Preprint. Final version in Chapter 9 of
"Innovations in Intelligent Machines - 1", Springer 2007
http://www.springer.com/978-3-540-72695-1

Toward Robot Perception through Omnidirectional Vision

José Gaspar†, Niall Winters‡, Etienne Grossmann†, José Santos-Victor†

†Instituto de Sistemas e Robótica,
Instituto Superior Técnico,
Av. Rovisco Pais, 1,
1049-001 Lisboa - Portugal.
(jag,etienne,jasv)@isr.ist.utl.pt

‡Media Lab Europe,
Sugar House Lane, Bellevue,
Dublin 8,
Ireland.
Niall.Winters@medialabeurope.org

"My dear Miss Glory, Robots are not people. They are mechanically more perfect than we are, they have an astounding intellectual capacity..."

From the play R.U.R. (Rossum's Universal Robots) *by Karel Capek, 1920.*

1 Introduction

Vision is an extraordinarily powerful sense. The ability to perceive the environment allows for movement to be regulated by the world. Humans do this effortlessly but still lack the understanding of how perception works. In the case of visual perception, many researchers, from psychologists to engineers, are working on this complex problem. Our approach is to build artificial visual systems to examine how a robot can use images, which convey only 2D information, in a robust manner to drive its actions in 3D space. The perceptual capabilities we developed allowed our robot to undertake everyday navigation tasks, such as *"go to the fourth office in the second corridor"*.

A critical component of any perceptual system, human or artificial, is the sensing modality used to obtain information about the environment. In the biological world, for example, one striking observation is the diversity of "ocular" geometries. The majority of insects and arthropods benefit from a wide field of view and their eyes have a space-variant resolution. To some extent, the perceptual capabilities of these animals can be explained by their specially adapted eye geometries. Similarly, in this work, we explore the advantages of having large fields of view by using an *omnidirectional camera* with a 360° azimuthal field of view.

Once images have been acquired by the omnidirectional camera, a question arises as to what to do with them. Should they form an internal representation of the world? Over time, can they provide intrinsic information about the world so as no representation is required? These fundamental questions have long been addressed by the computer vision community and go to the heart of our current understanding of visual perception. Before going on to detail our approach, a brief overview of this understanding will be provided.

1.1 Background

In the mid-20^{th} Century, Gibson put forward an ecological approach to vision. Emphasis was placed on the optic array: its invariant properties specify all information (structures and events) about the environment. The theory is that this information should be "picked

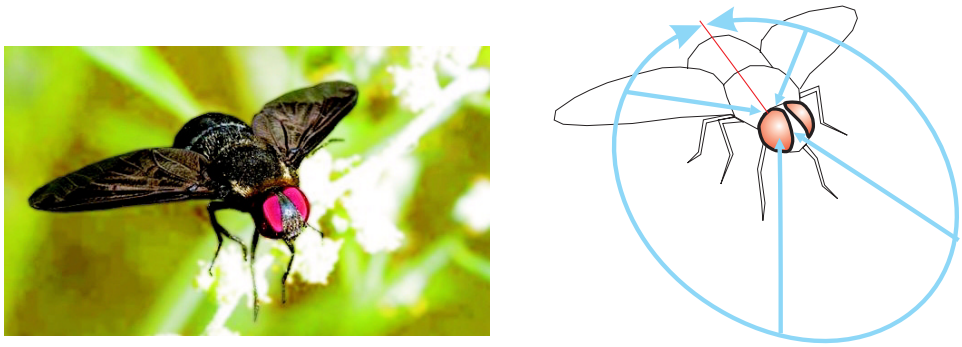


Figure 1: Photograph of a *true fly*. The azimuthal field of view of a *true fly* is about 360^o. *Photograph courtesy of Armando Frazão (http://photo.digitalg.net/).*

up" by an observer as they move through their environment. Thus, Gibson says that perception is *direct*: perception and action are tightly coupled with no need for internal representations. Given the complexity of the world, where not everything can be processed at once, this theory suggests that perception is based on selective attention mechanisms.

In 1982, Marr put forward a computational approach to vision, which was to act as the foundation of modern computer vision. In this work, he proposed a sequential, modular approach to visual perception, where internal representations were a crucial component. In this regard, Marr's *indirect* computational approach differs from Gibson's. In his computational approach, you begin with a *primal sketch* which contains information regarding image regions and boundaries. From this representation, you go on to build a $2\frac{1}{2}D$ *sketch* which specifies, from an observer's viewpoint, an object's orientation and depth. Finally, you build a viewer–independent *3D model* representation of the world.

1.2 Our Approach to Visual Perception

In our approach to visual perception, we follow the indirect approach and construct representations of the world. In this sense, we agree with Marr's mediated approach to perception, although we depart from the sequential building of representations. At no stage do we build full, or partial, 3D representations of the environment for autonomous navigation[1]. Instead, we focus on building representations *suitable* to the task at hand. For instance, when walking along a city avenue, it is sufficient to know our position to within an accuracy of one block. However, when entering our hall door we require much more precise movements. We propose that the internal environmental representations should be tailored to each navigation task, in line with the information perceived from the environment. This is supported by evidence from the biological world, where many animals make alternate use of landmark-based navigation and (approximate) route integration methods [67].

When our robot is required to travel long distances, an appearance-based environmental representation is used to perceive the world. For precise tasks, such as docking or door traversal, perception switches from the appearance-based method to one that relies on

[1]We will show that such representations are useful as a Visual Interface for a human when specifying tasks to a semi–autonomous mobile robot.

image features. We characterize these two modes of operation as: *Topological Navigation* and *Visual Path Following*, respectively.

Combining long-distance/low-precision and short-distance/high-accuracy perception modules plays an important role in finding efficient and robust solutions to the robot navigation problem. This distinction is often overlooked, with emphasis being placed on the construction of world models, rather than concentrating on how these models can be used effectively.

1.3 Human–Robot Interaction

A second aspect of our work is developing user interfaces for robots using (omnidirectional) images. From a perception perspective, our aim is to design an interface where an intuitive link exists between how the user perceives the world and how they control the robot. We hope to achieve this by generating a rich scene description of a remote location. The user is free to rotate and translate this model to specify a particular destination to the robot. Scene modelling, from a single omnidirectional image, is possible with limited user input in the form of co-linearity, co-planarity and orthogonality properties. While humans have an immediate qualitative understanding of the scene encompassing co-planarity and co-linearity properties of a number of points in the scene, robots equipped with an omnidirectional camera can take precise azimuthal and elevation measurements.

In Section 2, we present the modelling and design of omnidirectional cameras, including details of the camera designs we used. In Section 3, we present Topological Navigation and Visual Path Following. We provide details of the different image dewarpings (views) available from our omnidirectional camera: standard, panoramic and bird's–eye views. In addition, we detail geometric scene modelling, model tracking, and appearance–based approaches to navigation. In Section 4, we present our Visual Interface. In all cases, we demonstrate mobile robots navigating autonomously and guided interactively in structured environments. These experiments show that the synergetic design, combining perception modules, navigation modalities and human–robot interaction, is effective in real–world situations. Finally, in Section 5, we present our conclusions and future research directions.

2 Omnidirectional Vision Sensors: Modelling and Design

In 1843 [46], a patent was issued to Joseph Puchberger of Retz, Austria for the first system that used a rotating camera to obtain omnidirectional images. The original idea for the (static camera) omnidirectional vision sensor was initially proposed by Rees in a US patent dating from 1970 [56]. Rees proposed the use of a hyperbolic mirror to capture an omnidirectional image, which could then be transformed to a (normal) perspective image.

Since those early days, the spectrum of application has broadened to include such diverse areas as tele-operation [64, 71], video conferencing [55], virtual reality [45], surveillance [59], 3D reconstruction [25, 61], structure from motion [11] and autonomous robot navigation [27, 70, 68, 76, 78]. For a survey of previous work, the reader is directed to [75]. A relevant collection of papers, related to omnidirectional vision, can be found in [15] and [34].

Omnidirectional images can be generated by a number of different systems which can be classified into four distinct design groupings: Camera-Only Systems; Multi-Camera – Multi-Mirror Systems; Single Camera – Multi-Mirror Systems, and Single Camera – Single Mirror Systems.

Camera-Only Systems: A popular method used to generate omnidirectional images is the rotation of a standard CCD camera about its vertical axis. The captured information, i.e. perspective images (or vertical line scans) are then stitched together so as to obtain panoramic 360° images. Cao *et al.* [9] describe a system that uses a fish-eye lens [47]. Instead of relying upon a single rotating camera, a second camera-only design is to combine cameras pointing in differing directions [22]. Here, images are acquired using inexpensive board cameras and are again stitched together to form panoramas. Finally, Greguss [33] developed a lens, he termed the Panoramic Annular Lens, to capture a panoramic view of the environment.

Multi-Camera – Multi-Mirror Systems: This approach consists of arranging a cluster of cameras in a certain manner along with an equal number of mirrors. Nalwa [49] achieved this by placing four triangular planar mirrors side by side, in the shape of a pyramid, with a camera under each. One significant problem with multi-camera – multi-mirror systems is geometric registering and intensity blending the images together so as to form a seamless panoramic view. This is a difficult problem to solve given that, even with careful alignment, unwanted visible artifacts are often found at image boundaries. These occur not only because of variations between the intrinsic parameters of each camera, but also because of imperfect mirror placement.

Single Camera – Multi-Mirror Systems: The main goal behind the design of single camera – multi-mirror systems is compactness. Single camera – multi-mirror systems are also known as Folded Catadioptric Cameras [52]. A simple example of such a system is that of a planar mirror placed between a light ray travelling from a curved mirror to a camera, thus "folding" the ray. Bruckstein and Richardson [8] presented a design that used two parabolic mirrors, one convex and the other concave. Nayar [52] used a more general design consisting of any two mirrors with a conic-section profile.

Single Camera – Single Mirror Systems: In recent years, this system design has become very popular; it is the approach we chose for application to visual-based robot navigation. The basic method is to point a CCD camera vertically up, towards a mirror.

There are a number of mirror profiles that can be used to project light rays to the camera. The first, and by far the most popular design, uses a **standard mirror profile**: planar, conical, elliptical, parabolic, hyperbolic or spherical. All of the former, with obvious exception of the planar mirror, can image a 360° view of the environment horizontally and, depending on the type of mirror used approximately 70° to 120°, vertically. Some of the mirror profiles, yield simple projection models. In general, to obtain such a system it is necessary to place the mirror at a precise location relative to the camera. In 1997, Nayar and Baker [50] patented a system combining a parabolic mirror and a telecentric lens, which is well described by a simple model and simultaneously overcomes the requirement of precise assembly. Furthermore, their system is superior in the acquisition of non-blurred images.

The second design involves specifying a **specialised mirror profile** in order to obtain a particular, possibly *task-specific*, view of the environment. In both cases, to image the greatest field-of-view the camera's optical axis is aligned with that of the mirrors'. A detailed analysis of both the standard and specialised mirror designs are given in the following Sections.

2.1 A Unifying Theory for Single Centre of Projection Systems

Recently, Geyer and Daniilidis [30, 31] presented a unified projection model for all omnidirectional cameras *with* a single centre of projection. They showed that these systems (parabolic, hyperbolic, elliptical and perspective[2]) can be modelled by a two-step mapping via the sphere. This mapping of a point in space to the image plane is graphically illustrated in Figure 2 (left). The two steps of the mapping are as follows:

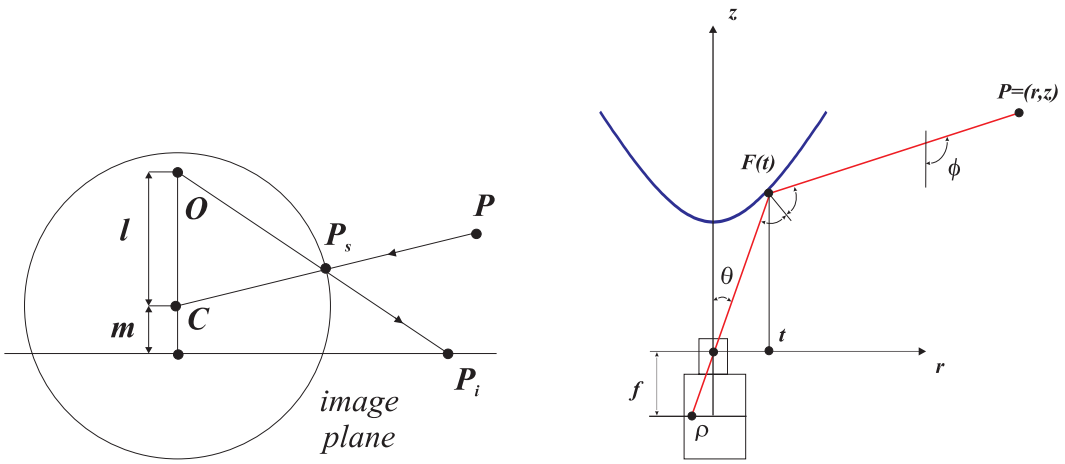


Figure 2: A Unifying Theory for all catadioptric sensors *with* a single centre of projection (left). Main variables defining the projection model of non-single projection centre systems based on arbitrary mirror profiles, $F(t)$ (right).

1. Project a 3D world point, $\mathbf{P} = (x, y, z)$ to a point $\mathbf{P_s}$ on the sphere surface, such that the projection is normal to the sphere surface.

2. Subsequently, project *to* a point on the image plane, $\mathbf{P_i} = (u, v)$ from a point, $\mathbf{O}$ on the vertical axis of the sphere, through the point $\mathbf{P_s}$.

The mapping is mathematically defined by Equation 1:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{l+m}{l \cdot r - z} \begin{bmatrix} x \\ y \end{bmatrix}, \text{where } r = \sqrt{x^2 + y^2 + z^2} \tag{1}$$

As one can clearly see, this is a two-parameter, (l and m) representation, where l represents the distance from the sphere centre, $\mathbf{C}$ to the projection centre, $\mathbf{O}$ and m the distance from $\mathbf{O}$ to the image plane. Modelling the various catadioptric sensors with a single centre of projection is then just a matter of varying the values of l and m in Equation 1. As an example, to model a parabolic mirror, we set $l = 1$ and $m = 0$. Then the image plane passes through the sphere centre, $\mathbf{C}$ and $\mathbf{O}$ is located at the north pole of the sphere. In this case, the second projection is the well known stereographic projection. We note here that the Unifying Theory can model standard perspective cameras (i.e. the pinhole model) when $l = 0$ and $m = 1$. In this case, $\mathbf{O}$ converges to $\mathbf{C}$ and the image plane is located at the south pole of the sphere.

[2]A parabolic mirror with an orthographic lens and all of the others with a standard lens. In the case of a perspective camera, the mirror is virtual and planar.

2.2 Model for Non-Single Projection Centre Systems

Non-single projection centre systems cannot be represented exactly by the unified projection model. One such case is an omnidirectional camera based on an spherical mirror. The intersections of the projection rays incident to the mirror surface, define a continuous set of points distributed in a volume[2], unlike the unified projection model where they all converge to a single point. In the following, we derive a projection model for non-single projection centre systems.

The image formation process is determined by the trajectory of rays that start from a 3D point, reflect on the mirror surface and finally intersect with the image plane. Considering first order optics [37], the process is simplified to the trajectory of the principal ray. When there is a single projection centre it immediately defines the direction of the principal ray starting at the 3D point. If there is no single projection centre, then we must first find the reflection point at the mirror surface.

In order to find the reflection point, a system of non-linear equations can be derived which directly gives the reflection and projection points. Based on first order optics [37], and in particular on the reflection law, the following equation is obtained:

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

where θ is the camera's vertical view angle, ϕ is the system's vertical view angle, F denotes the mirror shape (it is a function of the radial coordinate, t) and F' represents the slope of the mirror shape. See Figure (2 (right).

Equation (2) is valid both for single [30, 1, 77, 62], and non-single projection centre systems [10, 38, 13, 27]. When the mirror shape is known, it provides the projection function. For example, consider the single projection centre system combining a parabolic mirror, $F(t) = t^2/2h$ with an orthographic camera [51], one obtains the projection equation, $\phi = 2atan(t/h)$ relating the (angle to the) 3D point, ϕ and an image point, t.

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

$$\theta = atan\left(\frac{t}{F}\right) \quad , \quad \phi = atan\left(-\frac{r-t}{z-F}\right). \tag{3}$$

The solution of the system of Equations (2) and (3) gives the reflection point, (t, F) and the image point $(f.t/F, f)$ where f is the focal length of the lens.

2.3 Design of Standard Mirror Profiles

Omnidirectional camera mirrors can have standard or specialised profiles, $F(t)$. In standard profiles the form of $F(t)$ is known, we need only to find its parameters. In the specialised profiles the form of $F(t)$ is also a degree of freedom to be derived numerically. Before detailing the design methodology, we introduce some useful properties.

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

Proof: from Eq.(2) we see that the maximum vertical view angle, ϕ is obtained when t/F and F' are maximums. Since both of these values are monotonically increasing, then the maximum of ϕ is obtained at the maximal t, i.e. $t = t_M$.

□

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

$$t_2 \doteq \alpha t \quad \wedge \quad F_2(t_2) \doteq \alpha F(t). \tag{4}$$

This scaling preserves the geometrical property:

Property 2 ***(Scaling)*** *Given a catadioptric camera with a pin-hole at* $(0, 0)$ *and a mirror profile* $F(t)$*, which is a* C_1 *function, the* vertical view angle is invariant *to the system scaling defined by Eq.(4).*

Proof: we want to show that the vertical view angles are equal at corresponding image points, $\phi_2(t_2/F_2) = \phi(t/F)$ which, from Eq.(2), is the same as comparing the corresponding derivatives $F_2'(t_2) = F'(t)$ and is demonstrated using the definition of the derivative:

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

□

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

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

Standard mirror profiles are parametric functions and hence implicitly define the design parameters. Our goal is to specify a large vertical field of view, ϕ given the limited field of view of the lens, θ. In the following we detail the designs of cameras based on spherical and hyperbolic mirrors, which are the most common standard mirror profiles.

Cameras based on spherical and hyperbolic mirrors, respectively, are described by the mirror profile functions:

$$F(t) = L - \sqrt{R^2 - t^2} \quad \text{and} \quad F(t) = L + \frac{a}{b}\sqrt{b^2 + t^2} \tag{5}$$

where R is the spherical mirror radius, (a, b) are the major and minor axis of the hyperbolic mirror and L sets the camera to mirror distance (see Fig.3). As an example, when $L = 0$ for the hyperbolic mirror, we obtain the omnidirectional camera proposed by Chahl and Srinivasan's [10]. Their design yields a constant gain mirror that linearly maps 3D vertical angles into image radial distances.

Chahl and Srinivasan's design does not have the single projection centre property, which is obtained placing the camera at one hyperboloid focus, i.e. $L = \sqrt{a^2 + b^2}$, as Baker and Nayar show in [1] (see Fig.3 (right). In both designs the system is described just by the two hyperboloid parameters, a and b.

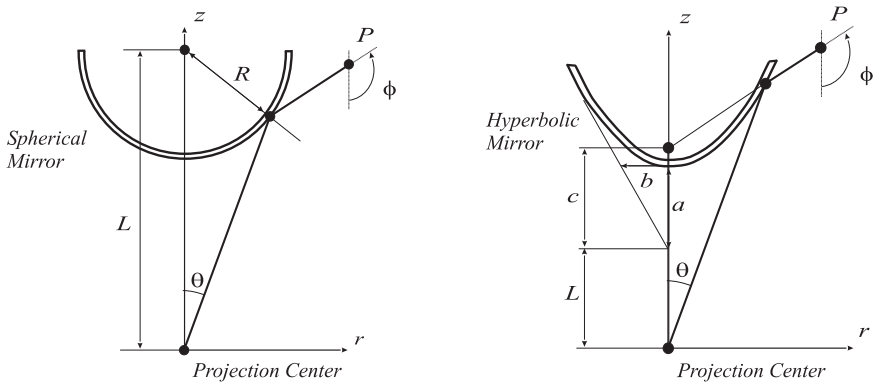


Figure 3: Catadioptric Omnidirectional Camera based on a spherical (left) or an a hyperbolic mirror (right). In the case of a hyperbolic mirror $L = 0$ or $L = c$ and $c = \sqrt{a^2 + b^2}$.

In order to design the spherical and hyperbolic mirrors, we start by fixing the focal length of the camera, which directly determines the view field θ. Then the maximum vertical view field of the system, ϕ, is imposed with the reflection law Eq.(2). This gives the slope of the mirror profile at the mirror rim, F'. Stating, without loss of generality, that the mirror rim has unitary radius (i.e. $(1, F(1))$ is a mirror point), we obtain the following non-linear system of equations:

$$\begin{cases} F(1) = 1/\tan\theta \\ F'(1) = \tan\left(\phi - \theta\right)/2 \end{cases} . \tag{6}$$

The mirror profile parameters, (L, R) or (a, b), are embedded in $F(t)$, and are therefore found solving the system of equations.

Since there are minimal focusing distances, D_{min} which depend on the particular lens, we have to guarantee that $F(0) \geq D_{min}$. We do this applying the scaling property (Eq.(4)). Given the scale factor $k = D_{min}/F(0)$ the scaling of the spherical and hyperbolic mirrors is applied respectively as $(R, L) \leftarrow (k.R, k.L)$ and $(a, b) \leftarrow (k.a, k.b)$. If the mirror is still too small to be manufactured then an additional scaling up may be applied. The camera self-occlusion becomes progressively less important when scaling up.

Figure 4: Omnidirectional camera based on a spherical mirror (left), camera mounted on a Labmate mobile robot (middle) and omnidirectional image (right).

Figure 4 shows an omnidirectional camera based on a spherical mirror, built in house for the purpose of conducting navigation experiments. The mirror was designed to have

a view field of 10^o above the horizon line. The lens has $f = 8mm$ (vertical view field, θ is about $\pm 15^o$ on a $6.4mm \times 4.8mm$ CCD). The minimal distance from the lens to the mirror surface was set to $25cm$. The calculations indicate a spherical mirror radius of $8.9cm$.

2.4 Design of Constant Resolution Cameras

Constant Resolution Cameras, are omnidirectional cameras that have the property of linearly mapping 3D measures to imaged distances. The 3D measures can be either elevation angles, vertical or horizontal distances (see Fig.5). Each linear mapping is achieved by specializing the mirror shape.

Some constant resolution designs have been presented in the literature, [10, 38, 13, 30] with a *different derivation* for each case. In this section, we present a *unified* approach that encompasses all the previous designs and allows for new ones. The key idea is to separate the equations for the reflection of light rays at the mirror surface and the mirror *Shaping Function*, which explicitly represents the linear projection properties to meet.

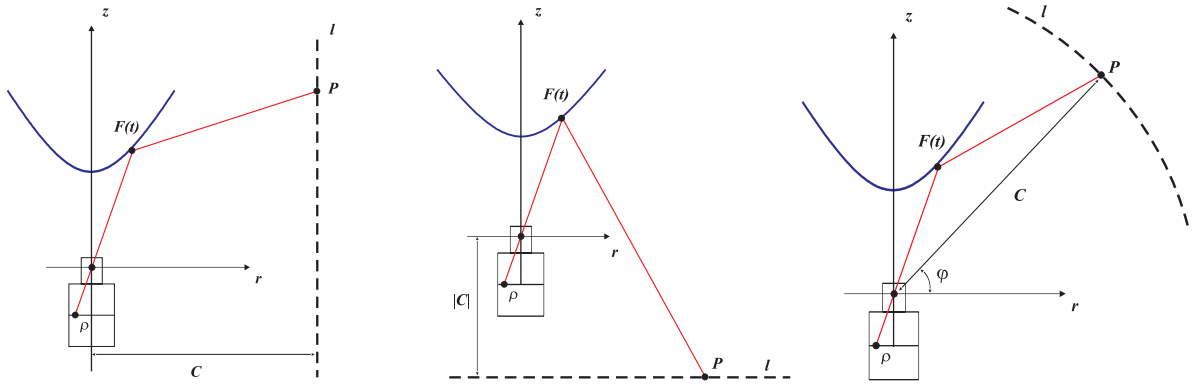


Figure 5: Constant vertical, horizontal and angular resolutions (respectively left, middle and right schematics). Points on the line l are linearly related to their projections in pixel coordinates, ρ.

The Mirror Shaping Function

Combining the equations that describe the non-single projection centre model (Eqs. (2) and (3)) and expanding the trigonometric functions, one obtains an equation of the variables t, r, z encompassing the mirror shape, F and slope, F':

$$\frac{\frac{t}{F} + 2\frac{F'}{1-F'^2}}{1 - 2\frac{tF'}{F(1-F'^2)}} = -\frac{r-t}{z-F} \tag{7}$$

This is Hicks and Bajcsy's differential equation relating 3D points, (r, z) to the reflection points, $(t, F(t))$ which directly imply the image points, $(t/F, 1)$ [38]. We assume without loss of generality that the focal length, f is 1, since it is easy to account for a different (desired) value at a later stage.

Equation 7 allows to design a mirror shape, $F(t)$ given a desired relationship between 3D points, (r, z) and the corresponding images, $(t/F, 1)$. In order to compute $F(t)$, it is convenient to have the equation in the form of an explicit expression for F' [3]. Re-arranging

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

Eq.(7) results in the following second order polynomial equation:

$$F'^2 + 2\alpha\, F' - 1 = 0 \tag{8}$$

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

$$\alpha = \frac{-(z-F)\,F + (r-t)\,t}{(z-F)\,t + (r-t)\,F} \tag{9}$$

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

Concluding, to obtain the mirror profile first we specify the shaping function, Eq.(9) and then solve Eq.(8), or simply integrate:

$$F' = -\alpha \pm \sqrt{\alpha^2 + 1} \tag{10}$$

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

Setting Constant Resolution Properties

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

For constant resolution mirrors, we want some world distances, D, to be *linearly* mapped to (pixel) distances, p, measured in the image sensor, i.e. $D = a_0.p + b_0$ for some values of a_0 and b_0 which mainly determine the visual field.

When considering conventional cameras, pixel distances are obtained by scaling metric distances in the image plane, ρ. In addition, knowing that those distances relate to the slope t/F of the ray of light intersecting the image plane as $\rho = f.\frac{t}{F}$. The linear constraint may be conveniently rewritten in terms of the mirror shape as:

$$D = a.t/F + b \tag{11}$$

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

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

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

The derivation described here follows closely that presented by Gaechter and Pajdla in [23]. The main difference consist of a simpler setting for the equations describing the

Linear Property	Mirror Shaping Function
$z = a.t/F + b$ $r = C$	$\alpha = \frac{-\left(a\frac{t}{F} + b - F\right) F + (C - t)\, t}{\left(a\frac{t}{F} + b - F\right) t + (C - t)\, F}$ (12)
$r = a.t/F + b$ $z = C$	$\alpha = \frac{-(C - F) F + \left(a\frac{t}{F} + b - t\right) t}{(C - F)\, t + \left(a\frac{t}{F} + b - t\right) F}$ (13)
$\varphi = a.t/F + b$ $r = C.cos(\varphi)$ $z = C.sin(\varphi)$	$\alpha = \frac{-\left(C\sin(a\frac{t}{F} + b) - F\right) F + \left(C\cos(a\frac{t}{F} + b) - t\right) t}{\left(C\sin(a\frac{t}{F} + b) - F\right) t + \left(C\cos(a\frac{t}{F} + b) - t\right) F}$ (14)

Table 1: Mirror Shaping Functions for constant vertical, horizontal and angular resolutions.

mirror profile. We start by specialising the linear constraint in Eq.(11) to relate 3D points of a vertical line l with pixel coordinates (see Fig.5). Inserting this constraint into Eq.(9) we obtain the specialised shaping function of Eq.(12).

Hence, the procedure to determine the mirror profile consists of integrating Eq.(10) using the shaping function of Eq.(12), while t varies from 0 to the mirror radius. The initialization of the integration process is done by computing the value of $F(0)$ that would allow the mirror rim to occupy the entire field of view of the sensor.

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

Figure 5 shows how the ground plane, l, is projected onto the image plane. The camera-to-ground distance is represented by $-C$ (C is negative because the ground plane is lower than the camera centre) and r represents radial distances on the ground plane. The linear constraint inserted into Eq.(9) yields a new shaping function (as in Eq.(13)), which after integrating Eq.(10) results in the mirror profile proposed by Hicks and Bajcsy [38].

Constant Angular Resolution - One last case of practical interest is that of obtaining a linear mapping from 3D points spaced by equal angles to equally distant image pixels, i.e. designing a constant angular resolution mirror. Figure 5 shows how the spherical surface with radius C surrounding the sensor is projected onto the image plane. In this case the desired linear property relates angles with image points. Then, placing the constraints into Eq.(9) we finally obtain Eq.(14).

Integrating Eq.(10), using the shaping function just obtained (Eq.(14)), would result in a mirror shape such as the one of Chahl and Srinivasan [10]. The difference is that in our case we are imposing the linear relationship from 3D vertical angles, φ directly to image points, $(t/F, 1)$ instead of angles relative to the camera axis, $atan(t/F)$.

Shaping functions for Log-polar Sensors - Log-polar cameras are imaging devices that have a spatial resolution inspired by the human-retina. Unlike standard cameras, the resolution is not constant on the sensing area. More precisely, the density of the pixels is higher in the centre and decays logarithmically towards the image periphery. The organisation of the pixels also differs from the standard cameras, as a log-polar camera consists of a set of concentric circular rings, each one with a constant number of pixels. Advantageously, combining a log-polar camera with a convex mirror results in an omnidirectional imaging device where the panoramic views are extracted directly due to the polar arrangement of the sensor.

In a log-polar camera, the relationship of the linear distance, ρ, measured on the sensor's surface and the corresponding pixel coordinate, p, is specified by $p = \log_k(\rho/\rho_0)$, where ρ_0 and k stand for the fovea radius and the rate of increase of pixel size towards the periphery.

As previously stated, our goal consists of setting a linear relationship between world distances (or angles), D and corresponding (pixel) distances, p. Combining into the linear relationship the perspective projection, $\rho = t/F$ and the logarithmic law of the log-polar camera, results in the following constraint:

$$D = a.\log(t/F) + b \tag{15}$$

The only difference in the form of the linear constraint when using conventional or log-polar cameras, Eqs. (11) and (15), is that the slope t/F is replaced by its logarithm. Hence, replacing the slope by its log directly in Eqs. (12), (13) and (14), results in the desired shaping functions for the log-polar camera.

Concluding, we obtained a design methodology of constant resolution omnidirectional cameras, that is based on a shaping function whose specification allows us to choose a particular linear property. This methodology generalises a number of published design methods for specific linear properties. For example the constant vertical resolution design results in a sensor equivalent to that of Gaechter et al [23]. Of particular interest is a constant angular resolution sensor, that is an implementation of a spherical sensor providing a constant number of pixels per solid angle. This is similar to Conroy and Moore's design [13], but with the difference that, due to the nature of the log-polar camera, we do not need to compensate for lesser pixels when moving closer to the camera axis.

Figure 6 shows an omnidirectional based on the prototype log-polar camera Svavisca [43]. The mirror is a combined design, encompassing constant vertical and horizontal resolutions, respectively, in the outer and in the two inner annular regions. Vertical and ground patterns in the real world are use to test for linear properties. The panoramic image results from a direct read out of the sensor and the bird's eye views are obtained after a change from polar to cartesian coordinates. In the panoramic image, the vertical sizes of black squares are equal to those of the white squares, thus showing linearity from 3D measures to image pixel coordinates. In the bird's eye views the rectilinear pattern of the ground was successfully recovered.

2.5 The Single Centre of Projection Revisited

A question related to the use of non-single centre of projection sensors is how different they are from single projection centre ones? What is the degree of error induced by a locus of viewpoints? We have studied this problem using the catadioptric sensor with a spherical mirror [25]. As outlined in Section 2.1, the Unifying Theory covers all catadioptric sensors with a single centre of projection. A projection model governing a catadioptric sensor with

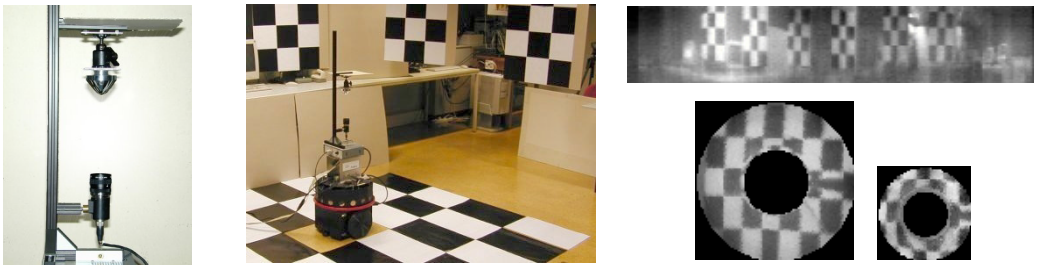


Figure 6: Svavisca camera equipped with the combined mirror (left) and world scene with regular patterns distributed vertically and over the floor (middle). Panoramic and bird's eye views (right). The bird's eye views have a transformation from cartesian to polar coordinates. The bird's eye view at right originated from the fovea area.

a generic mirror profile is given in Section 2.2. If the Unifying Theory can approximate a non-single centre of projection camera, one would expect that - using both models - the error between projecting 3D points to the image plane would be small. It turns out that for real-world points further than 2m away from the catadioptric sensor the error in the image plane is less than *1 pixel*.

Derrin and Konolige [18] also approximated a single centre of projection but used a concept they termed *iso-angle mapping*. They constructed a virtual system by displacing all incoming rays, each having a unique Euler angle, so as they converged at a single point. Thus, their method produced a camera with a single centre of projection, imaging a distorted scene. Since they did not derive an analytical expression for the distortion, it was measured as a change in the height of a small object, given a change in its elevation angle and remained less than 2.5%.

Concluding, many omnidirectional vision systems, despite not having a single projection centre, can be accurately described by a single projection centre model. In this way models based on the single projection centre property may become the most common, in the same way as the pin-hole model is used for standard cameras even when it is just an approximation valid for the tasks at hand.

3 Environmental Perception for Navigation

Traditionally, localisation has been identified as a principal perceptual component of the navigation system of a mobile robot [44]. This drove continuous research and development on sensors providing direct localisation measurements.

There is a large variety of self-localisation solutions available [4] in the literature. However, in general they are characterised by a hard and limiting tradeoff between robustness and cost. As paradigmatic and extreme examples we can refer to solutions based on artificial landmarks (beacons) and those based on odometry. Solutions based on beacons are robust but expensive in terms of the materials, installation, maintenance or configuration to fit a specific new purpose. The solutions based on odometry are inexpensive, but since they rely on the integration of the robot's internal measurements, i.e. not grounded to the world, errors accumulate over time.

We use vision to sense the environment as it allows navigation to be regulated by the world. In particular, we have noted the advantages of omnidirectional vision for navigation, including its flexibility for building environmental representations. Our robot combines two main navigation modalities: Visual Path Following and Topological Navi-

gation. In Visual Path Following, the short-distance / high-accuracy navigation modality, the orthographic view of the ground plane is a convenient world model as it makes simple representing / tracking ground plane features and computing the pose of the robot. Panoramic views are a complementary representation, which are useful in the identification and extraction of vertical line features. These types of views are easily obtained from omnidirectional cameras using image dewarpings.

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

In summary, both Visual Path Following and Topological Navigation rely upon environmental perception (self-localisation) for regulating the movement. The main point here is that perception is linked to internal representations of the world which are chosen according to the tasks at hand. We will now detail *Geometrical Representations* for precise self-localisation, necessary for Visual Path Following, and *Topological Representations* for global positioning leading, necessary for Topological Navigation.

3.1 Geometric Representations for Precise Self-Localisation

Robot navigation in cluttered or narrow areas, such as when negotiating a door traversal, requires precise self-localisation in order to be successful. In other words, the robot has to be equipped with precise environmental perception capabilities.

Vision-based self-localisation derives robot poses from images. It encompasses two principal stages: image processing and pose-computation. Image processing provides the tracking of features in the scene. Pose-computation is the geometrical calculation to determine the robot pose from feature observations, given the scene model.

Designing the image processing level involves modelling the environment. One way to inform a robot of an environment is to give it a CAD model, as in the work of Kosaka and Kak [42], recently reviewed in [19]. The CAD model usually comprises metric values that need to be scaled to match the images acquired by the robot. In our case, we overcome this need by defining geometric models composed of features of the environment directly extracted from images.

Omnidirectional cameras based on standard mirror profiles, image the environment features with significant geometrical distortion. For instance, a corridor appears as an image band of variable width and vertical lines are imaged radially. Omnidirectional images must therefore be dewarped in order to maintain the linearity of the imaged 3D straight lines.

Pose-computation, as the robot moves in a plane, consists of estimating a 2D position and an orientation. Assuming that the robot knows fixed points in the environment (landmarks) then there are two main methods of self-localisation relative to the environment: trilateration and triangulation [4]. Trilateration is the determination of a vehicle's position based on distance measurements to the landmarks. Triangulation has a similar purpose but is based on bearing measurements.

In general, a single image taken by a calibrated camera provides only bearing measurements. Thus, triangulation is a more "natural" way to calculate self-localisation. However, there are some camera poses / geometries that provide more information. For example, a bird's eye view image (detailed in the following subsection) provides an orthographic

view of the ground plane, providing simultaneous observation of bearings and distances to floor landmarks. Given distances and bearings, the pose-computation is simplified to the calculation of a 2D rigid transformation.

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

Image Dewarpings for Scene Modelling

Images acquired with an omni-directional camera, e.g. based on a spherical or hyperbolic mirror, are naturally distorted. Knowing the image formation model, we can correct some distortions to obtain Panoramic or Bird's Eye Views.

The panoramic view groups together, in each scan line, the projections of all visible points, at a constant angle of elevation. The bird's eye view is a scaled orthographic projection of the ground plane. These views are advantageous e.g. for extracting and tracking vertical and ground plane lines.

Panoramic and Bird's Eye Views are directly obtained by designing custom shaped mirrors. An alternative approach, as described next, is to simply dewarp the omnidirectional images to the new views.

Panoramic View: 3D points at the same elevation angle from the axis of the catadioptric omnidirectional vision sensor, project to a 2D circle in the image. Therefore, the image dewarping is defined simply as a cartesian to polar coordinates change:

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

where (u_0, v_0) is the image centre, α and R are the angle and radial coordinates. The steps and range of α and R are chosen according to the resolution, and covering all the effective area, of the omnidirectional image. One rule for selecting the step of α is to make the number of columns of the panoramic image equal to the perimeter of the middle circle of the omnidirectional image. Hence inner circles are over-sampled and outer circles are sub-sampled. This rule gives a good tradeoff between data loss due to sub-sampling and memory consumption for storing the panoramic view.

Bird's Eye View: In general, 3D straight lines are imaged as curves in the omnidirectional image. For instance, the horizon line is imaged as a circle. Only 3D lines that belong to vertical planes containing camera and mirror axis project as straight (radial) lines.

In order to dewarp an omnidirectional image to a bird's eye view, notice that the azimuthal coordinate of a 3D point is not changed by the imaging geometry of the omnidirectional camera. Therefore, the dewarping of an omnidirectional image to a bird's eye view is a radial transformation. Hence, we can build a 1D look up table relating a number of points at different radial distances in the omnidirectional image and the respective real distances. The 1D look up table is the radial transformation to be performed for all directions on an omnidirectional image in order to obtain the bird's eye view.

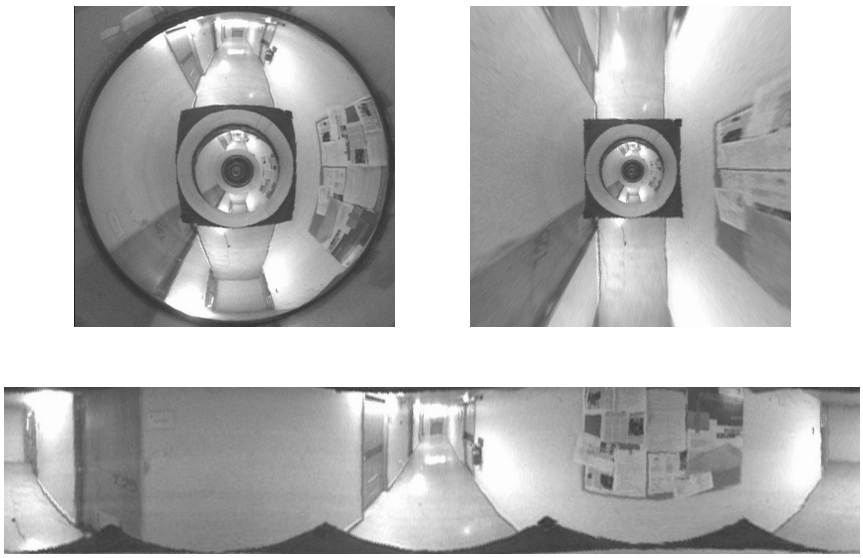


Figure 7: Image dewarping for bird's eye and panoramic views. (Top-left) original omnidirectional image, (top-right) bird's eye view and (bottom) panoramic view.

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

$$\rho_{img} = \mathcal{P}_\rho(\rho_{ground}, \vartheta) \tag{16}$$

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

Figure 7 illustrates the dewarpings of an omnidirectional image to obtain the Bird's Eye and Panoramic Views. Notice that the door frames are imaged as vertical lines in the Panoramic view and the corridor guidelines are imaged as straight lines in the Bird's Eye view, as desired.

As a final remark, notice that our process to obtain the look up table encoding the Bird's Eye View, is equivalent to performing calibration. However, for our purposes a good dewarping is simply the one that makes ground plane straight lines appear straight in the Bird's Eye View.

As long as the mirror, camera and support (mobile platform) remain fixed to each other, the dewarpings for panoramic and bird's eye views are time invariant and can be programmed with 2D lookup tables. The dewarpings are done efficiently in this way.

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

Geometric Scene Modelling and Model Tracking

Geometric models of the scene are collections of segments identified in Bird's Eye and Panoramic views [4]. Ground segments are rigidly interconnected in the Bird's Eye views while vertical segments will vary their locations according to the viewer location. Considering both types of segments, the models are "wire-frames" whose links change according to the viewpoint.

Each scene model must have a minimal number of features (line segments) in order to allow self-localisation. One line of the ground plane permits finding only the orientation of the robot and gives a single constraint on its localisation. Two concurrent ground lines, or one ground and one vertical, already determine robot position and orientation. Given three lines either all vertical, one on the ground, two on the ground (not parallel) or three on the ground (not all parallel), always permit us to compute the pose and therefore form valid models [5].

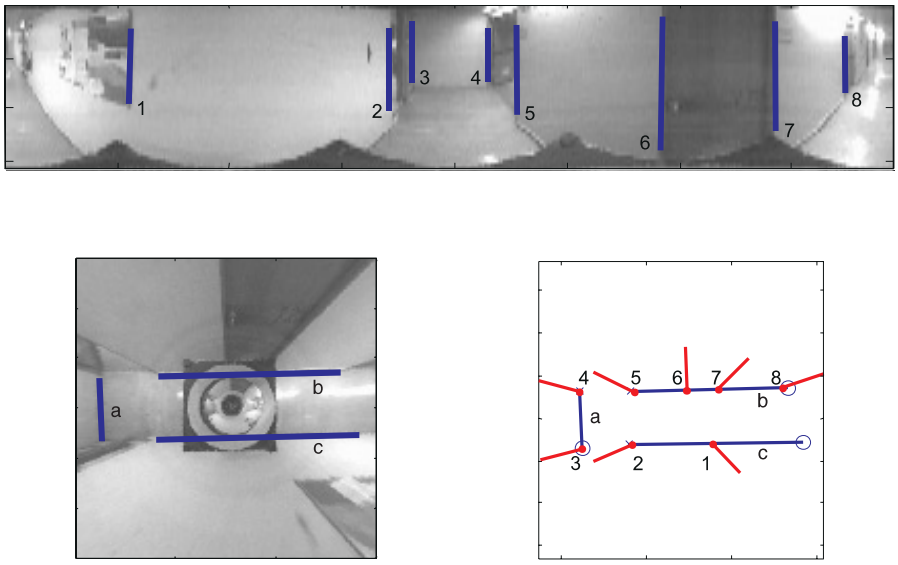


Figure 8: Geometric models for a door crossing experiment. The segments composing the model (bottom right) are illustrated in the panoramic and bird's eye view images, respectively (top and bottom left).

Figure 8 shows one example of modelling the scene using line segments observed directly in the scene. The model is composed of three ground lines, two of which are corridor guidelines, and eight vertical segments essentially defined by the door frames. A single door frame (i.e. two vertical lines) and one corridor guideline would suffice but it is beneficial to take more lines than minimally required in order to improve the robustness of self-localisation.

In order to represent a certain scene area, and to meet visibility[6] and quality criteria, a minimal number of segments are required. Models characterising different world regions are related by rigid 2D transformations. These transformations are firstly defined between every two neighbour models at locations where both models are (partially but with enough

[4]Despite the fact that localisation can be based on tracked image corners [58], more robust and stable results are obtained with line segments as noted for example by Spetsakis and Aloimonos in [60].

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

[6]see Talluri and Aggarwal in [63] for a geometrical definition of visibility regions

relevance) visible. Navigation is therefore possible in the area composed as a union of individual areas, provided by each individual model.

Assuming that the robot pose evolves smoothly over time, the model segments need to be detected only once – at the initialisation stage. From then on, we need only *track* them, which is much more efficient in computational terms. We track both edges lying on the ground plane and vertical edge segments, using respectively the bird's eye and panoramic views (details in [28]).

Pose computation

The self-localisation procedure is based on the tracking of the geometric models. The tracking of the models requires rigidity of the world structure (but naturally not rigidity of the observed model segments themselves).

A simple method of calculating pose from the models arises when the segments of the model intersect at ground points (as in the model shown in figure 8). In this case, the model, despite encompassing ground and vertical segments, is simplified to the case of a set of ground points. This set of points moves rigidly in the Bird's Eye View, and therefore self-localisation is in essence the computation of the 2D transformation tracking the movement of the points. This method requires intersecting segments, which is similar to tracking corners but in a much more stable manner. This is especially true when dealing with long segments, as the noise in the orientation of small segments may become significant, affecting the computation of the intersections and the quality of corner estimates.

Alternatively, localisation is achieved through an optimisation procedure, namely minimizing the distance between model and observed line segments, directly at the pose parameters. Intuitively, the best pose estimate should align the scene model and the observed lines as well as possible. This is computationally more expensive, but more robust to direction errors on the observed line segments [26].

Defining pose as $\mathbf{x} = [x\ y\ \theta]$ and the distance between the segments ab and cd as $d(cd, ab) = f(c-a, b-a) + f(d-a, b-a)$ where a, b, c, d are the segment extremal points and f is the normalised internal product, $f(\mathbf{v}, \mathbf{v_0}) = \left|\mathbf{v}^T.\mathbf{v_0^\perp}\right| / \left\|\mathbf{v_0^\perp}\right\|$, the problem of pose estimation based on the distance between model and observed segments can be expressed by the minimization of a cost functional:

$$\mathbf{x}^* = \arg_{\mathbf{x}} \min \sum_i d(s_i, s_{0i}(\mathbf{x})) \tag{17}$$

where s_i stands for observed vertical and ground line segments, and s_{0i} indicates the model segments (known a priori). The minimization is performed with a generic gradient descent algorithm provided that the initialisation is close enough. For the initial guess of the pose there are also simple solutions such as using the pose at the previous time instant or, when available, an estimate provided by e.g. a 2D rigid transformation of ground points or by a triangulation method [3].

The self-localisation process as described by Eq.(17), relies exclusively on the observed segments, and looks for the best robot pose justifying those observations on the image plane. Despite the optimization performed for pose-computation, there are residual errors that result from the low-level image processing, segment tracking, and from the method itself. Some of these errors may be recovered through the global interpretation of the current image with the a priori geometric model. Since the model is composed of segments associated with image edges, we want to maximize the sum of gradients, ∇I at every point of the model wire-frame, $\{P_i\}$. Denoting the pose by $\mathbf{x}$ then the optimal pose $\mathbf{x}^*$ is obtained

as:

$$\mathbf{x}^* = \arg_{\mathbf{x}} \max \mu(\mathbf{x}) = \arg_{\mathbf{x}} \max \sum_i |\nabla I\left(\mathcal{P}(P_i; \mathbf{x})\right)| \tag{18}$$

where $\mathcal{P}$ is the projection operator and $\mu(\mathbf{x})$ represents the (matching) merit function. Given an initial solution to Eq.(17), the final solution can be found by a local search on the components of $\mathbf{x}$.

Usually, there are model points that are non-visible during some time intervals while the robot moves. This is due, for example, to camera (platform) self-occlusion or to the finite dimensions of the image. In these cases, the merit matching merit function does not smoothly evolve with pose changes: it is maximized by considering the maximum number of points possible, instead of the true segment pose. Therefore, we include a smoothness prior to the function. One solution is to maintain the gradient values at control points of the model for the images when they are not visible.

Visual Path Following

Visual Path Following can be described in a simple manner as a trajectory following problem, without having the trajectory explicitly represented in the scene. The trajectory is only a data structure learnt from example / experience or specified through a visual interface.

Visual Path Following combines the precise self-localisation (detailed in the preceding sections) with a controller that generates the control signals for moving the robot, such as that proposed by de Wit et al [16].

Experiments were conducted using an omnidirectional camera with a spherical mirror profile (shown in Fig.4), mounted on a TRC labmate mobile robot. Figure 9 illustrates tracking and self-localization while traversing a door from the corridor into a room. The tracked features (shown as black circles) are defined by vertical and ground-plane segments, tracked in bird's eye view images.

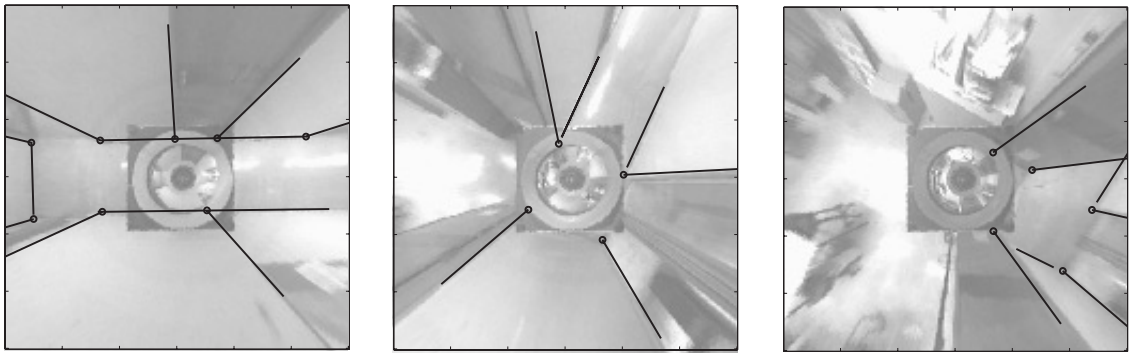
Currently, the user initializes the relevant features to track. To detect the loss of tracking during operation, the process is continuously self-evaluated by the robot, based on gradient intensities obtained within specified areas around the landmark edges (Eq.18). If these gradients decrease significantly compared to those expected, a recovery mechanism is launched.

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

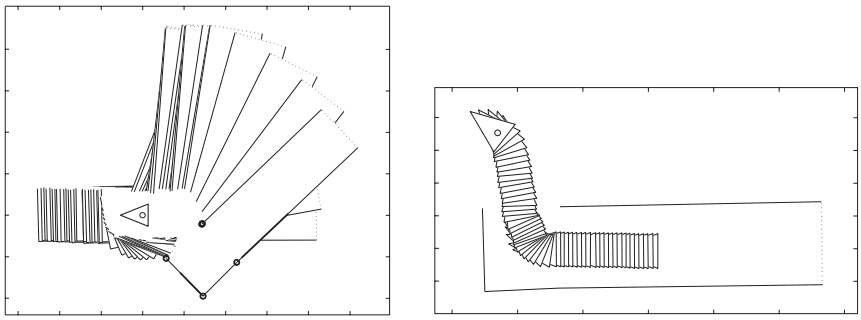
3.2 Topological Representations

A topological map is used to describe the robot's global environment and obtain its qualitative position when travelling long distances. It is represented as a graph: *nodes* in the graph correspond to landmarks, i.e. distinctive places such as corners. *Links* connect nodes and correspond to environmental structures that can be used to control the pose of the robot. In order to effectively use this graph the robot must be able to travel along a corridor, recognize the ends of a corridor, make turns, identify and count door frames. These behaviours are implemented through an appearance based system and a visual servoing strategy.

An appearance based system [48] is one in which a run–time image is compared to a database set for matching purposes. For example, in our corridor scene, the appearance



(a) Robot in corridor *(b) turning to the door* *(c) after crossing the door*



(d) Scene-model tracking *(e) Self-localisation*

Figure 9: Feature tracking at three instants (a,b,c), scene-model tracking in the robot coordinate system (d) and the self-localisation result obtained by fixing the tracked scene-model (e).

based system provides qualitative estimates of robot position and recognizes distinctive places such as corner or door entrances.

Therefore, the topological map is simply a collection of inter-connected images. To go from one particular locale to another, we do not have to think in precise metric terms. For example, to move the robot from one corner to the opposite one we may indicate to the robot to follow one corridor until the first corner and then to follow the next corridor until the next corner, thus reaching the desired destination, or to complete more complex missions such as "*go to the third office on the left-hand side of the second corridor*".

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

When compared to geometric approaches, topological maps offer a parsimonious representation of the environment, are highly computationally efficient [65], scale easily and can explicitly represent uncertainties in the real world [6].

3.3 Image Eigenspaces as Topological Maps

In general, sizeable learning sets are required to map the environment and so matching using traditional techniques, such as correlation, would incur a very high computational cost. If one considers the images as points in space, it follows that they shall be scattered throughout this space, *only* if they differ significantly from one other. However, many real-world environments (offices, highways etc.) exhibit homogeneity of structure, leading to

a large amount of redundant information within the image set. Consequently, the images are not scattered throughout a high dimensional space but – due to their similarity – lie in a lower dimensional subspace.

We implement dimensionality reduction using the classical procedure of *Principal Component Analysis* (PCA)[7], as described by Murase and Nayar in [48], and detailed by Winters in [73] or Gaspar, Winters and Santos-Victor in [27]. Simply put, Principal Component Analysis **reduces the dimensionality** of a set of linearly independent input variables, while still accurately representing most of the original data. The reconstruction of this original data is optimal in the sense that the mean square error between it and the original data is minimized.

Imagine that we represent images as L-dimensional vectors in $\mathbb{R}^L$. Due to the similarity between images (data redundancy) these vectors will not span the entire space of $\mathbb{R}^L$ but rather, they will be confined (or close, to be more precise) to a lower-dimensional subspace, $\mathbb{R}^M$ where $M << L$. Hence, to save on computation, we can represent our images by their co-ordinates in such a lower-dimensional subspace, rather than using all of the pixel information. Each time a new image is acquired, its capture position can easily be determined by projecting it into the lower-dimensional subspace and finding its closest match from the *a priori* set of points (images).

A basis for such a linear subspace can be found through PCA, where the basis vectors are denominated **Principal Components**. They can be computed as the **eigenvectors** of the **covariance matrix** of the normalised set of images acquired by the robot. The number of eigenvectors that can be computed in such a way is the same as the number of images in the input data, and the eigenvectors are the same size as the images.

Each reference image is associated with a *qualitative* robot position (e.g. half way along the corridor). To find the robot position in the topological map, we have to determine the reference image that best matches the current view. The distance between the current view and the reference images can be computed directly using their projections (vectors) on the lower dimensional eigenspace. The distance is computed between M-dimensional coefficient vectors (typically 10 to 12), as opposed to image size vectors (128×128). The position of the robot is that associated with the reference image having the lowest distance.

When using intensity images, comparison of images is essentially a sum of squared differences of brightness (radiance) values and because of this the robot is prone to miscalculating its location where *large non-uniform* deviations in illumination occur. This can be overcome by using edge images, although these are not robust to edge-point position errors. The solution therefore, is to compare shapes instead of edge-points, or more specifically the *distance between shapes* present in both images. There are several possible definitions of the distance between shapes. Two very well known are chamfer distance and the the Hausdorff distance.

Localisation Based on the Chamfer Distance

The chamfer distance is based on the correlation of a template edge-image with a *distance transformed image*. The distance transform of an edge-image is an image of the same size as the original, that indicates at each point the distance to the closest edge point [5, 29, 14].

The *chamfer distance transform*[8] is computed from an edge-image using the forward

[7] It is sometimes known as the application of the *Karhunen-Loève transform* [53, 66].

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

and backward masks shown in figure 10 [5, 29]. There are various possible values for the constants in the masks. We use the values according to Montanari's metric [14].

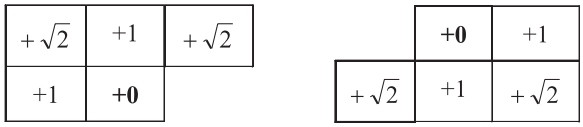


Figure 10: Forward and backward masks for computing the distance transform. The element in bold face indicates the centre of the mask.

The constants shown in the masks are added to each of the local values and the resulting value of the mask computation is the minimum of the set. Both masks are applied along the rows of the initialised image.

Figure 11 shows the distance transform of the edges of an omnidirectional image. We remove the inner and outer parts of the omnidirectional image as they contain artifact edges, i.e. edges not related to the scene itself, created by the mirror rim and the robot plus camera self-occlusion.

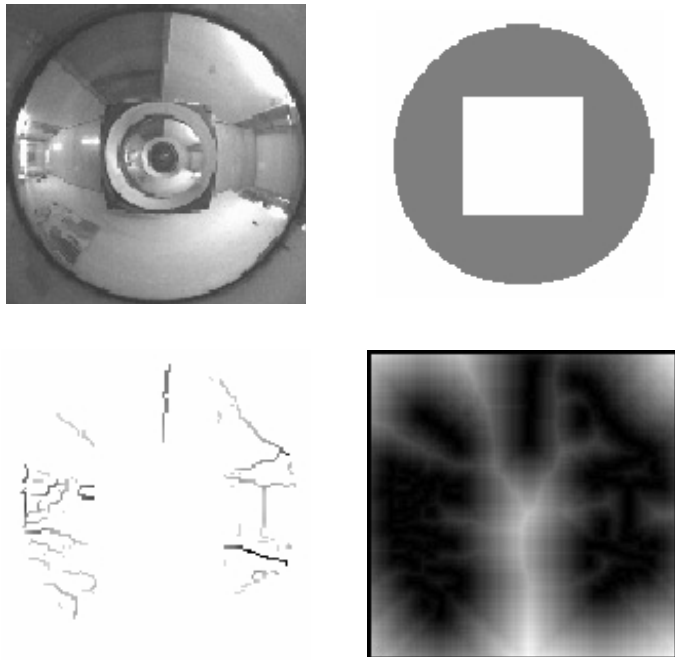


Figure 11: Distance transform: (top) original omnidirectional image and mask showing in grey the region of interest, (bottom) edges found in the region of interest and the distance transform of the edge-image.

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

$$d(D,T) = \frac{\sum_{i,j} D_{ij} T_{ij}}{\sum_{i,j} T_{ij}} \tag{19}$$

where T is a template edge-image and D is the distance transform of the edges of the current run-time image. As weaker edges (small gradient magnitudes) are more susceptible

to noise, we set T_{ij} to the gradient magnitudes of the template images, instead of binary edges. Hence, we give more weight to the strongest edges.

Equation 19 says that the *chamfer distance* is an average of the distances between corresponding points of the shapes. In a strict sense, it is an approximation as the underlying *chamfer distance transform* is itself an approximation to the Euclidean distance. In practice this difference is not relevant as typically the shapes to compare are at similar poses and the distances between the points are small enough to make negligible the difference of the chamfer and the Euclidean distances.

In the topological localisation application, we want to find the database image corresponding to the current run-time image. In order to find the best matching we search the database using the chamfer distance as the comparison measure. The comparison of images is done from an edge-image to a distance transformed edge-image. The distance transformation may be applied either to the run time or to the database images [29]. We apply the distance transform to the run time edge-images and leave to the template edge-images the role of selecting the relevant edge locations.

The distance as defined by Eq.(19) is zero for perfectly matching images. Therefore we search for the image matching the current image I_m in a set $T_1 \dots T_n$ by minimizing Eq.(19),

$$\hat{n} = \arg_n \min d\left(D(I_m), T_n\right). \tag{20}$$

Notice that, unlike recognition applications such as pedestrian and sign detection in an image [29], in the localisation application the template and run-time images have equal sizes. The search parameter is an image index instead of translation, rotation and scaling. The range of the index is the size of the database.

Usually there are a large number of database images, and thus undertaking localisation, as in Eq.(20), is computationally expensive. However it only needs to be performed once, when the robot is dropped-in-scene. During normal operation there is a causality constraint along the consecutive locations. We reduce the search range to a window around the last location, typically ± 5 images.

3.4 Eigenspace approximation to the Hausdorff fraction

The Hausdorff distance [57] (of which the Hausdorff fraction is a subset) is a technique whereby one can measure the distance between two sets of points, in our case edge images. A number of Hausdorff distance measures are defined by the following equations:

$$H(A, B) = \max(h(A, B), h(B, A)) \tag{21}$$

where

$$h(A, B) = \max_{a \in A} \min_{b \in B} \parallel a - b \parallel \tag{22}$$

Here A and B represent sets of points. $h(A, B)$ measures the distance from each point in A to the nearest point in B and the maximum distance is termed the *directed* distance from A to B, and is the normal choice for critical time dependent systems.

The Hausdorff distance is very sensitive to even a single outlying point in one of the shapes. The Generalised Hausdorff distance, defined by Huttenlocher et al in [39], is thus proposed as a similar measure but that is robust to partial occlusions. The generalised Hausdorff distance is an f^{th} quantile of the distances between all the points of one shape to the corresponding points of the other shape. The quantile is chosen according to the expected noise and occlusion levels.

In recognition applications, the generalised Hausdorff distance is further specialised to save computational power. The Hausdorff fraction, the measure we are interested in, instead of measuring a distance between shapes evaluates the percentage of superposition when one of the shapes is dilated. Furthermore, for computational efficiency, principal components analysis is included resulting in an eigenspace approximation to the Hausdorff fraction [40].

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

$$\hbar(I_m, I_n^d) = \frac{I_m^T I_n^d}{\|I_m\|^2} \tag{23}$$

An image, I_k can be represented in a low dimensional eigenspace [48, 72] by a coefficient vector, $\mathcal{C}_k = [c_1^k, \cdots, c_M^k]^T$, as follows:

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

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

$$\widehat{\hbar}(I_m, I_n^d) = \frac{C_m^T C_n^d + I_m^T \bar{I} + I_n^{dT} \bar{I} - \|\bar{I}\|^2}{\|I_m\|^2}. \tag{24}$$

One important issue, when approximating the Hausdorff fraction, is to include some tolerance in matching step. Huttenlocher et al. [40] build the eigenspace using both dilated and un-dilated model views and pre-process the run time edge images to dilate the edges. In our pre-processing we use low pass filtering instead of edge dilation. The purpose here is to maintain the local maxima of gradient magnitude at edge points while enlarging the matching area. We found this to be a good tradeoff between matching robustness and accuracy.

To test this view-based approximation we collected a sequence of images, acquired at different times, 11am and 5pm, near a large window. Figure 12 shows the significant changes in illumination, especially near the large window at the bottom left hand side of each omnidirectional image. Even so, the view based approximation can correctly determine that the unknown image shown in Figure 12(a) was closest to the database image shown in Figure 12(b), while PCA based on brightness distributions would fail. For completeness, Figure 12 (c) and (d) shows a run-time edge image and its corresponding retrieved image using the eigenspace approximation to the Hausdorff fraction.

Integrating Topological Navigation and Visual Path Following

When continuously operating, the mobile robot is usually performing topological navigation. At some points of the mission the navigation modality is required to change to the visual path following. Thus, the robot needs to retrieve the scene features (straight lines in our case) chosen at the time of learning to specific this particular visual path following task.

The search for the features can be approached as a general pattern matching problem using e.g. a generalised Hough transform as in [74, 21]. We approach the problem by coordinating the two navigation modalities. To find the features, the uncertainty of the

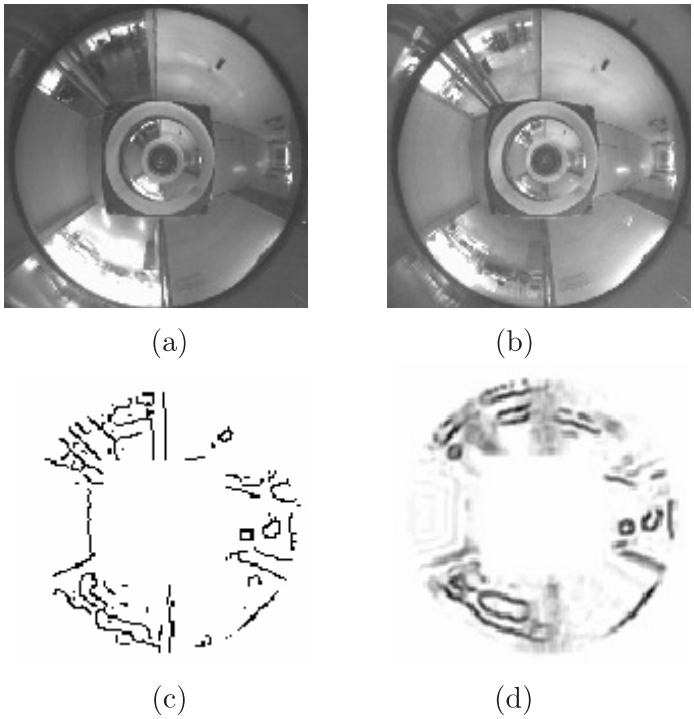


Figure 12: (a) An omnidirectional image obtained at 11:00, (b) one obtained at 17:00; (c) An edge-detected image and (d) its retrieved image.

location of the robot is controlled by using more detailed topological maps and by increasing the searching regions of the features otherwise bounded according to the maximum speed of the robot.

During system initialisation, the robot will normally begin at a known docking place and the undocking visual path following task may be immediately elicited. Of course, the robot may have to start at an unknown (within the topological map) position, i.e. a drop-in-scene case. Should this occur, then self-localisation is performed using the topological localisation module.

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

3.5 Topological Localisation Results

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

The experiments are based on three sequences of images: one database sequence describing the environment and two run-time sequences acquired along a fraction of the represented environment. One of the run time sequences was acquired at a time of the day different to the database set, resulting therefore in very different lighting conditions.

Experiment 1: the run time sequence, as compared to the database, is acquired

under similar illumination conditions, the length of the traversed path is about 50% of the original and the images are acquired at a different sampling frequency (distance between consecutive images). Figure 13 shows that the three methods give similar localisation results, as desired. The small differences among the methods are due to the distinct image database (appearance set) construction techniques. The figure shows that in this experiment the three methods, despite compressing information, preserve enough detail to distinguish each image relative to all the others.

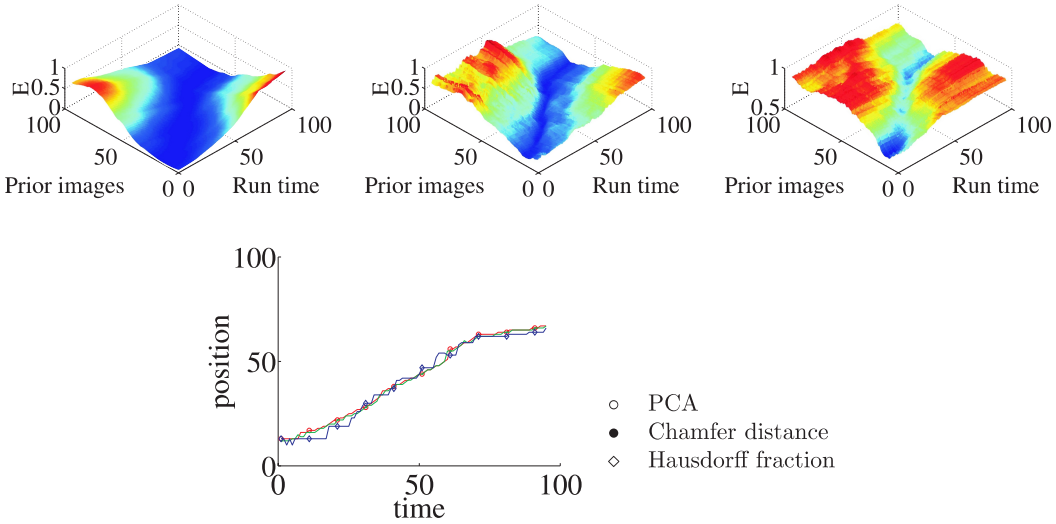


Figure 13: Three methods of topological localisation, (top, from left to right): localisation based on PCA, Chamfer distance and Hausdorff distance. The clear valleys show that there is enough information to distinguish the robot locations. (Bottom) localisation as found by each of the methods i.e. ordinates corresponding to minimum values found at each time instant on the 3D plots.

Experiment 2: figure 14 shows topological localisation performed by each of the methods for two sequences taken in the same path but at different times of the day, resulting in very different lighting conditions. As expected, the PCA-based method, i.e. the one using brightness values directly , fails to obtain correct locations in areas of large non-uniform illumination change (i.e. the last part of the test). The other two methods, which are based on edges, obtain better results.

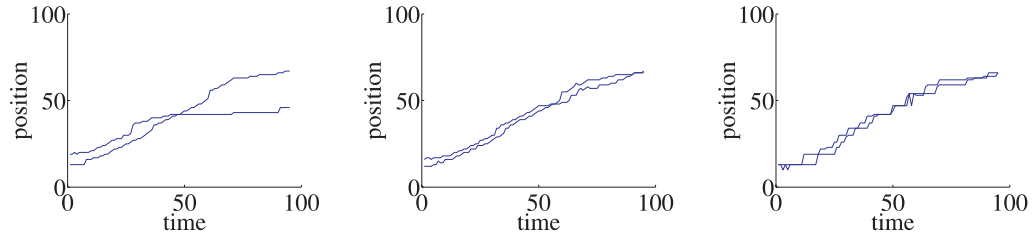


Figure 14: Topological localisation experiments using the three methods over two sequences acquired under different lighting conditions. From left to right: localisation based on PCA, Distance transform and Hausdorff distance.

As expected, the edges based methods are more suited to dealing with very different illuminations. In our navigation experiments we use mainly the PCA over brightness

values, as most of our scenario is not subject to large illumination changes, and using brightness values is more informative than using only edges. For the parts of the scene where illumination can change significantly we use the Hausdorff based method. The reason of its choice when compared to the Distance transform, is that it is faster for the first localisation at dropped-in-scene situations.

Integrated Navigation Experiments

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

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

The visual feedback is provided by the omnidirectional camera. We use *bird's eye views* of the floor, which simplifies the servoing task, as these images are a scaled orthographic projection of the ground plane (i.e. no perspective effects). Figure 15 shows a top view of the corridor guidelines, the robot and the trajectory to follow in the centre of the corridor.

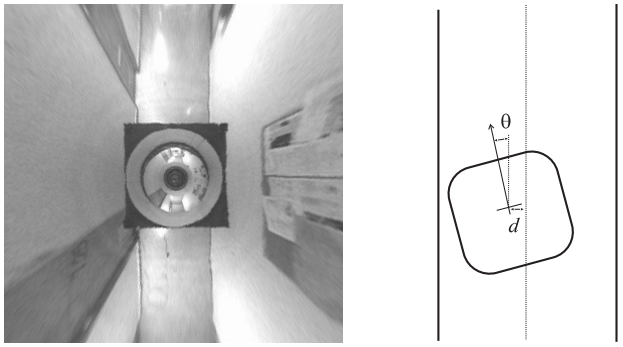


Figure 15: (Left) Bird's eye view of the corridor. (Right) Measurements used in the control law: the robot heading θ and distance d relative to the corridor centre. The controller is designed to regulate to zero the (error) measurements actuating on the angular and linear speeds of the robot.

From the images we can measure the robot heading with respect to the corridor guidelines and the distance to the central reference trajectory. We use a simple kinematic planner to control the robot's position and orientation in the corridor, using the angular velocity as the single degree of freedom.

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

Hence, the robot is equipped to perform Topological Navigation relying on appearance based methods and on its corridor following behaviour. This is a methodology for traversing long paths. For local and precise navigation the robot uses Visual Path Following as detailed in section 3.1. Combining these behaviours the robot can perform missions covering extensive areas while achieving local precise missions. In the following we describe one such mission.

The mission starts in the Computer Vision Lab. Visual Path Following is used to navigate inside the Lab, traverse the Lab's door and drive the robot out into the corridor.

Once in the corridor, control is transferred to the Topological Navigation module, which drives the robot all the way to the end of the corridor. At this position a new behaviour is launched, consisting of the robot executing a 180° turn, after which the topological navigation mode drives the robot back to the Lab entry point.

Figure 16: Experiment combining visual path following for door traversal and topological navigation for corridor following.

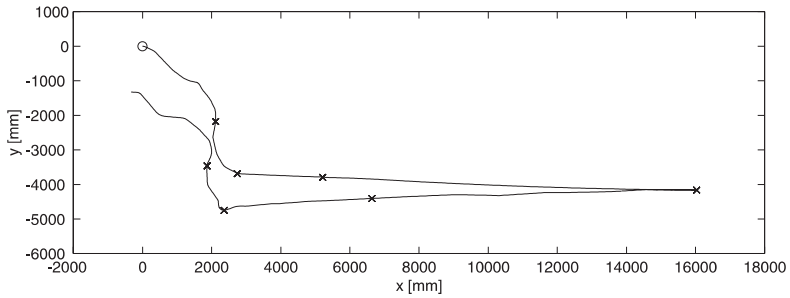
During this backward trajectory we use the same image eigenspaces as were utilised during the forward motion by simply rotating, in real-time, the acquired omnidirectional images by 180° . Alternatively, we could use the image's power spectrum or the Zero Phase Representation [54]. Finally, once the robot is approximately located at the lab entrance, control is passed to the Visual Path Following module. Immediately it locates the visual landmarks and drives the robot through the door. It follows a pre-specified path until the final goal position, well inside the lab, is reached. Figure 16 shows an image sequence to relate the robot's motion during this experiment.

In Figure 17(a) we used odometric readings from the best experiment to plot the robot trajectory. When returning to the laboratory, the uncertainty in odometry was approximately 0.5m. Thus, door traversal would not be possible without the use of visual control. Figure 17(b), shows the actual robot trajectory, after using ground truth measurements to correct the odometric estimates. The mission was successfully accomplished.

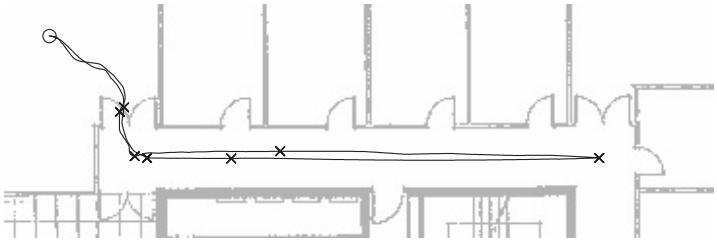
This integrated experiment shows that omnidirectional images are advantageous for navigation and support different representations suitable both for Topological Maps, when navigating between distant environmental points, and Visual Path Following for accurate path traversal. Additionally, we have described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

4 Complementing Human and Robot Perceptions for HR Interaction

Each omnidirectional image provides a rich description and understanding of the scene. Visualization methods based on panoramic or bird's eye views provide a simple and effective way to control the robot. For instance, the robot heading is easily specified by the user by simply clicking on the desired direction of travel in the panoramic image, and the desired (x, y) locations are specified by clicking in the bird's-eye view.



(a)



(b)

Figure 17: A real world experiment combining Visual Path Following for door traversal and Topological Navigation for long-distance goals. Odometry results before (a) and after (b) the addition of ground truth measurements.

Using 3D models further improves the visualization of the scene. A unique feature of such a representation is that the user can tell the robot to arrive to a given destination at a certain orientation simply by rotating the 3D model. Beyond the benefits of immersion, it allows to group the information of many views and get a global view of the environment.

In order to build the 3D scene models, we propose *Interactive Scene Reconstruction*, a method based on the complimentary nature of Human and Robot perceptions. While Humans have an immediate qualitative understanding of the scene encompassing co-planarity and co-linearity properties of a number of points of the scene, Robots equipped with omnidirectional cameras can take precise azimuthal and elevation measurements.

Interactive scene reconstruction has recently drawn lots of attention. Debevec et al in [17], propose an interactive scene reconstruction approach for modelling and rendering architectural scenes. They derive a geometric model combining edge lines observed in the images with geometrical properties known a priori. This approach is advantageous relative to building a CAD model from scratch, as some information comes directly from the images. In addition, it is simpler than a conventional structure from motion problem because, instead of reconstructing points, it deals with reconstructing scene parameters, which is a much lower dimension and better conditioned problem.

In [61] Sturm uses an omnidirectional camera based on a parabolic mirror and a telecentric lens for reconstructing a 3D scene. The user specifies relevant points and planes grouping those points. The directions of the planes are computed e.g. from vanishing points, and the image points are back-projected to obtain parametric representations where the points move on the 3D projection rays. The points and the planes, i.e. their distances to the viewer, are simultaneously reconstructed by minimizing a cost functional based on the distances from the points to the planes.

We build 3D models using omnidirectional images and some limited user input, as in Sturm's work. However our approach is based on a different reconstruction method and the omnidirectional camera is a generalised single projection centre camera modelled by the Unified Projection Model [30]. The reconstruction method is that proposed by Grossmann for conventional cameras [36], applied to single projection centre omnidirectional cameras for which a back-projection model was obtained.

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

4.1 Interactive Scene Reconstruction

We now present the method for interactively building a 3D model of the environment. The 3D information is obtained from co-linearity and co-planarity properties of the scene. The texture is then extracted from the images to obtain a realistic virtual environment.

The 3D model is a Euclidean reconstruction of the scene. As such, it may be translated and rotated for visualization and many models can be joined into a single representation of the environment.

As in other methods [41, 61], the reconstruction algorithm presented here works in structured environments, in which three orthogonal directions, "x", "y" and "z" shape the scene. The operator specifies in an image the location of 3D points of interest and indicates properties of alignment and planarity. In this section, we present a method based on [35].

In all, the information specified by the operator consists of :

- Image points corresponding to 3D points that will be reconstructed, usually on edges of the floor and of walls.
- Indications of "$x-$", "$y-$" and "z =constant" planes as and of alignments of points along the x, y and z directions. This typically includes the floor and vertical walls.
- Indications of points that form 3D surfaces that should be visualized as such.

The remainder of this section shows how to obtain a 3D reconstruction from this information.

Using Back-projection to form Perspective Images

In this section, we derive a transformation, applicable to single projection centre omnidirectional cameras that obtain images as if acquired by perspective projection cameras. This is interesting as it provides a way to utilize methodologies for perspective cameras directly with omnidirectional cameras. In particular, the interactive scene reconstruction method (described in the following sections) follows this approach of using omnidirectional cameras transformed to perspective cameras.

The acquisition of correct perspective images, independent of the scenario, requires that the vision sensor be characterised by a single projection centre [2]. The unified projection model has, by definition, this property but, due to the intermediate mapping over the sphere, the obtained images are in general not perspective.

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

The back-projection of an image pixel (u, v), obtained through spherical projection, yields a 3D direction $k \cdot (x, y, z)$ given by the following equations derived from Eq.(1):

$$a = (l + m), b = (u^2 + v^2)$$
$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{la - sign(a)\sqrt{a^2 + (1 - l^2)b}}{a^2 + b} \begin{bmatrix} u \\ v \end{bmatrix} \tag{25}$$
$$z = \pm\sqrt{1 - x^2 - y^2}$$

where z is negative if $|a| / l > \sqrt{b}$, and positive otherwise. It is assumed, without loss of generality, that (x, y, z) is lying on the surface of the unit sphere. Figure 18 illustrates the back-projection. Given an omnidirectional image we use back-projection to map image points to the surface of a sphere centred at the camera viewpoint [9].

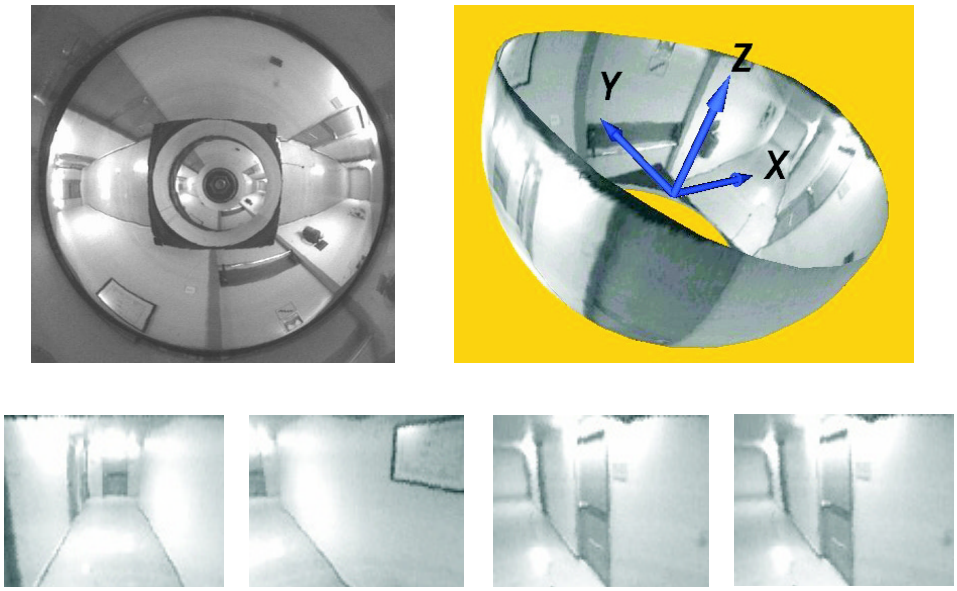


Figure 18: (Top) original omnidirectional image and back-projection to a spherical surface centred at the camera viewpoint. (Below) Examples of perspective images obtained from the omnidirectional image.

At this point, it is worth noting that the set $\{(x, y, z)\}$ interpreted as points of the projective plane, already define a perspective image. Rotation and scaling of the set $\{(x, y, z)\}$, allows to obtain specific viewing directions and focal lengths. Denoting the transformation of coordinates from the omnidirectional camera to a desired (rotated) perspective camera by R then the new perspective image $\{(u, v, 1)\}$ becomes:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \lambda K R \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{26}$$

[9] The omnidirectional camera utilized here is based on a spherical mirror and therefore does not have a single projection centre. However, as the scene depth is large as compared to the sensor size, the sensor approximates a single projection centre system (details in [25]). Hence it is possible to find the parameters of the corresponding unified projection model system and use Eq.(25).

where K contains intrinsic parameters and λ is a scaling factor. This is the pin-hole camera projection model [20], when the origin of the coordinates is the camera centre.

Figure 18 shows some examples of perspective images obtained from the omnidirectional image. The perspective images illustrate the selection of the viewing direction.

Aligning the data with the reference frame

In the reconstruction algorithm we use the normalised perspective projection model [20], as indicated by Eqs. (25) and (26):

$$p = \lambda RP \tag{27}$$

in which $p = [u\ v\ 1]^T$ is the image point, in homogeneous coordinates and $P = [P_x\ P_y\ P_z]^T$ is the 3D point.

The rotation matrix R is chosen to align the camera frame with the reference (world) frame. Since the z axis is vertical, the matrix R takes the form :

$$R = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{28}$$

where θ is the angle formed by the x axis of the camera and that of the world coordinate system. This angle will be determined from the vanishing points [12] of these directions.

A vanishing point is the intersection in the image of the projections of parallel 3D lines. If one has the images of two or more lines parallel to a given 3D direction, it is possible to determine its vanishing point [61].

In our case, information provided by the operator allows for the determination of alignments of points along the x and y directions. It is thus possible to compute the vanishing points of these directions and, from there, the angle θ between the camera and world coordinate systems.

Reconstruction Algorithm

Having determined the projection matrix R in Eq. (27), we proceed to estimate the position of the 3D points P. This will be done by using the image points p to linearly constrain the unknown quantities.

From the projection equation, one has $p \times RP = 0_3$, which is equivalently written

$$S_p RP = 0_3, \tag{29}$$

where S_p is the Rodrigues matrix associated with the cross product with vector p.

Writing this equation for each of the N unknown 3D points gives the linear system:

$$\begin{bmatrix} S_{p_1}R & & & \\ & S_{p_2}R & & \\ & & \ddots & \\ & & & S_{p_N}R \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_N \end{bmatrix} = A.\mathcal{P} = 0_{3N}. \tag{30}$$

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

Since only two equations from the set defined by Eq. (29) are independent, the co-rank of A is equal to the number of points N. The indeterminacy in this system of equations corresponds to the unknown depth at which each points lies, relatively to the camera.

This indeterminacy is removed by the planarity and alignment information given by the operator. For example, when two points belong to a z = constant plane, their z coordinates are necessarily equal and there is thus a *single* unknown quantity, rather than *two*. Equation (30) is modified to take this information into account by replacing the columns of A (resp. rows of $\mathcal{P}$) corresponding to the two unknown z coordinates by a single column (resp. row) that is the sum of the two. Alignment information likewise states the equality of two pairs of unknowns.

Each item of geometric information provided by the user is used to transform the linear system in Equation (30) into a smaller system involving only *distinct* quantities :

$$A'\mathcal{P}' = 0_{3N}. \tag{31}$$

This system is solved in the total least-squares [32] sense by assigning to $\mathcal{P}'$ the singular vector of A' corresponding to the smallest singular value. The original vector of coordinates $\mathcal{P}$ is obtained from $\mathcal{P}'$ by performing the inverse of the operations that led from Eq. (30) to Eq. (31).

The reconstruction algorithm is easily extended to the case of multiple cameras. The orientation of the cameras is estimated from vanishing points as above and the projection model becomes :

$$p = \lambda(RP - Rt) \tag{32}$$

where t is the position of the camera. It is zero for the first camera and is one of $t_1 \dots t_j$ if j additional cameras are present.

Considering for example that there are two additional cameras and following the same procedure as for a single image, similar A and $\mathcal{P}$ are defined for each camera. The problem has six new degrees of freedom corresponding to the two unknown translations t_1 and t_2 :

$$\left[\begin{array}{ccc|c|c} A_1 & & & & \\ & A_2 & & -A_2.\mathbf{1}_2 & \\ & & A_3 & & -A_3.\mathbf{1}_3 \end{array}\right] \left[\begin{array}{c} \mathcal{P}_1 \\ \mathcal{P}_2 \\ \mathcal{P}_3 \\ t_1 \\ t_2 \end{array}\right] = 0 \tag{33}$$

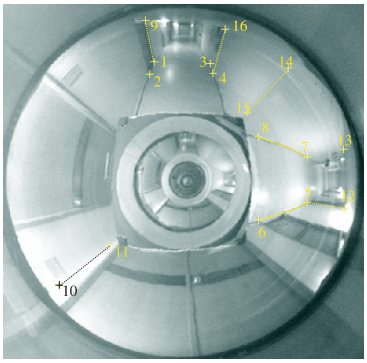
where $\mathbf{1}_2$ and $\mathbf{1}_3$ are matrices to stack the blocks of A_2 and A_3.

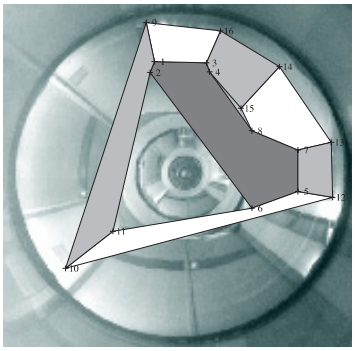
As before co-linearity and co-planarity information is used to obtain a reduced system. Note that columns corresponding to different images may be combined, for example if a 3D point is tracked or if a line or plane spans multiple images. The reduced system is solved in the total least-squares sense and the 3D points P are retrieved as in the single-view case. The detailed reconstruction method is given in [35].

Results

Our reconstruction method provides estimates of 3D points in the scene. In order to visualise these estimates, facets are added to connect some of the 3D points, as indicated by the user. Texture is extracted from the omnidirectional images and a complete textured 3D model is obtained.

Figure 19 shows an omnidirectional image and the superposed user input. This input consists of the 16 points shown, knowledge that sets of points belong to constant x, y or z planes and that other sets belong to lines parallel to the x, y or z axes. The table on the side of the images shows all the user-defined data. Planes orthogonal to the x and y axes are in light gray and white respectively, and one horizontal plane is shown in dark gray (the topmost horizontal plane is not shown as it would occlude the other planes).



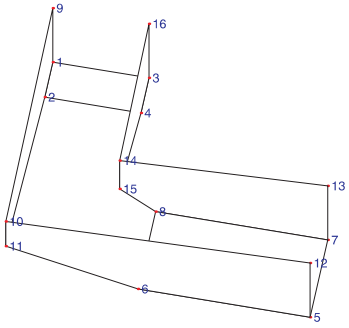


Planes orthogonal to the x axis:
$(1, 2, 9, 10, 11)$,
$(3, 4, 14, 15, 16)$,
$(5, 7, 12, 13)$

Planes orthogonal to the y axis:
$(5, 6, 10, 11, 12)$,
$(7, 8, 13, 14, 15)$,
$(1, 3, 9, 16)$

Planes orthogonal to the z axis:
$(1, 2, 3, 4, 5, 6, 7, 8)$,
$(9, 12, 13, 16)$

Line parallel to the x axis:
$(1, 2)$



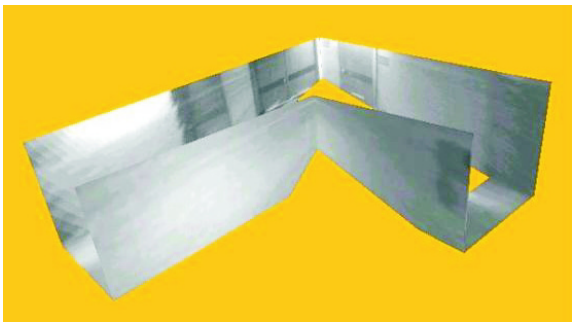


Figure 19: Interactive modelling based on co-planarity and co-linearity properties using a single omnidirectional image. (Top) Original image with superposed points and lines localised by the user. Planes orthogonal to the x, y and z axis are shown in light gray, white, and dark gray respectively. (Table) The numbers are the indexes shown on the image. (Below) Reconstruction result and view of the textured mapped 3D model.

Figure 19 shows the resulting texture-mapped reconstruction. This result shows the effectiveness of omnidirectional imaging to visualize the immediate vicinity of the sensor. It is interesting to note that just a few omnidirectional images are sufficient for building the 3D model (the example shown utilized a single image), as opposed to a larger number of "normal" images that would be required to reconstruct the same scene [41, 61].

4.2 Human Robot Interface based on 3D World Models

Now that we have the 3D scene model, we can build the Human Robot interface. In addition to the local headings or poses, the 3D model allows us to specify complete missions. The human operator selects the start and end locations in the model, and can indicate points of interest for the robot to undertake specific tasks. See figure 20.

Given that the targets are specified on interactive models, i.e. models built and used on the user side, they need to be translated as tasks that the robot understands. The translation depends on the local world models and navigation sequences the robot has in its database.

Most of the world that the robot knows is in the form of a topological map. In this case the targets are images that the robot has in its image database. The images used to build the interactive model are nodes of the topological map. Thus, a fraction of a distance on an interactive model is translated as the same fraction on a link of the topological map.

At some points there are precise navigation requirements. Many of these points are

identified in the topological map and will be invoked automatically when travelling between nodes. Therefore, many of the Visual Path Following tasks performed do not need to be explicitly defined by the user. However, should the user desires, he may add new Visual Path Following tasks. In that case, the user chooses landmarks, navigates in the interactive model and then asks the robot to follow the same trajectory.

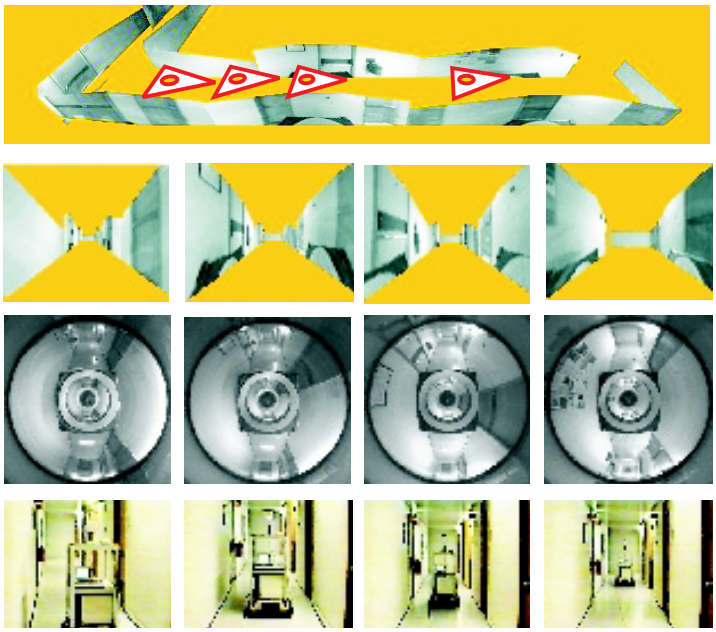


Figure 20: Tele-operation interface based on 3D models: (top) tele-operator view, (middle) robot view and (bottom) world view.

Interactive modelling offers a simple procedure for building a 3D model of the scene where a vehicle may operate. Even though the models do not contain very fine details, they can provide the remote user of the robot with a sufficiently rich description of the environment. The user can instruct the robot to move to desired position, simply by manipulating the model to reach the desired view point. Such simple scene models can be transmitted even with low bandwidth connections.

5 Conclusion

This chapter addressed the problem of mobile robot perception in the context of navigation. We presented a complete navigation system with a focus on building, in line with Marr's theory, mediated perception modalities. We addressed fundamental design issues associated with this goal; namely sensor design, environmental representations, navigation control and user interaction.

A number of omnidirectional vision setups were detailed. These included designs using standard and specialized mirror profiles. We identified the convenience of constant resolution profiles for navigation and proposed a novel uniform formalism for their design [24].

The internal representations used by the robot were tailored to the task at hand: Visual Path Following for local and precise navigation and Topological Navigation for traveling long distances [26, 27]. Methods based on image-edges for improving robustness

of Topological Navigation to large, non-linear illumination changes [69] were presented. By clearly separating the nature of the navigation tasks a simple and yet powerful navigation system was obtained [27].

We described a method for interactive reconstruction based on omnidirectional images. It combined a method designed for conventional cameras with the unified back-projection model that we proposed for single projection centre omnidirectional cameras [25]. Interactive Scene Reconstruction was used to build an expedient visual interface, showing how human and robot perceptions of the environment can complement each other for practical target selection.

Discussion

The challenge of developing perception as a key competence of vision-based mobile robots is of fundamental importance to their successful application in the real world. Vision provides information on world structure and compares favourably with other sensors due to the large amount of rich data available. In terms of perception, omnidirectional vision has the additional advantage of providing output views (images) with simple geometries. Our sensors output Panoramic and Bird's Eye views that are images as obtained by cylindrical retinas or pin-hole cameras imaging the ground plane. Panoramic and Bird's Eye views are useful for navigation, namely for servoing tasks, as they make localisation a simple 2D rigid transformation estimation problem. Successful completion of the door crossing experiment, for example, relied on the tracking of features surrounding the sensor. Such experiments are not possible with limited field of view (conventional) cameras. Even cameras equipped with pan-and-tilt mounting would be unable to perform the many separate landmark trackings of our experiments.

Designing navigation modalities for the task at hand is easier and more effective when compared to designing a single complex navigation mode [7]. Therefore, in this work, emphasis was placed on building appropriate representations rather than always relying upon highly accurate information about the environment. The decision to use this representation was partly inspired by the way in which humans and animals model spatial knowledge. Our combined navigation modalities, Visual Path Following and Topological Navigation, constituted an effective approach to tasks containing both short paths to follow with high precision and long paths to follow qualitatively.

Interactive Scene Reconstruction was shown to be an effective method of obtaining 3D scene models, as compared to conventional reconstruction methods. For example, the model of the corridor corner, in Section 4, was built from a single image. This constitutes a very difficult task for automatic reconstruction due to the low texture. These 3D models formed the basis for the human-robot interface. Unlike many other works, a unique feature of this representation was that the user could specify a given destination, at a certain orientation, simply by rotating the 3D model.

When considering the system as a whole, (i) our approach to visual perception was found to be useful and convenient because it provided world-structure information for navigation, tailored to the task at hand, (ii) the navigation modalities fulfilled the purpose of semi-autonomous navigation by providing autonomy while naturally combining with the human-robot interface, (iii) the human-robot interface provided intuitive way to set high level tasks, by combining limited user input with the simple output of the sensor (images).

In conclusion, the goal of this chapter was to work toward building a robot with visual perception capabilities. As suggested by the title, we believe there is a large amount of work still to be done before we have a full and true understanding of perception. We believe

that key challenges can be addressed by building artificial vision systems. In the future our understanding of perception will allow for robots with visual perception systems, robust enough to cope with new and novel environments. Then, as happened with computers, almost every person will have their very own robot, or what we may term the *personal service robot*.

References

[1] S. Baker and S. K. Nayar, *A theory of catadioptric image formation*, Proc. Int. Conf. Computer Vision (ICCV'97), January 1998, pp. 35–42.

[2] ———, *A theory of single-viewpoint catadioptric image formation*, International Journal of Computer Vision **35** (1999), no. 2, 175–196.

[3] M. Betke and L. Gurvits, *Mobile robot localization using landmarks*, IEEE Trans. on Robotics and Automation **13** (1997), no. 2, 251–263.

[4] J. Borenstein, H. R. Everett, and Liqiang Feng, *Navigating mobile robots: Sensors and techniques*, A. K. Peters, Ltd., Wellesley, MA, 1996
(also: Where am I? Systems and Methods for Mobile Robot Positioning,
ftp.eecs.umich.edu/people/johannb/pos96rep.pdf, 1996).

[5] G. Borgefors, *Hierarchical chamfer matching: A parametric edge matching algorithm*, IEEE Transactions on Pattern Analysis and Machine Intelligence **10** (1988), no. 6, 849–865.

[6] R. Brooks, *Visual map making for a mobile robot*, Proc. IEEE Conf. on Robotics and Automation, 1985.

[7] R. A. Brooks, *A robust layered control system for a mobile robot*, IEEE Transactions on Robotics and Automation **2** (1986), 14–23.

[8] A. Bruckstein and T. Richardson, *Omniview cameras with curved surface mirrors*, Proceedings of the IEEE Workshop on Omnidirectional Vision at CVPR 2000 (Hilton Head, SC, USA), June 2000, First published in 1996 as a Bell Labs Technical Memo, pp. 79–86.

[9] Z. L. Cao, S. J. Oh, and E.L. Hall, *Dynamic omni-directional vision for mobile robots*, Journal of Robotic Systems **3** (1986), no. 1, 5–17.

[10] J. S. Chahl and M. V. Srinivasan, *Reflective surfaces for panoramic imaging*, Applied Optics **36** (1997), no. 31, 8275–8285.

[11] P. Chang and M. Herbert, *Omni-directional structure from motion*, Proceedings of the 1st International IEEE Workshop on Omni-directional Vision (OMNIVIS'00) at CVPR 2000 (Hilton Head Island, South Carolina, USA), June 2000.

[12] R. Collins and R. Weiss, *Vanishing point calculation as a statistical inference on the unit sphere*, Int. Conf. on Computer Vision (ICCV), 1990, pp. 400–403.

[13] T. Conroy and J. Moore, *Resolution invariant surfaces for panoramic vision systems*, IEEE ICCV'99, 1999, pp. 392–397.

[14] Olivier Cuisenaire, *Distance transformations: Fast algorithms and applications to medical image processing*, Ph.D. thesis, U. Catholique de Louvain, October 1999.

[15] K. Daniilidis (ed.), *1st international ieee workshop on omnidirectional vision at cvpr 2000*, June 2000.

[16] C. Canudas de Wit, H. Khennouf, C. Samson, and O. J. Sordalen, *Chap.5: Nonlinear control design for mobile robots*, Nonlinear control for mobile robots (Yuan F. Zheng, ed.), World Scientific series in Robotics and Intelligent Systems, 1993.

[17] P. E. Debevec, C. J. Taylor, and J. Malik, *Modeling and rendering architecture from photographs: a hybrid geometry and image-based approach*, SIGGRAPH, 1996.

[18] S. Derrien and K. Konolige, *Approximating a single viewpoint in panoramic imaging devices*, Proceedings of the 1st International IEEE Workshop on Omnidiretional Vision at CVPR 2000 (Hilton Head, SC, USA), June 2000, pp. 85–90.

[19] G. DeSouza and A. Kak, *Vision for mobile robot navigation: A survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002), no. 2, 237–267.

[20] O. Faugeras, *Three-dimensional computer vision - a geometric viewpoint*, MIT Press, 1993.

[21] Mark Fiala, *Panoramic computer vision*, Ph.D. thesis, University of Alberta, 2002.

[22] J. Foote and D. Kimber, *Flycam: Practical panoramic video and automatic camera control*, Proc. of the IEEE Int. Conference on Multimedia and Expo, vol. III, August 2000, pp. 1419–1422.

[23] S. Gaechter, T. Pajdla, and B. Micusik, *Mirror design for an omnidirectional camera with a space variant imager*, IEEE Workshop on Omnidirectional Vision Applied to Robotic Orientation and Nondestructive Testing, August 2001, pp. 99–105.

[24] J. Gaspar, C. Deccó, J. Okamoto Jr, and J. Santos-Victor, *Constant resolution omnidirectional cameras*, 3rd International IEEE Workshop on Omni-directional Vision at ECCV, 2002, pp. 27–34.

[25] J. Gaspar, E. Grossmann, and J. Santos-Victor, *Interactive reconstruction from an omnidirectional image*, 9th International Symposium on Intelligent Robotic Systems (SIRS'01) (Toulouse, France), July 2001.

[26] J. Gaspar and J. Santos-Victor, *Visual path following with a catadioptric panoramic camera*, Int. Symp. Intelligent Robotic Systems (Coimbra, Portugal), July 1999, pp. 139–147.

[27] J. Gaspar, N. Winters, and J. Santos-Victor, *Vision-based navigation and environmental representations with an omni-directional camera*, IEEE Transactions on Robotics and Automation **16** (2000), no. 6, 890–898.

[28] José Gaspar, *Omnidirectional vision for mobile robot navigation*, Ph.D. thesis, Instituto Superior Técnico, Dept. Electrical Engineering, Lisbon - Portugal, 2003.

[29] D. Gavrila and V. Philomin, *Real-time object detection for smart vehicles*, IEEE, Int. Conf. on Computer Vision (ICCV), 1999, pp. 87–93.

[30] C. Geyer and K. Daniilidis, *A unifying theory for central panoramic systems and practical applications*, ECCV 2000 (Dublin, Ireland), June 2000, pp. 445–461.

[31] ———, *Catadioptric projective geometry*, International Journal of Computer Vision **43** (2001), 223–243.

[32] Gene H. Golub and Charles F. Van Loan, *Matrix computations*, third ed., Johns Hopkins Studies in the Mathematical Sciences, The Johns Hopkins University Press, Baltimore, MD, USA, 1996. MR 1 417 720

[33] P. Greguss, *Panoramic imaging block for 3d space*, US patent 4,566,763, January 1986, Hungarian Patent granted in 1983.

[34] P. Greguss (ed.), *Ieee icar 2001 workshop on omnidirectional vision applied to robotic orientation and non-destructive testing*, August 2001.

[35] E. Grossmann, D. Ortin, and J. Santos-Victor, *Algebraic aspects of reconstruction of structured scenes from one or more views*, British Machine Vision Conference, BMVC2001 (Manchester), September 2001, pp. 633–642.

[36] Etienne Grossmann, *Maximum likelihood 3d reconstruction from one or more uncalibrated views under geometric constraints*, Ph.D. thesis, Instituto Superior Técnico, Dept. Electrical Engineering, Lisbon - Portugal, 2002.

[37] E. Hecht and A. Zajac, *Optics*, Addison Wesley, 1974.

[38] R. Hicks and R. Bajcsy, *Catadioptric sensors that approximate wide-angle perspective projections*, IEEE Workshop on Omnidirectional Vision - OMNIVIS'00, June 2000, pp. 97–103.

[39] D. Huttenlocher, G. Klanderman, and W. Rucklidge, *Comparing images using the hausdorff distance*, IEEE Transactions on Pattern Analysis and Machine Intelligence **15** (1993), no. 9, 850–863.

[40] D. Huttenlocher, R. Lilien, and C. Olsen, *View-based recognition using an eigenspace approximation to the hausdorff measure*, IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999), no. 9, 951–956.

[41] S. B. Kang and R. Szeliski, *3d scene data recovery using omnidirectional multibaseline stereo*, CVPR, 1996, pp. 364–370.

[42] A. Kosaka and A. Kak, *Fast vision-guided mobile robot navigation using modelbased reasoning and prediction of uncertainties*, CVGIP: Image Understanding **56** (1992), no. 3, 271– 329.

[43] LIRA Lab, *Document on specification*, Tech. report, Esprit Project n. 31951 - SVAVISCA - available at http://www.lira.dist.unige.it - SVAVISCA - GIOTTO Home Page, May 1999.

[44] J. J. Leonard and H. F. Durrant-Whyte, *Mobile robot localization by tracking geometric beacons*, IEEE Trans. on Robotics and Automation **7** (1991), no. 3, 376–382.

[45] A. Majumder, W. Seales, G. Meenakshisundaram, and H. Fuchs, *Immersive teleconferencing: A new algorithm to generate seamless panoramic video imagery*, Proceedings of the 7th ACM Conference on Multimedia, 1999.

[46] B. McBride, *Panoramic cameras time line*, www page, http://panphoto.com/TimeLine.html.

[47] K. Miyamoto, *Fish-eye lens*, Journal of the Optical Society of America **54** (1964), no. 8, 1060–1061.

[48] H. Murase and S. K. Nayar, *Visual learning and recognition of 3d objects from appearance*, International Journal of Computer Vision **14** (1995), no. 1, 5–24.

[49] V. Nalwa, *A true omni-directional viewer*, Technical report, Bell Laboratories, Holmdel, New Jersey, USA, February 1996.

[50] S. K. Nayar, *Catadioptric image formation*, Proc. of the DARPA Image Understanding Workshop (New Orleans, LA, USA), May 1997, pp. 1431–1437.

[51] ———, *Catadioptric omnidirectional camera*, Proc. IEEE Conf. Computer Vision and Pattern Recognition (Puerto Rico), June 1997, pp. 482–488.

[52] S. K. Nayar and V. Peri, *Folded catadioptric cameras*, Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (Fort Collins), June 1999.

[53] E. Oja, *Subspace methods for pattern recognition*, Research Studies Press, 1983.

[54] T. Pajdla and V. Hlavac, *Zero phase representation of panoramic images for image based localization*, 8th Inter. Conf. on Computer Analysis of Images and Patterns CAIP'99, 1999.

[55] V. Peri and S. K. Nayar, *Generation of perspective and panoramic video from omnidirectional video*, Proc. DARPA Image Understanding Workshop, 1997, pp. 243–246.

[56] D. Rees, *Panoramic television viewing system, us patent 3 505 465*, postscript file, April 1970.

[57] W. Rucklidge, *Efficient visual recognition using the hausdorff distance*, Lecture Notes in Computer Science, vol. 1173, Springer-Verlag, 1996.

[58] J. Shi and C. Tomasi, *Good features to track*, Proc. of the IEEE Int. Conference on Computer Vision and Pattern Recognition, June 1994, pp. 593–600.

[59] T. Sogo, H. Ishiguro, and M. Treivedi, *Real-time target localization and tracking by n-ocular stereo*, Proceedings of the 1st International IEEE Workshop on Omni-directional Vision (OMNIVIS'00) at CVPR 2000 (Hilton Head Island, South Carolina, USA), June 2000.

[60] M. Spetsakis and J. Aloimonos, *Structure from motion using line correspondences*, International Journal of Computer Vision **4** (1990), no. 3, 171–183.

[61] P. Sturm, *A method for 3d reconstruction of piecewise planar objects from single panoramic images*, 1st International IEEE Workshop on Omnidirectional Vision at CVPR, 2000, pp. 119–126.

[62] T. Svoboda, T. Pajdla, and V. Hlaváč, *Epipolar geometry for panoramic cameras*, Proc. European Conf. Computer Vision (Freiburg Germany), July 1998, pp. 218–231.

[63] R. Talluri and J. K. Aggarwal, *Mobile robot self-location using model-image feature correspondence*, IEEE Transactions on Robotics and Automation **12** (1996), no. 1, 63–77.

[64] Geb Thomas, *Real-time panospheric image dewarping and presentation for remote mobile robot control*, Journal of Advanced Robotics **17** (2003), no. 4, 359368.

[65] S. Thrun and A. Bucken, *Integrating grid-based and topological maps for mobile robot navigation*, Proceedings of the 13th National Conference on Artifical Intelligence (AAAI'96), 1996.

[66] S. Watanabe, *Karhunen-loève expansion and factor analysis*, Transactions of the 4th Prague Conference on Information Theory, Statistical Decision Functions and Random Processes (Prague, Czech Republic), 1965, pp. 635–660.

[67] R. Wehner and S. Wehner, *Insect navigation: use of maps or ariadne's thread?*, Ethology, Ecology, Evolution **2** (1990), 27–48.

[68] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor, *Omni-directional vision for robot navigation*, Proceedings of the 1st International IEEE Workshop on Omni-directional Vision (OMNIVIS'00) at CVPR 2000 (Hilton Head Island, South Carolina, USA), June 2000.

[69] ———, *Omni-directional vision for robot navigation*, 1st International IEEE Workshop on Omni-directional Vision at CVPR, 2000, pp. 21–28.

[70] N. Winters and J.Santos-Victor, *Omni-directional visual navigation*, 7th International Symposium on Intelligent Robotics Systems (SIRS'99) (Coimbra, Portugal), July 1999, pp. 109–118.

[71] N. Winters and G. Lacey, *Overview of tele-operation for a mobile robot*, TMR Workshop on Computer Vision and Mobile Robots. (CVMR'98) (Santorini, Greece), September 1999.

[72] N. Winters and J. Santos-Victor, *Omni-directional visual navigation*, Proc. Int. Symp. on Intelligent Robotic Systems (Coimbra - Portugal), July 1999, pp. 109–118.

[73] Niall Winters, *A holistic approach to mobile robot navigation using omnidirectional vision*, Ph.D. thesis, University of Dublin, Trinity College, 2002.

[74] P. Wunsch and G. Hirzinger, *Real-time visual tracking of 3-d objects with dynamic handling of occlusion*, IEEE Int. Conf. on Robotics and Automation, April 1997, pp. 2868–2873.

[75] Y. Yagi, *Omnidirectional sensing and its applications*, IEICE Transactions on Information and Systems **E82-D** (1999), no. 3, 568–579.

[76] Y. Yagi, Y. Nishizawa, and M. Yachida, *Map-based navigation for mobile robot with omnidirectional image sensor COPIS*, IEEE Trans. Robotics and Automation **11** (1995), no. 5, 634–648.

[77] K. Yamazawa, Y. Yagi, and M. Yachida, *Obstacle detection with omnidirectional image sensor hyperomni vision*, IEEE ICRA, 1995, pp. 1062–1067.

[78] J. Zheng and S. Tsuji, *Panoramic representation for route recognition by a mobile robot*, International Journal of Computer Vision **9** (1992), no. 1, 55–76.