Visual Information Processing in Neural Architecture[*]

Werner von Seelen, Stefan Bohrer, Christof Engels, Walter Gillner, Herbert Janßen, Hartmut Neven, Gregor Schöner, Wolfgang M. Theimer, Bernd Völpel

Institut für Neuroinformatik, Ruhr–Universität Bochum, D-44780 Bochum, FRG
`institut@neuroinformatik.ruhr-uni-bochum.de`

Abstract. We report on the work done at the Institut für Neuroinformatik in Bochum concerning the development of a neural architecture for the information processing of autonomous visually guided systems acting in a natural environment. Since biological systems like our brain are superior to artificial systems in solving such a task, we use findings from neurophysiology and –anatomy as well as psychophysics for defining processing principles and modules that have been implemented on our mobile platform MARVIN. MARVIN is equipped with an active stereo camera system. On the basis of a *neural instruction set* for early information processing we define an *action for perception* approach. From the biological paradigm we use principles like active vision, foveation, two-dimensional cortical layers, mapping, and discrete parametric representations in a task-oriented way to solve problems like obstacle avoidance, path planning, scene recognition, tracking, and 3D perception. This paper has the character of an overview of the work done in this field at our institute.

1 Introduction

Autonomous systems are characterized by generating control decisions based on sensory perception of task and environment related features as well as accumulated knowledge. The interaction between an autonomous system and its environment is accomplished by using actuators or exchanging information.

The system should show the following properties:

1. The mobile robot must be able to adapt to variable environments and goals.
2. Solutions to simple problems should be used as preadaptations for complex ones leading to scalable complexity without re-design.
3. The degree of structural parallelism is high for *early vision* operations and is gradually replaced by functional parallelism at higher levels of information processing.

[*] Supported by the BMFT (ITR8800K4) and the VW–Stiftung (I/66354)

2 Design Principles and Hardware

As a general principle for the definition of a network structure, we advocate the view that for natural reasons small variations in the input signals correspond to small variations in the internal representations. Most of the advantageous properties of neural systems such as graceful degradation, fault tolerance and convergent self-organization stem from this isomorphic organization. This paradigm is further supported by the fact that natural evolution must lead to strongly causal systems. With preference to the evolution of information processing, strong causality amounts to the above constraint of isomorphic representation.

The following operations can be defined on the basis of cortical transformations. These operations are used as a "neural instruction set" to organize the "behavior" of our system.

1. Two dimensional maps with lateral interactions are used as space and time dependent filters which are either linear or nonlinear. By changing the scaling one can construct functional maps containing the dependency of two parameters expressed in 2D space and time.
2. The transport of data between different subsystems is combined with coordinate transformations. This topographic mapping is essential for a number of tasks.
3. A fovea is used to reduce the data flow as well as to organize - in a second design step - different representations in one layer. This is a special case of space–variant information processing as our general principle.
4. Discrete combination of various two–dimensional representations in one layer is a method to achieve functional specificity to the expense of spatial resolution with a task–dependent degree of parallelism.
5. Layering of two–dimensional nets combined with spatially three–dimensional operators is another way to combine representations.
6. Regularisation enables the use of continuity assumptions in reducing information.
7. A double camera system with three degrees of freedom for each camera acts as an active vision module.

The problems can be treated in continuous mathematics on the basis of a mean anatomy. The mathematical details are published in [vSBKT94].

The mobile platform and the cameras are two subsystems integrated into a unified control loop. We have developed a common command structure for both separate units. This approach guarantees a great flexibility due to its flexible open system structure. The whole system can easily be controlled by an external process that sends task-specific command sequences to the mobile robot. Furthermore the system is able to deal with multiple users simultaneously so that separated tasks can be performed, organized in a hierarchy of command priorities. More detailed descriptions of the hardware and software structure are given in [SBDF90] and [DFSKT94].

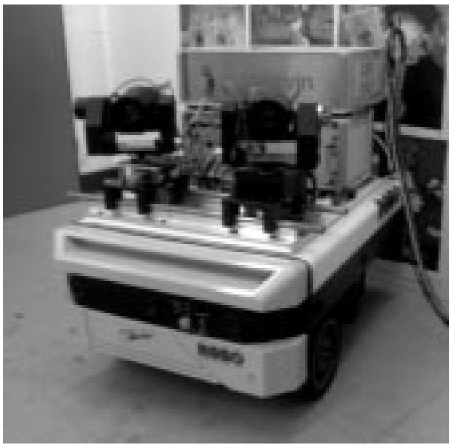

Fig. 1. The robot MARVIN consisting of a commercial mobile platform and an active binocular camera system. The platform and the camera system are integrated into a unified control loop.

3 Behavioral Organisation

In the following we describe some of the abilities of the system, showing the biological concern of the solutions, their performance and the architecture of problems coming up with the integration of "modules" to an entire "behavior".

Obstacle detection with inverse perspective mapping The term "inverse perspective mapping" (IPM) was introduced by Mallot et al. [MSS88, MBLB91] and does not correspond to an actual inversion of perspective, which is in general mathematically impossible. The inversion can be realized under the additional constraint that inversely mapped points should lie on the horizontal plane.

Consider a mobile robot with two translatory and one rotatory degree of freedom, which corresponds to a movement confined to a horizontal plane. In this paradigm, an *obstacle* can be defined as anything rising above the plane, i.e., objects that an observers path cannot cross. This is the minimal definition of an obstacle, requiring no additional information about the nature of the object.

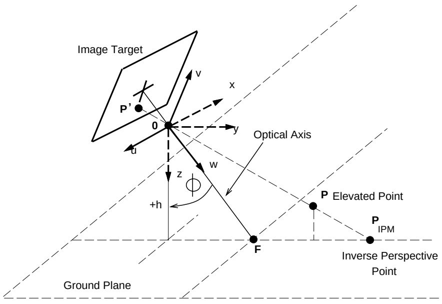

Fig. 2. Principle of the inverse perspective mapping paradigm. The coordinate system of the image plane is $W_I = \{u, v\}$. Its origin lies in the center of projection at a distance $-f$ (focal length) from the target. The world coordinate system $W_W = \{x, y, z\}$ also lies in the center of projection with a distance h (height component) from the motion plane. These assumptions hold for all the following calculations.

Fig. 3 shows the geometry of the perspective coordinate transform. In order to eliminate the perspective distortion due to the motion plane, all points $\mathbf{P}$ in space lying on the line through the center of projection and the intersection point $\mathbf{P}'$ on the image plane are projected into the motion plane. Elevated points are thus distorted and magnified.

In the case of pure inclination ϕ relative to the horizontal axis and no vergence, the *Inverse Perspective Mapping* ($\mathcal{IPM}$) is described by Eq. 2. A point P_I on the two-dimensional image plane is mapped into the point p_{IPM} on the two-dimensional inverse perspective world plane. The coordinates u, v denote the image coordinates, h is the height of the nodal point above the motion plane and f represents the focal length of the camera.

$$\begin{aligned} \mathcal{IPM} : \mathbb{R}^2 \mapsto \mathbb{R}^2 : \mathbf{p}_I = (u, v, -f, 1) & \qquad (1) \\ \mapsto \mathbf{p}'_{IPM} = \left(\frac{uh}{v\sin(\phi) - f\cos(\phi)}, \frac{(v\cos(\phi) + f\sin(\phi))\,h}{v\sin(\phi) - f\cos(\phi)}, h, 1, \right) & \end{aligned}$$

Stereoptical obstacle detection scheme with inverse perspective mapping In addition to the motion-based obstacle detection process, the inverse perspective mapping can be used as an efficient stage in a stereoptical obstacle avoidance procedure. The detection cue consists in this case of disparity changes relative to the ground plane which are caused by vertically extended objects. As before an obstacle is defined in a very simple way as any object having a height component from the ground floor. The topographic mapping transforms the ground plane into the zero–disparity plane. Obstacles can thus be extracted by simply subtracting the two inversely mapped images (for more details see [MSS88, BLM91]). The stereo approach has been extensively applied on an industrial mobile platform [MSS88] and is now being used in our group for the real-time lateral obstacle detection in normal street cars. Fig. 3 gives an example of this application.

3.1 Vergence and computation of disparity maps

2D phase method The main problem of stereoscopic vision is to find corresponding parts of the left image $l(\mathbf{x})$ and the right image $r(\mathbf{x})$, where $\mathbf{x}$ denotes the vector of image coordinates $(x, y)^\top$. The remaining section gives a rough sketch of the phase-based correspondence method. Details can be found in [TM93].

A *global* spatial shift $\mathbf{d}$ can be detected as a phase shift $\mathbf{k}^\top \mathbf{d}$ in the Fourier spectrum. Convolving the left and right image with complex Gabor filters yields *local* phase shifts between the images containing the information about varying spatial shifts [San88]. The Gabor profile at a certain position is called receptive field (RF). A set of complex Gabor functions $g_{ij}(\mathbf{x})$ with different mean frequencies $\mathbf{k}_{fij} = (k_{fi} \cos\vartheta_{fj}, k_{fi} \sin\vartheta_{fj})^\top$ differing in magnitude $k_{fi} = k_{f0}\,(\frac{1+t}{1-t})^i$ and orientation $\vartheta_{fj} = \pi\frac{j}{J}$ $(i = 0 \ldots I-1, j = 0 \ldots J-1)$ is defined via a quadratic form as

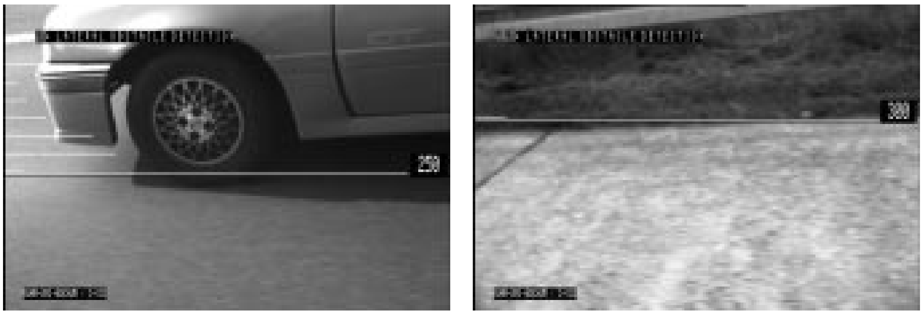

Fig. 3. Results of the stereoptic inverse perspective obstacle detection scheme. The experimental setup consists of two cameras fixed on the lateral car windows. The car is driving with an average speed of 80 to 100 *km/h*. The tilt angles have an amount of -65° and the pan angles are about $\pm 15^\circ$ with a baseline of $1.32\,m$. The angular parameters are known within a precision of $\pm 2^\circ$. Furthermore all camera parameters are disturbed due to the vibrations of the driving car. The figure gives an example of the robust character of the algorithm. By analyzing images of a spatial resolution of 512^2 pixels a cycle time of $200\,ms$ is reached. a) shows the side wheel of a car which is being passed by the experimental car. b) shows the border of the road where a slightly elevated strip of grass is detected. In both cases the white line defines the distance to the lowest detected part of the object. The depth estimation is computed relative to this border line.

$$g_{ij}(\mathbf{x}) = n_{ij}\ \exp\left(-\frac{1}{2}\mathbf{x}^\top \mathbf{A}_{ij}\mathbf{x}\right) e^{i\mathbf{k}_{fij}^\top \mathbf{x}} \ , \qquad (2)$$

where the matrix $\mathbf{A}_{ij}$ can be derived from a diagonal matrix $\mathbf{D}_i$ (Gabor filter at orientation 0^o) by multiplication with a rotation matrix $\mathbf{C}_j$:

$$\mathbf{A}_{ij} = \mathbf{C}_j \mathbf{D}_i \mathbf{C}_j^\top = \begin{pmatrix} \cos\vartheta_{fj} & -\sin\vartheta_{fj} \\ \sin\vartheta_{fj} & \cos\vartheta_{fj} \end{pmatrix} \begin{pmatrix} \frac{1}{\sigma_{xi}^2} & 0 \\ 0 & \frac{1}{\sigma_{yi}^2} \end{pmatrix} \begin{pmatrix} \cos\vartheta_{fj} & \sin\vartheta_{fj} \\ -\sin\vartheta_{fj} & \cos\vartheta_{fj} \end{pmatrix} \ . \qquad (3)$$

Convolution of the left and right image with each filter $g_{ij}(\mathbf{x})$ results in a set of complex filter responses $l_{ij}(\mathbf{x})$ and $r_{ij}(\mathbf{x})$ with phase angles $\varphi_{lij}(\mathbf{x})$ and $\varphi_{rij}(\mathbf{x})$.

Due to the aperture problem each phase difference $\Delta\varphi_{ij}(\mathbf{x}) = \varphi_{lij}(\mathbf{x}) - \varphi_{rij}(\mathbf{x}) + 2n\pi \in [-\pi, \pi)$ and its corresponding local frequency $\mathbf{k}_{sij}(\mathbf{x})$ determine a projection $\mathbf{p}_{ij}(\mathbf{x})$ of $\mathbf{d}$ onto the direction of $\mathbf{k}_{sij}(\mathbf{x})$:

$$\mathbf{p}_{ij}(\mathbf{x}) = \frac{\Delta\varphi_{ij}(\mathbf{x})}{||\mathbf{k}_{sij}(\mathbf{x})||} \frac{\mathbf{k}_{sij}(\mathbf{x})}{||\mathbf{k}_{sij}(\mathbf{x})||} \quad \text{with} \quad \mathbf{k}_{sij}(\mathbf{x}) = \frac{1}{2}\mathrm{grad}\left(\varphi_{lij}(\mathbf{x}) + \varphi_{rij}(\mathbf{x})\right) \ . \qquad (4)$$

The maximum disparity is bound by a phase difference of π and the minimum local frequency norm $||\mathbf{k}_{sij}(\mathbf{x})||$.

If a 3D patch projects onto the same points $\mathbf{x}$ on the left and right target the response amplitudes $|r_{ij}(\mathbf{x})|$ and $|l_{ij}(\mathbf{x})|$ are similar. With growing disparity between the left and right projections the amplitudes at $\mathbf{x}$ are assumed to differ more strongly. A confidence measure $c_{ij}(\mathbf{x})$ for correspondence is defined as

$$c_{ij}(\mathbf{x}) = 1 - \left| \frac{|l_{ij}(\mathbf{x})| - |r_{ij}(\mathbf{x})|}{|l_{ij}(\mathbf{x})| + |r_{ij}(\mathbf{x})|} \right| \in [0, 1] \ . \tag{5}$$

With increasing scale i the kernels and the disparity range detected by the Gabor filters become smaller. Therefore a coarse to fine search is employed [Mar82]: The disparity vector $\mathbf{d}_0(\mathbf{x}) = \Delta\mathbf{d}_0(\mathbf{x})$ is used as an initial estimate for the next filter scale $g_{1j}(\mathbf{x})$. The filter responses at scale i to be compared are separated between the left and right image by $\pm[\frac{1}{2}\mathbf{d}_{i-1}(\mathbf{x})]_r$ pixels so that they only have to yield an estimate $\Delta\mathbf{d}_i(\mathbf{x})$ for the remaining (smaller) disparity $\mathbf{d}(\mathbf{x}) - 2[\frac{1}{2}\mathbf{d}_{i-1}(\mathbf{x})]_r$, where $[\ ]_r$ denotes a rounding operation.

The disparity $\Delta\mathbf{d}_i(\mathbf{x})$ can be determined by at least two projections $\mathbf{p}_{ij}(\mathbf{x})$ which are not linearly dependent. Taking into account measurement errors of $\Delta\varphi_{ij}(\mathbf{x})$ and $\mathbf{k}_{sij}(\mathbf{x})$ the redundancy of more than two projections $\mathbf{p}_{ij}(\mathbf{x})$ can be used to minimize the weighted mean square error $e^2(\mathbf{x})$ for $\Delta\mathbf{d}_i(\mathbf{x})$:

$$e^2(\mathbf{x}) = \sum_{j=0}^{J-1} c_{ij}(\mathbf{x}) \left(\Delta\varphi_{ij}(\mathbf{x}) - \mathbf{k}_{sij}^{\top}(\mathbf{x}) \Delta\mathbf{d}_i(\mathbf{x}) \right)^2 \ . \tag{6}$$

A necessary condition for minimal $e^2(\mathbf{x})$ is

$$\frac{\partial e^2(\mathbf{x})}{\partial \Delta d_{ix}(\mathbf{x})} = 0 \ , \quad \frac{\partial e^2(\mathbf{x})}{\partial \Delta d_{iy}(\mathbf{x})} = 0 \tag{7}$$

leading to a set of linear equations for $\Delta\mathbf{d}_i(\mathbf{x}) = (\Delta d_{ix}(\mathbf{x}), \Delta d_{iy}(\mathbf{x}))^{\top}$. The disparity estimates $\mathbf{d}_i(\mathbf{x})$ are linearly combined with appropriate weights resulting in a disparity map $\mathbf{d}(\mathbf{x})$ and a confidence map $c(\mathbf{x})$:

$$\mathbf{d}(\mathbf{x}) = \frac{\sum_{i=0}^{I-1} \mathbf{d}_i(\mathbf{x}) \sum_{j=0}^{J-1} c_{ij}(\mathbf{x})}{\sum_{i=0}^{I-1} \sum_{j=0}^{J-1} c_{ij}(\mathbf{x})} \quad , \quad c(\mathbf{x}) = \frac{1}{IJ} \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} c_{ij}(\mathbf{x}) \in [0, 1] \ . \tag{8}$$

Application to vergence control Vergence control should have small response times to disparate input images. Furthermore, larger disparities must be discriminated than those found in a disparity map. Since only the global disparity $\mathbf{d}_{global}$ is needed, the algorithm operates on a lowpass filtered and subsampled version of the original stereo image. The filter kernels and the subsampled image are of equal size, resulting in a set of filter responses at only one position. The local frequency $\mathbf{k}_{sij}(\mathbf{x})$ is approximated by the filter mean frequency $\mathbf{k}_{fij}$. The maximum detectable disparity in the original image is $\frac{\pi}{||\mathbf{k}_{f00}||}$ multiplied with the subsampling factor. These global disparities are transformed into symmetric vergence movements via a pinhole model of the active stereo camera system in order to increase the overlap of the left and the right view. Measurements are iterated so that the vergence control behaves like a negative feedback system minimizing the global disparity. The cameras focus on the point where the optical axes have minimal distance [CE90]. This is demonstrated for a real scene, a plant at a distance of 3m (Fig. 4 and 5). The computation of disparity maps follows in section 3.2.

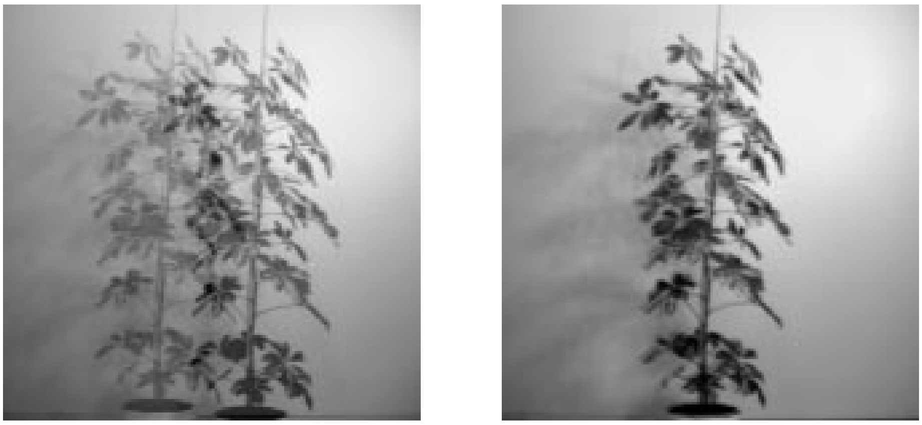

Fig. 4. Stereo image of plant scene *before* vergence (**left**) and *after* 9 iterations (**right**) printed as superimposed left and right image.

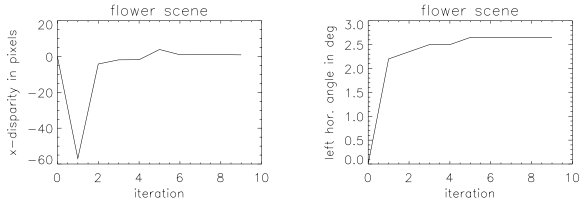

Fig. 5. Global x-disparity vs. iteration (**left**) and horizontal angle of the left camera (half of the vergence angle) vs. iteration (**right**).

3.2 Depth reconstruction

Localization uncertainty A point $\mathbf{X} = (X, Y, Z)^\top$ in the world is mapped onto target points $\mathbf{x_l} = (x_l, y_l)^\top$ and $\mathbf{x_r} = (x_r, y_r)^\top$ on the left and right target. This results in four linear equations, one for each target coordinate [VT94]. To recover a world point $\mathbf{X}$ one has to solve these equations for $\mathbf{X}$. Although, three out of the four equations are sufficient, we propose to recover $\mathbf{X}$ in terms of the least squared error $e^2(\mathbf{X})$.

Although the phase–based stereo algorithm calculates a disparity vector for each *RF*, it lacks the representation of the exact target coordinates x_l, y_l, x_r, y_r within the *RFs*. This localization uncertainty on the targets also corresponds to a *localization uncertainty* LU in the world.

The *3D LU* of the *RFs* under consideration is constrained by three factors:

- the *area of equal horizontal disparity* d_x,
- the *area of equal vertical disparity* d_y,
- the *volume of intersection VOI*, determined by that volume projecting to corresponding *RFs* in *both* targets. Thus $VOI = \{\, \mathbf{X} | \mathbf{x_l}(\mathbf{X}) \in RF_l \wedge \mathbf{x_r}(\mathbf{X}) \in RF_r \}$.

The intersection of the two isodisparity surfaces results in a curve representing the *curve of equal disparity* $\mathbf{d}$. Finally only that part of the curve intersecting the *RFs VOI* constitutes its associated LU.

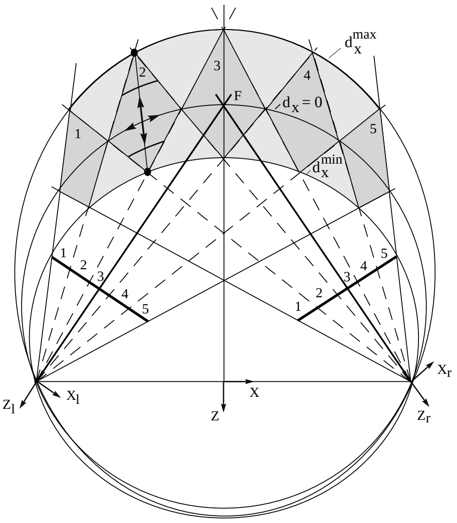

Fig. 6. *Localization Uncertainty LU*: The stereo cameras fixate **F**. Five non–overlapping *RFs* on each target are displayed exemplary. Dark shaded: Binocular field of view of the five exemplary *RFs*; Light+dark shaded: Area of detectable disparities d_x for continuously overlapping *RFs*; Furthermore the ellipses of equal disparity with $d_x = 0$ and the limiting cases $d_x = d_x^{max}$ and $d_x = d_x^{min}$ are shown.

Fig. 6 shows five exemplary *RFs* in addition with the corresponding world segments accessible to the stereo algorithm resulting from a cross section with $Y = 0$.

In contrast to global stereo algorithms, which analyze the whole 3D environment, the phase–based algorithm can in principle only analyze small parts of it, bounded by the disparities $[d_x^{min}, d_x^{max}]$ and $[d_y^{min}, d_y^{max}]$ (see Fig. 6). In general there is a set of valid disparity vectors $\mathbf{d}$ determined by those curves of equal disparity $\mathbf{d}$ which intersect the *RFs VOI*. If we assume all points X in the world to be equiprobable, as Blostein and Huang also do for the similar problem of pixel quantization [BH87], this assumption allows us to express the 3D–LU of any RF with disparity d_x in terms of the length of the corresponding line of equal disparity inside this RF. The LU is at its maximum for disparity $d_x = 0$ and decreases with negative or positive disparities (arrows in RF no.2). For the limiting disparities d_x^{max} and d_x^{min} the LU vanishes, denoted by the black dots.

Due to the error attached to the disparity estimates the reconstruction of a world point by the four world–target coordinate equations can lead to a recovered world point $\mathbf{X}$ which is outside the *RFs VOI*. Because of the horizontal baseline b even for a rather extreme optical setup the epipolar lines tend to be horizontal the closer they are to the target center. Thus, the possible d_y range within a receptive field tends to be small and an incorrect measure of d_y will easily lead to world points which are impossible within the *RFs VOI*.

If we base the localization solely on the knowledge of d_x, we can avoid the contradiction of recovering a point $\mathbf{X}$ which does not belong to the *RF's VOI*. This leads to a LU determined by the intersection of the area of equal disparity d_x with the corresponding *VOI*. Now we have to make a decision which point being subject to the LU is selected as the recovered $\mathbf{X}$. We assume all world points $\mathbf{X}$ to have equal probability. Furthermore from [Vog70], pp. 66 we know that $\min_{u \in \mathbb{R}} \mathbf{E}((x-u)^2) = \sigma^2$ for any distribution with $u = \mathbf{E}(x)$, the center of gravity. Thus, best we can do in terms of least squared error is to take the center of gravity of the distribution under consideration for u. For a more detailed description see [VT94].

Fusion of depth maps from different views In order to determine the depth profile for scenes with larger depth variations multiple views with fixation points shifted in depth must be fused [AA88].

Each vector $\mathbf{d}^{(i)}(\mathbf{x})$ in the disparity map of view i corresponds to a world point $\mathbf{X}^{(i)}(\mathbf{x})$. A cyclopean depth map $Z^{(i)}(\mathbf{x}_c)$ consists of the distances in Z-direction of each point $\mathbf{X}^{(i)}(\mathbf{x})$ registered at the intersection of the line $\overline{\mathbf{OX}}$ with a virtual (cyclopean) target in the middle of the camera base and tilt angle 0^o (see Fig. 7). Due to the discrete coordinate grid with step size x_m the intersection points $\mathbf{x}_c$ must be rounded to multiples of x_m.

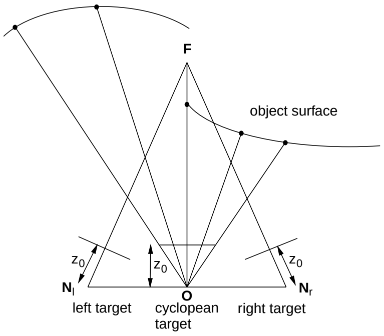

Fig. 7. Exemplary points of cyclopean depth map. Note the equidistant spacing of samples on the cyclopean target. The nodal point of this virtual target coincides with the origin **O** of the 3D coordinate system. The fixation point **F** is located in negative z-direction.

$$x_c = \left[-\frac{z_0 X}{x_m Z} \right]_r x_m \quad , \quad y_c = \left[-\frac{z_0 Y}{x_m Z} \right]_r x_m \tag{9}$$

Different depth maps $Z^{(i)}(\mathbf{x}_c)$ are integrated into $Z(\mathbf{x}_c)$ by pointwise selecting the depth value of the map with highest confidence $c^{(i)}(\mathbf{x})$. In general a depth map of a single view only contains contributions for a subset of grid points $\mathbf{x}_c$. In order to fill the gaps between contributions an interpolation of depth and confidence values is required.

Empirical observations suggest to smooth the underlying confidence maps in order to identify compact regions in the depth map receiving input only from one view without outlyers from another view [TM93].

The fusion of different stereo views acquired by our active camera system is examined. The left part of the stereo images shows a poster wall at a distance of 2.0 m and the right part a wall at a distance of 2.5 m from the camera system. In Fig. 8 the single cyclopean depth maps $Z^{(1)}(\mathbf{x}_c)$ and $Z^{(2)}(\mathbf{x}_c)$ with fixation points at a distance of 2.0 and 2.5 m in the median plane are shown:

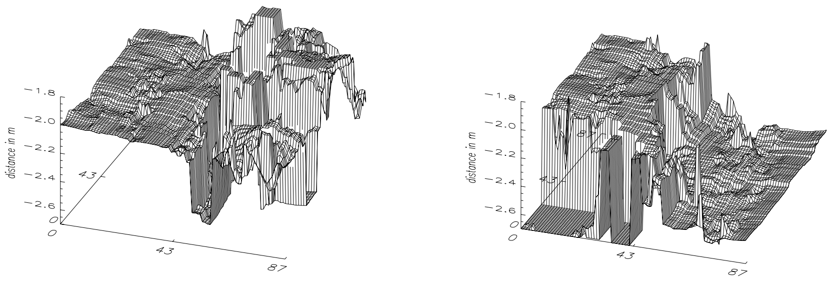

Fig. 8. Left: Cyclopean depth map of a stereo image with fixation point $\mathbf{F} = (0, 0, 2.0m)^\top$. **Right:** Cyclopean depth map of a stereo image with fixation point $\mathbf{F} = (0, 0, 2.5m)^\top$.

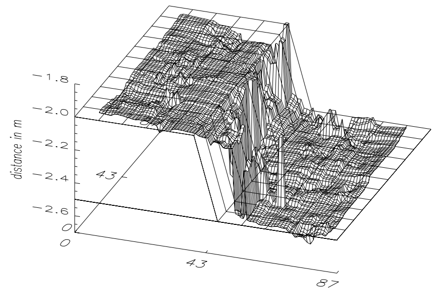

Fig. 9. Cyclopean depth map of the fused views. Most reconstruction errors are eliminated. The remaining errors are due to confidence values indicating that the wrong view (fixation point far apart from object surface) should be superior to the correct view (fixation point close to object surface).

It is obvious that the reconstruction from a single view fails if the 3D object surface leaves the bounded depth range. When applying the previously described procedure to both views, most of the reconstruction errors can be eliminated (Fig. 9).

3.3 Saccadic camera control and recognition

Attentional control is the most important mechanism in reducing the computational workload of visual processing. Furthermore it fits naturally into an *action for perception* approach if basic behaviors like eye movements are taken into consideration. Using the scene as an external memory buffer, complete image analysis and information storage become secondary to perceptual selectivity. Our approach is to define a model for saccadic camera movements as part of an integrated basic visual behavior architecture with visual exploration, scanning and recognition as commonly interesting abilities.

We propose a system for saccadic control, which is part of an architecture for basic visual behavior of our autonomous vehicle. The saccadic control is achieved by independent processing pathways for foveal and peripheral images. Scene recognition is obtained by classification of the foveal images and temporal integration of this classification with respect to their relative positions. Hypotheses about a scene are "tested" by the recognition system by trying to find expected but yet unfoveated parts of the scene. Expectations generate a tendency to gaze at a specific position as well as look for a specific feature, the latter by using selective masking of salient peripheral features. The system generates object specific emergent scanpathes but does not rely on them for recognition. Integration of saccadic control is done with an interest map, which implements competition and cooperation between different target demands.

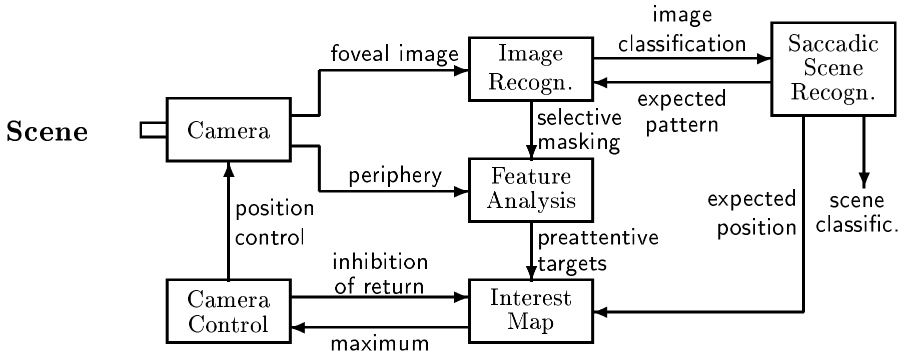

Fig. 10. Saccadic scene recognition, system scheme

Visual saliency In general visual saliency is an ill–defined term. A useful definition should include properties like *local distinctness* and *invariance* [HS93] but also temporal change and task–dependent clues. Our approach is to use a very simple saliency definition based on nonlinear filtering of the peripheral image to be able to find features in a reproducible way and extend this concept by selective masking in the case where expectation can serve as an additional saliency measure.

Therefore we compute saliency in absence of specific expectations by the product of the spatial derivatives of the image plus its temporal derivative

$$s_p(x,y,t) = \frac{\partial I(x,y,t)}{\partial x} \cdot \frac{\partial I(x,y,t)}{\partial y} + \mu \frac{\partial I(x,y,t)}{\partial t} \tag{10}$$

which basically defines a detector for corners, intensity peaks and movement. For feature–specific saliency we use an adaptive principal–component expansion of the image $I(x,y)$ into n vectors $\mathbf{v}_i$ [San89] together with the (small) vector $\mathbf{l}$ of expected coefficients to define an additional selective saliency term as

$$s_e(x,y,t) = \sum_{i=0}^{n} \| l_i(t) - (I(x,y,t) * \mathbf{v}_i(x',y')) \| \tag{11}$$

Pattern recognition Concerning the system behavior the pattern recognition method used is only relevant with respect to its performance. We use a linear classifier based on higher–order–autocorrelation features (for details see [KVJ94]), which supplies a *similarity vector* $\mathbf{c}$ between the actual foveal image and all stored foveal patterns. Its maximum value $c_{i'}(t) = \max_i(c_i(t))$ determines the actual recognized pattern i' at time t.

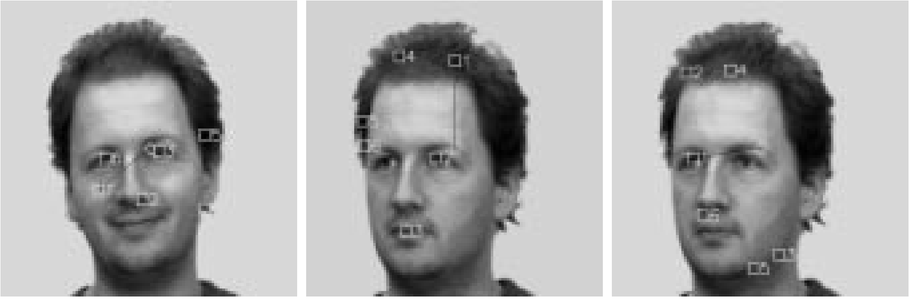

Fig. 11. a) Image 1 with sensor–driven saliency priority, b) Image 2 with sensor–driven saliency priority c) Image 2 with saliency selective to the left–eye feature of image 1

Generation of object hypotheses Scene recognition is performed as transsaccadic integration of the pattern recognition information. The evidence of the known u foveal patterns for the v objects is stored in the pattern–object relation matrix $\mathbf{Q} \in \mathbb{R}^{u \times v}$. $\mathbf{Q}$ constitutes the associative "what"–memory[UM82]. While the position of the foveal patterns of an object is stored in the"where"–memory $\mathbf{R} \in \mathbb{C}^{u \times v}$. The matrix $\mathbf{R}$ is complex to accommodate horizontal and vertical angles as the real and imaginary parts of one matrix element. $\mathbf{Q}$ and $\mathbf{R}$ are sparse for large numbers of foveal patterns and objects.

The cognitive object recognition process temporally integrates this information in a vector $\mathbf{a}(t)$:

$$\frac{d\mathbf{a}(t)}{dt} = - \underbrace{\tau_a \mathbf{a}(t)}_{\text{relaxation}} + \underbrace{\mathbf{Q}\mathbf{c}(t)}_{\text{input}} - \underbrace{\mathbf{e}(t)}_{\text{position error}} \tag{12}$$

An internal object hypothesis for object j' is stated if the condition $a_{j'} = \max_j(a_j) \wedge a_{j'} > \theta$ is valid, where θ is a threshold to suppress the statement of 'weak' hypotheses.

The *input term* is a vector containing the evidence of the current foveal image for the v learned objects. All position vectors are complex expressing horizontal angles as real parts and vertical angles as imaginary parts. The *position error term* reduces the value of the actual object hypothesis. It depends quadratically on the spatial difference between the most recently executed saccade (position $r'(t_{k-1})$ to $r'(t_k)$) and the last saccadic shift according to the "where'–memory (position $r_{i'(t_{k-1})j'(t_k)}$ to $r_{i'(t_k)j'(t_k)}$):

The *relaxation term* enables the system to "forget" acquired information.

This *emergent saccadic model* uses a very efficient and robust representation avoiding memory consuming 'fully connected graph' representations or explicit scan path storage[RB90].

The positional error can be neglected completely if the system is not stating an hypothesis and is small if there is no significant spatial disturbance between the patterns. In this case Eq. (12) becomes linear and the stability of a hypothesis is given if $(\mathbf{Q}\mathbf{c}(t))_k \geq \tau_a \theta$ where k denotes the index of the considered object.

For $(\mathbf{Qc}(t))_k > \tau_a \theta$ the object k will become a hypothesis if there is no other stronger evidence.

Learning of relevant patterns and objects is easily achieved by using the explorative saccadic scanning behavior.

Generation of cognitive targets A *pattern accumulator* $\mathbf{b}$, is used to generate the cognitive target demands. The relative size of the values in $\mathbf{b}$ denotes the urgency to foveate a pattern to verify the current hypothesis, while trying to avoid patterns that have lately been gazed at:

$$\frac{d\mathbf{b}(t)}{dt} = - \underbrace{\tau_b \mathbf{b}(t)}_{\text{relaxation}} - \underbrace{\mathbf{c}(t)}_{\text{recognition}} + \underbrace{\mathbf{Q}^T \mathbf{a}'(t)}_{\text{verification}} \tag{13}$$

$$\text{where} \qquad a'_j(t) = \begin{cases} a_j(t) & \text{for } j = j' \\ 0 & \text{for } j \neq j'. \end{cases} \tag{14}$$

A *recognition term* reduces all values b_i by the certainty they have already been assigned. The *verification term* is calculated by the back-projection of the current object hypothesis j' according to the matrix $\mathbf{Q}$, and contains the evidence of the foveal views belonging to the object hypothesis. Values of $\mathbf{b}$ are "forgotten" by temporal *relaxation.* By using the "where"–memory, the system generates a weighted list of discrete top–down target positions, which is transformed into a smooth excitation distribution s_o for the interest map. The most urgent target also defines the selective saliency computation.

Interest map and camera control While the saliency and recognition target demands $\mathbf{s_p}, \mathbf{s_e}, \mathbf{s_o}$ are projected into the *excitatory subsection* I_x of an *interest map*, the *camera control* enters its actual position via h_c into another subsection I_r for *inhibition of return* to positions already gazed at.

$$\frac{dI_x(\mathbf{x},t)}{dt} = -\tau_x I_x(\mathbf{x},t) + D_x \nabla^2 I_x(\mathbf{x},t) + s_p(\mathbf{x},t) + s_e(\mathbf{x},t) + s_o(\mathbf{x},t) \tag{15}$$

$$\frac{dI_r(\mathbf{x},t)}{dt} = -\tau_r I_r(\mathbf{x},t) + D_r \nabla^2 I_r(\mathbf{x},t) + h_c(\mathbf{x},t) \tag{16}$$

The relaxation terms τI again allow for "forgetting". The spatial diffusion $D\nabla^2 I$ locally distributes activity, so the camera control system can cope with the integration of positional errors.

The camera control calculates the spatial position $\mathbf{x}^*$ of the next target simply as the maximum position of the sum of the two subsections $I_x(\mathbf{x}, t_k) + I_r(\mathbf{x}, t_k)$.

3.4 Pyramidal optical flow computation and object tracking

One important task for mobile robot system is motion detection and object tracking. Motion detection can be used for segmentation tasks due to moving objects in natural environment or to stabilize a moving region of interest in the center of the camera target. This fixation task can give additional information about three-dimensional structure of the environment.

Algorithms for computation of optical flow either gradient-based or correlation–based are often time–consuming or yield only sparse vector fields. For our approach we use a greyvalue based correlation algorithm which has shown to calculate a optical flow fields with high confidence. This algorithm will not be modified but by implementation into scale-space structure it will lead to the sufficient density of vector fields which is necessary for segmentation task.

Correlation–based optical flow computation The original algorithm can be described by the following three steps [BLP89]:

I. Shift and Compare: The expected motion displacements are characterized by an 2D interval $D_\delta := [-\delta, \delta] \times [-\delta, \delta]$. For each node $\mathbf{x}$ and permissible displacement $\mathbf{d} \in D_\delta$, a comparison function $\phi(a, b)$ is evaluated. (Here, a, b denote either greylevels or intensities of preprocessed images.) The output of this step is a matching strength for each node and displacement, $m(\mathbf{x}, \mathbf{d}) = \phi(E_t(\mathbf{x}), E_{t+\Delta t}(\mathbf{x} + \mathbf{d}))$.

II. Local Summation: At each pixel $\mathbf{x}$ the matching strength for corresponding displacements from the pixels in a neighborhood $P_\nu(\mathbf{x})$ are accumulated. The output of this step is a combined matching strength which, again, is a point wise function:

$$M_{\mathbf{x}}(\mathbf{d}) := \sum_{\mathbf{y} \in P_\nu(\mathbf{x})} m(\mathbf{y}, \mathbf{d}). \tag{17}$$

III. Winner-Take-All: This step forces a consistent match from frame to frame. To each pixel $\mathbf{x}$, the displacement that received the highest matching strength M is assigned as its velocity value $V(\mathbf{x})$ by a winner-take-all scheme. That is, V is selected to satisfy the condition

$$M_{\mathbf{x}}(V(\mathbf{x})\Delta t) = \max_{\mathbf{d} \in D_\delta} M_{\mathbf{x}}(\mathbf{d}). \tag{18}$$

A large vote for one particular displacement is expected if the motion field is locally constant.

Pyramidial optical flow computation An aquired image sequence can be transformed into a Gaussian pyramid [Bur84] by convolution and subsampling. This process can easily be done by hardware or a signal processor system. The Gaussian pyramid representation splits up the image into a set of images with different resolution. By restricting the permissible displacement to a small region around each node this representation is used to calculate vector fields for different velocities at each level of the pyramid [WG93].

A way to define these velocity channels is to calculate the optical flow at each pyramidal level but to limit the permissible displacement window $D_\delta := [-\delta, \delta] \times [-\delta, \delta]$ to a maximum of $[-1, 1][-1, 1]$.

Starting with at pyramid level G_0 the Gaussian pyramid will be generated up to level G_{max} and the optical flow then be computed at each level G_i with the limited displacement window (see fig. 3.4). This formally means that the available velocity vectors $\mathbf{v}$ are described by

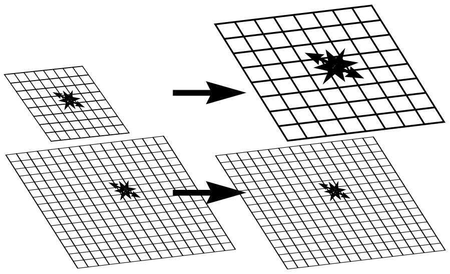

Fig. 12. The pyramidal flow scheme. The left hand side shows the permitted displacement range at two pyramid levels, the right hand side the detected range after interpolation of each level up to the highest resolution level G_0.

$$\mathbf{v} = (\pm x, \pm y) \quad \text{with} \quad x \in \{0, 2^i\} \quad \text{and} \quad y \in \{0, 2^j\} \quad \forall \quad 0 \le i < \delta_v, \quad 0 \le j < \delta_h \tag{19}$$

$$\text{or} \quad \mathbf{v} = (\pm x, \pm y) \quad \text{with} \quad x, y \in \{0, 2^i\}, \forall \quad 0 \le i < \delta_h = \delta_v \tag{20}$$

This range of displacement vectors is defined for each pixel. In this way the optical flow scheme combined with the Gaussian pyramid defines a band-pass relating to detectable velocities. The two-dimensional spectrum of achievable velocities is logarithmic.

Comparison of the computational costs The computational cost for the original algorithm defined in [BBM90]

$$L_{total} = (2dh_{max} + 1)(2dv_{max} + 1)(4XY - X) + C \tag{21}$$

can be extended for the pyramidal approach to

$$L_{quad} = 9 \sum_{d=1}^{dh_{max}=dv_{max}} \left(4\frac{XY}{4^{d-1}} - \frac{X}{2^{d-1}}\right) + C \tag{22}$$

where X, Y denote the picture size and dh_{max}, dv_{max} denote the displacement window. C describes the constant effort for the computational environment. Eq. (22) shows a significant decrease of the computational costs with an increasing number of pyramidal levels.

Tracking application The reduction in computational cost for pyramidal flow computation results in applicability for real-time tracking of moving objects. Figure 13 shows a sequence where a moving person is tracked by an active camera system. Segmentation is done by interpolation of all significant vectors of each pyramid level to the basis G_0. A succeeding integration step yields homogenous dense regions with a pyramidal weighted mean flow vector. This vector is used as input to the camera controller system.

3.5 Approximative vision and low dimensional neural dynamics for ratlike robot navigation

We derive from the ethology of rat spatial behavior a navigation strategy for an autonomous mobile robot. Exploration of a novel environment is organized

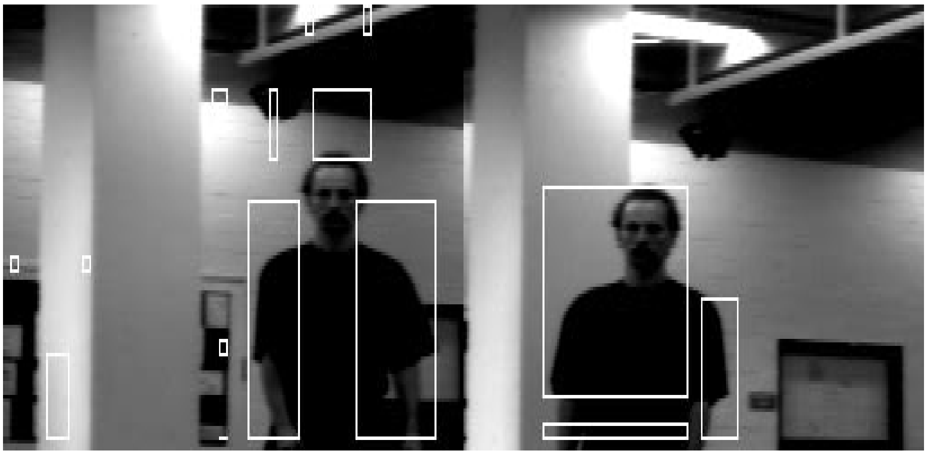

Fig. 13. Tracking of a moving person. Regions with homogeneous dense vector field are marked with white boxes.

around base points. The robot creates a base at a certain location by taking local views looking into several directions and computing distance estimates to the objects seen in these views. Additionally the robot memorizes the movements it performed to get from one base to the next. Thus space is represented in a graphlike structure.

The bases act as centers of force in a dynamics whose state variables estimate the robots position within its environment. Source of position information are the optical flow fields between an actual image and the memorized local views. These are computed by the correlation algorithm mentioned in 3.4. Through Taylor approximation we derive formulas such that each measured correspondence vector votes for a certain value of the position variables. At that point in state space an attracting potential is augmented proportionally to the correlational score with which the correspondence vector was measured.

A second source of position information are the movement commands the robot executes, giving rise to another attracting potential $s_{mov}(x,t)$ shown in Fig. 3.5 as the dotted entries. Note, that it generates the dominating force in regions where no reliable visual information can be obtained.

The sum of these “stimuli” is fed into a dynamics derived from neurophysiological considerations. It describes the dynamics of the center of a localized excitation of a neural tissue with lateral interactions.

$$\frac{d}{dt}x = \frac{1}{\tau}\int_{-\infty}^{\infty}(x'-x)e^{\frac{-(x'-x)^2}{k^2}}s(x',t)dx' + noise$$

Its useful features are that convergent information cooperates, divergent information competes in shaping the stable attractor states. This endows the system with the capability of decision making through bifurcation and stabilization of decision through hysteresis.

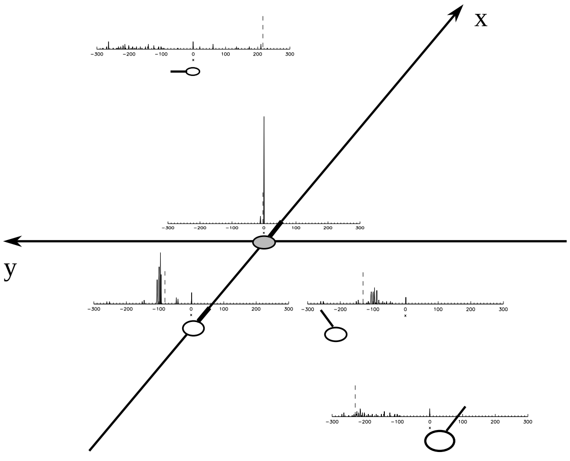

Fig. 14. The figure shows resulting potentials $s_{vis}(x,t)$ for the state variable estimating the x-position. Shown are potentials for different robot attitudes, indicated by circles with hair. The potentials result from the comparison of actual views with a memorized one which was taken at the position indicated through the filled symbol. Note, that obviously the potentials are the more pronounced the less actual position and memory position differ.

In an analogous fashion estimates of obstacle distances stemming from binocular stereo, motion stereo or from memory entries in the spatial representation can be fed into a neural dynamics which manages the planning variables “forward velocity” and “angular velocity”. So all variables within the robot architecture necessary for spatial behavior and changing on a fast time scale are treated with the same tool.

We found that spatial behaviors such as homing are already possible with a resolution as poor as 32x32 pixels for position and 64x64 pixels for distance sensing. So even on conventional serial hardware cycle times of the order of magnitude of one second can be reached.

3.6 Dynamic neural fields for autonomous robots

The design and operation of autonomous systems pose challenges, which beg new ideas and concepts: (a) If behaviors are to be produced with short reaction times and controlled in real time enormous efficiency of information processing and closed loop control is implied; (b) If a reasonably rich behavioral repertoire is aimed at, multiple behaviors must be available and must be coordinated, including the integration of sensory information; (c) To warrant flexibility, change of behavior in response to information about the environment must be possible; (d) If systems are to accumulate knowledge about their environment, memory in the sense of past sensory or behavioral experience affecting current behavior must be available. Recently, the fact that biological systems demonstrate elaborated solutions to those requirements has begun to influence the search for artificial solutions:

First, behavior-based robotics (review [Bro91]) which has been influenced by ethological findings has made the break away from artificial intelligence approaches to these problems. This meant renouncing symbolic representations and reasoning modules, and hence, avoiding the difficult problems of generating symbols from sensory information, of developing flexible and temporally structured plans in terms of rules, and of transforming symbolic plans into controllable actions in the world. The new methods have strengths in terms of efficiency and closed loop stability (point (a)), achieve some degree of integration (point (b)), and provide limited flexibility (point (c)). Strengths of the AI approach are sacrificed, in particular, as concerns memory and invariant representation of the environment (point (d)), but also, in terms of the generality of the theoretical language in which systems are defined. This affects the capability to scale up behavior-based systems to increased behavioral complexity. As such scaling is attempted, the lack of a firm theoretical foundation is acutely felt and the need for a general and unifying language recognized. At the same time, remnants of a symbolic approach linger in many of the behavior-based architectures. For instance, the individual contributions to potential fields used to control reactive navigation (see, e.g., [Ark90]) are essentially symbols: each perceptual or behavioral schema is instantiated and has identity. Similarly, the state variables used in finite state machines (see, e.g., [Bro91]) are themselves symbols, they maintain identity in time. This subtle reliance on symbols through instantiated variables might contribute to the difficulty to seamlessly integrate behavioral modules (e.g., the problem discussed by Payton, [PRK90]).

Second, neurophysiological evidence has been found that motor behavior is represented within a population code [GSK86].

Based on this findings we propose an architecture that provides a uniform language, which can, in principle, be used throughout the autonomous system. Generation of overt behavior is dealt with in the same manner as the generation of sensory information as well as the creation and updating of memorized information. The language is based on dynamic neural fields [Ama77] which are treated in the limit case of strong intra-field interaction. In this limit, which runs counter to typical neural network architectures, behaviors are represented by stationary and stable localized distributions of excitation over parameter fields, each of which defines a system level. Coupling among levels is local and, on average, weak, so that levels can be viewed as behavioral modules that interact while maintaining their individual functional characteristics.

Within the dynamic field architecture, both sensory integration and integration of elementary behaviors take place. There is no distinction between decision making and control: the field dynamics implements closed loop stability, and instabilities lead to decision making. Representations take the form of abstract behaviors, that is, they arise in fields in which sensory information is stored in a behavior-related way, that is, in terms of the behaviors specified by the sensory information.

To illustrate the approach in an exemplary model system we refer to the classic problem of target acquisition while avoiding obstacles for mobile platforms in the form given in [SD92]. Dynamic fields are introduced (a) for the representation of sensory data specifying target and obstacle areas, (b) for the creation, updating, and deletion of such information in memory, and (c) for the generation and control of vehicle trajectories.

Each field is defined in terms of parameters, in which the behavioral con-

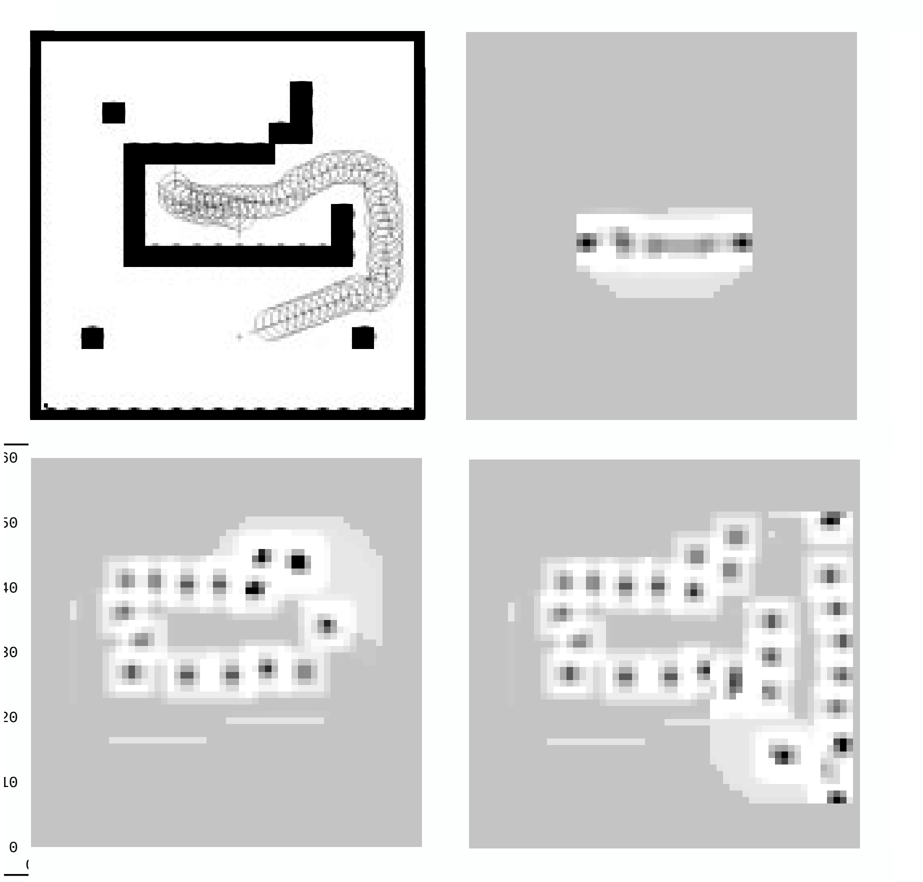

Fig. 15. (top left) The path of the robot in a complex environment. Note that the vehicle is moving in the opposite direction than the goal while it is leaving the central box (black regions denote obstacles, the cross indicates the target, the vehicle is drawn as circle with hair showing heading direction). Activity in the memory field as a result of the intra-dynamics at the initial (top right), at an intermediate (bottom left) and at the final position (bottom right) of the robot (black denotes peaks of positive activity). The spacing of the memory peaks is chosen such that no local minima occur in the external input to the planning field and depends on the vehicle size. The resulting behavior is based on local information from sensory fields and memory at the position of the robot only.

straints specific to the field can be expressed as points or simple sets in the field (see Fig. 15). These parameters are not dictated by the properties of sensor or effector systems, although ultimately transformation to their interfaces must be possible. Each field is governed by an Amari neural field equation [Ama77], in the limit of strong intra-field interaction. Behaviors consist of individual peaks of localized excitation (case of instantiation), representations consist of groups of localized peaks. In both cases, the shape and spacing of the peaks reflect the behavioral requirements at the particular level. Input to the fields takes the form of additive local excitation, which thus specifies a behavior at the given level. Any given field couples into other fields in an analogous manner. To achieve local coupling, transformations of the parameters spanning the fields may be required. Attractor fusion and instabilities are used to determine the shape of interactions within and across fields so that task requirements are met.

The build-up and maintenance of memory over time is demonstrated and its formal integration as an additional level of behavior is shown in Fig. 15.

In [ESed, SE] we present simulations to demonstrate the feasibility and properties of the approach.

4 Conclusion and Outlook

Those behavioral tasks interfering directly with the mobile robot have already been implemented on the MARVIN robot system. The visual obstacle detection process runs on the mobile robot and it is integrated with the path planning system. Saccadic exploration and recognition, vergence, depth estimation and tracking are integrated into a "camera–frontend" which controls these cemera movements autonomously and outputs various information (distance, depth, velocity, segmentation and identification). Each module for itself has been extensively tested on real data and works in a closed loop on the robot.

The main problem now is the integration of these partially cooperating modules into a common structure. The solution of this so–called "architecture problem" is the main topic of our future research activities. The aim is to design a flexible structure which allows a scalable complexity within the behavioral task hierarchy and does not require a redesign due to changing environmental conditions and task redefinitions.

The general structure of the integrated system we envision is highly modular. We do not want to introduce central representations for sensory data, memory or behavioral control. Instead we try to rely completely on the various distributed modules and representations, combined in serial, parallel or with feedback. Data is typically exchanged in the format of estimated parameters or parametric maps, eventually including confidence values. Control is handled by the agents themselves by using concurrence of resource allocation. Two examples where we have already implemented this interaction scheme are the interest map approach to saccadic sensor control (see section 3.3) and the camera–frontend integration. We are presently also evaluating flexible interaction networks, which are able to develop an optimized connection scheme between the modules. We believe that such network flexibility is a crucial condition for scalable architectures.

References

[AA88] A. Abbott and N. Ahuja. Surface Reconstruction by Dynamic Integration of Focus, Camera Vergence, and Stereo. In *Proceedings of ICCV 1988*, pages 532 – 543. IEEE Computer Society, 1988.

[Ama77] S Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977.

[Ark90] R C Arkin. Integrating behavioral, perceptual, and world knowledge in reactive navigation. *Robotics and autonomous control*, 6:105–122, 1990.

[BBM90] S. Bohrer, H.H. Bülthoff, and H.A. Mallot. Motion Detection by Correlation and Voting. In *ICNC-90*, pages 471–474, Düsseldorf, 1990. DGK, Elsevier, North Holland.

[BH87] S. Blostein and T. Huang. Quantization Errors in Stereo Triangulation. In *Proceedings of ICCV 1987*, pages 325 – 334. IEEE, 1987.

[BLM91] S. Bohrer, A. Lütgendorf, and M. Mempel. Using Inverse Perspective Mapping as a Basis for two Concurrent Obstacle Avoidance Schemes. In *ICANN–91*, Helsinki, 1991. Elsevier, North Holland.

[BLP89] H. H. Bülthoff, J. J. Little, and T. Poggio. A Parallel Algorithm for Real–Time Computation of Optical Flow. *Nature*, 337(9):549–553, February 1989.

[Bro91] R A Brooks. New approches to robotics. *Science*, 253:1227–1232, 1991.

[Bur84] P. J. Burt. The pyramid as a structure for efficient computation. In A. Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, Information Sciences, pages 6–35. Springer, 1984.

[CE90] H. Collewijn and C. J. Erkelens. Binocular eye movements and the perception of depth. In E. Kowler, editor, *Eye Movements and Their Role in Visual and Cognitive Processes*, pages 213 – 261. Elsevier Science Publishers B.V., Amsterdam, 1990.

[DFSKT94] M. Dose, S. Fuhrmann, E. Schulze-Krüger, and W. Theimer. An Autonomous Mobile Robot: A Development Tool for Operation in a Natural Environment. Internal Report IR-INI 94-02, Institut für Neuroinformatik, Ruhr-Universität Bochum, April 1994.

[ESed] C. Engels and G. Schöner. Dynamic fields endow behavior–based robots with representations. *Robotics and Autonomous Systems*, submitted.

[GSK86] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner. Neural population coding of movement direction. *Science*, 233:1416–1419, 1986.

[HS93] Robert$\tilde{M}$. Haralick and Linda G. Shapiro. *Computer and Robot Vision, Vol. 2.* Addison–Wesley, 1993.

[KVJ94] M. Kreutz, B. Völpel, and H. Janßen. Scale–invariant image recognition based on higher–order autocorrelation features. Internal Report 94-07, 1994.

[Mar82] D. Marr. *Vision.* W. H. Freeman, San Francisco, 1982.

[MBLB91] H. A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer. Inverse Perspective Mapping Simplifies Optical Flow Computation and Obstacle Detection. *Biological Cybernetics*, 64:177–185, 1991. Springer-Verlag.

[MSS88] H. A. Mallot, E. Schulze, and K. Storjohann. Neural Network Strategies for Robot Navigation. In *Proc. of nEuro'88*, pages 560–569, Paris, June 1988.

[PRK90] D. W. Payton, J. K. Rosenblatt, and D. M. Keirsey. Plan guided reaction. *IEEE Transactions on Systems, Man and Cybernetics*, 20(6):1370–1382, 1990.

[RB90] R. D. Rimey and C. M. Brown. Selective attention as sequential behavior: Modeling eye movements with an augmented hidden markov model. Technical Report TR327, Computer Science Department, University of Rochester, 1990.

[San88] T. Sanger. Stereo Disparity Computation Using Gabor Filters. *Biological Cybernetics*, 59:405 – 418, 1988.

[San89] T.D. Sanger. Optimal unsupervised learning in a sungle layer linear feedforward neural network. *Neural Networks*, 2:459–473, 1989.

[SBDF90] E. R. Schulze, S. Bohrer, M. Dose, and S. Fuhrmann. An Active Vision System for Task-Specific Information Processing. In *Mustererkennung 1990, 12. DAGM-Symposium*, Oberkochen-Aalen, September 1990. Springer-Verlag.

[SD92] G Schöner and M Dose. A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion. *Robotics and Autonomous Systems*, 10:253–267, 1992.

[SE] G. Schöner and C. Engels.

[TM93] W. Theimer and H. Mallot. Vergence Guided Depth Reconstruction Using a Phase Method. In Karl Goser and Jeanny Hérault, editors, *Neural Networks and their Industrial and Cognitive Applications: Neuro-Nîmes 93*, pages 299 – 308. EC2 Publishing, 1993.

[UM82] L. G. Ungerleider and M. Mishkin. Two cortical visual systems. In D. J. Ingle, M. A. Goodale, and R. J. W. Mansfield, editors, *The Analysis of Visual Behavior*, pages 549 – 586. The MIT Press, Cambridge, Ma., 1982.

[Vog70] Walter Vogel. *Wahrscheinlichkeitstheorie*, volume 22 of *Studia Mathematica*. Vandenhoeck & Rupprecht, 1970.

[vSBKT94] W. von Seelen, S. Bohrer, J. Kopecz, and W. Theimer. A Neural Architecture for Visual Information Processing. *International Journal of Computer Vision*, 1994. in press.

[VT94] B. Völpel and W. Theimer. Localization Uncertainty in Area-Based Stereo Algorithms. Internal Report IR-INI 94-06, Institut für Neuroinformatik, Ruhr-Universität Bochum, June 1994.

[WG93] V. Vetter W. Gillner, S. Bohrer. Objektverfolgung mit pyramidenbasierten optischen flußfeldern. In *3. Symposium, Bildverarbeitung '93*, pages 189–220, Esslingen, November 1993. Technische Akademie Esslingen.

This article was processed using the LaTeX macro package with LLNCS style