Analysis of Current Approaches in Automated Vision-based Navigation

Edited by: Fredrik Bergholm, KTH, Sweden
Panos Trahanias, ICS-FORTH, Greece
Stelios Orphanoudakis, ICS-FORTH, Greece

Contributions by: Antonis A. Argyros, KTH, Sweden & ICS-FORTH, Greece
Jens Arnspang, DIKU, Denmark
Fredrik Bergholm, KTH, Sweden
Wolfram Burgard, Univ. of Bonn, Germany
Kostas Chandrinos, ICS-FORTH, Greece
Raja Dravid, Univ. of Zurich, Switzerland
Rachid Deriche, INRIA, France
Dieter Fox, Univ. of Bonn, Germany
Knud Henriksen, DIKU, Denmark
Joachim Hertzberg, GMD, Germany
Lena Gaga, ICS-FORTH, Greece
Claus B. Madsen, Aalborg University, Denmark
Ralf Möller, Univ. of Zurich, Switzerland
Stelios Orphanoudakis, ICS-FORTH, Greece
Rolf Pfeifer, Univ. of Zurich, Switzerland
Erich Rome, GMD, Germany
Giulio Sandini, DIST-Univ. of Genova, Italy
Frank Schönherr, GMD, Germany
Jose Santos-Victor, Instituto Superior Tecnico, ISR, Portugal
Antonella Semerano, ROBOSOFT, France
David Sinclair, Technical Univ. Graz, Austria
Panos Trahanias, ICS-FORTH, Greece

July 24, 1997

Preface

This document has evolved as part of a survey, that comprises task 1 of the TMR project VIRGO[1] (`http://www.ics.forth.gr/virgo`). VIRGO involves ten participating organizations across Europe, and a number of research, educational and industrial organizations are indirectly associated to VIRGO. The subject pursued focuses on "vision-based navigation" and a broad spectrum of issues are addressed reflecting current research trends, strengths and interests of the participants.

The material presented in the next chapters has been contributed mostly by researchers within VIRGO partners, although in some cases contributions have been provided by researchers from non-participating organizations. An editing procedure has been followed to compile the material into coherent chapters. After a number of reviews and related revisions, the current document has resulted, featuring a broad review as well as recent results in "vision-based navigation".

Although the initial purpose of this document has been to serve research goals within VIRGO, we believe that in its present form it may also be used in various ways. Hopefully, it will form a reference text within VIRGO and the research community, whereas at this point there are plans to use—part or all of it—as class material for graduate courses addressing related subjects.

A text of the size and (hopefully) quality, like the current one, does not just happen. Instead, considerable effort is required at all levels: individual contributions, editing, revisions, compilation of coherent chapters. It is evident that in this case all contributors have strived to provide excellent pieces of text, and this is highly acknowledged. Last, but not least, the financial contribution of EC is greatly acknowledged since it has facilitated the operation of the VIRGO Network and the pursue of research at a European level in "vision-based navigation".

[1] EC Contract No ERBFMRX-CT96-0049

Contents

Chapter 1

Background

by Fredrik Bergholm

A mobile platform, an autonomous vehicle, a moving robot arm are - generally speaking - devices equipped with sensors for moving around, navigating, in an environment, simultaneously performing some tasks.

The emphasis of this overview will be on *visual-based navigation*, where other sensors, such as *sonar*, are looked upon as complementary sensors. There are of course at present many systems where the roles are opposite. Vision is a complement and sonar, odometry, laser range finders are the chief responsible sensors for collision-free navigation. A comprehensive overview can be found in [1].

Visual-based navigation, by necessity, comprises a number of basic capabilities. The moving pattern of light on the images - often called image flow or optical flow - contains information about the environment in many respects, such as the three-dimensional structure, independently moving objects, and time-to-collision, to mention a few. Basic *visual capabilities* (Chapter 2) deals with the extraction of such information from image data. Systems may be monocular, binocular or multi-ocular. The image and information processing in a binocular system can be done in quite different ways compared to the monocular set-up.

The process of navigation, in animals and humans, usually involves a cognitive representation of the environment that facilitates localization and the determination of the appropriate motion(s) to reach a navigation target, where *visual landmarks* play a key role (Chapter 3). Landmarks need to be *selected* and *recognized.* The capability of moving in a qualitative fashion from one environment landmark to the next—*visual homing*—obviously includes elements of learning, memory, strategic decisions (when selecting them) and pattern representation. This kind of visual processing is characterized by image processing in very long sequences of images. Chapter 4 is a vivid example of the links between zoology research and robotics. Desert ants take advantage of global light patterns in the sky, to approximately find their way back home after a stroll. Apart from being an example when global orientation information affects navigation (dead-reckoning[1] and homing), it also

[1] According to [1, p.3]: "*Dead reckoning* (derived from "deduced reckoning" of sailing days) is a simple mathematical procedure for determining the present location of a vessel by advancing some previous po-

illustrates a special case of visual homing - certain memorized snapshots may probably serve as fine-tuning mechanisms for finding the nest, a process so-called *visual piloting.*

The desert ant example also serves to illustrate some basic principles of autonomous behavior. *Instead* of *sense-think-act* cycles, continuous sensory-motor coordination from visual stimuli leads to seemingly intelligent behavior; the ants returning to their nests even in the presence of detours due to obstacle avoidance.

A classic issue in robotics is the use of internal world representations, where there is a range of different opinions (Chapter 5), one view being that they should be avoided in the first place (Brooksian) and another that we need to represent things at various levels rather meticulously, and a third that some mix of both paradigms is appropriate. Using sonars, certain so-called grid-based representations are popular. Animals in turbid water, like the hippopotamus, or dolphins use sonar. In brain sciences it is customary to speak of *neural images*, indicating that sensory data (including visual data) are re-assembled in various clever ways in the human brain. This also has some links to *short-term memory.* Aspects of learning, with relevance for vision-based navigation is also reviewed in Chapter 5.

Given that we have some pre-stored spatial knowledge of the workspace (or knowledge built up gradually over time from the sensors), that forms the so-called *maps* (often 2D) of the environment, a question arises on how should this information be employed by the autonomous agent for the purpose of determining appropriate motions. This enters topics such as *motion planning* and *path planning*, to which Chapter 6 is devoted. Whereas the classic approaches presuppose rather exact knowledge, it is obvious that many reasons would suggest to soften this assumption. Consequently, a body of theory and experimental work around *map-based navigation* and *path planning with uncertainty* has evolved. Occupancy maps, such as local *occupancy grids* from ultra-sonic data, are built up gradually while travelling, without the need to know absolute positions, sometimes augmented by vision (cf. Sec. 7.4).

Control issues (Chapter 7) are a core preoccupation in mobile robotics under all circumstances, irrespective of the choice of sensors. Some reflex-like behaviors in connection with obstacle and collision avoidance have been implemented in several laboratories, and have proved to work to some extent, even if vision is the *only* sensor (sometimes augmented by *peripheral vision*). A fundamental issue at this point is how to combine behaviors in a meaningful way. This has not been addressed sufficiently to-date, as it becomes obvious from the independent development of various behaviors. However, approaches for integrating strategic planning and local reactivity (fine motion control) have been developed, and such a design is presented in chapter 7.

sition through known course and velocity information over a given length of time... The vast majority of land-based mobile robotic systems in use today rely on dead reckoning.., and like their nautical counterparts, periodically null out accumulated errors with recurring ‘fixes’... The most simplistic implementation of dead reckoning is sometimes termed *odometry*; the term implies vehicle displacement along the path of travel is directly derived from some onboard ‘odometer’.”

1.1 A Guide to the Reader

The structure of this document is as follows. For each *main topic* there will be a number of topics. A main topic may be something like *visual competences*, relevant to the VIRGO project, and a topic under *visual competences* is, for example, *egomotion estimation.* A topic may also have subtopics.

For each topic there will, as a rule, be a review section and a conclusion section. The latter will cover things such as subjective comments, pointing at what seems to be missing in previous approaches together with suggestions for what appears to be worthwhile research directions, at the moment. Some of the conclusion sections have a more modest heading, such as 'remarks'. The union of all conclusion sections should, to some extent, serve to identify problem areas.

In addition to conclusion sections, some sections are more tuned towards analysis or discussions, as for example, Section 7.4 (on integration of control and planning), some subsections, such as Section 2.4.4 on time-to-collision, and Section 5.5, although, of course, the borderline between analysis and review is soft.

Existing systems are described in more detail in Chapter 4, Section 7.2 (control for fast collision avoidance) and Section 7.3 (visual behaviours for navigation).

As a service to the reader, the review section will often be preceded by a section on definitions, called 'definitions' or 'introduction'. These are to be looked upon as tutorial sections for the non-specialist, and an attempt to make the terminology more coherent.

One chapter is devoted to industrial applications of autonomous navigation technologies, with an example which by the way is not vision-based but rather ultrasound- and laserfinder-based, but nevertheless illustrative of future potentials and trends.

Bibliography

[1] Borenstein, J., Everett, H. R., and Feng, L. *Navigating Mobile Robots.* A K Peters, Wellesley, MA, 1996.

Chapter 2

On Visual Competences

by Antonis Argyros, Jens Arnspang, Fredrik Bergholm
Rachid Deriche, Knud Henriksen and Panos Trahanias

Imagine an autonomous mobile agent, navigating in an unstructured environment, that has been assigned with a set of goals (navigational tasks). In order to successfully accomplish its goals, the agent should be *equipped* with visual capabilities, much like the capabilities of biological creatures. Such capabilities are usually termed *visual competences*; here, we provide a review of the research work that has been done with respect to four basic visual competences, i.e. *structure and egomotion estimation*, *independent motion detection*, estimation of *time-to-contact*, and *tracking.* The literature on three-dimensional (3D) reconstruction has been somewhat de-emphasized (some is mentioned in Section. 2.2) since it is doubtful whether a mobile navigation system should spend too much time on more exact 3D map-building. Neverheless, binocular techniques have been and will be used in many vision-based mobile navigation systems, and a separate section on binocular depth measurements could have been included in the visual competences chapter, as well as a section on color processing. Color and stereo matching will, however, be touched upon in the other chapters, and they are an integrated part of many types of computations, and capabilities. Section 2.3 dealing with independent motion detection shows that the topic is closely related to binocular vision.

For a mobile robot that should be able to navigate in unknown and potentially dynamic environments, the solution to these problems provides a rich perceptual input that is crucial for determining the robot behavior. An effort is made to define the problems and to present some of the solutions that have been proposed up to now.

Section 2.1 deals with measurements of *2D motion*, which can be used for developing visual competences, whereas the following sections deal with visual competences per se.

2.1 Measurements of 2D motion

The relative motion between an observer and the 3D environment causes a point of the 3D space to project onto different points of the 2D image plane, in a temporal sequence of images. The problem of 2D motion computation amounts to the problem of computing the displacements of the projections of 3D points between consecutive image frames.

2.1.1 Definitions

A direct approach to motion computation addresses the *correspondence problem*, i.e. the positions of dots, local maxima/minima, endpoints of lines, corners, edge or line intersections (etc.) are computed in successive frames and the differences in their positions constitute the *displacement vectors*. This process is sometimes referred to as *explicit token matching*.

If the sparse displacement vectors approximate velocity vectors well, one speaks of *image flow fields*, *image flow* or *flow fields*, or, simply, *image velocity fields*. The image flow fields should ideally coincide with the *motion field*, which is defined as the velocity field generated by surface points in space (whether identified or not) when projected (through perspective projection) onto an image surface. Some image flow vectors correspond to other events than the motion field, such as moving shadows, and reflected light sources, to mention a few.

In the case that the displacements are not too large, a dense-like *flow field* is frequently computed from derivatives in time and image coordinates, typically denoted x and y. The space coordinates are often denoted (X, Y, Z) and, if assuming *perspective projection*, the transformation between space and image coordinates is simply $(X/Z, Y/Z, 1) = (x, y, 1)$ (see Fig. 2.1).

The term *optical flow*, although often used, is quite problematic. Some people use it as a synonym to *image flow*, a practice we will mostly avoid in this text. A common attempt to definition of *optical flow*, is that the relative motion of the observer with respect to the scene gives rise to motion of brightness patterns that are formed on the image plane. The instantaneous changes of the brightness patterns[1] are analyzed to derive the optical flow field, a two-dimensional vector field reflecting the image displacements (small time separation). Actually, as defined above, the dense vector field derived from the patterns could not be a velocity field in the image plane, but a set of vectors that sometimes are velocity vectors, sometimes just components of velocities (so-called *normal flow*) and sometimes neither because some interpolation process generates data where there is in general insufficient information to infer image velocities. Much efforts have been directed towards estimating the so-called *dense optical flow*.

For dense optical flow to be computed, additional assumptions have to be made. Common assumptions are related to smoothness or geometrical modeling of the shape of the scene. Employing additional smoothness assumptions to recover an unknown function when the available constraints are not sufficient is a general mathematical technique known as *regularization*. In many such approaches, normal flow is not (much) retained but replaced by values interpolated from other places in the image. In this case, we have image flow plus some interpolated flow (which in unfortunate cases may neither be close to true image flow nor true normal flow).

[1] Intensity, image intensity and brightness are often used as synonyms in computer vision. It may sometimes be some weighted average of red, green and blue images (RGB), or, an individual spectral band.

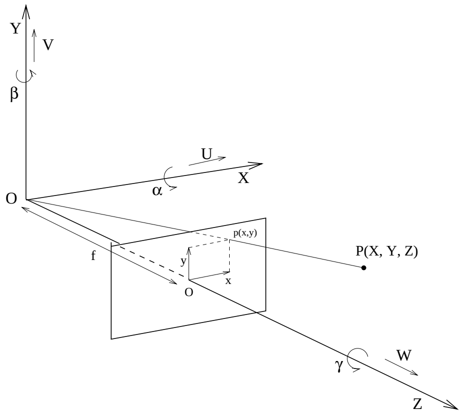


Figure 2.1: *Translational and rotational motion parameters for a camera-centered coordinate system. The origin can be thought of as the lens center for a thin lens.*

However, *sparse optical flow*, corresponding to velocities only, is also sometimes computed, for example using Eq. (2.1), below, with second order derivatives of brightness I, $\partial^2 I/\partial x^2 = I_{xx}, I_{yy}, I_{tx}, I_{ty}$, as for example in [14]. Employing *features*, such as *edges*, does not mean that explicit token matching need be done. It is quite possible for *edge-based flow* edges following the motion, to obtain optical flow (both image flow and normal flow) estimates anyway, as for example in [153]. This amounts to replacing the intensity (brightness) I in Eq. (2.1) by **another** function, namely $B(x, y)$ the blurred binary edge image, which thus is converted into a kind of intensity image.

In many cases, a sequence of images can be modeled as a continuous function $I(x, y, t)$ of two spatial (x, y) and one temporal (t) variables. $I(x, y, t)$ expresses the image intensity at point (x, y) at time t. Assuming that irradiance is conserved between two consecutive frames, and that u and v are the x- and y- components of the image flow, it is expected that the irradiance will be the same at point $(x + \delta x, y + \delta y)$ at time $t + \delta t$, where $\delta x = u\delta t$ and $\delta y = v\delta t$. This leeds to the *optical flow constraint equation*, originally mentioned in Horn and Schunk [71]:

$$I_x u + I_y v + I_t = 0 \tag{2.1}$$

where, $I_x = \partial I/\partial x$, $I_y = \partial I/\partial y$ and $I_t = \partial I/\partial t$ are the two spatial partial derivatives (in $x-$ and $y-$ direction) and the time derivative of the image intensity function, respectively. Equation (2.1) gives one constraint for the components u and v of image flow. Equation (2.1) can be written in the form of a dot product

$$(I_x, I_y) \cdot (u, v) = -I_t \tag{2.2}$$

geometrically interpreted as permitting the computation of the projection of the image flow along the intensity gradient direction (i.e. the perpendicular to the edge at that point). This projection is also known as *normal flow* (see also Fig. 2.5). Equation (2.2) can be viewed as the mathematical expression of the so-called *aperture problem* and shows why extra assumptions should be made in order to compute image flow.

One may replace I by some other function F to obtain

$$F_x u + F_y v + F_t = 0, \tag{2.3}$$

as was mentioned earlier. Sometimes, edge contours are explicitly calculated and the normals along those contours are discretely sampled and put in as (F_x, F_y), and F_t is replaced by the measurable edge displacement, a so-called normal displacement (small time separation). In this case F_x, F_y is not a gradient but sampled unit vectors normal to a contour. For the *velocity functional method*, [151], (u, v) are replaced by functions in image coordinates (see also [15]).

2.1.2 Sparse displacement fields - establishing correspondences

The correspondence problem has been addressed by many authors [88, 37, 70] and is usually treated in two steps [88]. First, a type of image primitive is selected (eg. raw pixels, lines and contours [157]) and a similarity criterion between such primitives is defined. In an effort to select tokens that differ from their surroundings, corners have also been employed [100] and points of high interest [97]. Then, for each primitive in one frame, a search is performed which tries to locate this primitive in the other frame. The primitive is said to be found at the position that gives the best match (in terms of the similarity criterion chosen). For this search (or mapping), various smoothness assumptions have been made about the spatial and temporal characteristics of motion, which basically restrict either the structure of the scene in view or the motions of the objects in view [37, 87]. Finally, in order to reduce the computational requirements of the search, heuristic rules have been employed. Such a formulation constrains a feature to move in a way similar to neighboring features [22, 54]. Trajectories are sometimes used as a means to obtaining more accurate approximations to instantaneous velocity (displacement) vectors. In [28] a continuous trajectory *consistent with* certain linear velocity operators is discussed.

2.1.3 Computing optical flow

The theory and numerical computation of *optical flow field* has attracted considerable research efforts for many years [71, 100, 67, 9, 66, 145, 27, 8, 128, 38]. The techniques for obtaining optical flow involve two computational steps that exploit two different constraints that are needed for its computation.

In a first step, assuming the conservation of some type of information within the image frames, the locally available velocity information is computed. Three different approaches can be distinguished [56, 128]: *gradient based approaches* which assume conservation of image intensity [71, 100], *correlation based techniques* which assume conservation of the local intensity distribution [158, 34, 8] and *spatiotemporal energy-based approaches* [2, 58, 66, 159] which are analogous to gradient-based approaches in spatiotemporal frequency space.

The second step of optical flow computation employs a regularization technique, which is based on the hypothesis that optical flow varies smoothly in most parts of the image [71]. However, the optical flow field is not smooth for a number of reasons. First, the smoothness of optical flow depends on the continuity of depth variable Z. In most scenes depth discontinuities do exist and, therefore, the optical flow cannot be regarded as smooth. Black and Anandan [29] provide a framework that can be applied to standard formulations of optical flow estimation in order to reduce their sensitivity in the presence of transparency, depth discontinuities, independently moving objects, shadows and specular reflections. Uras [145] requires that the gradient of the image brightness does not change over time. In other cases [132, 99, 151, 150], assumptions about scene geometry have been employed rather than smoothness of the optical flow field. A typical assumption of this type is that the world can be locally approximated by planar surfaces.

For a detailed presentation of different approaches to optical flow estimation the reader is referred to [128]. Furthermore, a comparative study of representative optical flow techniques can be found in [23].

2.1.4 Computing normal flow

The equation describing image flow, Eq. (2.2), gives only one local constraint on the flow values while there are two unknowns to be recovered for each image flow vector. Due to this lack of information, all methods that aim at recovering dense image flow need to employ additional assumptions, such as smoothness[2]. For this reason, many research efforts are targeted towards approaching visual problems by employing only the projection of image flow on the image gradient direction, i.e. the normal flow field.

The normal flow field has been used in the past for both egomotion estimation [56, 7, 127] and independent motion detection [105, 123]. However, the problem of egomotion estimation cannot be solved by using the normal flow field without any knowledge about the viewed scene. This is because each image point (in fact, each point at which the intensity gradient has a significant magnitude and, therefore, a reliable normal flow vector can be computed) provides one constraint on the 3D motion parameters. Furthermore, if no assumption is made regarding the depth Z, each point provides at least one new independent depth variable. Evidently, the problem is underconstrained for general motion if no additional information is available regarding depth.

2.2 Egomotion and structure estimation

Given an observer that moves rigidly in a stationary 3D environment, as illustrated in Fig 2.1 for a camera-centered coordinate system $OXYZ$, the problem of egomotion estimation can be defined as the problem of recovering the translational $\vec{t} = (U, V, W)$ and the rotational $\vec{\omega} = (\alpha, \beta, \gamma)$ parameters of the observer's motion. The problem of egomotion estimation is usually addressed by considering two different subproblems. The first subproblem concerns the type of motion information that can be computed from 2D images of a 3D scene, and the second focuses on how this 2D motion information is related to the 3D motion parameters of the observer. Provided some answers to the above questions, the main issue is then to develop efficient and robust algorithms to estimate egomotion. In the remaining of this section we will review the solutions that have been proposed in the literature regarding these problems.

We consider the problem of computing *structure from motion* as a subproblem of *egomotion estimation* since once the egomotion is known, the 3D structure is also more or less calculated, or, conversely if a "structure from motion calculation" is successful, then the egomotion problem is quite close to being solved.

[2] For sparse image flow, whether token-based or derivative-based, this is not needed.

2.2.1 Definitions

The plethora of methods for estimating egomotion can be classified based on the type of assumptions they make about the observer and the scene of view.

The *Focus of Expansion* (FoE) is the point where the direction of translation intersects the image surface. Thus, the coordinates of the FoE provide qualitative knowledge regarding egomotion. The imaging center (center of projection) and the FoE define the direction of translation, or, the direction of *translational velocity.* Unfortunately, there is an *unrecoverable scale factor* in the translation, i.e. a 3D translational motion (U, V, W) at a certain depth produces exactly the same image flow as motion $(2U, 2V, 2W)$ at depth $2Z$.

2.2.2 Structure and motion from motion

The 3D motion and structure of the scene can be computed from equations that relate the measurements of 2D motion to the 3D motion parameters. The problem of structure from motion has been extensively studied and a number of results concern the type of information about motion and structure that can be extracted by processing a sequence of images [143, 144]. Longuet-Higgins [91] and Tsai and Huang [140] showed that the image measurements in two consecutive image frames are related through the 3×3 *Essential Matrix*. From the nine parameters of the essential matrix, only eight can be determined because of the unrecoverable scale factor. Thus, the essential matrix can be computed given eight point correspondences and the 3D motion parameters are derived by matrix computations. However, this technique is very sensitive to errors in the computation of correspondences and, therefore, cannot give rise to a robust algorithm. Structure from motion methods have also been studied in the presence of noise [140, 130, 4] and a plethora of algorithms have been designed for the computation of the structure of a scene [149, 150, 155, 141, 131, 137, 125, 154, 89, 114].

Structure from motion, in cases where rotation has been *separated* from the translation, seems, if not easy, quite practical. A well-known example of structure estimation for pure translation in long time sequences, is the *weaving wall* representation, and computation [19].

An important issue regarding the observer is the number of cameras he is equipped with and their arrangement. A lot of research work has been done on trying to estimate the egomotion parameters based on various stereoscopic configurations, where motion information is combined with stereoscopic information in order to provide additional constraints regarding the scene structure [114, 48, 149, 126].

An additional degree of freedom is imposed from the camera model and, more specifically, from the projection model assumed. This is because the exact form of the equations relating the 2D motion with the 3D motion parameters and structure is determined by the type of projection assumed. The *orthographic* and the *perspective* models [101] are the most commonly used. The orthographic model results in linear equations, but is a realistic approximation in a small area around the center of the image (the projection of the optical axis on the image plane), or in the whole visual field of cameras with large focal length. Because of the relative simplicity of the orthographic projection model, it was the first that was considered in attempts to solve the structure from motion problem [144, 68, 6]. On the contrary, the perspective projection model is more realistic, but it results in a non-linear relation between the 3D motion and structure and the measurements of 2D motion.

Another class of assumptions about the observer is related to the type of his motion. In the general case, three translational and three rotational parameters should be estimated. One major simplification stems from an assumption of pure rotational or pure translational egomotion. In the first case, the 2D motion measurements do not depend on the image structure and therefore, the estimation of egomotion is greatly facilitated. For example, Aloimonos and Brown [5] have studied the case of purely rotational motion, where they formulated the problem as a least squares minimization of the difference between predicted and observed normal flow values. This formulation leads to linear equations relating the rotational motion parameters to the normal flow field. In the case of pure translation, again, FoE can be easily located based on geometrical properties of the flow field. Horn and Weldon [73] suggested methods for the problem of egomotion estimation for the case of pure translational motion. More specifically, they exploited the fact that the the term W/Z is positive (negative) for all points in the image, if the observer approaches (departs) from the scene. Therefore, every normal flow vector provides a constraint on the location of the FoE; it has to be in the half-plane (or hemisphere in case of spherical projection) which is on the opposite (same) side of the gray level edge of the normal flow vector. Constrained combinations of translation and rotation have also been investigated. For example, in [156] egomotion is estimated for an observer rotating around the direction of translation.

The problem becomes much more complicated in the case of combined translational and rotational egomotion because then it is quite harder to define quantities that depend on the egomotion parameters but not on the scene structure. This is also the most interesting case because it is not possible to guarantee that the translational or the rotational part of motion will be exactly equal to zero. A number of methods for egomotion estimation actually try to "derotate" the image flow field (i.e. remove the contribution of pure rotational motion) so that what remains is a purely translational field. Prazdny [115] and Burger and Bhanu [33] suggested techniques which decompose the flow field into its translational and its rotational components. The problem is addressed as a search for the rotation to be subtracted from a flow field in order to be left with a purely translational flow. The error function suggested was based on the variance of the intersections of one displacement vector with all the other vectors. Nelson and Aloimonos [106] proposed a decomposition method but using the spherical projection model. This work exploits the elegant geometric properties of the translational and rotational flow fields when they are projected on a spherical eye. Recently, a derotation method has been proposed by Lobo and Tsotsos [90] which is based on the Collinear Point Constraint (CPC). The CPC defines a way of selecting projections of optical flow vectors so that the combined motion for these vectors depends only on the translational motion of the observer. Given this, the Focus of Expansion and the parameters of rotational motion can be determined.

Various methods employ assumptions that require the external world can be modeled by a restricted class of shapes [132, 99, 151, 150, 104]. For example in [104], the problem of egomotion estimation is treated for scenes that can be modeled as either a plane or a quadratic patch.

In the approach by Fermüller [56], egomotion estimation is based on the geometrical properties of the normal flow field. More specifically, it is shown that the change of sign of the normal flow for specific classes of vectors[3] gives rise to simple patterns, the characteristics of which depend on the egomotion parameters. This approach is interesting in the sense that it is based on the extraction of qualitative information from a sequence of images (i.e. the signs of the normal flow vectors). On the other hand, in real-world images the patterns formed are quite sparse because the normal flow field is usually a sparse field. For this reason, the extraction of 3D motion information from the patterns is a difficult problem in its own right.

2.2.3 Conclusions

Most researchers constrain either the motion or the environment in order to obtain robust estimates of translational direction, rotation and structure. There seems to be no universal solution for the general case of unrestricted motion and structure, working in a predictable way for any environment. From the point of view of a moving navigating platform, on the other hand, it is quite possible that precise knowledge of structure or motion is not needed in the first place. However, it is an open research issue exactly what is needed.

2.3 Independent motion detection

Given an observer pursuing unrestricted 3D motion, the problem of independent motion detection can be defined as the problem of detecting objects that move independently of the observer in 3D space. This problem takes a special form if the observer does not move relative to the environment. In this case, all points of the static environment are projected at the same locations of the image plane, while, points belonging to moving objects are projected at different 2D coordinates as a function of time. Thus, in the case of a still observer, the problem of detecting moving objects can be treated as a problem of *change detection* [74, 129].

The situation is much more complicated when the observer is moving relative to the environment. In such a case, the points of both the environment and the moving objects project in different 2D locations on the image plane. If independent knowledge on the observer's motion parameters does not exist, change detection cannot anymore handle the problem of detection of moving objects. In this chapter we will treat the problem of independent motion detection for the case of a moving observer. This case is also the most interesting, because biological and man made visual systems usually move. Even if the body of an observer is still, the eyes are continuously moving [98].

[3] These classes are formed on the basis of the directions of the normal flow vectors

2.3.1 2D methods

Independent motion detection has often been methodologically approached as a problem of segmentation of the two dimensional motion representation that can been computed from a temporal sequence of images. The general algorithmic framework that is applied by such methods is the following:

1. Extract some sort of motion representation from a temporal sequence of images.
2. Assume that this information is described by some kind of 2D model.
3. Detect the discontinuities of this model and report them as 3D motion discontinuities.

Based on different alternatives regarding each of the above algorithmic steps, a large variety of methods have been proposed. Step (1) is implemented by any of the available image flow estimation methods or by employing normal flow. In step (2), an affine or a quadratic model [142] are usually assumed for the image flow. In [147], the parameters of an affine model are estimated locally in image patches. Patches are then combined in larger motion segments based on a k-means clustering scheme that merges two layers if the distance of their motion parameters is sufficiently small.

Some methods do not compute image flow, but they are based on normal flow. For example Irani et al. [80] and Torr and Murray [139] perform independent motion detection based on normal flow rather than image flow, employing again two dimensional models. Nordlund and Uhlin [108] estimate the parameters of an affine model of 2D motion. By constraining the spatial extent of the independently moving object, they assume that the estimation of the dominant 2D motion will not be affected considerably by the presence of the independently moving object. Therefore, independent motion can be achieved by determining the points where the residual between the measured flow and the flow predicted by the estimated parameters is large.

Major emphasis is usually put on step (3), for which a variety of techniques have been proposed. Bober and Kittler [31] combine robust statistics with the Hough transform to detect independently moving objects. In [75], robust statistics and more specifically M-estimators are used to distinguish the dominant 2D motion from the secondary 2D motions. A similar idea is exploited by Ayer et al [18], where two other robust estimators, namely Least Median of Squares and Least Trimmed Squares are used to discriminate the dominant from the secondary 2D motions.

Bouthemy and Francois [32] view 2D motion segmentation as a problem of statistical regularization using Markov Random Field models. Other methods, approach independent motion detection as a problem of *change detection*, a technique usually applied in the case of a still observer [74]. Paragios and Tziritas [111] extend the application of change detection in the case of a moving observer. First, the dominant 2D motion is estimated and then compensated. After compensating for the dominant (i.e. observer's) motion, a change detection algorithm is applied to isolate regions that move independently. Odobez and Bouthemy [109] also take the approach of camera motion compensation by employing multiscale MRF models.

Many of these methods produce good results in certain classes of scenes, but have two basic drawbacks that are related to some inherent problems of steps (1) and (2) of the general algorithmic framework presented earlier. The methods that are based on the computation of optical flow tend to eliminate the effect they are trying to detect, because the smoothness assumptions that are employed in order to compute optical flow, tend to compensate the existing motion discontinuities. Another drawback stems from the simplified 2D motion models that they employ. The projection of 3D motion on the 2D image plane depends on certain characteristics of the observer (e.g. focal length of the system), on the parameters of the relative 3D motion between an object and the observer and, on the depth of the scene. Therefore, discontinuities in the computed 2D motion field are not only due to 3D motion discontinuities (i.e. independently moving objects), but also due to depth discontinuities. The simplified 2D models that are employed by the previous methods do not take into account the depth of the scene. This is equivalent to assuming that depth is constant, which is in principle an unrealistic environmental assumption. For this reason, all the above methods perform well in scenes where the depth variation is small compared to the distance from the observer. In the case of scenes with large depth variations, these methods cannot provide reliable results.

2.3.2 3D methods

In an effort to overcome the inherent problems of employing 2D models of motion, the more accurate 3D models have been employed. By employing 3D models the problem becomes much more complicated because extra variables are introduced regarding the depths of the scene points. Thus, certain assumptions are made in order to provide additional constraints for the problem. Common assumptions of existing methods are related to the motion of the observer, the structure of the scene in view, or both. In many cases, the methods employing 3D models depend also on the accurate computation of the optical flow (image flow) field.

Wang and Duncan [148] present an iterative method for recovering the 3D motion and structure of independently moving objects from a sparse set of velocities obtained from a pair of calibrated, parallel cameras. Based on stereo correspondences, the method extracts the structure of the viewed scene and identifies independently moving objects by examining the consistency between the stereo disparities, the left/right image flows and the estimated 3D motion components. Jain [81] has considered the problem of independent 3D motion detection by an observer pursuing translational motion. In addition to imposing constraints on egomotion (the observer's motion cannot have rotational components), knowledge of the direction of translation is required. Adiv [3] performs segmentation by assuming planar surfaces undergoing rigid motion, thus introducing an environmental assumption. In order to determine the motion parameters of the moving plane and to group the motion vectors, he employs a Hough transform.

An interesting method is proposed by Lobo and Tsotsos [90] which is based on the Collinear Point Constraint (CPC). The method is basically an egomotion estimation method. If, however, there exist some independently moving object(s) that do not cover much of the visual field, their presence will not affect the estimation of egomotion Consequently, such objects can easily be detected because their flow will not be compatible with the one suggested by the estimated egomotion parameters.

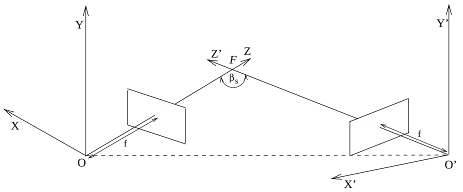


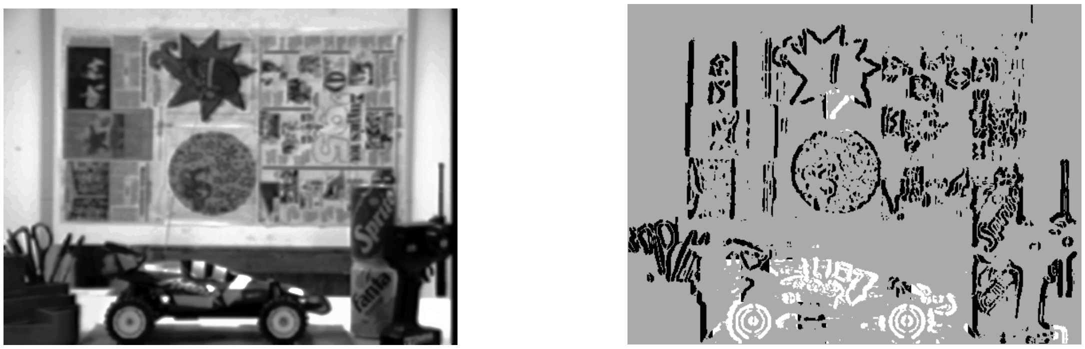
Figure 2.2: *Binocular Observer.*

In Torr and Murray [138], the problem of independent motion detection is approached by exploiting the epipolar constraint. More specifically, the problem of motion segmentation is transformed into one of finding a set of fundamental matrices which optimally describe the observed temporal correspondences. Thompson and Pong [136] derive various principles for detecting independent motion when certain aspects of the egomotion or of the scene structure are known. Although it is an inspiring work, the practical exploitation of the underlying principles is limited because of the assumptions they are based on and other open implementation issues. Finally, Thompson et al. [135] employ the rigidity constraint in order to detect independently moving objects based on robust regression, applied to point correspondences that have been established under an orthographic projection model.

Argyros et al. [11] present a method that assumes a binocular observer, illustrated in Fig 2.2. The stereoscopic information is used to segment an image into depth layers, i.e. regions (not necessarily connected) in which depth variations are small compared to the distance from the observer. The definition of depth layers enables the solution of 2D motion segmentation within each depth layer. After that, motion information from the various depth layers is combined to provide a full 3D motion segmentation of the whole scene. In Argyros et al [12], an even more qualitative approach is taken. Qualitative functions of depth estimated from stereo and motion are extracted in image patches. It is shown that if only one rigid motion is present in this patch, then the two depth functions ought to have a linear relationship. If, however, this linear relationship does not hold, then this is due to the presence of independent motion. The application of the latter method to an image sequence featuring a moving “toy-car” is presented in Fig 2.3. In Fig 2.3a, one frame from the original toy-car sequence is shown. Fig 2.3b presents the motion segmentation result, where the independent motion of the toy-car has been marked white; black pixels correspond to stationary points, and gray pixels indicate points where no reliable motion information (normal flow) could be computed.

2.3.3 Direct methods

Methods that fall in this category are the ones that do not rely directly on the estimation of the motion parameters in order to provide answers to the problem of independent motion detection, but rather exploit geometrical properties of the normal flow field. Thus, the solution of the structure from motion problem is by-passed. One such method has been developed by Sharma and Aloimonos [124], which, as in the case of [81], assumes known translational egomotion. For such an egomotion, it is known that normal flow vectors are constrained to lie on specific half planes. Normal flow vectors that violate this constraint can only be due to independent motion. Similarly, Nelson [105] presents a method for independent motion detection, which is also based on the normal flow field. It requires



(a) (b)

Figure 2.3: *Motion discontinuities detection for the "toy-car" sequence; (a) one frame from the "toy-car" sequence, (b) independent motion detection, marked by white.*

qualitative knowledge of egomotion parameters and assumes upper bounds on the depth of the scene. Based on the above, normal flow vectors are constrained to lie on specific areas of the plane. Normal flow vectors that do not meet this constraint are reported as due to independent motion.

2.3.4 Conclusions

The independent motion detection, being qualitative by nature, seems to be somewhat more tractable for visual navigation than, for example, egomotion estimation in the general case. In many cases, even the simplistic 2D methods produce acceptable results, when the depth variations of the scene are negligible. In the opposite case, 3D methods may provide more accurate results, since they rely on realistic models of the viewed scene. Such methods usually characterize independent motion by detecting inconsistencies in some appropriate function of the scene structure and/or motion.

A different approach is taken by the so-called *direct* methods, which exploit geometrical properties of the normal flow field. Such properties manifest themselves in the form of constraints that normal flow vectors should comform to; violations of such constraints are attributed to independently moving objects.

2.4 Estimating time-to-contact

For a moving observer various visual measures are useful while navigating through a scene, be it the ambulant human observer, as studied in [83, 84], or be it the autonomous robot, as studied in [72, 20]. One such very useful measure is "time-to-contact" to certain features in the visual field, i.e. the current estimate of time, that will elapse before the feature in question either hits or passes by the observer.

2.4.1 Definitions

When referring to optical flow in this section, we follow the definition in Section 2.1, whereby the term (in the absence of various dynamic illumination effects) refers to image flow *or* normal flow, *or both.*

The time-to-contact t_c (or *time-to-collision*, or *time-to-impact*) between the camera and a scene point is defined as $t_c = Z/W$, where Z is the depth of this point and W the relative forward translational speed of the camera with respect to the point. The estimation of time to contact is very important for navigational purposes. If we assume that the relative motion is constant in time, Z/W expresses the time left until the target will come into contact with the lens plane. Note that, as defined, the time-to-contact is neither a pure distance measure, nor a pure speed measure but a measure that is directly related with the observer's reaction time[4].

As will be apparent below, there are two classes of approaches, for estimating time-to-contact, one image *velocity field* based, and another which is based on *trajectories* projected onto the image plane. In the former case, divergence or other local measures of the image velocity field may yield an estimate of Z/W, whereas in the latter, accelerations along the projected trajectory gives us an estimate of Z/W. Denote the planar image velocity field by (u, v) and the total time derivative $(du/dt, dv/dt) = (a, b)$. Then, the *optic acceleration* is defined by:

$$a = \frac{\partial u}{\partial x}u + \frac{\partial u}{\partial y}v + \frac{\partial u}{\partial t} \tag{2.4}$$

$$b = \frac{\partial v}{\partial x}u + \frac{\partial v}{\partial y}v + \frac{\partial v}{\partial t} \tag{2.5}$$

Equations (2.4) and (2.5) are similar to expressions that appear in fluid dynamics and dynamic meteorology [69, Ch.2], where the first two terms on the right-hand sides, are called *advected velocity field.*

The above equations illustrate clearly the difference between, on one hand *divergence*, i.e., $\partial u/\partial x + \partial v/\partial y$, and on the other, *optic accelerations* (a, b).

2.4.2 Time-to-contact estimation approaches

There is a number of research efforts towards estimating the time-to-contact [39, 107, 120, 133]. More specifically, Cipolla and Blake [39] proposed estimating the parameters of an affine model of the motion field of a tracked object. Based on this, the method can extract information about time-to-contact. Tistarelli and Sandini [120] have presented a technique for estimating time-to-contact based on the properties of a space-variant optical sensor, the polar and log-polar camera. Nelson and Aloimonos [107] provide analysis and experiments on estimation of time-to-contact based on divergence. Subbarao [133] has derived bounds on the time-to-contact by using the first-order derivatives of the image

[4] For example, an obstacle at a distance of 3m from an observer that approaches it with a speed of 5m/s is more dangerous than an obstacle at a distance of 1.5m from an observer that approaches it with a speed of 1m/s, in the sense that, in the latter case, the observer's reaction time is larger.

flow. All these methods use divergence, or other local measures based on the velocity field, to compute the time-to-contact.

Arnspang et al. [16] provide some simple time-to-contact estimators using isolated points and curve segments based on image flow and optic acceleration, an approach that is discussed further, below.

A third branch of approaches that has been proposed is based on the utilization of constraints from fixation and tracking [57]. The change of rotation in order to accurately track a target, is related to the target's change in depth which in turn is related to the time-to-contact.

2.4.3 Time-to-contact estimation based on optic acceleration

While optical flow fields have been extensively studied for decades now (see [83, 152] for overviews), the use of optic acceleration has been suggested [13], but not directly used in time-to-contact estimates. It has been indicated experimentally, that animals and humans actually use optic accelerations in their visual flow field for every day behavioural purposes. Such experiments may be conducted by ambulant observers, carrying a virtual reality helmet. While the ambulant observer moves in voluntary ways, the actual visual scene, which the observer would have experienced without a virtual reality helmet, may be displayed on an internal screen in the helmet, or a modified visual scene may be displayed on this internal screen. Experiments with modifying optic accelerations in the displayed scene may then be performed. It has for example been shown, that the somewhat cycloid like motions, humans perform due to body saccades when 'walking straight', and the thus induced optic accelerations in the visual field, are of utmost importance for humans in order to estimate distances to obstacles (Insight Project experiments, [78]).

Within a robot navigation paradigm, optic accelerations may be employed in order to determine time-to-contact directly without any need to calibrate focal length or camera orientation relative to camera motion, nor to know the slope of a surface in question. Moreover, as will be shown, the normal flow and acceleration of an approaching curve element is sufficient to determine time-to-contact with the curve point in question. In this sense it is seen that the classic aperture problem, as discussed throughout the literature [72, Ch. 12], is not an obstacle for estimating time-to-contact. Consider a point moving with a constant spatial velocity in front of a perspective camera, as indicated in Fig. 2.4, showing some visual projections of a point with equal time between plots.

With the symbols of Fig. 2.4, we have the coordinate relations

$$fX + xZ = 0, fY + yZ = 0 \tag{2.6}$$

where f is the camera focal length, (X, Y, Z) are spatial coordinates of the point and (x, y) projected coordinates on the image plane. By taking time derivatives of these relations until first and second order we obtain

$$fU + xW + uZ = 0 \tag{2.7}$$

$$fV + yW + vZ = 0 \tag{2.8}$$

$$2uW + aZ = 0 \tag{2.9}$$

$$2vW + bZ = 0 \tag{2.10}$$

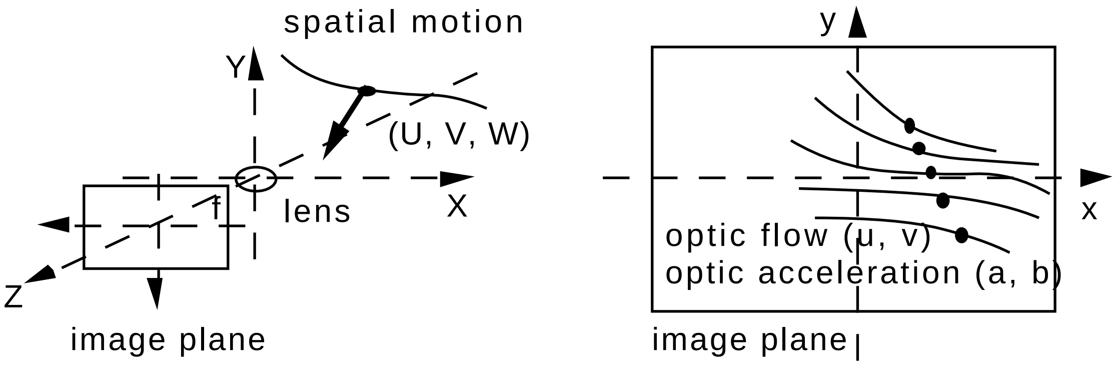


Figure 2.4: *Geometric and Kinematic Definitions.*

where (U, V, W) is spatial velocity of the point in question, (u, v) is image flow of the projected image point and (a, b) is the optic acceleration. From these relations we obtain simple expressions for $t_c = -Z/W$ as:

$$t_c = 2u/a \quad (2.11)$$

$$t_c = 2v/b \quad (2.12)$$

One may note, that the estimates of Eqs. (2.11),(2.12) are independent of the choice of coordinate axes on the image plane, which may be oriented and scaled arbitrarily. Furthermore the unit of time may be chosen freely. One obvious choice is to orient a coordinate axis on the image plane along the visual projected direction of motion and use the length of the image flow vector in this direction as metric unity in the image plane. Likewise, an obvious choice for time unit will be the camera shutter pulse, i.e. the time between each image taken in the sequence at hand. The measured image flow in this visual direction is consequently of magnitude 1 in this scale, and if the observed optic acceleration in the visual direction of motion is of magnitude a in this scale, then time-to-contact, expressed in units of the fixed camera shutter pulse, is given by:

$$t_c = 2/a \quad (2.13)$$

Using normal flow and acceleration

Consider Fig. 2.5 for interpreting the estimates in Eqs. (2.11),(2.12) in terms of normal flow and acceleration for a curve element. Since the image flow vector (u, v) and the optic acceleration vector (a, b) are directed along the same line in the image plane, the ratio of their length, used to estimate time-to-contact in Eqs. (2.11),(2.12) is the same as the ratio of their projection to the line, which is normal to the curve at the point in question:

$$t_c = 2u_N/a_N, \quad t_c = 2v_N/b_N \quad (2.14)$$

where (u_N, v_N) is the normal flow vector and (a_N, b_N) is the normal acceleration vector. Consequently only calculation of the normal flow and acceleration is needed in order to estimate time-to-contact at a chosen point on the curve in question.

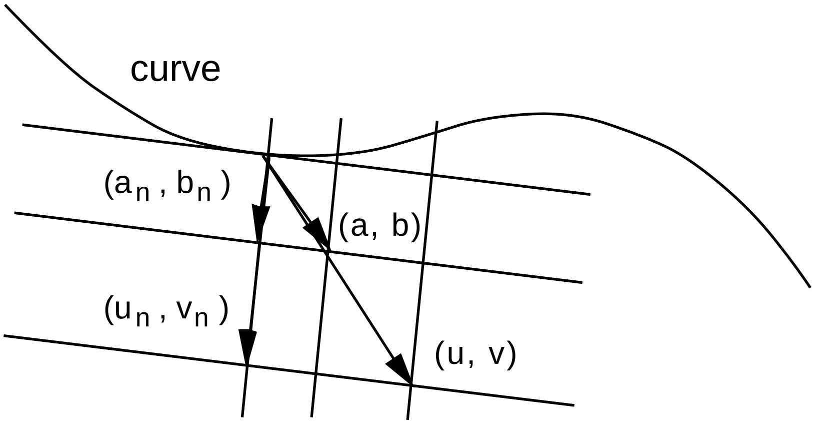


Figure 2.5: *Using Image Flow or Normal Flow, i.e., image velocity vectors versus image velocity components.*

Interpolating estimates during tracking

As a supplement to estimating time-to-contact from temporal local values of image flow and acceleration, the object in question may be tracked for some time and several estimates of time-to-contact obtained at different times. These estimates then in general vary with time, however in meaningful situations not in a priori unexpected ways. Note, that an estimate of t_c basically make sense only when a constant translation of the object in view, relatively to the observer may be assumed. If an unknown acceleration is involved, an estimate of t_c, based on the present motion situation hardly make sense. Then note, that during a constant translation estimates of t_c should vary linearly with time, i.e. one second later a new estimate of t_c should be one second smaller than the previous estimate ot t_c. This suggest a simple and robust interpolation, where a line of slope -1 is fitted in a time versus t_c diagram to the estimates at hand, obtained so far. The crossing point of the time axis and the fitting line will be the overall estimate of t_c. As the object in question approaches, it turns out that the individual estimates, based on image flow and optic acceleration becomes more accurate, meaning that the fitted line and its crossing point with the time axis becomes more accurate, the longer the object is tracked and the closer it gets to the observer [16]. This seems to be a meaningful and useful property of the technique.

2.4.4 Analysis — open questions

In this section two questions are raised, pointing out present research topics, concerned with time-to-contact and autonomous navigation, which constitute subjects of current research.

Non-differential estimates based on global templates

From the observations above, concerning temporal repeated estimations of time-to-contact, the question may be raised as to what kind of image sequence data, belonging to different instants of time may be used for providing a robust estimate of time-to-contact. Traditionally, time-to-contact has been estimated from purely local values of divergence or as above, purely local values of image flow and optic acceleration (u, v, a, b) at a certain time $t = 0$. Above, the basic projective relationships (Eqs. (2.7)-(2.10)) relating spatial position (X, Y, Z) and motion (U, V, W) with camera focal length f, image coordinates (x, y), image flow (u, v) and optic acceleration (a, b) are obtained by differentiating the basic coordinate transformation, Eq. (2.6), in time. This introduces the differential image sequence quantities such as (u, v, a, b), which are hard to estimate properly. Approaches based on such quantities may then introduce uncertainties in the estimates of Eqs. (2.11)-(2.13) and (2.14) of time-to-contact. If, instead, the basic coordinate transformation (2.6) are written as functions of time, evolving from a certain time $t = 0$ as

$$f(X0 + tU) + x(t)(Z0 + tW) = 0 \tag{2.15}$$

$$f(Y0 + tV) + y(t)(Z0 + tW) = 0 \tag{2.16}$$

we may by diving with spatial velocity along the optic axis W obtain a linear system in the quantities $X0/W, Y0/W, Z0/W, U/W$ and V/W, of which the latter three quantities are time-to-contact at time $t = 0$ and focus of expansion coordinates:

$$fX0/W + ftU/W + x(t)Z0/W + x(t)t = 0 \tag{2.17}$$

$$fY0/W + ftV/W + x(t)Z0/W + x(t)t = 0 \tag{2.18}$$

In such a linear system based on Eqs. (2.17)-(2.18), the coefficients are image coordinates $(x(t), y(t))$ at times $t = -1, 0, 1$. The system does not contain differentially defined quantities such as image flow and optic acceleration. An estimate of time-to-contact will in this case be based directly on image coordinates of the projected object, tracked over time, and will not be based on differential quantities like image flow and optic acceleration. A closed form solution for t_c in terms of a "temporal global t_c-template" can be derived by solving the resulting system of equations:

$$t_c = (x1 - x - 1)/(x1 - 2x0 + x - 1) \tag{2.19}$$

$$t_c = (y1 - y - 1)/(y1 - 2y0 + y - 1) \tag{2.20}$$

Preliminary experiments indicate that such temporal global estimates of time-to-contact are very robust indeed, compared to traditional temporal local estimates. Current efforts are directed towards employing such estimates in visual robot navigation tasks.

Helpful off-centre camera rotations

We further note, that in case there is no visual motion for a certain point, it may either actually not be moving, or it may be bound to hit you in the optic centre. Voluntarily rotating the camera around a point *different from the optic centre* may however sort out spatially stationary points from potentially colliding points[5]. Active vision experiments of this kind seem worthwhile doing.

[5] Such rotations are actually performed by humans during saccades, where the eyeball is rotated around a point, which is not the optic centre of the eye lens.

Time-to-contact estimates as a stereo cue

Going beyond the mere purpose of estimating t_c in order to use it for an obvious navigation purpose, t_c may be of use as a cue for stereo matching and hence facilitate depth calculations. Consider for example an ordinary stereo set up with parallel optic axes and a common lens plane for the two stereo cameras. Time-to-contact with an object, which is constantly translating, relatively to this stereo rig, could be estimated with the above techniques in either camera individually. Since the estimate for physical reasons should be identical for the two cameras, the t_c could be used for choosing candidates in a subsequent stereo match procedure. Traditionally, stereo paradigms are based on 2D image features, which have not been interpreted as a parameter of the 3D scene, before the stereo matching and depth map computations takes place.

2.4.5 Conclusions and remarks

A number of general remarks can be drawn for the approaches presented above for time-to-contact estimation; more specifically:

- For the methods mentioned which use divergence to compute time-to-contact, there are two basic disadvantages involved:
  - The computation based on divergence depends on the spatial derivatives of flow which may be hard to estimate accurately from a sequence of real images.
  - Divergence does not only depend on the time-to-contact, but also on on the translational component of motion that is parallel to the image plane and the orientation of the surface in the field of view [107].
- One problem with the approaches that are based on the utilization of constraints from fixation and tracking [57], is that they require the robot to keep track of its own exact motion during tracking; this is a difficult task if the robot is considered not to be static but moving itself.
- The acceleration-based approach has no problems with surface orientation, since it is trajectory based. However, some additions for dealing with camera rotations seem important. Accelerations can be replaced with second differences, and in the case of rectilinear moving objects, these two entities (as mentioned) coincide.
- Off-center camera rotations may help to avoid objects coming at a mobile platform, heading for the optic centre.

Summarizing, a common branch of methods for estimating time-to-contact use properties of optical flow fields, image flow fields, which may be hard to estimate correctly. Furthermore, the use of some divergence based methods assume knowledge of the underlying surface, producing the optical flow. However, when combining image flow and optic acceleration for estimating time-to-contact, no knowledge of an underlying surface is needed, but the problems with estimating image flow remains. Recently, methods have been suggested, using normal flow fields only and thus circumventing the classic aperture problem and at the same time using less computing power. Still, the problem of calculating differential quantities remain in these normal flow based techniques.

Current techniques, using coordinate based 'time-to-contact templates', rather than differential measures, or measures based on two camera views over time, seem to offer enhanced robustness to visual navigation. Based on these, classic stereo camera paradigms can be reformulated to use the insight of two camera estimates of time-to-contact to provide a general scene layout determination during a visual navigation.

2.5 Object tracking

Object tracking provides a good basis for estimation of 3D motion and/or structure from time varying images and thus, provides a rich support to the analysis of time-varying environments. It is clearly one of the most important issues within a wide class of applications including robotics, automated surveillance, traffic monitoring, and medical imagery.

2.5.1 Definitions

In tracking, standard techniques are used quite often. One branch comprises techniques using *Kalman filters* [10], or α, β trackers [30]. We refer to the above references for tutorial introductions to these filters. For our purposes it suffices to mention that the Kalman filters share many properties with *recursive least squares*, i.e., the iteration formula obtained by adding an observation, at each time step t, calculating least squares parameters. A measured position of image points (for simplicity the x-position) is, e.g., expressed as

$$\hat{x}_t(t-1) = \hat{x}_{t-1} + k \cdot (\hat{x}_{t-1} - x_{t-1}) + \hat{v} \tag{2.21}$$

which reads: The predicted value at time $t-1$ for time t $(= \hat{x}_t(t-1))$, is the previous position estimate $(= \hat{x}_{t-1})$ plus a correction due to the discrepancy between predicted and measured position at $t-1$, namely $\hat{x}_{t-1} - x_{t-1}$. The coefficient k is estimated recursively from data, at each time instant t and $\hat{v}_t$ is estimated velocity (in the x-direction) at t. This is also called the *prediction step* of Kalman filtering. In automatic image analysis, the points or features to be tracked must be identified, which amounts to the *matching step*. A time prediction of the new position of a point at time t, should facilitate matching between frame $t-1$ and frame t for that point.

A *snake*, is a curve γ in the image plane which deforms over time according to some constraints to stay close to some feature or object (say the edge around a closed image region or an independently moving object). Usually, some smoothness constraints on the deformation process are invoked. In other words, *snakes*[6] are deformable curves with some typical constraints governing their deformation. It is essentially a kind of interpolation technique, over space and time.

[6] Sometimes called *active contours.*

2.5.2 Early approaches

Early investigations in tracking were concerned with sparse image features such as points [121, 77, 43, 60, 64] or edge-line segments [47, 45, 61]. As a representative example the work described in [47] presents the development and implementation of a line segment based tracker. The tracking approach combines prediction and matching. The prediction step is a Kalman filtering based approach that is used in order to provide reasonable estimates of the region where a matching process has to search for a possible match between line features (also called line tokens). Correspondence in the local search region, defined by parameters in the algorithm, is done through the use of a similarity function based on *Mahalanobis* distance between carefully chosen attributes of the line segments. Different implementations for the prediction step are considered and an α, β tracker with decoupled equations is presented in order to deal with real time applications. The efficiency of the proposed approach is illustrated in several experiments that have been carried out on noisy synthetic data and real scenes obtained from the INRIA mobile robot.

2.5.3 Snake and region based approaches

As the use of vertices or edges may lead to a sparse set of trajectories and can make the procedure very sensitive to noise as well as to occlusions, more complex tracking models have been proposed to take into account global primitives. Due to the detailed quantitative information that can be provided about their evolving shape, snake based contour tracking and region model-based tracking approaches have been found to be very well suited to the tracking of rigid and non-rigid objects.

Using snakes for tracking objects in sequence of images was originally proposed by Kass et al. in [82], where an application to track the contours around a speaker's lips was considered. Since this pioneering work, active contour models have been largely used in order to develop powerful tracking systems, and many applications have demonstrated the effectiveness of this idea. The work described in [134] bridges *snake techniques* and *Kalman filtering*—the latter known to optimally integrate visual measurements from multiple sources and over time—to improve the accuracy of state estimates.

Some authors have proposed the use of *deformable B-spline curves* for tracking occluding contours of 3D objects (e.g. see the previously mentioned work (Section 2.4.2) by Cipolla and Blake in [39]). Experiments were done with a camera attached to a moving robot arm. The approach taken in [39] provides more realistic and global descriptions of curved edges than points or segments. Closed B-Spline snakes are used to localize and track closed image contours. Divergence and deformation are then estimated by tracking closed image contours with B-Spline snakes. This divergence and deformation are then successfully used to estimate time-to-contact (time-to-collision) and surface orientation. Similar approaches are also described in [40, 41].

In [46], the design of a real time dynamic contour that run at video rates is presented. The authors combine the use of parametric snakes defined as B-splines with parallel computing in the form of a small transputer network (11 processors) to achieve a video rate tracking performance for several contours simultaneously. Applications to surveillance of people and vehicles, robotics path-planning and grasp-planning have been considered.

Bascle et al. [25] describe an approach to contour tracking, using deformable curves with *motion constraints*, the deformable curve being constrained by a motion model. Several visual motion models are considered, corresponding each to a different type of 3D motion. The resulting curve deforms freely in order to better fit the image contours. This tracking process is carried out by minimizing a functional, specifically formulated for this problem. The performance of this curve-tracking method and its robustness to occlusion are illustrated by experimental results obtained on real images, including curved objects. Three interesting applications of this tracking approach are proposed, which aim at recovering 3D information from tracking. First, time-to-contact is estimated, using the visual motion model evaluated by tracking. Second, the spatio-temporal surface of the moving contour is constructed. This surface is a valuable piece of information, because its characteristics are related to 3D structure and motion [55]. Third, if tracking is performed between two calibrated binocular images, the corresponding 3D curved edge can be reconstructed. These applications are illustrated by real examples, in the above-mentioned references.

Tracking complex primitives along image sequences has also been addressed by integrating snake-based contour tracking and region-based motion analysis. The complex primitive consist both of contours, and local regions with spatio-temporal image gradient fields inside them. Bascle et al. [24], describe an approach where a snake first tracks the region outline and performs segmentation. Then the motion of the extracted region is estimated by a dense analysis of the apparent motion over the region, using spatio-temporal image gradients. Finally, this motion measurement is filtered to predict the region location in the next frame, and thus to guide (i.e. to initialize) the tracking snake in the next frame. Therefore, these two approaches collaborate and exchange information to overcome the limitations of each of them. The method is illustrated by experimental results on real images.

Following the work performed in [25, 24], [26] describes a complementary approach to tracking of complex shapes through image sequences, that combines deformable region models and deformable contours. A new deformable region model is presented; its optimization is based on texture correlation and is constrained by the use of a motion model, such as rigid, affine or homographic. The use of texture information (versus edge information) noticeably improves the tracking performances of deformable models in the presence of texture. Then the region contour is refined using an edge-based deformable model, in order to better deal with specularities and non planar objects.

Recently, a new level-set based framework for detecting and tracking moving objects in a sequence of images has been presented [112, 113, 35]. Using a statistical approach, where the *inter-frame* difference is modeled by a mixture of two Laplacian or Gaussian distributions, and an energy minimization based approach, the authors in [113] reformulate the motion detection and tracking problem as a front propagation problem. The Euler-Lagrange equation of the designed energy functional is first derived and the flow minimizing the energy is then obtained. Following the work by Caselles et al. [36] and Malladi et al. [94, 93], the contours to be detected and tracked are modeled as geodesic active contours evolving toward the minimum of the designed energy, under the influence of internal and external image dependent forces. Using the level set formulation scheme of Osher and Sethian [110], complex curves can be detected and tracked, and topological changes for the evolving curves are naturally managed. In order to reduce the computational time required by a direct implementation of the formulation scheme of Osher and Sethian [110], a new approach that exploits the best aspects of the *Narrow Band* [1] and *Fast Marching* [122] methods has been developed recently by the same authors and implemented. The idea of applying the curve evolution theory to the tracking problem has been also recently presented by Caselles in [35]. However this sequentially three step approach is very different from the unified approach presented in [113]. Following their previous work on geodesic active contours, the authors in [35] first start by detecting the contours of the objects to be tracked. An estimation of the velocity vector field along the detected contours is then performed using a completely separate approach, and finally another PDE is designed to move the contours to the boundary of the moving objects. These contours are then used as initial estimate of the contours in the next image, and the process is repeated.

These articles clearly show that the level set method can also be applied effectively for solving important vision problems, such as motion detection and tracking, and provide promising results. With this method, complex curves can be tracked and topological changes for the evolving curves are naturally managed. The final result is relatively independent of the curve initialization.

Figures 2.6 and 2.7 illustrate the results obtained on various real world video sequences, using the approaches described in [112, 113].

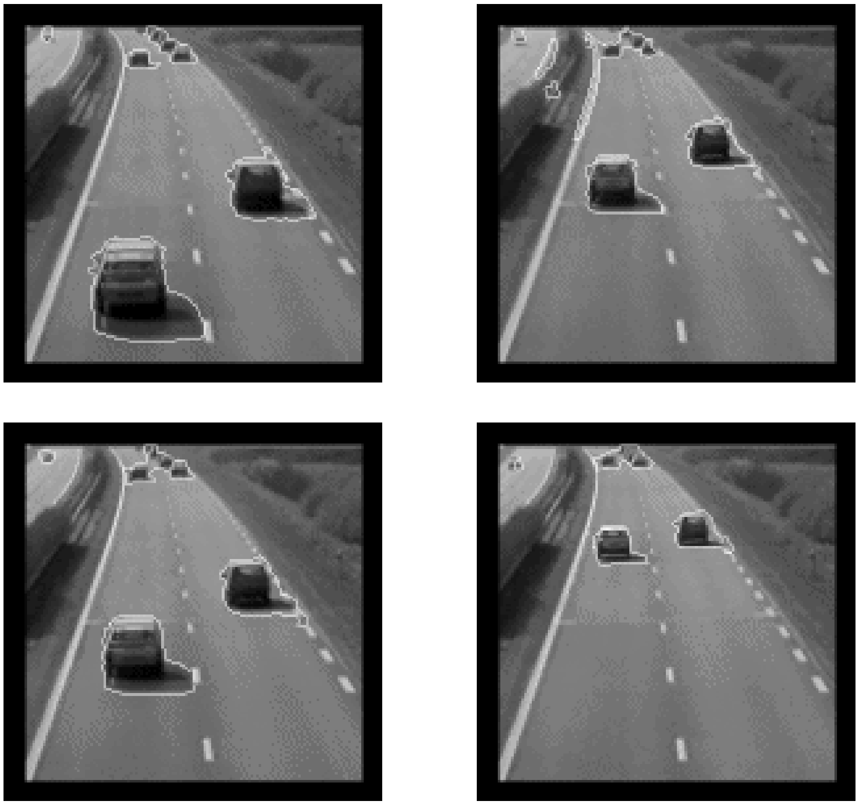


Figure 2.6: *Highway sequence tracking. 'Snake' marked by white around moving objects.*

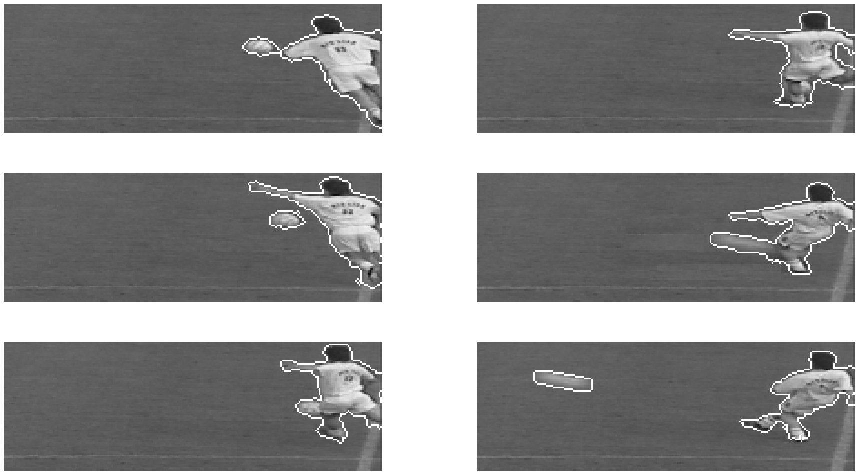


Figure 2.7: *Football sequence tracking.*

2.5.4 Real-time tracking systems

The above described advances in object tracking, combined with the recent developments in computer hardware performances, have made possible the development of low-cost, high-reliability and real time systems for model based motion tracking. Hence, exploiting the spatio-temporal continuity and using the Kalman filtering framework for the real time control of vehicles from moving image sequences, Dickmanns [52, 49, 53], [50, 51] successfully demonstrated the ability for guiding an experimental vehicle at its maximum speed of 98 km/hour on an empty AutoBahn in 1987, by tracking the road boundaries with sets of correlation type feature detectors.

In [92], a computer vision system has been developed for real-time motion tracking of 3D-Objects with many degrees of freedom, while running on relatively inexpensive hardware. Rates of 3 to 5 frames per second have been obtained, and the author claims that with moderate increases in computer speeds, as are already available with low-cost parallel systems of microprocessors, such a system could be used to track objects at 30 or 60 frames per second and provide real time visual input for robots.

In [44], the road detection system SCARF (Supervision Classification Applied to Road Following), which actively tracks the road location in a sequence of color images is described. A road surface likelihood image is first computed and then used by a model mathcing algorithm to select the road or intersection model that best matches this surface. SCARF has successfully driven the `Navlab` mobile robot on numerous occasions. `Navlab` is further discussed, from the point of view of learning, in Section 5.5.

One can also refer to the DROID system [65] that uses controlled egomotion to construct a 3D geometric description of the scene. It identifies and tracks features points such as corners and uses triangulation with Kalman filtering to estimate a depth map that can be used for obstacle avoidance and detection of navigational landmarks. The completion of a real time (5Hz) implementation of the DROID system was in the process to be done.

A real-time tracking system of image regions with changes in geometry and illumination is developed in [63]. The implemented system accommodates affine distortions and non-orthogonal illumination bases. It can perform frame rate tracking of human face (100 by 100 pixels) image regions.

2.5.5 Multitarget tracking

In traffic scenes, machine vision based surveillance systems have to cope with several moving objects which can interact with each other. Multitarget tracking and occlusion detection are thus required. Rao [116] addresses the problem of tracking *multiple bodies* while maintaining the continuity of each path in the presence of uncertainty. The author highlights the issues which influence the development of data association policies and describes the most commonly used solutions to the problem. An effective solution is demonstrated in the case of point objects observed by a multiple camera surveillance system. Applications involving tracking and data association can also be found in [21, 30, 62]

In Koller et al. [86], a contour tracker based on intensity and motion boundaries is designed for robust detection and tracking of multiple vehicles in road traffic scenes. The authors exploit the special camera pose, mounted above the road on a bridge and looking down along the driving direction, to decide about partial occlusion of vehicles. In [59], the authors improve their image sequence analysis system by using an explicit model-based recognition of 3D occlusion situations. Similarly, a purely image domain approach without requiring 3D information is presented in [96]. Tracking multiple objects is also treated in [118] and in [119] where an active tracking strategy for monocular depth inference over multiple frames is developed. Another related work is the model-based object tracking approach developed for traffic scenes [85].

2.5.6 Related applications

The work described in [76] addresses the problem of tracking non-rigid objects in complex scenes. Constrained deformable superquadrics are used and combined with non-rigid motion tracking in [95]. A visual analysis of high DOF articulated objects is developed in [117] and an application to hand tracking is considered. Finally, one can refer to the article [79], where an original and efficient stochastic based approach is developed for tracking rigid and non-rigid curves in dense visual clutter; this approach is shown to succeed in some cases where a Kalman filter fails.

Other applications of tracking are those related to the recovery of 3D motion and structure from stereo and 2D token tracking cooperation [102, 103, 146, 162, 163, 160, 161]. Medical imaging is another domain where tracking has been employed successfully in many applications. The work in [17] describes a medical imaging system for tracking anatomical structures in time sequences of noisy ultrasound echocardiographic images, using a sonar-space filtering method combined with a snake-based tracker.

2.5.7 Conclusions and remarks

Visual tracking of objects is a technical discipline, important to applications within automated surveillance, traffic monitoring, vehicle navigation, and also within computer assisted analysis of medical images.

Early tracking paradigms deal with moving points and line segments. The introduction of snake- and region-based approaches significantly widened the problem domains, for which tracking paradigms could be formulated and also proved to work in real-time applications - examples including tracking of deformable contours in 3D, for example lip movements, or, deformations of selfoccluding contours of smooth objects. Another example is tracking of the outline of cars, from time to time partly occluded by buildings or other traffic components, to be surveilled for some time. A third example is anatomical structures, tracked in echocardiac image sequences for diagnosis purposes.

Tracking of objects is a fundamental process in many visually based applications, including the automatic visual navigation. At present, a main stream of research is devoted to establishing techniques, which will allow simultaneous tracking of *multiple objects* or of objects breaking into several parts.

Bibliography

[1] D. Adalsteinsson and J. A. Sethian. A Fast Level Set Method for Propagating Interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995.

[2] E.H. Adelson and J.R. Bergen. Spatiotemporal Energy Models for the Perception of Motion. *Journal of the Optical Society of America A*, 2:284–299, 1985.

[3] G. Adiv. Determining Three Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(4):384–401, July 1985.

[4] G. Adiv. Inherent Ambiguities in Recovering 3D Motion and Structure from a Noisy Flow Field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(5):477–489, May 1989.

[5] Y. Aloimonos and C.M. Brown. Direct Processing of Curvilinear Sensor Motion from a Sequence of Perspective Images. In *Workshop on Computer Vision: Representation and Control*, pages 72–77, 1984.

[6] Y. Aloimonos and C.M. Brown. On the Kinetic Depth Effect. *Biological Cybernetics*, 60:445–455, 1989.

[7] Y. Aloimonos and Z. Duric. Estimating the Heading Direction using Normal Flow. *International Journal of Computer Vision*, 13(1):33–56, 1994.

[8] P. Anandan. A Computational Framework and an Algorithm for the Measurement of Visual Motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[9] P. Anandan and R. Weiss. Introducing a Smoothness Constraint in a Matching Approach for the Computation of Optical Flow Fields. In *3rd International Workshop on Computer Vision: Representation and Control*, pages 186–194, 1985.

[10] Brian D.O. Anderson, John M. Moore. *Optimal Filtering.* Prentice-Hall, Electrical Eng. series, New Jersey, 1979.

[11] A.A. Argyros, M.I.A Lourakis, P.E. Trahanias, and S.C. Orphanoudakis. Independent 3D Motion Detection Through Robust Regression in Depth Layers. In *British Machine Vision Conference (BMVC '96), Edinburgh, UK*, September 9-12 1996.

[12] A.A. Argyros, M.I.A Lourakis, P.E. Trahanias, and S.C. Orphanoudakis. Qualitative Detection of 3D Motion Discontinuities. In *IROS '96, Tokyo, Japan*, November 4-8 1996.

[13] J. Arnspang. Optic Acceleration. In *Proc. of International Conference on Computer Vision*, Tampa, December 1988.

[14] J. Arnspang. Notes on local determination of smooth optical flow and the translational property of first order optic flow. DIKU tech. report 88.1, 1988.

[15] J. Arnspang. Motion constraint equations, based on constant image irradiance. Image and Vision Computing, Vol. 11, No.9, Nov. 1993.

[16] J. Arnspang, K. Henriksen, and R. Stahr Estimating Time to Contact with Curves, avoiding Calibration and Aperture Problem. In *Proc. of CAIP95*, Prag, Springer Verlag, pages 856–861, Sept. 1995.

[17] Nicholas Ayache, Isaac Cohen, and Isabelle Herlin. Medical image tracking. In Andrew Blake and Alan Yuille, editors, *Active Vision*, chapter 17, pages 285–302. The MIT Press, 1993.

[18] S. Ayer, P. Schroeter, and J. Bigun. Segmentation of Moving Objects by Robust Motion Parameter Estimation over Multiple Frames. In *European Conference on Computer Vision*, 1994.

[19] H.H. Baker, R.C.. Bolles, The weaving wall. In *Proc of CVPR*, Ann Arbor, Michigan, 1988.

[20] D.H. Ballard, C.M. Brown. *Computer Vision.* Prentice Hall, 1982.

[21] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association.* Academic, New York, 1988.

[22] S.T. Barnard and W.B. Thompson. Disparity Analysis of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI 2:333–340, 1980.

[23] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[24] B. Bascle, P. Bouthemy, R. Deriche, and F. Meyer. Tracking complex primitives in an image sequence. In *Proceedings of the 12th IAPR International Conference On Pattern Recognition*, pages 426–431, Jerusalem, Israel, oct 1994.

[25] B. Bascle and R. Deriche. Stereo matching, reconstruction and refinement of 3-D curves using deformable contours. In *Proceedings of the 4th International Conference On Computer Vision*, Berlin, Germany, 1993.

[26] Bénédicte Bascle and Rachid Deriche. Region tracking through image sequences. In *Proceedings of the 5th International Conference on Computer Vision*, pages 302–307, Boston, MA, June 1995. IEEE Computer Society Press.

[27] F. Bergholm. Motion from Flow along Contours: A Note on Robustness and Ambiguous Cases. *International Journal of Computer Vision*, 3:395–415, 1989.

[28] F. Bergholm. On velocity estimation and mean square error. *Proc. of 8th Scandinavian Conference on Image Analysis*, Tromsoe, Norway, pages 1093-1100, May 1993.

[29] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996.

[30] S. S. Blackman. *Multiple-Target Tracking with Radar Application.* Artech House, Norwood, MA, 1986.

[31] M. Bober and J. Kittler. Estimation of Complex Multimodal Motion: An Approach Based on Robust Statistics and Hough Transform. *Image and Vision Computing*, 12:661–668, December 1994.

[32] P. Bouthemy and E. Francois. Motion Segmentation and Qualitative Dynamic Scene Analysis from an Image Sequence. *International Journal of Computer Vision*, 10(2):157–182, 1993.

[33] W. Burger and B. Bhanu. Estimating 3-D Egomotion from Perspective Image Sequences. *IEEE Trans. on Image Processing*, 12(11):1040–1058, Nov. 1990.

[34] P.J. Burt and C. Yen X. Xu. Multiresolution Flow Through Motion Analysis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 246–252, 1983.

[35] V. Caselles and B. Coll. Snakes in Movement. *SIAM Journal on Numerical Analysis*, 33:2445–2456, December 1996.

[36] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proceedings of the 5th International Conference on Computer Vision*, pages 694–699, IEEE Computer Society Press, Boston, MA, June 1995.

[37] C.J. Cheng and J.K. Aggarwal. A Two-stage Hybrid Approach to the Correspondence Problem via Forward Searching and Backword Correcting. In *International Conference on Pattern Recognition*, pages 173–179, 1990.

[38] T. Chin, W. Karl, and A. Willsky. Probabilistic and Sequential Computation of Optical Flow Using Temporal Coherence. *IEEE Transactions on Image Processing*, 3:773–788, November 1994.

[39] R. Cipolla and A. Blake. Surface Orientation and Time to Contact from Image Divergence and Deformation. In *Proc. 2nd ECCV, Santa Margherita Ligure, Italy*, pages 187–202, May 1992.

[40] Roberto Cipolla and Andrew Blake. The dynamic analysis of apparent contours. In *Proceedings of the Third International Conference on Computer Vision, Osaka, Japan*, pages 616–623, December 1990.

[41] Roberto Cipolla and Andrew Blake. Motion planning using divergence and deformation. In Andrew Blake and Alan Yuille, editors, *Active Vision*, chapter XII, pages 189–202. The MIT Press, 1993.

[42] J. C. Clarke and A. Zisserman. Detection and Tracking of Independent Motion. *Image and Vision Computing*, 14:565–572, 1996.

[43] I. Cohen, N. Ayache, and Sulger P. Tracking points on deformable objects using curvature information. In *Proc. 2nd ECCV, Santa Margherita Ligure, Italy*, May 1992.

[44] Jill D. Crisman. Color region tracking for vehicle guidance. In Andrew Blake and Alan Yuille, editors, *Active Vision*, chapter VII, pages 107–122. The MIT Press, 1993.

[45] J.L. Crowley and P. Stelmaszyk. Measurement and integration of 3-D structures by tracking edge lines. In O.D. Faugeras, editor, *Proceedings of the 1st European Conference on Computer Vision*, pages 269–280, Antibes, France, April 1990. Springer, Berlin, Heidelberg.

[46] Ruppert Curwen and Andrew Blake. Dynamic contours : Real-time active splines. In Andrew Blake and Alan Yuille, editors, *Active Vision*, chapter III, pages 39–58. The MIT Press, 1993.

[47] Rachid Deriche and Olivier Faugeras. Tracking line segments. *Image and Vision Computing*, 8(4):261–270, November 1990. (A short version appeared in the Proc. of the 1st ECCV.)

[48] U. R. Dhond and J. K. Aggarwal. Structure from Stereo - A Review. *IEEE Trans. on Systems, Man and Cybernetics*, 19(6):1489–1510, November/December 1989.

[49] E.D. Dickmanns. 4d-dynamic scene analysis with integral spatio-temporal models. In *Proc. ISSR'87*, pages 73–80, Santa-Cruz, 1987.

[50] E.D Dickmanns and V. Graefe. Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1:241–261, 1988.

[51] E.D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1:223–240, 1988.

[52] Ernst. D. Dickmanns. Expectation-based dynamic scene understanding. In Andrew Blake and Alan Yuille, editors, *Active Vision*, chapter 18, pages 303–335. The MIT Press, 1993.

[53] Ernst D. Dickmanns and Birger D. Mysliwetz. Recursive 3-D road and relative ego-state recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):199–213, February 1992.

[54] J.Q. Fang and T.S. Huang. Some Experiments on Estimating the 3-D Motion Parameters of a Rigid Body from Two Consecutive Frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:547–554, 1984.

[55] O. D. Faugeras and T. Papadopoulo. A theory of the motion fields of curves. *International Journal of Computer Vision.* To appear in 1993.

[56] C. Fermüller. *Basic Visual Capabilities.* PhD Dissertation, Center for Automation Research, University of Maryland, 1993.

[57] C. Fermüller and Y. Aloimonos. Tracking facilitates 3-D motion Estimation. *Biological Cybernetics*, 67:259–268, 1992.

[58] D.J. Fleet and A.D. Jepson. Computation of Component Image Velocity from Local Phase Information. *International Journal of Computer Vision*, pages 77–104, 1990.

[59] Thomas Frank, Michael Haag, Henner Kollnig, and Hans-Helmut Nagel. Tracking of oclluded vehicles in traffic scenes. In Springer-Verlag, editor, *Fourth European Conference on Computer Vision*, volume II, pages 485–494, Cambridge, UK, April 1996.

[60] J.P. Gambotto. Tracking points and line segments in image sequences. In *Proc. IEEE Workshop Visual Motion*, pages 38–45, Irvine, CA, March 1989.

[61] B. Giai-Checa, R. Deriche, T. Viéville, and O.Faugeras. Suivi de segments dans une séquence d'images monoculaire. Technical Report 2113, INRIA Sophia-Antipolis, France, December 1993.

[62] Per-Olof Gutman and Mordekhai Velger. Tracking targets using adaptive kalman filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 26(5):691–698, September 1991.

[63] Gregory D. Hager and Peter N. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 403–410, San Francisco, CA, June 1996.

[64] C.G. Harris. Determination of ego-motion from matched points. In *Proceedings of the 3rd Alvey Conference, Cambridge*, pages 189–192, September 1987.

[65] Chris Harris. Geometry from visual motion. In Andrew Blake and Alan Yuille, editors, *Active Vision*, chapter 16, pages 263–284. The MIT Press, 1993.

[66] D. Heeger. Optical Flow Using Spatiotemporal Filters. *International Journal of Computer Vision*, 1:279–302, 1988.

[67] E. Hildreth. Computations Underlying the Measurements of Visual Motion. *Artificial Intelligence*, 23:309–354, 1984.

[68] D.D. Hoffman. Infering Local Surface Orientation from Motion Fields. *Journal of the Optical Society of America A*, 72:880–892, 1982.

[69] James R. Holton. *An introduction to Dynamic Meteorology.* third edition, Geophys. series, Vol. 45, Academic Press Inc., 1992.

[70] R. Horaud and T. Skordas. Stereo Correspondence Through Feature Grouping and Maximal Cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1168–1180, November 1989.

[71] B.K.P. Horn and B. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.

[72] B.K.P. Horn. *Robot Vision.* MIT Press, 1986.

[73] B.K.P. Horn and J.R. Weldon. Computationally Efficient Methods for Recovering Translational Motion. *International Journal of Computer Vision*, 2:2–11, 1987.

[74] Y. Hsu, H.H. Nagel, and G. Rekkers. New Likelihood Test Methods for Change Detection in Image Sequences. *Computer Vision, Graphics and Image Processing*, 26:73–106, 1984.

[75] Y. Huang, K. Palaniappan, X. Zhuang, and J.E. Cavanaugh. Optic Flow Field Segmentation and Motion Estimation Using a Robust Genetic Partitioning Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1177–1190, December 1995.

[76] Daniel P. Huttenlocher, Jae J. Noh, and William J. Rucklidge. Tracking non-rigid objects in complex scenes. In *Proceedings of the 4th International Conference on Computer Vision*, pages 93–101, Berlin, Germany, May 1993. IEEE Computer Society Press.

[77] Vincent S.S. Hwang. Tracking feature points in time-varying images using an opportunistic approach. *Pattern Recog.*, 22(3):247–256, 1989.

[78] (Insight Project Experiments - ESPRIT) Working discussions and experiments (J. Arnspang and colleagues) at Utrecht Biophysics Research Institute and Technical Univ. at Delft during ESPRIT Insight Project, 1989-1990.

[79] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In Bernard Buxton, editor, *Proceedings of the 4th European Conference on Computer Vision*, volume I, pages 343–356, Cambridge, UK, April 1996.

[80] M. Irani, B. Rousso, and S. Peleg. Computing Occluding and Transparent Motions. *International Journal of Computer Vision*, 12(1):5–16, 1994.

[81] R.C. Jain. Segmentation of Frame Sequences Obtained by a Moving Observer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(5):624–629, September 1984.

[82] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision*, pages 259–268, June 1987.

[83] J. J. Koenderink, J. J. and A. J. van Doorn. Invariant Properties of the Motion Parallax Field due to the Movement of Rigid Bodies Relative to an Observer. *Optica Acta* 22, 1975.

[84] J. J. Koenderink. *Local Structure of Movement Parallax of the Plane. Journal of Optical Society. America*, July 1976.

[85] D. Koller, K. Daniilidis, T. Thorhallson, and Nagel H. Model-based object tracking in traffic scenes. In *Proc. 2nd ECCV*, Italy, pages 437–452, 1992.

[86] Dieter Koller, Joseph Weber, and Jitendr.a Malik. Robust multiple car tracking with occlusion reasoning. In Springer-Verlag, editor, *Third European Conference on Computer Vision*, volume I, pages 189–196, Stockhol, Sweden, May 1994.

[87] T. Lawton. Processing Translational Motion Sequences. *Computer Vision, Graphics and Image Processing*, 22:116–144, 1983.

[88] C.H. Lee and A. Joshi. Correspondence Problem in Image Sequence Analysis. *Pattern Recognition*, 26(1):47–61, 1993.

[89] L. Li and J. Duncan. 3-D Translational Motion and Structure from Binocular Image Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-15(7):657–667, July 1993.

[90] N. V. Lobo and J. K. Tsotsos. Computing Egomotion and Detecting Independent Motion from Image Motion Using Collinear Points. *Computer Vision and Image Understanding*, 64(1):21–52, July 1996.

[91] H.C. Longuet-Higgins. A Computer Algorithm for Reconstruction of a Scene from Two Projections. *Nature*, 293:133–135, 1981.

[92] D. Lowe. Robust model based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, 1992.

[93] R. Malladi, J. A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: A level set approach. *Pattern Analysis and Machine Intelligence*, 17(2):158–175, February 1995.

[94] R Malladi, J.A. Sethian, and B.C. Vemuri. A topology independent shape modeling scheme. *SPIE*, 2031:246, 1993.

[95] D. Metaxas and D. Terzopoulos. Constrained deformable superquadrics and nonrigid motion tracking. In *Computer Vision and Pattern Recognition*, pages 337–343, 1991.

[96] F. Meyer and P. Bouthemy. Region-based tracking in an image sequence. In *Proc. 2nd ECCV, Santa Margherita Ligure, Italy*, pages 476–484, May 1992.

[97] H. Moravec. Towards Automatic Visual Obstacle Avoidance. In *International Joint Conference on Artificial Intelligence*, pages 584–585, 1977.

[98] A. Movshon. *Images and Understanding*, pages 122–137. Cambridge University Press, 1990.

[99] D.W. Murray and B.F. Buxton. Reconstructing the Optic Flow from Edge Motion: An Examination of Two Different Approaches. In *First Conference on AI Applications*, 1984.

[100] H.H. Nagel. Displacement Vectors Derived from Second order Intensity Variations in Image Sequences. *Computer Vision, Graphics and Image Processing*, 21:85–117, 1983.

[101] V.S. Nalwa. *A Guided Tour of Computer Vision*, chapter 8. Addison-Wesley, 1993.

[102] N. Navab, R. Deriche, and O.D. Faugeras. Recovering 3D motion and structure from stereo and 2D token tracking cooperation. In *Proc. Third Int'l Conf. Comput. Vision*, pages 513–517, Osaka, Japan, December 1990. IEEE.

[103] Nassir Navab, Zhengyou Zhang, and Olivier Faugeras. Tracking, motion and stereo: A robust and dynamic cooperation. In *7th Scandinavian Conference on Image Analysis*, pages 98–105, Aalborg University, Denmark, August 1991.

[104] S. Negahdaripour and B.K.P. Horn. Direct Passive Navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:163–176, 1987.

[105] R.C. Nelson. Qualitative Detection of Motion by a Moving Observer. *International Journal of Computer Vision*, 7(1):33–46, 1991.

[106] R.C. Nelson and J. Aloimonos. Finding Motion Parameters from Spherical Motion Fields (or the Advantages of Having Eyes in the Back of Your Head. *Biological Cybernetics*, 58:261–273, 1988.

[107] R.C. Nelson and Y. Aloimonos. Obstacle Avoidance Using Flow Field Divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(10):1102–1106, October 1989.

[108] P. Nordlund and T. Uhlin. Closing the Loop: Detection and Pursuit of a Moving Object by a Moving Observer. *Image and Vision Computing*, 14:267–275, 1996.

[109] J.M. Odobez and P. Bouthemy. Detection of Multiple Moving Ojects Using Multiscale MRF with Camera Motion Compensation. In *1st IEEE International Conference on Image Processing, ICIP*, 1994.

[110] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed : algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[111] N. Paragios and G. Tziritas. Detection and Location of Moving Objects Using Deterministic Relaxation Algorithms. In *ICPR96*, Vienna, Austria, September 1996.

[112] Nikolaos Paragios and Rachid Deriche. Detection of Moving Objects: A Level Set Approach. In *Proceedings of SIRS'97, Stockholm, Sweden*, July 1997.

[113] Nikolaos Paragios and Rachid Deriche. A PDE-based Level-Set Approach for Detection and Tracking of Moving Object In *INRIA Research Report, Also Submitted to ICCV'97* , 1997.

[114] G. Patras, N. Alvertos, and G. Tziritas. Joint Disparity and Motion Field Estimation in Stereoscopic Image Sequences. In *ICPR96*, Vienna, Austria, September 1996.

[115] K. Prazdny. Determining Instantaneous Direction of Motion from Optical Flow Generated by a Curvilinear Moving Observer. *Computer Vision, Graphics and Image Processing*, 17:238–248, 1981.

[116] Bobby. Rao. Data association methods for tracking systems. In Andrew Blake and Alan Yuille, editors, *Active Vision*, chapter VI, pages 91–106. The MIT Press, 1993.

[117] James M. Rehg. *Visual Analysis of High DOF Articulated Objects with Application to Hand Tracking*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA, April 1995.

[118] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. AC*, 24:843–854, December 1979.

[119] G. Sandini and M. Tistarelli. Active tracking strategy for monocular depth inference over multiple frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):13–27, January 1990.

[120] G. Sandini, F. Gandolfo, E. Grosso, and M. Tistarelli. Vision During Action. In Yiannis Aloimonos, editor, *Active Perception*, chapter 4. Lawrence Erlbaum Associates, Hillsdale, NJ, 1993.

[121] Ishwar Sethi and Ramesh Jain. Finding trajectories of Feature Points in a Monocular Image Sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):56–73, January 1987.

[122] J. A. Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. In *Proc. Nat. Ac. Science*, volume 93, pages 1591–1694, 1996.

[123] R. Sharma. Robust Detection of Independent Motion: An Active and Purposive Solution. Technical report, Center for Automation Research, University of Maryland, CAR TR-534, College Park, MD, 1991.

[124] R. Sharma and Y. Aloimonos. Early Detection of Independent Motion from Active Control of Normal Image Flow Patterns. *IEEE Transactions on SMC*, SMC-26(1):42–53, February 1996.

[125] A. Shashua. Projective Structure from two Uncalibrated Images: Structure from Motion and Recognition. Technical Report A.I. Memo 1363, MIT, A.I. Lab, September 1992.

[126] Y.Q. Shi, C.Q. Shu, and J.N. Pan. Unified Optical Flow Approach to Motion Analysis from a Sequence of Stereo Images. *Pattern Recognition*, 27(12):1577–1590, 1994.

[127] D. Sinclair, A. Blake, and D. Murray. Robust Estimation of Egomotion from Normal Flow. *International Journal of Computer Vision*, 13(1):57–69, 1994.

[128] A. Singh. *Optical Flow Computation: A Unified Perspective.* PhD Dissertation, Department of Computer Science, Columbia University, New York, NY, 1990.

[129] K. Skifstad and R. Jain. Illumination Independent Change Detection for Real World Image Sequences. *Computer Vision, Graphics and Image Processing*, 46:387–399, 1989.

[130] M.E. Spetsakis and Y. Aloimonos. Optimal Motion Estimation. In *IEEE Workshop on Visual Motion*, pages 229–237, 1989.

[131] M.E. Spetsakis and Y. Aloimonos. Structure from Motion Using Line Correspondences. *International Journal of Computer Vision*, 4:171–183, 1990.

[132] M. Subbarao. *Interpretation of Visual Motion.* PhD Dissertation, Center for Automation Research, Univ. of Maryland, College Park, MD, 1988.

[133] M. Subbarao. Bounds on Time-to-Collision and Rotational Component from First-Order Derivatives of Image Flow. *Computer Vision, Graphics and Image Processing*, 50(3):329–341, 1990.

[134] Demetri Terzopoulos and Richard Szeliski. Tracking with Kalman snakes. In Andrew Blake and Alan Yuille, editors, *Active Vision*, chapter 1, pages 3–20. The MIT Press, 1993.

[135] W.B. Thompson, P. Lechleider, and E.R. Stuck. Detecting Moving Objects Using the Rigidity Constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(2):162–166, February 1994.

[136] W.B. Thompson and T.C. Pong. Detecting Moving Objects. *International Journal of Computer Vision*, 4:39–57, 1990.

[137] C. Tomasi and T. Kanade. Shape and Motion from Image Streams under Orthography: a Factorization Method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[138] P. H. S. Torr and D. W. Murray. Stochastic Motion Clustering. In J.-O. Eklundh, editor, *Proceedings of ECCV'94, LNCS, vol. 80*, pages 328–337, 1994.

[139] P.H.S. Torr and D.W. Murray. Statistical Detection of Independent Movement from a Moving Camera. *Image and Vision Computing*, 11:180–187, May 1993.

[140] 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*, PAMI-6(1):13–26, January 1984.

[141] G. Tziritas. Recursive and/or Iterative Estimation of Two-Dimensional Velocity Field and Reconstruction of Three-Dimensional Motion. *Signal Processing*, 16:53–72, January 1989.

[142] G. Tziritas and C. Labit. *Motion Analysis for Image Sequence Coding.* Elsevier, New York, 1994.

[143] S. Ullman. The Interpretation of Structure from Motion. In *Royal Society, London, B*, volume 203, pages 405–426, 1979.

[144] S. Ullman. *The Interpretation of Visual Motion.* MIT Press, Cambridge, MA, 1979.

[145] S. Uras, F. Girosi, and V. Torre. A Computational Approach to Motion Perception. *Biological Cybernetics*, 60:79–87, 1988.

[146] T. Viéville. Estimation of 3D-motion and structure from tracking 2D-lines in a sequence of images. In O.D. Faugeras, editor, *Proceedings of the 1st ECCV, Antibes*, pages 281–292. Springer-Verlag, Berlin, 1990.

[147] J.Y.A. Wang and E.H. Adelson. Representing Moving Images with Layers. *IEEE Transactions on Image Processing*, 3(5):625–638, September 1994.

[148] W. Wang and J. H. Duncan. Recovering the three-dimensional motion and structure of multiple moving objects from binocular image flows. *Computer Vision and Image Understanding*, 63(3):430–440, May 1996.

[149] A.M. Waxman and J.H. Duncan. Binocular Image Flows: Steps Toward Stereo-Motion Fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):715–729, November 1986.

[150] A.M. Waxman, B. Kamgar-Parsi, and M. Subbarao. Closed-form Solutions to Image Flow Equations for 3D Structure and Motion. *International Journal of Computer Vision*, 1:239–258, 1987.

[151] A.M. Waxman and K. Wohn. Contour Evolution, Neighborhood Deformation and Global Image Flow. *International Journal of Robotics Research*, 4:95–108, 1985.

[152] A.M. Waxman, K. Wohn. *Image Flow Theory.* In Advances of Computer Vision, ed. C. Brown. Erlbaum Publishers, 1986.

[153] AM.. Waxman, F. Bergholm, Wu. Convected activation profiles. In *Proc of CVPR*, Ann Arbor, Michigan, 1988.

[154] J. Weng, N. Ahuja, and T.S. Huang. Optimal Motion and Structure Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):864–884, September 1993.

[155] J. Weng, T.S. Huang, and N. Ahuja. Motion and Structure from Two Perspective Views:Algorithms, Error Analysis, and Error Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):451–476, May 1989.

[156] G. White and E.J. Weldon. Utilizing Gradient Vector Distributions to Recover Motion Parameters. In *International Conference on Computer Vision*, pages 64–73, 1988.

[157] K.Y. Wohn, J. Wu, and R.W. Brockett. A Contour-Based Recovery of Image Flow: Iterative Transformation Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(8):746–760, 1991.

[158] R.Y. Wong and E.L. Hall. Sequential Hierarchical Scene Matching. *IEEE Transactions on Computers*, 27:359–366, 1978.

[159] Y. Yeshurun and E. L. Schwartz. Cepstral Filtering on a Columnar Image Architecture: A Fast Algorithm for Binocular Stereo Segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(7):759–767, July 1989.

[160] Z. Zhang. Strategies for tracking tokens in a cluttered scene. In *Proc. British Machine Vision Conference BMVC93*, pages 207–216, University of Surrey, Guildford, UK, September 1993.

[161] Z. Zhang. Token tracking in a cluttered scene. *Int'l J. of Image and Vision Computing*, 12(2):110–120, March 1994. also Research Report No.2072, INRIA Sophia-Antipolis, 1993.

[162] Zhengyou Zhang and Olivier Faugeras. Tracking and motion estimation in a sequence of stereo frames. In L.C. Aiello, editor, *9th European Conference on Artificial Intelligence*, pages 747–752, Stockholm, Sweden, August 1990.

[163] Zhengyou Zhang and Olivier Faugeras. Tracking 3D line segments: New developments. In *5th International Conference on Advanced Robotics*, pages 1365–1370, Pisa, Italy, June 1991. IEEE.

Chapter 3

Visual Landmarks

by Fredrik Bergholm, Claus B. Madsen, Rachid Deriche and Panos Trahanias

In nature, the process of navigation usually involves a cognitive representation of the environment that facilitates localization as well as determination of the appropriate motion(s) to reach a navigation target. Such cognitive representations of the environment manifest themselves in many cases as images of distinct objects or views of the workspace, *landmarks*, that can be used to unambiguously characterize the environment. Landmarks are routinely used by biological systems as *reference points* during navigation. In contrast to traditional, metric approaches in navigation that use environment maps as a workspace representation, landmark-based approaches rely on qualitative representations and hence result in qualitative navigation. Such approaches are extremely useful in robotic navigation and many efforts have been put in their development. In this chapter we focus mostly on the issues of *visual homing* (navigating between landmarks), *landmark selection* and *landmark recognition* and review past and current approaches that address them.

3.1 Visual homing

The word *visual homing* was coined by Randal Nelson in his Ph.D. thesis [29], and refers to a kind of qualitative navigation where one advances from one *landmark* to the next, *without a map*, and *without knowing metric information* such as distances. The underlying goal is to move from point A to B (maybe by way of $B_1, B_2, ..$) using landmarks. The procedure is slightly analogous to the situation when a human finds his way in a city by having memorized certain views ($\sim$ landmarks) knowing what action to take upon recognition of each view. Insects, like bees, may perform something similar, finding their way back to the hive.

When considering visual homing, one would typically shun things like 2- or 3-D maps of the environment, since visual homing, by definition, should be qualitative navigation *without* a map. In a practical application one may wish to blend *visual homing* and *map-guided navigation*, but this will not be the topic of this section. To be deprived of a map, does, however, not mean that one has no idea of the spatial organization, at all. Qualitative notions such as 'far' and 'distant' are examples of crude qualitative environmental information.

3.1.1 Definitions

Visual homing contains elements of *learning*. To quote Nelson [29]: *"...the problem is interesting because it appears to be one of the earliest visual operations performed by biological systems that involves substantial amounts of learned information."*. In order to be fairly brief, we consider learning only in a quite limited sense in this chapter. Learning only enters in the form of a *training procedure* or some simple *associations*, here.

The training procedure is the phase during which the landmarks are automatically or semi-automatically (or interactively) entered into a memory. Association handling may involve 'forgetting' landmarks that are far from the current landmark, and 'activate' the memory of those that are nearby.

When storing a landmark in a memory the worst alternative would be to store the image itself, since the memory would quickly be filled, and it is quite unlikely that a certain view would be repeated that exactly when moving through a scene the next time. An abstraction of the view representing the landmark is more appealing, which is why the concepts of *pattern* and *pattern space* often are introduced.

A vector $(a_1, a_2, \ldots, a_n)$ is said to be a *pattern* where each quantity is called a *feature*. Each feature is part of a *pattern space* which is a set of possible integer values for each feature. For example, if $(a_1, a_2, \ldots, a_n)$ is a pattern, then $A_1 \times A_2 \times \cdots \times A_n$ is the corresponding pattern space, where for example, $|A_1| = 2$, $|A_2| = 8, \ldots$, where $|..|$ is the size of a set[1]. A feature whose pattern space is of size 2, is obviously a *binary* feature. A statement about a landmark that could be answered by 'yes' or 'no', would be an example of a binary-valued feature. If, say, color is quantized into 8 categories, then this is an eight-valued feature.

Given a pattern representation, the concept *recognition neighborhood* [29, p.72] refers to the points in space $\bar{x} \in \mathbb{R}^3$ and camera orientations $\bar{\phi} \in \mathbb{R}^3$, at $\bar{x}$, for which a memorized pattern triggers recognition, based on some similarity test. Landmarks that are specific for a certain environment are usually referred as *specialized landmarks*, [23, p.116]. Door handles, hinges, doors, etc. are examples of specialized landmarks in certain indoor environments[2].

Some objects exist only at a certain time and Kuhnert calls them *randomly existing objects* [23, p.117]. Certain shadows (only visible at a certain time of the day) would be an example of such an object. An additional chair in a room which is absent the next time the camera platform visits the room is another.

Landmarks may possess *attributes* or additional information attached to them. One natural attribute would be a label describing crude location. For example, in an indoor environment the names of various rooms may be a label to be attached to each landmark. A useful piece of information would also be the appropriate actions that should be taken when the current landmark is encountered. Attributes and related information are aquired during the training phase, i.e. during landmark selection and environment learning. Landmarks may also possess *links* to each other, with attributes such as 'far' or

[1] For example, $A_1 = \{1, 2\}$ and $A_2 = \{0, 1, 2, \ldots, 7\}$.

[2] *Special infrastructure* such as man-made markers [31, p.76], or lines could guide navigation. However, we consider this to be too specific or inflexible and do not treat it in this chapter.

'unblocked path', assigned to these links.

Apart from landmarks there are other ways of crudely characterizing certain aspects of the environment. A *freeway* [6], is defined as a rectangular elongated region of space free from obstacles. An example of a freeway in an indoor environment could be a memorized existence of a corridor, which is a freeway at the current moment if unblocked. The length dimensions of the freeway are not meant to be memorized[3] although some notion of needed time to traverse the freeway could be associated with it. A freeway must, of course, be detected somehow using the available sensors.

The process by which a mobile agent returns to the original position, its 'home', after a mission of some kind, we will refer to as *nest homing*, which will also be one topic in Chapter 4. Nest homing may involve visual homing, but not necessarily so. The example in Chapter 4 involves dead reckoning, and some global light information thereby largely by-passing the local visual landmarks, to be discussed below.

3.1.2 Approaches to visual homing

An early example of a vision-based autonomous ground vehicle using landmarks[4], instead of maps, for navigating is the `Harunobu-3` vehicle developed at Yamanashi University [27]. The vehicle followed a road lane using a downward tilted color TV-camera. The chosen landmarks belonged to a fairly restricted set of naturally appearing patterns, painted on the road. Landmarks were stored in a nongeometric topological map as a sequential list containing type of landmark (pattern) and attributes such as distances to next and previous landmarks.

When being in the vicinity of a landmark, it may be useful to know approximately how far away one is from it. A few landmarks could have a height attribute associated to them. This was used by Kuhnert in [23], for the vision-guided experimental vehicle `Athene`. It constitutes an example of a fairly standard operation, that of *position knowledge via height of features* [23, p.118]. With known height, one approximately knows how far away one is from the landmark. This is a borderline-case, whereby some crude metric information is inferred from memorized environment, and this estimation is somewhere in a gray zone between qualitative and non-qualitative navigation[5]. A cruder way of judging vicinity would be to store landmarks in multiple resolutions, and have associative links between the multiple resolutions. When the finer resolution view is more similar than the coarser one, then the moving observer is 'closer' to the landmark.

In the approach developed by Nelson [29], visual homing is performed based on image patterns. The latter are obtained by filtering an image in large subrectangles, letting a single filter response in each subrectangle, be a pattern. Such an approach is a rather general one, and many possibilities of defining pattern vectors apparently exist. If dividing the image into 25 blocks (subrectangles) and quantizing each filter response into 8 values, then we have a pattern vector of length 25, each feature having one of 8 possible values.

[3] A freeway is a rubbery map, indicating a qualitative aspect of the environment.

[4] No automatic landmark selection was done.

[5] A height attribute of a landmark is not really a map of the environment, but some sparse incomplete metric information.

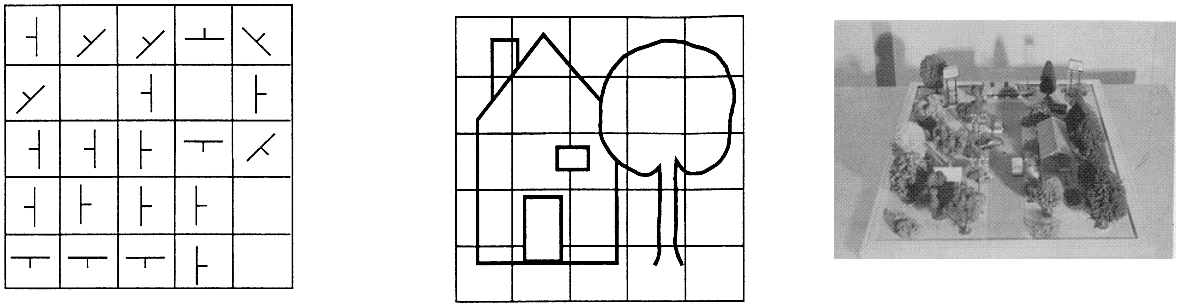


Figure 3.1: *(a) Edge direction representation of a simple scene (b), for memorizing and representing landmarks in visual navigation [29], (c)* `Tinytown`.

Such a pattern vector has also been employed by Nelson, by using the average edge direction in each block as feature, quantized into 8 directions (see Fig. 3.1). The size of the pattern space in this case can be regarded as “reasonable”, since it contains 8^{25} possible patterns and the probability of two patterns being *similar* by chance is roughly equal to

$$\sum_{i=13}^{25} \binom{n}{i} \left(\frac{1}{8}\right)^{i} \left(\frac{7}{8}\right)^{25-i} \approx 10^{-5} \tag{3.1}$$

When the *similarity measure* is computed as the number of features having exactly the same value, uniform probability distributions are assumed for the features, and two patterns are declared to be *similar* if more than 12 features have the same value.

The visual homing experiments to demonstrate the above approach were done with a robot arm traveling through a miniature landscape, christened `Tinytown`. The camera was set to move parallel to the ground and facing straight down. Thus it had two degrees of freedom, since there was no rotation involved. The size of the landscape was 40 by 60 cm, and the camera moved in a plane 30 cm above the ground plane of the landscape.

In these experiments it was observed [29] that, when the space of permissible camera motions is augmented to include rotations (which is the normal case in indoor robotics), then features that exhibit rotation-invariance should be used, to avoid excessive growth of the number of needed memorized views. In general, as Nelson puts it: *“There is still the problem of designing a suitable pattern space - there is no such thing as a free lunch..*, [29, p.102]. Determining the required number of refererence points (each point from which a landmark was recorded) and their distribution, is also a necessary step to make the approach more practical, according to the author.

Visual landmark representation without explicit world model

In the approach described above, the environment representation consists of patterns stored during a training phase. These patterns, upon traversing the environment again, will trigger recognition if the pattern extracted from the current view is sufficiently similar to one of the stored patterns. There could be an internal bookkeeping checking that recognized patterns (recognized landmarks) are traversed in a correct predefined order.

In this sense, navigation based on recognized patterns (landmarks) corresssponds to visual homing without using an external model of the world. This is a desirable property since, among other things, *"it results in a more easily analyzable system than one based on a global model, since all the interactions are local in nature"* [29, p.72].

Visual landmarks and high-level knowledge

It is possible that, in specific cases, landmark recognition as performed by humans is done in a totally different fashion, which is loosely related to image stimuli. Let us call this *recognition by functionality.* A door (doorway) may not be recognized by its appearance, but by its functionality. Any narrow passage which leads you from one open space to another, is a "door". A certain building, during a stroll downtown, may be recognized not beacuse the exact views are remembered but by observing what activity is performed in the place; something like: *'I remember passing a restaurant soon after turning left'.* Any view which yields the realization that the building in question houses a restaurant, is enough to trigger landmark recognition. Higher-level knowledge may also be important for landmark selection, since humans may have good intuition what is, and what is not *a permanent* landmark[6], and it may be that intelligent mobile agents, like humans, are quite superior in choosing good landmarks for visual homing.

3.1.3 Conclusions and remarks

There are several alternatives for storing landmark information. One may choose an *image-like* representation. The *edge direction* representation in strongly downsampled images, is an example of such a representation. From large sub-blocks of an image, some quantity is calculated. A variant of such a representation is to store vectors of quantities for each sub-block. Information such as *circular variance* [18], *color*, *depth*, etc. could also be stored. Landmarks may of course be stored in multiple resolutions to facilitate or enable recognition.

However, it is not obvious how robust such a representation is in an indoor environment for a 3D motion space[7]. It seems desirable to use a pattern representation which is fairly *rotation invariant*, to avoid the already mentioned excessive growth in the number of patterns that need to be stored. Color variables, such as *hue*, are rotation invariant. Moreover, recognition neighbourhoods should not be too narrow to be useful in practice.

Instead of storing whole hue patterns in image sub-blocks—which seems superfluous—one could store the existence of an 'unusual' hue in the current image, where 'unusual' would mean: not seen frequently during a certain span of time in a time sequence [15]. When memorizing the pattern something like the "20 questions game" [33] could be applied. Once the 'unusual hue' is memorized as a feature, one could ask questions such as: 'Does the hue form a region of a simple shape (ellipses and quadrilaterals are rotation invariant)? If the answer is yes, store a second feature "shape". A third question could be

[6] This bears resemblance to the concept of *randomly existing* objects, mentioned earlier.

[7] The 3D motion space is derived from the 2D horizontal motion assuming a planar floor for the autonomous ground vehicle and the pan motion of the camera. However, in many situations the motion space may be 4- or 5- dimensional space – 2D motion and two or three orientation angles of the optical axis.

whether there is another pattern of a certain hue residing in the interior of the unusual-hue region. If so, store that hue also. In order to perform the hue measurements needed in the above, one should rather perform some white balancing before calculating the hue to account for *color constancy.*

The ultimate success of visual homing hinges on the joint performance of the *landmark selection process* and chosen pattern representations. In designing software, it is tempting to work with pattern representations and landmark selection strategies that are fixed *beforehand.* A challenging task may be to design agents so that they are more adaptable, for it seems likely that both selection strategies and pattern representations in really long image sequences, covering several environments, must adapt, must change 'modes', according to current conditions.

3.2 Landmark selection

When a robot moves around in an environment there can be two purposes of having a process for automatic detection of potential landmarks in incoming images:

1. to select regions/features in the image that should be matched to stored landmarks in order to aid robot self-localization

2. to select regions in the image that could potentially be committed to memory for use as future landmarks

This section is devoted to aspects of finding 'areas of interest' in images using purely data-driven techniques. Naturally what could constitute potential landmarks and how to detect them, is intimately linked to how the landmarks are to be used in a system context, and how landmarks are recognized. Therefore, many aspects of landmark selection are closely related to landmark recognition (section 3.3).

For the two potential uses of a landmark selection process mentioned above, it is important that the process can operate in a bottom-up, data-driven manner, generating landmark hypotheses and pass them on for higher level processing. The subsequent brief survey is thus limited to methodologies that support this mode of operation. In [10] a directed search for doors using estimated robot position and a priori known locations of doors is described. Such approaches are top-down and model-driven and thus not included in this survey. Similarly, the survey does not include recovery of three dimensional structure and subsequent identification of potential landmarks. This is primarily because the direct use of 3D structure for landmarks is normally coupled to the use of ultra-sonic or laser range sensors – issues that are not covered in-depth by this report.

3.2.1 Definitions

In what follows we distinguish between two types of landmarks or landmark candidates: iconic and non-iconic. Iconic landmarks are essentially images (or sub-images) represented as pixel arrays, whereas non-iconic landmarks are features of higher level abstraction derived from images, e.g. edge segments.

For any type of bottom-up detection of potential interest areas in images, be it iconic or non-iconic, the issue of *saliency* is essential. Salient is another word for prominent or 'eye-catching' and is often used in the context of so called saliency maps [11] and conspicuity maps. Conspicuity maps are typically derived from images and are 2D discrete arrays of values just as images, but rather than coding intensity values, a pixel in a conspicuity map in general codes how prominent the corresponding image pixel is, according to some measure. For example the gradient magnitude image is a conspicuity map, coding the edge strength at image locations.

Sometimes construction of conspicuity maps is based on a weighted sum of values from many different maps. To be able to distinguish between maps computed directly through image processing, and maps computed as weighted sums of other maps, it is common to use two terms: conspicuity maps and saliency maps. A conspicuity map is directly derived from an image through processing and related to some well-defined property, e.g. the gradient magnitude is a conspicuity map. A saliency map is a combination of multiple conspicuity maps, typically through weighted sums. The distinction between conspicuity and saliency maps was probably introduced by Milanese [25, 26].

3.2.2 Landmark selection approaches

Non-iconic features detected by image processing

One of the simplest approaches to landmark selection is illustrated in the fairly common use of vertical straight lines, where some edge detection algorithm is employed directly to incoming images. An example of this is the work by Carlsson [7]. Naturally not all vertical edge segments are actual landmarks and it is necessary to match the detected edges to stored landmarks. Neira et al. [28] also use vertical edge segments and reconstruct the 3D position of the segments from multiple observations for including the segments into a model of the environment as the robot moves about.

Another simple technique for detecting potential landmarks from images is described in [20]. Images are acquired by letting a camera look down on a spherical mirror, providing a 360° panoramic view of the environment. Intensity values are averaged onto a one-pixel wide circular band corresponding to the real world height of the spherical mirror, which in effect results in a 1D image of intensity values around the robot. Zero-crossings of the second order derivative of this 1D image are detected and used as potential landmarks. These zero-crossings typically correspond to wall corners, edges of filing cabinets and door frames. In the described system the position of some of these physical landmarks are stored in advance. Associations of detected landmarks with stored ones (*recognition*) is performed by pairing each zero-crossing to the closest anticipated landmark, given an estimated robot position and subject to the constraint that landmarks must appear in 'proper order' along the 1D image signal.

Iconic features: schemes for attention selection

While the purpose of this section is to briefly survey mechanisms for driving visual attention, it would be an omission not to mention that entire images can serve as iconic visual landmarks as is also described in the next section on Landmark Recognition (c.f. Sec. 3.3); [9, 30] are examples of this approach. In both works images are collected by manually guiding the robot through some path. The collected images are stored in memory (possibly in some compressed format), and used for comparison to new, incoming images once the robot attempts to re-trace the original path on its own.

Work in recent years on active vision and agile camera heads have encouraged studies in finding techniques for selecting areas of images on which to direct attention. A thorough review of attention selection mechanisms is provided by Andersen in [1]. Many of these mechanisms are in some manner inspired by biological vision systems and the concept of receptive fields. Culhane and Tsotsos [12] described an algorithm using a multi-resolution approach to selecting attention, and illustrated it directly on intensity images, so that attention was given to the most prominent bright areas of the scene. Thus, in the terms defined in section 3.2.1 there is only one conspicuity map, which is also the saliency map.

In general it is possible to combine several different attention cues by constructing a saliency map from a weighted sum of multiple conspicuity maps, as demonstrated by Milanese [25, 26] and Trahanias et al. [42]. Figures 3.2 and 3.3 illustrate this principle for a particular scenario. The conspicuity maps are formed from 4 different features (Fig. 3.2): intensity, edges, corners and motion. These four feature images are converted into conspicuity maps by running a DoG filter directly over the feature images (Fig. 3.3). The purpose of filtering the feature images is to suppress regions with non-varying feature responses and to amplify areas where features are located in small clusters. This is a sound selection, since if all responses from a given feature are identical in a large area the feature is not very informative or salient for this region.

Figure 3.4 demonstrates a simple version of the attention selection scheme applied to the problem of detecting possible regions for iconic landmarks in a hallway. The conspicuity map applied here is based on local magnitude and orientation variance of the intensity gradient. It is seen how the technique attracts attention to regions that are descriptive and have a high likelihood of being possible to track a sequence of images.

It is possible to build conspicuity maps from any kind of feature that produces a response varying in strength over an image. A novel choice could be using a symmetry measure as the one proposed by Reisfeld [32].

A bottom-up generation of attention cues as described in this section can actually be combined with top-down, model-driven search for known landmarks using a coherent framework. Some examples of this kind of integration may be found in [24, 1, 42]. The approach described in [42] employs *a priori* knowledge of the workspace structure (indoor corridors and rectangular rooms) to segregate the image into *qualitatively* distinct areas: walls, ceiling, floor, far end. This information is combined with a saliency map, and landmark selection is performed only in designated areas (walls), potentially avoiding the selection of *randomly existing objects* as landmarks. Landmarks are finally extracted as image areas, using a region growing technique. This procedure is illustrated in Fig 3.5. The qualitative workspace segregation, superimposed on the saliency map, is shown in

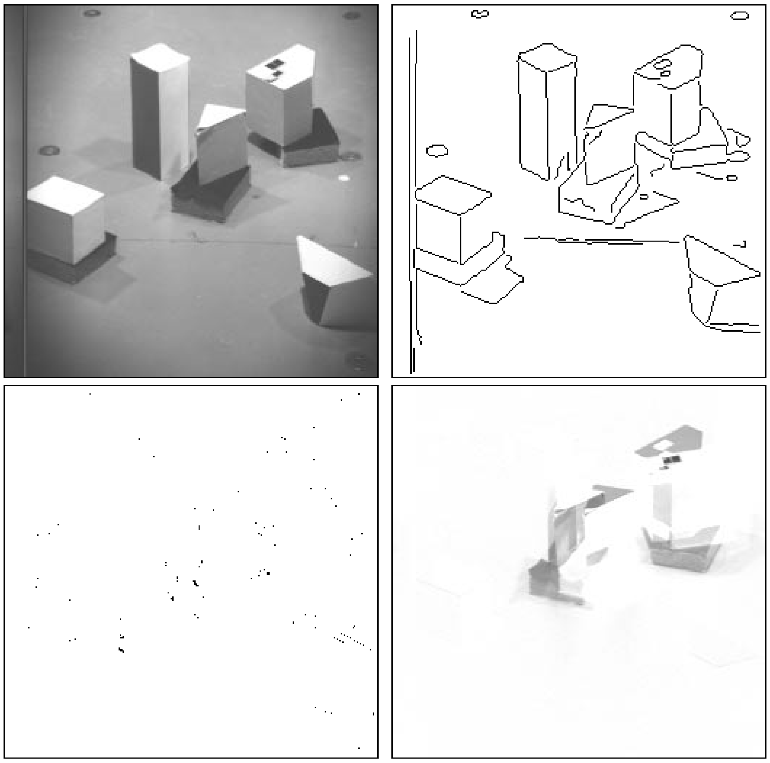


Figure 3.2: *Examples of features for creation of conspicuity maps. Top left: an image of a blocks world scenario. Top right: edges detected with a Canny filter. Bottom left: corners detected with a corner detector. Bottom right: motion detected by image differencing. Reprinted with permission from [1].*

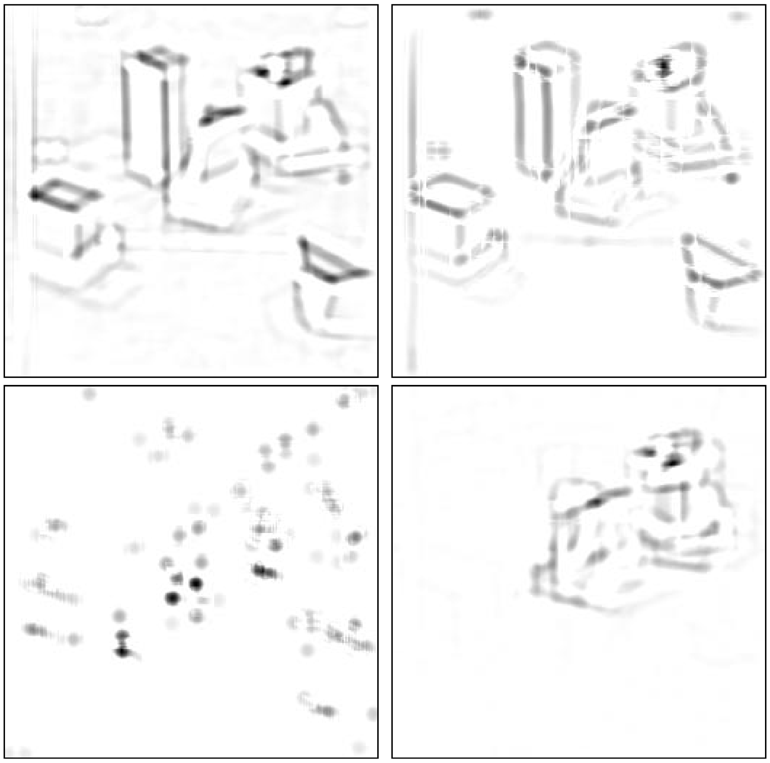


Figure 3.3: *Conspicuity maps generated from the feature images shown in Fig. 3.2, by filtering each feature images using a DoG (second order derivative) filter. These conspicuity maps can be combined into a single saliency map. Reprinted with permission from [1].*

Fig 3.5a; this result has driven attention to the points marked by "x", and through a heuristic procedure the final landmark location has been determined. After the application of a region growing algorithm, the landmark pattern shown in Fig 3.5b has been extracted.

3.2.3 Conclusions and remarks

In the beginning of this section we argued that a mobile robot navigation system based on visual landmarks has two uses of a bottom-up process locating potential landmarks in incoming images. The first use is for identifying landmark candidates to be matched to landmark models stored in a database; the other use is to enable automatic landmark acquisition and commitment to memory. Such a bottom-up process is thus a kind of attention mechanism. Attention is a necessity unless fairly accurate physical landmark position and robot position/heading estimates are available, since generally it would be computationally prohibitive to match all stored landmarks to every, entire image.

One aspect concerning automatic landmark acquisition or learning, which was not addressed here, is that of obtaining all the information concerning a new landmark, which is needed for the database. In a model-based system the positions of landmarks must be known and stored together with the visual appearances. Obtaining this information requires a stereo camera setup or multiple observations from different calibrated viewpoints.

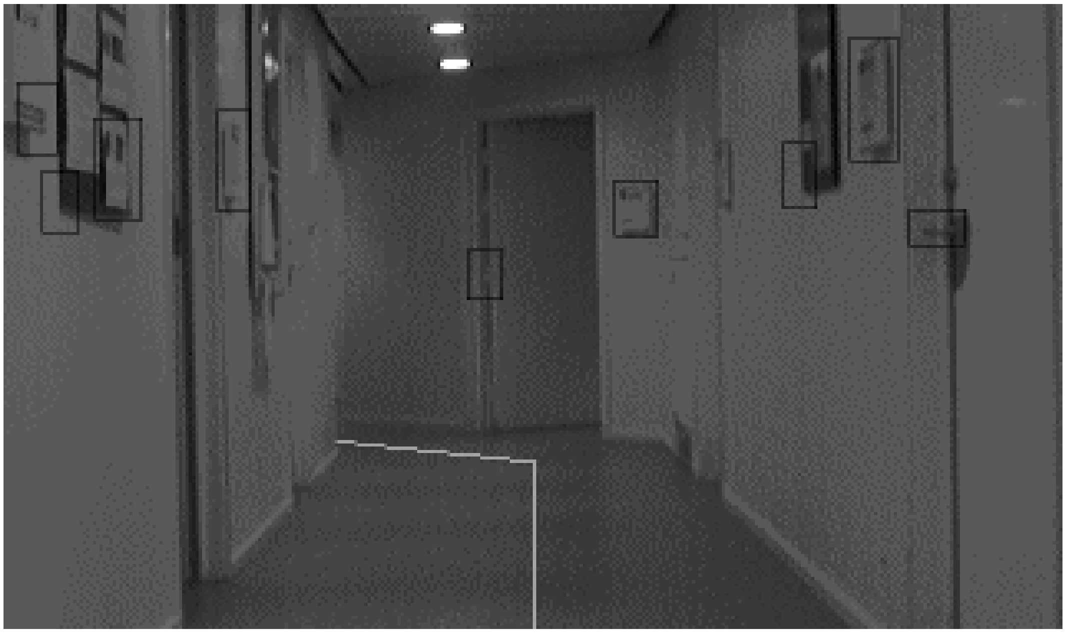
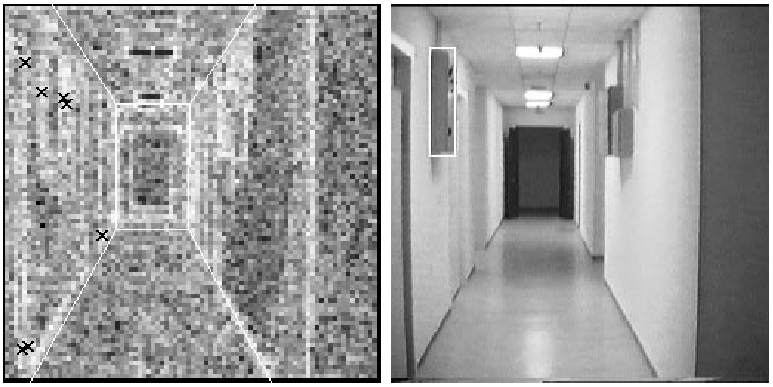


Figure 3.4: *Black rectangles indicate detected landmark candidates. White line is planned robot path (not important for the purpose of this illustration). Reprinted with permission from [2].*



(a) (b)

Figure 3.5: *(a) Saliency map, (b) extracted landmark.*

It is evident from the review in this report that there is an abundance of systems using pre-stored landmarks in many forms. On the contrary, there are only few efforts, mostly confined at research level, that have addressed the problem of automatic landmark acquisition or learning. However, increased importance is currently being paid to this area, which is expected to grow to a very active research area in the near future.

3.3 Landmark recognition

3.3.1 Definitions

It is impossible to discuss the subject of landmark recognition without simultaneously touching upon the subject of landmark representation. Recognition implies some kind of matching process between incoming, new data and stored model data, and such matching processes require incoming and stored data to have the same representation. One such example is the *edge direction representation* as defined in Section 3.1.

In the subsequent discussions it is important to bear in mind that a landmark can have three different guises: 1) the physical landmark in the 3D environment, e.g. a door, 2) the stored representation of a landmark in system memory, and 3) the appearance of a landmark (or the representation thereof) in an image.

For an actual system implementation, there is a trade-off between the recognition complexity needed for the employed landmarks, and the descriptive power (with a direct effect on self-localization and, therefore, navigation) of the landmark patterns. For example, Carlsson [7] uses vertical lines (edge segments) as landmarks, and 'recognition' of the individual lines degenerates to forming associations between stored edge segments and the ones detected in an incoming image. However, vertical lines may not suffice for resolving all possible ambiguities regarding localization, which is equivalent to inapropriate descriptive power of the selected landmarks.

Finally, the choice of representation for landmarks in system memory is intimately linked to the choice of algorithms or techniques for landmark recognition. As an example, section 3.3.2 describes the use of raw pixel arrays (templates) for representing landmarks in memory, combined with the use of template matching techniques for recognizing instances of the landmarks in incoming images.

3.3.2 Landmark recognition approaches

Template matching

A common choice for physical landmarks are planar patterns in the environment, e.g. door signs, light switches, posters, etc. Such landmarks can be represented directly by their appearance in images, i.e. as pixel arrays (templates). The problem with such a representation is that it is not rotation and scale invariant. That is, the appearance of the landmarks varies with the viewpoint and hence the robot position.

Therefore, this representation is more appropriate for model based navigation sys-

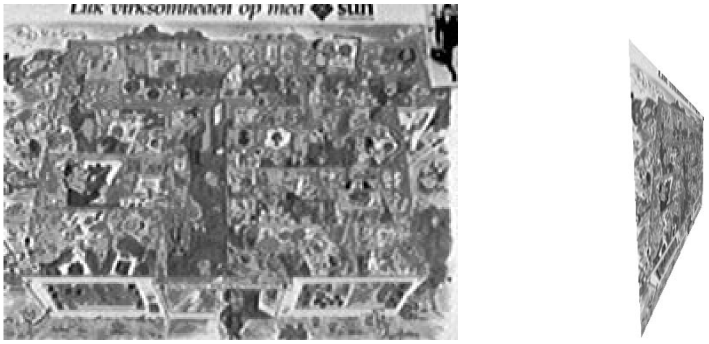
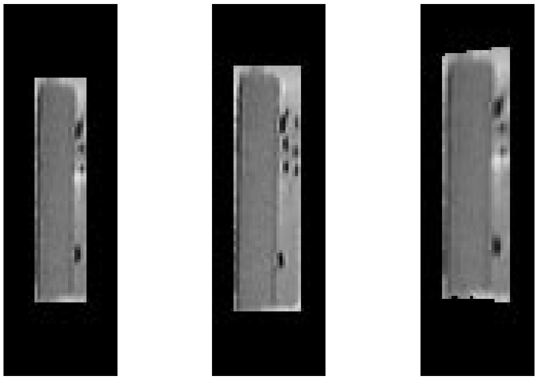


Figure 3.6: *Left: stored pixel data for a landmark. Right: landmark pixel data re-mapped to oblique view to obtain a template for recognizing the image location of the landmark.*



(a) (b) (c)

Figure 3.7: *Landmark pattern extracted in (a) learning phase, and (b) navigation phase; (c) pattern in (a) after landmark transformation.*

tems, where the position and orientation of the physical landmarks is known. Furthermore, a fairly accurate estimate of the robot position must be available. Given known camera focal length and pixel aspect ratio, a stored set of pixel data for a particular landmark can be transformed to a new set of pixel data that represent the landmark as it should appear in the image from a particular viewpoint and viewing direction. This is illustrated in Fig. 3.6, where the stored pixel data for a landmark has been transformed to another view. This transformed landmark template can then be used to locate the precise position of the landmark in the image.

Trahanias et al. [42] were able to perform the landmark transformation without the constraint of known landmark and robot positions. By assuming workspaces that form orthogonal parallilepipeds and constrained egomotion of the observer, landmark transformation is accomplished using only quantities that are readily available from the image data. A result of this procedure is demonstrated in Fig 3.7.

Landmark recognition can be performed using the normalized cross-correlation coefficient [19]. By computing the correlation between image and perspectively re-mapped landmark pixel data at various positions in the image (within a search window), it is possible to find the exact location of the landmark in the image. Figures 3.8, 3.9 and 3.10 illustrate this for a navigation example. These figures are reprinted with permission from

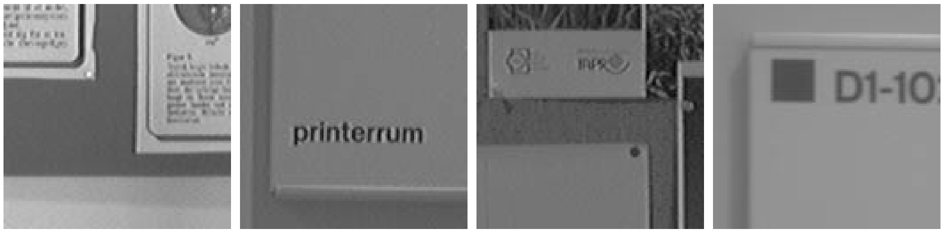


Figure 3.8: *Stored pixel data of 4 landmarks. The regions used as landmarks were selected automatically from an image of a hallway using an attention scheme as described in section 3.2.*

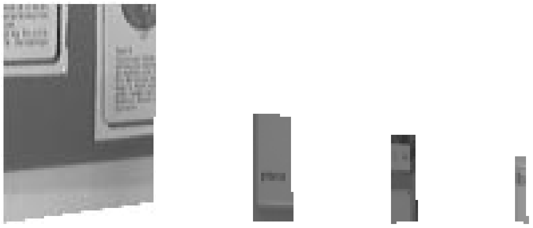


Figure 3.9: *Landmark templates obtained from perspective re-mapping of the stored pixel data shown in Fig. 3.8. The re-mapping is based on an estimated camera position corresponding to the hallway view in Fig. 3.10.*

[41] and [2].

Neural network based image matching

The above described template matching is a possible approach to iconic (pixel based) image matching. An alternative method used for mobile robot self-localization is described in [9], where a neural network (NN) is trained on images obtained from a grid in the environment, and the output of the NN is trained to encode robot position. Alternatively, training images are obtained during a run of a path by human guidance. The robot can then later retrace the path. Images are formed by pointing the camera to an aluminum cone acting as a conical mirror. This provides an omni-directional (360°) view of the environment. Images are collapsed into 1D signals and the values fed directly into the input layer of a NN.

This approach is mentioned in this context since it illustrates 1) an alternative technique to image matching, and 2) how widely the concept of landmarks can be stretched. In this example the notion of landmarks is used in a broad sense, since whole images are in essence used as landmarks. The image data are stored in advance and used for position correction by matching current visual input to the stored images. In other words, the stored images themselves effectively *become* landmarks.

A similar concept is employed in [30], with the only difference that the images are foveated views with the optical axis parallel to the ground plane; in this approach, a correlation measure is used for iconic image matching.

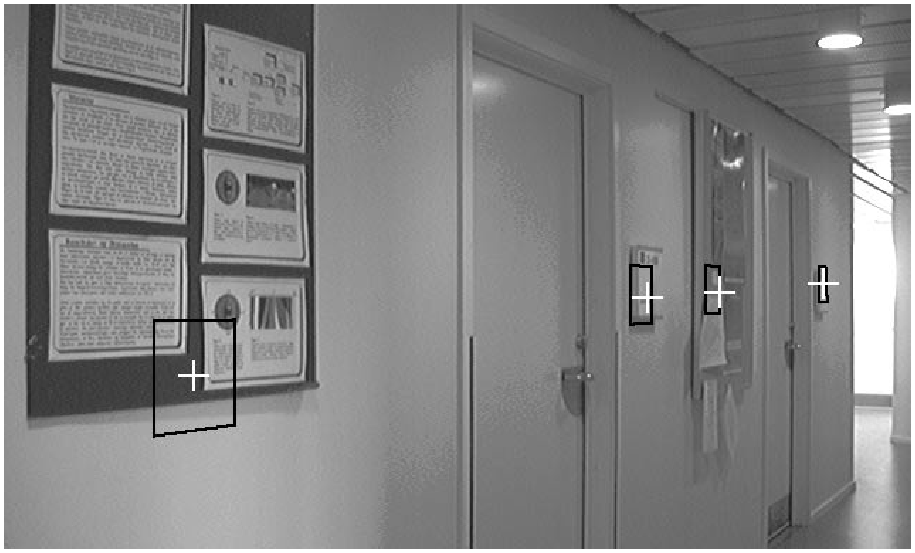


Figure 3.10: *Locations of four landmarks detected using normalized cross-correlation after performing perspective re-mapping of the stored landmark pixel data, (Fig. 3.9). The black boxes are the search regions and the white crosses are the loci of maximum correlation coefficient.*

Edge based landmark recognition

Structural landmarks, such as doors, can be represented and recognized based on perceptual groupings of edge segments. Christensen et al. [10] describe a system were doors are used as landmarks for aiding the robot self-localization process by recognizing doors using the three edge segments comprising the door frame. The generic door (and thus landmark) model consists of three co-planar line segments. If door locations are stored in a 3D environment model, and given a rough position and heading estimate for the robot, the door model can be projected to the image and matched to detected edge segments.

The described dedicated door-recognition approach is naturally a special case of using more general 3D object recognition schemes for recognizing structures in the environment and using them for landmarks. For example, [47] demonstrates an edge based 3D object recognition approach to localize pillars.

Landmark recognition using projective invariants

Various visual or visually-guided robotics tasks may be performed using only a projective representation, which shows the importance of projective information at different steps in the perception-action cycle. Since the introductory papers [16, 17, 21] on computing projective structure using uncalibrated cameras, there has been a high interest in developing reliable algorithms for uncalibrated stereo, for a binocular moving agent. One of the reasons is that a number of visual or visually-guided robotics tasks may be carried out using only a projective representation. We can mention here the obstacle detection and avoidance [37] or goal position prediction for visual servoing [5, 13].

Another reason is that more recently it has been shown, both theoretically and experimentally, that under certain conditions an image sequence taken with an uncalibrated camera can provide 3D Euclidean structure as well. The latter paradigm consists in recovering projective structure first and then upgrading it into Euclidean structure [3, 14]. A more complete review can be found in [38].

Additionally, it is generally accepted that computing structure without explicit camera calibration is more robust than using calibration because we need not make any (possibly incorrect) assumptions about the Euclidean geometry (remembering that calibration is itself often erroneous). The above demonstrate the importance of projective geometry both in computer vision and robotics, and the various applications show that projective information can be useful at different steps in the perception-action cycle. Since the study of every geometry is based on the study of properties which are invariant under the corresponding group of transformations, the projective geometry is characterized by the projective invariants. In active visual navigation, projective invariants offer an interesting alternative [4] for non-metric navigation, and constitute an active research approach, besides other contemporary approaches that employ model-based perception for mobile robot navigation [22] or landmark-based navigation for the acquisition of environmental models [34].

In order to model environment objects, the most common non-parametric model—either implicit or explicit—is to consider obstacles or landmarks as *planar* objects [45] and to introduce specific constraints on the vehicle displacement [40]. In both cases, these assumptions do not rely on calibration. By restraining models of the scene to the simple case of piece-wise planar patches, it has been possible to derive theoretical results and also to implement in a somewhat robust way image motion estimation, with a precision sufficient to infer the required 3D parameters of the scene.

Invariants for object recognition is the next most important topic related to our problem, when looking for landmarks in the visual surroundings. In fact, 3D object recognition using invariance is not specific to landmarks; recognition of planar objects has been addressed by employing semi-local projective invariants [8], which has also been extended to non-planar objects [48]. Generally, invariant descriptors for 3D object recognition and pose estimation [17] have been constructed, while using projective invariants for constant time library indexing in model based vision [39] has been also experimented.

3.3.3 Conclusions and remarks

Landmark recognition has been approached to-date using various techniques from the field of *pattern recognition*. A key issue, common to many recognition tasks, is that of finding landmark representations that are invariant to rotation (and possibly scaling) and recognition schemes that support these representations. Such representations do exist, for example straight, vertical edge segments, but typically suffer from lack of descriptive power.

Two-dimensional patterns such as door signs, posters, etc. are very salient and offer descriptive power, but do not in general offer any kind of rotational invariance. Specially designed patterns and added to the environment for the explicit purpose of serving as landmarks may improve on this situation. A design which lends itself well to the task in hand is a circle or square with four quadrants, where quadrant 1 and 3 are black, and

2 and 4 white (this pattern is often seen on crash test dummies). However, approaches that are based on such patterns can only function in pre-engineered environments and are excluded from dynamic and/or changing environments. In these cases, the research efforts are currently directed towards introducing appropriate attention mechanisms for landmark selection, whereas the employed pattern representation usually dictates the approach for landmark recognition. As such, traditional image matching techniques are usually adopted.

Approaches that use representations employing projective invariants offer an appealing alternative to the problem of landmark recognition. Invariant descriptors for 3D object recognition have been constructed, and the use of projective geometry in visual navigation tasks seems quite promising.

Bibliography

[1] C. S. Andersen. *A Framework for Control of a Camera Head*. PhD thesis, Laboratory of Image Analysis, Aalborg University, Denmark, January 1996.

[2] S. Andersen et al. Robot-navigation. Student project report (No number), Laboratory of Image Analysis, Aalborg University, Fr. Bajers Vej 7D1, DK-9220 Aalborg East, December 1996. In Danish.

[3] M. Armstrong, A. Zisserman, and P. Beardsley. Euclidean structure from uncalibrated images. In E. Hancock, editor, *Proceedings of the 5th British Machine Vision Conference*, pages 508–518, York, UK, Sept. 1994. BMVA Press.

[4] P. A. Beardsley, I. D. Reid, A. Zisserman, and D. W. Murray. Active visual navigation using non-metric structure. In *Proceedings of the 5th International Conference on Computer Vision* Boston, MA, pp.58–64, June 1995.

[5] R. C. Bolles and R. Horaud. 3dpo: A tree-dimensional part orientation system. *International Journal of Robotics Research*, 5(3):3–26, 1986.

[6] R.A. Brooks, "Visual map for making a mobile robot", in *Proc. IEEE Int. Conf. on Robotics and Automation*, St. Louis, MO, pp. 824-829, 1985.

[7] S. Carlsson. Relative positioning from model indexing. *Image and Vision Computing*, 12(3):179 – 186, April 1994.

[8] S. Carlsson, R. Mohr, T. Moons, L. Morin, C. Rothwell, M. Van Diest, L. Van Gool, F. Veillon, and A. Zisserman. Semi-local projective invariants for the recognition of smooth plane curves. *The International Journal of Computer Vision*, 19(3), pp.211-236, 1996.

[9] R. Cassinis, D. Grana, and A. Rizzi. Self-localization using an omni-directional image sensor. In *Proceedings: 4th International Symposium on Intelligent Robotic Systems, Lisbon, Portugal*, pages 215 – 222, July 1996.

[10] H.I. Christensen, N.O. Kirkeby, S. Kristensen, L. Knudsen, and E. Granum. Model-driven vision for in-door navigation. *Robotics and Autonomous Systems*, (12):199 – 207, 1994.

[11] J. Clark and N. Ferrier. Attentive Visual Servoing. In A. Yuille A. Blake, editor, *Active Vision*, Artificial Intelligence, chapter 9, pages 137–154. MIT Press, Cambridge, MA, 1993.

[12] Sean M. Culhane and John K. Tsotsos. An attentional prototype for early vision. In G. Sandini, editor, *Proceedings: Second European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 551–560, May 1992.

[13] G. Csurka. *Modélisation projective des objets tridimensionnels en vision par ordinateur.* PhD thesis, University of Nice, Sophia-Antipolis, France, 1996.

[14] F. Devernay and O. Faugeras. From projective to euclidean reconstruction. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 264–269, San Francisco, CA, June 1996. IEEE.

[15] J. Dias, F. Bergholm, P. Fornland, " VIRGO, Vision-Based Robot Navigation Research Network", Report, TRITA-NA-P9627, ISRN KTH/NA/P-96/27, Royal Inst. of Technology, Stockholm, Sweden, Sept. 1996.

[16] O. Faugeras. What can be computed in three dimensions with an uncalibrated stereo rig. *Journal of the Optical Society of America*, 1993. Submitted.

[17] D. Forsyth, J. L. Mundy, A. Zisserman, C. Coello, A. Heller, and C. Rothwell. Invariant Descriptors for 3D Object Recognition and Pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):971–991, Oct. 1991.

[18] J. Gårding, *Shape from Surface Markings*, Ph.D Thesis, TRITA-NA-P9110, ISRN KTH/NA/P-91/10, Royal Inst. of Technology, Stockholm, Sweden, May 1991.

[19] R.C. Gonzales and R.E. Woods. *Digital Image Processing.* Addison-Wesley Publishing Company, 1992.

[20] R. Greiner and R. Isukapalli. Learning to select useful landmarks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(3):437 – 449, June 1996. Special issue on learning in autonomous robots.

[21] R. Hartley. Projective reconstruction and invariants from multiple images. *Pattern Analysis and Machine Intelligence*, 16(10):1036–1040, 1994.

[22] D. Kriegman and T. Binford. Model-based perception for mobile robots. In *Multisensor Fusion and Environment Modelling*, 1989.

[23] K-D. Kuhnert, "Fusing dynamic vision and landmark navigation for autonomous driving", in *IEEE Int. Workshop on Intelligent Robots and Systems*, IROS '90, pp. 113 - 119, 1990.

[24] R. Milanese, S. Weschler, H. Gil, J.-M. Bost, and T. Pun. Integration of bottom-up and top-down cues for visual attention using non-linear relaxation. In *Proceedings: IEEE Conference on Computer Vision and Pattern Recognition, Seattle, Washington*, pages 781 – 785, June 1994.

[25] Ruggero Milanese. Detection of salient features for focus of attention. In *Proceedings: 3rd Meeting of the Swiss Group for Artificial Intelligenc eand Cognitive Science*, Biel-Bienne, Switzerland, October 1991. World Scientific Publishing.

[26] Ruggero Milanese. *Detecting Salient Regions in an Image: from Biological Evidence to Computer Implementation.* PhD thesis, Department of Computer Science at university of Geneva, Switzerland, Geneva, Switzerland, December 1993.

[27] H., Mori, H. Oowa, S. Kotani, S. Yasutomi, H. Ishiguro, Y. Chino, H. Niki, "Environmental understanding of Mobile Robot 'Harunobu-3' by picture interpretation language PILS V-3", pp. 158-161, 8:th ICPR, Paris, 1986.

[28] J. Neira, I. Ribeiro, and J. D. Tarós. Mobile robot localization and map building using monocular vision. In *Proceedings: 5th International Symposium on Intelligent Robotic Systems, Stockholm, Sweden*, July 1997. To appear.

[29] R. Nelson, "Visual Navigation", Ph.D Thesis, CAR-TR-380, Univ. of Maryland, College Park, Maryland, Aug. 1988.

[30] J. Nielsen and G. Sandini. Learning mobile robot navigation: A behaviour-based approach. In *Proceedings: IEEE Conference on Systems, Man and Cybernetics, San Antonio, Texas*, October 1994.

[31] Lars Nielsen, *Simplifications in Visual Servoing*, Ph. D thesis, Studentlitteratur, LUTFD2/(TFRT-1027)/1-153/(1985), Dept. of Automation Control, Lund Inst. of Technology, Sweden, 1985.

[32] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context-free attentional operators: The generalized symmetry transform. *International Journal of Computer Vision*, 14(2):119 – 130, March 1995.

[33] W. Richards, "How to play twenty questions with Nature and win", Tech. Report, A.I. Memo No. 660, Mass. Inst. of Technology, AI Lab, Massachussets, USA, December 1982.

[34] E. Riseman, A. Hanson, J. Beveridge, R. T. Kumar, and H. Sawhney. *Landmark-Based Navigation and the Acquisition of Environmental Models*, chapter 11, pages 317–374, in Yannis Aloimonos ed. "Visual Navigation : From Biological Systems to Unmanned Ground Vehicles". Lawrence Erlbaum, Mahwah, New Jersey, 1993.

[35] L. Robert, M. Buffa, and M. Hebert. Weakly-calibrated stereo perception for rover navigation. In *Proceedings of the ARPA Image Understanding Workshop*, pages 1317–1323. Defense Advanced Research Projects Agency, Morgan Kaufmann Publishers, Inc., 1994.

[36] L. Robert and O. Faugeras. Relative 3d positioning and 3d convex hull computation from a weakly calibrated stereo pair. Technical Report 2349, INRIA, Sept. 1994.

[37] L. Robert, C. Zeller, O. Faugeras, and M. Hébert. Applications of non-metric vision to some visually-guided robotics tasks. In Y. Aloimonos, editor, *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, chapter ? Lawrence Erlbaum Associates, 1996. to appear, also INRIA Technical Report 2584.

[38] C. Rothwell, G. Csurka, and O. Faugeras. A comparison of projective reconstruction methods for pairs of views. In *Proceedings of the 5th International Conference on Computer Vision*, Boston, MA, pp.932–937, June 1995.

[39] C. Rothwell, A. Zisserman, D.A.Forsyth, and J. Mundy. Using projective invariants for constant time library indexing in model based vision. In P. Mowforth, editor, *Proceedings of the 2nd British Machine Vision Conference*, pages 62–70, Glasgow, UK, Sept. 1991. Springer-Verlag.

[40] C. A. Rothwell, A. Zisserman, D. A. Forsyth, and J. L. Mundy. Planar object recognition using projective shape representation. *The International Journal of Computer Vision*, 16(1):57–99, Sept. 1995.

[41] J. S. Sørensen. A method for selecting landmarks minimizing the uncertainty of the estimated position for a landmark based navigation system. Master's thesis, Laboratory of Image Analysis, Aalborg University, Fr. Bajers Vej 7D1, DK-9220 Aalborg East, August 1996.

[42] P.E. Trahanias, S. Velissaris and T. Garavelos. Visual landmark extraction and recognition for autonomous robot navigation. In em Intl. Conf. on Intelligent Robots and Systems, IROS'97, Grenoble, France, Sept. 1997. To appear.

[43] T. Viéville, Q. Luong, and O. Faugeras. Motion of points and lines in the uncalibrated case. *International Journal of Computer Vision*, 17(1), 1995.

[44] T. Viéville, F. Romann, B. Hotz, H. Mathieu, M. Buffa, L. Robert, P. Facao, O. Faugeras, and J. Audren. Autonomous navigation of a mobile robot using inertial and visual cues. In M. Kikode, T. Sato, and K. Tatsuno, editors, *Intelligent Robots and Systems*, Yokohama, 1993.

[45] T. Viéville, C. Zeller, and L. Robert. Recovering motion and structure from a set of planar patches in an uncalibrated image sequence. In *Proceedings of the International Conference on Pattern Recognition*, pages 637–641, Jerusalem, Israel, Oct. 1994. Computer Society Press.

[46] T. Viéville, C. Zeller, and L. Robert. Using collineations to compute motion and structure in an uncalibrated image sequence. *The International Journal of Computer Vision*, 18(2), 1995.

[47] F. Wallner and R. Dillmann. Reactive sensor control for a mobile robot. In *Proceedings: 4th International Symposium on Intelligent Robotic Systems, Lisbon, Portugal*, pages 41 – 48, July 1996.

[48] A. Zisserman, D. Forsyth, J. Mundy, C. Rothwell, J. Liu, and N. Pillow. 3D object recognition using invariance. *Artificial Intelligence Journal*, 78:239–288, 1995.

Chapter 4

Biologically Inspired Approaches to Navigation

by Rolf Pfeifer, Raja Dravid and Ralf Möller

A new discipline has been established between 'Artificial Intelligence' and 'Cognitive Science', sometimes referred to as 'New AI' or 'Behavior-based AI'. The new field has provided scientists and engineers with new ways of thinking about intelligent systems. One of the core insights of 'New AI' is that the agent-environment interaction plays the central role in the understanding and design of intelligent systems. Based on this approach, a number of new design principles for autonomous agents have been derived.

These principles can be applied to the central ability of autonomous agents, namely navigation, that is, the ability of relocating to a region of the environment once occupied by the agent. In most approaches, navigation is considered as an information processing capability, attributing cognitive abilities to the agent. Only a few have investigated the case of simple rules capable of producing complex navigational behavior, i.e., behavior steered by direct information about the environment.

This chapter will provide examples for two vision-based, navigational mechanisms performed by biological agents—'path integration' and 'visual piloting'—which have been implemented on mobile robots.

4.1 Design principles for autonomous agents

4.1.1 Definitions

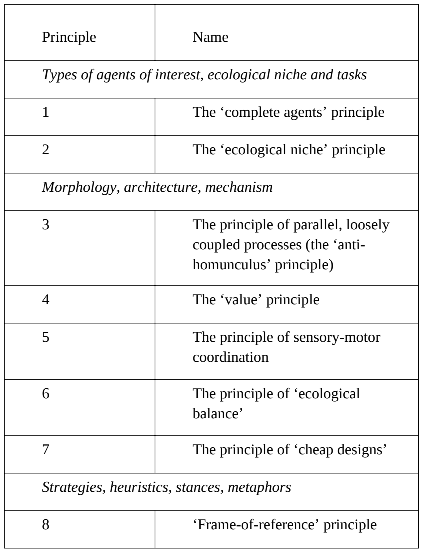
The word 'animat' has been coined to describe the discipline where animal behaviors may be described in terms of existing or new robot control paradigms and vice versa, robot control paradigms may be phrased in biological inspired terms.

An example of an animat would be a neural net with visual capabilities looking for food, choosing among candidates and navigating according to this.

4.1.2 An Overview of design principles

Design principles provide guidance on how to build animats. The way we build our animats is a manifestation of our views of intelligence. One purpose of the design principles is to make this knowledge explicit. There are three classes of design principles; an overview is given in Fig. 4.1: The first class concerns the kinds of agents and the behaviors that are of interest from a cognitive science perspective. The second class concerns the agent itself, its morphology, its sensors and effectors, its control architecture and its internal mechanisms. The third class contains principles that have to do with way of thinking and proceeding, with stances, attitudes and strategies to be adopted in the design process [9, 10].

In the following, we will concentrate on the design principles that are especially relevant to the models of navigation and their implementation on mobile robots.



Principle	Name
Types of agents of interest, ecological niche and tasks	
1	The 'complete agents' principle
2	The 'ecological niche' principle
Morphology, architecture, mechanism	
3	The principle of parallel, loosely coupled processes (the 'anti-homunculus' principle)
4	The 'value' principle
5	The principle of sensory-motor coordination
6	The principle of 'ecological balance'
7	The principle of 'cheap designs'
Strategies, heuristics, stances, metaphors	
8	'Frame-of-reference' principle

Figure 4.1: *Summary of design principles. After [10], with authors' permission.*

Principle 1: The 'fungus-eaters' principle

The 'Solitary Fungus Eater' (see Fig. 4.2) is a creature—in our terminology an autonomous agent—sent to a distant planet to collect uranium ore. The more ore it collects, the more reward it will get. It feeds on certain types of fungus which grow on the planet. The 'Fungus Eater' has a fungus store and means of locomotion, means for decision making and collection. Any kind of activity requires energy. If the level of fungus drops to zero, the 'Fungus Eater' stops working. The 'Fungus Eater' is also equipped with sensors, one for vision and one for detecting uranium ore.

This scenario (described by Toda [13]) is interesting as a design principle in a number of respects. The 'Fungus Eaters' must be *autonomous*: they are simply too far away to be

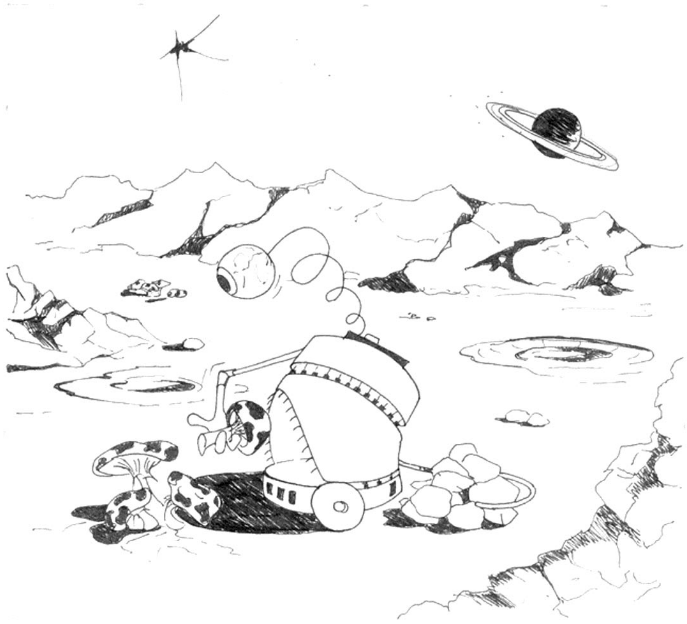


Figure 4.2: *A 'Fungus Eater' ingesting fungus on a distant planet. It has to perform its task autonomously while maintaining its energy supply (cartoon by Isabelle Follath, Zürich). Taken with authors' permission from [10].*

controlled remotely. They must be *self-sufficient*, since there are no humans to change the batteries and repair the robots, and they must be *adaptive*, since the territory in which they have to function is largely unknown.

Autonomous agents must be *embodied*, i.e., they must be realized as a physical system capable of acting in the real world [2]. Therefore, the experiments described below have been carried out with real robots. In difference to other approaches where the designer's perspective is imposed on the agent, the design principles emphasize *situatedness*: the whole interaction with the environment must be controlled by the agent itself, i.e., the world must always be seen from the perspective of the agent. Adaptivity can be seen as one aspect of situatedness, in that the agent has to be able to bring in its own experience in dealing with the current situation.

From the viewpoint of navigation, self-sufficiency and situatedness (including adaptivity) are crucial design principles: The agent has to find its way back to a feeding location at the right time (self-sufficiency) and it has to adapt to an initially unknown territory or even to special properties of the environment. An example for adaptivity in the context of navigation is the 'Adaptive Light Compass' illustrated below. This control architecture is adaptive by using a self-organizing neural network, that associates signals coming from light sensors with the corresponding motor actions.

Principle 2: The 'ecological niche' principle

Whenever designing an agent, we first have to specify the ecological niche, i.e., the environment in which the agent has to perform specific tasks. The agent can exploit properties of its environment, but on the other hand has to cope with circumstances that complicate the execution of a given task.

In the path integration example given below, the mobile robot 'Sahabot' is inspired by the desert ant 'Cataglyphis' and thus operates in the same ecological niche. The basic navigational task of the ant is to forage for food and carry it back to the nest. The extreme heat of the desert ground prohibits the use of pheromone trails for navigation, therefore, the ant has to rely on a different method of navigation. It uses the polarization pattern of the cloudless sky to extract compass information.

Principle 3: The principle of loosely coupled processes

This principle states that intelligence or cognition is emergent from a large number of parallel, loosely coupled processes. These processes run asynchronously and largely peripheral, requiring little or no centralized resources. Principle 3 could also be called the 'anti-homunculus' principle. It is directly motivated from biology. The 'adaptive light compass' described below incorporates this principle, in that two independent processes (avoidance network and adaptive light compass) are superimposed on the motor signals.

Principle 7: The principle of cheap design

The principle of 'cheap design' states that good designs are 'cheap'. A cheap design can be achieved by exploiting system-environment interaction. In navigation, this means, that homing does not have to rely on a cartesian map and a planning process. Instead, the agent makes a decision based on the current sensory situation. Through this action, it is transferred into the next situation, that triggers another decision, and so on [2].

Other design principles (4, 5, 6)

These principles are not directly related to navigation, but play an important role as basis for an alternative approach to cognition and categorization. The 'value principle' states that the autonomous agent has to be embedded in a value system, which lets it decide what is good for it and what isn't. The principle of 'sensory-motor coordination' says that the interaction with the environment is to be conceived as a sensory-motor coordination instead of being a sense-think-act cycle. The principle of 'ecological balance' states that there has to be a match between the 'complexity' of the sensors, the actuators, and the neural substrate.

4.2 Navigation of autonomous agents

The mechanisms of visual navigation to be discussed are based on work on the neurobiology of visually guided behavior in bees and ants and has been done by combining behavioral analysis, neuro-physiological and neuro-anatomical work with experimental work on real robots. There are two basic mechanisms:

1. *Dead reckoning (path integration):* It has been found, that certain animals continuously keep track of their own positions relative to home (homing-vector model) [14]. Of special interest is the type of information that natural agents use when they employ such a mechanism. The desert ant 'Cataglyphis', for example, uses polarized light information to determine its orientation.

2. *Visual piloting:* Besides dead reckoning, certain animals use a visual piloting system to finally pinpoint the position of a goal. An essential part of this piloting system is the neural formation of a visual 'snapshot' of the landmark skyline surrounding the goal. Dead reckoning is normally used with more fine-grained navigation mechanisms for precise homing. After having arrived in the vicinity of the starting position, certain animals switch to more precise mechanisms for finding the nest. This is normally called 'pin-point navigation' [4]. Albums of such snapshots offer a way for 'map-building'. We put map-building in quotes to indicate that we do not understand 'map' in the traditional sense of the word, but rather as a property emerging from sensory-motor coordination.

4.3 Path integration

4.3.1 Navigating with an adaptive light compass

One of the fundamental abilities required in autonomous mobile agents is the one of 'homing'. Natural agents like ants solve this problem by mainly using dead-reckoning mechanisms with an egocentric frame of reference. Here we present a biologically inspired orientation mechanism, an 'adaptive light compass', that was used for homing in a mobile robot equipped with infrared and ambient light sensors (see Fig. 4.3) [7]. The compass uses ambient light information to provide orientation cues.

The general design of the control architecture is shown in Fig. 4.4: Two neural networks are responsible for the locomotion of the robot. A self-organizing neural net (the 'adaptive light compass') takes input from the ambient light sensors and outputs to the motors (the light source used in the experiments was the laboratory window). The task of the network is to keep track of the current direction and steer the robot in such a way that it always moves in one direction. The second network is responsible for the obstacle avoidance behavior of the robot. It is a network with fixed weights inspired by Braitenberg [1].

The adaptive light compass is based on a self-organizing 'Extended Kohonen Map' [6, 11]. Its structure is depicted in Fig. 4.5. Each of the cells of the two-dimensional lattice associates an incoming weight vector (corresponding to the ambient light vector) with an

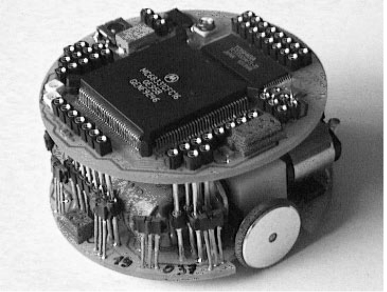


Figure 4.3: *The mobile robot (Khepera) used for the experiments with the adaptive light compass. It is equipped with 8 infra-red proximity sensors, also used to measure ambient light. Taken with author's permission from [7].*

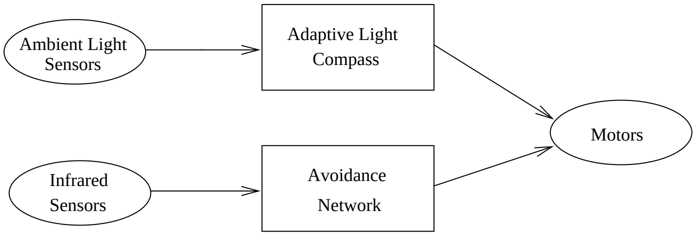


Figure 4.4: *Control architecture of robot for the adaptive light compass experiment. Taken with author's permission from [7].*

outgoing weight vector (corresponding to the motor signals). The extended Kohonen model allows the system to learn a control task, when a sequence of correct input-output pairs is available. In this case, the networks task is to compensate for the turns arising from the avoidance network. The input-output pairs are constructed as follows: every time the agent leaves its home position it performs an 'orientation movement' (usually a turn of 360 degrees) during which is collects data from the ambient light sensors and the associated direction as derived from the wheel encoders. This sets up the relation between the compass direction and the homing direction, and when deviating from the homing direction, the correct speed for the motors is set to be proportional to the angle that has to be compensated. In other words, if the robot has been forced to deviate from the original direction because of an avoidance movement, the contribution of the net to the motors will be proportional to the angle of deviation.

For an evaluation of the overall performance of the homing mechanism different environments like the one depicted in Fig. 4.6 have been used. For each experiment, the robot was made to cover a distance of 110 cm (20 times its body diameter) during the outward journey and then go back home. For leading the agent back home, the outgoing connections of the map to the motor units are exchanged.

Figure 4.7 shows a typical path of the robot. The activation of the network is displayed at several points corresponding to different headings of the robot. It is interesting, how the resulting activity on the net corresponds to different directional angles. Maximum

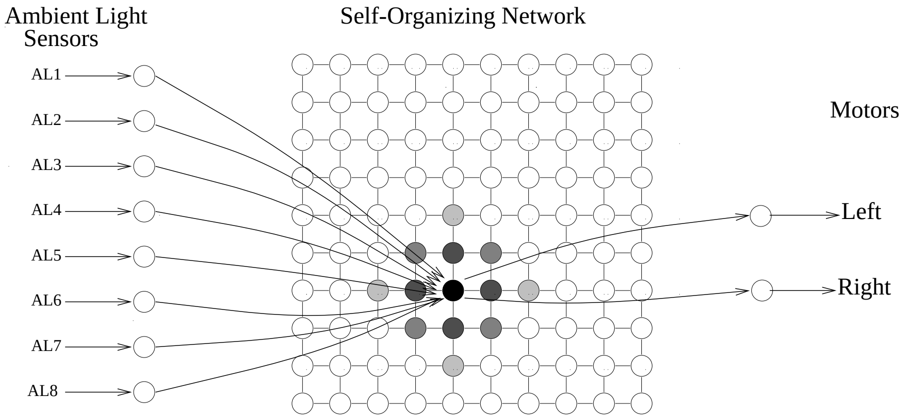


Figure 4.5: *Self-organizing neural network used for the adaptive light compass. Taken with author's permission from [7].*

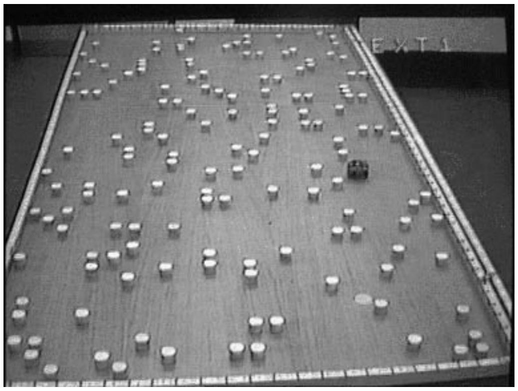


Figure 4.6: *Environment with a random distribution of obstacles used for the adaptive light compass experiments. Taken with author's permission from [7].*

activity occurs at specific cells of the net for certain angles. The maximum activation area is shifted around the net in a compass-like manner. For the environment shown in Fig. 4.6, the mean error between initial and final position is 3.6%.

4.3.2 Navigating with a polarized light compass

For dead-reckoning mechanisms, compass information has a crucial effect on the precision of homing. Certain insects (like the desert ant 'Cataglyphis') use the pattern of polarized light in the sky that arises due to scattering of sun-light in the atmosphere. The analysis of skylight polarization is mediated by specialized photoreceptors and neurons in the ant's visual system. Inspired by the insect's polarized light compass, [8] describes an implementation of a polarization navigation system that was successfully employed on the mobile robot 'Sahabot' (see Fig. 4.8).

There are 6 polarized light sensors (POL-sensors) arranged in pairs with each pair attached on a mechanical device for angular adjustment placed on the back part of the

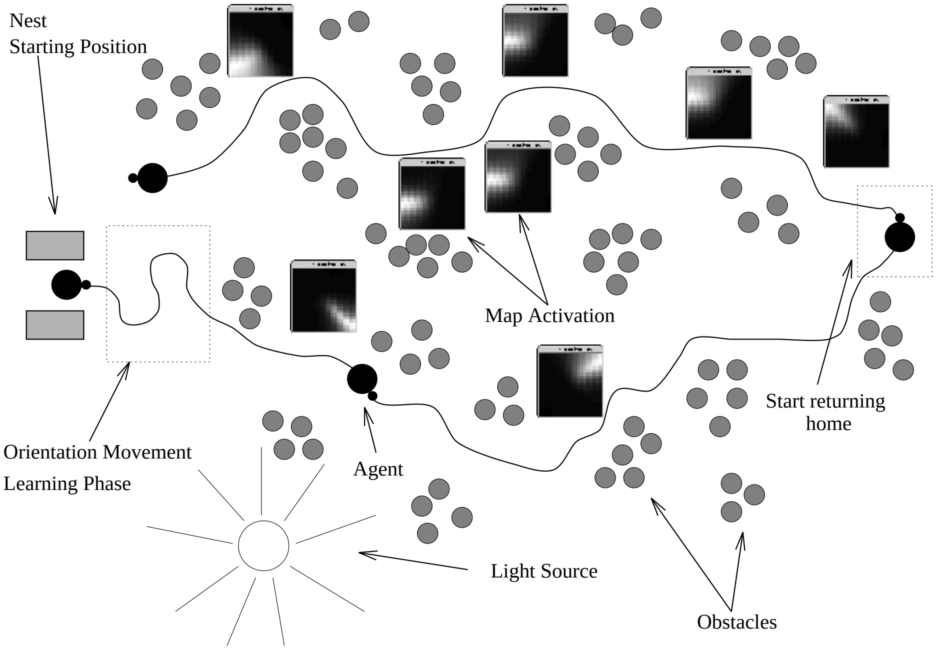


Figure 4.7: *A typical path of the robot in the adaptive light compass experiment. The activity of the map is shown at specific points of the path corresponding to different orientations from the light source. Taken with author's permission from [7].*

robot. Furthermore, the robot is equipped with a set of 8 directional light intensity sensors (DLI-sensors) mounted on the front part of the robot and with two incremental encoders placed on the motor axes.

One POL-sensor pair is enough to determine the orientation of the agent: The robot scans the sky by moving in a circle (mimicking the rotation of the ant about its body axis). The maximum response of the POL-sensor pair will occur when the body axis of the robot is parallel to the symmetry line of the polarization pattern (solar-antisolar meridian). The ambient light sensors are used to resolve the ambiguity between the solar- and the antisolar line. With two additional POL-sensor pairs, the response curve can be calculated that has sharper peaks during rotation.

The experimental procedure is depicted in Fig. 4.9, see the caption for a description. The average angular error between initial and final position of the robot was 1.5%, the positional error 2-3% of the total distance travelled (68m and 136m).

4.4 Visual piloting

It is known, that insects like ants and bees preferably rely on dead-reckoning mechanisms like those described in the previous section [15]. Additionally, there is evidence for the use

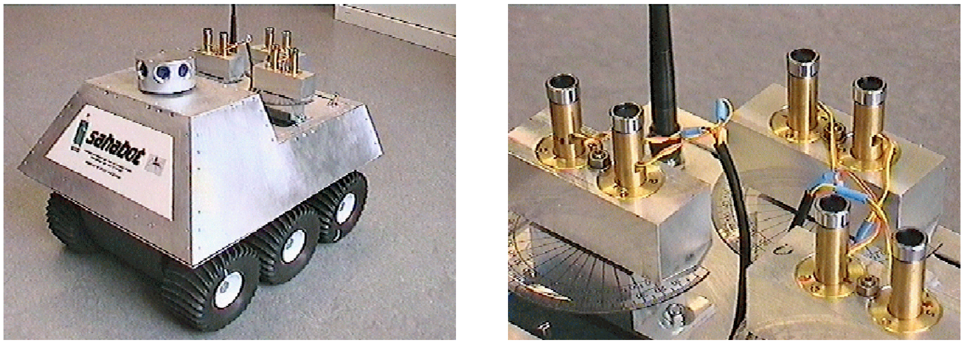


Figure 4.8: *Example of a robot designed for a particular ecological niche: the so-called Sahabot (for Sahara Robot). It has to operate in the Sahara desert. For this special niche, the robot is equipped with ambient light sensors (top part in the left figure) and UV polarized light sensors (depicted in the right figure). The robot is used for experiments involving the navigation behavior of the desert ant* Cataglyphis. *It has been built by Dimitrios Lambrinos, Hiroshi Kobayashi and Marinus Maris of the AI Lab Zürich together with the Zoology Department of the University of Zürich (headed by Prof. Wehner). Taken with authors' permission from [10].*

of visual landmarks for navigation. Cartwright and Collett proposed a so-called 'snapshot model' [3] describing the homing behavior of bees. The basic assumption of this model is, that the bee takes a visual snapshot of the nest's surrounding and uses this snapshot to finally pinpoint the goal when returning. This is accomplished by means of computation of disparity vectors between the current view seen by the bee and the snapshot at the position of the nest. For the computation of the home vector, different methods have been proposed. In [5], parametrized disparity fields are used to map the current view onto the snapshot. The application of Kohonen networks for the computation of disparity information is described in [12].

As pointed out in [15] and [3], the existence of an 'album' of such snapshots at different positions may be sufficient to describe the navigation behavior of insects without the need to introduce concepts like topographic and especially metric maps.

Visual piloting has not been implemented, but preliminary experiments have been made using the Khepera robot (see Fig. 4.3). Real-world tests are planned to be performed in the Sahara desert.

4.5 Summary and conclusions

The chapter presented a new approach to navigation based on a number of design principles for autonomous agents. Instead of concentrating on the development of increasingly sophisticated devices with higher resolution, higher precision and higher computational power, we concentrate on biologically motivated architectures with cheap design and inherent parallelism and hold to the paradigm, that complex behavior emerges from simple processes relying on sensory-motor coordination.

Furthermore, these principles could offer an alternative to the classical ‘information-processing metaphor’. The difficult task of building and maintaining a complex world model and performing a time-consuming planning process could be replaced by behavior that is directly steered by sensory information, thus enabling the robot to operate in a real-world environment.

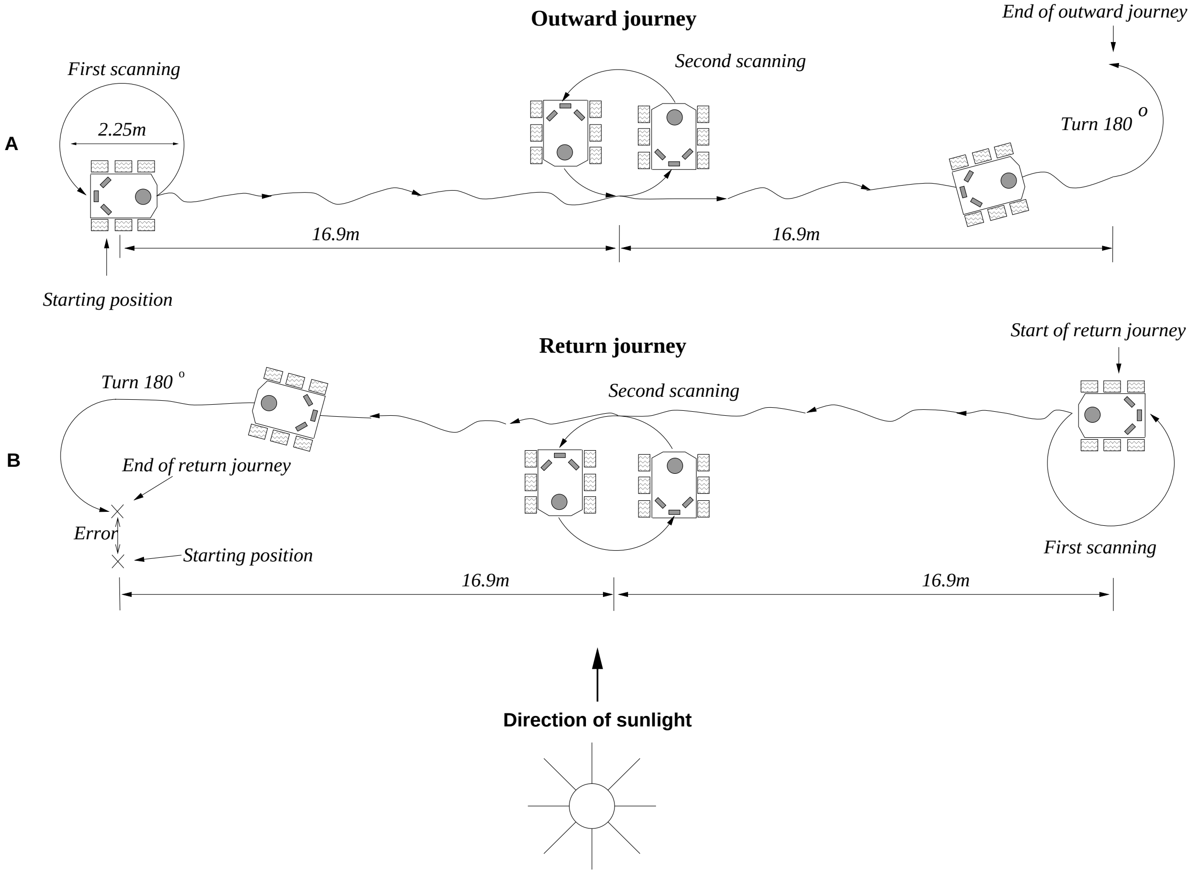


Figure 4.9: *The behavior of the robot during the polarized light compass experiments.* **A:** *Outward journey,* **B:** *Return journey. The robot starts from a predefined position, travels a certain distance and then returns back. At the beginning of the outward journey it performs a 360 degree scan during which it determines the solar azimuth. It then uses this information to calibrate its proprioceptors and set its course direction. After having covered a certain distance (half of outward journey) it performs another scan. At the end of the outward journey the robot turns 180 degrees. On the way back scanning is performed again twice, i.e. at the beginning and in the middle of the return journey. The robot stops when it has covered the same distance as in the outward journey. Taken with authors' permission from [8].*

Bibliography

[1] V. Braitenberg. *Vehicles - Experiments in Synthetic Psychology.* The MIT Press, Cambridge MA, London, 1984.

[2] R.A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.

[3] B.A. Cartwright and T.S. Collett. Landmark maps for honeybees. *Biological Cybernetics*, 57:85–93, 1987.

[4] M.L.S. Collett. Approaching and departing bees learn different cues to distant landmarks. *Journal of Comparative Physiology A*, 175:171–177, 1994.

[5] Matthias O. Franz, Bernhard Schölkopf, and Heinrich H. Bülthoff. Homing by parametrized scene matching. Technical Report 46/1997, Max-Planck Institut für biologische Kybernetik, Tübingen, 1997.

[6] T. Kohonen. *Self Organisation and Associative Memory.* Springer Verlag, Berlin, Heidelberg, New York, 3rd edition, 1989.

[7] Dimitrios Lambrinos. Navigating with an adaptive light compass. In *Proc. of European Conference in Artificial Life*, 1995.

[8] Dimitrios Lambrinos, Marinus Maris, Hiroshi Kobayashi, Thomas Labhart, Rolf Pfeifer, and Rüdiger Wehner. An autonomous agent navigating with a polarized light compass. *will be published in: Adaptive Behavior*, 1997.

[9] Rolf Pfeifer. Building 'fungus eaters': Design principles of autonomous agents. In *Proc. SAB'96*, 1996.

[10] Rolf Pfeifer and Christian Scheier. *An Introduction to New Artificial Intelligence.* will be published by MIT Press, 1997.

[11] H. Ritter, T. Martinetz, and K. Schulten. *Neuronale Netze - eine Einführung in die Neuroinformatik selbstorganisierender Netzwerke.* Addison-Wesley, Bonn, München, 1990.

[12] Thomas Röfer. Controlling a robot with image-based homing. Technical Report Bericht 3/95, ZKW, Universität Bremen, 1995.

[13] M. Toda. *Man, robot and society.* Nijhoff, The Hague, 1982.

[14] R. Wehner. The polarization-vision project: championing organismic biology. In K. Schildberger and N. Elsner, editors, *Neural Basis of Adaptive Behaviour*, pages 103–143. G. Fischer, Stuttgart, 1994.

[15] Rüdiger Wehner, Barbara Michel, and Per Antonsen. Visual navigation in insects: coupling of egocentric and geocentric information. *Journal of Experimental Biology*, 199:129–140, 1996.

Chapter 5

Internal World Representations

by David Sinclair, Wolfram Burgard, Fredrik Bergholm
Panos Trahanias, Lena Gaga and Stelios Orphanoudakis

The state-of-the-art in world representations for visual based navigation may perhaps be illustrated by the following quotation from Iyengar and Elfes [13]: *"Although sensor interpretation and world modeling are fundamental for robots to operate in the real world, robot perception is still one of the weakest components of current robotic systems"*. World representations (or, conversely, the absence of them[1]) are currently a hot topic for debate and this chapter will discuss some issues primarily related to world representations, and to a lesser extent, related to control of behaviors. Chapter 7 will analyze control architecture issues more closely.

An often repeated question in the context of navigation of autonomous robotic platforms is whether or not an explicit world model, an explicit representation, is needed, in the first place. This theme will recur from time to time in the sections below. Internal world representations are also discussed in the context of behavioral and brain sciences in Section 5.4.

5.1 Background: autonomous vehicles

Sensor competence is now sufficiently advanced to give *autonomous vehicles* the ability to derive usable information about their environment.

The main sensor types used for autonomous navigation are:

1. CCD camera(s); stereo, structure from motion, object identification, depth from focus, shape from shading.
2. Structured light; laser range finders, light stripes, IR time of flight.
3. Sonar; nearest reflecting surface, imaging sonar, stereo tracking sonar.

[1] in Brooksian behavior paradigm.

4. Tactile; odometry, collision detecting bumpers.

5. Radar.

It is now standard practice, in research labs, to build *autonomous ground vehicles* (from now on abbreviated AGVs) for internal service with a stereo head, sonar proximity sensors, a mechanical collision detecting bumper and odometry. The stereo head will typically be used to provide some kind of structural information based on epipolar geometry as well as possibly performing landmark, target, object or obstacle identification. If detailed knowledge of the terrain in front of the vehicle is required a laser range is then the sensor of choice.

A major issue with such a (multi-sensor) design is "what to do with the information derived from the sensors". In this context, Section 5.2 reviews a number of existing autonomous vehicle systems with respect to used principles for *combining and using information* in navigation. When combining and using information, some sort of internal representations of the world are used, in a number of existing AGVs; however, some modules may be totally behavior-based. Section 5.3 deals with *various kinds of maps* utilized in AGV navigation. In particular, the so-called grid-based maps are compared to the so-called topological maps. Also, in the context of landmark recognition, there is the question of whether an explicit world model should be used or not. In Section 3.1.2 an example of a landmark recognition scheme without an explicit model has been presented.

5.2 AGV - Behaviour vs centralized representation

The principle debate in recent AGV research focuses on whether to adopt a Brooksian behavior based approach to interaction with the world or build an explicit internal world model and perform deliberative planning. In order to shed light in this issue, we present here a survey[2], giving brief summaries of the world representations used in several different AGV projects; this survey is by no means claimed complete, but rather an effort towards highlighting some of the represention adopted by various groups. Uhlin et al. [47] also give an overview of current autonomous vehicles, but do not specialize on vision-based AGVs.

5.2.1 Definitions

Sensor fusion refers to strategies for combining input from different sensors, whereas *command fusion* is some voting procedure from independent modules, for selecting among a set of possible vehicle actions. Some systems (e.g. CMU's `Navlab`, Subsection 5.2.2) may employ command fusion, but avoid sensor fusion, in some modules.

Topological world model denotes a sparse model where only subsets of the 3D-environment are modeled (represented) and related to each other by various pointers and some attributes.

[2] Internet addresses are available for all the surveyed groups, Oct. 1996.

5.2.2 AGV platforms

In this section we survey briefly a number of systems which use primarily visual information for guidance. The purpose of this survey is to present the different solutions that have been adopted by groups worldwide in their designs, with emphasis on world representation issues.

CMU's Navlab

`Navlab` is a large van, designed to operate as a testbed for autonomous navigation concepts in outdoor environments[3]. As described in [40], it has a range of capabilities for handling different driving conditions, built into a suite of modules, namely:

- RALPH : Rapidly Adapting Lateral Position Handler.
- ALVINN : Neural Network Navigation.
- STORM : High Speed Stereo Vision.
- A Reactive System for Off-Road Autonomous Driving.
- 3-D Vision for Autonomous Navigation
- GANESHA : Grid Based Navigation.
- DAMN : A Distributed Architecture for Mobile Navigation.
- RACCOON : Car following at night.
- Panacea : Active sensor control for neural networks.
- SAPIENT : Situational awareness for driving in traffic.
- MPRF : Massively Parallel Road Following.

The individual modules have specialized capabilities and hence must be activated depending on the driving scenario. In the following, additional details regarding DAMN are presented, since this module is responsible for system control.

The *Distributed Architecture for Mobile Navigation* (DAMN) is a behavior-based architecture similar in spirit to the Subsumption Architecture [6]. In contrast to more traditional centralized AI planners that build a world model and plan an optimal path through it, a behavior-based architecture consists of specialized task-achieving modules that operate independently and are responsible for only a very narrow portion of vehicle control, thus avoiding the need for sensor fusion.

Within the framework of DAMN [40], behaviors provide the task-specific knowledge for controlling the vehicle. Each behavior runs completely independently and asynchronously, providing votes to its appropriate arbiter, each at its own rate and according

[3] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/alv/member/www/navlab_home_page.html

to its own time constraints. The arbiter periodically combines all the latest commands from each behavior and issues a command to the vehicle controller.

In order to allow multiple considerations to affect vehicle actions concurrently, DAMN uses a scheme where each behavior votes for or against each of a set of possible vehicle actions. An arbiter then performs command fusion to select the most appropriate action. The arbiter then computes a weighted sum of the votes it has received from each behavior, and the command choice with the highest value is selected and issued to the vehicle controller.

There is a *mode manager* which is able to select the most appropriate behavior at a given moment, through adjusting weights in the voting scheme, although task specific knowledge about the world is entirely contained within individual behaviors and the set of possible vehicle actions.

DRA Chertsey, DROID

This is a real time autonomous driving system mounted on an electric vehicle[4]. The system has a single camera and with vehicle motion explicitly recovers a series of *scene depths* and then fits a particular surface model to the points. The surface is taken to be a drivable surface. The system can also segment out moving obstacles. The system does not use other characteristics of typical driveable surfaces such as colour, texture or bounding white lines. There is no goal directed behavior, the system only has one behavior, namely that of driving down the middle of the current drivable region. More information can be found in the relevant citations [37, 38].

INRIA PRIMA and ROBOTVIS projects

Within the PRIMA project, a system called SAVA[5] has been developed to perform the integration of visual processing modules and robot control [10, 36, 9, 5, 11]. The core of the system is a rule-base interpreter written in CLISP; this central rule base is responsible for scheduling visual modules and procedures controlling robot action.

ROBOTVIS[6] aims at developing the theory and practice of machine visual perception [48, 2, 14, 49]. Much work has been done within ROBOTVIS on geometric vision. An algorithm for *combining visual, inertial and odometric heading data* was developed and tested. A *limited 3D world model* is used to position the robot relative to a driveable region.

Purdue University RVL Mobile robot navigation

In the past decade, the Robot Vision Laboratory[7] has developed two reasoning and control architectures for vision-guided navigation by indoor mobile robots [31, 27].

[4] http://www.dra.hmg.gb/ImageProcessing/smsmith/dra/group_home.html

[5] http://pandora.imag.fr/Prima/vap.html

[6] http://www.inria.fr/robotvis/

[7] http://rvl1.ecn.purdue.edu/mobile-robot-nav/mobile-robot-nav.html

The first, called FINALE, uses a *3D CAD model of an interior environment* together with vision for localization with respect to the model to perform navigation tasks. The system can cope with multiple moving or stationary obstacles, although obstacles are detected using sonars. Planning may be performed in the space of the 3D building model.

The second architecture for navigation is called FUZZY-NAV. A fuzzy supervisor is used to orchestrate the activation and deactivation of various neural networks. The world model is a *topological one*, containing useful features like *corridor junctions* or *door frames.* Semantic navigational instructions can be used. This permits the system to simultaneously navigate and avoid obstacles.

UBC SPINOZA robot

LCI at UBC[8] is pursuing several autonomous robotics projects including a mobile visually guided platform called SPINOZA. The project is at an early stage and it is unclear what architecture will be used to control the robot (see also [46]).

SRI FLAKEY project

FLAKEY[9] has a *local perceptual space* (LPS) in which information from all sensors and interpretation routines is posted. The system [35, 34] is hierarchical and runs a reactive planning system with a fuzzy controller, behavior sequencer, and deliberative planner. Different behaviors may run simultaneously using information from the LPS. Behaviours are defined as sets of *fuzzy rules.* The degree of applicability of a rule is determined by how well its firing conditions match the information in the LPS (see also the discussion in Section 7.4).

University of Chicago AI Lab

The Animate Agent Project[10] has produced a system to exploit the wide variety of different visual processing routines available from packages like Khoros and Vista (see also [32, 16]). The system features on-the-fly configuration of visual routines to exploit local temporal context. A package called `Gargoyle` is employed that allows a robot to configure, parameterize, and execute image-processing pipelines at run-time. Pipeline configurations and operator parameters can be stored as a library of visual methods appropriate for different sensing tasks and environmental conditions. Beyond this, a robot may reason about the current task and environmental constraints to construct novel visual routines that are too specialized to work under general conditions, but that are well-suited to the immediate environment and task. The system has been implemented on a mobile platform (called Chips) and can perform some real world tasks.

Planning for achieving goals is carried out by a reactive action package, the RAP system. The RAP execution system includes a sensor memory, representation language

[8] http://www.cs.ubc.ca/nest/lci/research

[9] http://www.ai.sri.com/people/flakey/welcome.shtml

[10] http://www.cs.uchicago.edu/ firby/aap/index.html

and interpreter.

Berkeley PATH project

This project is aiming at solving California's transport woes[11] (see also [22]). It's long term goal is to increase the carrying capacity of highways by building smarter cars. The project currently has a car with a roof mounted stereo pair capable of segmenting and tracking moving vehicles. The project is not far enough advanced to have generated a particular world representation.

5.2.3 Local environment representation and dynamic obstacles

In some cases, an *explicit internal representation* of the robot's local environment may be useful. This does not preclude the use of behaviors or neural network based routines for low level navigation and will readily permit the use of contextual information to drive behavior selection.

As an example consider a robot traveling along a corridor only to be confronted by an obstacle directly in its path; to which side of the obstacle should the robot pass? Given dynamic obstacles, such as people or other robots, the future predicted location of obstacles relative to the environment becomes important, and this is facilitated by an explicit local world model. For example, the expected or likely motion of an independently moving object depends, of course, on whether it is a person or not, and whether, for example, a door is nearby. Structure information may be flagged by the actions of dynamic obstacles (e.g. moving people), like the presence of a door or a corridor junction. A similar discussion also appears in [47, p.15]: *"In some situations ... it may be necessary to move the obstacle, and if there is nothing the robot can do by itself, it must call for help. To make such decisions the nature of the obstacle must be investigated ... Will it go away by itself?"*.

It may be desirable to have the local world model extend beyond the immediate camera field of view. This might require fusion of sonar, odometry and visually recovered structural information over an extended period of time. The required spatial and temporal extent of the local model is a matter for experiment.

Knowledge of the relative location of certain types of object relative to the robot will permit the initiation of specific behaviors, door traversing, wall following etc.

5.2.4 Conclusions

A large portion of an autonomous vehicle's navigational needs may be met by a set of simple behaviors; obstacles may be avoided, corridors may be followed and doorways traversed. But in all of the reviewed systems it is apparent that there is some kind of central control. This may be called a *behavior arbiter* or a *rule-base.* This is seen as corroboration of the fact that behaviors have only local applicability and may not be able

[11] http://http.cs.berkeley.edu/projects/vision/vision_group.html

of themselves to determine their domain of applicability. Still, the problem of combining behaviors is an open topic of research.

5.3 Topological and grid-based representations

To efficiently carry out complex missions in indoor environments, autonomous mobile robots must be able to acquire and maintain models of their environments. The task of acquiring models is difficult and far from being solved. The following factors impose practical limitations on a robot's ability to learn and use accurate models:

- *Sensors.* Sensors often are not capable to directly measure the quantity of interest (such as the exact location of obstacles).
- *Perceptual limitations.* The perceptual range of most sensors is limited to a small range close to the robot. To acquire global information, the robot has to actively explore its environment.
- *Sensor noise.* Sensor measurements are typically corrupted by noise, the distribution of which is often unknown (it is rarely Gaussian).
- *Drift/slippage.* Robot motion is inaccurate. Odometric errors accumulate over time.
- *Complexity and dynamics.* Robot environments are complex and dynamic, making it principally impossible to maintain exact models.
- *Real-time requirements.* Time requirements often demand that the internal model must be simple and easily accessible. For example, fine-grain CAD models are often disadvantageous if actions must be generated in real-time.

5.3.1 Definitions

Recent research has produced two fundamental paradigms for modeling indoor robot environments: the *grid-based (metric) paradigm* and the *topological paradigm.* Grid-based approaches, such as *occupancy grids* proposed by Moravec & Elfes [25], who performed work in connection with wide angle sonar, and many others, represent environments by evenly-spaced (2D) grids. Each grid cell may, for example, indicate the presence of an obstacle in the corresponding region of the environment. Topological approaches represent robot environments by *graphs.* Nodes in such graphs correspond to distinct situations, places, or landmarks (such as doorways). They are connected by arcs if there exists a direct path between them.

The occupancy grid, while being built from sensor information, may contain areas labelled *unknown.* In general, maps may be more or less complete, a topic discussed in Chapter 6. Yet another category of maps, not discussed here, are metric maps where sparse metric information is known here and there. This can also be thought of as an incomplete map.

Grid-based approach	Topological approach
+ easy to build, represent, and maintain + recognition of places (based on geometry) is non-ambiguous and view point-independent + facilitates computation of shortest paths – planning inefficient, space-consuming (resolution does not depend on the complexity of the environment) – requires accurate determination of the robot's position – poor interface for most symbolic problem solvers	+ permits efficient planning, low space complexity (resolution depends on the complexity of the environment) + does not require accurate determination of the robot's position + convenient representation for symbolic planners, problem solvers, natural language interfaces – difficult to construct and maintain in larger environments – recognition of places (based on landmarks) often ambiguous, sensitive to the point of view – may yield suboptimal paths

Table 5.1: *Comparison of grid-based and topological approaches to map building.*

5.3.2 Grid-based versus topological paradigm

Both approaches to robot mapping exhibit orthogonal strengths and weaknesses. Occupancy grids are relatively easy to construct and to maintain even in large-scale environments. Since the intrinsic geometry of a grid corresponds directly to the geometry of the environment, the robot's position within its model can be determined by its position and orientation in the real world—which, as shown below, can be determined sufficiently accurately using only sonar sensors, in environments of moderate size. As a desirable consequence, different positions for which sensors measure the same values (i.e. situations that look alike) are naturally disambiguated in grid-based approaches. This is not the case for topological approaches, which may determine the (sometimes qualitative) position of the robot relative to a model based on landmarks or distinct sensory features. For example, if the robot traverses two places that look alike, topological approaches often have difficulty determining if these places are the same or not (particularly if these places have been reached via different paths). Also, since sensory input usually depends strongly on the view-point of the robot, topological approaches may fail to recognize geometrically nearby places.

On the other hand, grid-based approaches suffer from their enormous space and time complexity, and their complexity often prohibits efficient planning and problem solving in large-scale indoor environments. This is because the resolution of a grid must be fine enough to capture every important detail of the world. Topological representations, although difficult to learn in large-scale environments, can be used much more efficiently due to their compactness, which is a key advantage of such representations. Topological maps are usually more compact since their resolution is determined by the complexity of the environment. Consequently, they permit fast planning, facilitate interfacing to symbolic planners and problem-solvers, and provide more natural interfaces for human instructions. Since topological approaches usually do not require the exact determination of the geometric position of the robot, they often recover better from drift and slippage—phenomena that must constantly be monitored and compensated in grid-based approaches. To summarize, both paradigms have orthogonal strengths and weaknesses, which are summarized in Table 5.1.

5.3.3 Conclusions

Since both grid-based (metric) and topological representations offer advantages and suffer from disadvantages, an appropriate representation might be one that combines (integrates) them. Such an approach may construct a grid-based model of the environment by interpreting sensor data, using an artificial neural network to map them into probabilities for occupancy. Multiple interpretations are integrated over time using Bayes' rule[12] [44, 7, 43]. On top of the grid representation, more compact topological maps may be generated by splitting the metric map into coherent regions, separated by the so-called *critical lines* [45]. These critical lines correspond to narrow passages, such as doorways. By partitioning the metric map into a small number of regions, the number of topological entities is several orders of magnitude smaller than the number of cells in the grid representation. Therefore, the integration of both representations has unique advantages that cannot be found for either approach in isolation: the grid-based representation, which is considerably easy to construct and maintain in environments of moderate complexity (e.g. 20 by 30 meters), models the world consistently and disambiguates different positions. The topological representation, which is grounded in the metric representation, facilitates fast planning and problem solving.

5.4 World representations in behavioral and brain sciences

5.4.1 Definitions and concepts

If restricting ourselves to the link between vision and world representations, there is a bulk of literature in brain sciences, psychophysics, ophthalmology and related sciences, discussing this topic. Our focus here is to understand how seeing creatures with their brain and ocular system accumulate knowledge of the external world. To quote Barlow [4, p.21]: *"Vision is sometimes defined as the sense that tells one* **what** *and* **where** *in the space around us; for that one needs to comprehend the relationship between something in the image and some property in the external world, like the position of an object, or its edibility. But there are many relationships to be understood* **within** *an image, and these must be taken into account before objects in the world can be recognized effectively."*

The image on the retina is not directly accessible to the brain, so the first "image" the brain perceives is the one in primary visual cortex, which is not an iconic image, but rather a number of properties of the retinal image reassembled in various ways. Most authors refer to this as *local feature detection* [4], or the *stable features frame* [15, p.266], denoting information that can be visualized as a stack of images with various features such as hue, lightness, saturation, motion, direction of movement, disparity, shape, and texture, each *coded with a few bits.* Words like 'detection' and 'stable frame' are probably used to avoid the word 'image', since the stored structures may not at all be image-like.

It is believed that these features of the external world are subject to a second, third

[12] See also Section 6.5.1.

etc. re-assembly, because information is more easily processed if brought together, focused, in various clever ways. In this context, the notion of *neural image* [4, p.22] appears. In the ophthalmology literature, we come across the term *ocular image*, which is a rather vivid example of how different some of the 'images' in the brain are from what we mean by images in the everyday sense of the word.

5.4.2 Representations

Some of the 'brain images' are a kind of primitive representations of the world, somewhat intermingled with short term memory functionality[13]. An interesting example is the already mentioned *ocular image* [1, pp.539-540]. The term is used to describe the impression formed in higher brain centers through the vision of one eye (at a time). When light strikes a point on the retina, say left eye, a so-called corresponding point in the other retina (right eye) is associated with it by nerve connections, exciting the same place (very local region) in cortex. In the language of ocular images, one may say that corresponding points on the retina are capable of exciting the same 'pixel' on the ocular image. This 'pixel' is believed to store the property of *subjective direction* relative to an egocentric center. The amazing property of the ocular image is that even if one eye is closed, the same pixel is excited *just by one eye*, and the same subjective impression of 3D direction is excited in the brain. Roughly speaking, the ocular image works like an extra *cyclopean eye*.

When discussing world representations for visual-guided robotics, there are representations at various levels. The more primitive representations (like a cyclopean eye) are easily recognized as useful[14]. Even if one camera (eye) is occluded by something, the system may still have a perception of a cyclopean subjective direction to a moving object or a landmark. The ocular image, the brain locus of corresponding points, also serves as a tool for focus of attention. When fixating, all points which are corresponding form a curve out in space (the so-called *horopter based on the subjective direction criterion*) which in some sense are at the same subjective distance from an observer, a moving agent.

Turning to more high-level representations, Feldman [15, p.274] refers to *general world knowledge* and *the environmental frame*[15]. In a sense, the environmental frame is a representation which has directions from the observer as arguments, and is thought to activate certain memorized information coordinated with eye fixation. This has clear links to landmark selection and landmark recognition. It is the 'where' aspect of the world representation, (as seen from the observer, not a prestored map). It should, however, be noted that the *'what'* aspect need not be part of this. You do not need to recognize that the red cylindrical object is a fire extinguisher on a wall, in order to conjure up the memory of a red cylindrical landmark opposite to which there should be vast open space (the memory of a *situation*).

Haber [20, p.296] mentions that mapping the retinal image onto eye positions is not

[13] For example the (neural) hue image, is different if a red pencil from the periphery moves into a central field of view or if it is the other way round. In the first case, the pencil changes color from gray to red, and in the second it remains red.

[14] A learning component is probably involved as well, for setting up the association between egocentric center and perceived subjective direction.

[15] In his representation, *general world knowledge*, *the environmental frame* and *the stable features frame* are three fundamental representations of information.

needed to *stabilize* the image during eye movements. An invariant pattern under translation is an eye movement indication, and is not interpreted by the brain as independent motion. There is no need to stabilize the image on the retina because it is not the retinal image that is used, but rather the stable features frame. In navigation this means that we probably do not need to meticulously compensate for induced camera rotations, and eye rotations can often be detected without oculomotor feedback.

What ways are there to make general object knowledge available in a visual-based navigation system? Richards [33, p.2] mentions the technique of rapid convergence into object category, by repeated (prestored) questions: *"Consider the classical children's game of 20 Questions, where the goal is to identify an object. The first questions usually attempt to identify the general class of object. Is it animal, vegetable or mineral? Subsequent questions attempt to determine the size, shape or mass, or the sounds 'it' might make, how 'it' moves..."*. The sensor system is supposed to yield, gradually, the 'answers' to the questions.

5.4.3 Remarks

In brain sciences, the world representations can roughly be divided into primitive representations such as, for example, the ocular image (=cyclopean eye), independent motion maps[16], etc., and higher-level representations such as the *environmental frame* linked with triggering memories of situations, and object knowledge, and even more general knowledge. However, things like stored maps, even 2D maps, seem rather farfetched, at least those that are not quite local. For narrowing down identification of objects, a series of questions could be practical. For visual-based navigation there is (cf. Section 5.2.3) a clear practical difference whether an object is, say a moving person, another robot, a stationary object, a doorway, a corridor, a landmark, an illumination phenomenon, and so on.

5.5 Aspects on machine learning in vision-based navigation

The navigation of robotic platforms based on visual information is generally considered as a very difficult problem, primarily because of the great variability of the real world scenes and the noise introduced in the process of image acquisition. Systems based on contemporary image analysis and pattern recognition/matching techniques may perform well under certain conditions but face significant problems with others. The processing performed by these systems is finely tuned for very specific environments, but usually fails when applied to different environments.

Machine learning aims to overcome some of the current limitations, by enabling a robot to collect knowledge during its operation, through real-world experimentation and interaction with the environment. A learning robot can adapt or change its behavior according to the specific task requirements and environmental conditions.

The field of machine learning in navigation is relatively young and the applications are few and in most cases simple, at least when compared to what we would like the robots

[16] in the stable features frame

to be able to do. Most of the work in this area has been done either on toy examples, where the answers are simple, or in simulated environments. However, good performance in toy examples does not guarantee that the method is even applicable to real-world situations. Moreover, it is very difficult to provide simulations that are realistic enough. In the case of visual navigation, there are additional difficulties. For example, there are considerable problems in synthesizing video inputs by using computer graphics. Most researchers from the learning community use sonar measurements as an alternative input to their navigation systems, avoiding thus completely to deal with vision-based navigation [8, 50].

However, recently there have been some attempts to extend machine learning methodologies in visual navigation. One, and probably the most promising, application of machine learning in visual navigation is Pomerleau's ALVINN [28, 29, 30], an artificial neural network based perception system which learns to control Carnegie Mellon's `Navlab` vehicle by "watching" a person driving the vehicle. The network receives video images from an onboard camera as the person drives, and is trained to output the vehicle's steering direction. The backpropagation algorithm is used to train the network by altering the weights of the connections between the units so that the appropriate steering response is produced. One of the major problems with using supervised learning from examples for this task is that the system will almost never be presented with examples of what to do when it strays too far from the center of its lane. To compensate for this lack of real training data, the system uses images shifted by various amounts relative to the road's center. The correct steering direction is determined by the amount of shift introduced into the images. ALVINN has been trained to drive in a variety of situations including multiple road types and variations in lighting. The approach taken is to use a pool of pre-trained networks in which each network is trained to work well in a different situation. The same group is recently directed towards the use of genetic algorithms to find the appropriate structure of the network for the task of navigation [3]. The number of layers, the connectivity and the corresponding weights are defined through the genetic search process.

Thrun [41] uses an explanation based neural learning algorithm (EBNN) for a simple indoor navigation task based on trial-and-error. EBNN is a hybrid learning mechanism, which integrates inductive (neural network) and analytical (explanation-based) learning. In this case, the robot is equipped with a color camera but it uses vision primarily to augment sonar information for building occupancy grids. Lately [42], work has been reported on landmark-based navigation by using neural networks as landmark detectors. Each detector maps sensor input to a single value estimating the presence or absence of a particular landmark. Sensor inputs include sonar scans and image encoding. The networks are trained by minimizing the average error in robot localization.

Suzuki and Arimoto [39] propose a self-organizing model for pattern learning and apply this model to position identification and obstacle avoidance tasks. For these tasks the robot learns the model that associates a set of images with its current position or proper motion.

Most research efforts in machine learning use neural networks to deal with the problem of visual navigation and in some cases, like ALVINN, the results are promising. These approaches usually require a long, time consuming training phase and very careful choice of the examples in order to acquire generalized knowledge. This characteristic implies problems in realistic, dynamic environments.

An alternative learning method which seems to be more suitable to robotic applications is *reinforcement learning* [21]. Some researchers have presented work in this field, but they employ mostly proximity sensors such as bumper and sonar [23, 24]. The use of vision is very rare due to its high processing cost.

Nakamura and Asada [26] propose a *vision-based* reinforcement learning method to deal with the problem of reaching a target while avoiding obstacles. In their approach, they incorporate stereo and motion disparity estimation processes that yield information about occlusion status of the target and its neighbor area. Based on this information they build a state space. They model the world as a Markovian process and apply one-step Q-learning to learn the optimal policy. Their method takes a long time to converge and also assumes that neither the target nor the obstacles are moving.

In a different direction, the work by Greiner and Isukapalli [19] tackles the problem of accurate estimation of a robot's position, which is a crucial issue in map-based navigation. They assume a known environment and a set of (manually selected) real-world objects at known locations that are used as landmarks. Based on dead-reckoning the robot has an initial estimate of its position and attempts to locate landmarks and improve this estimation. They acknowledge the fact that some landmarks may not be visible or confused with other landmarks. To address these problems they propose a method which uses previous experiences to learn a selection function that, given the set of landmarks that might be visible, returns the subset that can be used to provide an accurate registration of robot's position.

The above indicate that progress in machine learning in vision-based navigation has been limited todate. However, it is recognized by many researchers that learning should form an indispensable part of autonomous agents, operating in real workspaces. Towards this goal, recent approaches focus on tackling this problem and developing real applications [12, 8, 17, 18, 21].

Bibliography

[1] A. Ames, K.N. Ogle, and G. H. Gliddon. Corresponding Retinal Points, the Horopter and Size and Shape of Ocular Images. *Journal of Optical Soc. America*, 22:538–574, Oct. 1932.

[2] Nicholas Ayache and Olivier D. Faugeras. Maintaining Representations of the Environment of a Mobile Robot. *Robotics and Automation*, 5(6):804–819, dec 1989. also INRIA report 789.

[3] S. Baluja. Evolution of an Artificial Neural Network Based Autonomous Land Vehicle Controller. *IEEE Trans. on Sys., Man and Cybern.*, 26(3):450–463, June 1996.

[4] H. Barlow, C. Blakemore, M. Weston-Smith (Eds) *Images and Understanding.* Cambridge University Press, 1990.

[5] J. Bedrun and J. Crowley. Sava iii: a testbed for integration of and control of real time active vision. In *CIRFFSS*, 1994.

[6] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE J. Robotics Auromat.*, RA-2(7):14–23, Apr. 1986.

[7] Joachim Buhmann, Wolfram Burgard, Armin B. Cremers, Dieter Fox, Thomas Hofmann, Frank Schneider, Jiannis Strikos, and Sebastian Thrun. The mobile robot Rhino. *AI Magazine*, 16(2):31–38, Summer 1995.

[8] J.H. Connell and S. Mahadevan Eds. *Robot Learning.* Kluwer, Reading, MA, 1993.

[9] J. Crowley. Planning and execution control for an autonomous mobile robot. In *IEEE conference on robotics and autonomous systems*, 1994.

[10] J. Crowley. Integration and control of reactive visual processes. *Robotics and autonomous systems*, 15(1), 1995.

[11] J. Crowley and H. Christensen, editors. *Vision as Process.* Springer, 1994.

[12] M. Dorigo, Ed. Special issue on learning autonomous robots. *IEEE Trans. on Sys., Man and Cybern.*, 26(3), June 1996.

[13] S.S. Iyengar, and A. Elfes, editors. *Autonomous Mobile Robots: Perception, Mapping, and Navigation.* IEEE Computer Soc.Press, Los Alamitos, CA, USA, 1991.

[14] Olivier Faugeras, Bernard Hotz, Hervé Mathieu, Thierry Viéville, Zhengyou Zhang, Pascal Fua, Eric Théron, Laurent Moll, Gérard Berry, Jean Vuillemin, Patrice Bertin, and Catherine Proy. Real time correlation based stereo: algorithm implementations and applications. *Int. J. Computer Vision*, 1993.

[15] A. J. Feldman. Four frames suffice: A provisional model of vision and space *The Behavioral and brain sciences*, 8: 265-289, 1985.

[16] J. Firby, P. Prokopowicz, M. Swain, R. Kahn, and D. Franklin. Programming chips for ijcai-95 robot competition. *AI Magazine*, 1996.

[17] J. Franklin, T. Mitchell and S. Thrun, Eds. Special issue on robot learning. *Machine Learning*, 23(2-3), 1996.

[18] P. Gaussier, Ed. Special issue on moving the frontiers between robotics and biology. *Robotics and Autonomous Systems*, 16(2-4), 1995.

[19] R. Greiner and R. Isukapalli. Learning to Select Useful Landmarks. *IEEE Trans. on Sys., Man and Cybern.*, 26(3):437–449, June 1996.

[20] R. N. Haber. Three frames suffice: Drop the retinotopic frame. *The Behavioral and brain sciences*, 8: 295-296, 1985.

[21] B. Kroese, Ed. Special issue on reinforcement learning and robotics. *Robotics and Autonomous Systems*, 15(4), 1995.

[22] Q. Luong, D. Weber, D. Koller, and J. Malik. An integrated stereo-vision approach to automatic vehicle guidance. In *ICCV 95*, 1995.

[23] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Trans. on Robotics and Autom.*, 8(3):304–312, Jun. 1992.

[24] S. Mahadevan. Enhancing Transfer in Reinforcement Learning by Building Stochastic Models of Robot Actions. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 290–299, Aberdeen, Scotland, July 1992.

[25] H. Moravec and A. Elfes. High-resolution Maps from Wide Angle Sonar. *IEEE International Conference on Robotics and Automation*, pp.116–121, St. Louis, MO, 1985.

[26] T. Nakamura and M. Asada. Stereo sketch: Stereo Vision-Based Target Reaching Behavior Acquisition with Occlusion Detection and Avoidance. In *IEEE Int. Conf. on Robotics and Automation*, pages 1314–1319, April 1996.

[27] D. Pan, J. Pack, A. Kosaka, and A. Kak. Fuzzy-nav: a vision-based robot navigation architecture using fuzzy inference for uncertainty-reasoning. In *Proceedings of the world congress on neural networks*, volume 2, pages 602–607, 1995.

[28] D.A. Pomerleau. Neural network based autonomous navigation. In C. Thorpe, editor, *Vision and Navigation: The CMU Navlab.* Kluwer Academic Publishers, 1990.

[29] D.A. Pomerleau, J. Cowdy, and C.E. Thorpe. Combining artificial neural networks and symbolic processing for autonomous robot guidance. *Engineering Applications of Artificial Intelligence*, 4(4):279–285, 1991.

[30] D.A. Pomerleau. *Neural network perception for Mobile Robot Guidance.* Norwell, MA: Kluwer, 1993.

[31] J. Pan and A. Kak. Design of a large-scale expert system using fuzzy logic for uncertainty-reasoning. In *Proceedings of the world congress on neural networks*, volume 2, pages 703–708, 1995.

[32] P. Prokopowizc, M. Swain, J. Firby, and R. Kahn. Gargoyle: an environment for real-time context-sensitive active vision. In *National Conference on Artificial Intelligence AAAI-96*, 1996.

[33] W. Richards, "How to play twenty questions with Nature and win.", Tech. Report, A.I. Memo No. 660, Mass. Inst. of Technology, AI Lab, Massachussets, USA, December 1982.

[34] A. Saffiotti, K. Konolige, and E. Ruspini. A fuzzy controller for flakey, an autonomous mobile robot. Technical Report 529, SRI Artificial Intelligence Center, 1993.

[35] A. Saffiotti and L. Welsley. Perception-based self-localization using fuzzy locations. In *Reasoning with uncertainty in robotics.* Springer LNCS, 1996.

[36] B. Scheile and J. Crowley. Certainty grids: perception and localization for a mobile robot. *Robotics and autonomous systems*, 12(3):163–172, 1994.

[37] S. Smith. Integrated real-time motion segmentation and 3d interpretation. In *ICPR 13.* IEEE computer society press, 1996.

[38] M. Stephens, R. Blissett, D. Charnley, E. Sparks, and J. Pike. Outdoor vehicle navigation using passive 3d. In *CVPR89*, pages 556–562, 1989.

[39] H. Suzuki and S. Arimoto. Visual Control of Autonomous Mobile Robot Based on Self-Organizing Model for Pattern Learning. *Journal of Robotic Systems*, 5(5):453–470, 1988.

[40] C. Thorpe, editor. *Vision and navigation - The Carnegie Mellon Navlab.* Kluwer Academic Publishing, 1995.

[41] Sebastian Thrun. An Approach to Learning Mobile Robot Navigation. *Robotics and Autonomous Systems*, 1995.

[42] Sebastian Thrun. A Bayesian Approach to Landmark Discovery and Active Perception in Mobile Robot Navigation. Technical Report CMU-CS-96-122, School of Computer Sc., Carnegie Mellon Univ., 1996.

[43] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schimdt. Map learning and high-speed navigation in RHINO. In D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors, *AI-based Mobile Robots: Case studies of successful robot systems.* MIT Press, Cambridge, MA, to appear.

[44] Sebastian Thrun. Exploration and model building in mobile robot domains. In *Proceedings of the ICNN-93*, pages 175–180, San Francisco, CA, March 1993. IEEE Neural Network Council.

[45] Sebastian Thrun and Arno Bücken. Integrating grid-based and topological maps for mobile robot navigation. In *Proc. of the 14th National Conf. on Artificial Intelligence*, 1996.

[46] V. Tucakov, M. Sahato, D. Murray, A. Mackworth, J. Little, S. Kingdom, C. Jennings, and R. Barman. Spinoza: a stereoscopic visually guided mobile robot. In *HICSS97*, 1997.

[47] T. Uhlin, K. Johansson, Autonomous Mobile Systems, a study of current research NUTEK, Swedish National Board for Industrial and Tech. Development, report R 1996:4.

[48] Thierry Viéville, François Romann, Bernard Hotz, Hervé Mathieu, Michel Buffa, Luc Robert, P.E.D.S. Facao, Olivier Faugeras, and J.T. Audren. Autonomous navigation of a mobile robot using inertial and visual cues. In M. Kikode, T. Sato, and K. Tatsuno, editors, *Intelligent Robots and Systems*, Yokohama, 1993.

[49] Zhengyou Zhang and Olivier Faugeras. A 3-D World Model Builder with a Mobile Robot. *International Journal of Robotics Research*, 11(4):269–285, Aug 1992.

[50] B. Yamauchi and R. Beer. Spatial learning for navigation in dynamic environments. *IEEE Trans. on Sys., Man and Cybern.*, 26(3):496–505, June 1996.

Chapter 6

On Path Planning

by Erich Rome, Joachim Hertzberg and Frank Schönherr

Mobile robots are complex, integrated systems of hardware and software, which are designed to perform a number of tasks. On a strategic level, such a task could be described by specifying a goal, for example, "Deliver a parcel to office X on floor Y". The task to achieve this *strategic goal* can be broken down into a number of subtasks, like navigating in hallways while avoiding collisions, evaluate sensory data to localize orientation points (*landmarks*), doors, and elevators, update the position and correct the trajectory, which in turn can be broken down into other, more specific subtasks.

Navigation, for instance, could be subdivided into three phases. Let us assume that the environment is completely known, the robot has an abstract description of the environment (a *map*), knows its current position in the environment, and the position of its goal location. In order to reach the goal location, the robot could search the map for a path to the goal that avoids obstacles and that is wide enough to let the robot fit through. This first phase is called "path planning". Its output could be a set of action descriptions (the *plan*) that have to be executed by the robot in a second phase, called "plan execution". Since errors may occur during the execution of the plan, it is desirable to detect and correct these errors. This is done in a phase called "action control".

In this chapter, we will review the state of the art in path planning techniques for the task of navigation of mobile robots. Plan execution and action control are covered in chapter 7.

6.1 Definitions

The partition of the general navigation task into three phases is taken from Crowley [13, p. 708f], who describes these phases more precisely:

- *Path Planning.* A path-planning system develops a plan to achieve a goal using an abstract model of the domain. In most cases the elements of this model have been prelearned (or constructed by hand) and cannot be directly sensed at the time of planning.

- *Plan Execution.* A plan-execution system uses a dynamically maintained model of the local environment to monitor the execution of the plan and to execute actions. The execution system must verify that each step can be executed and watch for opportunities to simplify the plan.
- *Action Control.* Low-level control of actions often involves techniques from control theory. In sophisticated systems such controllers are based on the dynamics of the device and on the dynamically maintained model of the structure of the environment.

All phases of the navigation task are equally important for proper performance. According to Latombe [30, p. 4f], the planning and execution phases may be performed consecutively if the environment is completely known. The two phases need to be interwoven if complete knowledge is not available. If during the execution of a path plan the action control detects that the goal cannot be reached—due to an unexpected obstacle, for example—then the plan has to be modified. It cannot be excluded that replanning the path also affects a more strategic task, rendering the achievement of a strategic goal impossible. This example illustrates that robot tasks sometimes are not independent of each other and that they may interact during execution. Therefore, it is sometimes hard to tell where one task ends and the other begins.

Although the three phases are equally important, the literature usually emphasizes the first phase, path planning. This is a special case of "motion planning", a problem which Latombe [30, p. ix] informally describes by the question: "How can a robot decide what motions to perform in order to achieve goal arrangements of physical objects?" The special case of path planning could analogously be described informally as: "How can a robot decide what motions to perform in order to travel from a given start location to a specified goal location?" The location specification may be in form of global world coordinates or in form of descriptions of arrangements of objects in the environment.

A fact that makes it hard to analyze the state of the art in path planning is that there is neither a commonly accepted general method for robot navigation in general nor for path planning in particular. Instead, a large variety of methods have been published that are more or less special solutions for specific strategic tasks, specific machines, and specific types of environments. A notable fraction of this work is either theoretic or has been tested only in simulations. This means that additional work has to be done in order to test whether these methods will work in real robots.

According to Latombe [30, p. 12], all these methods can be related to three different classes of approaches to the path planning problem:

1. "Roadmap Methods",
2. "Cell Decomposition Methods", and
3. "Artificial Potential Field Approaches".

There is another class of navigation and path planning methods that has a nonempty intersection with the three classes above, called "Navigation under Uncertainty". This class of methods takes into consideration different types of uncertainties: Uncertainty of sensory data, uncertainty of kinematics, and uncertainty of map information.

In this chapter, we will characterize the four approaches and point out their strengths and weaknesses. For readers who are interested in an extensive overview and thorough description and analysis of path planning problems, citation [30] may be consulted. It should be noted that many of the *behavior-based* approaches to robot control programs are pure reactive approaches that operate without explicit path planning (cf. [20, 19]). A very interesting variation of this approach, the *biologically inspired approach to visual navigation*, is described in chapter 4.

Since the focus of this report is on vision-based navigation, a reader might ask the following questions: How does a particular sensor configuration affect the path planning process? Does the use of a camera sensor require different planning methods than the use of an ultrasound sensor? The answers depend on the kind of navigation task. If the robot operates in a completely known environment and already has a map of it, then it does not matter whether the map is given by a programmer, or pre-learned and extracted from the data of a particular sensor configuration.

However, the robot's sensor configuration may have influence in two cases. First, it may influence the other navigation phases, plan execution and action control. The actions required to perform a localization may differ with respect to the sensor configuration at hand. If a camera sensor is used, then the robot may have to perform certain actions to get into a position that yields the proper camera view on a particular landmark, i.e., the view that is needed to identify the landmark. For ultrasound sensors, a landmark "looks" different, and so the strategy to approach the landmark in order to identify it may be quite different from the strategy used for a camera.

Second, if the environment is not completely known, then a robot may be programmed to explore it. The exploration strategy may also differ with respect to the actual sensor configuration.

In the descriptions of the path planning methods in this chapter, we abstract from particular sensor configurations. If sensors are mentioned, then they are taken as an example or they refer to the original work that we review. Popular methods for the abstract representation of an environment, like maps, are reviewed in chapter 5 of this report. Important building blocks of maps are landmarks. Chapter 4 describes methods for the detection and recognition of visual landmarks.

6.2 Roadmap methods

Roadmap methods can be characterized as path planning methods based on graph search. This class of methods usually assumes a completely known and static environment, which is represented by a two-dimensional map. A graph of possible paths between a given start location and a goal location is constructed and the actual planning is performed as graph search. The latter part is done using well-investigated graph search methods like the A^* algorithm or the Dijkstra algorithm and similar ones. The constructed path has to satisfy one or more *optimality criteria*. These include:

- Minimal length,
- collision avoidance,

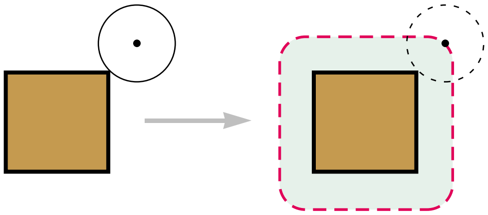


Figure 6.1: *Construction of an "inflated" obstacle in Lozano-Pérez's configuration space method. The robot is shrunk to a point, the circle is the smallest circle that encloses the robots dimensions in an orthogonal projection onto the map. The obstacle is inflated by this circle's radius. This can be imagined as rolling the circle once around the obstacle's perimeter. The traced course of the circle's center point is the inflated obstacles boundary.*

- minimal travel time,
- minimal time to compute the path,
- maximum security, and
- minimum total rotation of the robot.

The more difficult part is the construction of the graph, and a number of elaborated methods have been developed to achieve this. In this section, we will characterize four representative methods.

6.2.1 Visibility graph method

Jorgensen *et al.* [21] presented a simple method that demonstrates how path planning can be done in a known, static environment. They employ the *visibility principle* and assume that a static, completely known environment is given. It is represented by a two-dimensional map that records the coordinates of obstacle positions and of start and goal locations. In order to reduce computational effort, they abstract from the robot's physical dimensions and reduce it to a point. To ensure that the robot does not get stuck in gaps between the obstacles that are too narrow to pass, the obstacles are inflated by a distance that is half the largest diameter of the robot plus a tolerance margin.

The idea of shrinking the robot to a point and inflating obstacles by the robot's largest half-diameter to reduce computational effort has first been presented in Lozano-Pérez's [35] *configuration space* approach. Figure 6.1 illustrates how the representation of the obstacles is constructed in this approach, which became very popular in the 1980s.

Figure 6.2 further illustrates the above concept. Figure 6.2.1 shows an example environment with two obstacles. In Fig. 6.2.2 the inflated obstacles and walls are indicated by dashed lines. The construction of the search graph in the visibility method is done as follows:

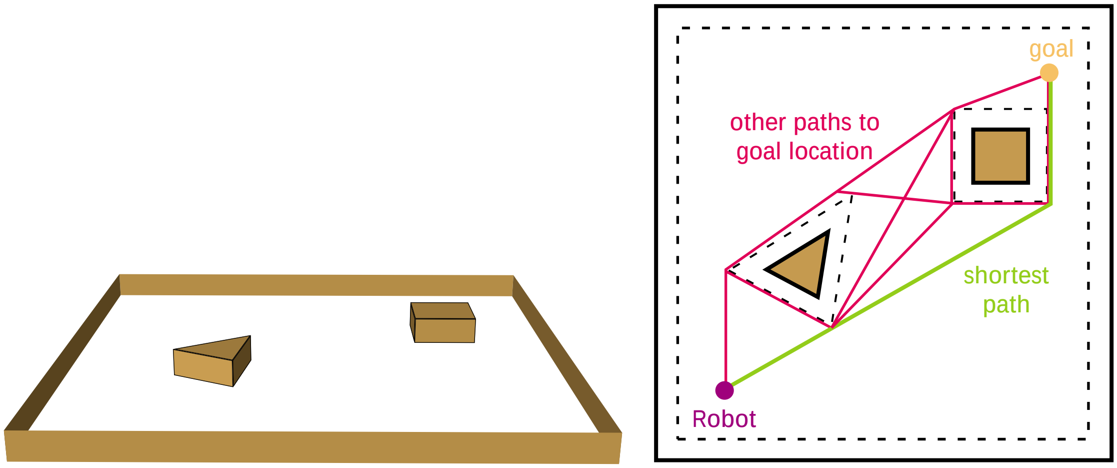
1. The robot's start position becomes the root node of the graph.

2. Starting from the robot's initial position, lines are drawn to the *visible*—i.e. non-occluded—inflated corners of all obstacles. A corner B is visible from corner A if the line connecting both corners does not intersect with an obstacle. The lines obtained with this method are represented as edges in the graph, their end points are new nodes in the graph. The edges are labelled, e.g., with the length of the lines they represent.

3. Step 2 is repeated for all new nodes.

4. The algorithm stops if the goal location is reached or no more corners are visible.

If the goal location is not among the found nodes, then there is no path from start to goal location. Otherwise, the graph is searched by an A^*-like algorithm that is able to find a shortest path.



6.2.1 *An artificial navigation environment with two obstacles.*

6.2.2 *2D map with all found paths and shortest path between start and goal locations.*

Figure 6.2: *Path planning using visibility graph principle.*

6.2.2 Methods using free space representations

Usually, an environment may be partitioned into obstacles and free space, and this is reflected by the representations that are used in the respective literature. Since both representations are complementary, i.e. one may be obtained from the other, we will describe only one of them, the free space representations.

Convex Regions (Crowley)

Crowley [12] constructs a free space representation in the following way. The free space is partitioned into *convex regions*, i.e. regions with a convex polygonal shape. These regions are then shrunk by half the robot's maximum diameter (complementary to inflating the obstacles). These regions may be disconnected. In a second step, disconnected regions are connected through *doorways*.

The graph structure is then constructed as follows:

1. For each convex region, and for each doorway that leads to that region,

   a) generate a node for the location inside the convex region that lies close to the middle of the part of the border that is shared between the region and the doorway, and

   b) generate all possible edges between all the nodes of this region.

2. For each doorway, generate an edge between those two nodes that are "assigned" to this doorway in those two regions that are connected via this doorway.

Figure 6.3 shows how such a representation could look like, with the search graph superimposed. After the construction of the graph is finished, path planning is performed with the Dijkstra graph search algorithm.

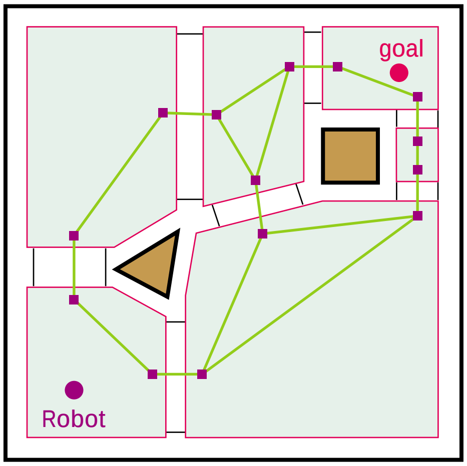


Figure 6.3: *Representation of free space by convex regions.*

Inherent advantages of the approach include:

- Paths inside a convex region are free of collisions.
- Each location inside a convex region can be reached from any other location inside the region by travelling along a straight line.

In the approach described in [12], the map representation is constructed from the robot's sensor data during an initial phase in which the robot systematically explores the environment. However, if the exploration fails to cover the entire environment, then the resulting map is incomplete. Also, it may be imprecise due to sensing errors.

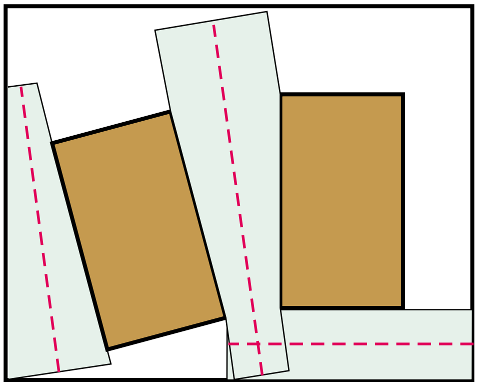


Figure 6.4: *Representation of free space by generalized cylinders.*

Generalized Cylinders (Brooks)

Brooks [4] uses *generalized cylinders* to represent the free space. The map of the static environment is given and the free space between obstacles is filled with generalized cylinder shapes[1].

As in other methods, the basic representation of the environment is a two-dimensional map. Generalized cylinders are constructed between each pair of neighbouring obstacles. The spine is a curve in the middle between the two obstacles. The cylinder is extended beyond the obstacle edges at both ends until its corners "touch" other obstacles. Figure 6.4 shows an example of the generalized cylinders representation of free space.

In a first step all cylinders which are too narrow for the robot to pass through are eliminated. Thereafter, all spine crossings are determined and become nodes in a graph structure. The spine sections that connect two spine crossings are marked as edges between the corresponding nodes. Path planning is then performed as graph search.

The computational complexity of Brooks' method is quite high, $O(n^3)$, which is a clear disadvantage. If the environment is cluttered with obstacles, then it is possible that not enough spine crossings can be constructed, i.e. the method is not able to find all possible paths.

A "nice to have" feature of the generalized cylinder method is the fact that the cones' spines are always in the middle between two obstacles. A robot that travels along a spine will always keep the maximum possible distance from the obstacles.

To conclude the subsection on free space representations, it should be mentioned that other free space representation methods exploit *Voronoi diagrams* or their dual, the *Delaunay triangulation*. Several different algorithms have been proposed to compute such representations. The implementation of these algorithms is not straightforward, may be subject to rounding errors in floating point arithmetic, and is of varying computational complexity, which ranges from a "normal" $O(n^2)$ to as good as $O(n)$, where n is the number of input vertices.

[1] A generalized cylinder is defined by a curve, called the *spine*, and a *sweep function*, which describes the generalized cylinder's volume by sweeping a—possibly varying—cross section along the spine.

6.2.3 Remarks

In general, the Roadmap approach has an easy part and a hard part. The hard part is the construction of the representation of the environment, and the easy part is the actual path planning that exploits this representation and is implemented using graph search algorithms.

Drawbacks of "Roadmap Methods"

Besides the drawbacks that are specific to particular techniques, general drawbacks of "Roadmap Methods" for path planning include:

- The construction of the free space representation may become very time-consuming, for instance, when Voronoi diagrams or Generalized Cones are used.
- Most of the respective literature says little or nothing about plan execution and action control.
- Most of the presented methods exclude some possible narrow passages between obstacles, making themselves less suited for environments that are cluttered with obstacles.
- Some of the methods account very poorly for the kinematics of a robot.

Advantages of "Roadmap Methods"

Specific advantages of particular Roadmap Methods have already been mentioned in the respective sections. As a general conclusion, one may state the following advantages:

- Most of the methods presented in this section are well-suited for known and static environments which are not too cluttered with obstacles.
- Some of the methods are easy and fast to compute.

6.3 Cell decomposition methods for path planning

Cell decomposition methods fall into two classes: *Exact* and *approximate*. The methods of the first class partition the entire free space of a known navigation environment into contiguous, non-overlapping cells. The methods of the latter class construct an area of contiguous, non-overlapping cells of a given size or of different sizes down to a maximum resolution. This area does not necessarily cover the entire free space of the environment under consideration, i.e., it may be only an approximation of the free space. The planning algorithms for exact cell decomposition methods use—as in other approaches—graph search to find a path between start and goal location.

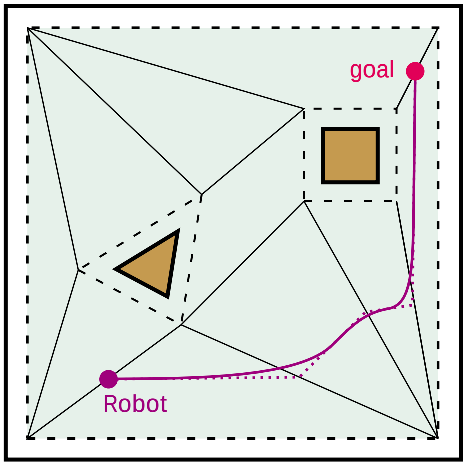


Figure 6.5: *Exact cell decomposition of free space using polygonal cells.*

6.3.1 Exact cell decomposition methods

An exact cell decomposition method is composed of three phases. The first phase is the decomposition of free space into contiguous, non-overlapping cells of varying shape. One method for doing this is the partitioning of free space into convex polygonal cells, as shown in Fig. 6.5.

The second phase is the construction of a path between start and goal location. This is done in three steps. The first step consists of the construction of a *connectivity graph* with a node for each cell. The graph's edges represent the adjacency relations between cells. In a second step, a graph path is searched that connects the cell that contains the start location with the cell that contains the goal location.

After a graph path has been found, the final navigation path is constructed in a third phase. This is done in the following way. First, a line is drawn between the start location and the midpoint of the line that is the common border between the start location's cell and the next adjacent cell in the graph path. Second, for each pair A/B and B/C of adjecent cells in the graph path, a line is drawn from the midpoint of the common border between A/B to the midpoint of the common border between B/C. And finally, a line is drawn between the end point of the last line segment and the goal location. Figures 6.5 and 6.6 show the resulting path for our sample environment. In a post-processing phase, these polygonal paths may be smoothed with suitable techniques, e.g. *spline curve fitting*. The splines are in many cases shorter then the original polygonal paths. Figure 6.5 shows the polygonal path as dotted line and the superimposed fitted spline as solid line.

This sounds rather simple, but an implementation has to be designed rather carefully. In general, according to Latombe [30, p. 200f], an exact cell decomposition method should have the following basic properties in order to make it useful for path planning and to make it efficiently computable: The cells should have a simple geometry, the adjacency test should be simple to implement, and it should be easy to find a path crossing cell boundaries.

In order to reduce search time and to find optimal or near-optimal paths, it seems desirable to have the minimum possible number of cells in a partition of free space. A

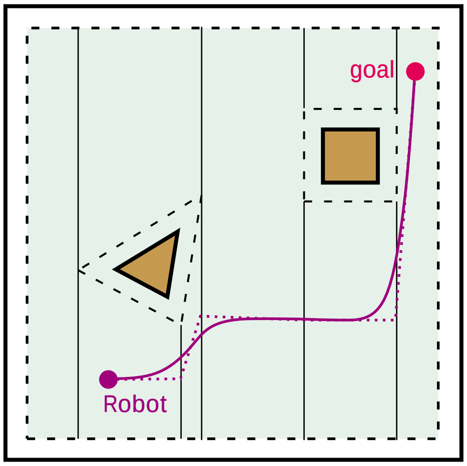


Figure 6.6: *Exact cell decomposition of free space using trapezoidal cells.*

decomposition of a polygon into the minimum possible number of convex polygons is called an *optimal decomposition*. Usually, the free space is bounded externally by a polygon and internally by the polygonal shapes of the obstacles, which can be viewed as holes inside the external bounding polygon. Unfortunately, it has been shown that the optimal decomposition of such a polygon is NP-hard [33].

Latombe [30, p. 116] states that there are non-optimal partitions which are efficiently computable, such as the trapezoidal decomposition shown in Fig. 6.6. Such a decomposition can be generated by a line sweeping algorithm with time complexity of $O(n \ \log \ n)$, where n is the number of vertices of the "polygon with holes" that represents the free space.

Latombe [30, p. 200f] requires an exact cell decomposition method to have the following general, basic properties in order to be useful for path planning and also efficiently computable: The cells should have a simple geometry, the adjacency test should be simple to implement, and it should be easy to find a path crossing cell boundaries.

In such a non-optimal decomposition, it is rather hard to construct optimal or near-optimal paths. For a trapezoidal decomposition, Latombe [30, p. 206] suggests a modification of the connectivity graph construction and a modified A^* search algorithm.

Extensions of the basic exact cell decomposition methods have been formulated to account for the kinematics of a robot, e.g. [46, 47]. The computational complexity of some of these extensions may limit their use for practical applications. Parts of the algorithm proposed in [46] have a time complexity of $O(n^5)$, where n is the number of all edges and vertices of polygonal obstacles in an environment [30, p. 224].

6.3.2 Approximate cell decomposition methods

This approach has been originated by Lozano-Pérez and Brooks [34, 5]. Approximate cell decomposition can also be divided into three phases, like their exact counterparts. The second phase, construction and search of the connectivity graph, is the same in both approaches. The major difference is the first phase, the decomposition method. Here, all cells have a pre-specified shape, e.g. a rectangular one. In general, it is impossible to find a tesselation of the free space with non-overlapping, adjacent cells of a given shape that covers the free space exactly. The free space can only be approximated with such cells.

In order to provide solutions for the path planning problem, the resolution of the decomposition, i.e. the cell size has to be chosen carefully. If the resolution is too coarse, then some possible paths cannot be found. If the resolution is too fine, then the computation time may be too long for practical applications.

A typical approximate cell decomposition method is the *quadtree* decomposition illustrated in Fig. 6.7. It approximates the free space in a hierarchical fashion. In a first step, the entire space is divided into four rectangular cells. Cells which are entirely in free space are labelled "empty" and marked light gray, cells which are entirely inside an obstacle boundary are labelled "full" and marked dark gray, and cells which cover both free and obstacle space are labelled "mixed" and marked medium gray. Only the latter ones are considered in the second iteration. Each of these cells is again divided into four rectangles and marked in the same fashion as in the first step. The iteration stops, if the cell size reaches a pre-specified minimum size. The cell description can be represented using a tree structure with degree 4—hence the name quadtree—i.e. each node has either four successors or is a leaf node. Each iteration step generates a new level in the tree.

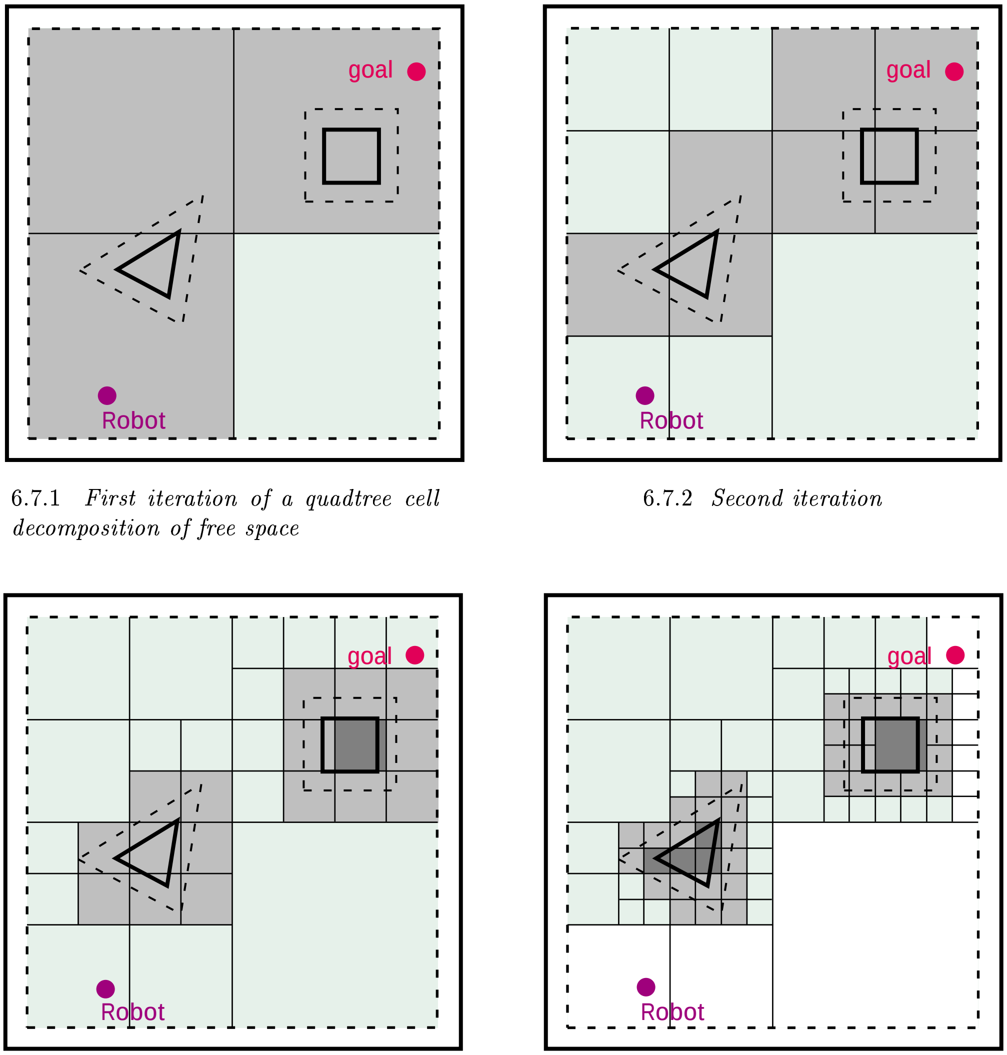
Next, the connectivity graph is constructed and searched for a graph path in a fashion similar to that described in subsection 6.3.1. The search algorithm returns a sequence of "empty" cells—called an *e-channel*—that connects the cells which contain the start and the goal location, rather than a polygonal or spline-like trajectory that other path planning methods produce. The e-channel may be seen as an area that constraints the robot's possible motions and provides some freedom for local decisions of a navigation algorithm.

Table 6.1 shows the searched part of the quadtree for the example environment of Fig. 6.7.4. The leaves of the tree constitute an e-channel from start to goal. Cells are numbered clockwise from 1 to 4, starting with the upper left cell of a decomposition. The sequence of cells in the e-channel is: 4.4, 4.3, 3, 2.3.2.3, 2.3.2.2, 2.3.3.3, 2.3.3.2, 2.2.3.3, 2.2.3.2, 2.2.2.

root									
4		**3**	2						
4.4	**4.3**		2.3				2.2		
			2.3.2		2.3.3		2.2.3		**2.2.2**
			2.3.2.3	**2.3.2.2**	**2.3.3.3**	**2.3.3.2**	**2.2.3.3**	**2.2.3.2**	

Table 6.1: *Part of a quadtree with leaves forming an e-channel.*

Decomposition and path search may be interwoven in order to reduce the number of decompositions. This leads to hierarchical graph search and may require more sophisticated search algorithms—for example, introducing a memory of the search results of higher levels—in order to increase search efficiency.



6.7.1 *First iteration of a quadtree cell decomposition of free space*

6.7.2 *Second iteration*

6.7.3 *Third iteration*

6.7.4 *Fourth iteration and resulting channel between the cells containing the start and the goal location*

Figure 6.7: *Path planning using approximate cell decomposition. Cells that lie entirely inside an obstacle are marked dark gray, cells that are entirely in free space are marked in light gray and cells that cover free space as well as obstacle area are marked medium gray. The white cells in the right hand subfigure constitute a* channel *between the cells that contain the start and the goal location.*

6.3.3 Conclusions

For the remainder of this subsection, the acronyms ECDM and ACDM are used for exact, respectively approximate, cell decomposition methods.

Drawbacks of "Cell Decomposition" methods

- One drawback of ECDMs is the high computational effort, especially of those variants that take the kinematics of a robot into account.
- ECDMs are difficult to implement because they require exact numerical computation algorithms and appropriate search methods.
- ACDMs are not guaranteed to find a free path if one exists.
- For the case of three-dimensional environments, ACDMs may have a larger time complexity than other approaches.

Advantages of "Cell Decomposition" methods

- ECDMs always find a path between start and goal location, if there is one. For this class of methods, the usage of exact numerical computation methods and suitable search algorithms is crucial.
- ACDMs are in general easier to implement than exact decomposition methods because they do not require exact numerical computation methods.
- For two-dimensional environments, ACDMs have a performance that does not fall far behind that of other approaches.

6.4 Potential field approaches to path planning

Khatib [22] and Krogh [28] introduced *Artificial Potential Fields* for mobile robot navigation and for planning the motions of manipulators. In this section, only navigation applications are considered.

Natural potential fields, like the gravitational (Newtonian) potential or the electrostatic potential, exert attractive or repulsive forces on bodies. This basic principle is adopted in the Artificial Potential Field approach. The representation of a goal location is surrounded by an attractive artificial potential field, and all obstacles are surrounded by repulsive artificial potential fields. The robot is modelled as a "particle" and its course between the obstacles and towards the goal is determined by the artificial forces which result from the combined artificial potential fields.

In general, Artificial Potential Field approaches for mobile robot navigation tasks work as follows. The basic representation of the robot's local or global environment is a two-dimensional map. The coordinates of the goal position and the positions of the grown obstacles are computed with respect to a fine-grained grid. Subsequently, the

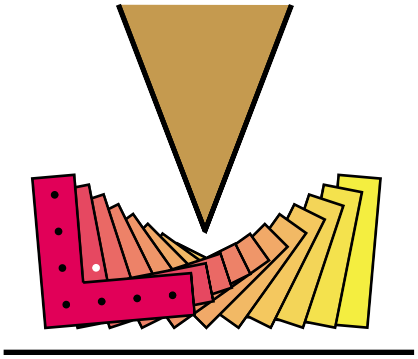


Figure 6.8: *Translation and rotation of an L-shaped form through a narrow gap using Chuang's and Ahuja's extension of the Artificial Potential Field approach. The center of translation is depicted as a white spot near the inner angle of the leftmost L-shape.*

goal representation is equipped with an attractive artificial force field and the obstacle representations are equipped with strong repulsive force fields. The field distribution across 2D space is described by an *artificial potential function (APF)*. In contrast to natural potential functions, the APFs of obstacles do not spread to infinity. To reduce computational complexity, they usually have a limited range, called *potential width*.

For each grid point, the artificial force resulting of the combined values of the goal's APF and the obstacles' APFs at the respective point is computed. The result is stored as a vector in this location. The vector's negated direction indicates into which direction the robot could be "pushed" at the corresponding point. The path planning process then consists of tracking the gradients between the robot's current location and the goal location. Many APFs give rise to local minima in the combined field forces of cluttered environments. The resulting vector is zero at these points, and the robot may get "trapped" in these minima before it reaches the goal location.

In this section, we review basic work on and extensions of APFs. More work may be found in [23, 3, 48, 36].

6.4.1 Extensions of the approach

Accounting for kinematics

In order to be able to follow all possible changes of direction, some simpler implementations of Artificial Potential Fields require a *holonomic* robot, i.e. a robot that is able to perform instantaneous heading changes without additional maneuvering.

Chuang and Ahuja [10] showed that it is possible to incorporate the handling of kinematic constraints into the Artificial Potential Field approach, thus making it suitable for non-holonomic robots as well. Figure 6.8 shows an example from [10]. The L-shaped form cannot be maneuvered through the gap between the wall (represented by the horizontal line) and the triangular obstacle by a simple translation. Instead, it must be rotated slowly during the translation.

In order to achieve this, Chuang and Ahuja use a local optimization method. Each of several points on the axes of the L-shaped form is equipped with a repulsive potential, as well as the wall and the obstacle. Additionally, a center of rotation is defined with respect to the L-shaped form. After a short translation of the L-shaped form towards the gap, the total potential of the new configuration (form, wall, and obstacle) is computed. The L-shaped form is rotated with respect to the center of rotation in such a way, that the total potential of the new configuration is minimal. The resulting movement appears to be smooth.

Of course, one could argue against this extension that optimizations like this one require a considerable computational effort. But the method has to be applied only in those cases, where the gap between two obstacles is too narrow to proceed with a simple translation.

Diffusion in potential fields

Another extension to the Artificial Potential Field Approach is Dautenhahn's approach called "Diffusion in Potential Fields" (DIP) [14]. This approach combines Potential Fields and a diffusion spreading method.

In nature, the process that compensates different concentration levels is called diffusion. The first part of the DIP algorithm models this process. In a first step, the goal location gets a high "concentration" (numeric value) and the free space a zero concentration. The concentration levels are iteratively computed by a simple diffusion rule. Spreading from the goal location, more and more free space locations get a non-zero value. If the start location gets a non-zero value, then the diffusion step ends. In the resulting configuration, the concentration values have a gradient that leads "uphill" from the start location towards the goal location.

In a second step, APFs are computed for the environmental arrangement and their values are combined with the concentration values by another simple rule. The combined algorithm results in smoother paths that do not end in local minima. However, it is not guaranteed that this method will always find an optimal path.

6.4.2 Conclusions

Drawbacks of "Potential Field" methods

One may imagine the obstacles as (potential) mountains and the free space between them as (potential) valleys, with a globally deepest point at the goal location. The robot's course may then be compared to that of a ball which is rolling down the valleys towards the goal (cf. Fig. 6.9). With this depiction it is immediately clear that the robot can be trapped in local minima of the potential distribution, where the resulting force is zero. This is one of the drawbacks of this approach.

A standard workaround for this problem is to construct a graph of the locations of all local minima and modify the path planning process in such a way that local minima are avoided. Some papers describe specially designed APFs that completely avoid local

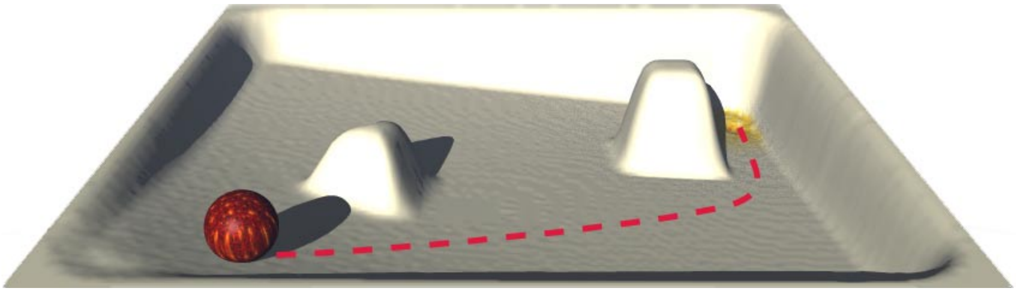


Figure 6.9: *A visualization of the Artificial Potential Field approach to robot navigation.*

minima [24, 1].

Other drawbacks and difficulties that are reported in the respective literature include:

- Some implementations do not always find existing globally optimal paths.
- The definition of the APFs is crucial for the success of the implementation.
- Some implementations are very time consuming, what makes them critical for certain applications.
- Some implementations require holonomic robots.

Advantages of "Potential Field" methods

Artificial potential field approaches have a number of advantages over other path planning approaches:

- As in most other path planning approaches, most contributions to this approach deal with two-dimensional maps, that is, the environment is projected onto a plane, and the APFs are defined as two-dimensional functions. Unlike other approaches, the Artificial Potential Field approach is easily extensible to three-dimensional maps, which are required to solve three-dimensional navigation tasks, since two-dimensional APFs are just simplifications of the three-dimensional natural potential functions.
- Although some implementations of Artificial Potential Fields may be quite time consuming, some authors have shown that the approach is suited for real time navigational tasks, for instance [3].
- It is not a prerequisite for this approach that the environment is completely known. Instead, navigation algorithms based on this approach may also work with an incomplete local map that is constructed from the robot's sensor readings and that is updated while the robot proceeds on its course.
- Kinematic constraints may be considered with Chuang's and Ahuja's [10] local optimization method.

6.5 Navigation under uncertainty

A pervasive fact that affects all three navigation phases, i.e., "path planning", "plan execution", and "action control" is *uncertainty:* sensor errors, fault or imprecision of actions, and gaps or flaws in the world model. Although many techniques exist to reduce the amount of uncertainty [2], it seems to be common understanding in research on autonomous mobile robots that methods for their navigation should account for uncertainty of sensor data, and other types of uncertainties. The uncertainty topic in general is very broad and leads far into the heart of general AI. In this section, we sketch some of its aspects as far as they are central for robot navigation. For a more comprehensive treatment of this topic in the spirit of AI, the reader is referred to [44, Part V].

Particularly crucial for robot navigation—and best covered in the literature—is *uncertainty of sensor data* [31, 37, 38, 39]. All sensors are subject to two classes of measuring errors. First, *systematic errors* occur from the sensor construction and the physical principles that are used for measurement; examples are optical errors in the lenses of video sensors, misalignment errors of wheels in odometry, or resolution errors in extremely close or very far distances in ultrasound transducers. Second, *unsystematic errors* occur from environment conditions and from the interaction between the robot and its environment; examples for such conditions are differences in lighting of the same scene in image processing, wheel slippage in odometry, and problems of reflection from polished surfaces for ultrasound transducers.

The other two types of uncertainty are typically treated in combination with sensor errors, e.g., [6, 7, 15, 16, 17, 18, 25, 27, 41]. They constitute a practical problem if and only if sensing is imperfect, because they could be easily avoided or corrected if sensors were arbitrarily exact and quick.

Action fault or imprecision is induced by imperfect actuators, where "action" means "motion" in the context of robot navigation, such as a turn or a translation. Imprecision, then, is a difference between the "ideal", intended or calculated motion and the real physical motion. Again, there are systematic and non-systematic errors, such as misalignment of wheels or wheel slippage, respectively. If odometry is done directly in the actuators, i.e., in gears, axles, or wheels, then sensors and actuators are not independent; consequently, sensor errors and motion errors are then just two sides of the same coin. This need not be the case; controlling the travelled distance using a laser-range finder, e.g., would be independent from motion fault.

Uncertainty of the world model in the context of navigation means inaccuracy of the (topological and/or metric) map. There does not seem to be work on systematizing such map errors; however, absence of some map item, presence of an item that does not exist in reality, and local (metric and/or topological) inaccuracy of a mapped item arguably yield different problems and should be handled or resolved in different ways. Note that map errors are not simply a result of sloppy world modeling; dynamic environments naturally imply that maps have to be updated.

In this section, we summarize the following clusters of literature that propose to handle different aspects of uncertainty or to handle uncertainty with different classes of techniques. Section 6.5.1 describes work on how to explicitly model positional uncertainty in grids. Section 6.5.2 reviews the use of Markovian decision processes to account for known sensor and action errors. Section 6.5.3 sketches approaches to build and correct maps in order to update the world model and, finally, our conclusions are drawn in Section 6.5.4.

6.5.1 Occupancy and certainty grids

Occupancy grids or *certainty grids* were introduced by Elfes and Moravec [15, 16, 41, 40] for modelling free-space. This approach uses a two-dimensional array of evenly-spaced cells corresponding to a horizontal grid imposed on the area to be mapped, providing a probabilistic, tessellated representation of spatial occupancy. The occupancy information is derived from robot sensor data, which are allowed to be imprecise. To construct the map, Moravec and Elfes obtain the cell state estimation by interpreting information about the distance to the next physical object in a given direction—as provided by, say, readings of an ultrasound transducer—using *probabilistic sensor models.* Such models are defined by a probabilistic density function of the form $\Pr(r|z)$ that relates a reading r to the true parameter space range value z (usually a one- or two-dimensional Gaussian range sensor). Informally, this function models the probability that the points inside the beam are empty and estimates the location of the point that caused an echo, if any.

The cell states are treated as *independent* random variables, so that Bayes' theorem allows occupancy grid cells to be updated sequentially, using readings taken from possibly multiple sensors over multiple points in space. The map is initialized with all cells set to a value representing *unknown;* it improves as readings are added. Overlapping readings of empty points in space reinforce each other, as do readings of occupied ones; on the other hand, evidence that a cell is empty would weaken its probability of being occupied, and vice-versa.

The absolute position of the robot need not be known when building the occupancy grid—only the relative position of range measurements with respect to each other must be known approximately. Any position uncertainty is incorporated into the map by a blurring operation (generally Gaussian) performed on the occupancy grid before integrating a new measurement. Finally, a deterministic world model can be obtained by using decision rules to assign discrete states to the cells, such as *occupied, empty,* or *unknown.*

The occupancy grid technique has been implemented in different systems, like [45], using a local and a global grid for position estimation. Moreover, [49, 51, 50, 8] have used this approach to extract different types of geometric features from the robot's environment. Figure 6.10 shows such an occupancy grid.

Based on this approach, [6] has introduced *position probability grids* as a counterpart to occupancy grids. The occupancy grid approach has been adjusted to estimate the absolute position of a mobile robot in a known environment. It is based on *Markov localization* over fine-grained possible positions (*states*, see Section 6.5.2). A *position probability grid P* is a three-dimensional array that contains in each field the probability that this field refers to the robot's current *pose* (position and orientation). Therefore, the robot must employ a given metric, e.g., CAD, model of the world. If *a priori* knowledge is available, this is applied to initialize the grid. Otherwise, a unimodal distribution is used.

The position probability for each grid-cell z is obtained by integrating movements and

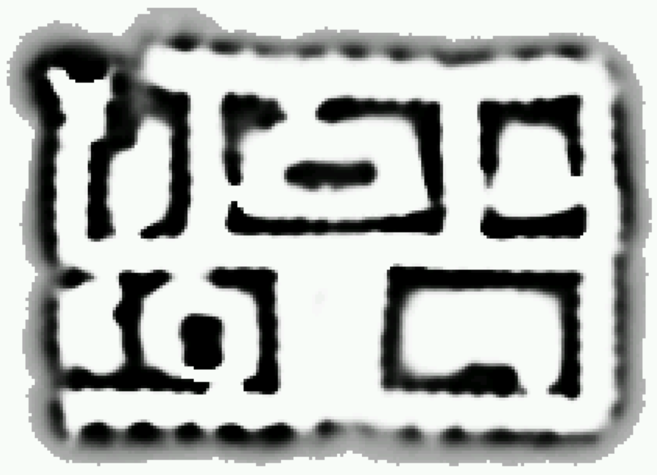


Figure 6.10: *An occupancy grid generated for the AAAI '94 competition. (Adapted from [6].)*

the likelihoods of sensor readings r, supposing the center of z is the robot's current pose. Recursive Bayesian estimation allows an incremental grid update for each new reading r_{n+1}:

$$\Pr(z|r_1 \wedge \cdots \wedge r_{n+1}) = \alpha \Pr(r_{n+1}|z) \Pr(z|\, r_1 \wedge \cdots \wedge r_n)$$

where α is a normalizer ensuring that the position probability sums up to 1 and $\wedge$ stands for 'and'. Thereby, the robot generates an internal belief as of where it might be.

In [7] probability grids are used for tracking the position of a mobile robot if its initial position is known. Therefore, its possible positions are reduced to a small section of the global position probability grid centered around its current estimated position. This center is changed according to the position information provided by the odometry data of the robot. To model the uncertainty of these measurements, an additional smoothing operation is applied (e.g., Gaussian). Sensor interpretations work as described in the global localization approach, limited to the small local grid. This approach is fast enough for real-time tracking in office environments.

6.5.2 POMDP Approaches

A number of approaches to robot navigation under uncertainty, such as [8, 18, 42, 49], are using classical *Markovian decision processes* (MDPs) as a representation for the problem of generating a good next move. The uncertainty of sensor readings and of motion control is cast into the concept of *partial observability* of the environment and the robot's actually executed actions, leading to *partially observable MDPs* (POMDPs); cf. [9, 49], on which this sketch is based. MDPs and POMDPs can be used to reason about action under uncertainty in general—not just about robot motions. In this section we confine ourselves to 'robot motions', which are implied whenever 'actions' are referred.

Formally, a POMDP consists of

- S, a finite set of *states* that represent situations in the environment,
- A, a finite set of *action representations,*

- T, the *state transition model,* i.e., a function mapping $S \times A$ into discrete probability distributions over S; we use *transition probabilities* $\Pr(s'|s,a)$ for all $s, s' \in S$ and $a \in A$ for denoting the probability that executing a in state s results in state s',
- R, a *reward function* from $S \times A$ into the real numbers that specifies the immediate reward that the robot gets from executing an action in a state,
- O, a finite set of *observation representations* representing observations to be made in the situations, and
- a function mapping S into discrete probability distributions over O; we use *observation probablities* $\Pr(o|s)$ for denoting the probability that $o \in O$ is observed in state $s \in S$.

Note that possible action and observation faults are represented by non-zero probabilities for more than one state or observation that results from a certain action in a certain state or from observing in a state, respectively. (A *fully observable* MDP is an instance of a POMDP, where the agent is able to tell with certainty from the observations in which situation it is.) Expected value operator is denoted by E, cf. Eq. (6.5.2).

In both MDPs and POMDPs, the reward function is used to plan for *good* moves rather than just any moves. Assume that a *policy* π is a mapping from S to A that specifies particular moves to be made in particular positions. Then the value $V_\pi(s)$ of a state $s \in S$ is intuitively the sum of expected rewards yielded by π in the future, where rewards are discounted by how far in the future they are expected. Let t denote a sequence of time steps and γ a discount factor, where $0 \leq \gamma < 1$. So, a way of formalizing V_π is

$$V_\pi(s) = E(\sum_{t=0}^{\infty} \gamma^t R_t(s,\pi))$$

where $R_t(s,\pi)$ is the reward yielded by the tth step of π when starting its execution in s. Obviously, (6.5.2) cannot be computed effectively as is, and will be reformulated below.

Turning to the uncertainty issue, the subjective belief of the agent of being in some situation is represented as a probability distribution over the set of states S. Every such distribution is called a *belief state.* For $S = \{s_1, \ldots, s_n\}$ we represent a belief state as $b = [b_1, \ldots, b_n]$, i.e., b_i represents the agent's subjective degree of belief of being in the situation represented by s_i. The b_i will sloppily be called *states* whenever the context makes clear that the value b_i stands for the "real" state that it represents (which, in turn, represents a situation in the world).

Belief states have to be updated to model the execution of an action and to include the results of observations. To start with the actions, let $b_{i,prio}$ and $b_{j,post}$ denote the degree of belief of being in s_i before and in s_j after executing an action a in s_i, respectively. Then for all $j \in \{1, \ldots, n\}$, we compute $b_{j,post}$ by

$$b_{j,post} = K \cdot \sum_{s_i \in S} \Pr(T(s_i, a) = s_j) b_{i,prio}$$

K is a normalizing factor which guarantees that the posterior belief state is a well-defined probability distribution, i.e., its components sum up to 1. If the transition model T is given by transition probabilities $\Pr(s'|s,a)$, then simply $b_{j,post} = \sum_{s_i \in S} \Pr(s_j|s_i,a) b_{i,prio}$.

For reckoning the information that results from an observation o into the recent belief state, note that an observation is assumed to leave the situation constant. Consequently,

$$b_{j,post} = K \cdot \Pr(o|s_j) b_{j,prio}$$

An obvious approach, as used in [18], for formulating the normalizing factor K, is $K = 1/\sum_{i=1}^{n} \Pr(o|s_i) b_{i,prio}$, i.e.,

$$b_{j,post} = \frac{\Pr(o|s_j) \cdot b_{j,prio}}{\sum_{i=1}^{n} \Pr(o|s_i) b_{i,prio}}$$

Let us briefly come back to the problem of computing good policies in the framework of POMDPs. Note first that the chances of finding an efficient algorithm for optimal policies in general are pretty dim, as this problem is known to be PSPACE-complete [43]. Second, note that having introduced the concept of belief state, we have cast a POMDP into the framework of a *belief* MDP (BMDP), i.e., something like a completely observable MDP over a different state set, namely, the set of belief states—so there might be some hope to solve it efficiently using one of the standard Operations Research techniques, such as linear programming. However, a BMDP is a *continuous space* MDP, its state space being infinite. So the best efficient algorithms we can expect are for finding good (rather than provably optimal) policies or for finding optimal ones in special or most cases (rather than in the general case). Paper [8] gives references to some BMDP algorithms, which, however, are practically useful only for very small sets of states and observations.

Cassandra [8], also states and compares several intuitively plausible ways of finding good policies. To give examples, these ways include

- the *most likely state* policy (MLS): pick the/a state with the highest probability weight in the current belief state and execute the action as dictated by the optimal *deterministic* (i.e., supposedly fully observable) policy; and
- the *action entropy* policy of following a policy such as MLS as long as the probability weight of the most likely state sticks out sharply or different highly likely states all induce the same action; however, if the entropy is high among likely states that would suggest different actions, then do an action that is expected to maximally reduce it.

For details, which are out of the scope of this text, the reader is referred to [8].

Paper [18] uses the POMDP model in a different way. It assumes that there is given a *deterministic,* "ideal" plan that has to be followed, such as a fixed path to be inspected. The POMDP model serves for running an internal model of the execution and observation failures that can occur when executing the given plan. The model includes—in terms of the recent belief state—an expectation about recently possible execution or observation errors; this expectation gets used for finding back to the original path, whenever unexpected observations occur.

6.5.3 Map building and map correction

Previous sections have mentioned maps as basic information sources for robot navigation, and it is in fact common and convenient to use them (see [2, 11] for overviews). There are two broad types of maps: *metric/grid-based* and *topological.* Maps of the first type typically represent space using occupancy grids (see Section 6.5.1). Topological maps are typically graphs, representing the environment as locally distinct places (nodes) connected by paths between two places (arcs).

Not all maps fall from heaven—in some cases, they have to be built by the robot in the first place, or they must be corrected if and when they have turned false in a changing environment. If sensors and actuators were arbitrarily precise, then map building would be easy; in reality, they are not. Therefore, map building or map learning are fundamental problems in autonomous robot navigation.

Before reviewing briefly some relevant approaches, let us state a caveat. There is no way for a real, physical robot of learning effectively without knowing previously, given that the learning has to occur within its short life time and given that it is subject to hard physical constraints: Whenever a robot has tumbled down an unforseen staircase, it has no longer the opportunity to learn that there was a staircase, that there exists such a thing as a staircase, or that floors need not be flat and infinitely large in the first place. So the most principled limitation for map building is that even though no map need be given in advance, the robot has to start out with a sensor equipment and set of concepts that fit the environment. In some cases, providing these may be considered a harder task than map building is.

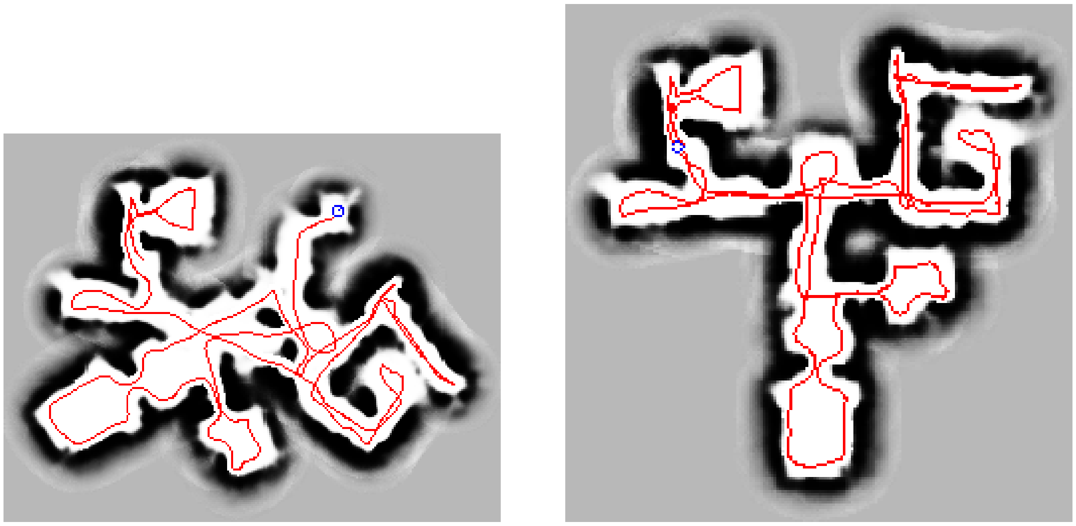
Keeping this in mind as a general constraint, here is a list of features that make map learning technically difficult:

- Sensor uncertainty caused by *noise*.
- Positional uncertainty caused by *drift* and *slippage* (dead-reackoning errors).
- *Interpretation* uncertainty: Sensor data often do not allow to clearly *distinguish* between different objects like closed doors and walls.
- Uncertainty about the *initial position* of the robot.
- Space and time *complexity* of the very map learning procedures.
- *Dynamic nature* of the environment, such as people moving around the robot.

In [53] an attempt to solve some of these problems with active exploration is presented. To construct a grid-based model of the environment, sensor values are processed by an artificial neural network interpreting them in the context of neighbouring sensors. The output of this network is a mapping into probabilities for occupied-ness (see [53] for details). Multiple interpretations are integrated using Bayes' rule. Therefore, the approach requires high accuracy in the alignment of the robot's coordinate system and in the global world coordinates. In order to achieve this, three kinds of information are combined:

1. Wheel encoders,
2. correlation between a local and the constructed global map, and
3. wall orientation (assuming that typical indoor environments have right angles between walls).

Figure 6.11 shows an example for the effectiveness of this technique.



6.11.1 *Only the wheel encoders are used.*

6.11.2 *All three position estimation techniques as described (wheel encoders, correlation, wall orientation) are used.*

Figure 6.11: *Two maps as constructed from the same environment with different position estimation mechanisms (adapted from [53]).*

Based on these data, Thrun and Bücken can generate a topological maps by splitting the occupancy grid into coherent regions, separated by *critical lines,* using a *Voronoi diagram.* Note that the number of regions is far smaller than the number of cells in the grid. In consequence, the complexity of path planning is by far lower for such maps than for maps that are based on the grid representation only, while the planned paths are typically only a little longer than optimal. Thrun has also evaluated an encouraging approach that allows the robot to select (topological) landmarks by itself. For details, which are out of the scope of this text, see [52].

Simmons and Koenig [51] try to overcome some of the above-mentioned position uncertainty problems using POMDP parameter adaption (see Section 6.5.2 for a brief description of POMDPs). Their approach starts with a rough Markov model of the environment. A simple *expectation-maximization* algorithm adjusts the initial probabilities of the POMDP by observing the robot's interaction with its environment, the *execution trace.* This data is generated automatically by the POMDP navigation. The learning algorithm maximizes the probability of generating the observations $o \in O$ for the given actions $a \in A$. Supplementary constraints, such as symmetry of sensors and in the map, are employed to reduce the amount of data needed for training. Note that this approach leaves unchanged the initial structure of the POMDP; moreover, learning is *unsupervised;* and *passive*, in the sense that the learning algorithm itself is running in the background, not interfering with the robot control.

Although Simmons and Koenig's approach can adapt to most kinds of uncertainty simultaneously, they describe in [50] a refinement that focuses on adjusting distance probabilities. The initial uncertainty of the metric distance between two landmarks is given as a uniform distribution over distance intervals from 1 to 3 meters. The robot learns a new probability distribution over the possible distances by using the above-mentioned algorithm. The initial structure of the POMDP is changed if the highest probability for some distance gets close to the upper bound of its interval, increasing this interval accordingly. Real distances are not reflected perfectly in learned distances, because they correspond rather to *perceived* corridor lengths due to dead-reckoning errors.

6.5.4 Conclusions

Before concluding by summing up the advantages and drawbacks of navigation under uncertainty, let us first emphasize that the matter to be discussed is not whether or not uncertainty is *per se* good or bad. Uncertainty simply exists in many interesting robot applications. It is no virtue to be able to handle it, but it is a burden if the application area does not permit to avoid handling it. Certain advantages and drawbacks of the individual approaches for navigation under uncertainty are listed hereunder.

Advantages of approaches to navigation under uncertainty

Occupancy and certainty grids

- This approach allows different types of sensors to be integrated over time. Multiple sensors can be used to improve world models and provide higher levels of fault tolerance, compensating for "blind spots" of a single type of sensor.

- No *geometric features,* such as object patches or line segments, need to be extracted from sensor data. Hence the information loss by early sensor data interpretation can be avoided.

- Various robot tasks can be addressed through operations performed directly on the occupancy grid. In path planing, for example, the cost of a path can be directly related to the corresponding cell probabilities by *value iteration* (cf. [53]).

- Uncertainty can be handled explicitly, as by *decision rules* and *probabilistic sensor models.*

- They are simple and easy to construct and maintain even for complex environments.

- The occupancy grid approach often leads to any-time algorithms that can come to decisions regardless of the available computation time, where (static) decision quality typically improves with computation time.

POMDP approaches

- Action faults and observation errors can be handled in a common framework with well-defined mathematical properties.

- Position uncertainty can be easily expressed that involves likelihood for different, spatially disconnected positions.

Map building and map correction

- No need to provide pre-existing maps, let alone correct ones.
- Fast and easy adaption to new or changed environments.
- No need for pre-defined landmarks.

Drawbacks of approaches for navigation under uncertainty

Occupancy and certainty grids

- In general, grid-based approaches imply a high space and time complexity depending on the number of grid-cells (which can, however, be reduced by using a map resolution hierarchy [40]).
- Discretization of a given environment poses a fundamental modeling problem: neither a sensible grid cell size is known in advance, nor is it obvious where cell boundaries should be.
- Change in the environment (moving objects) leads to blurred maps as the occupancy information is integrated over time.
- Sensor errors must be known and cast into a probabilistic error model.

POMDP approaches

- All (interesting) positions in the world must be cast into a finite, discrete set. This set must not change, or else the probability weights in the belief states may turn meaningless.
- The map must be (topologically) correct and cannot be changed.
- The relative probabilities of action and observation errors have to be known (estimated) in advance.
- If transitions from a state to another state can possibly be overlooked (due to observation errors, e.g.) then the sets of possible successor states of actions, as defined in the transition model, may grow very large.
- There is no general effective way for calculating optimal policies.

Map building and map correction

- For a robot with its limited operation time and physical speed, learning requires prior knowledge centered around the to-be-learned data.
- *Revisiting:* It cannot be guaranteed that the robot notices if it is visiting the same spot twice; not noticing it leads to inconsistent maps.

- *False déjà vue:* It cannot be guaranteed that the robot does not interpret two similar-looking spots as identical; identifying them leads to inconsistent maps.

6.6 General remarks

The reviewed approaches to path planning for navigation of mobile robots seem to be separable into two larger categories. The first category, as introduced in sections 6.2 and 6.3, may be described as "pure" path planning or "classical" path planning. In this category, the "path planning problem"—as it is typically called in the respective literature—is approached from a rather theoretical point of view, where the goals are mainly to solve the problem, finding "exact" or "optimal" solutions, or finding a method that guarantees to find a path if one exists.

Typical features of this category are:

- A completely known, static, environment is assumed. The exact positions of the obstacles, the outer limits of the environment, and the start and goal locations are represented in a two-dimensional map.
- The obstacles and boundaries are "inflated" by an amount that depends on the geometry of the robot under consideration. Simultaneously, the robot's dimensions are shrunk to a point. This is done in order to be able to abstract from the robot's kinematics, and thus to reduce the complexity of computations.
- A trajectory for the idealized point-like "robot" is constructed via graph search and a possible successive pre-processing stage.

Classical approaches may work well for static, known environments, which are not too cluttered, and for holonomic robots. Unfortunately, the applicability of the described methods for robust navigation of real robots is not always clear, since in many papers on path planning, no statements are made with respect to the other phases of navigation.

However, path planning usually is not just a task *per se*, but serves a certain goal: A path plan that has been computed by a planner is meant to be executed by a robot. The plan may be seen as a basic set of navigational instructions that have to be put into operation in a consecutive phase of navigation, called plan excecution. As pointed out in the introduction to this section, the execution of the planned actions and their operationalized descriptions is to be controlled in a phase called action control. One reason for this is, for instance, that the robot must be able to cope with its own possible deviations (imprecision of motion) from the planned trajectory. The problem of detecting such deviations between the robot's actual position and the planned position is called the *localization problem.* In order to implement a reliable action control, this problem has to be tackled (among others). Some planning methods, like Occupancy Grids and POMDP-based methods (cf. section 6.5), provide a basis for action control.

Both methods—as most of the methods described in sections 6.4 and 6.5—belong to the second category of approaches to path planning, which we call "practical" path planning. Typically, these approaches take into account some of the factors that are required for robust navigation of real robots. These factors include:

- uncertainty of motion, sensor data, and map information,
- dynamic nature and three-dimensionality of environments,
- non-triviality of robot kinematics, and
- suitability for real-time computation.

Artificial potential field methods, for instance, are well suited for real-time computation and application in dynamic or unknown environments. However, extending these methods to three-dimensional environments and/or non-point non-holonomic robots and/or handling uncertainty increases their computational complexity and may severely diminish their real-time suitability.

While there is a wealth of well-investigated classical path planning methods[2], it seems that practical path planning methods deserve further investigation. The review of the literature on mobile robot navigation has also shown that, at present, there is no universally accepted "general purpose" navigation method. This is not surprising, since there is a vast variety of robots, drives, sensors, applications, and environments. It may be assumed that, at best, a number of robust navigation methods will be developed that may work well for certain common systems of robot, application, and environment. In order to approach this goal, a number of challenging research topics remain to be further investigated, including

- path planning in dynamic environments,
- map building, map maintenance, and map matching,
- navigation under uncertainty of motion, sensor data, and map information,
- path planning for three-dimensional navigation, and
- interweaving of the three phases of path planning, plan execution, and action control.

[2] The following textbooks and overviews of the subject [2, 30, 32, 13, 29] may serve as additional reading material on path planning or, more generally, navigation for mobile robots

Bibliography

[1] Al-Sultan, K. S., and Aliyu, M. D. S. A new potential field-based algorithm for path planning. *Journal of Intelligent and Robotic Systems 17* (1996), 265–282.

[2] Borenstein, J., Everett, H. R., and Feng, L. *Navigating Mobile Robots.* A K Peters, Wellesley, MA, 1996.

[3] Borenstein, J., and Koren, Y. Real-time obstacle avoidance for fast mobile robots in cluttered environments. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation* (Los Alamitos, CA, 1990), vol. 1, IEEE, IEEE Computer Society Press, pp. 572–577. (Cincinatti, OH, May 13–18, 1990).

[4] Brooks, R. A. Solving the find-path problem by good representation of free space. *IEEE Transactions on Systems, Man, and Cybernetics SMC-13*, 3 (1983), 190–197.

[5] Brooks, R. A., and Lozano-Pérez, T. A subdivision algorithm in configuration space for findpath with rotation. *IEEE Transactions on Systems, Man, and Cybernetics SMC-15*, 2 (1985), 224–233.

[6] Burgard, W., Fox, D., Hennig, D., and Schmidt, T. Estimating the absolute position of a mobile robot using position probability grids. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI '96)* (Menlo Park, CA, 1996), vol. 2, AAAI, AAAI Press/MIT Press, pp. 896–901. (Portland, OR, August 4–8, 1996).

[7] Burgard, W., Fox, D., Hennig, D., and Schmidt, T. Position tracking with position probability grids. In *Proceedings of the First Euromicro Workshop on Advanced Mobile Robots (EUROBOT '96)* (Los Alamitos, CA, 1996), IEEE, IEEE Press, pp. 2–9. (Kaiserslautern, Germany, October 9–11, 1996).

[8] Cassandra, A., Pack Kaelbling, L., and Kurien, J. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '96)* (1996).

[9] Cassandra, A., Pack Kaelbling, L., and Littman, M. Acting optimally in partially observable stochastic domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI '94)* (Menlo Park, CA, 1994), vol. 2, AAAI, AAAI Press, pp. 1023–1028. (Seattle, WA, August 1–4, 1994).

[10] Chuang, J.-H., and Ahuja, N. Path planning using the Newtonian potential. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation* (Los Alamitos, CA, 1991), vol. CH2876-1, IEEE, IEEE Computer Society Press, pp. 558–563. (Sacramento, CA, April 9–11, 1991).

[11] Cox, I., and Wilfong, G., Eds. *Autonomous Robot Vehicles.* Springer Verlag, Berlin, 1990.

[12] Crowley, J. L. Navigation for an intelligent mobile robot. *IEEE Journal on Robotics and Automation RA-1*, 1 (1985), 31–41.

[13] Crowley, J. L. Path planning and obstacle avoidance. In *Encyclopedia of Artificial Intelligence*, S. C. Shapiro and D. Eckroth, Eds., vol. 2. Jon Wiley & Sons, New York, 1987, pp. 708–715.

[14] Dautenhahn, K. *Pfadplanung beim Menschen.* Ph.d., University of Bielefeld, Faculty of Biology, Bielefeld, Germany, January 1993.

[15] Elfes, A. Sonar-based real-world mapping and navigation. *IEEE Journal on Robotics and Automation RA-3*, 3 (1987), 249–265.

[16] Elfes, A. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer* (June 1989), 46–57.

[17] Gutierrez-Osuna, R., and Luo, R. C. Lola: Probabilistic navigation for topological maps. *AI Magazine 17*, 1 (SPRING 1996), 55–62.

[18] Hertzberg, J., and Kirchner, F. Landmark-based autonomous navigation in sewerage pipes. In *Proceedings of the First Euromicro Workshop on Advanced Mobile Robots (EUROBOT '96)* (Los Alamitos, CA, 1996), IEEE, IEEE Press, pp. 68–73. (Kaiserslautern, Germany, October 9–11, 1996).

[19] Horswill, I. Polly: A vision-based artificial agent. In *Proceedings of the Eleventh Conference on Artificial Intelligence (AAAI '93)* (Menlo Park, CA, 1993), AAAI, AAAI Press, pp. 824–829. (Washington, D.C., July 11–15, 1993).

[20] Horswill, I. *Specialization of Perceptual Processes.* Ph.d., MIT, Cambridge, MA, 1993.

[21] Jorgensen, C., Hamel, W., and Weisbin, C. Autonomous robot navigation. *Byte* (January 1986), 223–235.

[22] Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research 5*, 1 (1986), 90–99.

[23] Khoditscheck, D. E. Exact robot navigation by means of potential fields. In *Proceedings of the 1987 IEEE Conference on Robotics and Automation* (Piscataway, NJ, 1987), vol. 1, IEEE, IEEE Computer Society Press, pp. 1–6. (Raleigh, NC, March 31–April 3, 1987).

[24] Kim, J.-O., and Khosla, P. K. Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation 8*, 3 (June 1992), 338–349.

[25] Kosaka, A., and Kak, A. C. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *CVGIP: Image Understanding 56*, 3 (1992), 271–329. Invited paper. See also erratum [26].

[26] Anonymous. Erratum: Volume 56, number 3 (1992): Akio Kosaka and Avinash C. Kak, "Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties," pp. 271-329. *Computer Vision, Graphics, and Image Processing. Image Understanding 57*, 2 (March 1993), 263.

[27] Kosaka, A., Meng, M., and Kak, A. C. Vision-guided mobile robot navigation using retroactive updating of position uncertainty. In *Proceedings of the 1993 IEEE International Conference on Robotics And Automation* (Los Alamitos, CA, 1993), R. Werner and L. O'Conner, Eds., vol. 2, IEEE RAS, IEEE Computer Society Press, pp. 1–7. (Atlanta, GA, May 2–6, 1993).

[28] Krogh, B. A generalized potential approach to obstacle avoidance control. In *Robotics Research: The Next Five Years and Beyond* (1984), pp. 1–15. (Bethlehem, PA, August 14–16, 1984).

[29] Kuipers, B. J., and Levitt, T. S. Navigation and mapping in large-scale space. *AI Magazine 9*, 2 (SUMMER 1988), 25–43.

[30] Latombe, J.-C., Ed. *Robot Motion Planning*, vol. Fourth Printing 1996. Kluwer Academic Publishers, Boston, MA, 1991.

[31] Lazanas, A., and Latombe, J.-C. Motion planning with uncertainty: A landmark approach. *Artificial Intelligence 76*, 1–2 (1995), 287–317.

[32] Levitt, T. S., and Lawton, D. T. Qualitative navigation for mobile robots. *Artificial Intelligence 44* (1990), 305–360.

[33] Lingas, A. The power of non-rectilinear holes. In *Proceedings of the 9th Colloquium on Automata, Languages, and Programming* (Berlin, 1982), LNCS, Springer-Verlag, pp. 369–383. (Aarhus, Denmark, 1982).

[34] Lozano-Pérez, T. Automatic planning of manipulator transfer movements. *IEEE Transactions on Systems, Man, and Cybernetics SMC-11*, 10 (1981), 224–238.

[35] Lozano-Pérez, T. Spatial planning: A configuration space approach. *IEEE Transactions on Computers C-32 2* (February 1983), 108–120.

[36] Masoud, A. A., and Bayoumi, M. M. Robot navigation using the vector potential approach. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation* (Los Alamitos, CA, May 1993), R. Werner and L. O'Conner, Eds., vol. 1, IEEE, IEEE Computer Society Press, pp. 805–811.

[37] Miura, J., and Shirai, Y. Hierarchical vision-motion planning with uncertainty: Local path planning and global route selection. In *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots And Systems (IROS '92)* (Los Alamitos, CA, 1992), IEEE and RSJ, IEEE Computer Society Press, pp. 1847–1854. (Raleigh, NC, June, 1992).

[38] Miura, J., and Shirai, Y. Vision-motion planning with uncertainty. In *Proceedings of the 1992 IEEE International Conference on Robotics And Automation* (Los Alamitos, CA, 1992), vol. 3, IEEE RAS, IEEE Computer Society Press, pp. 1772–1777. (Nice, France, May 12–14, 1992).

[39] Miura, J., and Shirai, Y. An uncertainty model of stereo vision and its application to vision-motion planning of robot. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI '93)* (San Mateo, CA, 1993), vol. 2, AAAI, Morgan Kaufmann, pp. 1618–1623. (Chambery, France, August, 1993).

[40] Moravec, H. P. Sensor fusion in certainty grids for mobile robots. In *Sensor Devices and Systems for Robotics*, A. Casals, Ed. Springer-Verlag, Heidelberg, 1989, pp. 253–276.

[41] Moravec, H. P., and Elfes, A. High resolution maps from wide angle sonar. In *Proceedings of the 1985 IEEE Conference on Robotics and Automation* (Piscataway, NJ, 1985), IEEE, IEEE Computer Society Press, pp. 116–121. (Saint Louis, MO, March 25–28, 1985).

[42] Nourbakhsh, I., Powers, R., and Birchfield, S. DERVISH: An office-navigating robot. *AI Magazine 16*, 2 (1995), 53–60.

[43] Papadimitriou, C., and Tsitsiklis, J. The complexity of Markov decision processes. *Mathematics of Operations Research 12*, 3 (1987), 441–450.

[44] Russell, S., and Norvig, P. *Artificial Intelligence: A Modern Approach.* Prentice Hall, Englewood Cliffs, NJ, 1995.

[45] Schiele, B., and Crowley, J. L. Comparison of position estimation techniques using occupancy grids. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '94)* (1994).

[46] Schwartz, J. T., and Sharir, M. On the piano movers problem: I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. In *Planning, Geometry, and Complexity*, J. T. Schwartz, M. Sharir, and J. Hopcroft, Eds., ablex series in artificial intelligence. Ablex Publishing Corporation, Norwood, NJ, 1987, pp. 1–50. Reprint of article in *Communications on Pure and Applied Mathematics 36*, 1983, pp. 345–398.

[47] Schwartz, J. T., and Sharir, M. On the piano movers problem: II. General techniques for computing topological properties of real algebraic manifolds. In *Planning, Geometry, and Complexity*, J. T. Schwartz, M. Sharir, and J. Hopcroft, Eds., ablex series in artificial intelligence. Ablex Publishing Corporation, Norwood, NJ, 1987, pp. 51–96. Reprint of article in *Advances in Applied Mathematics 4*, 1983, pp. 298–351.

[48] Shahidi, R., Shayman, M., and Krishnaprasad, P. S. Mobile robot navigation using potential functions. In *IEEE R& A* (Washington, D.C., 1991), IEEE RAS, IEEE Computer Society Press, pp. 2047–2053. (Sacramento, CA, April 9–11, 1991).

[49] Simmons, R., and Koenig, S. Probabilistic robot navigation in partially observable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '95)* (San Mateo, CA, 1995), Morgan Kaufmann, pp. 1080–1087. (Montréal, Canada, August 20–25, 1995).

[50] Simmons, R., and Koenig, S. Passive distance learning for robot navigation. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML'96)* (1996), pp. 266–274.

[51] Simmons, R., and Koenig, S. Unsupervised learning of probabilisitic models for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '96)* (1996), pp. 2301–2308.

[52] Thrun, S. A Bayesian approach to landmark discovery and active perception in mobile robot navigation. Tech. Rep. TR CMU-CS-96-122, Carnegie Mellon University, Pittsburgh, PA 15213, May 1996.

[53] Thrun, S., and Bücken, A. Learning maps for indoor mobile robot navigation. Tech. Rep. TR CMU-CS-96-121, Carnegie Mellon University, Pittsburgh, PA 15213, April 1996.

Chapter 7

On Control and Behaviors

by Giulio Sandini, José Santos-Victor, Panos Trahanias
Lena Gaga, Stelios Orphanoudakis, Wolfram Burgard and Fredrik Bergholm

Loosely speaking, *visual behaviours* refer to certain subtasks performed by a navigating sensor-based mobile platform, requiring use of vision. Some typical sub-tasks of visual-based navigation are (a) obstacle and collision avoidance, (b) landmark-based navigation of visual homing type, or, map-based or active-vision based landmark navigation, (c) wall and corridor-following, (d) ground-floor monitoring, (e) docking upon reaching destination, (f) keeping fixed distance to moving object, (g) monitoring heading direction (Ch.4), (h) path planning[1] as well as a host of other behaviors triggered by the current situation (e.g. move-into-a-room). These subtasks, in turn, usually requires visual competences of the type mentioned in Chapter 2.

Some control theory aspects of sub-tasks (c) and (f) can be found in [7, 33]. In this chapter, after a brief introduction to behavioral approach in visual-based navigation, Section 7.2 presents an example of a control strategy for sub-task (a), taking into account the *kinematics* of the mobile platform. Then, a description of experiments in the 90-ies addressing wall/corridor following and docking is included. This is not a complete review of work performed in the field, but rather a concrete example of relevant on-going work in Europe, on these topics. This section does not cover the interesting problem how to combine these behaviors. Section 7.4 reviews attempts to integrate planning and control.

7.1 Behavioral approach to visual-based navigation

From the methodological point of view the so-called "behavioral approach" may be employed in navigational trials, to demonstrate that it is indeed possible to close the motor-control loop on the basis of direct visual information, extracted in real-time from a camera mounted on the vehicle to be controlled.

Breaking up a task into behaviours is a wellknown approach in sensor-based navigation, with advantages such as:

[1] Classical path planning as described in the previous section can apply to the visual homing case also, if one assumes some memory of the 2D layout of free space and obstacles, around some visual landmarks.

- the possibility of breaking down the general problem into tractable, yet meaningful parts and consequently
- the possibility of demonstrating the performance of the system since the beginning (i.e. without having to implement the overall system before starting the experimental part);
- the possibility of identifying the requirements in terms of processing power and specific modalities to define the common algorithms and their efficient integration;

In [7] the general problem of visual-based navigation is tackled in the general framework of the behavioral approach, and is formulated as a problem that can be stratified on two common behaviors:

> *Visual Homing*, that includes definition of landmarks, their recognition and selection;
>
> *Visual Navigation*, that includes the use of image motion information to controlling the position and attitude of the vehicle that carries the vision system.

Figure 7.1 describes graphically this situation where the two behaviors are applied. The *visual homing* behavior is responsible for monitoring the landmarks, identifying them, checking the sequence in which they appear and could select better ones to achieve the goal. A special case of this behavior is that which involves mainly selecting new landmarks, while progressing along a path between very sparse landmarks, say from point A to B, possibly for remembering how to return to the starting point, A. A basic visual capabilitby in the visual homing context is that of identifying and detecting the approximate whereabouts of these landmarks, as well as strategic selection of new landmarks. The *Visual Navigation* behavior is responsible for continuosly monitoring some visual motion parameters that can be useful to continuos control of the trajectory between (sparse) landmarks. These parameters include peripheral motion and obstacle detection.

7.2 Fast collision avoidance for synchro-drive robots

One of the ultimate goals of indoor mobile robotics research is to build robots that can safely carry out missions in hazardous and populated environments. For example, a service-robot that assists humans in indoor office environments should be able to react rapidly to unforeseen changes, and perform its task under a wide variety of external circumstances. Most of today's commercial mobile devices scale poorly along this dimension. Their motion planning relies on accurate, static models of the environments, and therefore they often cease to function if humans or other unpredictable obstacles block their path. To build autonomous mobile robots one has to build systems that can perceive their environments, react to unforeseen circumstances, and (re)plan dynamically in order to achieve their missions.

The work at Univ. of Bonn, as described in this section, focused on one particular aspect of the design of such a robot: the reactive avoidance of collisions with obstacles. In this development, the following requirements have been considered:

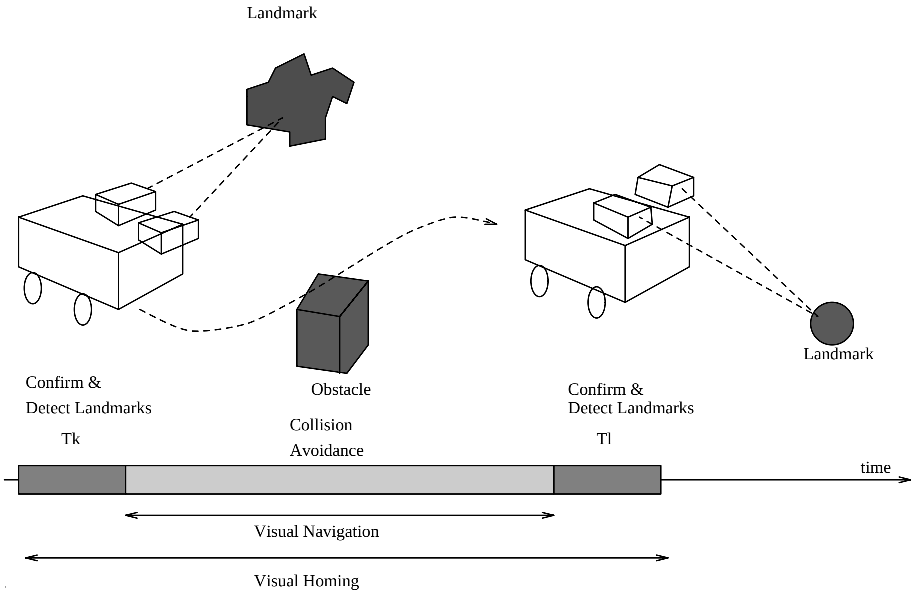


Figure 7.1: *The figure illustrates the concepts used in [7] for stratification of the mobile robot navigation problem. During the navigation process two different situations are anticipated: one related with the landmark identication, selection and continuously monitoring of them and another related the continuous control of the trajectory between landmarks.*

- The robot must travel safely even with high speed. It therefore must take the dynamic constraints into account.
- The robot should react adequately and rapidly to unforeseen circumstances. This requires fast techniques for the detection of obstacles and the selection of appropriate steering commands.
- The robot should make maximum progress towards the goal. Whenever advantageous, should modify its travel direction to stay away from obstacles.

The *dynamic window approach* developed at Univ. of Bonn [38, 41] meets the above requirements. It is based on an approximation of the exact motion equations of synchro-drive robots[2] by sequences of circular arcs. The dynamic window approach first prunes the overall search space of trajectories to a 2D space of circular trajectories. Then, the search space is reduced to the admissible velocities allowing the robot to stop safely without colliding with an obstacle. Finally, the *dynamic window* restricts the admissible velocities to those that can be reached within a short time interval given the limited accelerations of the robot. This way the dynamic constraints are properly taken into account. After that, the robot picks a trajectory at which it can maximize its translational velocity and the distance to obstacles, yet minimize the angle to its goal relative to its own heading direction. This is done by maximizing an appropriate objective function. The combination of all objectives leads to a very robust, efficient, and reactive collision avoidance strategy.

As local world model the dynamic window approach uses an obstacle line field [37] (cf. Sec. 5.3.3), which is a two-dimensional description of sensory data relative to the

[2] A brief tutorial on synchro-drive robots (synchro drive = 3 or more wheels mechanically coupled in a certain way) can be found in [2, pp.13-14].

robot's position. For the purpose of safety, the approach is conservative and integrates every sensor reading into the local world model.

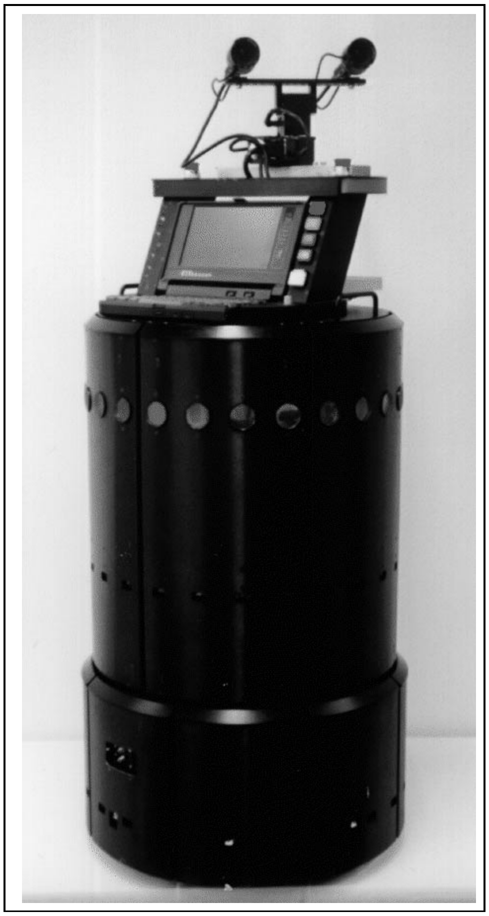


Figure 7.2: *The mobile robot RHINO*

The dynamic window approach has been implemented and tested using RHINO (see Figure 7.2), a B21 robot manufactured by Real World Interface Inc. In extensive experimental evaluations using ultrasonic proximity sensors for the construction of the obstacle line fields, the method has proven to avoid collisions reliably with speeds of up to 95 cm/sec on several robots in several indoor environments The method has also successfully been operated based on cameras and infrared detectors as sensory input.

In the case when there also are cameras onboard, the proximity information is extracted from color images using the segmentation technique described in [39, 40], where the cameras were used to detect different objects on the floor and to classify them. The objects were added to the obstacle line field so that the robot was able to move around small objects on the floor (for example bottles) which are not visible using only the ultrasonic sensors. In this way, the sonar-based map is augmented by vision information.

The approach differs from previous approaches in that it is derived directly from the motion dynamics of a mobile robot and, therefore, takes the inertia of the robot into account—which is particularly important if a robot with torque limits travels at high speed. In the experiments performed with this approach in cluttered and dynamic environments, the mobile platform has been safely operated with speeds of up to 95 cm per second. We envision this approach to be particularly useful for robots that travel at even higher speeds and for low-cost robots with limited motor torques, for which the constraints imposed by the motion dynamics are even more imperative.

7.3 Visual behaviors for robot motion

This section presents some of the experiments performed at DIST, University of Genoa, regarding visual perceptual behaviors. These experiments, and the approaches adopted, are illustrative of the behavioral approach in vision-based navigation.

7.3.1 Wall and corridor following

With this behavior experiments of a real-time navigation system driven by two cameras pointing laterally to the navigation direction (called *Divergent Stereo*) were performed [33]. Similarly to what has been proposed in [12, 6], the approach in [33] assumes that, for navigation purposes, the driving information is not distance (as it is obtainable by a stereo setup) but motion and, more precisely, by the use of qualitative optical flow information computed over non-overlapping areas of the visual field of two cameras.

Following this idea, a mobile vehicle has been equipped with a pair of cameras looking laterally (much like honeybees) and a controller based on fast, real-time computation of optical flow has been implemented. The control of the mobile robot is based on the comparison between the apparent image velocity of the left and the right cameras.

The approach is based on the use of two cameras mounted on a mobile robot with the optical axes directed in opposite directions such that the two visual fields do *not* overlap (*Divergent Stereo*[3]). Range is perceived by computing the apparent image speed on images acquired during robot motion.

A real-time computation of optical flow has been implemented, based upon the constraints imposed by the geometry of the cameras and by the navigation strategy. In [33, p.436] the normal flow coincided with the image flow, because it was assumed that vertical image flow component was zero.

A PID controller was used to close the visuo-motor control loop. The closed loop behavior was studied, based on models of the different control system components. The analysis of the control system design led to a suitable configuration for the PID controller.

The approach has been tested using real-time experiments to accomplish different navigation tasks such as: sharp turn around an obstacle, wall following, passage through a narrow opening. The influence of the control parameters on the system behavior was studied and the results confirmed, within the assumptions made, the discussion on the

[3] The word ‘stereo’ here should not be associated with stereopsis, because there is no overlapping of the visual fields.

control system design. A controller for the robot forward velocity was also studied and implemented. Experiments have been made to show the improvement achievable, by including this control loop in cluttered environments.

Finally, through the insertion of a sustained behavior, the robot is able to operate in environments rather more complex than a simple corridor, showing the capability of operating in sparsely textured corridors, and following unilaterally textured walls.

All the experiments were performed without the need for accurate depth or motion estimation, nor required a calibration procedure (besides the manual positioning of the two cameras).

The main features of the implementation can be summarized as follows:

- **Purposive** definition of the sensory apparatus and the associated processing. The approach proposed, in fact, cannot be considered general but, with limited complexity, solves a relevant problem in navigation: the control of heading direction in a cluttered environment.

- Use of **qualitative and direct** visual measures. In our opinion this is not only a "religious" issue but, more importantly, a way to achieve a reasonable autonomy with limited computational power. Successful examples of this approach have recently appeared in the literature both with respect to reflex-like behaviors for obstacle avoidance [31, 8] and in relation to more "global" measures of purposive navigation parameters [9].

- **Continuous use of visual measures**. A further aspect worth mentioning is the attempt made at developing a sensory system providing a continuous stream of environmental information. A first advantage is the increased robustness implicit in the use of repeated measures (no single mistake produces catastrophic errors) and a secondary, and potentially more important, advantage is the possibility of implementing sensory-motor strategies where the need for a continuous motor control is not bounded by an "intermittent" flow of sensory information. This paradigm is, in our opinion, a non trivial evolution of some active vision implementations where the motion of the (active) observer is seen "only" as a way of taking advantage of the stability of the environment (e.g. by moving the vehicle along pre-programmed, known trajectories to reduce uncertainty). The use of vision **during** action [32], on the contrary, may be a very powerful extension of the concept of active observer by exploiting the use of dynamic visual information not only at the "reflexive" level of motor control.

- **Simplicity**. This feature is often regarded as an engineering and implementation aspect and, as such, is not explicitly considered a scientific issue. This view, in our opinion, must be changed if reasonable applications of computer vision are addressed. The issue of simplicity, however, should not be considered, within specific aspects of intelligent actor's design (such as, sensory systems, mechanical design, computational architecture etc.) but must be considered at system level. The divergent stereo is an example of such an holistic view of simplicity where the purpose is achieved by a comprehensive analysis and integration of visual processing, sensor design, sensor placement, control law and vehicle structure. In this respect low-level animals (and insects in particular) are extremely interesting examples of "simple" actors where all engineering aspects are mixed exploiting not only "computational" issues but, more importantly, the cooperation of "intelligent" solutions which, if considered separately, may look like interesting implementational tricks but, once acting together, produce intelligent behaviors.

7.3.2 Docking

Visual-based behaviors for docking operations in mobile robotics have been implemented in [34]. Two different situations have been analyzed. In the *ego-docking*, the robot is equipped with a camera and the egomotion controlled when docking to a surface, whereas in the *eco-docking*, the camera and all the necessary computational resources are placed in a single external docking station, which may serve several robots. In both situations, the goal consists in controlling both the orientation, aligning the camera optical axis with the surface normal, and the approaching speed (slowing down during the maneuver). The *station keeping* in the *eco-docking* is here mentioned, because it can be solved with similar techniques as the docking behaviour.

These goals are accomplished without any effort to perform 3D reconstruction of the environment or any need to calibrate the setup, in contrast with traditional approaches. Instead, we use image measurements directly to close the control loop of the mobile robot.

In the approach proposed, the robot motion is directly driven by the first order time-space image derivatives or, equivalently, the normal flow field which is the only component of the image flow field that can be accurately estimated due to the aperture problem.

The docking system is operating in real-time and the performance is robust both in the *ego-docking* and *eco-docking* paradigms. The approach is based on the direct use of image measurements to drive the motion controller, without any intermediate reconstruction procedure. As the visual input, only normal flow field is used.

An affine model is fitted to the measured motion field, and a fast estimation procedure robust to outliers was used. The affine parameters of the flow field are expressed as a function of the robot motion and directly used to close the motor control loop. The closed loop strategy proposed uses direct visual measurements to control the robot forward speed (based on time to crash measurements) and heading direction. (The same control structure is used to for the *ego-docking* or *eco-docking* cases.)

A real time implementation was realized and a robust docking behavior achieved. A major issue is the fact that there is no need to calibrate the camera intrinsic or extrinsic parameters nor is it necessary to know the vehicle motion.

7.3.3 General remarks

The behaviors described in this section, besides the peculiarities summarized below, stress the concept of visuo-motor coordination in, at least, two ways :

1. The visual measure used (normal flow) is elicited by the motion of the robot.

2. The perception/action loop is not decoupled in the sense that the performance of the perceptual processes is also a function of the control ones.

The consequences of this approach may be, in our opinion, very general particularly in the area of navigation and manipulation.

A purposive motor action coupled to a specific perceptual process directly elicits a behavior (a behavior emerges as Brooks puts it), without the need for "understanding" the structure of the scene or continuously monitoring the geometric features of the environment. In doing that, the system behaves in a parsimonious way by utilizing the minimum amount of information necessary to achieve the current goal (even if it is obvious, it is worth noting that only one goal at a time can be pursued and that, even in case of concurrent processes, the motor commands must be unique).

In the all experiments performed the behavior of the robot is solely controlled by the direct link between the optical flow estimation and the motor commands generated by the controller: no matter what the robot "sees" it will end up in front of the "docking wall" and perpendicular to it or avoid an obstacle, or maintain a given distance from a wall.

For all these visual behaviors there is no need to know the calibration and/or the vehicle motion parameters and, moreover, they are all based on the same visual information (optical flow). Two factors characterize the different behaviors:

1. The part of the visual field analyzed (in which part of the visual field is the attention focused on).

2. The control law adopted (the direct link between visual information and rotation of the wheels).

One challenge is how to combine these different behaviors to accomplish more complex tasks. The simplest solution would be to design a "planner" eliciting the appropriate behavior according to the current situation. For example, the centering behavior if the robot is navigating along a corridor or the wall following or the docking behavior to stop in front of a door or the obstacle detection to avoid obstacles. The problem, then, is no more to understand the environment (each behavior embeds all the perceptual processes necessary to understand the aspects of the environment strictly necessary) but to understand (or to know) the situation. Of course, this is not necessarily simpler than understanding the environment, however, the fact that it may not be necessary to "tune" a perceptual process, interpret the perceptual information and transform this into motor commands, but, on the contrary, "appropriate action" is totally embedded inside the single behaviors, seems to be a very powerful way of breaking a complex problem into simpler ones and, consequently, of designing incremental systems whose capabilities are bounded by the number of behaviors implemented and do not require a general purpose architecture to be developed beforehand.

7.4 Integrating planning and control

Classical approaches to motion planning compute trajectories based on previous knowledge of the environment and the goals of the mobile agent. Such approaches are usually unable to cope with the uncertainty and the dynamics of real-world workspaces and a number of reactive approaches have been proposed to overcome this limitation [10, 15, 13, 36]. Reactivity provides immediate response to unpredicted environmental situations by considering only local (in time and space) information. However, reasoning about future consequences of actions (strategic planning) is still needed in order to intelligently solve complex tasks. Saffiotti et al. [26, 27, 28, 29, 30, 22, 23, 24] have developed an approach for integrating strategic planning and local reactivity by adopting a "complex controller" model, i.e. a controller model that simultaneously satisfies strategic goals coming from the planner, and low-level innate goals (e.g. obstacle avoidance).

The complex controller model is built bottom-up, starting from small units of control, called *control schemas*. These are simple and basic movement capabilities and are implemented using multivalued functions from internal states to control actions. Control schemas are made *flexible* in real environments by not committing to specific effector commands, but rather by computing a measure of desirability over the space of these commands (since many commands can generate the same type of movement).

Complex controls can emerge from control schemas by appropriate combination of the latter. When two (or more) control schemas are not conflicting, then a *conjunctive composition* satisfies their conjunction. For example, two control schemas that follow a corridor and move fast, respectively, can be composed to follow a corridor moving fast. Blending of reactive and purposeful movement can (in many cases) be achieved by a *chaining composition* of control schemas. Consider for example a control schema to reach a goal with no obstacles present and a second one to avoid obstacles. Composing these two schemas creates a complex control that reaches a goal in the presence of obstacles.

Two key notions are used in the combination of control schemas and the execution of actions (goals): the notion of *embedding* in the environment, and the notion of *context*, or circumstances, of execution which gives the applicability conditions for a schema. Embedding facilitates the execution of certain actions in an environment. That is, only when the agent faces an open door, the action "move into the room" will be accomplished. The approach to embedding is model-based: a perceptual subsystem is used to tie (anchor) representations of the objects in the internal state to their counterparts in the real world.

In order for the control system to be highly reactive and, at the same time, exhibit "intelligent behavior", the perceptual subsystem is organized in a bottom-up fashion. A *Local Perceptual Space* (blackboard-like mechanism) is used to keep track of sensor information. Sensor readings are registered into that, and in some cases are used unfiltered (e.g. by an obstacle-avoidance scheme). When needed, higher-level filters group these data into surface information, which are finally grouped into recognized objects.

In the above framework, actions, or *behaviors*, are the results of executing control schemas in an environment under certain assumptions. A behavior does not uniquely identify one sequence of movements, or execution, but a set of executions that are possible. Behaviors are linked to *goals*, expressed as sets of satisfactory executions. The *good behaviors* for a goal are those that, when executed under their own context, produce executions that satisfy the goal.

As pointed out by the developers of the above approach [26] "the complex control methodology is not a radical departure from the reactive planning methods ... Rather, it is a theoretical approach to two significant problems with these architectures: how to form complex movements, and how to link these up with more abstract deliberation processes."

From our point of view, the complex control methodology presents an elegant approach to robot motion planning and control, that is also amenable to rigorous mathematical treatment. The latter stems from the employment of multivalued logic as the formal framework for developing the approach. The interesting feature of using multivalued logic arises from the nature of the control problem, which involves complex trade-offs between competing goals.

7.4.1 Comparisons

By contrasting the above approach with others from the literature, it is obvious that certain similarities and influences, as well as differences can be identified. The two methodologies of subsumption architectures [3, 4] and situated automata [25, 16] aim at producing embedded agents that perform complex tasks. The complex control methodology borrows from them the idea of decomposing complex behavior into the composition of simple behaviors. It departs in that it prefers explicit model-based perception and analogical representations of the world as part of embedding the controller.

Several proposals and implementations of reactive planning architectures have recently appeared, which combine a low-level control mechanism with one or more deliberative layers [5, 14, 35, 11, 10, 19, 1, 20, 21]. The complex control methodology relates with these architectures, concentrating on the relation between the control and sequencing layers, and their tying to the more abstract planning level.

Finally, the complex control methodology bears similarities to methods originating from the field of control theory. The *artificial potential fields*, mentioned in Section 6.4, are a typical representative of this class of methods, and a way to connect (path) planning and control (robot's path following [17, 18]. The presented methodology shares with the potential field methods the intuition of describing basic units of control as local preferences, expressed on a continuous scale, and then combining these preferences to build complex controls. An important difference on the technical level stems from the use of multivalued logic to express goals and controls. While the vectors generated by pseudo-forces are *summary* descriptions of the preferred control, the desirability functions are assignments of utility values to *each* possible control. As a result, the combined preferred controls are the preferred controls of the combined preference criteria, and not the combination of the controls individually preferred by each criterion.

7.4.2 General remarks

In conclusion, the complex control methodology seems to offer (at least in theory) adequate tools and power to model complex goals and behaviors of autonomous mobile agents.

This has also been demonstrated by its designers through its implementation on a mobile robotic platform [26, 29]. However, it can be argued that in many real applications this methodology can not be readily applied, due to a number of issues that are not sufficiently covered:

- The desirability functions, as well as other functions introduced (trajectory, context, goal), can not be easily constructed in practice. Although they are rigorously defined, their realization in a particular application is left open, relying mostly on the intuition of the designer.
- The dynamic modification of complex behaviors, when "things do not proceed as planned", is an important issue that arises frequently in practice. Although not included at the moment, the framework could be further enhanced with the development of adequate indices of performance and their use to patch an existing plan [26].
- More importantly, the relation of perception and action is only superficially covered by this approach. Although it is shown how perceptual behaviors can be used to relate perception and action, the very nature of perceptual processes is left untackled. This can not be overlooked, since the success of many simple control schemas relies on effective recognition and registration of environmental structures.

We believe that the last item above deserves particular attention, if we aim at applications addressing realistic workspaces. At the same time, we understand that perception of the environment is usually the bottleneck in these cases, being a research field still in its infancy. Under this ascertainment, we may sum up by acknowledging the contribution of the complex control methodology, as a rigorous mathematical approach that integrates into the same framework strategic planning and local reactive schemas. Moreover, the methodology can be seen as a unification of many previous approaches to planning, since it borrows and employs desirable features from them.

Bibliography

[1] R.C. Arkin. Integrating behavioral, perceptual and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6:105–122, 1990.

[2] Borenstein, J., Everett, H. R., and Feng, L. *Navigating Mobile Robots.* A K Peters, Wellesley, MA, 1996.

[3] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE J. Robotics Auromat.*, RA-2(7):14–23, Apr. 1986.

[4] J. Connell. *Minimalist Mobile Robotics: A Colony-style Architecture for an Artificial Creature.* Academic Press, 1990.

[5] J. Connell. SSS: A hybrid architecture applied to robot navigation. In *Proc. of the IEEE Conf. on Robotics and Automation*, pages 2719–2724, 1992.

[6] D. Coombs and K. Roberts. Centering behaviour using peripheral vision. In D.P. Casasent, editor, *Intelligent Robots and Computer Vision XI: Algorithms, Techniques and Active Vision*, pages 714–21. SPIE, Vol. 1825, Nov. 1992.

[7] J. Dias, F. Bergholm, P. Fornland, " VIRGO, Vision-Based Robot Navigation Research Network", Report, TRITA-NA-P9627, ISRN KTH/NA/P-96/27, Royal Inst. of Technology, Stockholm, Sweden, Sept. 1996.

[8] W. Enkelmann. Obstacle detection by evaluation of optical flow field from image sequences. In *Proc. of the 1st. European Conference on Computer Vision*, pages 134–138, Antibes (France), 1990. Springer Verlag.

[9] C. Fermüller. Global 3d motion estimation. In *Proc. of the IEEE CVPR*, New York, USA, June 1993.

[10] J.R. Firby. An investigation into reactive planning in complex domains. In *Proc. AAAI Conf.*, 1987.

[11] J.R. Firby. Adaptive execution in complex dynamic worlds. Technical Report 672, Dept. of Computer Sci., Yale University, 1989.

[12] N. Franceschini and J. Pichon and C. Blanes. Real time visuomotor control: from flies to robots. In *Proc. of the Fifth Int. Conference on Advanced Robotics* Pisa, Italy, June 1991.

[13] E. Gat. *Reliable Goal-Directed Reactive Control for Real-World Autonomous Mobile Robots.* PhD dissertation, Virginia Polytechnic Institute and State University, 1991.

[14] E. Gat. Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots. In *Proc. AAAI Conf.*, 1992.

[15] L.P. Kaelbling. An architecture for intelligent reactive systems. In M.P. Georgeff and A.L. Lansky, editors, *Reasoning About Actions and Plans.* Morgan Kaufmann, 1987.

[16] L.P. Kaelbling. Compiling operator descriptions into reactive strategies using goal regression. Technical Report TR-90-10, Teleos Research, Palo Alto, CA, 1990.

[17] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robotics Res.*, 5(1):90–98, 1986.

[18] J. C. Latombe. *Robot Motion Planning.* Kluver Academic Publishers, Boston, MA, 1991.

[19] D. McDermott. Planning reactive behavior: A progress report. In *Proc. of the DARPA Workshop on Innovative Appr. to Planning, Scheduling and Control*, 1990.

[20] D.W. Payton, J.K. Rosenblatt, and D.M. Keirsey. Plan guided reaction. *IEEE Trans. Systems, Man and Cybern.*, 20(6), 1990.

[21] D.W. Payton. Exploiting plans as resources for action. In *Proc. of the DARPA Workshop on Innovative Appr. to Planning, Scheduling and Control*, 1990.

[22] E.H. Ruspini. Fuzzy logic in the flakey robot. In *Proc. of the Int. Conf. on Fuzzy Logic and Neural Nets. (IIZUKA)*, pages 767–770, Japan, 1990.

[23] E.H. Ruspini. On the semantics of fuzzy logic. *Int. Journal of Approximate Reasoning*, 5:45–88, 1991.

[24] E.H. Ruspini. Truth as utility: A conceptual systasis. In *Proc. of the 7th Conf. on Uncertainty in AI*, Los Angeles, CA, 1991.

[25] S.J. Rosenschein. The synthesis of digital machines with provable epistemic properties. Technical Report 412, SRI Artificial Intelligence Center, Menlo Park, CA, 1987.

[26] A. Saffiotti, K.G. Konolige, and E.H. Ruspini. A multivalued logic approach to integrating planning and control. Technical Report 533, SRI Artificial Intelligence Center, Menlo Park, CA, Jun. 1993.

[27] A. Saffiotti, E.H. Ruspini, and K.G. Konolige. Integrating reactivity and goal-directedness in a fuzzy controller. In *Proc. of the 2nd Fuzzy-IEEE Conf.*, San Fransisco, CA, 1993.

[28] A. Saffiotti, E.H. Ruspini, and K.G. Konolige. Robust control of a mobile robot using fuzzy logic. In *Proc. of the European Congress on Fuzzy and Intell. Technologies*, Aachen, Germany, 1993.

[29] A. Saffiotti, E.H. Ruspini, and K.G. Konolige. A fuzzy controller for flakey, an autonomous mobile robot. Technical Report 529, SRI Artificial Intelligence Center, Menlo Park, CA, 1993.

[30] A. Saffiotti. Some notes on the integration of planning and reactivity in autonomous mobile robots. In *Proc. of the AAAI Spring Symposium on Foundations of Automatic Planning*, pages 122–126, Stanford, CA, 1993.

[31] G. Sandini and M. Tistarelli. Robust obstacle detection using optical flow. In *Proc. of the IEEE Intl. Workshop on Robust Computer Vision*, pages 396–411, Seattle, (WA), October 1990.

[32] G. Sandini, F. Gandolfo, E. Grosso, and M. Tistarelli. Vision during action. In Y. Aloimonos, editor, *Active Perception.* Lawrence Erlbaum Associates, 1993.

[33] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo for robot navigation: Learning from bees. In *IEEE International Conference on Computer Vision and Pattern Recognition.*, 1993.

[34] J. Santos-Victor and G. Sandini. Visual Behaviors for Docking. *Computer Vision and Image Understanding*, 65(3), May 1997.

[35] R. Simmons. An architecture for coordinating planning, sensing and action. In *Proc. of the DARPA Workshop on Innovative Appr. to Planning, Scheduling and Control*, 1990.

[36] J. Yen and N. Pfluger. A fuzzy logic based robot navigation system. In *Proc. AAAI Fall Symposium on Mobile Robot Navigation*, pages 195–199, Boston, MA, 1992.

[37] Joachim Buhmann, Wolfram Burgard, Armin B. Cremers, Dieter Fox, Thomas Hofmann, Frank Schneider, Jiannis Strikos, and Sebastian Thrun. The mobile robot Rhino. *AI Magazine*, 16(2):31–38, Summer 1995.

[38] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Controlling synchro-drive robots with the dynamic window approach to collision avoidance. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.

[39] Thomas Hofmann, Jan Puzicha, and Joachim Buhmann. Unsupervised segmentation of textured images by pairwise data clustering. In *Proceedings of the IEEE International Conference on Image Processing, Lausanne*, 1996.

[40] Thomas Hofmann, Jan Puzicha, and Joachim Buhmann. Deterministic annealing for unsupervised texture segmentation. In *EMMCVPR, Venice*, Lectures Notes in Computer Science. Springer Verlag, 1997.

[41] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, to appear.

Chapter 8

Industrial Applications

by Antonella Semerano, Jens Arnspang and Kostas Chandrinos

Applications of Machine Vision have manifested themselves throughout Europe and the rest of the world during the last decade as stable tools for quality inspection, production control, and robot guidance. Examples can be found in several industrial branches, of which we mention but a few.

Tuborg and Carlsberg, Danish breweries, use automatic visual inspection of boxes for the purpose of identifying defects and foreign logos. Holmegard, a Danish glass-ware producer, use automatic visual quality measures in their production line of bottles. H-profil, a Norwegian wood company, use automatic visual inspection of wood quality and identification of knots. Sidmar, a Belgium steel producer, use stereo techniques for measuring widths of steel plates during production. Rover, the British/German car manufacturer, use 3D visual guidance of complicated robot motion in the assembly line. These examples of industrial machine vision systems are all characterised by daily use and stable performance within a priori specified limits. Proven consequences include higher production quality, lower production cost and stable production lines.

In fact, there are many examples based on static cameras in various industrial applications. On the other hand, there are also examples of autonomous vehicles incorporating vision-based navigation in industrial applications. Most well-known *camera-based mobile platforms*, used in real-life applications, *do not use* vision in the normal sense of the word, but revert to sensor based navigation techniques. To take one example, TMR's `Helpmate`, which carries trays and lab tests in hospitals, projects a light grid on the floor which a camera monitors. Suitable engineering of the environment allows for the platform to redetermine and track it's own position at any given time without fixed tracks or guide wires. The reason for the lack of *pure vision* applications in robotic navigation is that up to-date technology cannot guarantee the robustness required by industrial tasks. Unfortunately, this has led to self-posed limitations of potential industrial applications: until vision-based navigation becomes trusted for industrial level implementation we can only design low-level navigational tasks based at best on other sensor information fusion. VIRGO intends to review the current status as well as propose and demonstrate ways in which vision for navigational purposes will attain industrial maturity and, when fused with other sensors, will allow for state-of-the-art robotic applications.

8.1 Autonomous guided vehicles

A number of autonomous guided vehicles (AGVs) have appeared in the past few years. They can be divided in two broad categories: Industrial strength AGVs that are in effect 'intelligent' forklifts empowered with sensor abilities for detecting cargo, obstacles, position, and passage of product through a line and for transmitting data such as addresses, ID numbers, etc. Such devices are already used in an increasing number of warehouse plants accross Europe. Current AGVs require certain artificial guidance features (e.g. whitre lines) in their environment so taht they can successfully navigate through it. However, the goal of placing an AGV in unknown terrain and allowing it to map its environment and find paths for itself has still not been reached, although significant progress has been made with the use of passive camera techniques. Extensions to the classic ideas surrounding AGVs, have led to Autonomous Underwater Vehicles (AUVs) like *Marius* from the Technical University of Denmark as well as 'Rover', the autonomous vehicle participating on NASA Pathfinder mission on planet Mars.

On the other hand, there are also attempts to adapt vision based navigation for the needs of ordinary vehicles. Such an attempt is the NavLab project (see also Section 5.2.2) from Carnegie Mellon University in Pittsburgh, PA where a multitude of visual techniques have been incorporated to benifit the navigation of a test vehicle amidst standard traffic. This vehicle combines a scanning laser rangefinder with radars, sonars and a trinocular stero vision system to attain 3-D perception data for several levels of representation. A number of autonomous behaviors that have been demonstrated in the past years include finding a parking space and parallel parking, convoy following and position estimation. The most striking European example is perhaps VaMP, from Universitat der Bundeswehr Munchen (UBM). This is an adapted 5-ton Mercedes 500 SEL from Daimler-Benz which also participates in the project, equipped with front and rear camera along an intelligent cruise control system. This vehicle is capable of autonomous travelling in highways and has demonstrated abilities of driving in dense traffic including autonomous lane changes. To that direction, many researchers are proposing refinements on how to tackle visual information for navigation purposes such as the work presented recently by the Advanced Computer Research Centre, at the University of Bristol, concerning road junction recognition and turn-offs for autonomous road vehicle navigation.

8.2 Rehabilitation robots

Driven by the needs of elderly and disabled people RTD efforts that address intelligent robotic aids are underway. Such efforts are also backed by economic considerations, since the market prospects for such systems seem quite optimistic. A number of different products have surficed in the market trying to combine strengths built from mobile robotics and 'smart' manipulators and actuators. In fact a standard interface (M3S) has already been proposed that allows modular development of 'smart' parts to be integrated on mobile platforms. The *Handy 1* rehabilitation robot from Rehab Robotics Ltd. is on example of such assistive technology. It is a robot comprising a set of actuators that can facilitate a physically challenged person in daily routine, like eating, shaving, teeth cleaning, make-up etc. The Finish Technical Research (VTT) on the other hand has devoted work on building a product module that helps controlling the movement of a wheelchair, employing ultrasonic omnidirectional scanning sensors. In a similar vein, PERMOBIL demonstrates navigational abilities through the use of sensors and suitable environment engineering. A box of photocells is mounted on the robotic wheelchair. Reflective tape is placed on the floor showing the path that the chair should follow. The line-steering can be augmented with an ultrasound-based collision detection and prevention system which stops the wheelchair in case of obstacles. In fact, most navigational approaches in current generation robotic wheelchairs are based on the measurements obtained by a ring of sensors, which can be readily interpreted for obstacle avoidance tasks. There are, however, certain limits to what can be achieved by using only local range measurements. Although they support reactivity to local environment features, they are inadequate for autonomous navigation. Current work focuses on navigation towards user-selected targets. Computer vision techniques are employed for target tracking; sonar based reactivity is employed for local, fine control of motion. An example application of such techniques has emerged from the SENARIO project undertaken by a consortium led by ZENON S.A. where a prototype of the proposed 'sensor aided intelligent wheelchair navigation' system was produced and tested on a commercial powered wheelchair. This combination can provide the end user with extended capabilities of obstacle avoidance, path selection and goal satisfaction through personalization of environment maps by means of a teaching period. There on, the robotic platform can function in a semi-autonomous or fully autonomous mode.

8.3 Autonomous scrubbing machines

Trade centres, covering thousands of square metres, recently multiplied all over the world. Approximately 1000 such centres are found in France. The open hours of these centres are from 9am to 10pm, which makes the late night hours attractive for floor cleaning in the sense, that neither customers nor employees are present then. However, managers prefer to minimise overnight electricity costs by scheduling restocking of supplies and cleaning of floors simultaneously in the early day light hours before the centres open. This causes crowding of restocking and cleaning staff while supply boxes on the floor hinder efficient cleaning of the floors of the trade centres. A better solution would involve a robot floor scrubber which can work overnight.

A European consortium has produced 3 prototypes of such autonomous scrubbing machines (from now on abbreviated ASMs), which are capable of four hours of continuous operation. Each machine is performing satisfactory to an extent that the rising of a new and more advanced industrial need has been justified, namely the need for central remote supervision of several ASMs, working independently in different and remote centres. The system, which has been developed, resembles a typical human operated scrubbing machine. The autonomous navigation version has been developed on the basis of advanced technologies, which include a navigation system for steering, ultrasonic sensors for collision avoidance, lasers for location recognition, and an on board computer. The robot records the steps needed to clean the floors and may be programmed for repeating the steps ('Teaching by Showing' method).

Problems like slippage on the floor may cause the machine to lose its track. For this reason the system is capable of redetermining its position by shooting laser beams at various wall landmarks. If the ultrasound system encounters a pallet or a person on the floor, the machine stops until the obstruction is removed. In order to minimise damage from collisions, the ASM moves slowly (2 km/h). The reduced speed allows cleaning formulas, which are less polluting for the environment.

One inherent problem with ASMs still has to be taken into account. If an ASM gets stuck or does not clean properly, the failure might be discovered at the end of the cleaning period a few hours before the centre reopen. A possible solution is to provide a monitoring strategy, in which a supervising operator may survey several ASMs and intervene in case of events, exterior to the preprogrammed action patterns of the ASMs. The operator may reprogram or redirect the ASMs in question or may call a local person to intervene[1]. The development of the ASM prototype described above, has been made possible by the recently ended EUREKA project EU 1094 - CLEAN. At present the partners are are performing several on site tests on the CLEAN machine and aiming at its dissemination and industrial exploitation. Based on actual performance tests the research and industrial partners are of the opinion that a world wide distribution of such ASMs may be possible by the end of 1997.

8.4 Future Trends

Robots have made great progress in factory applications, being actively used in such manufacturing processes as welding, painting, handling and assembly, enabling long strides forward in rationalization and labor savings for production. Now, such robots are beginning to gradually step out from inside the factory. In the not too distant future, delivery robots will distribute parcels in office buildings and exploratory robots will roam the surface of other planets. Reliability and efficiency are key issues in the design of such autonomous systems. They must deal reliably with noisy sensors and actuators and with incomplete knowledge of the environment. They must also react efficiently in real time to overcome dynamic situations that include obstacle avoidance, navigation, path planning and task scheduling. Such situated agents must exhibit goal-directed behavior in real time and will therefore require a high level of sensing capabilities, particularly purposive vision.

[1] In fact, relying on a *mobile steering camera system*, and overall on an effective Human Computer Interface, the supervising operator will be given opportunities to detect undesired difficulties which may occur on remote sites.

Chapter 9

Summary

by Fredrik Bergholm and Panos Trahanias

The preceding chapters have focused on various issues encountered in vision-based navigation: visual competences, environment landmarks, biologically inspired approaches to navigation, world representations, path planning, control and behaviors; moreover, industrial applications that involve autonomous navigation have been briefly reviewed. The above issues constitute basic research topics in vision-based navigation, and the in-depth and rigorous study of them is expected to greatly contribute to the advancement of the field. Throughout this text, we have made an effort to overview the related areas and, at the same time, present important results in more detail. In this chapter we review briefly the material presented and summarize our concluding remarks.

Research on extracting information on egotranslation direction and three-dimensional structure from motion, from optical flow or image flow, has shown that it is not easy to achieve this, in the presence of unknown general rotation. For practical usage in vision-based robotics, it seems that the only (safe) way of obtaining this information is by *short-circuiting* some of the equations, either by constraining the motion, or by assuming that information about some motion parameters is available by other means.

Given that (Section 2.2) the structure from motion (SFM) and the motion from motion (MFM) problems are hard to solve, it is tempting to by-pass them by attempting to estimate *qualitative* things. Independent motion detection, for example, is a qualitative decision made from motion data, where a number of reasonably efficient methods exist, and are being developed. Another example, is perception of planarity, or, of objects not being in a certain plane. This is also a mature field, where considerable success has been made in recent years (cf. Chapter 7, Section 3.3). For obstacle avoidance, which is a key capability in mobile robotics, structure maps of this qualitative kind are quite valuable.

Checking *planarity* is, under all circumstances, a key visual perception ingredient. When visual-based autonomous vehicles operate in indoor environments, identifying and montoring the floor plane must be an essential preoccupation. Extended vertical planar surfaces, walls, are important places from which planar visual landmarks may be selected. Such landmarks are valuable for several reasons, e.g., they are associated with projective invariance: once four points have been identified in a pattern, the whole image may be mapped, as seen from any other viewpoint disregarding resolution (cf. Section 3.3).

Object(s) tracking is a technical discipline, serving fields such as mobile surveillance, medical imaging, autonomous vehicles, image coding, and which has shown a vigorous growth in recent years. In vision-based navigation a coming trend, anticipated by both industrial needs and researchers, is the fact that several mobile agents should cooperate with[1] each other, and, also interact with moving people in a sensible way. This is a kind of *mobile surveillance* and, hence, the growth of useful tracking methods and algorithms is a fortuitious trend matching these future aims.

Autonomously mobile systems need to monitor time-to-collision or time-to-contact for safe navigation. For time-to-collision (time-to-contact), there are two main branches of estimation schemes, one image velocity-field based using *divergence*, and another based on image trajectories measuring image *accelerations*. However, the latter has not been used much in computer vision and robotics. The two approaches have different advantages and disadvantages and further work comparing and combining them seems worthwhile.

Binocular systems seem to have many usages beside the obvious one—stereopsis. It may (Chapter 2) for example play a role in independent motion detection, or time-to-collision estimates. Even a binocular system with no stereo matching at all is highly useful, as in the *divergent stereo* set-up, described in Section 7.3, since peripheral vision is a powerful obstacle avoidance module.

Visual homing (Chapter 3) involves very long image sequences, and techniques must be found that 'fight the memory usage explosion'. Current efforts are directed towards a *selective* approach in memorizing image patterns. In this context, salient image areas are detected, and the corresponding patterns may be stored for future reference. Still, such patterns need to be unambiguously recognized during a later, navigation session, to effectively facilitate localization in the workspace and, therefore, autonomous navigation. Pattern descriptors that are invariant under view-point transformations are very well suited for this task. Unfortunately, descriptors that are view-point invariant and, at the same time, powerful in representing the image patterns, are quite difficult to find. Current research efforts are addressing this topic, employing results from projective geometry (projective invariants).

The action control version of visual homing is *visual piloting*, whereby images (memorized views) are exploited to adjust a navigation path coming closer to a landmark destination. The desert ant studies in Chapter 4 are in this context of interest. Relevant experiments have shown that simple dead-reckoning mechanisms are rendered more powerful by a global compass-like information. By monitoring the discrepancy between desired (memorized) and actual heading direction, a *sensory-motor control* law may continuously compensate for detours due to obstacle avoidance. The examples in Chapter 4 concerns polarization patterns in the sky not visible to the human eye, or, a dominant light source (indoors). For outdoors autonomous navigation with human-like vision, the position of the sun, or other global patterns on the sky, may serve as short-term visual compass, and similar mechanisms may be exploited. Qualitative dead reckoning in terms of 'elapsed time' taking into account the extra time needed during detour operations could also be conducted, for navigation tasks such as " head north for so-many minutes and return to roughly to original position, all the time circumnavigating obstacles causing detours from heading direction."

[1] Or: supervise, monitor each other.

Hybrid representations have a tendency to be more powerful than the individual ones. Let us mention two examples from the text. First, a metric *grid-based* representation with a *topological* map on top has unique advantages that cannot be found for either approach in isolation: the grid-based representation, which is considerably easy to construct and maintain in environments of moderate complexity (e.g. 20 by 30 meters), models the world consistently and disambiguates different positions. The topological representation, which is grounded in the metric representation, facilitates fast planning and problem solving. Secondly, *reflex-like sensory-motor* behavior, such as is performed by any crawling insect, is truly useful for obstacle and collision avoidance, but the combined behaviour where some simple *"reasoning" about dynamics* in the environment for example, something like: "since a lot of people stream by that spot there is probably a door or an elevator nearby", will potentially be still more powerful.

More cooperation between zoology and robotics is likely to be a fruitful strategy, not only for visual-based navigation, but also for sonar and other sensors. In the marine world there are dolphins, and many species of fish, including deep sea fish, with quite advanced non-visual proximity sensors from which dynamic behaviors can be studied. Moreover, as illustrated in Chapters 4 & 7, approaches that employ low level reflex-like behaviors, where the sense-think-act cycle is not needed but replaced by sensory-motor control, are very promising in autonomous navigation trials. Such approaches, although simple in their implementation, achieve seemingly intelligent behaviour, fulfilling tasks such as crude visual homing, and obstacle avoidance. A more complete list of tasks for which this approach can be used would be valuable for the field of visual-based navigation.

As it may have become evident from the individual chapters presented, research in computer vision and robotic navigation has provided theoretical results and algorithmic implementations that address most of the issues involved in visual-based navigation. However, efforts to combine such results in meaningful ways and integrate them into working systems have been limited to-date. Although many pieces of the "puzzle" have been contributed, we are still lacking the framework that would facilitate their integration into a complete system. Therefore, besides pursuing individual research topics, future efforts should also be directed into such large-scale developments that would demonstrate advanced navigational capabilities.