Appeared in: 6th EuroMicro Workshop on Real-Time Systems,
June 94, Vaesteraas, Sweden, pp. 34-39

Supporting Real-Time Update of an Environment Representation for Autonomous Mobile Robots

Gunter MAGIN, Achim RUSS

Department of Process Control Computers, Prof. Dr. G. Färber
Technical University of Munich

Abstract – **A model of the environment is a mandatory requirement for the autonomy of a mobile robot. As the environment may change over time, methods are necessary to update the model information, keeping pace with the sensor frame rate. Our approach to cope with this real time problem is to predict virtual sensor images, which are then compared against the real perceptions. Two alternative methods are detailed: model based prediction and history based prediction.**

Index Terms – **sensor-specific environment modelling, spatial indexing, Z-buffer, feature tracking, matching**

1. Overview

Information flow

To ensure actuality, it is necessary to compare the sensor perceptions with the internal representation continually (Fig. 1). For each sensor frame we generate a synthetic feature image based on the current model of the environment, which limits the information to be processed in the real image. This principle is also used in [7].

Features e.g. line segments or other natural landmarks, can easily be extracted from the raw sensor data by fast preprocessing algorithms with the help of the synthetic feature map. Correspondences between the features of both lists are established in the match process, where three types of results are possible: features visible in both lists, features only visible in the synthetic picture, and features only visible in the real picture.

There is additional information from preprocessing telling how difficult it was to extract a feature, e.g. signal to noise ratio, and feature specific information, e.g. contrast, etc. Match results and preprocessing information are then filtered over time to eliminate transient errors. This means, for example, a change in visibility of a feature must occur in a couple of frames, before it is assumed permanent, and therefore integrated in the environment representation. After that, model based feature prediction supplies the match with only the ”good” features, and makes the verification process more robust. The match results can also be used to perform a localisation of the robot. A position estimation is necessary to calculate the visible features for a sensor at the current position.

Environment Representation

The environment of the mobile robot is modeled three-dimensionally by several layers describing the geometry of the world objects and their features. From the variety of ways to model real world objects [2] an attributed surface boundary representation (e.g. figure 2) was choosen.

Thereby the geometry of the objects is approximated by convex polygons which are attributed with surface normal and

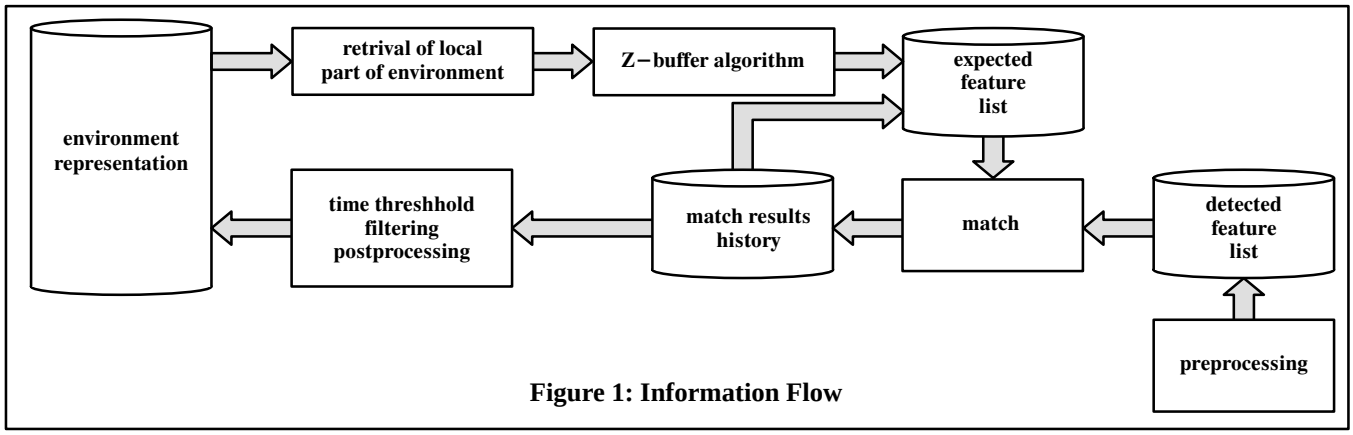

Figure 1: Information Flow

Appeared in: 6th EuroMicro Workshop on Real-Time Systems,
June 94, Vaesteraas, Sweden, pp. 34-39

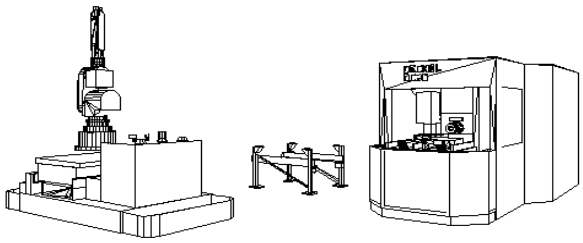

Figure 2: Manufacturing Scenario

material characteristics (e.g. reflection factor). This geometric description of the environment is common for all sensor systems besides specialities like glass doors which are seen by some sensors sometimes.

The feature layers - one for each sensory system - contain the corresponding features. The representation scheme of the features includes spatial and temporal uncertainties. These are dynamically modified by the matching results between predicted and real images, so that the environment model can also be used in non-static environments.

Modelling of the Sensor Systems

The models of the sensor systems specify how the synthetic sensor image is generated from the internal representation. There is a trade-off between quality of the predicted sensor images and the computation time needed. Experiments proved that for this application field it is sufficient to model the sensors as ideal systems, because the aim is to define regions of interest rather than photorealistic images. In case of video sensor system a simple pinhole camera model with radial distortions [4] is used. Resolution is 512 x 512. Range imaging sensors are also modelled as ideal systems, meaning that multiple reflections and ray expansion is not regarded. A 3D-Laser-Range-Camera [1] was modelled with a resolution of 321(81) x 41 and a scan range of 10 m and a 3D-Imaging-Microwave-Radar [6] was modeled with a resolution of 3600 x 1 and a scan range of 50 m.

2 Update Policies

Modification during Navigation Process

Those differences detectable in moving phases are typically slowly changing ones. Typical examples are shadow movements and illumination changes. The results of the match process are used to update the environment representation by incorporating the new feature properties and by modifying the features' spatial uncertainty. Fig. 3 shows a distance

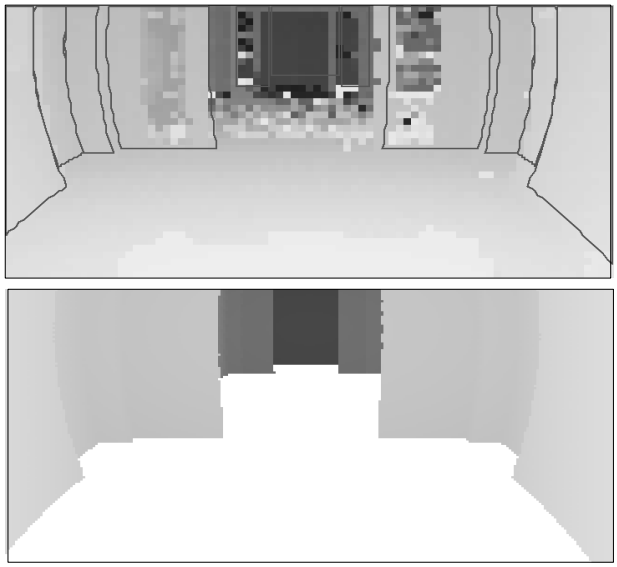

Figure 3: Real and Synthetic Laser Camera Images

image of a gangway scene shot with a laser range camera, and a synthetic image of the same scene, based on model information.

Modification during Collision Detection Process

When the robot is moving, it must check whether the place ahead is free to avoid a collision with unexpected obstacles. Therefore, the free space is queried from the environment model (Fig. 4). The sensors can now distinguish known from

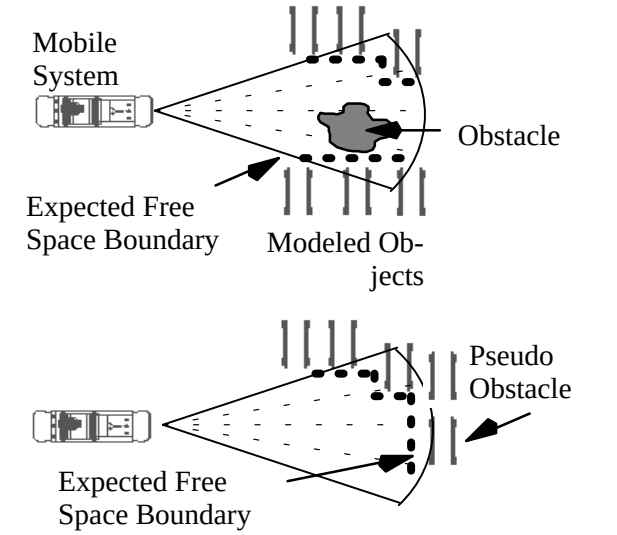

Figure 4: Collision Avoidance

new obstacles. Passing known obstacles is the responsibility of the robot's navigator module, and not covered here. Once an unexpected obstacle is detected, a seperate module tries to recognize it, using a-priori-knowledge of possible objects

in the environment. If this succeeds, the environment representation can be updated by inserting the new object.

3 Model Based Prediction

Spatial Indexing Mechanism

A typical manufacturing environment consists of hundreds of objects whose geometry is described by a total number of more than 50.000 polygons. The area overlooked by the sensor systems at a given point of view however contains only about 5% of the total number of polygons. The computation time for generating the synthetic sensor image can be decreased significantly if a spatial access mechanism is used to retrieve only locally relevant polygons for the following perspective projection process. Spatial geometric searching is similar to non-spatial multi-key searching although the used index structures must be suitable for non-zero sized spatial objects. The main requirement for a spatial indexing mechanism for use in autonomous mobile robots is selectivity, meaning that the part of the environment information that is overlooked by the sensor system should be retrieved as precise and fast as possible.

Access structures for extended spatial objects are surveyed in [5]. The authors suggest a hierarchical structure – the spatial-kD-tree (see figure 5) with k=2 or 3 for indexing non-

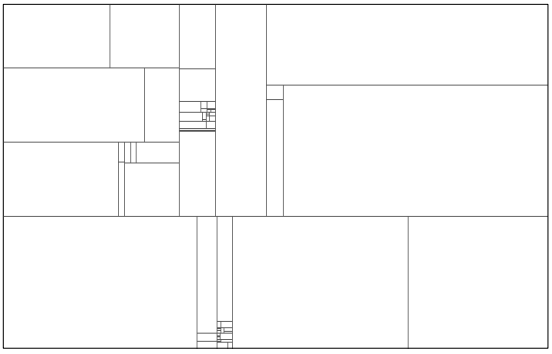

Figure 5:
Example for Subdivision by Spatial 2D-Tree

zero sized objects for geographic information systems. The applications in mobile robots differ in that in most cases the query region for the data base access is not a rectangle or a cuboid but a circular or spherical segment and that the data base is main memory resident – demanding low memory usage.

For dynamically sized query regions an implementation of the spatial-2D-search-tree has outperformed implementations of two indexing mechanisms based on dynamic hashing techniques: equidistant grid file and piecewise linear order preserving hashing [3]. The efficient realization of the algorithm that determines the subspaces overlapped by the query region showed to be one main aspect besides setting parameters for the index mechanism like minimum subspace area or maximum objects per subspace. The access time resulting for an environment consisting of 30.000 elements in an area of 1.200 m^2 for a circular segment shaped query region with a radius of 10 m and an angle of 60° is less than 30 ms.

The selectivity of the spatial-kD-tree becomes better if k=3 instead of k=2, so that the search space is divided into three dimensions. As a drawback the access time increases compared to k=2. In case of sensor systems like 2D-laser range scanners or microwave radar systems which cover the environment only in a parallel plane to the ground floor, the intersection test between query region (spherical segment) and subspace (rectangular parallelepiped) can be realized as two intersection tests between circular segment and rectangle. As a result access time is doubled only and overall the prediction of the sensor image is accelerated due to enhanced selectivity.

Prediction of Sensor Images

The prediction of the sensor image for a given point of view is done in a two-stage procedure with the model information retrieved via the indexing mechanism. The surface polygons of the objects are used to determine the visible parts of the sensor specific features of the objects from an estimated point of view. In the first stage hidden surface polygons are removed with the Z-buffer algorithm, which can easily be implemented in hardware. In a second stage the depth image resulting from the first stage is used to determine the visible parts of the feature while keeping the symbolic information of the feature, e.g. line from start point to end point.

The time consuming Z-Buffer algorithm for the hidden surface removal has been implemented on an Intel i860 microprocessor which hosts a special graphics unit for Z–Buffering. Due to the fact, that standard 'C'-compilers do not support the integrated graphic unit, minor parts of the software – less than 20 instructions – have been coded in assembler language. Figure 6 compares the computation time for

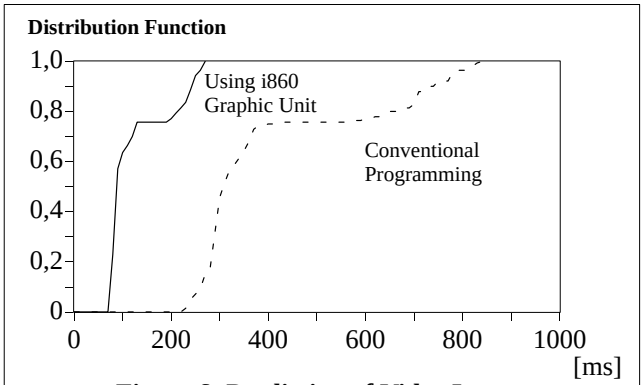

Figure 6: Prediction of Video Images

synthetic video images (resolution 512x512):

The standard Z-Buffer algorithm is originally designed for sensor systems like video sensor systems which use a planar surface for projection in contrast to range imaging sensors

which typically use spherical surface for projection. Because the Z-Buffer is the range image of the scene, images for range imaging sensors can be predicted by 'distorting' of the video sensor's Z-Buffer. Prerequisite is the correspondence of the discrete angular positions of the sensor beam to pixels of the video sensor for this uniform prediction (Figure 7).

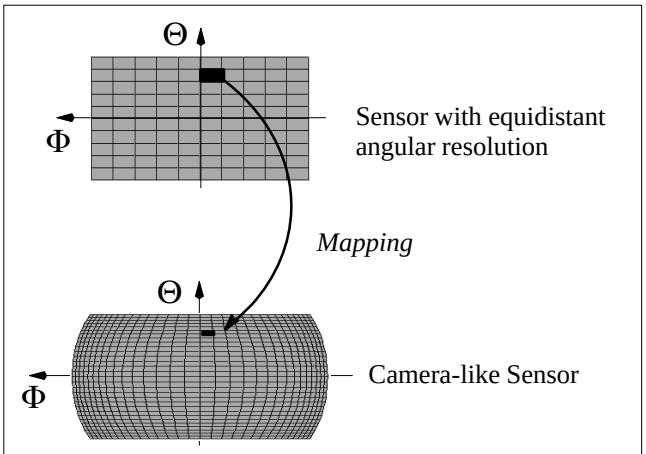

Figure 7: Uniform Prediction of Sensor Images

To obtain an error less than k range sensor pixels for a range imaging sensor with an equidistant angular resolution n_{range} the resolution n_{video} for the video image must be set as follows:

$$n_{video} \geqq 2 \cdot \frac{\tan\left(\frac{\alpha}{2}\right)}{\tan\left(\frac{k \cdot \alpha}{n_{range}}\right)}$$

α: Viewing Angle (Azimuth or Elevation)

Experiments showed that the drawbacks of the quantization errors caused by the Z-Buffer algorithm can be reduced to a negligible extend by artificially increasing the resolution of the sensor system. Experiments also showed that the computation of the sensor image is done in shorter time compared to ray shooting or object space algorithms despite of the increased resolution.

Regions of Interest

The predicted sensor images consists of a list of visible features defining regions of interest in the real sensor image to accelerate preprocessing of the sensor data. The size of these regions is determined dynamically by the spatial uncertainty of the current position and orientation (x_0, y_0, γ) of the mobile robot, the spatial uncertainty of the position of the feature (x_1, y_1) and the relative position between robot and feature. Figure 8 visualizes the basic idea for a vertical line feature assuming uncorrelated gaussian distribution of the spatial uncertainties.

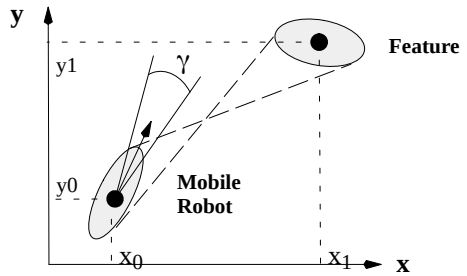

Figure 8: Dimensioning Regions of Interest

The main effect of the regions of interest is that local procedures can be used instead of global procedures when matching the synthetic and the real sensor image.

Real-Time Guarantee

In a moving robot the sensor images must be predicted within a given time interval, typically 0.25 - 0.5 seconds. The amount of computation time t_{total} is determined as follows:

$$t_{total} = t_{access} + n_{polygon} \cdot t_{polygon} + n_{pixel} \cdot t_{pixel\_polygon} + n_{feature} \cdot t_{feature} + n_{pixel} \cdot t_{pixel\_feature}$$

in quantitative terms:

$$t_{total} = 45\ ms + n_{polygon} \cdot 28\ \mu s + n_{pixel} \cdot 115\ ns + n_{feature} \cdot 4\ \mu s + n_{pixel} \cdot 5\ \mu s$$

After the data base traversal the number of retrieved polygons and features together with statistical knowledge about the mean size of polygons and features give a rough figure of t_{total}. If the expected computation time is above the real-time limit, the data base traversal is done again with a query region reduced using assumption about the spatial distribution of objects and features.

There is now a good chance to fulfill the real-time requirements, but no guarantee in critical moments. Therefore a new prediction of t_{total} is done after modeling transformation, clipping and rasterization with an error of the estimation of t_{total} which is now less than a few percent. The resolution of the predicted sensor image is now reduced according to the linear relation between the resolution (number of pixels) of the sensor systems and t_{total}. Figure 9 shows an exam-

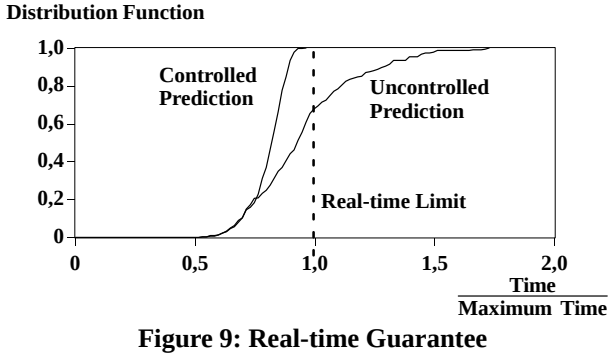

Figure 9: Real-time Guarantee

ple the combined effect of the two methods to meet the real-time requirements.

4 History Based Prediction

It seems to be unnatural to spend computation time to go through the procedure of model access, projection, etc. for every single sensor frame, as most of the produced feature positions differ only slightly from the results of the previous frame. This is because the features don't move far between successive frames, if the time between the shots is not too long. For instance, a translation of 5 cm into viewing direction results in a displacement of 2 to 5 pixel for a standard video camera.

So as an alternative we track the features to record a history of movement, and then predict, where each individual feature will show up in the next frame. The initial position is determined by a model inquiry or full frame preprocessing.

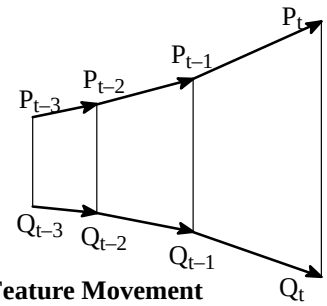

Figure 10: Feature Movement

Establishing correspondences for a feature in different frames is a crucial task in the tracking process. The number of candidates is limited by the feature's surrounding region of interest, just like in the model based prediction.

Motion Estimation Algorithm

As the movement of a feature in picture space is caused by the egomotion of the vehicle or a movement of an object in the scene or both, and those movements follow physical laws, movements of the features in the image plane follow the same laws, which are based on Newton mechanics, i.e. speed, acceleration, etc. There are exceptions, like blinking lights, which can of course not be covered.

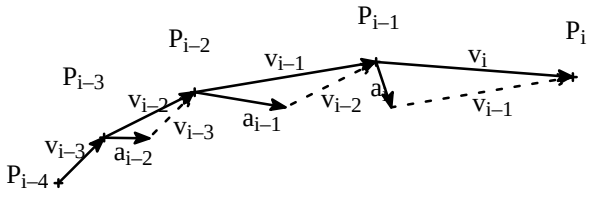

Figure 11: Feature's Speed & Acceleration

Feature history consists of the speed and acceleration in the image plane. Speed is determined by the position difference in successive frames, acceleration by the difference of speed. Higher order derivatives have not been found useful, as they do not contribute significantly to the prediction process, but add noise. Speed and acceleration of a feature in frame *i* is calculated as follows (see Fig. 11, where only one endpoint of an edge feature is shown):

$$\vec{v}_i = \frac{\vec{p}_i - \vec{p}_{i-1}}{t_i - t_{i-1}} = \frac{\vec{p}_i - \vec{p}_{i-1}}{\Delta t_i}$$

$$\vec{a}_i = \frac{\vec{v}_i - \vec{v}_{i-1}}{\Delta t_i} = \frac{\vec{p}_i - \vec{p}_{i-1}}{\Delta t_i^2} - \frac{\vec{p}_{i-1} - \vec{p}_{i-2}}{\Delta t_i \, \Delta t_{i-1}}$$

where p_i is the feature position. The frames are captured with a constant time interval, so Δt_j can be set equal to 1:

$$\vec{a}_i = \vec{p}_i - 2\,\vec{p}_{i-1} + \vec{p}_{i-2}$$

A new feature position is predicted by:

$$\vec{p}_{i+1} = \vec{p}_i + \vec{v}_i + \vec{a}_i = 3\vec{p}_i - 3\vec{p}_{i-1} + \vec{p}_{i-2}$$

This simple linear expression only contains the feature positions of the latest shots. The effort for evaluation is very low.

Correspondence Establishment

Solving the correspondence problem between features in successive frames is essential for feature tracking. Similar to the model based approach a region of interest is established around the predicted position in the sensor image, in which feature extraction takes place.

The size of this window must be large enough to ensure capturing of a feature even when movement starts to change (beginning of a curve after a long straight movement). On the other hand, it must be as small as possible to exclude correspondence candidates and to reduce preprocessing load. Horizontal and vertical window size is determined by the maximum of the feature acceleration and a constant value:

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \max \left\{ A \cdot \left| \begin{pmatrix} p_x \\ p_y \end{pmatrix}_i - 2\begin{pmatrix} p_x \\ p_y \end{pmatrix}_{i-1} + \begin{pmatrix} p_x \\ p_y \end{pmatrix}_{i-2} \right| , \begin{pmatrix} C_x \\ C_y \end{pmatrix} \right\}$$

A, C_x and C_y depend on the maximum possible feature acceleration and on the quality of preprocessing. A=3, C_x=C_y=5 have been found useful values.

Only the features found in this preprocessing window are taken for establishing a correspondence to the predicted feature, so the number to be checked is usually small. To find the best match out of those, feature properties are considered. For a video sensor, whose features are grey level edges, feature properties are the direction of the edge, amount and

sign of grey level gradient, properties of neighbour edges, etc.

For now, only a local algorithm for feature correspondence is used. It is sufficient in case of differential feature movements. If the time difference between shots becomes longer, or motion speed is increasing, a more sophisticated algorithm might be necessary.

Once the correspondence is established, feature history is updated. In case no correspondence for a feature could be established, the old history is extrapolated. After 3 successive frames without correspondence, this feature is considered lost. With this, temporal disturbances can be eliminated.

After a couple of frames, features disappear over the image borders. In case the number of tracked features becomes too small, a new model inquiry or another full frame processing phase is required.

5 Model Update

Access Functions

The presented concept of detecting changes in the environment by the evaluation of differences between predicted and detected features is based on incorporating minor changes in the environment - e.g. caused by varying illumination - by modifying attributes for spatial and temporal uncertainty of the environmental model's elements. This serves in a second function as a filter for transient irritations of the sensor systems.

The temporal integration of minor changes in the environment results in major changes in the environmental model executed by insertion and deletion of objects and features.

Real-Time Update

Modification of the attributes for spatial uncertainty of a line feature using Kalman filtering takes less than 3 ms in our implementation (Base: 20 SpecMarks). The insertion or deletion of an element of the environmental model takes about 0.2 ms.

6 Conclusion

Feature prediction in real-time to support the update of an environment representation is possible. The two methods, model based and history based prediction, have been presented, which have turned out to meet the real-time requirements of the sensor systems. The combination of both methods is subject of current research effort.

7 Remarks

The work presented is sponsored by the German Science Foundation (Deutsche Forschungsgemeinschaft), Bonn, Germany as part of the interdisciplinary research project "Information Processing in Autonomous Mobile Robots".

REFERENCES

[1] C. Fröhlich, F. Freyberger, G. Schmidt: "A Three-dimensional Laser Range Camera for Sensing the Environment of a Mobile Robot", in *Sensor and Actuators.* Elsevier Sequoia 1991.

[2] R.C. Jain, F.J. Besl: "Three-Dimensional Object Recognition", in *IEEE, Autonomous Mobile Robots*, Vol. 1, pp. 241–311, 1991

[3] H. Kriegel, B. Seeger: "Multidimensional Order Preserving Linear Hashing with Partial Extensions", in *Proceedings Int. Conf. on Data Base Theory.* pp. 203–220, 1986.

[4] R. Lenz: "Linsenfehlerkorrigierende Eichung von Halbleiterkameras mit Standardobjektiven für hochgenaue 3D-Messungen in Echtzeit". in *Informatik Fachberichte Bd. 149. Springer Verlag*, pp. 212–216, 1987.

[5] B.C. Ooi: "Efficient Query Processing in Geographic Information Systems". Springer Verlag, 1990.

[6] M. Rozmann, J. Detlefsen: "Environment Exploration based on a Three-dimensional Imaging Radar Sensor", in *Proceedings IEEE/ RSJ International Conference on Intelligent Robots and Systems*, Raleigh, 1992.

[7] D. Dickmanns, R. Behringer, V. v. Holt: "Road and Relative Ego-State Recognition", in *Proc. Conference on Intelligent Vehicles '92*, Detroit, 1992