Feature-based image segmentation for indoor robot-navigation

S. Weber[1], M. Krabbes[2], V. Stephan[1], H.-J. Böhme[1], H.-M. Groß[1]

[1] University of Technology, Ilmenau (Germany)
Department of Computational Neuroscience
98684 Ilmenau
stefan.weber@rz.tu-ilmenau.de
{vstephan,hans,homi}@informatik.tu-ilmenau.de

[2] Otto-von-Guericke-University Magdeburg (Germany)
Institute of Automation
krabbes@infaut.et.uni-magdeburg.de

Abstract: *This contribution presents an architecture, which enables collision-avoiding robot navigation in indoor-environments based on monocular visual information. This is achieved by neural linkage of image features with depth information, which can be extracted from the stream of visual data. A first approach only segments camera images into* obstacle *and* passable underground *regions with the exclusive help of suitably trained neural gases. An extended approach is introduced that merges the results of different segmenting procedures into a stable result. In this paper the efficiency of the algorithm will be described, problems in the real world application and their appropriate solutions will be pointed out.*

1 Introduction

In the past, very promising results could be achieved, when neural processing structures where applied to navigate autonomous mobile systems. The basic principle thereby mostly consisted in a direct feed-forward-processing of certain image features towards a motor control command. However, since from a single image no depth information is extractable, these procedures depend the presence of stable environmental features, which enable the visual sensor to distinguish between *passable ground* and *obstacles*. While this condition is fulfilled for road vehicles [7], in indoor environments stable relations of interpretable visual features and the desired driving path is hard to guarantee [2,4]. In order to develop the visual sensor for local robot navigation, it becomes necessary to combine the usable image features adaptively with the depth information available.

The hybrid architecture presented here (Figure 1) utilizes an *adaptive feature model* to

map depth information, retrieved from image sequences to other image features such as color and texture. Thus, depth information sufficient for collision-free navigation can be extracted from each image region at any time. But the model assumptions about the visual environmental characteristics that make *obstacles* distinguishable from *ground*, need to be developed.

The extraction of necessary depth information is accomplished by evaluating optical flow on inverse-perspectively transformed images [1]. The local image shift computed from each pixel determines whether this pixel belongs to the plane of motion (≙ *ground*) or it possesses an height offset to this plane of motion (≙ *obstacle*).

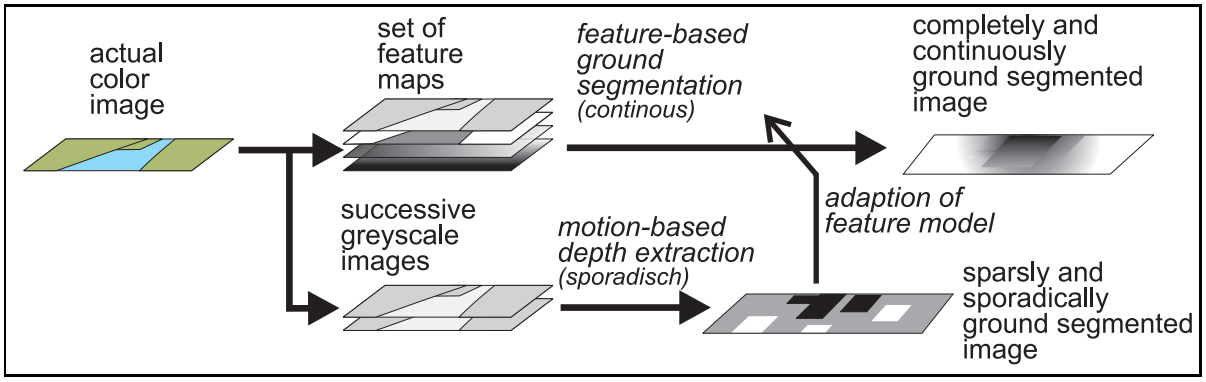

Figure 1: *Hybrid architecture: An adaptive feature model segments the images and is sporadically adapted to changes of environmental characteristics..*

2 Architecture

2.1 Inverse-perspective Mapping

Monocular visual navigation of an autonomous mobile robot is based on an image of the drive path from a camera (mounted tilted) onboard. The perspective distortions in the two dimensional image are eliminated by a so-called inverse-perspective transformation. According to a two-dimensional reference level (here the plane of motion of the robot), a transformation results in a virtual top view of this plane of motion (Figure 2.1). This top view homogenizes the movement dependent pixel shift for all pixel belonging to the ground plane. An obstacle sticking out of the motion plane is mapped with distortion and therefore produces a higher rate of movement (compared to the ground) in a sequence of several inverse-perspectively transformed images and can be segmented [5].

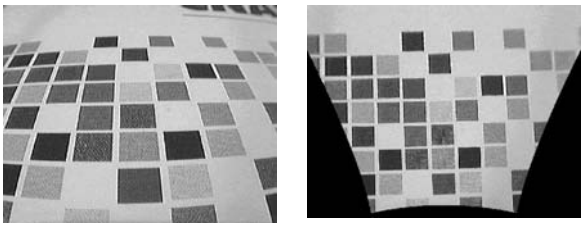

Figure 2: *Principle and effect of the inverse-perspective mapping. left: Original image, camera inclination: 45°; right: after compensation of lens distortion and elimination of perspective distortions a virtual top view results..*

2.2 Evaluating optical flow

For the comparison of the current image with a movement compensated image taken at an earlier point in time, a correlation-based displacement of pixel is detected (optical flow). The determination of local displacements will malfunction particularly in weakly textured image regions and a continuous calculation is time intensive. Additionally, a monocular configuration requires constant robot movements. Therefore the sporadic comparison of successive taken images extracts examples to train the adaptive feature model. However, for the ensuing training, a selection of the determined shifts according to their trustworthiness is necessary. Only those local shift vectors are used, where the difference of the minimal *sum of absolute differences (SAD)* between corresponding pixels exceeds a minimum value to the next higher SAD [8].
The implementation of such a hybrid architecture for visual robot navigation was investigated on two different systems:

- First, the miniature robot KHEPERA (Figure 3, left) was used in a special environment with structured walls. Its movement accuracy between two image recordings is not exactly controllable due to drive regulation. Hence it evaluates optical flow on the current image and a previous taken and movement compensated image.
- The second assigned robot system is MILVA (Figure 3, right), which operates in real indoor environments. The relative motion of the robot between two successive image recordings can be controlled very exactly. Using the movement information, the calculation of absolute differences of pixel values in two corresponding images becomes possible. However, this difference information is only interpretable in textured image regions. Therefore a SAD-based selection, using the algorithm described above, selects interpretable image regions.

The following allocation of shift vectors or pixel differences to one of the two classes *„ground“* or *„obstacle“* is based on plausible assumptions: All slight modifications between movement-compensated recordings are classified as *„ground“*; vectors with substantial vertical shift or all substantial pixel differences can be assigned to the class *„obstacle“*, since the effect of the larger shift described above is present here. This optical

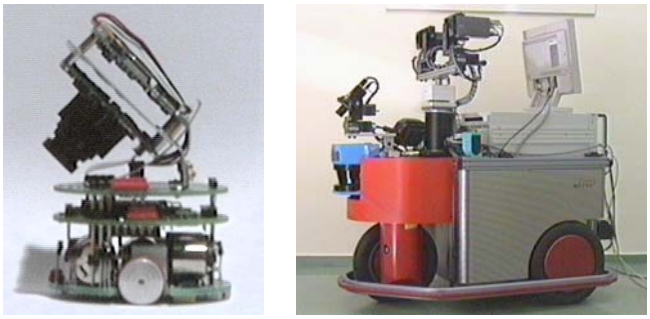

Figure 3: *The used robot platforms. Left: Miniature robot* KHEPERA *with installed inclined camera module; on the right: mobile robot platform* MILVA *(navigation camera: middle center)..*

flow based segmentation is then merged with the results of the other cues: segmentation using the **adaptive feature model** (Section 2.3) and a input prediction, derived from earlier segmentation results.

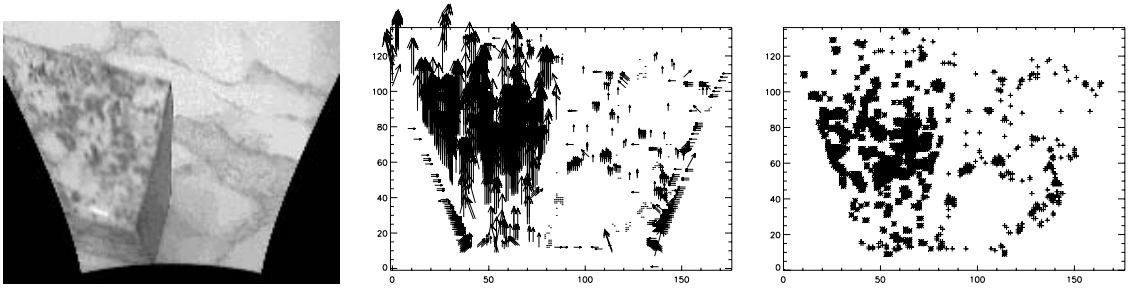

Figure 4: *Results of the processing steps on* KHEPERA. *Left: one of the two images used for shift calculation: the fine-structured obstacle to the left is to be recognized on the marbled ground; middle: selected vector field as result of the shift calculation; right: result of the vector classification (+ ground, * obstacle)..*

2.3 Adaptive feature model

The features of the two different classes are represented by one neural gas (NG) for each class [6]. The application of a separate, unsupervised vector quantifier for each class has proven to be more efficient than a single, supervised learning architecture with an output layer. Decoupling of the statistics of the training quantities in the two classes is achieved. This has advantageous effects, since the number of training examples for each class, extracted from an individual image, varies extremely due to different surface portions and occurring disturbances (e.g. brightness variation).

The model of KHEPERA The feature space consists of the following local image features, assignable to each pixel in the input color image: chromatic color (color information), standardized image coordinates (position information) and responses of three isotropic bandpass filters in successive frequency ranges (texture information). For

the *sporadic* training of the two NG's, a training pattern of features, that belongs to the position of a selected and classified shift or difference, is presented to the corresponding network as a training sample. For a representation of the 7-dimensional feature space, approximately 6 - 8 neurons in each network have been found sufficient.

The model of MILVA The feature space here only consists of the chromatic color (color information). Training examples for the *sporadic* training are obtained from the merging result of all the different segmentation algorithms. Therefore information of the other cues enters the feature model via training.

A *continuous* network-based image segmentation proceeds by assigning that class to each pixel of an input image, by whose NG the local image features, described above, are reproduced best. Thus, a pixel of the image belongs to the class of the NG with the smaller distance to the respective Best-Matching-Neuron. If both networks point to large reproduction errors, a still unknown object must be assumed.

3 Extension for real world use

The segmentation results, achieved in an artificial environment, demonstrates that an obstacle can be segmented from the ground by analyzing certain features. The optical flow supplies sufficient training information for the cluster algorithm. But for

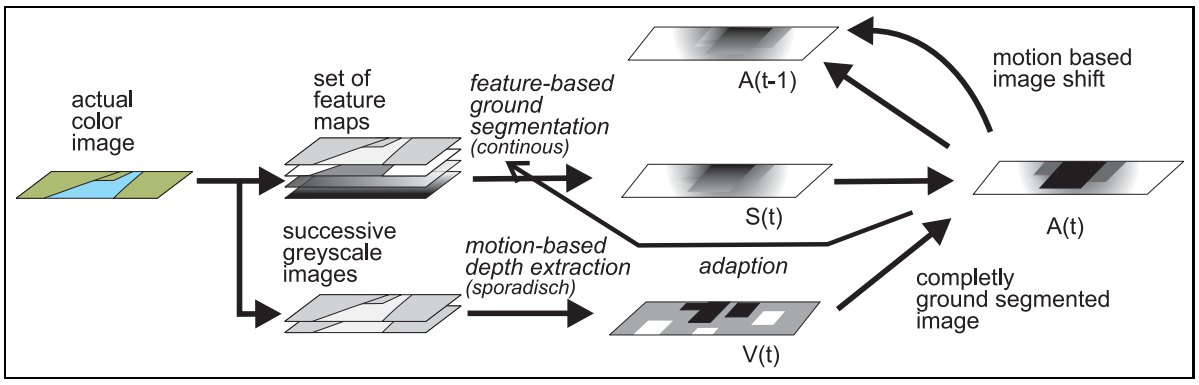

Figure 5: *Principle of suggested extended architecture. The final result is a merge of 1.) optical flow based image segmentation 2.) direct shift information and 3.) result of the preceding calculation step (movement corrected).*

real world application, an extension of the described hybrid architecture is suggested, whose implementation is presently investigated (Figure 5). In a feed back structure

all available *obstacle / ground* information flows into a common result, which then forms the basis for adapting the feature model: feature-based segmentation is fused with both the directly available movement information and with the final result of the previous step (shifted according to the relative motion of the robot). In this way the entire usable information can be used for the development of safe hypotheses to derive correct model assumptions.

The merging algorithm can be described in a difference equation,

$$A(t) = \alpha_1 \cdot P\left(A(t-1), \Delta x\right) + \alpha_2 \cdot V(t) + \alpha_3 \cdot S(t); \quad \sum_i \alpha_i = 1 \tag{1}$$

where Δx represents the distance the robot has moved since the last fusion result, $A(t-1)$ is the result of the previous step, $P\left(A(t-1), \Delta x\right)$ compensates movement, optical flow algorithms produce the pixel displacement $V(t)$, and $S(t)$ is the segmentation result of the adaptive feature model. $A(t)$ is the resulting coefficient dependent merge. First experiments with this architecture produced promising results.

4 Embedding the architecture in a navigation layer

As the segmentation result represents a virtual top view, distances to and between obstacles can be extracted directly. The distance information available can be fed into a global navigation layer (i.e. Occupancy-Grids [9]). However, in the ongoing work the resulting image serves as direct input activation for a neural network according to an ALVINN-like architecture, where segmented images are mapped with adequate robot control commands (i.e. front wheel angle (MILVA) or speed proportion for the two drives of a KHEPERA-robot). A *Multi Layer Perceptron* (MLP) is used to approximate the relation between the input image and the appropriate behavior generated during a test ride in the specific environment. The MLP consists of a 20×16 input retina, one hidden layer and a 25×15 output motor map. It is trained using ordinary *Back-Propagation* algorithms.

4.1 Generating Training Patterns

During a test ride along the institutes floor, more than 600 input patterns were recorded. It was tried to present the most common obstacles, such as walls and doors. The collected set of images was first invers-perspectively transformed and subsampled. The output set contained values for both speed and turn angle of the robot, as

they were chosen by an expert. These two data sets were given to the offline segmentation process, were the segmentation process described above took place. Every segmented image and its corresponding teacher vector were transformed into 6 different new patterns. After mirroring the image, both the original and the mirrored image were rotated with different angles. The desired angle was transformed due to the operations done on the images. For a final generation of the teach vectors, it be-

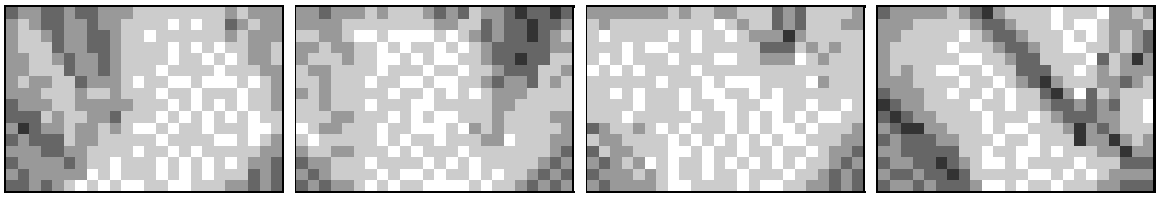

Figure 6: *A view typical input patterns for the MLP. Dark areas represent obstacles, such as walls and doors. White regions signalize passable ground. Resolution:* 24×16 *pixel..*

came necessary to code the two target values in a $speed \times turnangle$ motor map, using a two-dimensional GAUSS-function. The advantage of this two dimensional coding results in a separable output space for the single output neurons of the MLP. Finally a database, consisting of 3000 examples was created and divided into three different sets: a training set (1500 examples), a validation set (750 examples) and a test set (750 examples). The MLP was then trained with the mentioned training set. During the networks training, the database was presented in random order. After approximately 30 cycles of direct learning the MLP derived stable output assumptions. Despite the difficulties in an indoor environment, the architecture could learn to steer in the institutes floor and avoid obstacles.

The presented approach is able to extrapolate the feature based predication *ground/obstacle* onto image regions **without texture** and therefore without detectable shift information. Controllable calculation effort and the advantage of a single frame segmentation (despite the monocular configuration) provide the prerequisites for visual navigation in real environments, whereby the architecture is also portable to binocular camera systems.

References

[1] Bohrer, S.: Visual obstacle recognition by analysis of the optical flow in inverse-perspective scenes. VDI-Verlag (1994)

[2] Faeustle, P., Daxwanger, W., and Schmidt, G.: Controlling of local driving manoeuvres by direct coupling of illustrating sensor technology to an artificial neural network. congress volume from the 10. Autonomous Mobile Systems '94, Springer-Verlag (1994) 214–225

[3] Jaehne, B.: Digital image processing: concepts, algorithms, and scientific applications. Springer-Verlag, 3. Edition (1995)

[4] Krabbes, M., Boehme, H.-J., Stephan, V. ,and Gross, H.-M.: Extension of the ALVINN-Architecture for Robust Visual Guidance of a Miniature Robot. EUROBOT'97 – Proceedings of the 2nd EUROMICRO Workshop on Advanced Mobile Robots, IEEE Computer Society Press (1997) 8–14

[5] Mallot, H. A., Buelthoff, H. H., Little, J. J., and Bohrer, S.: Inverse Perspective Mapping Simplifies Optical Flow Computation and Obstacle Detection. Biological Cybernetics **64** (1991) 177–185

[6] Martinetz, T. and Schulten, K.: A "Neural Gas" Network Learns Topologies. Artificial Neural Networks – Proceedings of the ICANN'91, Elsevier Science (1991) 397–402

[7] Pomerleau, D. A.: Neural Network Perception for Mobile Robot Guidance. Kluwer Academic Publishers (1993)

[8] Stoeffler, N. O. and Faerber, G.: An Image Processing Board with an MPEG Processor and Additional Confidence Calculation for Fast and Robust Optic Flow Generation in Real Environments. Proceedings of the ICAR'97 (1997) 845–850

[9] Thrun, S., Fox, D., and Burgard, W.: Probabilistic Methods for State Estimation in Robotics. Proceedings of the Workshop SOAVE'97, VDI-Verlag (1997) 195–202