza and A. C. Kak, "Vision for Mobile Robot Navigation: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pp. 237-267, February 2002.

[4] H.P. Moravec and A. Elfes, "High Resolution Maps from Wide Angle Sonar", in Proceedings of IEEE International Conference in Robotics and Automation, pp. 116-121, 1985.

[5] J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots", IEEE Transactions Systems, Man, and Cybernetics, vol. 19, no. 5, pp. 1179-1187, 1989.

[6] D. Kim and R. Nevatia, "Representation and Computation of the Spatial Environment for Indoor Navigation", in Proceedings of International Conference in Computer Vision and Pattern Recognition, pp. 475-482, 1994.

[7] D. Kim and R. Nevatia, "Symbolic Navigation with a Generic Map", in Proceedings of IEEE Workshop Vision for Robots, pp. 136-145, August 1995.

[8] J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments", in Proceedings of IEEE International Conference in Robotics and Automation, pp.

572-577, 1990.

[9] J. Borenstein and Y. Koren, “The Vector Field Histogram–Fast Obstacle Avoidance for Mobile Robots”, IEEE Transactions Robotics and Automation, vol. 7, no. 3, pp. 278-288, June 1991.

[10] G. Oriolo, G. Ulivi and M. Vendittelli, “On-Line Map Building and Navigation for Autonomous Mobile Robots”, in Proceedings of IEEE International Conference in Robotics and Automation, pp. 2900-2906, May 1995.

[11] A. Ohya, A. Kosaka and A. Kak, “Vision-Based Navigation by Mobile Robots with Obstacle Avoidance Using Single-Camera Vision and Ultrasonic Sensing”, IEEE Transactions Robotics and Automation, vol. 14, no. 6, pp. 969-978, December 1998.

[12] P. Zingaretti and A. Carbonaro, “Route Following Based on Adaptive Visual Landmark Matching”, Robotics and Autonomous Systems, vol. 25, no. 3-4, pp. 177-184, November 1998.

[13] J. Horn and G. Schmidt, “Continuous Localization of a Mobile Robot Based on 3D-Laser-Range-Data, Predicted Sensor Images, and Dead-Reckoning”, Robotics and Autonomous Systems, vol. 4, no. 2-3, pp. 99-118, May 1995.

[14] J. Borenstein, H. R. Everett and L. Feng, “Navigating Mobile Robots: Systems and Techniques”, eds. Wellesley, Mass.: AK Peters, 1996.

[15] H.P. Moravec, “The Stanford Cart and the CMU Rover”, Proc. IEEE, vol. 71, no. 7, pp. 872-884, July 1983.

[16] C. Thorpe, “An Analysis of Interest Operators for FIDO”, in Proceedings of IEEE Workshop Computer Vision: Representation

and Control, pp. 135-140, April/May 1984.

[17] I.J. Cox, "Modeling a Dynamic Environment Using a Bayesian Multiple Hypothesis Approach", Artificial Intelligence, vol. 66, no. 2, pp. 311-344, April 1994.

[18] N. Ayache and O. D. Faugeras, "Maintaining Representations of the Environment of a Mobile Robot", IEEE Transactions Robotics and Automation, vol. 5, no. 6, pp. 804-819, 1989.

[19] T. Nakamura and M. Asada, "Motion Sketch: Acquisition of Visual Motion Guided Behaviors", in Proceedings of 14th International Joint Conference in Artificial Intelligence, vol. 1, pp. 126-132, August 1995.

[20] T. Nakamura and M. Asada, "Stereo Sketch: Stereo Vision-based Target Reaching Behavior Acquisition with Occlusion Detection and Avoidance", in Proceedings of IEEE International Conference in Robotics and Automation, vol. 2, pp. 1314-1319, April 1996.

[21] E. Huber and D. Kortenkamp, "Using Stereo Vision to Pursue Moving Agent with a Mobile Robot," in Proceedings of IEEE International Conference in Robotics and Automation, vol. 3, pp. 2340-2346, May 1995.

[22] A. Bernardino and J. Santos-Victor, "Visual Behaviours for Binocular Tracking", Robotics and Autonomous Systems, vol. 25, no. 3-4, pp 137-146, November 1998.

[23] J. Santos-Victor, G. Sandini, F. Curotto and S.Garibaldi, "Divergent stereo for robot Navigation: learning from bees", in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York 1993.

[24] A. Dev, B.J.A. Krose and F.C.A. Groen, “Navigation of a Mobile Robot on the Temporal Development of the Optic Flow”, in Proceedings of IEEE International Conference in Intelligent Robots and Systems, pp. 558-563, September 1997.

[25] A. Rizzi, G. Bianco and R. Cassinis, “A Bee-Inspired Visual Homing Using Color Images”, Robotics and Autonomous Systems, vol. 25, no. 3-4, pp. 159-164, November 1998.

[26] P. Gaussier, C. Joulain, S. Zrehen and A. Revel, “Visual Navigation in an Open Environment without Map”, in Proceedings of IEEE International Conference in Intelligent Robots and Systems, pp. 545-550, September 1997.

[27] C. Joulian, P. Gaussier, A. Revel and B. Gas, “Learning to Build Visual Categories from Perception-Action Associations”, in Proceedings of IEEE International Conference in Intelligent Robots and Systems, pp. 857-864, September 1997.

[28] D. Kim and R. Nevatia, “Recognition and Localization of Generic Objects for Indoor Navigation Using Functionality”, Image and Vision Computing, vol. 16, no. 11, pp. 729-743, August 1998.

[29] S. Tsugawa, T. Yatabe, T. Hirose and S. Matsumoto, “An Automobile with Artificial Intelligence”, in Proceedings of Sixth International Joint Conference in Artificial Intelligence, pp. 893-895, 1979.

[30] T.M. Jochem, D.A. Pomerleau and C.E. Thorpe, “Vision-Based Neural Network Road and Intersection Detection and Traversal”, in Proceedings of IEEE International Conference in Intelligent

Robots and Systems, vol. 3, pp. 344-349, August 1995.

[31] R. Pagnot and P. Grandjean, “Fast Cross Country Navigation on Fair Terrains”, in Proceedings of IEEE International Conference in Robotics and Automation, pp. 2593-2598, May 1995.

[32] B. Wilcox and D. Gennery, “A Mars Rover for the 1990’s”, J. Brit. Interplanetary Soc., vol. 40, pp. 484-488, 1987.

[33] B. Wilcox, L. Matthies, D. Gennery, B. Copper, T. Nguyen, T. Litwin, A. Mishkin and H. Stone, “Robotic Vehicles for Planetary Exploration”, in Proceedings of IEEE International Conference in Robotics and Automations, pp. 175-180, May 1992.

[34] L. Boissier, B. Hotz, C. Proy, O. Faugeras and P. Fua, “Autonomous Planetary Rover: On-Board Perception System Concept and Stereovision by Correlation Approach”, in Proceedings of IEEE Int’l Conf. Robotics and Automation, pp. 181-186, May 1992.

[35] E. Krotkov and M. Hebert, “Mapping and Positioning for a Prototype Lunar Rover”, in Proceedings of IEEE International Conference in Robotics and Automation, pp. 2913-2919, May 1995.

[36] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe and T. Litwin, “Mars Microrover Navigation: Performance Evaluation and Enhancement,” in Proceedings of IEEE International Conference in Intelligent Robots and Systems, vol. 1, pp. 433-440, August 1995.

[37] L. Matthies, T. Balch and B. Wilcox, “Fast Optical Hazard Detection for Planetary Rovers Using Multiple Spot Laser

Triangulation", in Proceedings of IEEE International Conference in Robotics and Automation, vol. 1, pp. 859-866, April 1997.

[38] S. Denasi, C. Lanzone, P. Martinese, G. Pettiti, G. Quaglia and L. Viglione, "Real-time system for road following and obstacle detection", in Proceedings of the SPIE on Machine Vision Applications, Architectures, and Systems Integration III, vol. 2347, pp. 70–79, October 1994.

[39] M. Lützeler and E.D. Dickmanns, "Road recognition with MarVEye", in Proceedings of the IEEE International Conference on Intelligent Vehicles, pp. 341–346, October 1998.

[40] H. C. Longuet-Higgins, "A computer algorithm for reconstruction a scene from two projections", Nature, vol. 293, pp. 133-135, 1981.

[41] M. E. Spetsakis and J. Aloimonos, "Structure from motion using line correspondences", International Journal of Computer Vision, vol. 4, pp.171-183, 1990.

[42] R.Y. Tsai and T.S. Huang, "Uniqueness and estimation of three dimensional motion parameters of rigid objects with curved surfaces", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, no. 1, pp. 13-26, 1984.

[43] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", International Journal of Computer Vision, vol. 47, no. 1, pp. 7-42, 2002.

[44] C. Zhang, "A Survey on Stereo Vision for Mobile Robots", Technical report, Dept. of Electrical and Computer Engineering,

Carnegie Mellon University, Pittsburgh, PA, 15213, USA, 2002.

[45] D.A. Pomerleau, "ALVINN: An Autonomous Land Vehicle in a Neural Network", Technical Report CMU-CS-89-107, Carnegie Mellon University, 1989.

[46] D.A. Pomerleau, "Efficient Training of Artificial Neural Networks for Autonomous Navigation", Neural Computation, vol. 3, pp. 88-97, 1991.

[47] D.A. Pomerleau, "Reliability Estimation for Neural Network Based Autonomous Driving", Robotics and Autonomous Systems, vol. 12, pp. 113-119, 1994.

[48] Y. Aloimonos, I. Weiss and A. Bandopadhay, "Active Vision", International Journal on Computer Vision, vol. 2, no. 1, pp. 333–356, 1988.

Chapter 3

Our System Architecture

"... Love guides me in the dark, as the trail of an albatross guides a desperate trembling crew, lost in a black and empty night. ..."
"... L'amore mi fa strada nel buio come l'albatro segna la scia alla tremante ciurma disperata, persa nel vuoto nero della notte. ..."
A.L.

3.1 Introduction

This chapter is devoted to describe the architecture of the whole system for obstacle detection and avoidance. An architecture for computer vision can follow several methodologies, or better different philosophies. In fact, the computer vision is an hard field because a visual image is inherently ambiguous and perception is essentially a matter of resolving ambiguities by using knowledge from the external world. For this reason, the kinds of information we consider and the way how this information is used inside the system characterize one approach from each other. Therefore, in the next section we present a brief survey on computer vision methodologies. Then our system architecture is described (Section 3) with reference to the chosen methodology and the constraints imposed on the problem.

3.2 A Vision Architecture: methodologies and issues

In general, there are many way to manage the development of a computer vision architecture. Vision is an under constrained problem and therefore that it is necessary to find or develop new constraints in order to make the problem solvable. The used methodologies are different depending on the different sources of constraints they consider. Then, what happens if someone asked "what is vision?" ? Several answers can be given: It is an information process to achieve a representation of the external world from a visual sensor; It is the capability for a machine to actively interact with its surrounding; It is an instrument to reach some purposes in different contexts; It is the three answers together. This is a possible classification of different methodologies in Computer Vision as reported in [1]:

- Recovery Approach
- Active Vision
- Goal-directed Vision
- Systemic Approach

Recovery Approach

The recovery approach is mainly due to the contribute of David Marr. The philosophy of Marr is detailed in the presentation of *Vision* [2] and is known as the *recovery school*. In fact, according to this theory, the main goal of vision is to derive a representation of the external

world. Marr conceives a general system in which there are not *a priori* knowledge on the goal and there are no constraints on the observer. In this approach, to render a vision system solvable, Marr propose the introduction of constraints on the nature of observed scene (for example hypothesis on the nature of the surfaces, etc.). So, in terms of source of constraints, we can say that in recovery approach we do not have any constraints on the goal and on the observer and we have only constraints on the observed scene. The goal of this methodology is to create a general observer that construct en efficient symbolic descriptions from images of the world. There are not any goal of creation of new objectives neither any goal of change of the scene. It is assumed that any further goal can be achieved from the data stored in the representation of the external world. Another characteristic of this approach is the fact that the observer is mainly passive.

Active Vision

The concept of active vision has been proposed by Aloimonos *et al.* [3] in order to overcome most of problems with which the visual recovery approach has to deal. The main idea is that "perceptual activity is exploratory and searching". The authors want to understand how an active observer interacts with its environment. The active vision approach adds general constraints on the observer (e.g. the observer moves with a know motion). Furthermore the active vision provide specifics objectives (e.g. obstacle avoidance in robotic vision). In any case the objective goal is still to extract structure from a scene or a sequence of scene. So the goal of the methodology, as the

recovery approach, is to create a general observer without any change at the scene and without any creation of new objectives.

Goal-directed Vision

An introduction to this approach to vision is due to Bajcsy [4] and can be summarized by "we do not just see, we look". Therefore, the focus of this approach is to act toward the external world in order to achieve a particular goal. So the purpose of vision can no longer be reduced to description of scenes, but it tries to directly solve a set of visual tasks using appropriate information, representations, algorithms, etc. This approach can be seen as a generalization of the active vision, where new constraints, extracted from the goal, have been added in order to make the vision problems more tractable.

Systemic Approach

This methodology was introduced for the first time in the field of Computer Vision in 1994 from Jolion [1]. The general idea is to consider constraints from all types of sources. In this approach we consider constraints coming from the scene, from the objective and from the observer. In particular, the nature of the observer (architecture type, memory capacity, processing time, etc.) is taken into account to choose the technique of processing. Furthermore systemists argue that we can found other source of constraints by looking the system as a global entity. The properties of this entity, in fact, should hopefully make solvable the subtasks. Besides in this approach great emphasis on the communication network between

system components; and other constraints can be inferred from analyzing the set of relations embodied in the system.

3.3 Our System Architecture

Our goal is to guarantee an autonomous navigation of a mobile platform in an unknown environment. We face this general problem investigating several subtasks and defining some constraints on the scene, the observer and the objective. For this reason our system can be classified as a systemic approach. We can divide the whole system into three most significant tasks (see **Figure 1**): a suitable *representation of the environment*; image analysis in order to *detect obstacles*; image understanding for *obstacle avoidance* and *behavioral analysis*.

Our system is also an active system, in fact, the mobile platform interacts with the external environment so the system is able to change its parameters according to the change of the real world or according to the behavioral analysis. For example, if an interesting object is in the scene the system can acquire images with an higher resolution or can refine the representation of the scene only around that object. For this reason in **Figure 1** a feedback loop is shown.

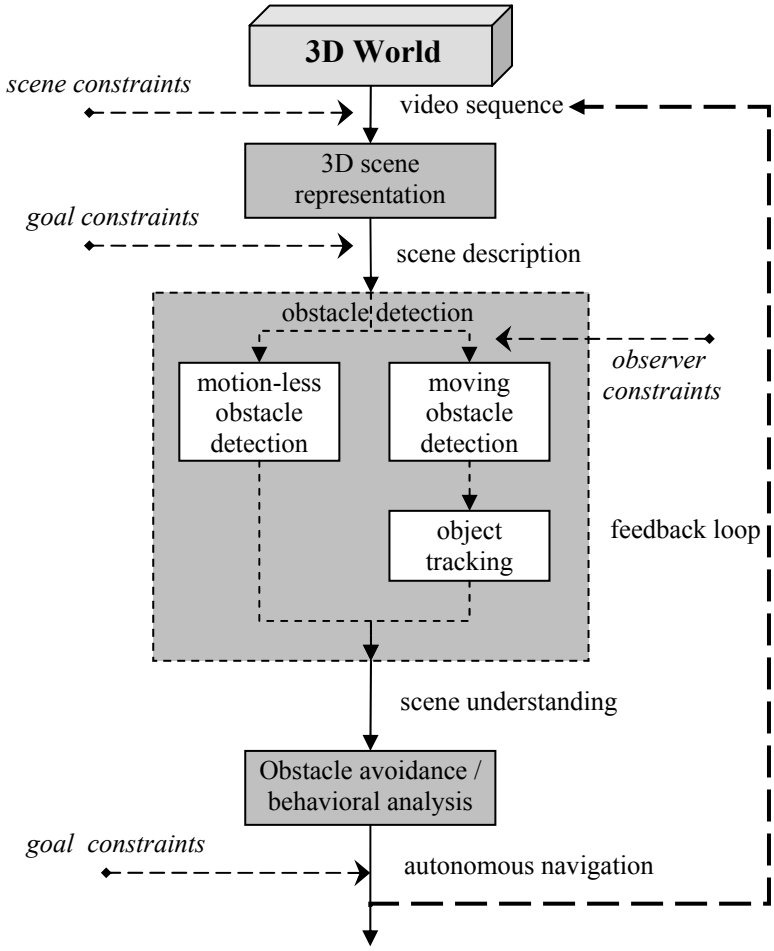


Figure 1: Our system architecture: *representation of the environment*; *obstacle detection*; *obstacle avoidance* and *behavioral analysis*.

As said in chapter 2, the obstacle detection can be faced using two different paradigms: 3-D motion estimation and 3-D space reconstruction. The first one is the most traditionally used, because easy and fast. Only in the last years, the researchers have started to use the second approach. A 3-D space reconstruction is a more heavy task than 3-D structure from motion, so that it can seem redundant for

autonomous navigation. Anyway, an opportune space representation allows a deeper knowledge of the environment, and some constraints can be considered in order to reduce computation time and to adapt the 3-D representation for our aims. The major contribution of the thesis concerns this "perceptive" representation of the environment, that it is not a "passive" representation, but related to the final goal of autonomous navigation. Now, we will describe briefly the different modules, and in the following chapters, we will explain in details the steps of 3D scene representation and obstacle detection (motion-less and moving obstacle detection).

Video Acquisition

The inputs of the system are the frames from a camera. The camera can be both an analog or a digital camera, anyway the input of the system has to be a digital frame: an array of pixels represented in some coding. In our case we use RGB values for each pixel. We work with uncompressed frames data. The acquisition of a frame happens as follow: the camera captures a frame from the scene and sent it to the system; an hardware for the acquisition of images (i.e. frame grabber) is provided in the system and it is responsible for storing the data in a video memory; the system takes the images from video memory and process it. The camera does not capture any other images from the scene until the system finish the processing. So, if the processing is too slowly we should lost some interesting events occurred in the scene. In particular, we use a stereo camera in a parallel focuses configuration. In fact, the 3-D scene representation is achieved using the stereo vision paradigm. The two cameras are supposed to have the

same physical proprieties, as resolution, focal length, lighting sensitivity, etc. However, the stereo vision approach is highly sensitive to any changes between the left and right image. Our system overcomes this difficulty during the phases of 3-D representation and obstacle detection.

3-D scene representation

Stereo cameras are built to simulate the way biological Human Vision System (HVS) works to obtain depth information from the captured images. By calculating the vertical displacement of each point between the two captured images, stereo cameras can tell how far the point is from the observer. We immediately anticipate (it will be more clear in the chapter about 3-D representation) that a punctual reconstruction of the scene is out of our aims. We are not interested in the depth of a point, but it is enough to segment the image into several regions of interest and recover depth information on them. In this way, the 3-D representation of the scene is *suitable* to navigation aims. Moreover, this representation guarantees a higher robustness in real contexts, that is in cases of small changes in the data between left and right image.

Obstacle detection

The obstacle detection task can be divided into motion-less obstacle detection and moving obstacle detection. In the first case a simple connected components analysis [5] of regions with the same value of depth can segment an entity that represents an obstacle for the mobile platform. The moving obstacles are detected using both 3-D representation and optical flow information. In this way, one

information tries to limit the uncertainties coming from the other one, and vice versa. Then the moving objects are followed along time (object tracking) only to understand the dynamic of the objects in the scene. At the end of this phase we have a certain understanding of the scene, so we are ready to move.

Obstacle avoidance and behavioral analysis

Starting from a representation and understanding of the scene it is possible to guide a mobile platform through a safety path. Moreover, some application tasks can be defined, as follow a particular object in the scene, or just move around for exploration aims avoiding obstacles. This last part of the system is out of the focus of this thesis.

3.4 Our constraints

As we told before our system belongs to the systemic vision approach. Here we show the constraints we have to consider to achieve the described goal (obstacle detection).

As regard the scene:

- The scene does not have to be too complex: normally in the autonomous navigation framework it is true, the scene is characterized by few objects (5-10) and a low dynamism (the speed of the objects is comparable with the speed of the mobile platform).
- The light should be almost uniform in the scene and in time: the stereo vision and optical flow are sensitive to changes in the data, even if our system guarantee a good robustness in

cases of small changes.

These constraints are considered and positively used for the 3-D space representation (see **Figure 1**).

As regard the goal:

- The obstacle detection task has to be scalable in time: the mobile robot has to quickly react to the external events, so the real-time requirement is needed. However, a finer but slower investigation of the environment can be made by our system, in fact a robot could decide to stop to look better and then move again.
- The obstacle detection task has to be scalable in performance: it is the same of above, but with reference to the details of the 3-D representation and of the detected obstacles.

As shown in **Figure 1**, these constraints are considered before the obstacle detection phase and during the navigation.

As regard the observer:

- Good quality cameras are taken into account: a good input image is necessary for a good representation of the scene. We consider a couple of standard CCD cameras with a resolution at least 384x288 pixels and a frame rate at least 15 fps (frame per sec). Different focal lengths are used in order to test our system with different angles of view. The cameras are calibrated only in the start-up of the system.
- The mobile platform moves slowly. Each adjacent couple of frames in the image sequence has to be overlapped for about 80%. Therefore, the speed of the mobile platform and the

frame rate of the cameras have to guarantee this constraint.

- The motion vector of the mobile platform is known from other sensor data. The required incertitude depends on the scene dynamics.
- Mechanical vibrations of the cameras are limited: the stereo vision and optical flow are sensitive to local and global perturbations in the images. Our system is enough robust in cases of small vibrations.

References

[1] J.M. Jolion, “Computer Vision Methodologies”, CVGIP: Image Understanding, vol. 59, no. 1, pp. 53–71, January 1994.

[2] D. Marr, “Vision”, Freeman, New York, 1982.

[3] Y. Aloimonos, I. Weiss and A. Bandopadhay, “Active Vision”, International Journal on Computer Vision, vol. 2, no. 1, pp. 333–356, 1988.

[4] R. Bajcsy, “Active perception”, in Proceedings of the IEEE (Special issue on computer vision), vol. 76, no. 8, pp. 996-1005, 1988.

[5] D. H. Ballard and C. Brown, “Computer Vision”, Prentice-Hall, 1982.

Chapter 4

Stereo Vision

"... Where are my doubts?
Affected motions of the Soul, rebelling against banal life of the mortals. ..."
"...Dove sono andati i dubbi miei?
Commossi moti dell'animo che alla banale vita dei mortali si ribella. ..."
A.L.

4.1 Introduction

This chapter is devoted to describe the state of art in Stereo Vision paradigm. Many authors have expressed their conviction that a robotic vision system should aim at reproducing the human vision system, and so should be based on stereo vision. The greatest advantage of stereo vision with respect to other techniques (e.g. optical flow) is that depth can be inferred with no prior knowledge of the observed scene (in particular the scene may contain unknown moving objects and not only motionless background elements). Several methods are proposed in the literature sometimes to improve the efficiency and sometimes to improve the accuracy of the solution.

4.2 Stereopsis

Stereopsis refers to the ability of assigning a depth to the objects in the scene. Human brain can percept in a single image the two different images coming from the eyes. In these images we can see the same object from two different points of view: stereoscopic vision uses these two different views to get a 3D vision of reality. Computational stereopsis is the process in which we get depth information about the scene observed by a cameras pair. The cameras give images of the scene from two different points of view. Two main problems arise in this context: 3D reconstruction and correspondence calculus. Correspondence problem refers to the research in the two images of points that are projection of the same point in the scene (see **Figure 1**); such points constitute a *conjugate pair*. Correspondence problem can be solved since the two images are only slightly different. We will see that we need some constraints in the conjugate pair research since it is possible to have many *false pairs*. Usually the most important of these constraints is the *horizontal epipolar constraint* which says that the correspondent point in an image can be found on the same horizontal line *(the epipolar line)* on the other image. Using this constraint we, dramatically, reduce the complexity of the correspondence problem since this constraint reduces the search space from two-dimensions to one-dimension. In chapter 5 we will also see, on the other hand, how such constraint can not be included in the autonomous navigation context and we shall introduce our solution to evaluate the depth of the objects in the scene relaxing this constraint.

4.2.1 3D Reconstruction

Once we have found conjugate pairs in the two images, it is possible to get the depth of the correspondent points in the scene if we know: the mutual position of the cameras *(extrinsic parameters)* and the sensors parameters *(intrinsic parameters)*.

In **Figure 1** we have two cameras parallel to each other, with coinciding retinal planes (i.e. fixation points at the ∞).

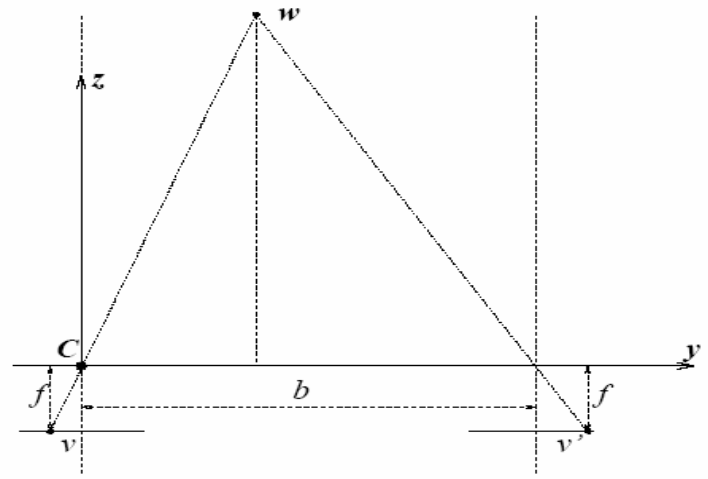


Figure 1: stereoscopic reconstruction.

It is easy to understand that the disparity is only on the x-axis (i.e. is a horizontal disparity), that is why **Figure 1** is two-dimensional. If we consider the reference frame as the left camera, we can write the following equations:

$$\begin{cases} \dfrac{-f}{z} = \dfrac{v}{y} \\ \dfrac{-f}{z} = \dfrac{v'}{b-y} \end{cases} \quad (1)$$

obtaining that:

$$z = \frac{bf}{v - v'} \tag{2}$$

so if we know the system geometry (*b* and *f* in this example) and the disparity *(v-v')* we can calculate the depth by using the last equation. Please note that the *baseline b* works as a scale factor in our problem: a point disparity is proportionally dependent on the baseline. Once we get the z-coordinate we can infer the real x and y coordinates by using the following equations:

$$x = \frac{x_l z}{f} \qquad y = \frac{y_l z}{f} \tag{3}$$

where x_l and y_l are the correspondent coordinates on the projective plane.

It should be noted that all these equations assume no incertitude but, in real conditions, they must be rewritten in order to take incertitude into account.

4.2.2 Correspondence Calculus

We now deal with the main problem of stereo vision: the correspondence calculus (also called *stereo matching*). Let the disparity be the difference (vector) between pixels belonging to a conjugate pair when the two images are overlapped. Correspondence calculus refers to the ability of evaluate disparity for points in the reference image (all the points or a meaningful subset). The result is the *disparity map* (dense or sparse disparity map).

In this evaluation we make the assumption that the two images acquired from the left and right are not so different, i.e. it is possible that a given point in the scene is present in both the images. Using similarity concepts, a point in the image 1 can be seen as correspondent to many points in the image 2: this is the problem of *false correspondences*, it makes harder to solve this task. Moreover, other problems arise in this context; here it follows a short summary of them:

- **Occlusions**. Since the cameras take the images from different points of view, there will be some points in an image that are not going to have a correspondent in the other one. Obviously, no disparity can be calculated for such points [1].
- **Photometric distortion**. Since the surfaces are not perfectly diffusing (i.e. they are not exactly Lambertian), their brightness change according with the angle from which they are viewed, so cameras will acquire a pretty different value for the same points of the scene.
- **Projective distortion**. Due to perspective projection, an object is projected in a different way in the two images.

In order to minimize false matches, some matching constraints have been imposed. Below is a list of the commonly used constraints:

- **Similarity** (or **compatibility** - Grimson, 1981 [2]). The matching pixels must have similar intensity values (i.e. differ lower than a specified threshold) or the matching windows (defined in some area-based methods as said later) must be highly correlated.

- **Uniqueness** (Marr and Poggio, 1979 [3]). Almost always, a given pixel from one image can match *no more than one* pixel from the other image. This constraint can fail if transparent objects are present in the scene. Furthermore, given a pixel *m* in one image, its "corresponding" pixel may be occluded in the other image. In this case, no match should be assigned to *m*.
- **Continuity** (Marr and Poggio, 1979 [3]). The cohesiveness of matters suggests that the disparity of the matches should vary smoothly almost everywhere over the image. This constraint fails at discontinuities of depth, for depth discontinuities cause an abrupt change in disparity.
- **Ordering** (Baker and Binford [4]). If $m \leftrightarrow m'$ and $n \leftrightarrow n'$ and if *m* is *to the left* of *n* then *m*' should also be *to the left* of *n*' and vice versa. That is, the ordering of pixels is preserved across images.
- **Epipolar**. Given a feature point *m* in the left image, the corresponding feature point *m*' must lie on the corresponding epipolar line. As said before, this constraint reduces the search space from two-dimensions to one-dimension.

4.3 Stereo matching methods: State of Art

We will present a brief description of the most important methods for stereo matching; for more details, there is a good taxonomy proposed by Scharstein and Szeliski [5], and a survey on stereo vision for mobile robots by Zhang [6]. Moreover, we will describe, in the following section, some important algorithms we also used during the experimental phase to compare our results.

We can divide the methods for the correspondence calculus in two main categories [5] [6] [7]:

Feature-based: these algorithms try to extract features from the two images (i.e. point or set of points of interest). Matching is applied to the features attributes. These algorithms produce a sparse map that can become a dense one after an interpolation step. They critically depend on the feature extraction stage.

Area-Based: these algorithms consider a window of pixels in an image searching for the most similar in the other one. A correlation measure among intensity values (or a function of them) is used. This process is iterated for each point (so we have a window, at each iteration, centred in the considered pixel), the result is a dense map. Uniform textured regions are a problem for this kind of algorithms; moreover the intensity value of a pixel acquired by a camera is dependent on the point of view (photometric distortion).

4.3.1 Feature based methods

The majority of the feature based algorithms consist of two steps:

Feature detection: salient and distinctive objects (closed-boundary regions, edges, contours, line intersections, corners, etc.) are manually, or, preferably, automatically detected. For further processing, these features can be represented by their point representatives (centers of gravity, line endings distinctive points) who are called control points (CPs) in the literature.

Feature Matching: in this step, the correspondence problem between the features in the sensed image and those detected in the reference image is established. Various feature descriptors and similarity measures along with spatial relationships among the features are used for that purpose.

The implementation of each registration step has its typical problems. First, we have to decide what kind of features is appropriate for the given task. The feature should be distinctive objects, which are frequently spread over the images and which are easily detectable. Usually, the physical interpretability of the features is demanded. The detected feature sets in the reference and sensed images must have enough common elements, even in situations when the images do not cover exactly the same scene or when there are object occlusions or other unexpected changes. The detection methods should have good localization accuracy and should not be sensitive to the assumed image degradation. In an ideal case, the algorithm should be able to detect the same features in all projections of the scene regardless of

the particular image deformation.

In the feature matching step, problems caused by incorrect feature detection or by image degradations can arise. Physically corresponding features can be dissimilar due to the different imaging conditions and/or due to the different spectral sensitivity of the sensors. The choice of the feature description and similarity measure has to consider these factors. The features descriptors should be invariant to the assumed degradations. Simultaneously, they have to be influenced by slight unexpected feature variations and noise. The matching algorithm in the space of invariants should be robust and efficient. Single features without corresponding counterparts in the other image should not affect its performance.

4.3.2 Area based methods

We can divide the area based approaches in: *local* (window-based) and *global* approaches. The local area-based approaches [8,9,10] provide a correspondence for each pixel of the stereo pair. They assume that each pixel is surrounded by a window of pixels having similar disparity; these windows are matched using correlation or a similar technique. They produce a dense disparity map (i.e. a map providing a disparity for each pixel). They can be quite unreliable, not only in homogeneous regions, but also in textured regions for an inappropriately chosen window size. On the other side, the global area-based approaches (that also yield a dense map) try to propagate disparity information from a pixel to its neighbours [11,12], or they

define and minimize some energy function over the whole disparity map [13,14,15]. They have a better performance in homogeneous regions, but they frequently have parameters which are difficult to set, and are highly time-consuming. The area based methods usually make implicit smoothness assumptions

4.4 Stereo matching algorithms

The algorithms we will describe in this section are area based ones. They make the assumption that the epipolar lines are horizontal, so conjugate pairs are on the same x-axis. Using this constraint we, dramatically, reduce the search space from two-dimensions to one-dimension (as said before). It is a very strong constraint in our context (autonomous navigation) because of the mechanical vibrations of the cameras. Moreover, there is no consequence if you work with continuous signal but this is not the case with discrete image and non horizontal epipolar lines are related to bad discrete geometry properties.

4.4.1 SSD

Given a pixel *(u,v)* in the I_1 image we consider the correspondent pixel (u+d,v) in I_2, then we have a window centred in *(u,v)* of *(2n+1)(2m*+1) dimensions; this window is then compared with a window of the same dimensions taken in I_2 at the same x-axis. Since the images are rectified (i.e. horizontal epipolar line) we consider *(u+d,v), d* $\in$

$[d_{min},d_{max}]$. The disparity is the offset corresponding to the max similarity between the grey values of the window.

The adopted metric is the so-called SSD (*Sum of Squared Differences*):

$$SSD(u,v,d)=\sum_{k,l}\left[I_1(u+k,v+l)-I_2(u+k+d,v+l)\right]^2 \quad \textbf{(4)}$$

where $k \in [-n,n]$, $l \in [-m,m]$ and $I(u, v)$ is the grey level of the pixel (u,v).

$$d(u,v)=\arg\min_{d} SSD(u,v,d) \quad \textbf{(5)}$$

In order to evaluate the stereo matching algorithm with SSD, please have a look at the following two figures.

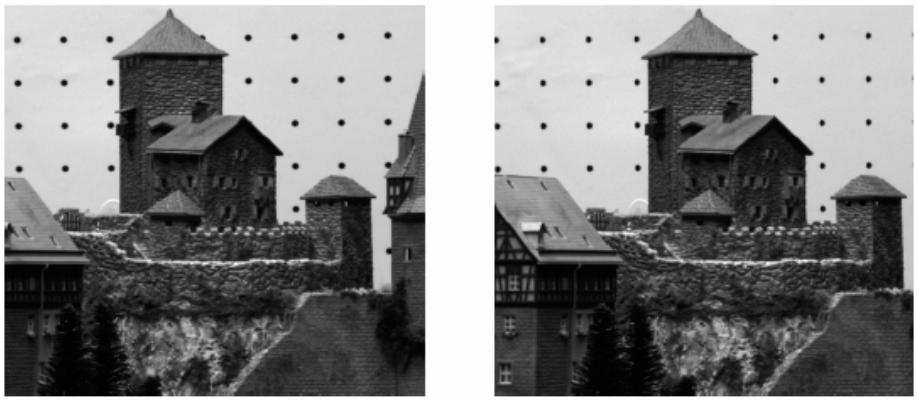


Figure 2: The rectified stereo pair.

The output is shown in Figure 3 in two different ways disparity map and height map.

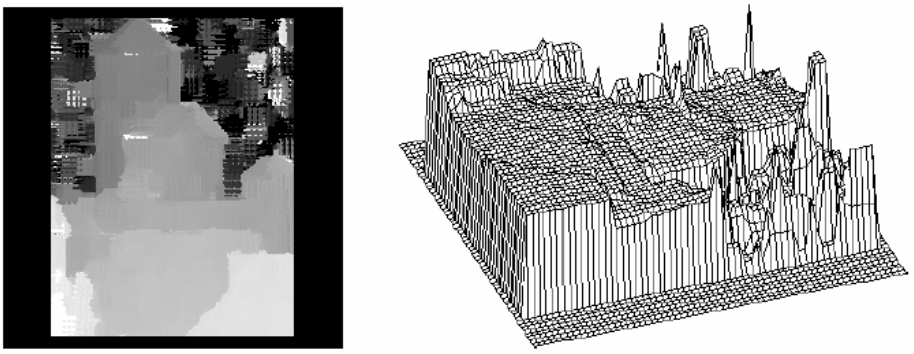


Figure 3: SSD stereo matching based algorithm output.

Depth information acquired by an area based algorithm has not the same reliability for each pixel in the entire image. For example, there is no information available in the textureless and occluded regions. This incomplete information can be integrated using information coming from other sensors; in this case we will need a reliability estimation of the stereo matching algorithm.

4.4.2 Dynamic Programming

I. Cox *et al.* in [17] refer to an approach that consists in assigning a weight to each bad pair (for example, two pixels with very different grey values). The grey values differences follow a Gaussian distribution, with regards to the occlusion the weight is a constant one. If we make the assumption of a known occlusion probability, cost function can be defined on a max similarity criterion.

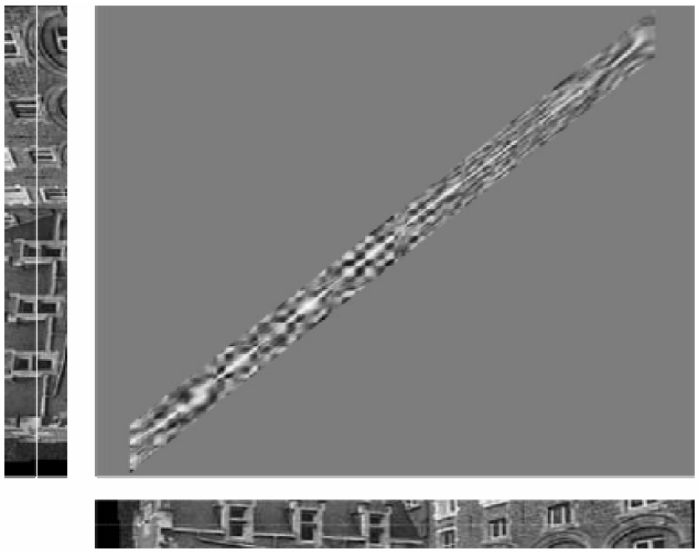


Figure 4: High correlation is expressed by a fair colour

Dynamic programming approach is shown in Figure 4. Correlation functions about the two epipolar line are computed and stored in a DB. Fair areas express a high correlation, dark area a poor one. Solution is given by a "best path" dynamic program problem (i.e. we solve the so called Travelling Salesman Problem).

Furthermore, we can apply to the cost function the ordering and smoothness (continuity) constraints. For example, we can consider the Birchfield and Tomasi cost function [18] [19] :

$$\gamma(M) = \mathrm{N}_{\mathrm{occ}} \cdot k_{occ} - N_m \cdot k_r + \sum_{i=1,N_m} d(x_i, y_i) \qquad \textbf{(6)}$$

where N_{occ} e N_m are the occlusion and matching numbers, k_{occ} is the occlusion weight, k_r is the constant due to a reliable matching and d(x,y) is the dissimilarity function between x and y, such a function could be the SSD.

The execution time (see Figure 5) is, for a 640*480 resolution, on

a workstation Indy by Silicon Graphics, of about 9 seconds with a max disparity equal to 14. Optimizing the implementation, the complexity becomes acceptable.

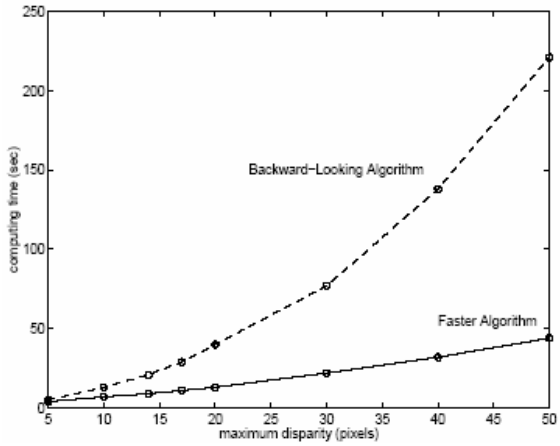


Figure 5: Execution times with different disparities, according to two different implementations: dotted line is not optimized.

4.4.3 Graph Cut

A problem arises in the previous approaches: each epipolar line is independently processed. Solutions obtained in such a way can change a lot and have great artefacts. The graph cut algorithm [16] [1] globally optimizes the solution. The *coherence constraint* takes the place of the *ordering constraint*. The coherence constraint forces locally similarity of the disparities in each direction. For this reason, epipolar lines are grouped in a correlation cube as in Figure 6.

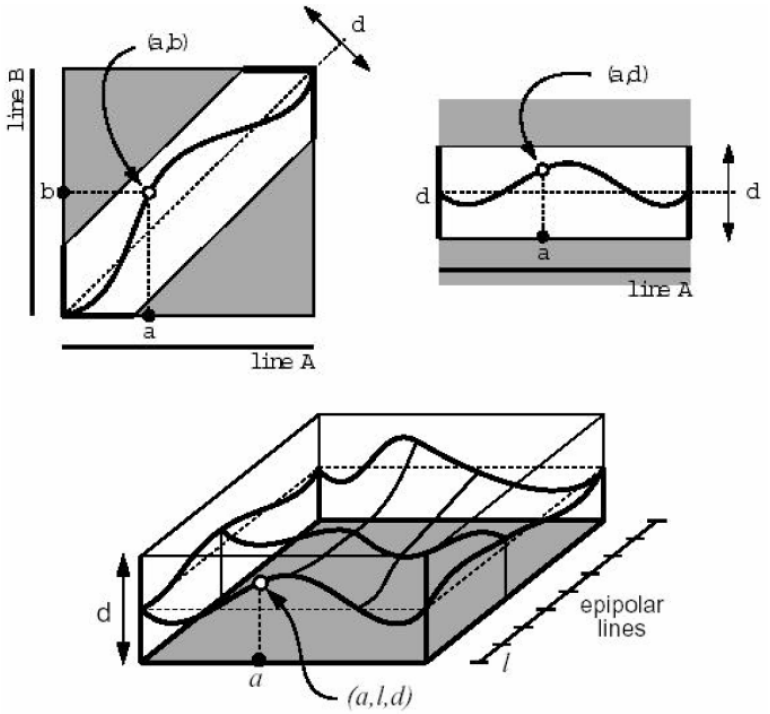


Figure 6: Matching in the entire image. All the epipolar lines *l* is grouped in a correlation cube. The aim is to find the best surface s.t, the coherence constraint that minimizes the total cost

Problem can be now seen as a max flow over a graph. If we add a source and a sink node and we consider all the points in the cube with integer coordinates as vertices in the graph, then the max-flow, between source and sink node, corresponds to the best disparity map.

This algorithm provides good results in fact it is a global approach that minimizes an energy function in the overall image. The main disadvantage is the execution time, (for a 640*480 resolution, on a notebook Intel P4 1.5 GHz 512 Mb RAM, of about 100-500 seconds depending on the maximum value of disparity), so that this algorithm can't be used in a real-time system.

References

[1] H. Hirschmuller, P. R. Innocent and J. Garibaldi, "Real-Time Correlation-Based Stereo Vision with Reduced Border Errors", International Journal of Computer Vision, vol. 47, no. 1, pp. 229-246, 2002.

[2] W.E.L. Grimson, "A computer implementation of a theory of human stereo vision", Philosophical Transactions of the Royal Society of London, vol. B, no. 292, pp. 217-253, 1981.

[3] D. Marr and T.A. Poggio, "A computational theory of human stereo vision", Philosophical Transactions of the Royal Society of London, vol. B, no. 204, pp. 301–328, 1979.

[4] H. H. Baker and T. O. Binford, "Depth from Edge and Intensity-based Stereo", in Proceedings of 7th International Joint Conferences on Artificial Intelligence, pp. 631-636, 1981.

[5] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", International Journal of Computer Vision, vol. 47, no. 1, pp. 7-42, 2002.

[6] C. Zhang, "A Survey on Stereo Vision for Mobile Robots", Technical report, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213, USA, 2002.

[7] B. Zitova and J. Flusser, "Image registration methods: a survey", Image and Vision Computing, vol. 21, pp. 977–1000, 2003.

[8] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment", IEEE Transaction on

Pattern Analysis and Machine Intelligence, vol. 16, no. 9, pp. 920-932, 1994.

[9] A. Fusiello and V. Roberto, “Efficient stereo with multiple windowing”, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 858–863, 1997.

[10] O. Veksler, “Stereo matching by compact windows via minimum ratio cycle”, in Proceedings of the International Conference on Computer Vision, vol. I, pp. 540–547, 2001.

[11] D. Marr and T.A. Poggio, “Cooperative computation of stereo disparity”, Science, vol. 194, no. 4262, pp. 283–287, 1976.

[12] C.L. Zitnick and T. Kanade, “A cooperative algorithm for stereo matching and occlusion detection”, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 22, no. 7, pp. 675–684, 2000.

[13] D. Geiger, B. Ladendorf and A. Yuille, “Occlusions and binocular stereo”, International Journal of Computer Vision, vol. 14, pp. 211–226, 1995.

[14] S. Roy, “Stereo without epipolar lines: A maximum-flow formulation”, International Journal of Computer Vision, vol. 34, no. 2/3, pp. 1–15, 1999.

[15] Y. Boykov, O. Veksler and R. Zabih, “Fast approximate energy minimization via graph cuts”, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222–1239, 2001.

[16] R. Szeliski and R. Zabih, “An Experimental Comparison of Stereo Algorithms”, in Proceedings of the International Workshop

on Vision Algorithms: Theory and Practice, pp. 1-19, September 1999.

[17] I. Cox, S. Hingorani and S. Rao, "A maximum Likelihood Stereo Algorithm", Computer Vision and Image Understanding, vol. 63, no.3, pp. 542–567, 1996.

[18] S. Birchfield and C. Tomasi, "Depth Discontinuities by Pixel-to-Pixel Stereo", International Journal of Computer Vision, vol. 35, no. 3, pp. 269-293, 1999.

[19] S. Birchfield and C. Tomasi, "Multiway Cut for Stereo and Motion with Slanted Surfaces", in Proceedings of the International Conference on Computer Vision, pp. 489-495, 1999.

Chapter 5

Our Approach

"... Where is the bitter smile, caused by the Leopardian pain?
What about the tormenting anxiety, the wild tempest
That prevent my heart from loving? ..."
"...Dove ho represso l'amaro sorriso che il leopardiano patir mi suscitava?
Cosa è stato dell'inquietudine struggente, della tempesta infuriata
che impediva al cuore mio d'amare? ..."
A.L.

5.1 Introduction

In the literature there are a lot of approaches that process the depth information starting from a couple of images acquired from a stereo camera, but many times these approaches need a strong pre-processing phase, i.e. rectification or calibration process. Furthermore a good depth map is typically time consuming so that it can not be used in real-time environment as *Automated Guided Vehicles* (AGV) and *Autonomous Mobile Robots* (AMR). This chapter is devoted to show a new approach for stereo matching in AMR and AGV applications. In this framework an accurate but slow reconstruction of the 3D scene is not needed; rather, it is more important to have a fast localization of the obstacles to avoid them. All the methods in the literature are based on a punctual correspondence, but they are inefficient in realistic contexts for the presence of uniform patterns, or some perturbations

between the two images of the stereo pair. Our idea is to face the stereo matching problem as a matching between homologous regions, instead of a point matching. We propose two different approaches: *graph based method* and *correlation based method*. The last section is devoted to explain as the disparity map (environment representation) can be used for moving object and obstacle detection aim.

5.2 Why a new approach

A pair of images acquired from a stereo camera implicitly contains depth information about the scene: this is the main assumption of stereo vision, based on the binocular parallax property of the human visual system. The main difficulty is to establish a correspondence between points of the two images representing the same point of the scene; this process is called *disparity matching*. The set of displacements between matched pixels is usually indicated as *disparity map*. The following figure presents a stereo pair from the Tsukuba data set.

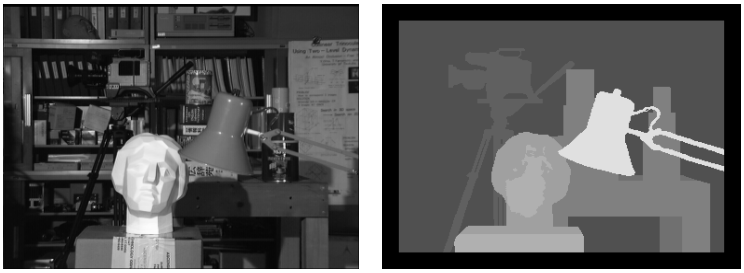


Figure 1: The reference image (on the left) and the ground truth of the disparity map (on the right), from Tsukuba data set. A point, in the disparity map, has a higher grey level (corresponding to a high disparity between the two images) the closer it is to the camera.

The second image in **Figure 1** represents the ground truth of the disparity map. An object has a higher grey level (corresponding to a high disparity between the two images) the closer it is to the camera, i.e. the lamp is in front of the statue that it in front of the table, etc.

The local area-based algorithms (see chapter 4) provide a correspondence for each pixel of the stereo pair. They produce a dense disparity map, redundant for AMR aims. Furthermore, they can be quite unreliable not only in homogeneous regions, but also in textured regions for an inappropriately chosen window size. On the other side, the global area-based approaches try to propagate disparity information from a pixel to its neighbors, so they have a better performance in homogeneous regions, but they frequently have parameters which are difficult to set, and are highly time-consuming. Finally, the feature-based approaches detect and match only "feature" pixels (as corner, edges, etc.). These methods produce efficient results, but compute sparse disparity maps (only in correspondence to the feature points). Therefore, AMR applications require more details, in fact some information about the size; also a rough shape of the objects is needed for guiding a robot in the environment or for basic recognition tasks (e.g. in industrial applications, or for platooning of robots).

All the proposed methods, as is clear, look for a punctual matching in the stereo pair. Therefore, some constraints both on the scene and on the input images have been introduced, since the first works on the stereopsis by Marr and Poggio [1,2], in order to guarantee good results and to reduce the complexity. To guarantee these constraints, the

stereo pair is supposed to be acquired from a sophisticated system, so that the energy distributions of the two images are as similar as possible. Moreover, a pre-processing phase is needed, before the correspondence finding step, to compensate the hardware setup (*calibration* phase), or to assume an horizontal epipolar line (epipolar *rectification*). Unfortunately, in realistic applications of mobile robot these constraints are not easy to guarantee. The two images of the stereo pair could have a different lighting, the motion of the mobile platform on a rough ground should produce mechanical vibrations of the cameras, and consequently local or global perturbations between the two images, that could undermine the initial phases of calibration and rectification.

We want to relax some constraints on the input images in order to consider a more realistic acquiring system, and consequently we add some constraints on our goal. We propose an extension of the disparity property, namely we define a disparity value for a whole region of the scene starting from the two homologous views of it in the stereo pair. The main reason of this extension is that a punctual approach is redundant for AMR and AGV applications. In fact, in this framework, it is not very important to have a good reconstruction of the surfaces, but it is more important to identify adequately the space occupied by each object in the scene (as soon as possible to avoid collisions), even by just assigning to it a single disparity information. Moreover the punctual approaches are lacking in robustness in some realistic frameworks, especially for video acquired from a mobile platform. Most of the algorithms available in off-the-shelf systems

[3,4] are unable to deal with large uniform regions or with vibration of the cameras. On the other hand, some efforts have been done in the literature to improve the robustness of the algorithms, but at the price of a significant increase of the running time. Our method estimates the average depth of the whole region by an integral measure, and so has fewer problems with uniform regions than other methods have. The estimate of the position of the regions is sufficiently accurate for navigation, also in the mentioned cases, and it is fast enough for real time processing.

5.3 The Rationale

In this thesis we propose, as said before, an extension of the disparity concept. The main idea is to determine a unique disparity value for a whole region of the scene and not for a pixel. In fact, even if we can suppose a unique correspondence between each pixel in the left and right images from an optical point of view (as said by the *uniqueness constraint*), in some cases we can not have enough information to find this correspondence looking just at a single pixel. Let us consider three kinds of situations:

Pixels inside homogeneous areas

As shown in **Figure 2**, it is a very hard task to compute the disparity value for a pixel inside a textureless region. In fact, the features-based algorithms are unable to find an appropriate feature in this case. The local area-based techniques must define a big correlation area in order

to pick enough information for the matching. Finally, the global area-based methods produce a propagation of the error depending on the energy minimization.

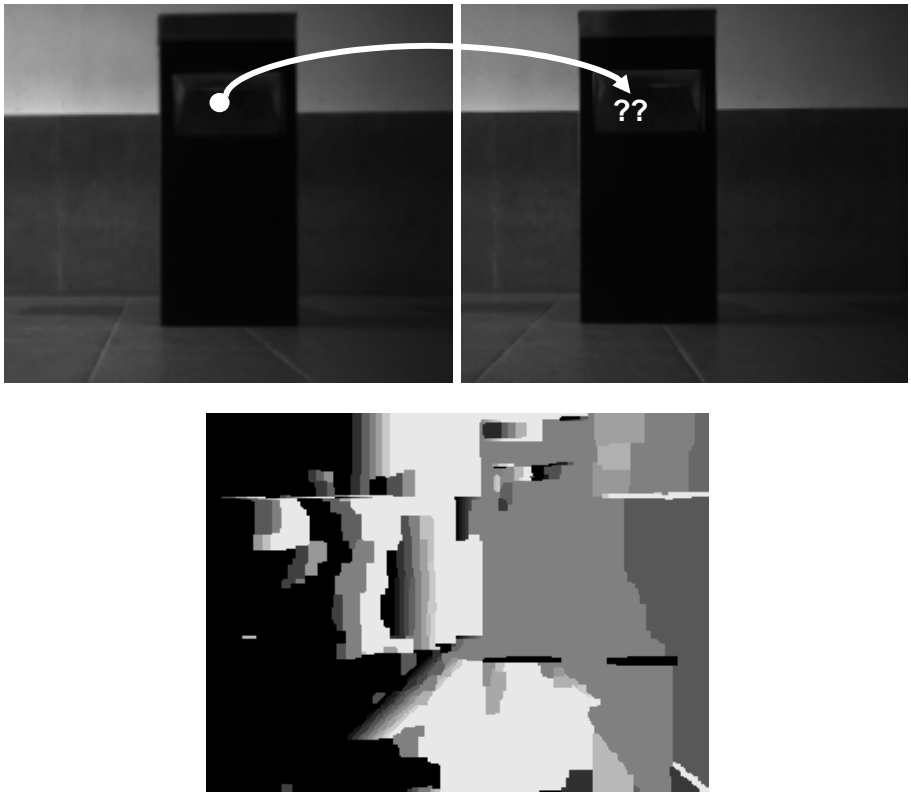


Figure 2: On the top of the figure a stereo pair with only one textureless object. On the bottom, the result of the algorithm by Boykov et al. [5], that is a global area-based method using graph cut. This algorithm produces a propagation of the error depending on the energy minimization.

Local and global perturbation of the stereo pair depending on the vibration of the mobile platform

The motion of the robot produces mechanical vibrations of the cameras with a consequent loss of the horizontal epipolar line constraint, which is assumed from all the methods in the literature.

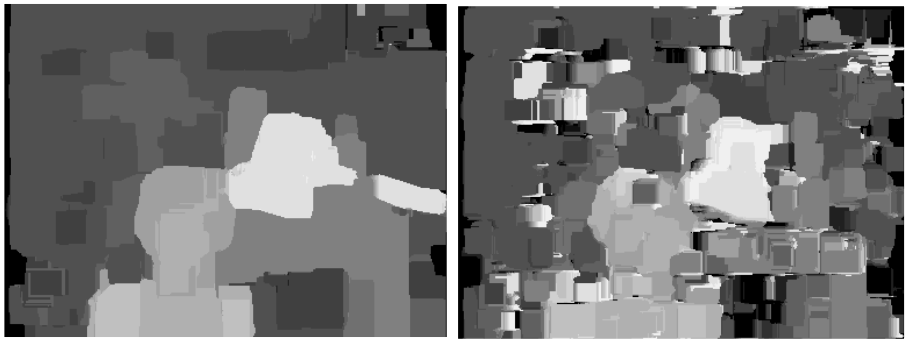


Figure 3: On the left side the result of the local area-based algorithm by Fusiello and Roberto [6] on a stereo pair with the horizontal epipolar line hypothesis. On the right side the result of the same algorithm after an horizontal misalignment of 2 pixels (upon 228 pixel of height) between the left and right images.

A different energy distribution between the left and right images

In a realistic framework the stereo pair could have some pixels suffering from perspective or photometric distortions, with a consequent loss of the compatibility constraint. Moreover, the two cameras could have different acquiring parameters, i.e. focus, or exposure, etc. In **Figure 4** there is an example of two images with different lighting. In ideal conditions all the pixels belonging to the same depth level have two energy patterns between the left and right images with a unique horizontal displacement (disparity value). In real condition (i.e. lighting differences) the two energy patterns are no longer a simple horizontal translation of each by one, consequently a punctual matching could be unsuitable.

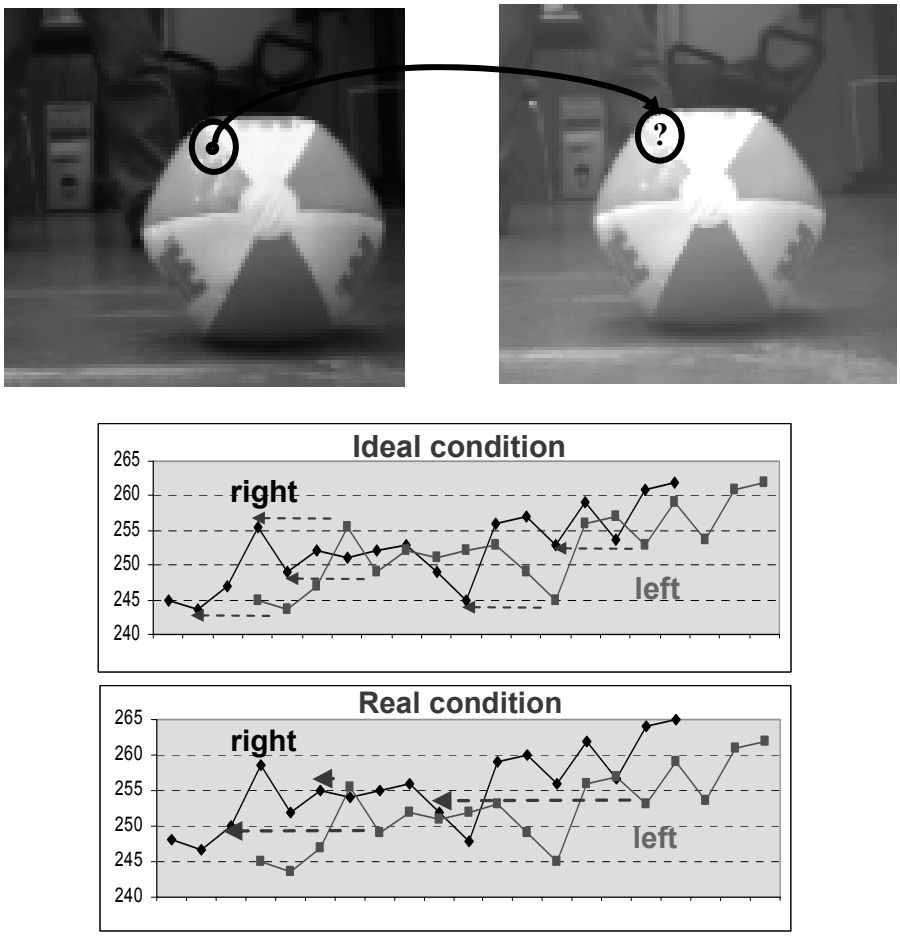


Figure 4: An example of a different energy distribution between the left and right images. On the top of the figure, the stereo pair. The two graphs show the energy pattern (for the selected area) in ideal and real conditions. In ideal conditions each pixel has the same horizontal displacement, instead in real conditions (a different lighting between the left and right images, as in the example) a vertical misalignment of energy causes the lost of the punctual correspondence.

A region-based algorithm is proposed to face up the limitations of the punctual stereo matching approaches. The corresponding entity is no longer the pixel, but a region; the matching of regions provides a lowering of resolution, but an increasing of robustness in a realistic environment. In fact, a uniform area is considered as a unique segment

for the matching, as like the local and global perturbations of the stereo pair less influence the solution. Finally, an integral matching, on a whole region, is able to mitigate the lack of homogeneity between the left and right images. Therefore, a good tread-off, between an efficient solution (to guarantee an autonomous navigation) and the robustness in a realistic framework, is investigated. Moreover, the real-time requirement is guaranteed.

5.4 A graph based definition

In this section, the *graph based method* is presented. The stereo matching is based on a region segmentation of the two images and a graph representation of these regions, to face the matching problem as a graph matching problem. The computational process is simple and fast, because we consider only some significant regions, i.e. big areas, or some areas selected by a specific target.

5.4.1 Overview of the method

The main idea of our approach is to obtain a disparity map looking at the distance between homologous regions (instead of pixels) in the stereo images. Let these regions be called *blobs*. In this way the computation of the disparity map is carried out on a set of pixels having the same spatial and color properties, producing a more robust performance with respect to local and global perturbations in the two images.

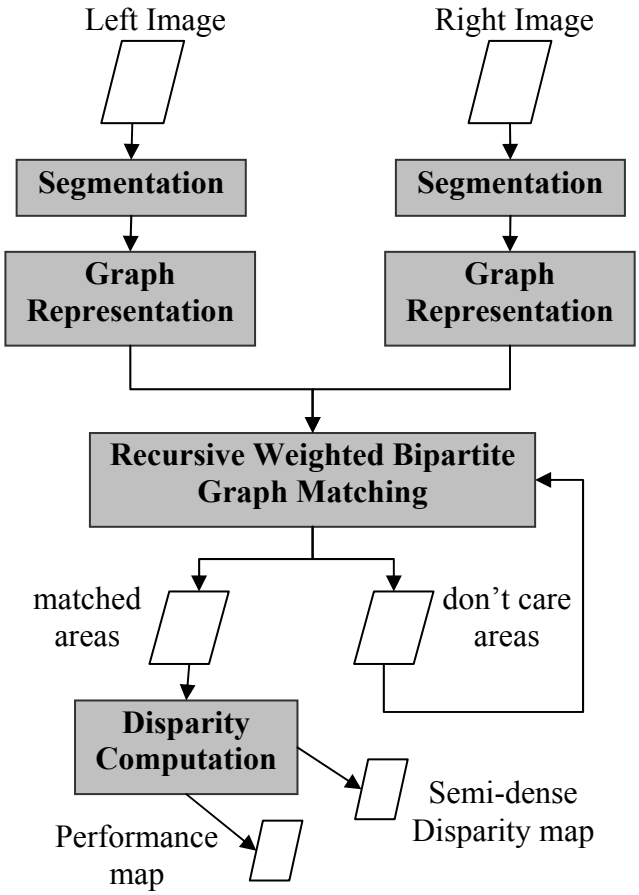


Figure 5: A scheme of our approach. The left and right images are segmented and each area identifies a node of a graph. A bipartite graph matching between the two graphs is computed in order to match each area of the left image with only one area of the right image. By calculating an horizontal displacement between the corresponding areas, a depth is found for those areas of the reference image (i.e. left image). The list of the don't care areas, instead, could be processed in order to refine the result.

It should be noted that a blob is not an object; objects are decomposed into several blobs, so the overall shape of the object is however reconstructed, except for uncommon pathological cases. An

example of pathological case can be a uniform object almost along the line of sight, but it has been satisfactorily dealt with only by global criteria optimization, which is extremely time consuming.

In our approach (see **Figure 5**), the left and right images are segmented and each area identifies a node of a graph. The segmentation process is simple and very fast. In fact, we are not interested in a fine segmentation, because we do not pursue a reconstruction aim. Anyway, we need similar segments between the left and right images in order to correctly find homologous regions. This objective is possible, in fact the stereo images are likely similar because they represent two different view points of the same scene. Moreover, the segmentation process does not influence the rest of algorithm, because a recursive definition of the matching and a performance function (see following) guarantee a recovery of some segmentation problems. A bipartite graph matching between the two graphs is computed in order to match each area of the left image with only one area of the right image. This process yields a list of reliably matched areas and a list of so-called *don't care* areas. By calculating an horizontal displacement between the corresponding areas, a depth is found for those areas of the reference image (i.e. left image). The list of the don't care areas, instead, could be processed in order to refine the result.

As it is clear, this approach is robust even in case of uniform texture and it does not need a strong calibration process because it looks for area correspondence and not pixel correspondence. On the other hand, an effort is required in graph matching to assure real-time

requirements. The application time is reduced using some constraints for a quicker computation of the bipartite graph matching. Our method can be classified as a systemic approach [7], in fact we consider constraints coming from the scene, from the objective and from the observer. In particular, with regard to scene constraints, we assume a strong continuity constraint for each selected region, and the compatibility and the uniqueness constraints are applied on the whole region and not longer on each pixel. The horizontal epipolar line constraint is generalized in a *horizontal epipolar band* (see 5.4.2), to take the nature of the mobile observer into account. Moreover, the observer is supposed to move in an indoor environment and not too fast. Finally, the objective is considered to be real-time and highly related to the AMR applications. Therefore, all these constraints are taken into account to achieve our goal.

5.4.2 The algorithm

The algorithm is composed of three phases: Segmentation and Graph representation, Graph Matching and Disparity Computation.

Segmentation and Graph representation

The first phase of the algorithm is the segmentation of the stereo images and their graph representation. We need a very fast segmentation process that produces similarly segmented areas between the left and right images. We have used a simple multi-threshold segmentation. It is essentially based on the quantization of

the histogram in some color ranges (of the same size). The left and right segmentations are very similar, considering an adaptive quantization for each image according to its lighting condition. A connected component detection procedure is applied on each segmented image to obtain 4-connected areas of the same color. Each connected area (blob) is then represented as a node of an attributed graph. Each node has the following attributes:

- ***colMean***: the RGB mean value of the blob (*m_r, m_g, m_b*);
- ***size***: the number of pixels in a connected area;
- ***coord***: the coordinates of the box containing the blob (*top, left, bottom, right*);
- ***blobMask***: a binary mask for the pixels belonging to the blob.

It is easy to understand that a segmentation yielding many segments can be more accurate but creates lots of nodes, consequently requiring a more expensive graph matching process. On the other hand, a rougher segmentation process generates matching nodes that are very dissimilar in size and shape. As a compromise, we consider a segmentation process tuned to over-segment the image, and subsequently we filter the image in order to discard small noisy areas.

Graph Matching

Formally our matching algorithm can be described in the following way. A number of nodes is identified in each frame (left and right images) and a progressive label is associated to each node (blob). Let $G^L = \{N_0^L,\ldots,N_n^L\}$ and $G^R = \{N_0^R,\ldots,N_m^R\}$ be the two graphs representing the left and right images respectively (region adjacency

graph). The solution of the spatial matching problem, between two stereo frames, is an injective mapping between a subset of G^L and a subset of G^R. The problem at hand can be represented by using a matrix whose rows and columns are respectively used to represent the nodes of the set G^L, and the nodes of the set G^R (correspondence matrix). The element (i,j) of the matrix is 1 if we have a matching between the element N_i^L with the element N_j^R, it is 0 otherwise. Each row contains no more than one value set to 1. If the j-th row or the i-th column contains only zeros, it means that it is a don't care node. The bijective mapping τ: $G^L \rightarrow G^R$ solves a suitable Weighted Bipartite Graph Matching (WBGM) problem. A Bipartite Graph (BG) [8] is a graph where nodes can be divided into two sets such that no edge connects nodes in the same set. In our problem, the first set is G^L, while the second set is G^R. Before the correspondence is determined, each node of the set G^L is connected with each node of the set G^R, thus obtaining a Complete BG. In general, an assignment between two sets G^L and G^R is any subset of $G^L \times G^R$, i.e., any set of ordered pairs whose first elements belongs to G^L and whose second elements belongs to G^R, with the constraint that each node may appear at most once in the set. A maximal assignment, i.e. an assignment containing a maximal number of ordered pairs is known as a matching (BGM) [9]. A cost function is then introduced, so that each edge (N_i^L, N_j^R) of the complete bipartite graph is assigned a cost. This cost takes into account how similar are the two nodes N_i^L and N_j^R. The lower is the cost, the more suitable is that edge. If the cost of an edge is higher than a threshold (thrMatch), the edge is considered unprofitable and is

removed from the graph (its cost is considered to be ∞).

Let us now introduce the cost function:

$$Cost = \frac{colCost + d\,i\,mCost + posCost}{3} \quad \textbf{(1)}$$

Where:

$$colCost = \frac{\sum_{i \in \{m\_r, m\_g, m\_b\}} \left| colMean_i^L - colMean_i^R \right|}{3 * 256}$$

$$d\,i\,mCost = \frac{\sum_{\substack{i \in \{bottom, right\} \\ j \in \{top, left\}}} \left| (i^L - j^L) - (i^R - j^R) \right|}{width + height}$$

$$posCost = \frac{\sum_{i \in \{bottom, right, top, left\}} \left| i^L - i^R \right|}{2 * (width + height)}$$

where *width* and *height* are the dimensions of the frame. The matching with the lowest cost among the ones with maximal cardinality is selected as the best solution. The problem of computing a matching having minimum cost is called Weighted BGM (WBGM). This operation is generally time-consuming; for this reason the search area (that is the subset of possible couples of nodes) is bounded by the ***epipolar*** and ***disparity bands*** (see **Figure 6**).

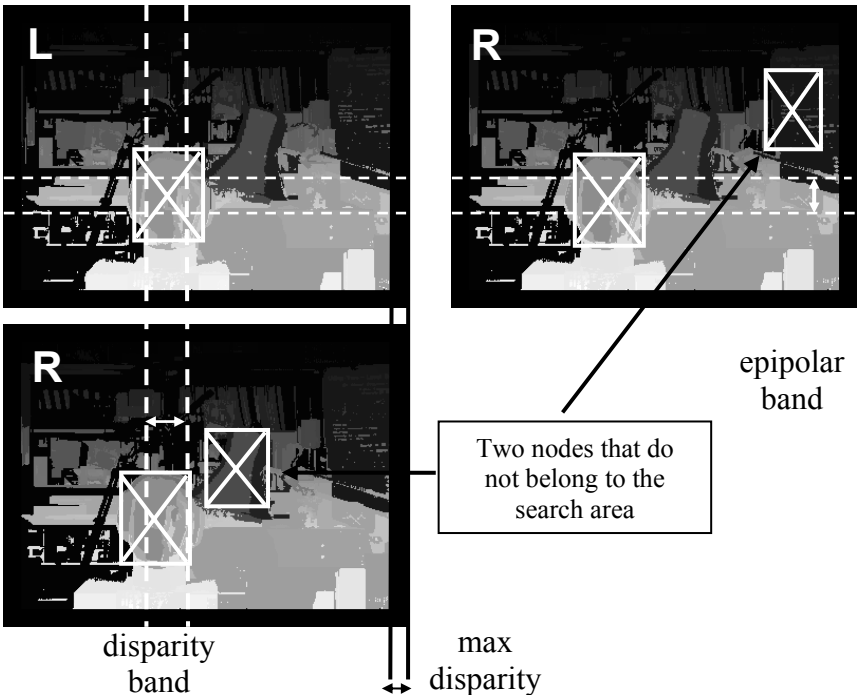


Figure 6: Epipolar and disparity bands: some constraints to optimize the WBGM. The epipolar band is a generalization for epipolar line, that is the maximum vertical displacement of two corresponding nodes. Disparity band, instead, is an horizontal displacement related to the maximum value of disparity.

These constraints come from stereo vision geometry, but in our case they represent a generalization. The epipolar band is a generalization for epipolar line, that is the maximum vertical displacement of two corresponding nodes (generally its value can be a few pixels). Disparity band, instead, is an horizontal displacement, so a node of the right image can move on the left almost of α*maxdisparity pixels (with α a small integer). These two displacements are computed with respect to the centers of the bounding box of the two blobs.

The graph matching process yields a list of reliably matched areas and a list of so-called *don't care* areas. The matched areas are

considered in the following section for the disparity computation. The list of the don't care areas, instead, is processed in order to group adjacent blobs in the stereo pair and consequently reduce split and merge artifacts of the segmentation process. Finally, a new matching of these nodes is found. The recursive definition of this phase assures a reduction of the don't care areas in few steps, but sometimes this process is not needed because don't care areas are very small.

Disparity Computation

The disparity computation is faced superimposing the corresponding nodes until the maximum covering occurs. The overlapping is obtained moving the bounding box of the smallest region into the bounding box of the largest one; precisely, the bounding box with the minimum width is moved horizontally into the other box, and the bounding box with the minimum height is moved vertically into the other box. The horizontal displacement, corresponding to the best fitting of the matched nodes, is the disparity value for the node in the reference image (left image).

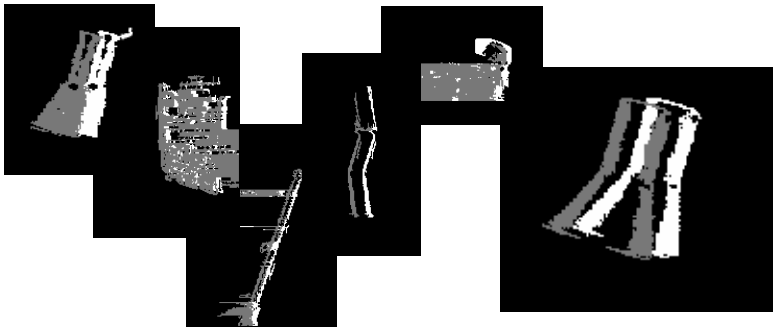


Figure 7: Some examples of matched regions. In grey color the region from the right image and in white color the region from the left image.

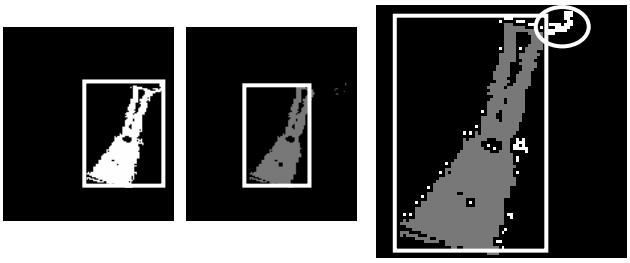


Figure 8: The overlapping process minimizes the mismatching between the two matched regions. On the right side, it is shown an appendix depending on the different segmentation between left and right images.

A lot of objects have some appendices (see **Figure 8**) depending on the different segmentation between left and right images. However, this process finds the correct value for the disparity, minimizing the mismatching between the two matched regions. Moreover, we propose a performance measurement for the disparity computation in order to consider also some cases with larger errors coming from both segmentation and matching process.

$$performance = \frac{\max Fitting}{\max(sizeL, sizeR)} \qquad (2)$$

It is the percentage value of the best fitting area size (*maxFitting*) with respect to the maximum size of the two matched regions (*sizeL* and *sizeR*).

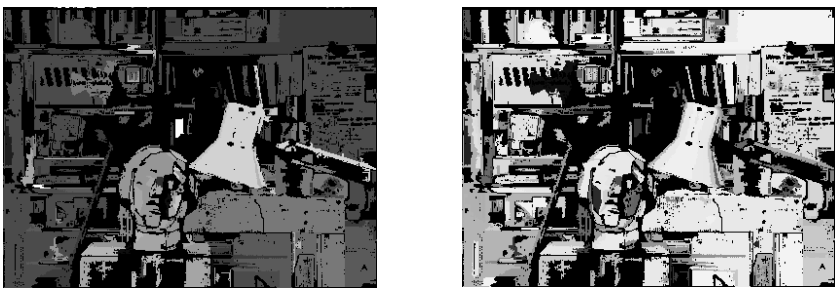


Figure 9: On the left side our disparity map; on the right side a graphical representation of the performance function (a brighter region has the upper value of performance).

The result of our algorithm can be represented in a graph, the so-called *disparity graph*, and, as it is clear from the **Figure 9**, the nodes of this graph can have a *don't care* attribute or, alternatively, the couple of disparity and performance attributes. Therefore, we could select a minimum performance value, and label the regions below this value as don't care. All these don't care areas could be processed again in the WBGM, as said in section 5.4.1, if we should need to refine the result. Anyway, in our experimental results, we use a simple post-filtering in order to reduce don't care regions. Each 4-connected don't care area is labeled choosing the most frequent among the disparities of the adjacent regions. This assumption comes from the continuity constraint, but it is clear that it is applicable only inside a region and not between two different regions, so it is checked that most of the adjacent regions have the same disparity value. An example of the post-filtering use is shown in the **Figure 10**.

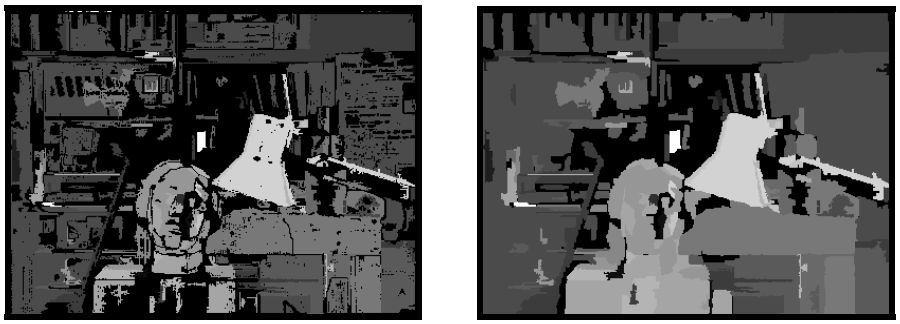


Figure 10: On the left side the original disparity map; on the right side the result after applying the post-filtering.

5.5 A correlation based definition

The graph based method has a lot of interesting features. For example, it completely avoids the strong epipolar constraint and this makes it suitable for AMR applications. On the other hand, some problems arise with this approach, especially in the segmentation phase of the algorithm. As it was explained before, the segmentation algorithm must be as fast as possible (in order to be suitable for real time applications) but it also has to ensure a "symmetric" segmentation in both the reference and sensed images. By "symmetric" I mean a segmentation that gives in the two images, as output, the same number of blobs which share the same semantics of the real scene. For this reason in this section we propose an other algorithm that starts from the same motivations of the graph based method, but overcoming its limitations.

5.5.1 Overview of the method

Let us consider R_l and R_r, as the projection of the region R of the scene into the stereo pair $\{I_l , I_r\}$ (see **Figure 11**). As shown in the figure, we assume the well-known pinhole model. It is composed of an *image plane* (*I*), also called *retina plane*, and of an optical centre (*C*), spaced *f* (*focal length*) from the plane. The line passing through C point and orthogonal to R is called *optical line*.

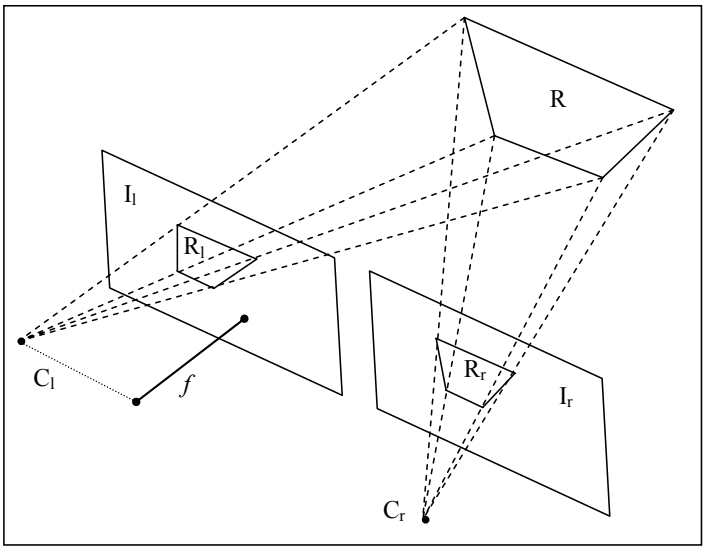


Figure 11: Projection of the region R in the stereo pair

Some projection errors could occur. The homologous regions can suffer from perspective distortion, as bigger as further are the cameras each by one, or photometric distortion, because of no perfectly lambertian surfaces. The mechanical vibration of the mobile platform can introduce other errors, undermining the epipolar rectification. Finally, the digitalization process (also depending on the acquiring parameters) can produce border errors. We determine the disparity value for the whole region as the horizontal displacement between the regions. The detection of the homologous regions is, of course, a difficult problem. In fact, a same segmentation method, separately applied on the left and right images, should divide in different parts the same region of the scene, or should produce border errors, undermining a correct detection of the disparity for the whole region. In our algorithm a segmentation of the left image (reference image) is performed and each region of the reference image, selected from each

segment, is overlapped on the sensed image (right image). The disparity value of the region is the horizontal displacement, corresponding to the minimization of a *best fitting function* between the two regions. This integral measurement of the disparity can mitigate some null integral border errors, as segmentation, digitalization, and photometric errors. An approximation can be obtained for the border errors from perspective distortion, that is not right with null integral.

5.5.2 The algorithm

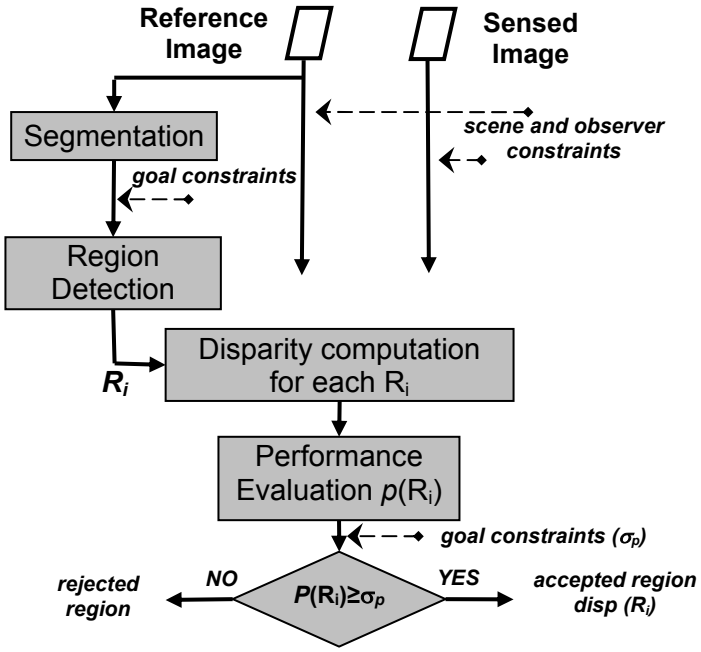


Figure 12: A schema for our algorithm of region-based stereo matching. The segmentation of the reference image is performed in order to detect interesting regions in the image. Each segment is used as a selection mask on the left and right images in order to select the homologous regions. The disparity is the horizontal displacement corresponding to the best fitting of the homologous regions. A region is rejected if the performance index is lower than an imposed tolerance ($P(R)<\sigma_p$).

As shown in **Figure 12** the algorithm is composed of four steps:

Segmentation of the reference image

Several segmentation methods (mean shift, pyramid, multi-threshold) have been tested. The algorithm has a similar behavior towards all the methods, taking care not to under segment the image. In fact, an under segmentation could merge regions belonging to different depth level. The over segmentation has not a big influence in our method, because the best fitting function is enough accurate. Anyway, a multi-threshold segmentation method has been used in the algorithm. It is essentially based on the quantization of the histogram in some color ranges (of the same size).

Region Detection

A connected component analysis is performed to detect connected segments. Looking at the experimental results, a 4-connected analysis has been enough for our aim. This step is also devoted to select a subset of regions among all. The selection is made using some constraints on the goal (*goal constraints*). Namely, a minimal knowledge about the obstacle (i.e. the maximum size of an obstacle is an upper-bound for the maximum size of a region; color information if any) or the desired resolution of the result (i.e. the minimum size of the region). In this way, the computation time can be reduced.

Disparity Computation

Each segment from Region Detection step is used as a selection mask

on the left and right images in order to select the homologous regions. The selection of the right region is displaced from 0 to the maximum value of disparity. The disparity is the horizontal displacement corresponding to the best fitting of the homologous regions. Formally, let $E_L(x,y)$ be the energy value for each pixel (x,y) on the left image, and $E_R(x,y)$ be the energy distribution of the right image. Let $G_L(x,y)$ and $G_R(x,y)$ be the gradient map of the left and right images. Finally, let R_i be the generic segment from step 2, the following equations are defined:

$$d(R_i) = \arg\min_{0 \le d \le d_{\max}} (\varepsilon_i(d)) \tag{3}$$

$$\varepsilon_i(d) = \alpha \cdot \varepsilon_i^{col}(d) + \beta \cdot \varepsilon_i^{grad}(d)$$

The best fitting function uses color and gradient information of the homologous regions, in order to consider the energy distribution of pixels inside each region and also texture information. The values for the weights α and β are experimentally found.

$$\varepsilon_i^{col}(d) = \frac{1}{|R_i|} \sum_{(x,y) \in R_i} \left| \left(E_L(x,y) - \mu_i^L\right) - \left(E_R(x-d,y) - \mu_i^R(d)\right) \right| \tag{4}$$

where:

$$\mu_i^L = \frac{1}{|R_i|} \sum_{(x,y) \in R_i} E_L(x,y) \qquad \mu_i^R(d) = \frac{1}{|R_i|} \sum_{(x,y) \in R_i} E_R(x-d,y)$$

$$\varepsilon_i^{grad}(d) = \frac{1}{|R_i|} \sum_{(x,y) \in R_i} |G_L(x,y) - G_R(x-d,y)|$$

The best fitting color function, $\varepsilon_i^{col}(d)$, is normalized on the mean color

(μ_i^L , μ_i^R) of the region R_i on the left and right images. In this way a good matching is found also in case of a no homogeneous distribution of energy between the left and right images.

Performance Evaluation

The previous step provides the disparity value for each region R_i, but also a performance index for the matching, $p(R_i)$. In fact, for each region the minimum value of the fitting function has been used as matching error, $\varepsilon(R_i)$, and the reliability index is:

$$p(R_i) = 1 - \frac{\varepsilon(R_i)}{\max_{R_i}(p(R_i))} \qquad (5)$$

where :

$$\varepsilon(R_i) = \min_{0 \le d \le d_{\max}} (\varepsilon_i(d))$$

A region is rejected if the performance index is lower than an imposed tolerance ($p(R)<\sigma_p$). This is an other goal constraint because we can choose a reliability level depending on the requested efficacy of the solution. Therefore the disparity map is a semi-dense map with some *don't care* regions.

As said in chapter 3, our method can be classified as a systemic approach [7], in fact we consider constraints coming from the scene, from the goal and from the observer (as shown in **Figure 12**). In particular, with regard to scene constraints, we assume a strong continuity constraint for each selected region, and the compatibility and the uniqueness constraints are applied on the whole region and not

longer on each pixel. Moreover, the observer (mobile platform) moves slowly so that only little vibrations of the cameras are possible. Finally, the obstacle detection task (our goal) is scalable in time and performance: a robot, having more time, can carry out a finer investigation of the environment, asking to the system a better solution.

5.6 Moving Object and Obstacle Detection

What is an obstacle for a mobile platform? What about moving objects? A motionless obstacle can be identified as a connected regions that belong to a chosen range of distances, in fact an obstacle is an object so close to the mobile platform to forbid the navigation. Therefore a good 2D ½ representation of the scene can be enough to detect motionless obstacles and to suggest a safety path for the navigation.

Stereo vision can provide an adequately accurate 2D ½ map of a scene, but does not produce an estimate of the trajectories of the objects in the scene, which is important if those objects are to be followed or avoided. For this reason a robot vision system must include a moving obstacle detection phase (as it is shown in the whole system, see chapter 3). As regards moving object detection, in the literature there are three main approaches: temporal differencing [10], background subtraction [11], and optical flow approaches [12,13,14].

The first two approaches are best suited to a fixed camera hypothesis. On the other hand, an optical flow approach can be

suitable for either a fixed camera hypothesis or a moving camera in a fixed scene hypothesis. We propose a system in which optical flow is combined with the disparity information for determining both object and robot motion. The main advantage is that the method works in contexts with a moving camera in a scene with multiple moving objects whose shape is not known a priori allowing the determination of the trajectory of these objects.

5.6.1 The Entire System

In AGV and AMR applications the scene cannot be simply segmented into a static background with moving foreground objects. The system must be able to detect both *autonomous moving objects* and *motionless objects*, which have to be considered in a different way. The problem is difficult because the camera is moving in the 3D environment, so the motion vectors alone are not sufficient to recognize the moving objects. However, using together the optical flow and the disparity map it is possible to separate the motion of the camera (ego-motion) from the motion of the objects. An overview of our system for Moving Object and Obstacle Detection (MOOD) is shown in **Figure 13**.

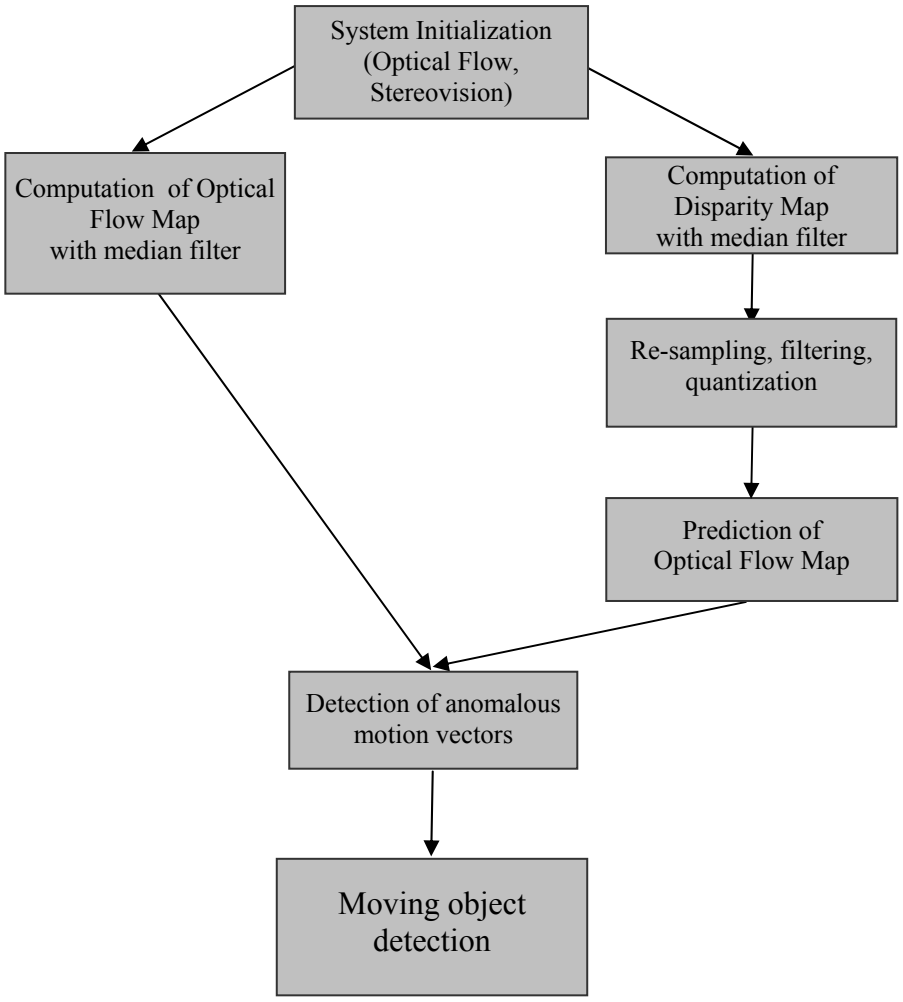


Figure 13: Our Moving Object and Obstacle Detection System (MOODs). Using together the optical flow and the disparity map it is possible to separate the motion of the camera (ego-motion) from the motion of the objects.

The first step is a re-sampling and a quantization of the disparity map (computed according to our algorithm - see 5.5.2). This step is needed to respect the resolution chosen for the solution (the same resolution is used for the optical flow). A median filtering is performed before the quantization.

In order to estimate the motion of the object in the scene, we use

the optical flow computed with an area-based algorithm [15]. Namely, given two images $I_1(x,y)$ and $I_2(x,y)$ captured in consecutive frames, the motion vector (v_x , v_y) of point (x, y) is obtained minimizing the function (*Sum of Absolute Differences -SAD*):

$$SAD(x, y, v_x, v_y) = \sum_{i,j} \left| I_1(x+i, y+j) - I_2(x+i+v_x, y+j+v_y) \right| \quad (6)$$

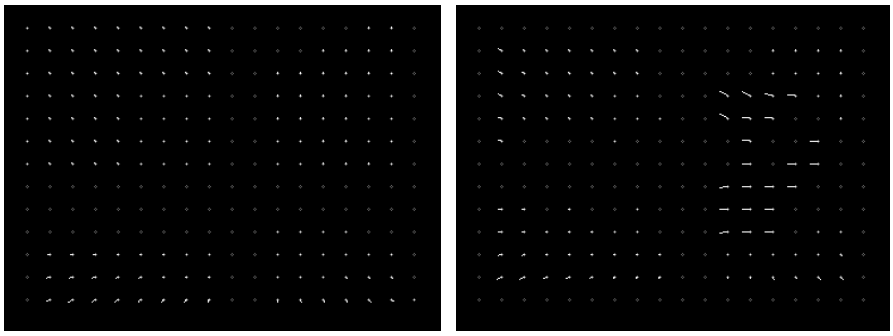
where $-m/2 \leq i \leq m/2$; $-m/2 \leq j \leq m/2$ and m is the size of the correlation window. Actually, to make this computation more robust we average the value of SAD over three adjacent frames. In the minimization phase we use a threshold on the SAD value to reject spurious motion vectors. Furthermore, we perform a quantization of (v_x , v_y) to reduce the computational cost. We have used a median filter, in post-processing phase, to attenuate local differences of the vectors, after Horn and Schunk [13] hypothesis of space continuity of the optical flow. At the end, *sharpness* and *distinctiveness* constraints are also enforced for disambiguating the minimum [16, 17].

Therefore, combining optical flow with the disparity information we can detect moving objects. In fact, assuming that the velocity vector of the camera is known, we can predict the displacement (between adjacent frames) of each point in the disparity map under the hypothesis that the point is motionless. This displacement is a 3D vector; the projection of this vector on the image plane gives a prediction of the motion vector in the optical flow. Of course, if the point belongs to a moving object, its observed motion vector (v_x , v_y)

will differ from the predicted one *(v'x , v'y)*. We compute the difference of the modules and phases between these two vectors and compare them with a module threshold and a phase threshold.

$$\left.\begin{array}{c} m=\sqrt{v_x{}^2+v_y{}^2} \quad m'=\sqrt{v_x{}'^2+v_y{}'^2} \\ \varphi=tg^{-1}\dfrac{v_y}{v_x};\ \varphi'=tg^{-1}\dfrac{v_y{}'}{v_x{}'}\Rightarrow\theta=\cos^{-1}\left(\dfrac{<\vec{v},\vec{v}'>}{|x|\cdot|y|}\right) \end{array}\right\} \begin{array}{l} |m-m'|\geq M\_Threshold \\ \theta\geq D\_Threshold \end{array} \tag{7}$$

The points that exceed one of the thresholds are marked as *anomalous motion vectors* and are used for the last step, *moving object detection*. This step uses a standard algorithm for detecting the connected components in an image. Each component is considered a detected moving object and described by means of its bounding box (see **Figure 14**).



a) b)

c)

Figure 14: Some output of our system: a) *Predicted Optical Flow*; b) *Observed Optical Flow*; c) *Moving Obstacle Detection*.

Some detected moving objects are discarded by a post-processing filter, that evaluates constraints based on object size, uniformity, distance and position with respect to the ground level [18].

References

[1] D. Marr and T.A. Poggio, "Cooperative computation of stereo disparity", Science, vol. 194, no. 4262, pp. 283–287, 1976.

[2] D. Marr and T.A. Poggio, "A computational theory of human stereo vision" RoyalP, vol. B, no. 204, pp. 301–328, 1979.

[3] K. Konolige, "Web site", http://www.ai.sri.com/software/SVS, 2006.

[4] K. Konolige, "Small vision systems: hardware and implementation", in Proceedings of the International Symposium On Robotics Research, pp. 111–116, 1997.

[5] Y. Boykov, O. Veksler and R. Zabih, "Fast approximate energy minimization via graph cuts", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222–1239, 2001.

[6] A. Fusiello and V. Roberto, "Efficient stereo with multiple windowing", in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 858–863, 1997.

[7] J.M. Jolion, "Computer Vision Methodologies", CVGIP: Image Understanding, vol. 59, no. 1, pp. 53–71, January 1994.

[8] H. Baier and C. L. Lucchesi, "Matching Algorithms for Bipartite Graphs", Technical Report DCC-03/93, DCC-IMECC-UNICAMP, Brazil, March 1993.

[9] H.W. Kuhn, "The Hungarian Method for the Assignment Problem", Naval Research Logistics Quarterly, vol. 2, pp. 83-97, 1955.

[10] C. Anderson, P. Burt and G. van der Wal, "Change detection and

tracking using pyramid transformation techniques", in Proceedings of SPIE Intelligent Robots and Computer Vision, vol. 579, pp. 72-78. 1985.

[11] C. Stauffer and W.E.L. Grimson, "Learning patterns of activity using real-time tracking", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 22, pp. 747-757, 2000.

[12] G. Halevy and D. Weinshall, "Motion of disturbances: Detection and tracking of multi-body nonrigid motion", Machine Vision Application, vol. 11-3, pp. 122–137, 1999.

[13] B. Horn and P. Schunk, "Determining Optical Flow", Artificial Intelligence, vol. 17, pp. 185-203, 1981.

[14] K.T. Song and J. H. Huang, "Fast Optical Flow Estimation and Its Application to Real-time Obstacle Avoidance", in Proceedings of IEEE International Conference on Robotics and Automation, pp. 2891-2896, 2001.

[15] C. Zhang, "A Survey on Stereo Vision for Mobile Robots", Technical report, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213, USA, 2002.

[16] L. Di Stefano, M. Marchionni and S. Mattoccia: "A Fast Area-Based Stereo Matching Algorithm", Image and Vision Computing, vol. 22, no. 12, pp. 983-1005, October 2004.

[17] J. H. Wang, R. F. Hsieh and H. C. Chiu, "A progressive constraint search approach for disparity matching in stereo vision", in Proceedings of the National Science Council - Part A, vol. 23, no. 6, pp. 789-798, 1999.

[18] S. Rougeaux and Y. Kuniyoshi, “Velocity and Disparity Cues for Robust Real-Time Binocular Tracking”, in Proceedings of IEEE Computer Vision and Pattern Recognition, pp. 1-6, 1997.

Chapter 6

Experimental Results

"... Now a new tempest – of passion and flame and thunder – dispels
The dark sky of the heart, that tender and tired of fighting,
Flops in the quiet tremor of a shaking harmonious warmth. ..."
"...Tempesta d'altra natura – di passione e fiamme e tuoni – squarcia ora
il cielo buio del cuore che tenero e stanco di lottare
si abbandona al tremore calmo di uno sconvolgente armonico tepore. ..."
A.L.

6.1 Introduction

In this chapter, we report the experimental results of our method for moving object and obstacle detection (that we have called MOOD system). The following section is devoted to the obstacle detection using stereo vision paradigm. The results of our method for stereo matching are shown in a comparison with the best algorithms in the literature. It is also proposed a quantitative measurement for performance evaluation, with a reference to our specific goal of the obstacle detection in autonomous navigation framework. The last section presents the results of the moving object detection. A synthetic database has been created to precisely evaluate our system.

6.2 The Obstacle Detection: The Results

In the literature, the tests for stereo matching algorithms are usually performed with standard databases composed of static images, well-calibrated and acquired in uniform lighting. The Middlebury web site by Scharstein and Szeliski [1] is a good reference for some stereo images and to compare some stereovision algorithms.

In this section we want to show our qualitative results and discuss some errors of the best algorithms in the literature, when applied to real cases. Nowadays, in AMR and AGV applications it is not defined a quantitative measurement for performance evaluation. It is proposed a quantitative performance evaluation for disparity map by Scharstein and Szeliski [2], but in case of reconstruction aims. For this reason in this thesis, it is also proposed a quantitative method to compare stereo algorithms when the goal is the obstacle detection and no longer the 3D reconstruction of the scene.

We have reported the results of the correlation based algorithm because more accurate and efficient than the graph based algorithm, in a lot of cases.

The following figure presents a stereo pair from the Tsukuba data set.

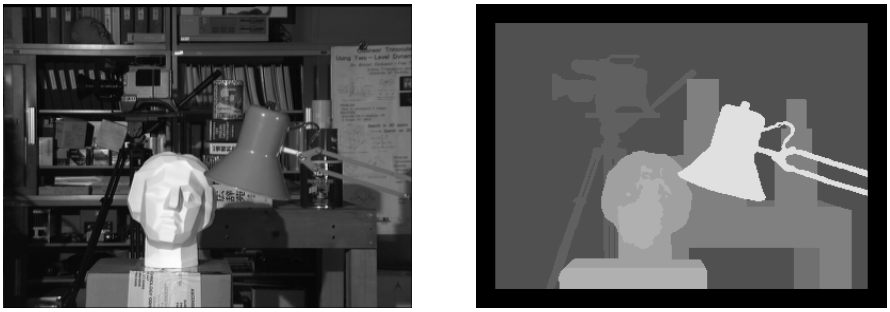
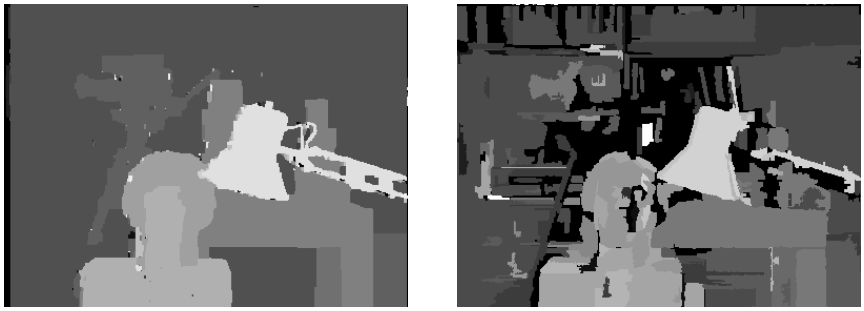


Figure 1: The reference image (on the left) and the ground truth of the disparity map (on the right), from Tsukuba data set. A point, in the disparity map, has a higher grey level (corresponding to a high disparity between the two images) the closer it is to the camera.

The second image in **Figure 1** represents the ground truth of the disparity map. An object has a higher grey level (corresponding to a high disparity between the two images) the closer it is to the camera, i.e. the lamp is in front of the statue that it in front of the table, etc. The following **Figure 2** shows our result on the Tsukuba DB and a comparison with other approaches. We have selected the best methods in the literature: squared differences (SSD), dynamic programming (DP) and graph cuts (GC) [**2**]. The first is a local area-based algorithm, the other two ones are global area-based algorithms. The experiments have been performed on a notebook Intel P4 1.5 GHz, 512 Mb RAM, and we have considered a resolution of 384x288 pixels.

SSD: Time < 1 sec DP: Time 2 sec



Graph Cut: Time 70 **OUR: Time 1.14**

Figure 2: A comparison with other approaches.

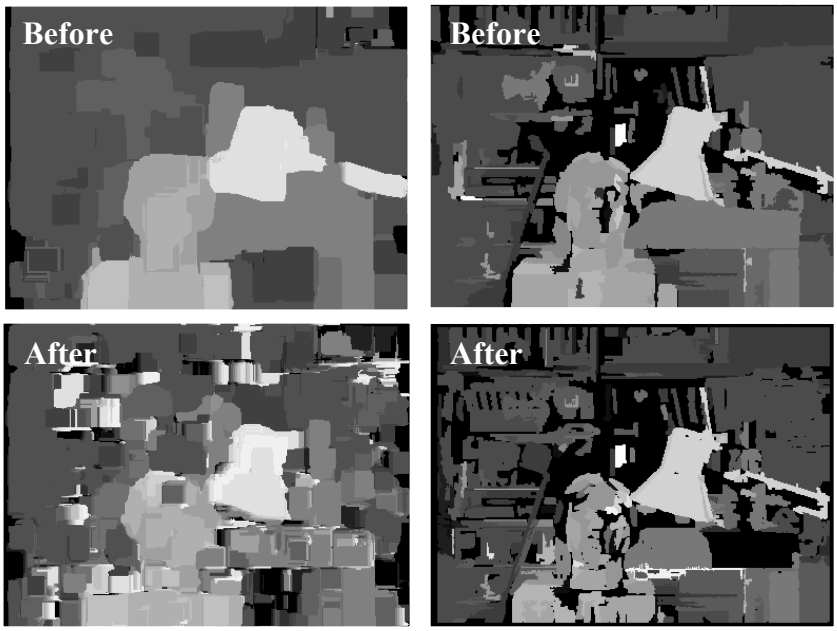
The real-time requirement is guaranteed, in fact our execution time is comparable to the SSD algorithm that is the most used in real-time context.

We have used the following parameters and constraints (see the algorithm in section 5.5.2):

Table 1: Parameters and constraints

Description	**Value**
Scene and observer constraints	Respected
Numbers of ranges for segmentation	20
Goal constraint for region detection	Not used
$[\alpha , \beta]$ for disparity computation	[0.4, 0.6]
Threshold for performance evaluation (σ_p)	0.8

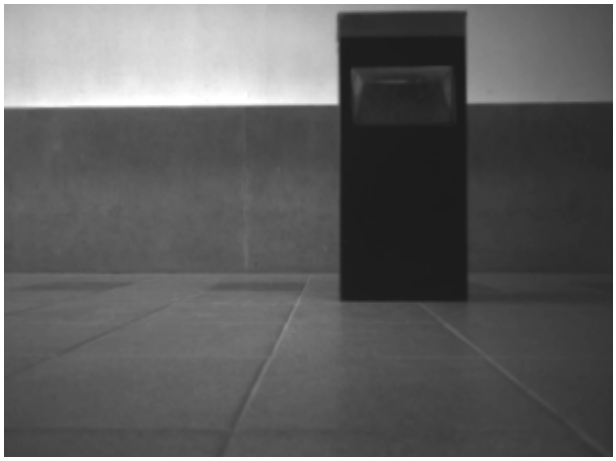
The goal constraint for region detection is not used in order to not compromise the comparison with the punctual approaches (that can't use such constraint). The parameters are obtained by experimental evidences.

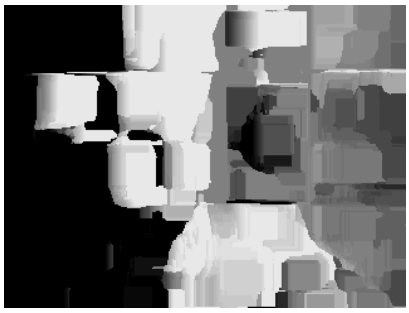


SSD: Time < 1 sec **OUR: Time 1.14 sec**

Figure 3: SSD and Our approach after a vertical translation of 2 pixels.

In **Figure 3** it is clear the robustness of our approach in relation to the loss of the horizontal epipolar constraint (we have imposed a vertical translation of 2 pixels between the left and right images). The SSD algorithm has a lot of false detections, in fact some pixels are labeled with an higher disparity and other ones with a lower disparity. Our result is more robust, in fact we have just little changes in don't care pixels.





SSD: Time < 1 sec

DP: Time 2 sec

Graph Cut: Time 50 sec

OUR: Time 0.7 sec

Figure 4: Results on our stereo pair: it is characterized by only one homogeneous object.

The presence of texture-less regions (very frequent in real contexts)

causes serious problems to the best algorithms of the literature as shown in **Figure 4**.

In order to consider a quantitative comparison of the algorithms for obstacle detection aim, we define a simple module that detects the obstacles from the disparity map. Each 4-connected region with the same disparity value is identified with a bounding box and its distance from the observer. We select the obstacles as the connected regions that belong to a chosen range of distances, in fact an obstacle is an object so close to the mobile platform to forbid the navigation. Therefore two performance index are defined in order to valuate: the capability of the algorithm to identify adequately the space occupied by each obstacle (*occupancy performance*); the correctness of depth computation for each obstacle (*distance performance*). For each frame of the video sequence acquired from the platform, let R_G be the real obstacle regions (*Ground Truth*), let R_D be the obstacle regions detected by the algorithm, and let R_I be the subset of regions correctly detected as obstacles by the algorithm ($R_I = R_G \cap R_D$). The occupancy performance is evaluated with the measures of *precision* and *recall*:

$$recall = \frac{R_I}{R_G} \qquad precision = \frac{R_I}{R_D} \tag{1}$$

The distance performance is evaluated with the following *relative distance error (rde)*:

$$rde = \frac{\left| \text{detected distance - real distance} \right|}{\text{real distance}} \tag{2}$$

The distance of an obstacle is related to its disparity value following the relation:

$$\text{distance} = k_{px/m} \frac{baseline \cdot focal\ \ lenght}{disparity} \quad (3)$$

where $k_{px/m}$ is the conversion factor from pixel to meter. It should be noted that for each real obstacle (*Ground Truth*) could be more than one overlapped obstacle regions detected by the algorithm. The detected distance for that obstacle is supposed to be a weighted mean distance of all the overlapped regions. The weights are set up to the sizes of each overlapping area. We report some results obtained on a realistic video acquired from our mobile platform. The video sequence (100 frames) is characterized by camera vibration, light changing, uniform obstacles (see **Figure 5**).

Figure 5: Some frames of the video sequence.

The proposed method is compared with the Small Vision System (SVS) by Konolige [4,5] that is the most popular system in off-the-shelf systems. Namely, the SSD stereo matching algorithm has been implemented in SVS, taking care the real-time requirement and filtering the solution to reject false stereo matches. We consider two different version of that algorithm: *SSD* and *SSD multi-scale*.

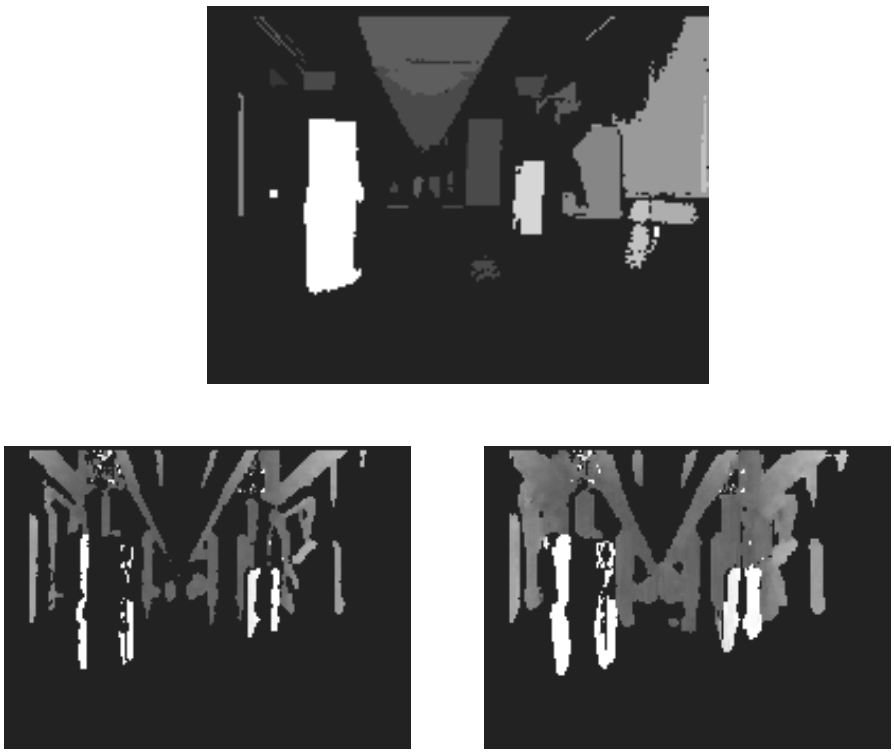


Figure 6: Disparity Map Results: On the top our method, on bottom left the SSD, and on bottom right SSD multi-scale .

Table 2: Precision and Recall

algorithm	recall	precision
our method	**0.91**	**0.63**
SSD	0.21	0.48
SSD multi-scale	0.45	0.35

Table 3: Relative Distance Error

algorithm	relative distance error
our method	**0.11**
SSD	0.19
SSD multi-scale	0.18

The results in the previous tables show that our method is much better than the other two, especially for the occupancy performance. In fact, as it is clear from **Figure 7** and **Figure 8**, our approach can better overlap the space occupied by the real obstacles, as like the SSD multi-scale algorithm has big opening areas inside the obstacles.

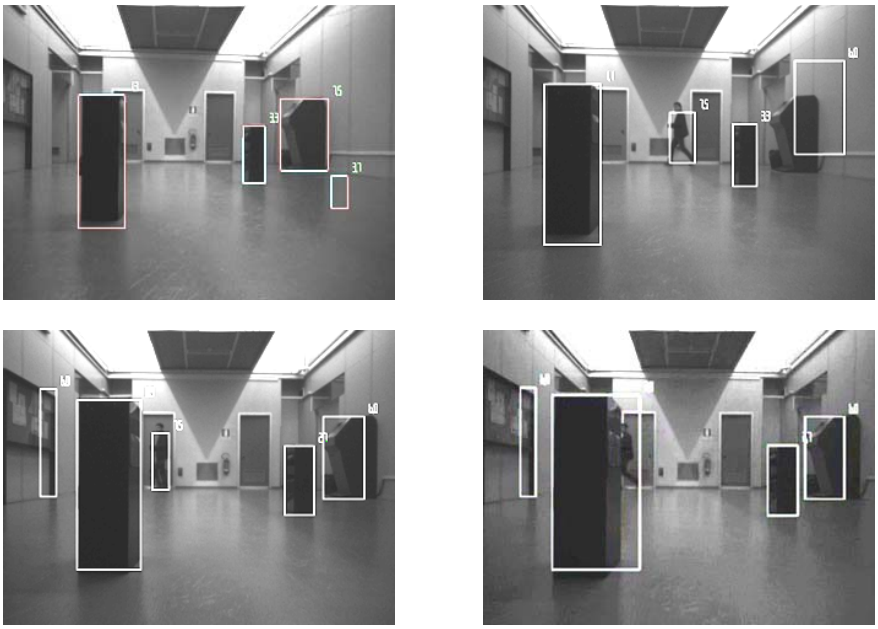


Figure 7: Some results of our obstacle detection algorithm.

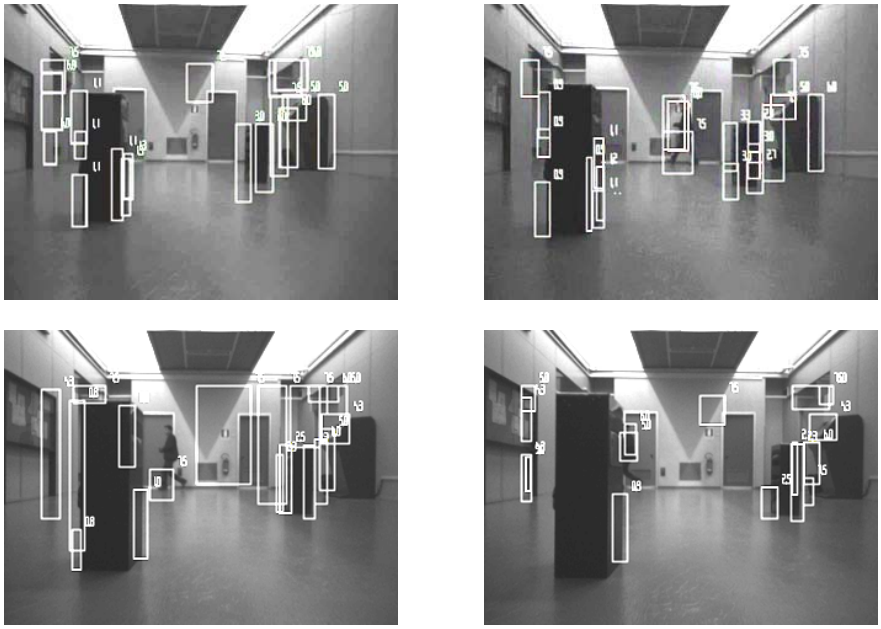


Figure 8: Some results of obstacle detection from SSD multi-scale stereo algorithm.

6.3 The Moving Object Detection: The Results

The Moving Object Detection subsystem (see section 5.6.1) has been performed using a synthetic database [6], created ad hoc using a rendering software (3D Studio Max). We have considered a resolution of 384x288 pixel^2 and a frame rate of 8 fps. Cameras are placed in the scene with parallel focuses, with a baseline of 10 cm, and at 80 cm above ground. The objects in the scene move according to different trajectories and speeds (from 0.5 m/s to 4 m/s). The robot follows both rectilinear and curvilinear routes at a speed of 0.5 m/s and 1 m/s. We have built a database of 34 video clips as *Training Set* and one larger video clip as our *Test Set* (see **Figure 9**). The Test Set contains

complex motion scenes, as a wall falling down, a pendulum, etc. The experiments have been performed on a notebook Intel P4 1.5 GHz, 512 Mb RAM.

Figure 9: Some examples of our database.

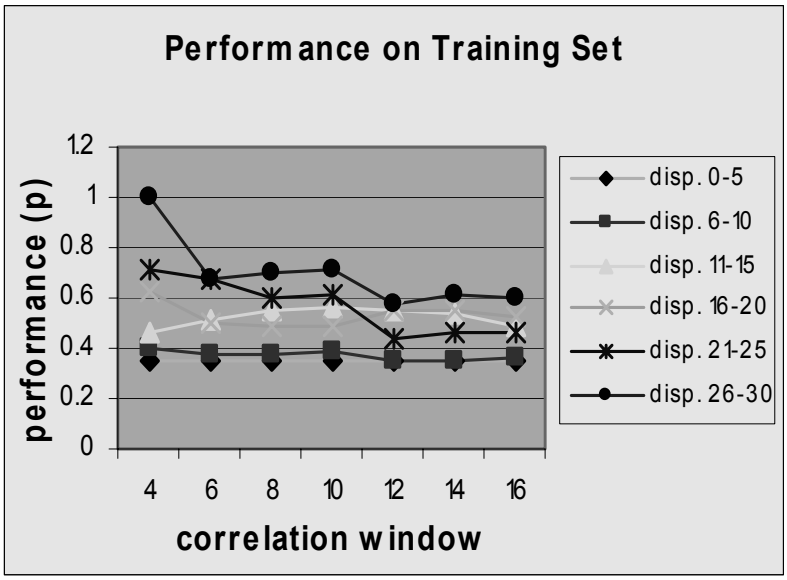
We use the *occupancy performance* defined in the previous section. In this case the bounding boxes are not related to the 4-connected regions with the same disparity value, but are related to the anomalous vectors (as defined in section 5.6.1), that detect the moving objects. Furthermore, trying to get a single evaluation value to be optimized, a unique *performance function* has been defined, depending from precision and recall:

$$p = \gamma_1 \cdot precision + \gamma_2 \cdot recall \qquad \textbf{(4)}$$

This function gives a higher weight to recall ($\gamma_2 > \gamma_1$), since in

autonomous navigation it is more important to find obstacles to avoid than a fine detection of them. We have tested different combinations of values for γ_1 and γ_2 and for each combination we have found the best parameters for our system. Afterwards, we have chosen the combination giving the best qualitative behavior of the robot in the environment (experimental values for γ_1 and γ_2 are set to 0.35 and 0.65). Another evaluation term is the frame-rate (*fps*) for respecting our real-time goal. The parameters of our system that need training are the following: *Correlation-window* for optical flow; *sampling step* for optical flow; *module and phase thresholds* for detection of anomalous motion vectors; *movie sampling*.

We have chosen the best set of parameters by optimizing the performance function and the frame rate on the training set. We have reported the performance function with respect to disparity quantized into six levels (the larger the disparity, the closer is the object). In the following are shown some of the results on the Training Set, used to find the optimum value of the main parameters (see **Figure 10**).



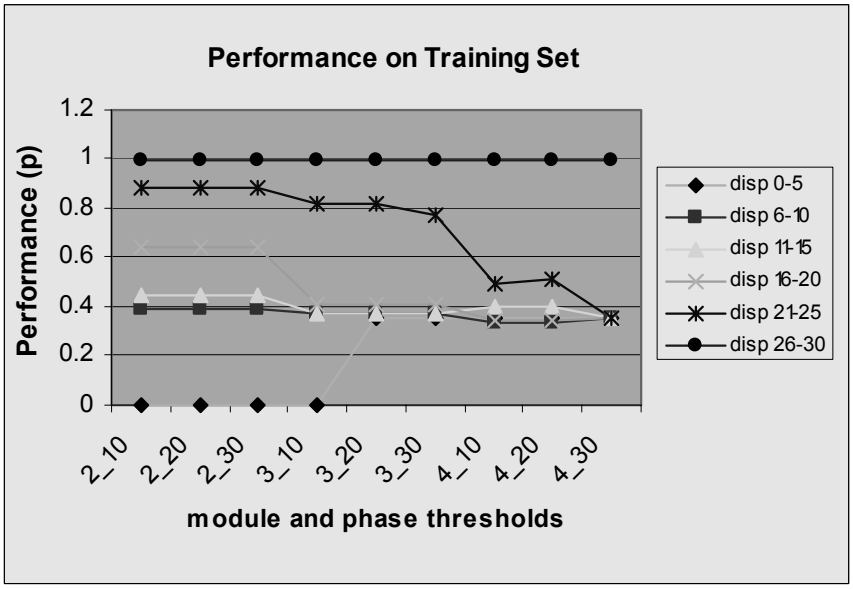


Figure 10: Performance functions with respect to disparity quantized into six levels, for the training of the correlation window and the module and phase thresholds (on Training Set)

A cumulative performance function is defined for assigning different

weighs to different disparity ranges (for rewarding central levels):

$$p_{cumul} = \sum_{i=0}^{5} w_i \cdot p(i\_level) \quad (5)$$

We have used the following vector of weighs, obtained by experimental evidences: $w_i = \{0, 0.125, 0.25, 0.25, 0.25, 0.125\}$

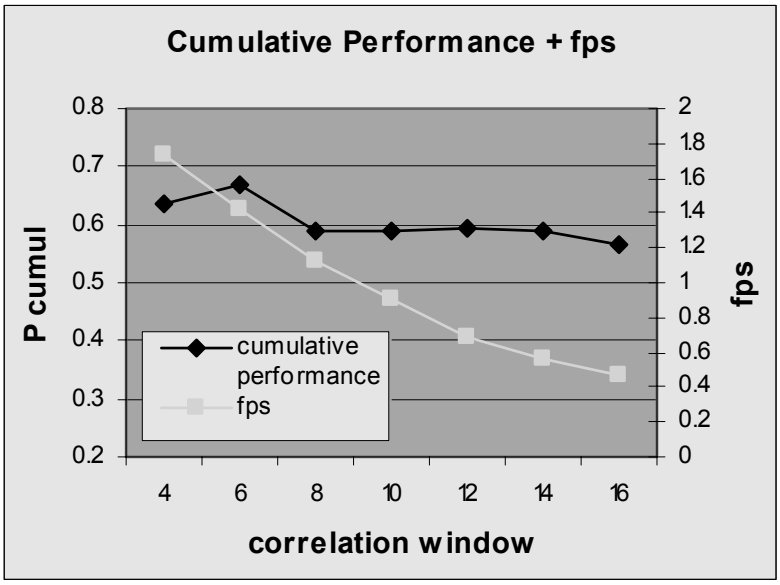


Figure 11: Cumulative Performance and Frame rate with respect to correlation window, to choose the best compromise

In **Figure 11** it is shown that a correlation window of 6 pixels gives the best cumulative performance at an acceptable frame rate. **Figure 12** shows Precision and Recall for each video clip. Notice that in some cases performance is not too high because of motion conditions that make more complex the moving object detection task, such as low speed and purely longitudinal trajectories.

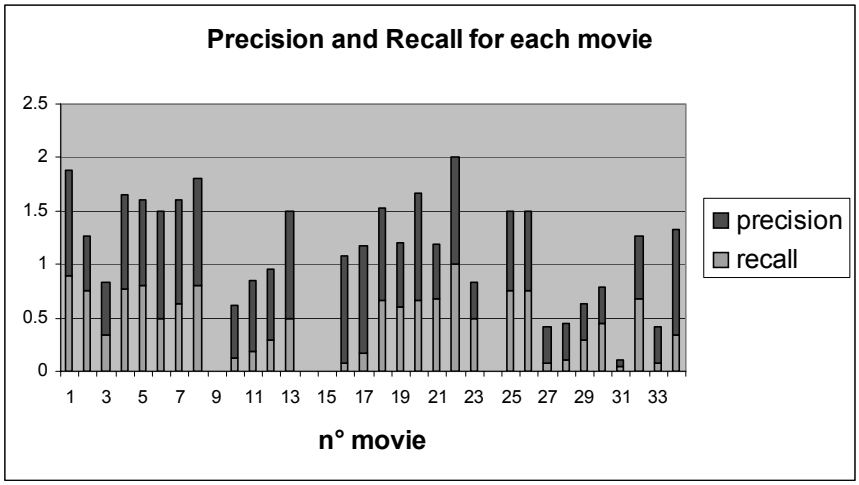


Figure 12: Precision and Recall for each video clip (of the Training Test) for different trajectories of the robot and of the objects

Finally the results on the Test Set are presented in **Figure 13**.

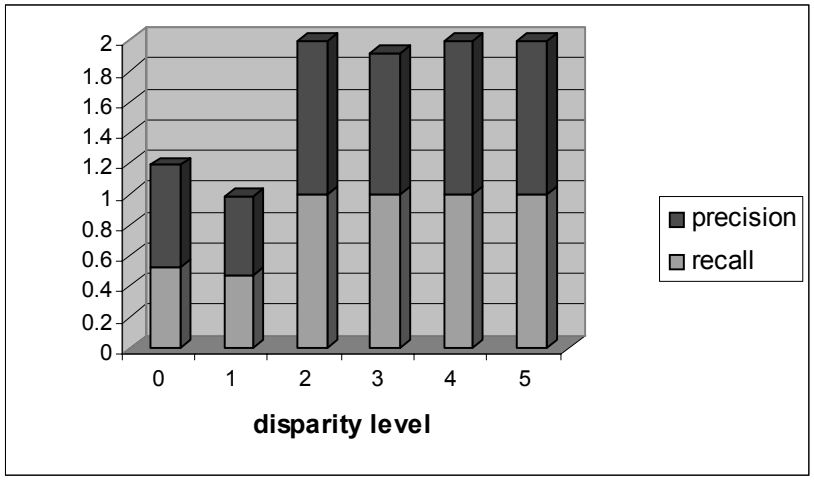


Figure 13: Some results on Test Set with respect to disparity level

It can be seen that our system has a good performance for objects that are at a low to medium distance from the robot, while degrades for far objects. This can be explained by the fact that the quantization and sampling noise become comparable to the disparity and optical flow

information. In **Figure 14** a visual display of the algorithm output is presented for some of the scenes in the Test Set.

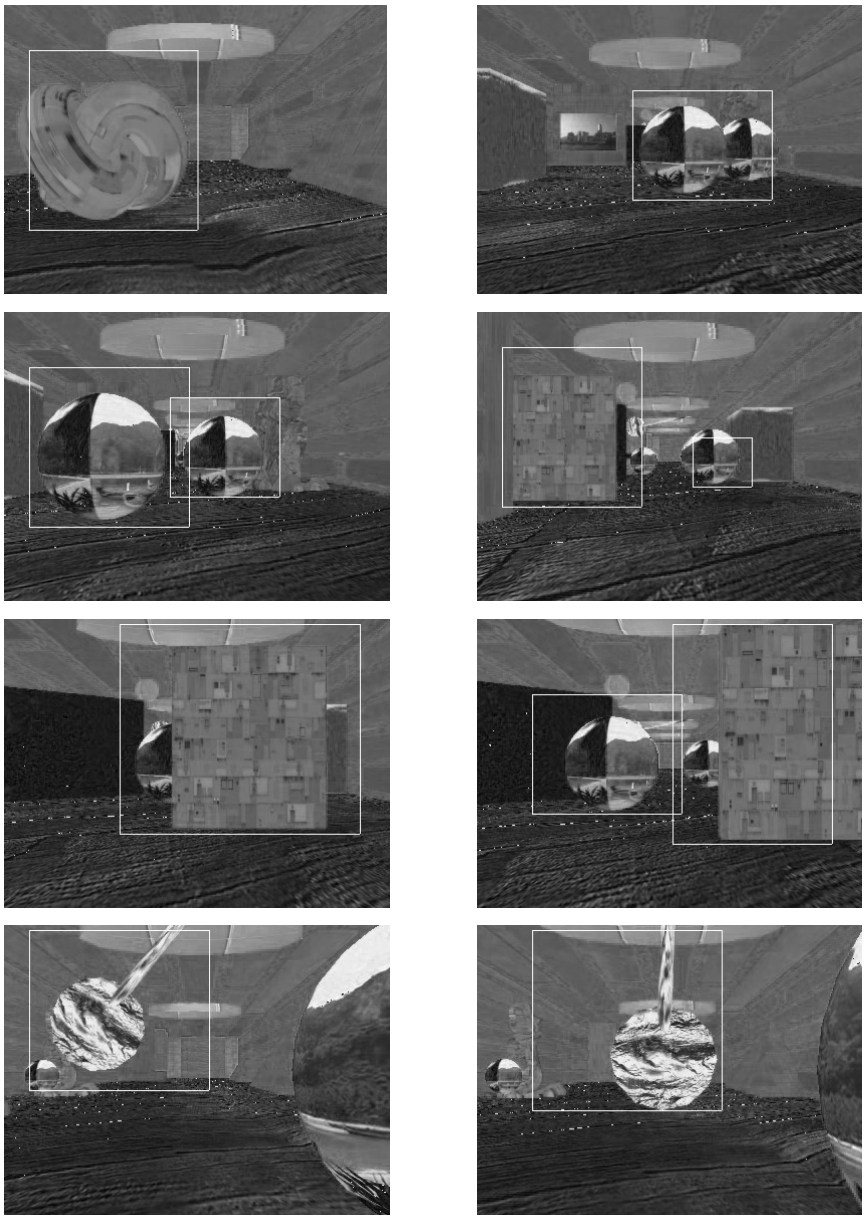


Figure 14: Some visual results.

References

[1] http://cat.middlebury.edu/stereo/

[2] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", International Journal of Computer Vision, vol. 47, no. 1, pp. 7-42, 2002.

[3] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light", in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 195-202, 2003.

[4] K. Konolige, "Web site", http://www.ai.sri.com/software/SVS, 2006.

[5] K. Konolige, "Small vision systems: hardware and implementation", in the 8th Proceedings of International Symposium On Robotics Research, pp. 111–116, 1997.

[6] G. Brosca and A. Limongiello, "AMR Synthetic Database", http://nerone.diiie.unisa.it/robot/, 2004.

Chapter 7

Conclusions

"... And so much fervently burns the Heart
Tormenting the flame of Love."
"...E sì nell'ardore brucia il cuore
tormentando la fiamma dell'amore."
A.L.

In this thesis we have proposed a new real-time video analysis system for autonomous navigation of a mobile platform. Our system presents several improvements as regards the state of the art of such systems. In particular we have addressed our attention to the challenging problem of the "Obstacle detection and avoidance" in unstructured environment, and we have analyzed the two phases of obstacle detection and moving object detection.

The obstacle detection problem, in the general framework of unstructured environment, is very hard to solve, in fact we do not have a large knowledge of the environment and of the objects in the scene, so that a robot has to build a wide understanding of the scene, in order to avoid obstacles. The motion of the camera, mounted on the robot, makes the video analysis very difficult and the most algorithms, in the literature, fail. Finally, an autonomous navigation needs a real-time elaboration to guide quickly the mobile platform through the safety path.

The entire system has been described according to a precise methodology for vision system development, called *systemic approach*. In this way, our system has some using specifications and guarantees a good performance for the specified application domain (AMR and AVG). The video analysis has been faced thinking about the nature of the environment, the nature of the mobile platform and the kind of goal we suppose to reach.

The major contribution of this work concerns a "perceptive" representation of the environment, that it is not a "passive" representation, but related to the final goal of autonomous navigation. It is based on the stereo vision paradigm and detect obstacles and moving objects in the scene right according to the autonomous navigation goal, that is obtaining a result as fine as it is enough for our aims. Therefore, we define a scalable system that works with a required resolution in a specific context.

The greatest advantage of stereo vision with respect to other techniques (e.g. optical flow, or model-based) is that it produces a full description of the scene, can detect motionless and moving obstacles (without defining a complex obstacle model), and is less sensitive to the environmental changes (the major disadvantage of optical-flow techniques). The stereo vision provides a 3D representation (or at least an approximation like a 2D ½ representation) of the scene, producing information about objects in the environment that may obstacle the motion. In stereo vision, the main difficulty is to establish a correspondence between points of the two images representing the

same point of the scene (disparity matching). All the approaches, in the literature, are based on this pixel correspondence.

We have proposed an extension of that concept, namely we have defined a disparity value for a whole region of the scene starting from the two homologous views of it in the stereo pair. The main reason of this extension is that a pixel-matching approach is redundant for AMR and AVG applications. In fact, in this framework, it is not very important to have a good reconstruction of the surfaces, but it is more important to identify adequately the space occupied by each object in the scene. Moreover the pixel-based approaches are lacking in robustness in some realistic frameworks, especially for video acquired from a mobile platform. Our method estimates the average depth of the whole region by an integral measure, and so has fewer problems with uniform regions than other methods have. The estimate of the position of the regions is sufficiently accurate for navigation and it is fast enough for real time processing.

The results of our method for stereo matching have been shown in a comparison with the best algorithms in the literature. We have reported some results obtained on a realistic video acquired from our mobile platform. The video sequence is characterized by camera vibration, light changing, uniform obstacles, in order to underline the limits of the algorithms present by now in the literature that are pixel-based. It is also proposed a quantitative measurement for performance evaluation, with a reference to our specific goal of the obstacle detection in autonomous navigation framework. The experimental results we carried out show that the proposed idea is very promising.