Omar Ait Aider, Philippe Hoppenot and Etienne Colle: " A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences" - Robotics and Autonomous Systems, vol. 52, p. 229-246, 2005.

# A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences

Omar Ait Aider, Philippe Hoppenot and Etienne Colle

*Laboratoire Systèmes Complexes - University of Evry, France,*
*oaider|hoppenot|ecolle@iup.univ-evry.fr*

## *Abstract*

*A model-based method for indoor mobile robot localization is presented herein; this method relies on monocular vision and uses straight-line correspondences. A classical four-step approach has been adopted (i.e. image acquisition, image feature extraction, image and model feature matching, and camera pose computing). These four steps will be discussed with special focus placed on the critical matching problem. An efficient and simple method for searching image and model feature correspondences, which has been designed for indoor mobile robot self-location, will be highlighted: this is a three-stage method based on the interpretation tree search approach. During the first stage, the correspondence space is reduced by virtue of splitting the navigable space into view-invariant regions. In making use of the specificity of the mobile robotics frame of reference, the global interpretation tree is divided into two sub-trees; two low-order geometric constraints are then defined and applied directly on 2D-3D correspondences in order to improve pruning and search efficiency. During the last stage, the pose is calculated for each matching hypothesis and the best one is selected according to a defined error function. Test results illustrate the performance of this approach.*

## 1. Introduction

In conjunction with development of the service robotics concept, such as cleaning robots and disabled-person assistance, we have witnessed over the past few years an extension in the field of mobile robot applications to indoor environments (e.g. offices, homes). Autonomy becomes a key issue when considering such an extension of potential uses. In this context, autonomy refers to the ability to perform navigational missions, explore the environment and attain goals without human intervention all while avoiding obstacles. It is neither elegant nor practical to equip or modify the home and office environments, as opposed to solutions designed for the industrial environment. Autonomy can indeed be obtained if one important task is correctly accomplished: self-localization.

Localization entails positioning the robot relative to its environment during navigational sessions. More formally, it indicates computing and updating three parameters $(x,y,\theta)$, which respectively define the orientation and position of the robot within the environmental frame. While GPS is considered as the best localization system for outdoor applications, no equivalent system has yet to be identified for indoor localization (Borenstein et al. 1997).

Researchers have devised a variety of techniques for mobile robot positioning. Solutions can be categorized into two groups: relative localization (dead-reckoning) and absolute localization. Relative localization utilizes information from proprioceptive sensors as odometer readings. Dead-reckoning algorithms are simple and fast; some factors however, such as slippage, cause error to increase and locational uncertainty to rise. Dead reckoning therefore proves insufficient. Absolute localization is based on exteroceptive sensor information. These methods yield a stable locating error but are more complex and costly in terms of computation time. One very popular solution for achieving online localization consists of combining the relative and absolute methods. Relative localization is used with a high sampling rate in order to maintain the robot pose up-to-date, whereas absolute localization is applied periodically with a lower sampling rate to correct relative positioning misalignments.

As regards absolute localization within indoor environments, map-based approaches naturally come to mind. In the large majority of cases, it is assumed that a map (model) of the workspace has been established. Three main types of environmental representations are available: topological maps, grid maps and the geometric primitive model. The latter two representations are called metric maps. The environmental model can either be pre-stored (architect's drawing or CAD model) or built online simultaneously with the localization using sensor data fusion or structure from motion technics (SLAM) as in Se et al. 2002.

The classical approach to model-based localization consists of matching the local representation of the environment built from sensor information with the global model map. When using monocular vision, the localization process is composed of the four following stages (DeSouza and Kak 2002):

- image acquisition from current robot position,
- image segmentation and feature extraction,
- matching 2D-image and 3D-model features,
- camera pose computation.

It is observed that model-based localization requires a succession of several processing steps and that each step may be time-consuming due to the large amount of data involved. The strategy ultimately adopted for each phase must then be very well-assessed for both online and real-time use.

The matching step represents one of the most important and complex phases in model-based localization; it entails determining the correct correspondence between image features and model features. Most methods designed to treat this problem have been developed for the object-recognition domain (Pennec 1998). In the field of mobile robotics, this problem is equivalent to considering local parts of the indoor environment model as objects to be recognized; however, certain particularities must be taken into account. Matching methods can be classified into two groups: methods that search for a solution within the “correspondence space” such as alignment (Ayache 1986, Lowe 1987), geometric hashing (Lamdan 1988) or interpretation tree search (Grimson 1987, 1990a) and methods that search within the “transformation space” such as the generalized Hough transform (Grimson 1990b).

A considerable body of work exists on model-based robot localization using visual sensors. Some research has focused specially on the matching problem. One popular approach is described by Kosaka and Kak 1992 and by DeSouza 2002. In these works, both robot orientation and position are updated from an initial setting, in accordance with the navigational commands. The new robot position and orientation obtained are then used to project model features on the image plane and compare them with the actual image being viewed. Uncertainty on the new robot position and orientation is propagated to predict uncertainty zones in the image, where a search for corresponding features is required. Uncertainty propagation is performed using covariance matrix propagation by means of the Jacobian matrix and Mahalanobis distance and by assuming local linearity of the command and projection equations. The correct correspondences are selected through applying the nearest neighbor rule. Once matching has been achieved, the pose is corrected using Kalman filtering. Due to the dimensions of sensed features as well as to perspective effects, the uncertainty propagation process may produce oversized zones on the image plane. Since uncertainty grows as the distance browsed between two image acquisitions increases, this approach necessitates a high sampling rate. Moreover, it must be recalled that not every image acquisition provides sufficient information to accurately update the robot pose. An alternative approach has been adopted by Talluri and Aggarwal 1996. In this work, the localization problem is handled by introducing the notion of Edge Visibility Regions, which are equivalent to aspect graphs within the object-recognition domain. These regions are used to restrict the matching process to a small subset of features in the global model; an interpretation tree search method is then applied. For the global localization problem, a generalized Hough transform is implemented to select a small set of candidate regions. The investigation is unfortunately restricted to horizontal features representing rooftop edges. Conversely, Munoz and Gonzalez 1998 used only vertical-line correspondences and an interpretation tree search in order to determine the interpretation that maximizes the number of correspondence pairs. A restriction is introduced on the global model edges thanks to dead-reckoning pose estimation. The geometric constraint added to eliminate false correspondences is based on the order of model vertices stored in a circular list and involves identifying the order of appearance of corresponding segments on the image. Confining the search to a particular category of segments constitutes a real shortcoming in the indoor application. One disadvantage of model-based localization is in fact the required minimum number of distinguishable features eligible to be used for matching. Eliminating some of the environmental features would not seem to be a wise strategy, especially when the reliability of a matching result is correlated with the number of correspondence pairs.

It thus appears that a method not requiring a high sampling rate would be preferable. In some cases, it may actually prove necessary to delay application of absolute localization to the subsequent image acquisition, thereby increasing pose estimation uncertainty. It is not guaranteed that a sufficient number of viewed features would be contained at each image acquisition. In addition, mobile robots are generally multitask systems and localization is not the only process running during navigation. For some image acquisitions therefore, system resources may already be allocated. Consequently, the periodicity of absolute localization is variable, which necessitates developing a localization method less sensitive to pose estimation quality.

In this paper, an incremental model-based localization method using monocular vision will be presented. The classical four-step general approach of model-based localization, as discussed above, has been adopted. The most important contribution herein is the development of an effective strategy for resolving the crucial problem of finding the correct match between image and model features. The aim of the approach derived is to preserve matching efficiency without increasing the sampling rate of absolute location operations.

The method is based on straight-line correspondences. The environment is assumed to be partially known (modeled), which implies that the environment is constructed offline and pre-stored. Both the matching and pose-computing methods are based on a full-perspective camera model. The equations that model perspective projections of 3D features on the image are non-linear. A large number of object-recognition applications make use of simplified camera models as para-perspective and weak-perspective models (Horaud and Monga 1993). In mobile robotics applications, the dimension of viewed features is high in comparison with distance to the camera focal center. It then becomes necessary to use a full-perspective model. A pinhole camera model has been obtained by means of offline calibration.

The global localization process will be presented in the next three sections. The indoor environment representation used in the method will be developed in Section 2. The model contains the main parts of the environment that are not supposed to move, such as wall and door edges, or whose movements are known and cannot be due to randomness, e.g. heavy pieces of furniture. Furthermore, the navigation plane has been split into View-Invariant Regions (or VIRs) (Simsarian et al. 1996) in order to improve efficiency by taking the occlusion phenomenon into account during the matching process, as will be shown below. The construction of VIRs will be briefly developed in this section as well. The algorithm generated to solve the critical feature-matching problem will then be discussed in detail in Section 3. The method chosen is based on the interpretation tree search approach, with the originality lying in the subdivision of the correspondence space using a constraint on the orientation of model edges and image features. Two interpretation trees are then obtained and the dimension of the space to be analyzed is reduced exponentially. Another contribution of this work consists of defining low-order geometric constraints, which are directly applicable on 2D-3D correspondences, in order to prune these interpretation trees. In addition, some of the restrictions on existing methods due to singular model edge configurations have been eliminated. Lastly, the camera pose-computing method based on numerical optimization has been simplified by taking the specific configuration of our robot into account. This method is then evaluated with both synthetic data and real images. Results, presented in Section 4, indicate that the correct correspondence combination is derived in more than 98% of cases with great efficiency and that the method is less sensitive to the quality of the initial pose estimate. Moreover, localization accuracy satisfies contextual requirements and enables achieving a navigational session within an indoor environment.

## 2. Indoor environment modeling

The model adopted for the present work is of the geometric feature type. This type is well-suited to applications using visual sensors. One of its advantages is concision and the possibility of representing uncertainty and frame changing in order to predict what is expected to be seen from a predicted camera pose. Existing camera pose-computing methods (for a given set of 3D-2D correspondences) generally use point- or straight-line correspondences. Image features result from image segmentation into contours which correspond to physical elements within the indoor work space, such as edges constituted by intersections between flat surfaces. These edges tend to be straight segments. Lines are easier to extract from contour images and their characterization, by means of polygonal approximation, is reliable even in the presence of noise. Partial occlusion (due to either the view angle or the presence of non-modeled objects) does not affect line representation parameters; using straight lines would therefore seem to be more judicious.

Indoor environments are manmade and semi-structured. This kind of environment can generally be viewed as a polygonal navigable space limited by vertical planes (walls). The environment can then be simply represented by a wire frame model composed of a set of line segments corresponding to the edges of these planes. The model is fixed and built offline; given the capability of camera pose estimation in a semi-structured environment (e.g. indoor environments), it is relatively straightforward to predict which parts will be sensed and then accurately describe these parts. This procedure allows avoiding online map construction (SLAM), which necessitates major processing of 3D reconstruction from stereovision or image sequences and introduces noise on model feature parameters. The 3D environment model is thus merely a wire frame representation comprising a set of straight segments.

One disadvantage with the wire frame representation is its inability to incorporate the occlusion phenomenon. The model appears to be transparent; this situation is managed by applying the notion of view-invariant regions. This adaptation of the aspect graphs used in object recognition to the domain of mobile robotics has been proposed by Guibas et al. 1992;

Talluri and Aggarwal 1996 suggested that the polygonal navigable space (which defines all possible positions of the camera) is clustered into non-overlapping polygons, called Edge-Visibility Regions (*EVR*), while Simsarian et al. 1996 introduced the notion of View-Invariant Regions (*VIR*), with a visibility list associated to each region. Each list is created from the subset of segments visible from the corresponding region (see Figure 1) within the global environment model.

The method employed to construct these VIR has been well-explained in Simsarian et al. 1996. Beginning with a global polygon, the principle herein is to first define the region boundaries. Each boundary is a straight-line segment defined using the convex vertices of the model yielding the occlusion phenomenon. Each boundary splits the polygons it crosses into two new polygons. Crossing a boundary causes either the disappearance or appearance of at least one model edge. Once all boundaries have been constructed using all model vertices, the polygons so defined represent the model VIRs.

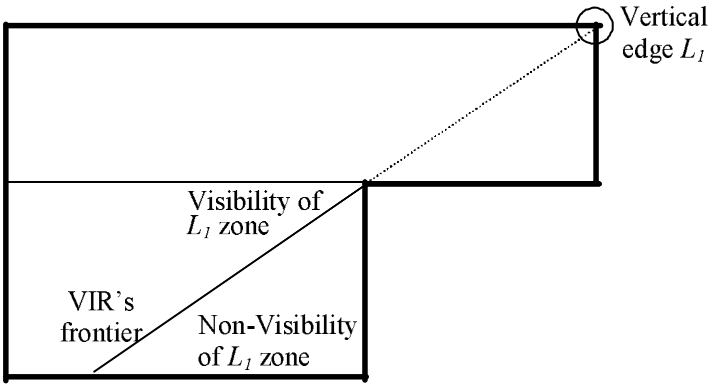

***Figure 1:** Simple example of view-invariant regions*

As range of orientations (from which the edge could be viewed) is associated with each segment of each visibility list. The way to determine the orientation range [$\phi_{\min}$,$\phi_{\max}$] is shown in Figure 2:

- $\phi_{\min} = \min(\phi_1, \phi_2, \ldots)$

- $\phi_{\max} = \max(\phi_1, \phi_2, \ldots)$

where ($\phi_1$, $\phi_2$,…) are the view angles of the edge from each VIR corner.

The same reasoning may be applied to a non-vertical edge using the projections of its extremities on the horizontal plane.

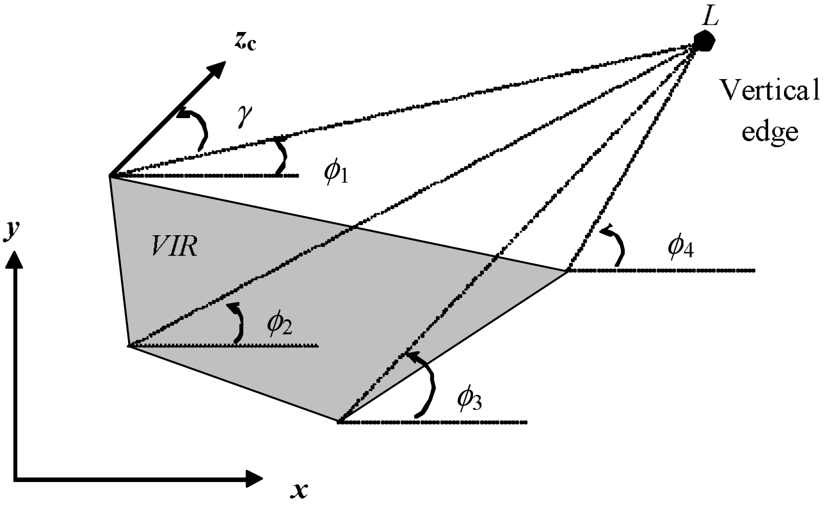

***Figure 2:*** *Definition of a range of orientations from which a model edge is visible*

## 3. Straight-line matching strategy

In this section, we will focus on the critical phase of identifying the correct match between the set of image lines and model lines. The most popular method is the interpretation tree search using point correspondences, as presented by Grimson and Lozano-Perez 1987. Murray and Cook 1988 developed a variant using line correspondences. Considering a set of $m$ model segments $\boldsymbol{L}=\{L_1,L_2,\ldots,L_m\}$ and a set of $n$ image lines $\boldsymbol{l}=\{l_1,l_2,\ldots,l_n\}$, the interpretation tree represents all possible combinations of $\boldsymbol{L}$ and $\boldsymbol{l}$ element couples. The theoretical number of possibilities (in recognition of the fact that one or more model lines may not be present in the image due to occlusion or non-detection during contour extraction) is $(n+1)^m$. Yet a considerable portion of the tree never gets examined. Some geometric constraints are introduced in order to prune the tree. The most commonly-used constraints are unary and binary. Unary constraints must be satisfied by each $L_i$-$l_j$ couple, such as length. Binary constraints specify that for each set of couple pairs $L_i$-$l_j$, $L_k$-$l_l$, the angle between $L_i$ and $L_k$ must be the same as the estimated angle between $l_j$ and $l_l$. Higher-order geometric constraints can then be introduced. One weakness in the approach is that the two sets $\boldsymbol{L}$ and $\boldsymbol{l}$ must be expressed in the same dimensional space number. Model segments lie in a 3D space since image lines are in 2D space. The 3D parameters of each 2D image line must then be estimated by means of the stereovision technique. Another eventual possibility would be to introduce a verification step into the search process by computing the camera pose from current hypothetical matching. In projecting each model line onto the image using the computed pose and then comparing the lines obtained with viewed lines, current matching can either be accepted or rejected. This procedure unfortunately dramatically increases computational costs and makes the algorithm unusable in real-time applications.

Our goal here is to develop a matching method well-adapted to mobile robotics by improving both the efficiency and computational costs of the search process. To attain this objective, certain constraints had to be imposed:

- correspondence space dimensions must be reduced to the greatest extent possible by making use of the available robot pose estimate, in the aim of limiting the number of possible combinations;
- geometric constraints must be directly applicable on 2D-3D correspondences in order to minimize the number of pose-computing operations;
- a full-perspective camera model must be adopted for expressing equations.

We are proposing a three-stage method that works within the correspondence space and utilizes the specificity of the mobile robotics context. The first stage consists of reducing the correspondence space. Based on approximate knowledge of the camera pose, one or two VIRs, which possibly contain the robot pose, are selected. Only the edges of their visibility lists are then retained during the subsequent stages of the matching process. Each of the two sets of features (model and image features) is divided into two subsets by use of a constraint on their orientation. Two interpretation trees are then built with these feature sets. In the second stage, a finite and reduced set of correspondence combination hypotheses is generated by pruning the interpretation trees. Two geometric constraints are used at this point to discard inconsistent correspondences by testing their local coherence. In the third stage, the most globally-consistent hypothesis is selected. A pose is computed for each correspondence combination. Model features are then projected into the image plane. Matching hypotheses are ordered according to an error function defined using the principle of the Hausdorf distance between projected and viewed features. Details of these three stages will be presented in the next subsections following a mathematical formulation of the localization problem.

In order to provide a mathematical formulation of the camera-localization problem, let's consider two coordinate systems: the global coordinate system $(\boldsymbol{O},\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ related to the environment; and the camera coordinate system $(\boldsymbol{O_c},\boldsymbol{x_c},\boldsymbol{y_c},\boldsymbol{z_c})$, where $\boldsymbol{O_c}$ is the camera optical center and where the image plane is orthogonal to $\boldsymbol{z_c}$, situated at a distance $f$ (focal lens) from $\boldsymbol{O_c}$. In this work, it has been assumed that the homogeneous transformation between the robot frame and a camera frame tied to the robot base is known at each moment. Localizing the mobile robot is thus considered equivalent to computing the camera pose. Camera location is characterized by a translation vector $\boldsymbol{T} = [x\ y\ z]^t$ and a rotational matrix $\boldsymbol{R}$. $\boldsymbol{T}$ represents the translation between $\boldsymbol{O_c}$ and $\boldsymbol{O}$. $\boldsymbol{R}$ is composed of the three elementary angles $\Psi$, $\theta$ and $\phi$ about the $\boldsymbol{x_c}$, $\boldsymbol{y_c}$ and $\boldsymbol{z_c}$-axes, respectively. $\boldsymbol{R}$ and $\boldsymbol{T}$ carry the camera frame onto the global frame. In general, indoor mobile robots operate within a 3D environment, yet their displacements lie in a 2D space. The camera thus moves in a horizontal plane at a known and constant height. $z$ and $\theta$ are then assumed to be known and $\Psi$ is zero. At this point, the system features 3 DOF (degrees of freedom) with parameters $\phi$, $x$ and $y$.

### 3.1 Reducing correspondence space (Selecting model segments)

A starting assumption necessary for the following method is that the robot has an approximate estimation of its location. This assumption is not very restricting. Indeed, most of mobile robots are equipped with wheel encoders which provide an estimation of the pose. In addition, a relatively large interval of poses is needed to start the matching process.

The complete model of the indoor work space contains generally a great number of segments. Having an estimation of the camera orientation $\phi_e$ and position $[x_e, y_e]$ and higher bounds $\delta\phi$, $\delta t$ of orientation and position estimation error, only few regions can be retained. Thus only segments belonging to the visibility lists of these regions will compose the set $\boldsymbol{L}$ of model segments participating to the matching process.

### 3.2 Matching hypothesis generation

**Unary geometric constraint definition.** Let us consider a 3D line segment $L_i$ of $\boldsymbol{L}$ defined by its direction vector $\boldsymbol{v}_i$ and its position vector $\boldsymbol{p}_i$ in the global coordinate system. $\boldsymbol{v}_i'=\boldsymbol{R}\boldsymbol{v}_i$ and $\boldsymbol{p}_i'=\boldsymbol{R}(\boldsymbol{p}_i-\boldsymbol{T})$ are the expressions of $\boldsymbol{v}_i$ and $\boldsymbol{p}_i$ in the camera coordinate system. Assuming that $l_j$ is the projection of $L_i$ in the image plane, $L_i$ and $l_i$ belongs to an interpretation plane passing through the focus point $\boldsymbol{O_c}$. Let $\boldsymbol{n}_j$ be the unit vector normal to this plane expressed in the camera coordinates system (Figure 3). It is possible to calculate $\boldsymbol{n}_j$ knowing $l_j$ and the intrinsic camera parameters $u_0$, $v_0$, $\alpha_u$, $\alpha_v$ (Horaud and Monga 1993). Expressing the dot products $\boldsymbol{n}_j\,.\boldsymbol{v}_i$' and $\boldsymbol{n}_j.\boldsymbol{p}_i$', one can write

$$\boldsymbol{n}_j^t \boldsymbol{R} \boldsymbol{v}_i = 0 \qquad (1.a)$$

$$\boldsymbol{n}_j^t (\boldsymbol{R}(\boldsymbol{p}_i - \boldsymbol{T})) = 0 \qquad (1.b)$$

According to the mobile robotics context presented before, the rotation matrix can be written as follows:

$$\boldsymbol{R} = \begin{pmatrix} \cos(\theta)\cos(\phi) & -\cos(\theta)\sin(\phi) & \sin(\theta) \\ \sin(\phi) & \cos(\phi) & 0 \\ -\sin(\theta)\cos(\phi) & \sin(\theta)\sin(\phi) & \cos(\theta) \end{pmatrix}$$

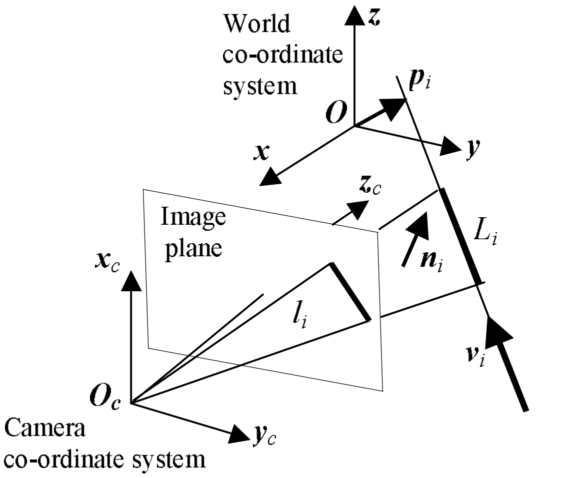

***Figure 3:** projection of a model segment*

By replacing $\boldsymbol{R}$ we obtain two equations of the following forms:

$$a_{ij}.\cos(\phi) + b_{ij}.\sin(\phi) + c_{ij} = 0 \qquad (2.a)$$

$$A_{ij}.x + B_{ij}.y + C_{ij} = 0 \qquad (2.b)$$

with :

$$\begin{cases} a_{ij} = (n_{jx}\cos(\theta) - n_{jz}\sin(\theta))v_{ix} + n_{jy}v_{iy} \\ b_{ij} = n_{jy}v_{ix} - (n_{jx}\cos(\theta) - n_{jz}\sin(\theta))v_{iy} \\ c_{ij} = (n_{jx}\sin(\theta) + n_{jz}\sin(\theta))v_{iz} \\ A_{ij} = -(n_{jx}\cos(\theta) - n_{jz}\sin(\theta))\cos(\phi) - n_{jy}\sin(\phi) \\ B_{ij} = -n_{jy}\cos(\phi) + (n_{jx}\cos(\theta) - n_{jz}\sin(\theta))\sin(\phi) \\ C_{ij} = [(n_{jx}\cos(\theta) - n_{jz}\sin(\theta))x_i + n_{jy}y_i]\cos(\phi) + [n_{iy}x_i - (n_{jx}\cos(\theta) - n_{jz}\sin(\theta))y_i]\sin(\phi) + \\ \qquad\qquad [n_{jx}\sin(\theta) + n_{jz}\cos(\theta)](z_i - z) \end{cases}$$

Thus, each correspondence permits to calculate $\phi$ using (1.a) and then to replace it by its value in (1.b) to constrain the position of the robot to belong to a straight line in the $xy$-plane whose equation is $A_{ij}.x + B_{ij}.y + C_{ij} = 0$.

A unary geometric constraint is then expressed as follows: a correspondence $L_i$-$l_j$ is an acceptable matching if the corresponding calculated orientation verifies $|\phi-\phi_e|<\delta\phi$ and the calculated straight line intersects with the circle whose center is $[x_e,y_e]$ and whose radius is $\delta t$ (figure 4).

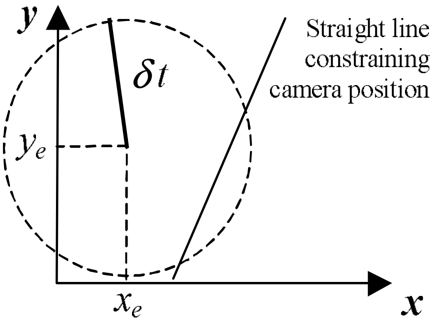

***Figure 4:*** *unary geometric constraint*

Unfortunately, if the model segment is vertical, parameters $a_i$ and $b_i$ are zero. The orientation can not be calculated in this way. Unary geometric constraint is then unusable. We propose the following solution.

**Binary geometric constraint definition.** Several works, as Betke and Gurvits 1997, uses vertical lines to calculate the camera pose. They generally assume that the angle $\theta$=90°, so that the projection of a vertical segment of the model is a vertical line in the image. A classical triangulation method is then used to constrain the camera position. In practice, in most mobile robots $\theta$ is known but variable (pan and tilt camera systems). The projection of a vertical segment is then not necessarily vertical in the image. The triangulation method can not be used directly. We propose to extend the method to the case where $\theta\neq$ 90°. The solution is first presented for $\theta$= 90° and then extended to the general case.

Let us consider a subset of vertical segments $\boldsymbol{L_v}$ of $\boldsymbol{L}$ and a subset of vertical image lines $\boldsymbol{l_v}$ of $\boldsymbol{l}$. The equation of $l_{vi}$ in the image plane can be written $v=v_i$ (figure 5).

Considering a couple of correspondences ($L_i$-$l_j$,$L_k$-$l_l$) observed from an unknown camera position $(x,y)$. The angle of view $\omega$ between $l_j$ and $l_l$ can be calculated from the image measurements $v_j$, $v_l$ and the camera intrinsic parameters as follows by computing first the view angle of each feature:

$$\omega_j = \arctan\left(\frac{v_j - v_0}{\alpha_v}\right) \text{and } \omega_l = \arctan\left(\frac{v_l - v_0}{\alpha_v}\right)$$

and then by :

$$\omega_{jl} = \left|\omega_j - \omega_l\right| \qquad (3)$$

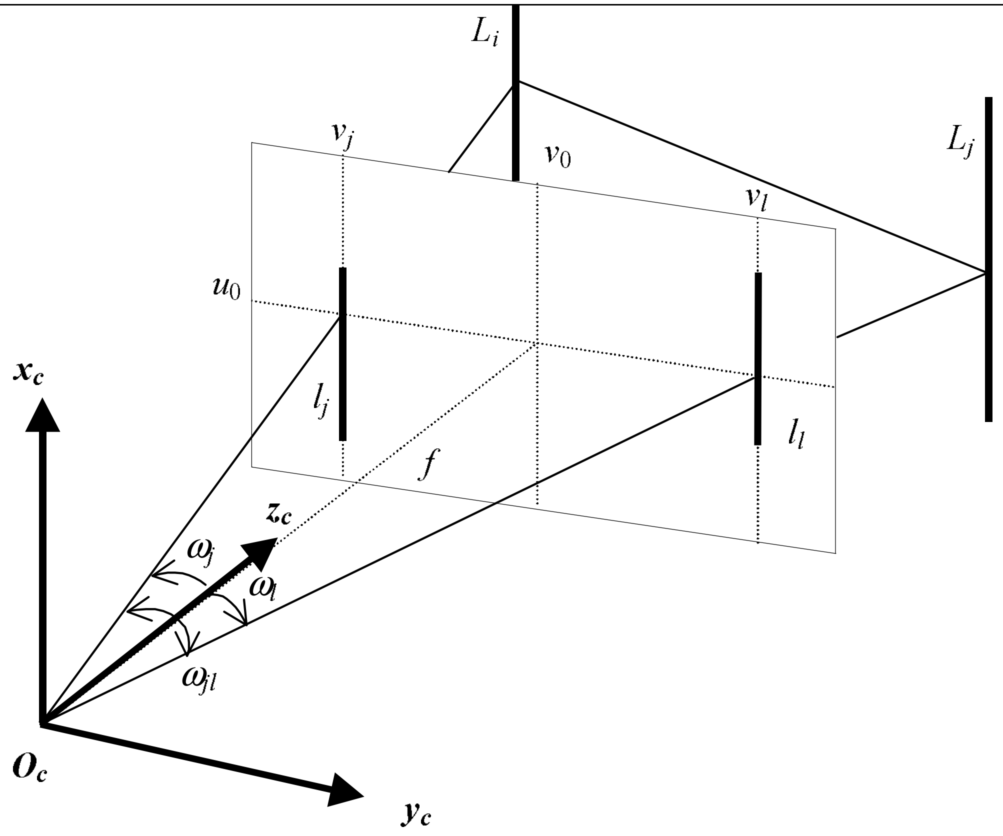

***Figure 5:*** *angle of view of vertical edges for θ=90°*

The relationship between the view angle and the coordinates of the two vertical segments can be written:

$$d_{ik}^2 = d_i^2 + d_k^2 - 2d_i d_k \cos(\omega_{jl}) \qquad (4)$$

where

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2}$$
$$d_k = \sqrt{(x_k - x)^2 + (y_k - y)^2}$$

According to this relationship, the view angle restricts the possible camera positions to an arc of a circle as shown in figure 6. Theoretically, two circles can satisfy the angular constraint. The false one can be eliminated by satisfying if the right to left order of the vertical lines in the image is consistent with each range of positions.

Then, a binary geometric constraints can be expressed as follows: a couple of correspondences ($L_i$-$l_j$,$L_k$-$l_l$) is acceptable if the corresponding circle arc intersects with the circle whose center is [$x_e$,$y_e$] and whose radius is $\delta t$ and the right to left image order of $l_j$ and $l_l$ is preserved.

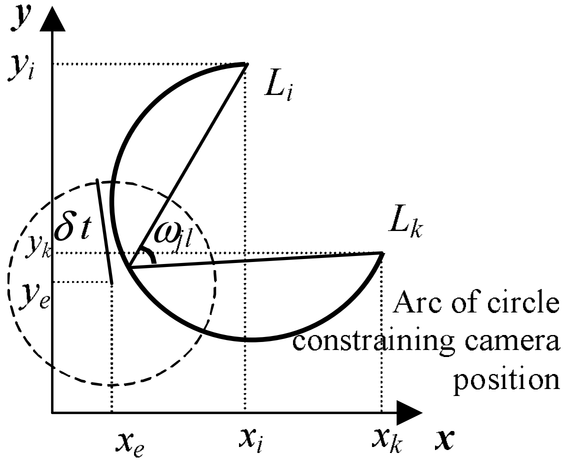

***Figure 6:** binary geometric constraint (top view of two vertical model segments $L_i$ and $L_k$)*

To generalize the binary constraint to the case where $\theta$ is not equal to 90°, one have first to identify the image lines which potentially correspond to vertical segments in the model. An image line $l$ is defined in the image plane by its slope $\rho$ and its distance to the origin $d$. $l$ may correspond to a vertical segment in the model if some rules are verified:

- if $0<\theta<90°$ then

  $\rho_{min}<\rho<\rho_{max}$ when $l$ is contained in the image right half, $-\rho_{max}<\rho<-\rho_{min}$ otherwise.

- if $90°<\theta<180°$ then

  $-\rho_{max}<\rho<-\rho_{min}$ when $l$ is contained in the image right half, $\rho_{min}<\rho<\rho_{max}$ otherwise.

$\rho_{min}$ and $\rho_{max}$ are proportional to $\theta_{min}$ and $\theta_{max}$ which generally can be situated around 45° and 135° respectively (figure 7).

The limits imposed by these rules must be made fuzzy in introducing flexible thresholds to take into account noisy data such as line parameters estimation and camera calibration errors.

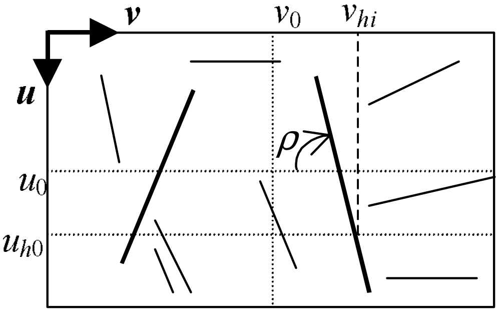

***Figure 7:** Only lines in bold are selected as possibly corresponding to vertical edges.*

We now have a set of image lines which can possibly correspond to vertical edges. To apply the binary constraint we need to calculate the $v_i$ measurements. If the camera was in an horizontal position ($\theta$=90°) the lines would be vertical and have as equation $v=v_i$ . We define a so called "horizon" line in the image whose equation is $u=u_{h0}$. $u_{h0}$ can easily be calculated knowing $\theta$ as follows :

$$u_{hor} = \tan\left(\theta - \frac{\pi}{2}\right)\alpha_u + u_0 \qquad (7)$$

To use the triangulation as in the case where $\theta = 90°$, one need to retrieve the measurements $v_i$. In other words, one has to build a virtual image corresponding to the case where $\theta = 90°$ of observed segments at any value of $\theta$. $v_i$ Measurements can be obtained from the intersections $v_{hi}$ of each line with the horizon line (figure 8) as follows :

$$v_i = (v_{hi} - v_0).\sin(\theta) + v_0 \qquad (8)$$

The demonstration is given in Appendix A.

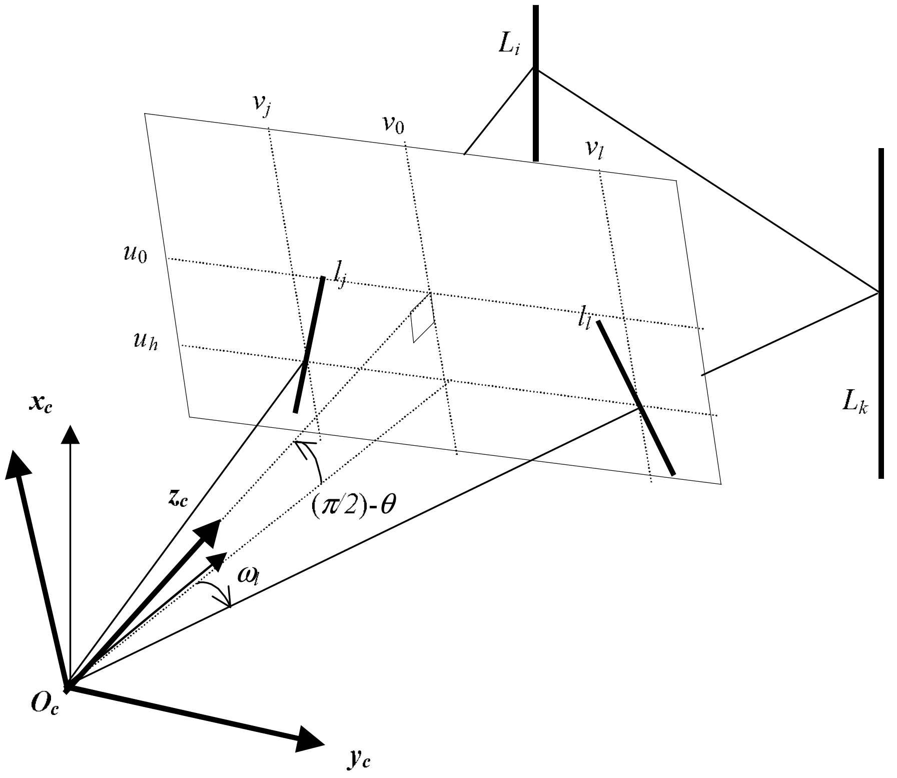

***Figure 8:*** *binary constraint in the general case ($\theta \neq 90°$)*

**Constructing the matching hypothesis list.** The matching hypothesis generation process can now be summarized as follows:

- Starting from a set of $m$ model lines $\boldsymbol{L}=\{L_1,L_2,\ldots,L_m\}$ and a set of $n$ image lines $\boldsymbol{l}=\{l_1,l_2,\ldots,l_n\}$, rather than using a classical interpretation tree constructed from $\boldsymbol{L}$ and $\boldsymbol{l}$ elements, $\boldsymbol{L}$ is first divided into two groups of vertical segments $\boldsymbol{L_v}=\{L_1,L_2,\ldots,L_{mv}\}$ and non-vertical segments $\boldsymbol{L_h}=\{L_1,L_2,\ldots,L_{mh}\}$. A subset $\boldsymbol{l_v}$ is extracted from $\boldsymbol{l}$ using the previous procedure. Two smaller interpretation sub-trees are constructed, one from $\boldsymbol{L_h}$ and $\boldsymbol{l}$ elements and the other from couples of $\boldsymbol{L_v}$ and couples of $\boldsymbol{l_v}$ elements.

- The unary geometric constraint is applied to prune the first sub-tree. Only retained correspondences serve to compose a list of combinations of $\boldsymbol{L_h}$ and $\boldsymbol{l}$ elements.

- The binary geometric constraint is used to prune the second sub-tree. It is applied on the couples of correspondences formed from $\boldsymbol{L_v}$ and $\boldsymbol{l_v}$. Only retained couples serve to compose a list of combinations of $\boldsymbol{L_v}$ and $\boldsymbol{l_v}$ elements. One more constraint is applied on the combinations of this list. It verifies the consistency of right to left apparition order of the segments on the image. The correct order can be computed offline and does not change when the camera moves inside the same VIR. The way to compute this order is presented in Appendix B. Combinations which do not verify the constraint are discarded.

The obtained two lists are then combined to form a complete list of possible matching hypothesis of ***L*** and ***l*** elements. The combinations which do not contain a sufficient number of correspondences are discarded. This number represents a low threshold on the fraction of hypothesized matches among model edges which are supposed to be viewed to eliminate the possibility of accidental correspondence combinations, also called false positives (Pennec 1998, Grimson and Huttenlocher 1991).

### 3.3 Matching Hypothesis verification

The thresholds used in unary and binary constraints are not tight enough to eliminate all the false matching hypothesis. The previous step serves to verify the local coherence of each correspondence or couple of correspondences. But a combination composed of locally consistent correspondences may be non globally consistent. The last step is to select the optimal match (the most globally consistent), if it does exist, among the set of retained hypothesis. Several approaches exist taking into account the fraction of correctly matched model features, the probability of accidental pairings and image reconstruction errors.

At this level, a small set of matching hypotheses are retained. Thus, it becomes now feasible to test each hypothesis by computing the corresponding camera pose. Indeed, for each correspondence combination, a camera pose is computed by mean of least square method. Model edges are projected using this pose onto image plan. The degree of fit between projected segments and viewed segments is estimated using a distance function based on Hausdorff distance. In the theoretical case, the right matching hypothesis would produce null errors. In practice, image measurements contain noise introduced at different stages (image acquisition, segmentation into contours, polygonal approximation). In addition, calibration errors make the projection of model segments uncertain. Thus, the best hypothesis according to this function is selected as the correct correspondence combination. The pose computing and the distance function definition are explained in the following sub-sections.

**Camera pose computing.** The method used to compute ***R*** and ***T*** is an adaptation of Phong-Horaud (Phong and Horaud 1995) method to mobile robotics context. It is a method based on numerical non-linear least square optimization of the error criterion obtained by the equation system (1) expressed with respect to $\phi$, $x$, $y$. The Levenberg-Marquardt algorithm is used.

**Distance between projected and viewed segments.** One may think that it is enough to use the residuals at convergence of the error criterion (1) as a metric to evaluate a distance between projected and viewed segments on image plan. This is what is done in (Chang and Tsai 1999) where a $\chi^2$ test is derived from this residuals to evaluate the reliability of hypothesized matching. This is not a good idea because this criterion considers in fact image segments as infinite straight lines. This means that a pose which makes a projected segment co-linear with its hypothetical viewed correspondent can be interpreted as correct even if the two segments are not superimposed. The function used in this work is inspired from the Hausdorff distance between two sets of features. Hausdorff distance is the longest metric distance among the distances between each feature of one set and its nearest neighbor in the second set. The problem is then to define the metric distance between two straight segments with different orientations positions and lengths. The solution adopted consists in measuring the sum of distances between each point on the projected segment to its nearest point on the viewed segment.

Let us consider two segments, a viewed segment $l$ and a projected segment $l_p$ defined by their starting and ending points in the image plan $m^s, m^e$ and $m_p^s, m_p^e$ respectively. Let $y=ax+b$ be the equation of the segment $l$ in the 2D frame $\left(m_p^s, \boldsymbol{x}, \boldsymbol{y}\right)$ where $\boldsymbol{x}_p = \frac{\overrightarrow{m_p^s m_p^e}}{\left\| m_p^s m_p^e \right\|}$, $\boldsymbol{y}_p = \frac{\overrightarrow{m_p^s m_i}}{\left\| m_p^s m_i \right\|}$ and $m_i$ the intersection of the straight line containing $l$ and the straight line perpendicular to $l_p$ and passing through $m_p^s$ (figure 9). Let $m(x_m, y_m)$ be a point on the segment $l$. the nearest point to $m$ on $l_p$ is :

- $m_p^s$ if $x_m \leq 0$,
- its perpendicular projection on $l_p$ if $0 < x_m < \lambda_p$ (where $\lambda_p$ is the length of $l_p$),
- $m_p^e$ if $x_m \geq \lambda_p$.

Thus, the distance between the two segments is defined by

$$D(l_p, l) = \frac{1}{\lambda_p} \int_{x^s}^{x^e} d(x) dx \qquad (9)$$

where

- $d(x_m) = \sqrt{x_m^2 + \left(y_m - y_p^s\right)^2}$ if $x_m^s \leq 0$,
- $d(x_m)$=(perpendicular distance to $l_p$ ) if $0 < x_m < \lambda_p$,
- $d(x_m) = \sqrt{\left(x_m - x_p^e\right)^2 + \left(y_m - y_p^e\right)^2}$ if $x_m \geq \lambda_p$.

It is possible now to define the distance between two sets of projected and viewed segments, $L_p = \{l_{p1}, l_{p2}, \ldots, l_{pm}\}$ and $L = \{l_1, l_2, \ldots, l_m\}$, using the Hausdorff principle, by :

$$h(L_p, L) = \max_{i=1,\ldots,k} \left(k \min(D(l_{pi}, l_i))\right) \qquad (10)$$

where $k$min represents the $k$ shortest distances $D$ among the distances between the segments of correspondences composing the matching hypothesis. $k$ is the threshold on the number of correspondences defined above.

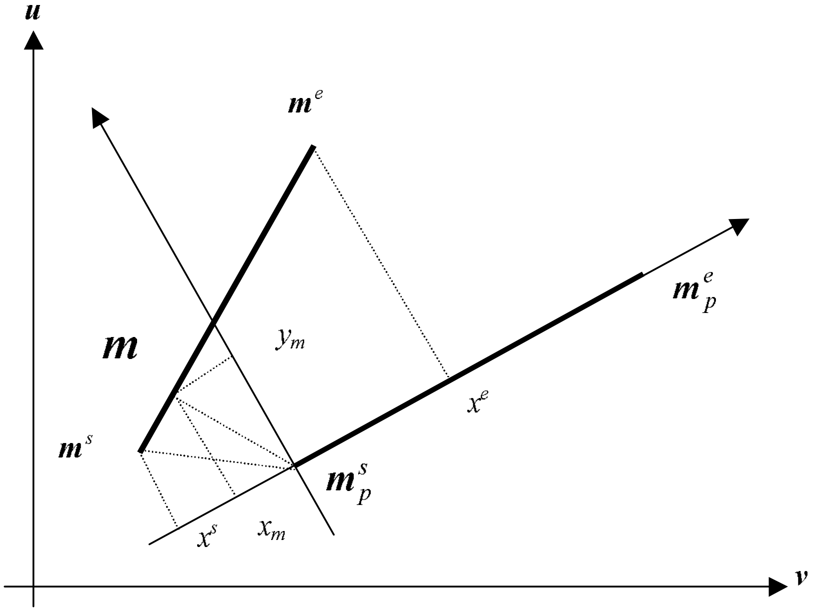

***Figure 9:*** *distance between two segments in image plan*

### 3.4 Taking uncertainty due to noise into account

The unary and binary constraints used to test the local coherence of correspondences and to prune the two interpretation sub-trees are based both on pose estimation with error margin and on image segments. Each image segment is represented by the vector normal to its interpretation plane $\boldsymbol{n}$. $\boldsymbol{n}$ components are noisy due to errors in contour detection as well as to polygonal approximation, with this uncertainty propagating to the parameters of the pose constraint regions (a straight line represented by three parameters $A_{ij}$, $B_{ij}$, $C_{ij}$ in the case of a unary constraint and an arc of a circle represented by its center coordinates $x_c$, $y_c$ and its radius $d$ in the case of a binary constraint). In order to estimate the variation margins of parameters in these constraint regions, a Monte Carlo method is applied offline. This approach is most useful due to the strong non-linearity of the equations yielding constraint region parameters with respect to $\boldsymbol{n}$ image components.

**Uncertainty in the unary constraint:** For each model $L_i$ segment, a set of synthetic image segments $l_j$ represented by their normal interpretation plane vector $\boldsymbol{n_j}$ is used. For each correspondence, $L_i$-$l_j$ random samples of $\boldsymbol{n}_j^s$, belonging to a normal distribution of mean value $\boldsymbol{n_j}$ and standard deviation matrix $\boldsymbol{\sigma_{nj}}$, are generated. $\phi^s_{ij}$, $A^s_{ij}$, $B^s_{ij}$ and $C^s_{ij}$ are computed for each sample. Mean values and standard deviations of parameters $\phi^s_{ij}$, $A^s_{ij}$, $B^s_{ij}$ and $C^s_{ij}$ are then estimated. During the online application of geometric constraints on a hypothetic correspondence $L_i$-$l_j$, the orientation $\phi_{ij}$ and parameters $A_{ij}$, $B_{ij}$, $C_{ij}$ of the straight-line constraint are computed. The test thus consists of searching within the zone established by the offline-computed standard deviations around $\phi_{ij}$ and $A_{ij}$, $B_{ij}$, $C_{ij}$, provided a straight line intersecting the circle defined by pose-estimation margins exists. If at least one such straight line actually exists, local coherence of the correspondence is then validated.

**Uncertainty in the binary constraint:** The same reasoning is applied for the binary constraint. For each couple of model vertical segments $L_i$-$L_j$, a set of synthetic image segment couples $l_i$-$l_j$, as represented by their respective measurement of intersection $v_i$-$v_j$ with the horizon line, is used. For each correspondence couple ($L_i$-$l_i$, $L_j$-$l_j$), random samples of $v^s_i$-$v^s_j$ belonging to a normal distribution of mean value $v_i$-$v_j$ and standard deviation matrix $\boldsymbol{\sigma_v}$ are generated, with the mean values and standard deviations of parameters $x^s_{cij}$, $y^s_{cij}$, $d^s_{ij}$ of the corresponding constraint arc being computed offline. During the online application of geometric constraints on a hypothetic correspondence ($L_i$-$l_i$, $L_j$-$l_j$), the arc of constraint parameters $x_{cij}$, $y_{cij}$, $d_{ij}$ is computed; the test then consists of searching inside the zone established by the standard deviations around $x_{cij}$, $y_{cij}$ and $d_{ij}$, provided an arc intersecting the circle defined by the pose-estimation margins exists.

### 3.5 Summary of the matching strategy

The complete matching process may be summarized by the following diagram (Figure 10):

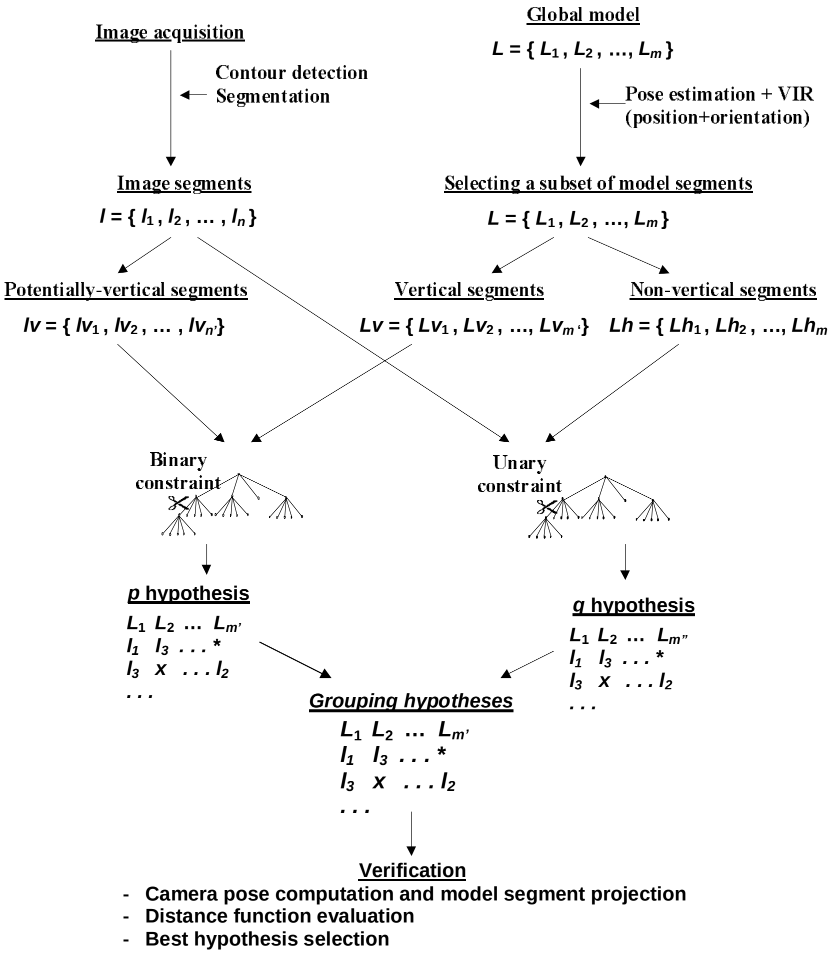

***Figure 10:** Summary of the correspondence search strategy*

## 4. Experimental evaluation

The method presented has been tested on both synthetic and real images. The goal of the test with synthetic data is to evaluate method performance from a statistical point of view using a great number of various practical situations. The test with real images and a real indoor environment serves to illustrate the feasibility of the approach under realistic conditions.

### 4.1 Evaluation with synthetic images

This test consists of generating synthetic views of a virtual indoor model from known poses and then running the proposed algorithm in order to both determine the correct correspondence combination and locate the camera.

A geometrical model of a hall composed of 65 edges was used for purposes of this test. A large number of camera poses are generated randomly; for each pose, a synthetic image is obtained by projecting the viewed model segments using a camera simulator. Specifications of the camera simulator are as follows:

- resolution = 640x480,
- intrinsic parameters $u_0 = 240$, $v_0 = 320$, $\alpha_u = -1000$ and $\alpha_v = 1000$.

Uncertainty due to calibration errors has been modeled by two noise parameters εm (multiplicative error) and εa (additive error), which belong to a Gaussian distribution with null mean values and respective standard deviations $\eta_m$ and $\eta_m$, introduced into the projection model as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} (1+\varepsilon_m)\alpha_u \dfrac{x}{z} + u_0 + \varepsilon_a \\ (1+\varepsilon_m)\alpha_v \dfrac{x}{z} + v_0 + \varepsilon_a \end{bmatrix}$$

In order to model noise due to an imaging process and line-extraction program, uniformly-distributed noise was added on the synthetic image line parameters as follows:

$$\begin{bmatrix} \rho \\ d \end{bmatrix} = \begin{bmatrix} \rho + \varepsilon_\rho \\ d + \varepsilon_d \end{bmatrix},$$

with $\varepsilon_\rho$ and $\varepsilon_d$ belonging to a Gaussian distribution with null mean values and respective standard deviations of $\eta_\rho$ and $\eta_d$. The actual values used in this test have been displayed in Table 1.

| Noise parameter | Value |
|---|---|
| $\eta_m$ | 0.02 |
| $\eta_a$ | 500 (*f*) |
| $\eta_\rho$ | 1.0 (°) |
| $\eta_d$ | 3 (pixels) |

***Table 1:** Noise parameter values used in the synthetic data test*

To simulate the presence of non-modeled objects in the scene, randomly-generated lines were added to the synthetic image. Initial pose estimations necessary to the algorithm were randomly generated as well. The number of model segments viewed in each image varied between 5 and 12; the number of added (non-modeled) segments also varied from 5 to 12. Images not containing sufficient viewed segments were discarded. A set of 430 images was ultimately selected for the test.

As mentioned, initial pose estimations necessary to the algorithm were generated randomly. A pose estimation quality (*QI*) parameter, which determines the difference between reference pose and estimated pose, was defined. Three series of tests were conducted by varying the *QI* value (see Table 2 below):

| *QI* | *1* | *2* | *3* |
|---|---|---|---|
| $\delta t$ (m) | 0.30 | 0.50 | 0.75 |
| $\delta\phi$ (°) | 10 | 15 | 20 |

***Table 2:*** *PEQ values used for the tests*

First of all, the capacity of the algorithm to identify a sufficiently-reliable combination of correct correspondences is studied. The results presented in Table 3 have been categorized into four groups:

- Success: a correct combination has been found;
- False negatives: no combination found;
- Consistent false positives: the proposed combination is composed of at least one false correspondence, but the interpretation is still globally consistent and the computed camera pose remains accurate;
- Inconsistent false positives: the algorithm has proposed a combination with several incorrect correspondences and the global interpretation is false.

| *QI* | 1 | 2 | 3 |
|---:|---|---|---|
| Success | 421 | 418 | 415 |
| False negatives | 5 | 5 | 5 |
| Consistent false positives | 4 | 7 | 7 |
| Inconsistent false positives | 0 | 0 | 3 |

***Table 3:*** *Synthetic data test results*

The second result concerns efficiency of the matching strategy. Figure 11 shows the number of matching hypotheses retained after pruning interpretation trees using both unary and binary constraints. Samples were ordered in accordance with the maximum number of combinations.

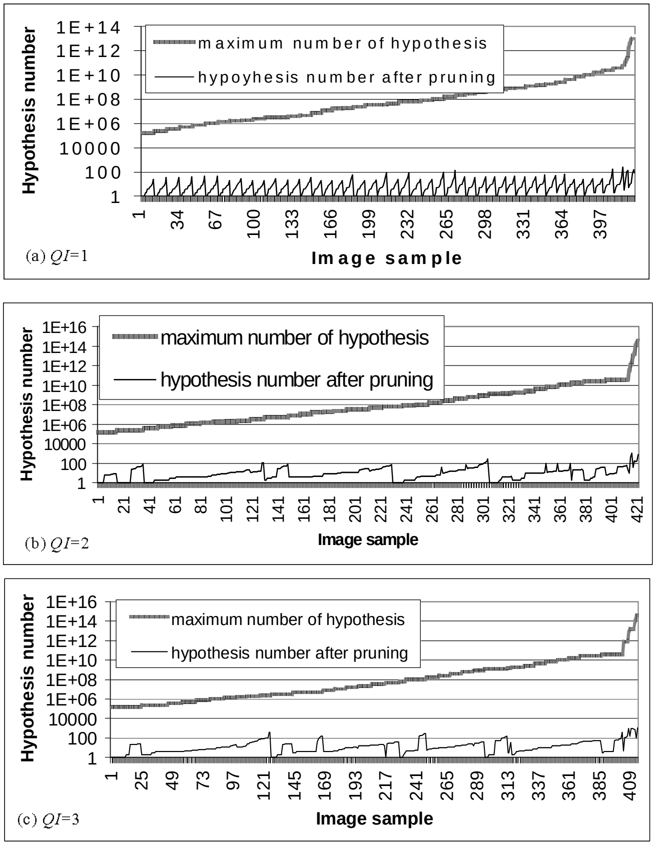

***Figure 11:** Pruning performance with unary and binary constraints*

### Results analysis

Two remarks can be forwarded based on the results obtained with synthetic data. First, the pruning strategy using unary and binary constraints offers a good level of selectivity. As shown in Figure 11, the number of retained hypotheses does not actually rise considerably with respect to the number of model and image segments. The exponential explosion in the maximum number of correspondence combinations is avoided. The second remark pertains to the capacity of the distance function in selecting the most globally-consistent correspondence combination. Table 3 reveals that the number of algorithm failures is very small (less than 2%, if the false negatives and inconsistent false positives are considered as failures with $QI = 3$). Furthermore, results show that method performance does not decrease when the quality of the initial pose estimate is deteriorated ($QI=3$).

## 4.2 Evaluation with real images

### Camera calibration

A SONY FCB-IX70P camera with a 640x480 resolution was used for this test. To obtain the intrinsic parameters of the pinhole camera model, the Zhang camera calibration method (Zhang 2000) was implemented; this method is better adapted to the mobile robotics context by virtue of its enhanced flexibility. As opposed to the classical Faugeras-Toscani method, the Zhang method does not degenerate with coplanar points. Several images of a planar pattern are introduced. The four intrinsic parameters $\alpha_u$, $\alpha_v$, $u_0$ and $v_0$, along with the extrinsic model for each view, are computed using a least-squares optimization technique. The transformation (rotation and translation) need not be known between the pattern poses in each view. This procedure enables modeling a large space around the camera without requiring very accurate or costly equipment. In addition, this method yields results comparable to those of classical methods. The intrinsic camera parameters are then computed using the Zhang method.

### Image feature extraction

The image contains a great amount of unprocessed information. For each robotics application, especially real-time applications, it is necessary to identify the parts of the image that contain pertinent information and then to represent such information in a form usable by the system. This is the role of low-level image processing. In the present work, contours have been extracted by means of a Canny filter. Among the contour pixels, linear chains longer than a set threshold are detected using a Hough transform of the image plane. To increase segment detection efficiency, the voting process in the Hough table is limited for each contour pixel to cells corresponding to a range of orientations defined by the gradient orientation of the pixel. Each chain is then represented by two parameters ($\rho$,$d$) obtained by polygonal approximation of the pixel coordinates (Horaud and Monga 1993).

### Experimental evaluation

The experimental test consists of locating the robot during a navigational session in the real-world environment that corresponds to the previous model. A sequence of images (some samples are presented on figure 12) is saved during a robot navigational session along a path defined in the navigation plane, as shown in Figure 13-a. The ground truth value is measured for each acquisition, with graduations being expressed in meters. For each image, the initial position estimate is represented by a black circle and the orientation margin by a blue angle. This initial estimate is obtained by updating the pose in light of the navigation commands; the value of $QI$ considered in the test is 3. Computed robot poses are symbolized by red triangles. Figure 13-b shows the wire frame of the hall. The edges selected thanks to VIRs are plotted in red. An example of an image used for location with correctly-matched segments is also shown in Figure 13-c. Results from the localization process are presented in Table 4.

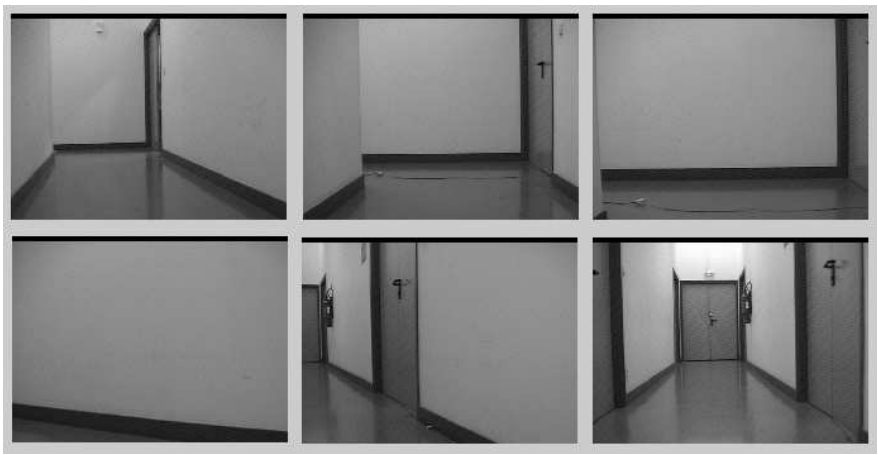

***Figure 12:** examples of images taken during the navigational session*

These results demonstrate that localization accuracy is maintained throughout the navigational session ($\delta\phi$ < 3° and $\delta T$ < 0.2 m). The number of correspondence combinations retained is quite low. The relatively poor initial pose estimation quality allows adopting a low sampling rate for image acquisition. As could be expected, an acquisition with insufficient features occurred during navigation (Image 8). The tolerance on initial pose estimation quality was then increased during the next acquisition, but this change did not affect algorithm performance.

### Results analysis

Results obtained with real images have confirmed expected performance, according to results of the test conducted using synthetic data. The algorithm served to efficiently update the robot pose. Performance in terms of localization accuracy satisfies the context requirement and allows the robot to accomplish automatic indoor navigational sessions.

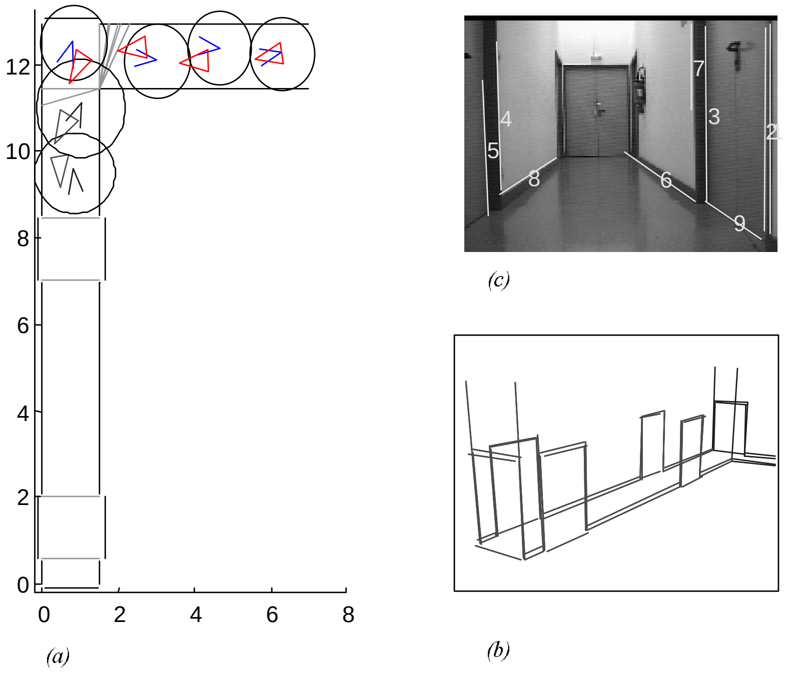

***Figure 13:*** *Localization process: (a) updating the robot pose (red triangle), starting from an initial position (black circle) and orientation estimate (blue angle); (b) selection of a subset of model features using VIRs; (c) viewed image and correctly-matched features*

| Image | Location error | | Hypothesis | Correspondence |
|---|---|---|---|---|
| number | $\delta\phi$ (°) | $\delta T$ (m) | number | number |
| 1 | 2.1 | 0.15 | 65 | 5 |
| 2 | 1.8 | 0.12 | 55 | 4 |
| 3 | 1.5 | 0.19 | 24 | 5 |
| 4 | 2.5 | 0.13 | 120 | 6 |
| 5 | 2.0 | 0.16 | 65 | 5 |
| 6 | 2.1 | 0.09 | 180 | 4 |
| 7 | 2.3 | 0.13 | 121 | 4 |
| 8 | - | - | - | - |
| 9 | 0.5 | 0.08 | 360 | 4 |
| 10 | 1.8 | 0.20 | 391 | 9 |

***Table 4:*** *Results of real image test*

## 5. Conclusion and subsequent work program

A method for incremental model-based localization of an indoor mobile robot using monocular vision and straight-line correspondences has been presented herein. All steps composing the localization process were studied, with special attention focused on the main contribution of this work, which consists of a strategy for the online matching of image and model correspondences.

According to simulation and experimental results, the previous objective of this work, which was to guarantee online matching search efficiency and absolute robot pose computing, has been attained. This efficiency is due to several factors. First, the interpretation tree subdivision enables avoiding the exponential explosion in the number of correspondence combinations. Second, the low order and good selectivity of geometric constraints make it possible to retain just a small number of correspondence combinations. In addition, these constraints are directly applicable on 2D-3D correspondences. Finally, accurate pose computing using each one of the small set of retained global interpretations serves to make a robust and efficient selection of the correct matching hypothesis.

Another advantage of this method is its ability to considerably reduce the sampling rate of absolute pose computing thanks to relative insensitivity to the initial pose estimate quality. This characteristic enables applying the method even without any dead-reckoning system. The necessary pose estimate can thus be obtained by merely integrating the commands generated by the navigation system.

Current interest is focused on introducing color and texture parameters to accelerate the matching process. These parameters can be used as additional constraints for testing the local consistency of hypothetical correspondences. Another area of concentration is the global localization problem, which consists of identifying a pose estimate when the robot is completely lost. This case may arise, for example, upon resetting the system. The approach entails introducing recognition using content-based image retrieval.

## 6. Appendix A

Let $l_i$ be an image line selected as possibly corresponding to a vertical edge (see figure 7) and viewed with a camera orientation $\boldsymbol{R}(\phi, \theta, 0)$ and translation $\boldsymbol{T}([x\ y\ z]^t)$. $v_{hi}$ is the coordinate of its intersection with the horizon line. This intersection point is the projection of a point $\boldsymbol{p_i}(x_i, y_i, z_i)$ on the corresponding model segment. In projecting $\boldsymbol{p_i}$ we have

$$(x_i', y_i', z_i') = \boldsymbol{R}(\boldsymbol{p}_i - \boldsymbol{T})$$

$$(u_{hi}, v_{hi}) = (\alpha_u . \frac{x_i'}{z_i'} + u_0, \alpha_v . \frac{y_i'}{z_i'} + v_0)$$

Since $z_i$=$z$ we obtain

$$v_{hi} = \frac{\alpha_v .((x_i - x).\sin(\phi) + (y_i - y).\cos(\phi))}{(-(x_i - x).\cos(\phi) + (y_i - y).\sin(\phi)).\sin(\theta)} + v_0$$

The goal is to calculate $v_i$ if $\theta$ was equal to 90°.

$$v_i = \frac{\alpha_v .((x_i - x).\sin(\phi) + (y_i - y).\cos(\phi))}{-(x_i - x).\cos(\phi) + (y_i - y).\sin(\phi)} + v_0$$

Thus $v_i = (v_{hi} - v_0).\sin(\theta) + v_0$

## 7. Appendix B

Let us consider a set $\{L_{v1}, L_{v2}, \ldots, L_{vi}\}$ of vertical model segments and their respective intersection measurements with the horizon line $\{v_1, v_2, \ldots, v_i\}$. Let $[\phi_{\min}, \phi_{\max}]$ be an estimation of the camera orientation and $\omega$the angle (figure 14).

The angle of view of a segment $L_{vi}$ is:

$$\omega_i = \arctan\left(\frac{v_i - v_0}{\alpha_v}\right) - \phi_{\max} - \omega$$

The order of apparition from right to left on the image of the segments is inversely proportional to $\omega_i$.

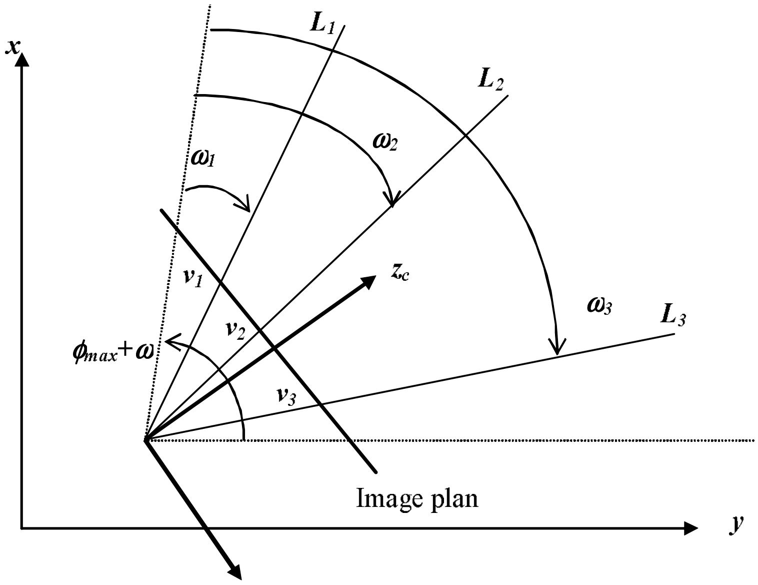

***Figure 14:*** *computing the left to right order of apparition of vertical segments on image*

### References

Ayache N., Faugeras O. and Hyper O. D. 1986. A New Approach for the Recognition and Positioning of Two-Dimensional Objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1): pp. 44-54.

Betke M. and Gurvits L. 1997. Mobile Robot Localization Using Landmarks. *IEEE Trans. on Robotics and Automation*, 13(2): pp. 251-263.

Borenstein J., Everett H.R., Feng L. and Wehe D. 1997. Mobile Robot Positioning: Sensors and Techniques. *Journal of Robotic Systems*, 14(4): pp. 231-249.

Chang C. C. and Tsai W. H. 1999. Reliable Determination of Object Pose from Line Features by Hypothesis Testing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(12): pp. 1235-1241.

DeSouza G. N. and Kak, A. C. 2002. Vision for mobile robot navigation : a survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(2): pp. 237-267.

Grimson W. E. L. 1990. Object Recognition: The Role of geometric Constraints. *MIT Press.*

Grimson W. E. L. and Huttenlocher D. P. 1990. On the Sensitivity of the Hough Transform for Object Recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(3): pp. 255-274.

Grimson W. E. L. and Huttenlocher D. P. 1991. On the Verification of Hypothesized Matches in Model-Based Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(12): pp. 1202-1213.

Grimson W. E. L. and Lozano-Perez T. 1987. Localizing Overlapping Parts by Searching the Interpretation Tree. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(4): pp. 469-481.

Guibas L., Motwani R. and Raghavan P. 1992. The robot localization problem in two dimensions. *In Proc. Third ACM-SIAM Symposium on Discrete Algorithms*: pp. 259-268.

Horaud R. and Monga O. 1993. Vision par ordinateur, outils fondatmentaux. *Hermes, Paris, France*.

Kosaka A. and Kak A.C. 1992. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *Computer Vision, Graphics and Image Processing – Image understanding*, 56(3): pp.271-329.

Lamdan Y. and Wolfson H. J. 1988. Geometric hashing: A General and Efficient Model-Based Recognition Scheme. *Proc. of Second ICCV*: pp. 238-289.

Lowe D. G. 1987. Three-dimensional object recognition from single two dimensional images. *Artificial Intelligence*, 31(3): pp. 355-395.

Munoz A. J. and Gonzalez J. 1998. Two-dimensional landmark-based position estimation from a single image. *In proc. IEEE int. Conf. Robotics and Automation*, pp. 3709-3714.

Murray D. W. and Cook D. B. 1988. Using the Orientation of Fragmentary 3D Edges Segments for Polyhedral Object Recognition. *International Journal of Computer Vision*, 2: pp. 153-169.

Pennec X. 1998. Toward a Generic Framework for Recognition Based on Uncertain Geometric features. *Videre: Journal of Computer Vision research, Quarterly Journal, The MIT Press*, 1(2): pp. 58-88.

Phong T.Q., Horaud R. and Tao P. D. 1995. Object Pose from 2-D to 3-D Point and Line Correspondences. *Int. Journal of Computer Vision*, 15: pp. 225-243.

Se S., Lowe D. G. and Little J. 2002. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research,* 21(8): pp. 735-758.

Simsarian K. T., Olson T. J. and Nandhakumar N. 1996. View Invariant Regions and Mobile Robot Self-Localization. *IEEE Trans. on Robotics and Automation*, 12(5): pp. 810-815.

Talluri R. and Aggarwal J. K. 1996. Mobile Robot Self-Location Using Model-Image feature Correspondence. *IEEE Trans. on Robotics and Automation*, 12(1): pp. 63-77.

Zhang Z. 2000. A flexible new technique for camera calibration. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 22(11): pp. 1330-1334.