Int. J. Appl. Math. Comput. Sci., 2005, Vol. 15, No. 1, 73–88

UNCERTAINTY MODELS OF VISION SENSORS IN MOBILE ROBOT POSITIONING

PIOTR SKRZYPCZYŃSKI

Institute of Control and Information Engineering
Poznań University of Technology
ul. Piotrowo 3A, 60–965 Poznań, Poland
e-mail: ps@ar-kari.put.poznan.pl

This paper discusses how uncertainty models of vision-based positioning sensors can be used to support the planning and optimization of positioning actions for mobile robots. Two sensor types are considered: a global vision with overhead cameras, and an on-board camera observing artificial landmarks. The developed sensor models are applied to optimize robot positioning actions in a distributed system of mobile robots and monitoring sensors, and to plan the sequence of actions for a robot cooperating with the external infrastructure supporting its navigation.

Keywords: mobile robots, vision sensors, positioning, uncertainty, action planning, multi-agent systems

1. Introduction

Reliable localization is a fundamental problem for mobile robots. Wheeled mobile robots use odometry, which is able to provide the robot with an estimate of its pose (position and orientation $\mathbf{X}_R = [x_r \; y_r \; \theta_r]^T$) at any time. This method alone is insufficient, and the pose has to be corrected by using measurements from exteroceptive sensors. There are numerous approaches to sensor-based localization known from the literature (Feng *et al.*, 1996). Unfortunately, self-localization methods require expensive on-board sensors (Castellanos and Tardòs, 1999). Currently, CCD cameras are the most compact and low cost sensors for mobile robots (DeSouza and Kak, 2002). However, most vision-based localization methods fail under common environmental conditions, due to occlusions, shadows, etc.

A solution for limited environments such as warehouses or factories is to develop an external infrastructure, which provides pose estimates to robots. Such an infrastructure was proposed by (Ishiguro, 1997) as a distributed vision system consisting of multiple cameras embedded in an environment. Stationary external cameras have also been successfully used to navigate experimental AGVs (Kruse *et al.*, 1998).

The modification of the environment is an obvious disadvantage. However, when a group of mobile robots shares a limited number of cameras mounted in carefully chosen locations (e.g., at corridor junctions, docking stations), the use of the external sensors is justified. Also artificial landmarks, which improve operational characteristics of on-board vision sensors, become more acceptable when a minimal number of cheap and unobtrusive visual cues is used (Bączyk *et al.*, 2003).

Localization based on the external infrastructure proposed here uses both fixed and on-board cameras, and exploits artificial visual cues in the form of passive, printed landmarks and active LED markers on robots. Due to these visual cues, simple and fast image processing methods could be employed, resulting in reliable and accurate positioning of mobile robots with respect to the global reference frame.

In a system with many robots performing their individual tasks, possibly along many very different routes, an optimal placement of stationary cameras and artificial landmarks, ensuring a complete coverage of the environment within the given uncertainty bounds, becomes infeasible. When artificial navigation aids are sparsely distributed in the environment and shared by several robots, an important issue is to ensure an appropriate strategy of positioning for particular robots.

This paper discusses methods used to ensure a reliable estimation of the mobile robot pose from artificial navigation aids sparsely distributed in the workspace, and shared by several robots working in a distributed system. The workspace is a closed environment and its general layout is known to the robots, so there is no need for exploration. Unexpected or moving obstacles (other robots) are handled by reactive navigation procedures. It is assumed that wheeled robots move in 2D on a flat floor. This work builds upon the author's previous results in distributed sensing (Kasiński and Skrzypczyński,

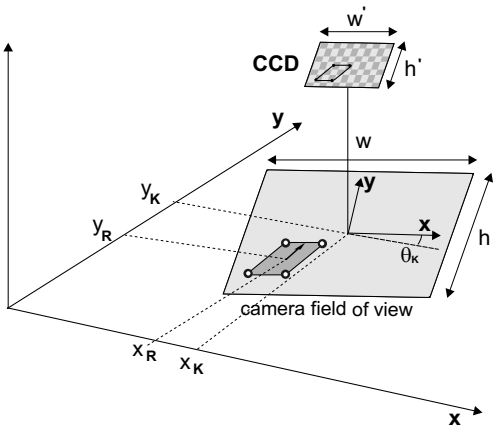


Fig. 1. Overhead camera system geometry.

1998; 2001) and sensor uncertainty modelling (Bączyk and Skrzypczyński, 2003; Skrzypczyński, 2004b). The main contributions of this paper are two methods aimed at minimizing the whole cost of positioning actions undertaken by robots, while ensuring the best available positioning quality (lowest uncertainty), or keeping the uncertainty bounded to the required value.

2. Vision-Based Sensors for Positioning

2.1. Distributed Overhead Cameras

The global vision system uses CCD cameras mounted to the ceiling. The cameras are equipped with wide angle (fish-eye) lenses, and their optical axes are orthogonal to the ground plane (Fig. 1). Low-cost *AverMedia* BT848 frame grabbers are used to digitize frames from standard B/W industrial cameras connected to PC computers being nodes of a LAN in the laboratory (Bączyk and Skrzypczyński, 2001). The entire image processing is performed in software on PCs. The extraction of robot characteristic points from grey-level images is hard and unreliable due to the complex shapes of the robots and the varying illumination conditions (Kasiński and Hamdy, 2003). Because of that, the *Labmate* robots have been equipped with active LED markers attached symmetrically to the corners. The detection of a robot is performed on the difference image, which is computed from a pair of images taken when the robot LEDs are on, and then off.

The procedure for the computation of the position and orientation of a mobile robot from overhead camera images consists of the following steps:

- Acquisition of two images, with the diodes on (Fig. 2A) and off, respectively.
- Elimination of the uneven natural lighting.
- Computation of the difference image and tresholding of this image (Fig. 2B).
- Cluster labelling and pruning of too small and too large clusters (Fig. 2C).
- Computation of the centres of the remaining clusters (Fig. 2D).
- Correction of the fish-eye distortions for the found points (Fig. 2E).
- Search for the pattern of points in the image, which matches the layout of the diodes on the robot (Fig. 2F).
- Computation of the position and orientation of the robot in the camera coordinates.
- Conversion of the found robot pose to the global coordinates.

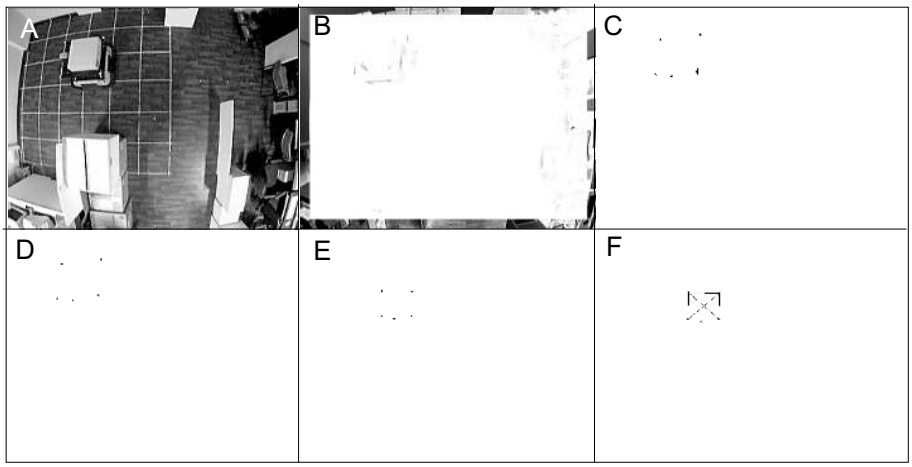


Fig. 2. Main stages of overhead camera images processing.

All operations in the robot recognition procedure are applied to the part of the whole image which is defined by the region of interest (ROI) (Bączyk and Skrzypczyński, 2001). The position of this ROI is computed from the previous estimate of the robot pose.

To eliminate the uneven scene illumination, a homomorphic filter was used (Kasiński and Hamdy, 2003). Thanks to this filter the illumination component of the images was suppressed. In many realistic situations the illumination component varies slowly, while the reflectance component changes rapidly. Using a high-pass filter, one can eliminate the illumination component, and obtain an image with an emphasized reflectance component. The separation of the components is not perfect since the reflectance component contains low-frequency residuals from the illumination component, but the difference image computed from two frames treated with the filter has much less noise than without the homomorphic filtration. The computation of the binary image $I_B(u,v)$ from a pair of frames $I_1(u,v)$ and $I_2(u,v)$ is described by

$$I_B(u,v) = \begin{cases} 1 & \text{for } |I_1(u,v) - I_2(u,v)| \geq I_T, \\ 0 & \text{for } |I_1(u,v) - I_2(u,v)| < I_T, \end{cases} \quad (1)$$

where u and v are the image coordinates of the pixel, and I_T is a given threshold of the pixel value. A group of connected pixels (cluster) in the image is potentially a representation of the sought object (a LED in this case). The labelling algorithm finds all connected pixels of an object, and assigns to them a unique value, called the label (Jain *et al.*, 1995). The groups of pixels which are too small or too large to represent a LED are deleted. The centres of the found clusters are established by computing the centre of mass:

$$\widetilde{u} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} iI_B(i,j)}{\sum_{i=1}^{n}\sum_{j=1}^{m} I_B(i,j)}, \qquad \widetilde{v} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} jI_B(i,j)}{\sum_{i=1}^{n}\sum_{j=1}^{m} I_B(i,j)}, \quad (2)$$

where $\widetilde{u}$ and $\widetilde{v}$ are first-order moments, which define the centre of a cluster in the $n \times m$ pixel image. In the above computations only the pixels which belong to a given cluster are taken into account.

To use a distorted image from the fish-eye camera for robot pose computation, this image has to be corrected by using a geometry transformation. The camera has to be internally calibrated (to exactly know its focal length (Heikkilä, 2000)), and calibrated with respect to the global coordinate system. The geometry correction procedure transforms image coordinates of the pixels to the positions these pixels would have if the picture was taken from a much greater distance, but by using idealized lenses (of a very long focal length), which do not introduce any geometric distortions (Fig. 3). There exist methods for the

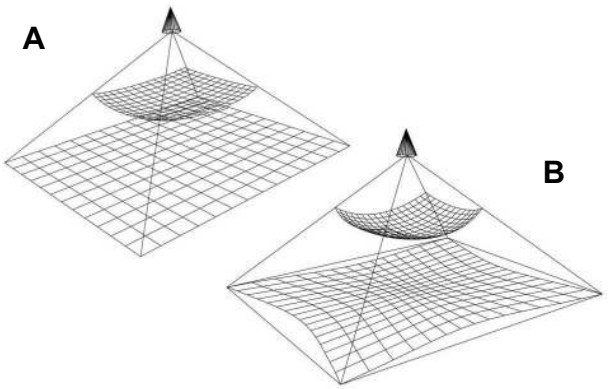


Fig. 3. Image geometry for a normal (A) and a fish-eye (B) lens.

fish-eye effect correction (Shah and Aggarwal, 1994), but here an original approach proposed by (Bączyk, 2001), called the spherical transformation, was used. In this simple method the image coordinates are expressed in the polar coordinate system, and only the distance of the given pixel from the centre of the image (the optical axis of the camera) is corrected.

In spite of its simplicity, the fish-eye correction of a full-frame image is quite time consuming. However, when the camera is used to detect positions of few clusters, which represent the robot LEDs, the correction can be applied after the initial stages of image processing, only to the remaining points being potentially the centres of the LEDs. Performing the correction only on few points makes the whole processing much faster.

When the corrected coordinates of the points being the centres of the found clusters are known, the set of four points, which is a representation of the robot diodes, is sought. To find the LED pattern, it is checked which of the corrected points fulfil the geometric constraints of the known dimensions of the LED pattern. Three LEDs must be visible to form a minimal detectable pattern (triangle). The distances between the found points are checked, as well as the angles between the line segments defined by the points. If the distance and angle conditions are fulfilled, the three found points form a right-angled triangle. All the found triangles are then checked whether or not they belong to the same rectangle defined by the four diodes on the robot. Depending on the number of the visible diodes (one of them can be occluded by the robot itself), the number of triangles is either 1 (three LEDs found) or 4 (all diodes visible).

The vector of the position and orientation of the robot in the image coordinates $\mathbf{U}_R = [u_r \; v_r \; \vartheta_r]^T$ is computed from the known positions of the LEDs. The centre of the LED pattern is computed for each found right-angled triangle:

$$u_r = \frac{1}{2}(u_A + u_B), \quad v_r = \frac{1}{2}(v_A + v_B), \quad (3)$$

where $\mathbf{U}_A = [u_A \; v_A]^T$ and $\mathbf{U}_B = [u_B \; v_B]^T$ are the image coordinates of the end points of the triangle's hypotenuse. The image coordinates of the centre can be averaged over all found triangles.

To compute the orientation of the rectangle constituted by LEDs, the coordinates of only two points are necessary:

$$\vartheta_r = \arctan\left(\frac{v_B - v_A}{u_B - u_A}\right). \tag{4}$$

When $u_A = u_B$, the robot orientation is 90°. If all four diodes are visible, the orientation can be computed also with respect to the remaining two points, and the results can be averaged. The transformations (3) and (4) can be written in a compact vector form:

$$\mathbf{U}_R = f_{uv}(\mathbf{U}_A, \mathbf{U}_B). \tag{5}$$

Once the position and orientation in image coordinates are known, the robot pose in the camera coordinates $\mathbf{X}_R^{'} = [x_k^{'} \; y_k^{'} \; \theta_k^{'}]^T$ can be computed by using the known dependence between the dimensions in the image and in the reality:

$$x_r^{'} = \frac{\left(u_r - \frac{w^{'}}{2}\right) w}{w^{'}}, \quad y_r^{'} = \frac{\left(v_r - \frac{h^{'}}{2}\right) h}{h^{'}}, \tag{6}$$

$$\theta_r^{'} = \vartheta_r,$$

where $w^{'}$ and $h^{'}$ are the dimensions of the CCD matrix, w and h are the dimensions of the field of view. The robot orientation remains unchanged. Then the robot pose $\mathbf{X}_R$ in global coordinates is computed by using the following equations:

$$\begin{aligned} x_r &= x_k + x_r^{'}\cos\theta_k - y_r^{'}\sin\theta_k, \\ y_r &= y_k + x_r^{'}\sin\theta_k + y_r^{'}\cos\theta_k, \\ \theta_r &= \theta_k + \theta_r^{'}, \end{aligned} \tag{7}$$

where $\mathbf{X}_K = [x_k \; y_k \; \theta_k]^T$ is the vector of the camera position and orientation in the global frame. The above relationship can be written in the following general form:

$$\mathbf{X}_R = f_{kx}(\mathbf{X}_K, \mathbf{X}_R^{'}). \tag{8}$$

2.2. On-Board Vision with Artificial Landmarks

The on-board vision system of the robot works on colour images. The artificial landmarks are made of A4 paper sheets. They have black orthogonal frame and bright green filling (Kasiński and Bączyk, 2001). There is a chessboard-like pattern placed inside. The particular fields of the chessboard are black or green according to the unique code of the given landmark. Up to 256 unique landmarks may co-exist in the environment.

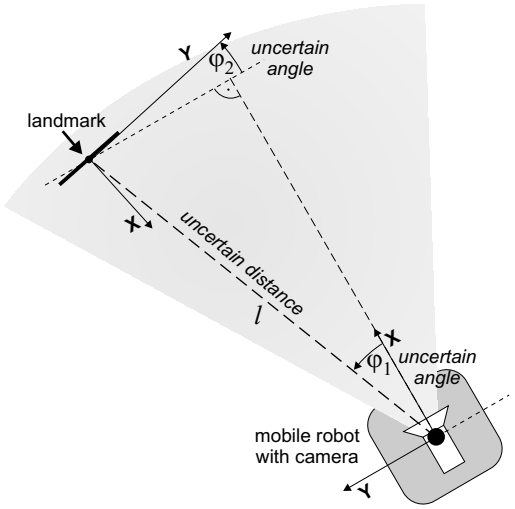


Fig. 4. Positioning with the landmark.

The landmark recognition process consists of three steps: the detection of ROIs in the image, the recognition of landmarks in particular ROIs, and the determination of image-coordinates of their reference points. ROIs, having the forms of rectangular windows, are determined by using colour segmentation methods (see (Bączyk *et al.*, 2003) for details). Further processing is restricted only to ROIs and based on the grey-scale image. The internal edges of the landmark frame are searched by computing gradient maxima. Straight lines are fitted to those maxima. Their crossections determine the picture-location of potential frame-corners with subpixel accuracy. Having determined the image coordinates of the landmark frame corners, one can establish the centres of chessboard fields, and read the landmark code.

During the robot pose computation, a pin-hole camera model is used. The camera is internally calibrated (Heikkilä, 2000). The image coordinates are expressed in [mm] by taking into account the physical dimensions and vertical/horizontal resolution of the CCD-matrix of the camera. The elements of the vector $\mathbf{P} = [x_1 \; y_1 \; x_2 \; y_2]^T$ containing the x-coordinates (x_1 and x_2) of the centres of the left and right frame edge, and the half-lengths (y_1 and y_2) of the vertical left and right landmark frame edge are calculated. These data, obtained from the image, are used to calculate the vector $\mathbf{L} = [l \; \varphi_1 \; \varphi_2]^T$ (Fig. 4) determining the robot pose with respect to the landmark:

$$l = v_l \frac{\sqrt{\lambda^2 + \left(\frac{x_1 y_2 + x_2 y_1}{y_1 + y_2}\right)^2}}{\frac{2 y_1 y_2}{y_1 + y_2}}, \tag{9}$$

$$\varphi_1 = -\arctan\left(\frac{\frac{x_1 y_2 + x_2 y_1}{y_1 + y_2}}{\lambda}\right),$$

where λ is the focal length, and v_l is the half-width of the landmark frame,

$$\varphi_2 = \arctan\left(\lambda \frac{y_1 - y_2}{-x_1 y_2 + x_2 y_1}\right). \tag{10}$$

The above transformations can be expressed in a vector form as

$$\mathbf{L} = f_{pl}(\mathbf{P}, \lambda, v_l). \tag{11}$$

As the locations of the landmarks in the global frame are known *a priori*, the computation of the robot pose in the global coordinates

$$\mathbf{X}_R = f_{lx}(\mathbf{X}_{L_i}, \mathbf{L}), \tag{12}$$

where $\mathbf{X}_{L_i} = [x_{l_i}\ y_{l_i}\ \theta_{l_i}]^T$ are the coordinates of the i-th landmark in the global frame, is straightforward:

$$\begin{aligned} x_r &= x_{l_i} + l\cos(\theta_{l_i} + \varphi_2 - \varphi_1), \\ y_r &= y_{l_i} + l\sin(\theta_{l_i} + \varphi_1 - \varphi_2), \\ \theta_r &= \theta_{l_i} - \varphi_2. \end{aligned} \tag{13}$$

3. Uncertainty of Vision-Based Positioning

3.1. Spatial Uncertainty Model

The robot pose uncertainty is described by a covariance matrix:

$$\mathbf{C}_R = \begin{bmatrix} \sigma_x^2 & \sigma_{yx} & \sigma_{\theta x} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{\theta y} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 \end{bmatrix}. \tag{14}$$

The uncertainty analysis uses a first-order covariance propagation (Haralick, 1996), and is focused on the influence of the relative position and orientation between the robot and the elements of the external infrastructure on the uncertainty of the pose estimate. The uncertainty caused by the quantization error is considered. Errors due to electronic noise in the image are not taken into account, because they do not depend on the spatial configuration of the robot with respect to the landmark or camera. The covariance propagation is based upon the linearization of non-linear equations describing dependencies between the variables in the measurement process by Taylor series expansions, and the computation of the proper Jacobians. This is a standard approach in robotics (Smith and Cheeseman, 1987), used also in computer vision (see (Miura and Shirai, 1993), for instance).

The analysis enables us to predict pose uncertainty before any sensing action (i.e., before taking and processing an image), and to decide which overhead camera should provide the pose estimate, and/or which landmark should be used. Moreover, by evaluating the predicted uncertainty over a given area, it is possible to construct a kind of uncertainty map for the given external sensor or landmark. To construct a 2D grid map, uncertainty must be expressed as a scalar value describing how "good" a particular cell in the map is. To this end, the equiprobability ellipsoid computed from the $\mathbf{C}_R$ matrix was adopted. The ellipse obtained by projecting this ellipsoid on the floor plane shows the area which contains the robot position $\mathbf{X}_{R_{xy}} = [x_r\ y_r]^T$ with the given level of probability (Smith and Cheeseman, 1987). The whole 3×3 $\mathbf{C}_R$ covariance matrix defines an ellipsoid, while the ellipse is defined by the 2×2 sub-matrix which describes the uncertainty of the robot position $\mathbf{X}_{R_{xy}}$. The ellipse is centred on the nominal $[x_r\ y_r]^T$ values of the position vector. If the covariances are non-zero, the angle between the major axis of the ellipse and the x axis of the coordinate system is computed from

$$\varphi = \frac{1}{2}\arctan\frac{2\sigma_{xy}}{\sigma_x^2 - \sigma_y^2}. \tag{15}$$

Introducing $\tau = \sqrt{(\sigma_x^2)^2 + (\sigma_y^2)^2 - 2\sigma_x^2\sigma_y^2 + 4\sigma_{xy}^2}$, one can write the lengths of the major and minor axes of the ellipse as

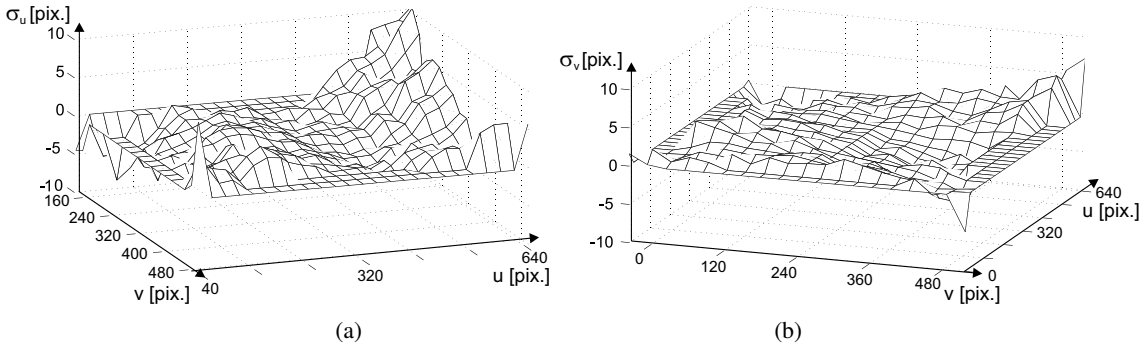
$$\begin{aligned} a_{maj} &= \sqrt{\frac{k}{2}(\sigma_x^2 + \sigma_y^2 + \tau)}, \\ a_{min} &= \sqrt{\frac{k}{2}(\sigma_x^2 + \sigma_y^2 - \tau)}, \end{aligned} \tag{16}$$

where k is a constant corresponding to the requested probability level

$$k = -2\ln\left(1 - P(x, y)\right). \tag{17}$$

For the probability of 95%, we have $k = 5.99$.

The above positional uncertainty ellipse does not directly capture the uncertainty of the robot orientation θ_r, but an improvement in the orientation estimate (e.g., due to a positioning action) influences indirectly the positional uncertainty in the subsequent path points, because elements of the robot state vector $\mathbf{X}_R = [x_r\ y_r\ \theta_r]^T$ are coupled through vehicle kinematics (Crowley, 1996). Having the positional uncertainty ellipse, one can deduce how pose uncertainty influences the clearance between the robot body and the surrounding obstacles, as was shown by (Moon *et al.*, 1999). However, in the uncertainty maps presented here the area (measured in [cm^2]) of the predicted ellipse for the 95% probability is employed as the positioning goodness value. Unlike the uncertainty in the distance computed along a given direction (e.g., the x or the y axis of the global coordinate system, which could be irrelevant to the current configuration of the obstacles), this area is a convenient overall scalar measure of positional uncertainty, which can be computed in advance



(a) (b)

Fig. 5. Distribution of errors in u (a) and v (b) of a corrected fish-eye image.

without full knowledge about the geometry of the environment, and can be used later to decide where positioning actions should be undertaken to obtain the best results.

Uncertainty maps are similar in concept to the sensory uncertainty field (SUF) proposed by (Takeda *et al.*, 1994), and then used by others, also for vision sensors (Adam *et al.*, 2001). However, the SUF is obtained by simulating the sensing many times. The uncertainty analysis presented below enables us to obtain a closed-form formula expressing the covariance matrix as a function of the robot configuration $[x_r\ y_r\ \theta_r]^T$ with respect to a given external navigation aid. Therefore, it is possible to predict uncertainty at run time or while planning using these equations. Such an approach is more flexible than the use of a pre-computed uncertainty field. Particularly, it is not necessary to define a constant grid resolution for uncertainty maps and, instead, the uncertainty measure can be computed for any instantaneous robot configuration.

3.2. Overhead Cameras

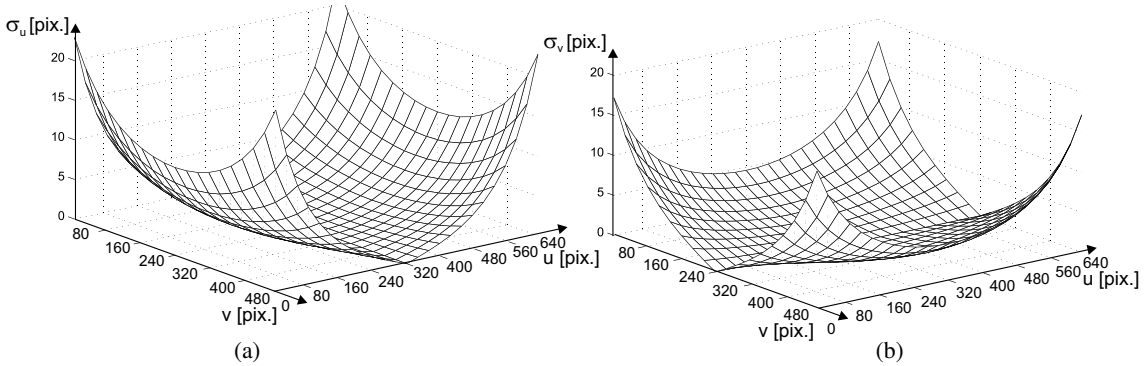
The spatial uncertainty of a robot localized by the overhead camera depends mainly on the uncertainty of the location of the points of the LED-pattern in the camera image. The correction of the fish-eye distortion results in shifting pixels from their original positions (Bączyk, 2001). Errors arise also because the assumption about the orthogonality of the optical axis to the floor plane is not perfectly satisfied. The spatial distribution of the errors in the pixel location (after correction) was evaluated by comparing the image of a calibration pattern with the ground truth. This experimental assessment of errors in the overhead camera images provides the primary uncertainty for the calculation of the estimated spatial uncertainty of the robot.

The input to the uncertainty computation procedure are the positions of the found diodes in the CCD matrix of the overhead camera $\mathbf{U}_D = [u_d\ v_d\ ]^T$, and their covariance matrices:

$$\mathbf{C}_D = \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}. \tag{18}$$

Because the centre of the robot is computed from (3), the coordinates of two diodes $\boldsymbol{U}_{D_A}$ and $\boldsymbol{U}_{D_B}$ are taken into account. The covariance matrices $\boldsymbol{C}_{D_A}$ and $\boldsymbol{C}_{D_B}$ are computed on the basis of the known errors of a single pixel location, which depend on the position of the given pixel in the CCD matrix:

$$\boldsymbol{C}_D(u,v) = \boldsymbol{C}_{mount}(u,v) + \boldsymbol{C}_{fish\_eye}(u,v), \tag{19}$$



(a) (b)

Fig. 6. Distribution of errors due to the 1^o non-vertical camera mounting, in u (a) and in v (b).

where the matrix $\boldsymbol{C}_{fish\_eye}$ represents the errors due to the fish-eye geometric transform, and $\boldsymbol{C}_{mount}$ represents the errors resulting from an imperfect vertical mounting of the camera to the ceiling. The elements of these matrices are evaluated upon primary uncertainty maps, which contain standard deviations σ_u and σ_v (in pixels) for all pixels of the CCD matrix, obtained experimentally. The experimental procedure was reported in detail in (Bączyk, 2001). The resulting primary uncertainty maps are depicted in Figs. 5 and 6 as 3D plots of the standard deviation values taken from the worst-case experimental results (Bączyk and Skrzypczyński, 2001).

To compute the covariance matrix $\mathbf{C}_{U_R}$ of the robot position and orientation in the image coordinates $\mathbf{U}_R$, the following expression is used:

$$\mathbf{C}_{U_R} = \mathbf{J}_{U_A}\mathbf{C}_{D_A}\mathbf{J}_{U_A}^T + \mathbf{J}_{U_B}\mathbf{C}_{D_B}\mathbf{J}_{U_B}^T, \qquad (20)$$

where $\mathbf{J}_U = \partial f_{uv}/\partial \mathbf{U}_R$ is the Jacobian of the transformation (5) with respect to $\mathbf{U}_R$. The robot position and orientation are computed from the positions of two LEDs, by using three equations. Therefore, the Jacobian is a 3 $\times$ 2 matrix.

The above computations describe the uncertainty of the robot pose in image coordinates. Then the resulting covariance matrix $\mathbf{C}_{U_R}$ has to be converted to a matrix describing uncertainty in terms of the camera coordinate system. Because the transformation (7) is linear, there is no need for an approximation:

$$\mathbf{C}_R^{'} = \begin{bmatrix} \sigma_{x'}^2 & \sigma_{y'x'} & \sigma_{\theta'x'} \\ \sigma_{x'y'} & \sigma_{y'}^2 & \sigma_{\theta'y'} \\ \sigma_{x'\theta'} & \sigma_{y'\theta'} & \sigma_{\theta'}^2 \end{bmatrix} \qquad (21)$$

$$= \begin{bmatrix} \frac{w}{w'} & 0 & 0 \\ 0 & \frac{h}{h'} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \sigma_u^2 & \sigma_{vu} & \sigma_{\theta u} \\ \sigma_{uv} & \sigma_v^2 & \sigma_{\theta v} \\ \sigma_{u\vartheta} & \sigma_{v\vartheta} & \sigma_{\vartheta}^2 \end{bmatrix}.$$

The uncertainty of the pose $\mathbf{C}_R$ in the global coordinates is computed from

$$\mathbf{C}_R = \mathbf{J}_K\mathbf{C}_R^{'}\mathbf{J}_K^T, \qquad (22)$$

where $\mathbf{J}_K = \partial f_{kx}/\partial \mathbf{X}_R^{'}$ is a 3 $\times$ 3 Jacobian matrix of the transformation (8) with respect to the vector $\mathbf{X}_R^{'}$. It is assumed that the camera position and orientation in the global frame are exactly known. Therefore, the uncertainty due to the pose vector $\mathbf{X}_K$ is not taken into account.

The use of several independent pose measurements (when all LEDs are visible) may lead to an improved pose estimate, which has smaller uncertainty when compared with that obtained from uncertain positions of only two diodes (according to (3)). If four diodes have been observed, and two different right-angled triangles have been built, then also two independent pose vectors and two covariance matrices can be computed. Using a simple static Kalman filter, one can fuse the pose estimates $\mathbf{X}_{R_1}$ and $\mathbf{X}_{R_2}$, taking into account their uncertainties $\mathbf{C}_{R_1}$ and $\mathbf{C}_{R_2}$:

$$\begin{aligned} \mathbf{K} &= \mathbf{C}_{R_1}(\mathbf{C}_{R_1} + \mathbf{C}_{R_2})^{-1}, \\ \mathbf{X}_R &= \mathbf{X}_{R_1} - \mathbf{K}(\mathbf{X}_{R_1} - \mathbf{X}_{R_2}), \\ \mathbf{C}_R &= (\mathbf{I} - \mathbf{K})\mathbf{C}_{R_1}, \end{aligned} \qquad (23)$$

where $\mathbf{X}_R$ is the fused pose, $\mathbf{C}_R$ is its covariance matrix, and $\mathbf{K}$ is the filter gain.

The resulting pose uncertainty in the global frame can be evaluated over the field of view of the overhead camera to build an uncertainty map. Figure 7 shows the positional uncertainty map computed in this way.

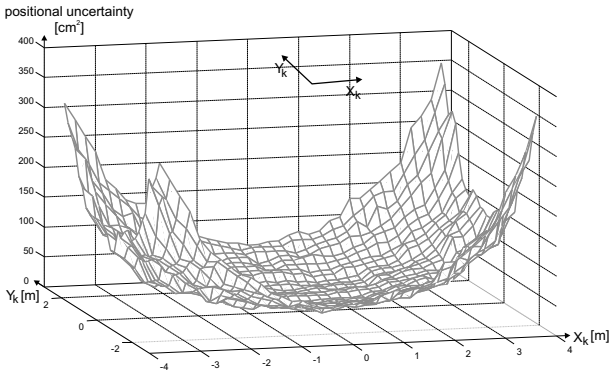


Fig. 7. Positional uncertainty as a function of robot location with respect to the overhead camera optical axis.

3.3. On-Board Cameras

The distances and angles between the camera and the landmark are computed from the relations between the known dimensions of the landmark pattern, and the dimensions of the image of this pattern appearing in the CCD matrix. The camera is a tool for measuring image dimensions. Measurement resolution is bounded by the CCD matrix pixel size. Although the positions of landmark reference points are computed with sub-pixel resolution, the standard deviation of this measurement equals the size of a pixel. This is the primary uncertainty, introduced by the limited resolution of the camera, and it is then propagated to the uncertainty of the vector $\mathbf{L}$ parameters, and then to the uncertainty of the robot pose $\mathbf{X}_R$. It is assumed that primary uncertainty depends on the errors in the computation of the coordinates $\mathbf{P} = [x_1\ y_1\ x_2\ y_2]^T$ in the image, and is expressed by the standard deviations

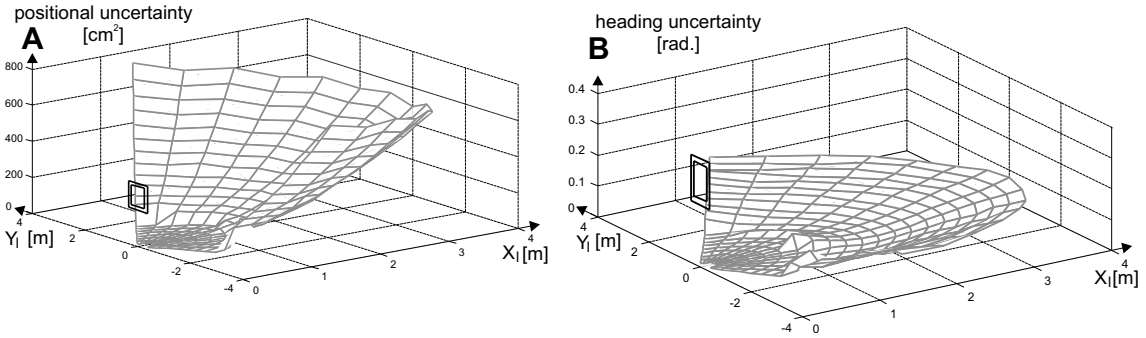


Fig. 8. Positional and angular uncertainty as a function of robot location with respect to the landmark.

$[\sigma_{x1}\ \sigma_{y1}\ \sigma_{x2}\ \sigma_{y2}]^T$. Thus, primary uncertainty is represented by the 4 $\times$ 4 matrix

$$\mathbf{C}_P = \begin{bmatrix} \sigma_{x1}^2 & 0 & 0 & 0 \\ 0 & \sigma_{y1}^2 & 0 & 0 \\ 0 & 0 & \sigma_{x2}^2 & 0 \\ 0 & 0 & 0 & \sigma_{y2}^2 \end{bmatrix}. \tag{24}$$

The standard deviation σ_y in the y coordinate is defined as the vertical discretization error

$$\sigma_y = \frac{h^{'}}{R_h}, \tag{25}$$

where R_h is the vertical resolution of the CCD matrix, and $h^{'}$ is its height (here R_h=480, $h^{'}$=3.6 mm). Similarly, the standard deviation σ_x in the x coordinate is defined as

$$\sigma_x = \frac{w^{'}}{R_w}, \tag{26}$$

where R_w is the horizontal resolution of the CCD matrix, and $w^{'}$ is its width (here R_w=640, $w^{'}$=4.8 mm). For the case under study, σ_x=σ_y, and thus all primary uncertainties are equivalent.

The uncertainty of $\mathbf{L}$ is described by the covariance matrix

$$\mathbf{C}_L = \begin{bmatrix} \sigma_l^2 & \sigma_{\varphi_1 l} & \sigma_{\varphi_2 l} \\ \sigma_{l\varphi_1} & \sigma_{\varphi_1}^2 & \sigma_{\varphi_2\varphi_1} \\ \sigma_{l\varphi_2} & \sigma_{\varphi_1\varphi_2} & \sigma_{\varphi_2}^2 \end{bmatrix}. \tag{27}$$

This matrix is computed from the primary uncertainty matrix $\mathbf{C}_P$ of the vector $\mathbf{P}$. The parameters of this vector constitute input data for the procedures calculating $\mathbf{L}$. The transformation (11) between $\mathbf{P}$ and $\mathbf{L}$ contains only the parameters of $\mathbf{P}$ and constants being the parameters of the landmark and the camera. Because this transformation is nonlinear, the covariance matrix $\mathbf{C}_L$ is computed as a first order approximation:

$$\mathbf{C}_L = \mathbf{J}_P \mathbf{C}_P \mathbf{J}_P^T, \tag{28}$$

where $\mathbf{J}_P = \partial f_{pl}/\partial \mathbf{P}$ is the 3 $\times$ 4 Jacobian matrix of the transformation (11) with respect to $\mathbf{P}$.

The last stage of uncertainty propagation shows how the uncertain distance and angles to the landmark influence the position and orientation of the robot in the global frame. The uncertainty in $\mathbf{X}_R$ is described by the covariance matrix $\mathbf{C}_R$ (14), which is a result of the uncertainty propagation from the vector $\mathbf{L}$ described by $\mathbf{C}_L$. Because the relation (12) between $\mathbf{L}$ and $\mathbf{X}_R$ is nonlinear, the matrix $\mathbf{C}_R$ is computed from the following approximation:

$$\mathbf{C}_R = \mathbf{J}_L \mathbf{C}_L \mathbf{J}_L^T, \tag{29}$$

where $\mathbf{J}_L = \partial f_{lx}/\partial \mathbf{L}$ is the 3 $\times$ 3 Jacobian matrix of (12) with respect to $\mathbf{L}$.

Figure 8A shows the positional uncertainty map for the artificial landmark, while Fig. 8B represents the standard deviation σ_θ, being the uncertainty in the robot heading.

4. Negotiation Framework for the Distributed System

4.1. Distributed System Structure

The vision-based positioning methods described in Sections 2.1 and 2.2 have been implemented in a distributed system of mobile robots and monitoring sensors

(Skrzypczyński, 2004a) to complete pose tracking methods relying on range sensing. The mobile robots and stationary cameras in this system are treated as independent, autonomous agents (Müller, 1996). The concept of agents is used to model mobile robots, overhead cameras, and the human operator interface.

The perception agent (PA) is conceptually similar to the Vision Agent proposed by (Ishiguro, 1997), which monitors the environment and provides various information to the robot agents (RA). Overhead cameras are the hardware part of perception agents, which localize robots with respect to the global reference frame. Perception agents compete for serving positioning data to robots. A specific agent in the system is the operator agent (OA), which initializes, configures and monitors robots and PAs.

Each robot uses its on-board range sensors to keep track of its own position and orientation. When the spatial uncertainty exceeds the acceptable value, or the robot fails to update its pose from the range sensing, it either uses its on-board vision to localize artificial landmarks or asks for the localization service from external agents. The on-board sensors are used also for local, reactive navigation (obstacle avoidance, etc.), which is implemented on mobile robots in a manner transparent to the positioning task considered here (Brzykcy *et al.*, 2001).

Each PA knows how positioning uncertainty depends on position of the robot with respect to the overhead camera of this agent. This type of knowledge is used in the negotiation mechanism based on the Contract Net Protocol (CNP) (Smith, 1980). Robot agents use this protocol to choose the best positioning data from the available sources. A dedicated, message-passing communication layer is used to support the information interchange between the agents (Kasiński and Skrzypczyński, 2002). The negotiation framework enables the robots to address the proper camera agent even if the robot does not know where the external cameras are placed (Bączyk and Skrzypczyński, 2003). This helps to make the distributed system open to modifications and robust to failures of particular external sensors.

4.2. Negotiation Protocol

For the positioning task, the message exchange is initiated by the robot which needs to know its position (manager). It sends to all Perception Agents (bidders) a request message. Whenever a PA is able to satisfy the request, it sends a bid message with the parameters informing the robot about the estimated localization uncertainty represented by the covariance matrix $\mathbf{C}_{RA}(PA)$ computed from the overhead camera uncertainty model presented in Section 3. It also signals how long it takes to complete the localization task, as the time T_{loc} depends on the number of localization requests already awaiting in the queue. An agent which is a manager in negotiations on the particular topic listens for bids, but does not accept requests with the same topic (i.e., requests for the same type of information/action), which prevents the protocol from looping. The manager waits for a prescribed time (depending on the number of PAs currently active in the system), to complete bids from all agents interested in pursuing the localization task (Kasiński and Skrzypczyński, 2002). The robot looks at the T_{loc} value, and excludes the bids from those PAs which are too busy. Then, the robot evaluates the uncertainty $\mathbf{C}_{RA}(PA)$ in the remaining bids to find the perception agent, which can localize it within the smallest positional uncertainty ellipse, assuming the 95% probability level. The robot compares proposals from particular PAs and awards the contract to the one which offers the best pose estimate, by sending an acknowledgement message. Next, the RA and the PA communicate in the peer-to-peer mode to transfer the localization data.

There are some tasks which cannot be executed by a single agent, because information from several agents is needed to complete the task. In such cases, the contractor may split up the task and award contracts to some other agents (such as an overhead camera-based perception agent). The agents operating in the presented system are not *a priori* designated as managers or bidders. Some of them can take both roles, depending on the current task context. Agents which cannot process data to yield results in the requested time (e.g., due to a sensor failure) are not considered as potential contractors by an agent that issued that particular task. For example, a robot-agent which wants localization data from camera-based perception agents can award the contract to another robot equipped with the on-board camera, whenever in the particular situation reliable pose estimate cannot be provided by the overhead cameras.

The communication mechanism is based on the User Datagram Protocol (Kasiński and Skrzypczyński, 2002). At the start of the system, the address of the operator agent is known to all agents. An agent entering the system contacts the OA, sending its symbolic name, type (RA or PA), IP address, and port number. This message also includes a list of specific tasks the agent can perform. The following positioning task types have been specified:

- `ovr_loc_active` – a perception agent can localize robots,
- `ovr_loc_passive` – a robot with LED markers can be positioned by a PA,
- `cam_loc_active` – a robot with a camera can localize landmarks,
- `cam_loc_passive` – a robot with a landmark can be positioned by another RA.

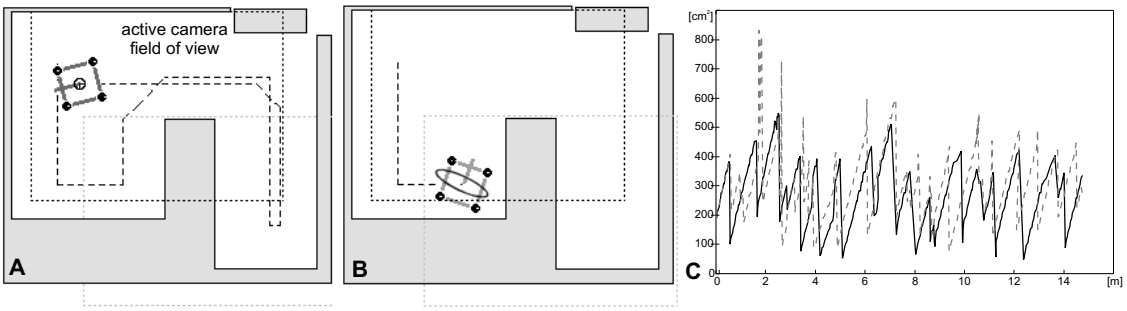


Fig. 9. Comparison of localization results, with and without negotiations.

The OA sends this information to all agents in the system. Because the agents know the specification of the tasks which other agents in the system can perform, it is possible to use point-to-point messages addressed to a subset of agents instead of broadcasting. Such an approach reduces the network traffic (Coulouris *et al.*, 1996) and allows us to skip the eligibility specification (Smith, 1980) in CNP announcement messages.

4.3. Results of Positioning with Negotiations

To show the advantages of negotiations between the robot agents and the overhead camera-based PAs, the results of two localization experiments are compared. The robot followed a predefined path (Fig. 9A), at an average speed of about 0.6 m/s, relying for localization only on its odometry and on the external camera agents. Figure 9C compares localization results as a function of the travelled distance for the case when the negotiation procedure is used (solid line), and the case when a simple choice of the "first available" PA is made (dashed line). The positional uncertainty ellipse area is used as the performance measure. When the robot did not use negotiation, but simply asked the first available camera-based agent to localize it, the number of localization requests was much bigger, and many of them were unsuccessful. Positional uncertainty exceeded at some points 700 cm^2 when the robot made several unsuccessful requests to the same agent. In many cases the robot requested the localization service being almost at the border of the field of view of the contacted PA (Fig. 9B), where positioning uncertainty is considerably higher (cf. Fig. 7), and occlusions of LED markers are more possible. The bid evaluation ensures that the robot uses the best offered localization service, and initializes localization when its position with respect to the field of view of the co-operating camera-based agent is acceptable. However, it should be noted that at some points of the path in Fig. 9C the uncertainty resulting from negotiations is higher than the one obtained with the naive strategy. This is caused by the local nature of optimization provided by the negotiations. The robot always uses the best positioning data available at a given point of the path, but the choice of the point itself is somewhat random – it depends on the current positional uncertainty, and hence on the whole sequence of the previous positioning actions. A robot which obtained a "better" position estimate at a given point can go for a longer distance without external positioning, and it could happen that it issues the next positioning request at a point which is very inconvenient to all PAs – negotiations do not help much in such a case. The observation of this problem gave rise to the global planning of the positioning actions sequence presented in the next section.

The next experiment demonstrates the ability of the distributed system to recover from sensing failures. Figure 10A shows a situation in which the robot `A` is located in a corner of the field of view of perception agents, while the robot `B` is near the centre of this field.

When the robot `A` needs to know exactly its pose, it sends a positioning request to the agents which are able to perform the `loc_active` tasks.

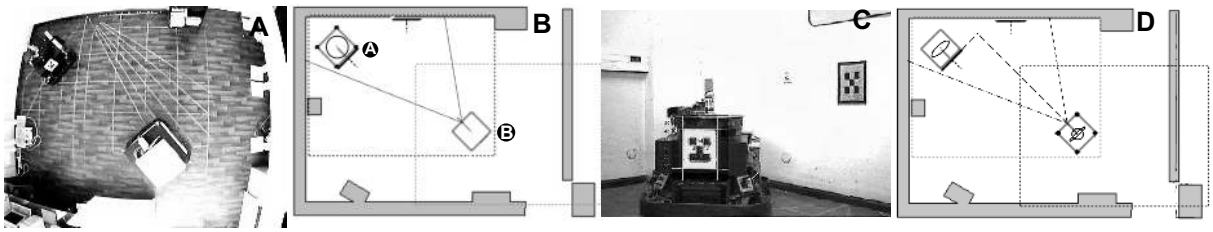


Fig. 10. Uncertainty in vision-based co-operative positioning.

The request contains the parameters $[x_0\ y_0\ \theta_0]^T$ of the current pose. One of the overhead-camera perception agents answers with a bid, but the predicted localization quality is low because of the location of the robot (Fig. 10B).

Another potential contractor is the robot B, equipped with an on-board camera. It predicts the relative pose uncertainty according to the model presented in Section 3.3, and estimates the time needed to find and localize the landmark attached to the robot A. However, to compute the pose of the robot with a landmark in the global frame, the robot with the camera needs to know its own pose. Although the robot B can use the artificial landmark attached to the wall to compute this pose, the predicted positional uncertainty is quite high, and the robot B sends a positioning request to the perception agents. The robot B receives a bid from the perception agent containing the predicted positional uncertainty, and accepts it because this uncertainty is small due to the robot position under the overhead camera. The robot with the camera computes the final predicted estimate of the robot A pose, and then sends the bid. The robot with the landmark evaluates the received bids and accepts the one from the robot B by sending an acknowledgement message. The contractor finalizes the contract with the perception agent receiving the current pose estimate with uncertainty information. Then it performs the actual positioning procedure by taking the image, recognizing the landmark attached to the robot A (Fig. 10C), and estimating the pose of this robot with respect to its local frame. At the final step, the robot B computes a new pose estimate $[x_n\ y_n\ \theta_n]^T$ of the robot A in the global frame (Fig. 10D).

5. Planning Positioning Actions

5.1. Motivation for Global Action Planning

If a mobile robot has a complete knowledge about the external cameras available in the system, and it also knows where the artificial landmarks are placed in the environment, it can compute in advance the optimal positioning strategy for the given path, which it has to follow. The aim of optimization is to minimize the time spent by the robot on communication and sensing actions (requests to Perception Agents and observations of artificial landmarks), which are necessary to keep pose uncertainty within some bounds.

In the literature, there are well-known path planning methods which take into account localization uncertainty (Latombe, 1991). However, most of these works assume continuous sensing of the environment (usually by some range sensors (Takeda *et al.*, 1994) and a complete, static environment model available to the robot. In most cases further simplifications are made, such as perfect sensors

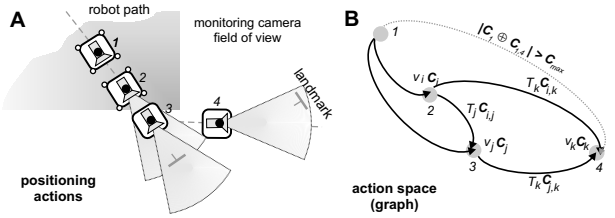


Fig. 11. Action space for positioning actions.

or landmarks providing constant pose uncertainty within a given region (Lazanas and Latombe, 1995). Few authors consider a probabilistic representation of uncertainty in planning (Lambert and Fraichard, 2000; Moon *et al.*, 1999).

Here a different approach is adopted, in which the sensing is opportunistic (i.e., the robot updates its pose only when it sees some landmarks or it is seen by an external camera), and the robot knows only the locations of the elements of the external navigation infrastructure: cameras and landmarks. Because the path is known in advance, the planning method yields only the positioning strategy, i.e., the sequence of positioning actions undertaken by the robot in order to travel to the goal, keeping pose uncertainty within bounds. The nominal path is computed in advance by a separate "geometric" planner taking into account the start point, the goal point, known obstacles and geometric limitations imposed by vehicle kinematics. The action sequence planning method relies on this geometric path planner for providing a feasible, collision-free route from the start to the goal. In the implementation an approach based on the generalized Voronoi diagram has been used (Takahashi and Schilling, 1989). This method yields reasonably short paths that give the robot maximal clearance around the obstacles. The geometric planning is performed for a disc-like shaped robot being able to turn on the spot, to reflects the basic features of the *Labmate* platform used in the experimental part of the research.

The planned sequence does not minimize the pose uncertainty over the whole path, but minimizes the overall cost of the positioning actions, ensuring that the positional uncertainty at any point of the path is lower than a given threshold. Such an approach is justified from a practical point of view: usually the robot needs only to be positioned with a given accuracy to perform its task (e.g., to pass a door), while optimal usage of the external navigation aids, deployed sparsely and shared with other robots, is of high importance. The optimization of the actions also enables the robot to achieve its goal position in a shorter time, because undertaking a positioning action requires to stop the vehicle – data cannot be gained while the robot is in motion. For overhead cameras this is caused mostly by the time needed for the acquisition of images (about 2[s]),

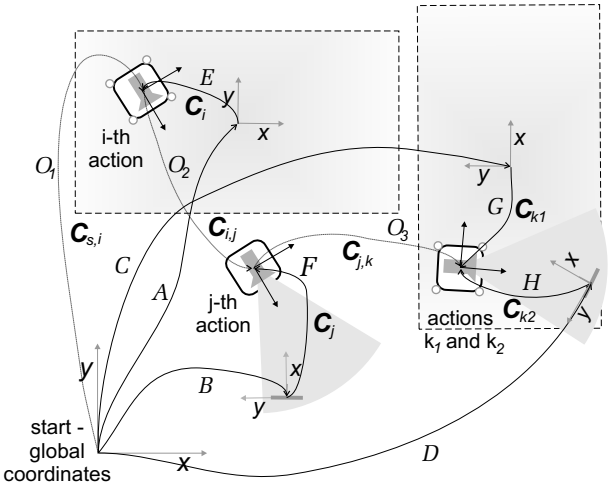


Fig. 12. Uncertainty sources of example positioning actions.

while for landmark recognition the image processing time dominates.

5.2. Construction of the Action Space

The proposed action planning framework is based on a classic approach to the search of a shortest path in a graph. The first step is to generate a discrete action space. The concept of this space is shown in Fig. 11A. At first, the nominal path of the robot is uniformly sampled, and the possible positioning actions which can be undertaken by the robot along this path are generated. An action at the i-th path point is described by the pose of the robot $\mathbf{X}_{R_i}$, the action type, the covariance matrix $\mathbf{C}_{S_i}$ describing the pose uncertainty resulting from a given type of positioning action performed in this particular configuration, and the cost of positioning T_i.

The covariance matrix is predicted according to the method described in Section 3, using closed-form formulae. The cost is an integer value generated upon a simple look-up table holding the experimentally determined time (in seconds) the robot spends at positioning depending on the type of sensing and/or communication action and the position with respect to the external navigation aid. Actions are nodes of the graph $G(V,E)$, $V = \bigcup_{i=1}^{n} v_i$ (Fig. 11B).

The node v_i is connected to the node v_j by an edge $e_{i,j}$ if it is possible for the robot to move from v_i to v_j, keeping pose uncertainty below a given threshold defined by the uncertainty ellipse area C_{max} (scalar value). As it is assumed that the robot may update its pose only at action nodes, the odometry model of a differential drive robot (Crowley, 1996) is used to compute the maximum admissible distance between two connected nodes. This model reflects the kinematics of the *Labmate* robot used in the experiments. If a different kind of robot platform (e.g., a synchro-drive) is being used, the odometry model should be changed to an appropriate one. Because the on-board camera is fixed to the robot body, several orientations at each point from which a landmark can be observed must be considered when the robot is turning (it turns on the spot). It is also possible that more than one positioning action are available at a given (x, y) point, e.g., the robot is under a ceiling-mounted camera, and it sees a landmark, or two landmarks are in the field of view of the camera. As a result, several nodes having the same $\mathbf{X}_R$ but different $\mathbf{C}_S$ and T can be generated. The edges of the graph are labelled with the costs of positioning actions. The edge $e_{i,j}$ has the cost T_j, according to the assumption that moving to a particular node implies the execution of the positioning action associated with this node. There is also a covariance matrix $\mathbf{C}_{i,j}$ associated with this edge, which reflects the growth of pose uncertainty while moving between the two nodes, due to odometry errors.

The sources of the uncertainty of the particular positioning actions are shown in Fig. 12, where the capital letters *A*,...,*D* denote poses of external infrastructure elements with respect to the global coordinate system, and *E*,...,*H* denote the particular poses of the robot with respect to overhead cameras and landmarks, while O_1,O_2,O_3 mean robot movements under the odometric control. The uncertainty evaluation along the robot path is achieved by compounding (denoted by $\oplus$) the serially linked uncertainties on the path, and by iterative merging (denoted by $\otimes$) of the obtained results with the uncertainties of the positioning actions. The compounding and merging operations are defined according to the proposals in (Smith and Cheeseman, 1987). For example, for the j-th action from Fig. 12 robot pose uncertainty before undertaking this action (i.e., before positioning by the observation of the landmark B) can be in this convention symbolically denoted by $(O_1 \otimes (A \oplus E)) \oplus O_2$.

The resulting action space is a directed graph $G(V,E)$. Because a positioning action can be performed only once by the robot travelling along a given path, the graph is acyclic.

5.3. Outline of the Planning Algorithm

Although a simple search in the action space will return the shortest path in the sense of a minimal action cost (minimal time), it cannot guarantee that positional uncertainty will be kept all the time below a given threshold. Some works on planning with uncertainty circumvent the problem by defining a graph whose edges have a modified cost being a combination of the actual cost (distance, time, etc.) and the uncertainty measure, with some scaling factors added (Lambert and Fraichard, 2000). With such an implicit trade-off (the values have different physical meanings), uncertainty becomes difficult to control.

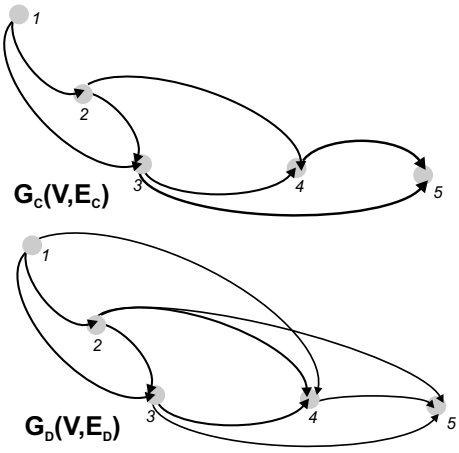


Fig. 13. Action graphs generated with two different methods.

The positioning action planning is treated as a constrained discrete optimization problem. An obvious solution is to construct the action space in such a way so that any path in the graph guarantees the required positioning precision from the start node v_s to the goal v_g. The positional uncertainty at the node v_k of a particular edge $e_{k,l}$ depends on the previous positioning actions undertaken along the path from v_s to v_k. However, assuming conservative initial uncertainty of an edge, which is yielded by the positioning action at v_k (known in advance), permits us to build a "conservative" graph $G_C(V, E_C)$. Two given nodes in this graph are connected by an edge $e_{k,l} \in E_C$ only if the merged uncertainty of the edge traversal (from odometry), and the positioning action undertaken at v_k is below C_{max}. The Kalman filter used to merge the pose estimates (Smith and Cheeseman, 1987) guarantees that the result is not worse than the best estimate taken as an input, while it is known that one of the input estimates at v_k has the uncertainty of $\mathbf{C}_{S_k}$. A search by means of the Dijkstra algorithm (Sysło *et al.*, 1983) with respect to the positioning cost T yields an optimal sequence of actions and guarantees pose uncertainty within the given bounds.

However, a robot can traverse between two action nodes keeping the uncertainty under C_{max} even if these nodes are not connected by an edge according to the above-mentioned conservative approach. The robot can achieve this by acquiring pose information in other nodes on its path, thus having the pose estimate at v_k (after merging) better than $\mathbf{C}_{S_k}$. As a result, the search in the conservative graph may result in a failure (no safe strategy found), even if a sequence of actions keeping the uncertainty below the given threshold does exist. An action space which permits a search taking into account the actual accumulated uncertainty can be constructed by using at the initial node v_k of a given edge the uncertainty value smaller than $\mathbf{C}_{S_k}$. The smallest pose uncertainty the robot can ever achieve is the uncertainty of the best (most precise) positioning action known to the system, i.e., the smallest $\mathbf{C}_{S_i}$ in the whole action space. When this uncertainty is used, the resulting graph $G_D(V, E_D)$ has the same set of nodes V, but more edges E_D, because starting with smaller $\mathbf{C}_S$ permits the odometry to take the robot further without violating the C_{max} constraint.

An initial approach to searching in the new action space was to modify the classical Dijkstra algorithm by adding a constraint satisfaction function, which checks whenever the move from the current node to the next node chosen by the search algorithm violates C_{max} (Skrzypczyński, 2004b). However, it turned out soon that such a method yields paths arbitrary suboptimal with respect to the cost. From a theoretical point of view, an effective search in $G_D(V, E_D)$ is equivalent to the restricted shortest path problem (RSP). This problem is known to be NP-complete (Ahuja *et al.*, 1993) but, recently, effective approximate algorithms have been developed to solve it, due to the importance of this problem for network routing with the Quality of Services (QoS) (Kuipers *et al.*, 2002).

Below, a pseudo polynomial dynamic programming solution to the action planning problem treated as an RSP is provided, which can be turned into an FPAS (fully polynomial approximation scheme) by using the approximation from (Lorenz and Raz, 2001). In the following algorithm, $\mathbf{C}[v, t]$ denotes a vector associated with each node v, which stores the minimum uncertainty on any path from v_s to v that has a total cost of t. T_{max} is the maximum cost of a path from v_s to v_g in the graph, obtained by a search with respect to the cost information only. It is a stop condition for the dynamic program. When the FPAS is used, T_{max} makes sure that the scaling error is not too large. $\mathbf{C}_{u,v}$ represents the uncertainty evolved by odometry while traversing the edge $e_{u,v}$, $\mathbf{C}_u$ is the uncertainty of the positioning action at node u. The notation $|\mathbf{C}|$ means the computation of a scalar value (ellipse area) from a given covariance matrix.

procedure
ACTIONPLANNINGASRSP($G(V, E), v_s, v_g, C_{max}, T_{max}$)
1 **for each** $v \in V - \{v_s\}$ $\mathbf{C}[v, 0]$:=∞
2 $\mathbf{C}[v_s, 0]$:=0
3 **for** $t := 1$ **to** T_{max} **do**
4 **for each** $v \in V$ **do**
5 $\mathbf{C}[v, t]$:=$\mathbf{C}[v, t-1]$
6 $P[v, t]$:=*nil*
7 **for each** $e_{u,v} \in E$ **do**
8 $\mathbf{C}_{temp}$:=$\mathbf{C}[u, t - T_{u,v}] \otimes \mathbf{C}_u \oplus \mathbf{C}_{u,v}$
9 **if** $|\mathbf{C}_{temp}| < |\mathbf{C}[v, t]|$ **and** $|\mathbf{C}_{temp}| \leq C_{max}$ **then**
10 $\mathbf{C}[v, t]$:=$\mathbf{C}_{temp}$
11 $P[v, t]$:=u *{update sequence}*
12 **if** $C[v_g, t] \leq C_{max}$ **then** RETURNSEQUENCE($P[v_g, t]$)
13 RETURNFAILURE *{T_{max} has been exceeded}*

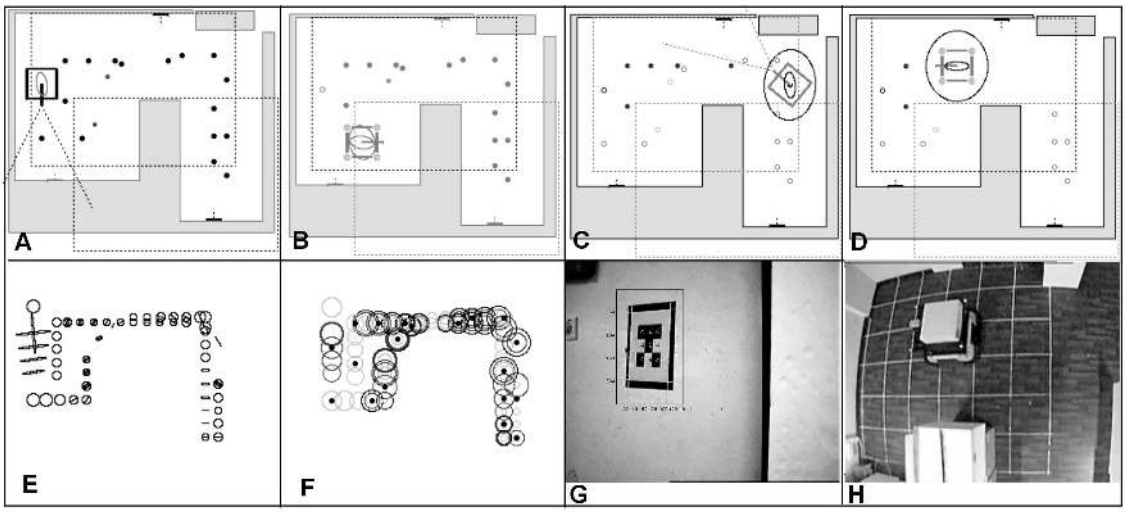


Fig. 14. Experimental results – the planned sequence of actions.

5.4. Action Planning Results

To validate the approach, experiments with the *Labmate* robot have been performed. The robot followed a preplanned path undertaking positioning actions according to the optimal strategy obtained from the above method. The experimental environment with two overhead cameras and three artificial landmarks is outlined in Fig. 14. Figures 14A–D show the simulated robot following the optimal sequence of positioning actions. For Figs. 14C and D the corresponding positioning actions of the real robot are shown in Figs. 14G and H. Figure 14E shows the predicted positional uncertainty ellipses for the generated positioning actions, in Fig. 14F the found optimal sequence of nodes (black dots) is shown – the circles in this figure have the diameter proportional to the positioning action cost, and the shade of grey means the type of the external localization aid used at this node (landmark, overhead camera, or both).

The results of the experiment are summarized in Fig. 15. From this figure it can be observed that the optimal sequence of positioning actions outperforms the simple strategy of using external localization aids. Using the simple strategy the robot observes its positional uncertainty. When the uncertainty exceeds the threshold value, the robot uses the on-board camera to measure the distance and angle to an artificial landmark or, if there is no landmark in the field of view, it asks for the positioning service from the external cameras. In the experiment, the threshold has been defined as the area of the positional uncertainty ellipse below 500 cm^2, for both strategies. From Figs. 15A and B it can be seen that the simple strategy does not guarantee the requested localization quality. In some areas the robot cannot perform a successful positioning action. If it enters such an area already with quite large uncertainty, the constraint is violated. The optimized strategy protects the robot from such failures by choosing good positioning actions before entering an area with lesser external aids. From Fig. 15C it is clear that the positioning cost (in this case it is simply the time) of the non-optimal strategy is higher, because the robot undertakes many unsuccessful actions, especially with respect to the external cameras. In such a case, the positioning cost grows, while the robot is still at the same point on the path.

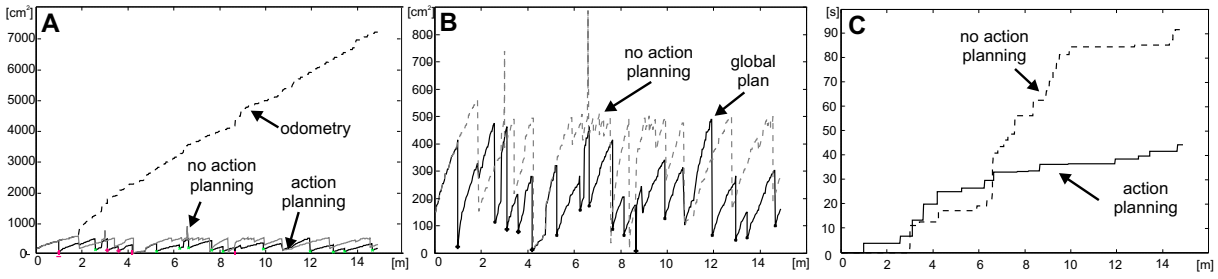


Fig. 15. Comparison of the results of using the planned actions and a simple positioning strategy.

6. Final Remarks

In this paper, methods of planning and optimizing the positioning actions of a mobile robot cooperating with an external navigation infrastructure have been presented. The infrastructure contains distributed overhead cameras and artificial landmarks enhancing the capabilities of on-board vision-based positioning sensors. Other types of stationary sensors (e.g., wall-mounted cameras) can be incorporated into this infrastructure if their uncertainty models are available.

The proposed negotiation protocol addresses the issues of extendability and fault tolerance in distributed systems, while trying to provide the best available positioning data to the robots. The global planning procedure sacrifices a part of this flexibility (the actual system configuration must be known in advance), but provides a mathematically solid method for obtaining an optimal sequence of positioning actions along a given path. This method takes into account in an exact way both the action cost and the positioning uncertainty. The action planning algorithm can be easily generalized to any other type of stationary sensor if the uncertainty related to positioning with respect to this sensor is described by closed-form formulae. A generalization to other types of robot vehicles is also possible, provided that the same geometric path planner can be used. The robot kinematics and sensing capabilities influence the construction of the action space (Section 5.2), which requires the usage of a proper odometry model, and an adequate definition of action costs. For example, if a robot is equipped with a pan/tilt camera, looking around for artificial landmarks no longer yields additional pose uncertainty. Because of this, unlike in the case of a fixed-camera robot considered in this paper, all possible positioning actions generated by changing the orientation of the camera will have the same pose uncertainty from the odometry. However, the costs of these actions can be different, because moving the camera to an angle more distant from the neutral position usually requires more time.

A more problematic issue is extending the action planning to a robot with severe kinematic limitations to maneuverability (e.g., to a car-like robot). The path planner for such a robot takes into account these limitations (see, e.g., Lambert and Fraichard, 2000), and produces a much more complicated geometric path which can contain also backward motions. Such a path cannot be handled by the current version of the positioning action planning method.

A subject of further research is also to extend the action planning procedure to a unified positioning action and a shortest/fastest path planning framework for robots cooperating with external sensors. This is possible when a network of collision-free routes is used to build the action space instead of a single path.

References

Adam A., Rivlin E. and Shimshoni I. (2001): *Computing the sensory uncertainty field of a vision-based localization sensor*. — IEEE Trans. Robot. Automat., Vol. 17, No. 3, pp. 258–267.

Ahuja R., Magnanti T. and Orlin J. (1993): *Network Flows: Theory, Algorithms and Applications*. — Englewood Cliffs: Prentice Hall.

Bączyk R. and Skrzypczyński P. (2001): *Mobile robot localization by means of an overhead camera*. — Proc. Conf. *Automation 2001*, Warsaw, Poland, pp. 220–229.

Bączyk R. (2001): *Methods of correcting image distortions in a localization system of a mobile robot*. — Proc. 7-th Nat. Conf. *Robotics*, Wrocław, Poland, pp. 185–194.

Bączyk R., Kasiński A. and Skrzypczyński P. (2003): *Vision-based mobile robot localization with simple artificial landmarks*. — Prep. 7th IFAC Symp. *Robot Control*, Wrocław, Poland, pp. 217–222.

Bączyk R. and Skrzypczyński P. (2003): *A framework for vision-based positioning in a distributed robotic system*. — Proc. Europ. Conf. *Mobile Robots*, Warsaw, Poland, pp. 153–158.

Brzykcy G., Martinek J., Meissner A. and Skrzypczyński P. (2001): *Multi-agent blackboard architecture for a mobile robot*. — Proc. IEEE/RSJ Int. Conf. *Intelligent Robots and Systems*, Maui, USA, pp. 2369–2374.

Castellanos J. and Tardòs J. (1999): *Mobile Robot Localization and Map Building. A Multisensor Fusion Approach*. — Dordrecht: Kluwer.

Coulouris G., Dollimore J. and Kindberg T. (1996): *Distributed Systems. Concepts and Design*. — Boston: Addison Wesley.

Crowley J. L. (1996): *Mathematical foundations of navigation and perception for an autonomous mobile robot*, In: Reasoning with Uncertainty in Robotics (L. Dorst, Ed.). — Berlin: Springer.

DeSouza G. and Kak A. C. (2002): *Vision for mobile robot navigation: A survey*. — IEEE Trans. Pattern Anal. Mach. Intell., Vol. 24, No. 2, pp. 237–267.

Feng L., Borenstein J. and Everett H. (1996): *"Where am I ?" Sensors and methods for autonomous mobile robot positioning*. — Tech. Rep., Univ. of Michigan.

Haralick R. M. (1996): *Propagating covariance in computer vision*. – Int. J. Pattern Recog. Artif. Intell., Vol. 10, No. 5, pp. 561–572.

Heikkilä J. (2000): *Geometric camera calibration using circular control points*. — IEEE Trans. Pattern Anal. Mach. Intell., Vol. 22, No. 10, pp. 1066–1077.

Ishiguro H. (1997): *Distributed vision system: A perceptual information infrastructure for robot navigation*. — Proc. Int. Joint Conf. *Artif. Intell.*, Nagoya, Japan, pp. 36–43.

Jain R., Kasturi R. and Schunck B. (1995): *Machine Vision*. — Singapore: McGraw-Hill.

Kasiński A. and Bączyk R. (2001): *Robust landmark recognition with application to navigation*. — Proc. Conf. *Computer Recognition Systems (KOSYR)*, Wrocław, Poland, pp. 401–407.

Kasiński A. and Hamdy A. (2003): *Efficient illumination suppression in a sequence by motion detection combined with homomorphic filtering*. — Proc. 27th Workshop AAPR *Vision in a Dynamic World*, Laxenburg, Austria, pp. 19–26.

Kasiński A. and Skrzypczyński P. (1998): *Cooperative perception and world-model maintenance in mobile navigation tasks*, In: Distributed Autonomous Robotic Systems 3 (T. Lüth *et al.*, Eds.). — Berlin: Springer, pp. 173–182.

Kasiński A. and Skrzypczyński P. (2001): *Perception network for the team of indoor mobile robots.: Concept, architecture, implementation*. — Eng. Appl. Artif. Intell. Vol. 14, No. 2, pp. 125–137.

Kasiński A. and Skrzypczyński P. (2002): *Communication mechanism in a distributed system of mobile robots*, In: Distributed Autonomous Robotic Systems 5 (H. Asama *et al.*, Eds.). — Tokyo: Springer, pp. 51–60.

Kruse E., Gutsche R. and Wahl F. (1998): *Intelligent mobile robot guidance in time varying environments by using a global monitoring system*. — Proc. IFAC Symp. *Intell. Autonomous Vehicles*, Madrid, Spain, pp. 509–514.

Kuipers F. A., Korkmaz T., Krunz M. and Van Mieghem P. (2002): *A review of constraint-based routing algorithms*. — Tech. Rep., Delft Univ. Technol.

Lambert A. and Fraichard T. (2000): *Landmark-based safe path planning for car-like robots*. — IEEE Int. Conf. *Robot. Automat.*, San Francisco, pp. 2046–2051.

Lazanas A. and Latombe J.-C. (1995): *Motion planning with uncertainty: A landmark approach*. — Artif. Intell., Vol. 76, No 1–2, pp. 287–317.

Lorenz D. and Raz D. (2001): *A simple efficient approximation scheme for the restricted shortest path problem*. — Oper. Res. Lett., Vol. 28, No. 5, pp. 213–219.

Latombe J.-C. (1991): *Robot Motion Planning*. — Dordrecht: Kluwer.

Miura J. and Shirai Y. (1993): *An uncertainty model of stereo vision and its application to vision-motion planning of robot*. — Proc. Int. Joint Conf. *Artif. Intell.*, Chambéry, France, pp. 1618–1623.

Moon I., Miura J. and Shirai Y. (1999): *On-line viewpoint and motion planning for efficient visual navigation under uncertainty*. — Robot. Autonom. Syst., Vol. 28, No 2, pp. 237–248.

Müller J. (1996): *The Design of Intelligent Agents: A Layered Approach*.— Berlin: Springer.

Shah S. and Aggarwal J. (1994): *A simple calibration procedure for fish-eye (high distortion) lens camera*. — IEEE Int. Conf. *Robot. Automat.*, San Diego, pp. 3422–3427.

Skrzypczyński P. (2004a): *A team of mobile robots and monitoring sensors – From concept to experiment*. — Adv. Robot. Vol. 18, No. 6, pp. 583–610.

Skrzypczyński P. (2004b): *Using sensor uncertainty models to optimize the robot positioning actions*, In: Intelligent Autonomous Systems 8 (F. Groen *et al.*, Eds.). — Amsterdam, IOS Press, pp. 299–308.

Smith R. and Cheeseman P. (1987): *On the estimation and representation of spatial uncertainty*. — Int. J. Robot. Res., Vol. 5, No. 4, pp. 56–68.

Smith R. G. (1980): *The contract net protocol: High-level communication and control in a distributed problem solver*. — IEEE Trans. Comput., Vol. 29, No. 12, pp. 1104–1113.

Sysło M. M., Deo N. and Kowalik J. S. (1983): *Discrete Optimization Algorithms with Pascal Programs*. — Englewood Cliffs: Prentice-Hall.

Takahashi O. and Schilling R. J. (1989): *Motion planning in a plane using generalized Voronoi diagrams*. — IEEE Trans. Robot. Automat., Vol. 5, No. 2, pp. 143–150.

Takeda H., Facchinetti C. and Latombe J.-C. (1994): *Planning the motions of a mobile robot in a sensory uncertainty field*. — IEEE Trans. Pattern Anal. Mach. Intell., Vol. 16, No. 10, pp. 1002–1017.

Received: 20 April 2004
Revised: 26 July 2004