Combined space-variant maps for optical flow based navigation

Gregory Baratoff, Christian Toepfer, Heiko Neumann
Dept. of Neural Information Processing, Faculty of Computer Science,
University of Ulm, 89069 Ulm, Germany

{baratoff, ct, hneumann}@neuro.informatik.uni-ulm.de

Abstract : A robot navigating in an unstructured environment needs to avoid obstacles in its way and determine free spaces through which it can safely pass. We present here a set of optical flow based behaviors which allow a robot moving on a ground plane to perform these tasks. The behaviors operate on a purposive representation of the environment called the "virtual corridor" which is computed as follows : The images captured by a forward-facing camera rigidly attached to the robot are first remapped using a space-variant transformation. Then, optical flow is computed from the remapped image stream. Finally, the virtual corridor is extracted from the optical flow by applying simple but robust statistics. The introduction of a space-variant image preprocessing stage is inspired by biological sensory processing, where the projection and remapping of a sensory input field onto higher-level cortical areas represents a central processing mechanism. Such transformations lead to a significant data reduction, making real-time execution possible. Additionally, they serve to "re-present" the sensory data in terms of ecologically relevant features, thereby simplifying the interpretation by subsequent processing stages. In accordance with these biological principles we have designed a space-variant image transformation, called the polar sector map (PSM), which is ideally suited to the navigational task. We have validated our design with simulations in synthetic environments and in experiments with real robots.

1 Introduction and Motivation

An autonomous system - be it biological or technical - navigating through an unknown or only partially known environment needs to be able to detect obstacles in its heading direction and turn to avoid collisions with them. Furthermore, it needs to be able to gauge the width and height of free spaces to determine whether it can safely pass through. Neither of these tasks requires a detailed internal reconstruction of the three-dimensional structure of the environment [2]. In section 2 we present a description of a set of behaviors which we have implemented in simulations and in real robots, and derive a purposive representation [1] of the environment in terms of a "virtual corridor", based on which the robot reliably performs these navigational tasks. Our virtual corridor is an extension of Coombs *et al.*'s [4, 3] "conceptual corridor" and of other "minimalist" representations designed for the basic navigational tasks considered here [15, 19].

The inspiration for such representations comes from studies by Srinivasan [17] on the flight behavior of bees, which showed that bees perform seemingly complex flight maneuvers reliably based on very coarse information extracted from the optical flow field. These bee behaviors can be successfully used in ground-based robotic vehicles [4, 3, 15, 21]. In section 3 we present a short summary of the properties of the optical flow field during typical ground-plane motion, and show how the information necessary for performing the navigational tasks can be derived from it.

Since the robot moves, it is required to react in time to obstacles in its heading direction, e.g. by triggering an avoidance maneuver. For this to be possible, sensory processing must be fast enough to guarantee real-time reaction. At the present time it is, however, not possible to process optical flow at high enough speed at full camera resolution with standard PC hardware. Some kind of data reduction is therefore necessary. Simple uniform subsampling of the input image is not a viable option, because it reduces the resolution in the central region where high resolution is crucial for detection of imminent collisions. A space-variant transformation of the input images is called for. Here, biological principles provide further inspiration. The projection and remapping of a sensory input field onto higher cortical areas is a central feature of biological sensory processing, especially in higher animals. An important aspect of many of

mappings relevant to our problem are the models of the retino-cortical projection in primates [16, 13]. Besides introducing a significant data reduction, which makes real-time execution possible, these mappings simultaneously allow a coarsely resolved wide field of view and a region of high visual resolution for more detailed processing. The former is necessary for monitoring the environment, the latter for object discrimination and recognition. Additionally, of particular interest in the navigational context is the suggestion by Wright & Johnston[24] that *"... the morphology of cortical maps may have a functional significance ... related to the visual control of locomotion ..."*. In section 4 we discuss Schwartz's [16] model of the primate retino-cortical projection in terms of the complex logarithmic mapping (CLM), and analyze it with respect to its suitability for our navigational tasks. This model, however, does not satisfy all of the requirements. Therefore, we have designed a new space-variant transformation, the *polar sector map (PSM)*, which is optimized with regard to the subsequent interpretation of the optical flow by the navigation and collision avoidance behaviors. It satisfies the following design goals :

1. In the central image region optical flow should directly signal "time-to-collision"[9], i.e. the time until collision with an obstacle would occur if the robot were to continue at the current speed in the same direction.

2. In the image periphery optical flow should be a monotonic function of (a) the side-ways distance in left and right peripheral image regions and of (b) the height above ground in the lower and upper peripheral image regions.

The first property allows the robot to avoid collisions with obstacles in the heading direction. The second property leads to an *ordinal representation*[1] of the periphery that is *invariant over time under forward translational motion*. Due to these properties, the PSM provides a representation from which the control signals driving the navigation behaviors can be efficiently extracted. Indeed, as we show in section 5, simple robust statistics of the optical flow field in the central and peripheral image regions yield stable control signals for the collision avoidance and navigation behaviors, respectively. Since the choice of these statistics reflects the structure of the environment as well as the task to be performed, we have termed them "purposive statistics". We also show how the virtual corridor is built from these purposive statistics, give the control laws for the behaviors, and demonstrate the performance of the system by experiments in a synthetic simulation environment and with real robots. In section 6, we discuss our approach and summarize our contributions.

2 Navigational Behaviors and Purposive Representation

2.1 The behavioral repertoire

In order to achieve collision-free navigation in an unknown environment, we propose the following repertoire of behaviors for a ground-based mobile robot :

- The **collision-avoidance (CA)** behavior detects obstacles in front of the robot and turns to avoid them.
- The **centering (CE)** behavior steers the robot through the middle of a passage.
- The **adaptive speed (AS)** behavior adjusts the robot's speed in proportion to the width of a passage, yielding safe navigation through narrow alleys.
- The **free-space (FS)** behavior monitors the width of a passage and triggers an avoidance maneuver whenever it is too narrow for the robot to pass through.

The CA behavior has the highest priority. It can interrupt the other behaviors whenever it determines that the robot is on collision course with an obstacle in its heading direction. Similarly, the FS behavior can abort the CE and AS behaviors whenever it determines that the passage gets too narrow for the robot to pass through.

[1] An ordinal representation is one in which absolute distances are compressed, but the depth order along a given direction is preserved.

The simple behaviors introduced above do not have to operate on a rich internal representation of the 3D structure of the environment. Indeed, a representation of the environment consisting of only five quantities, which we term "virtual corridor", is sufficient to perform these navigational tasks. Figure 1 illustrates this representation. The virtual corridor represents the free space around the robot, and consists

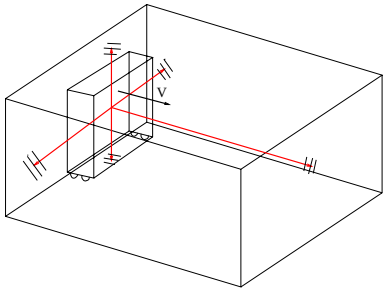

Figure 1: Virtual corridor consisting of left and right side walls, ground floor and ceiling, and frontal wall.

of a ground plane, a ceiling, left and right side walls, and a frontal wall. Each "wall" is determined by the closest obstacle in the corresponding region of robot-centric space. The information about obstacles in the heading direction required by the collision-avoidance behavior is provided by the frontal, lower and upper walls. The other three behaviors use the information about the left and right walls as input. The centering behavior navigates the robot down the middle of a passage by keeping the left and right virtual walls at the same distance. The width of the passage computed from the left and right walls is used by the adaptive speed behavior to adjust the robot speed and by the free-space behavior to determine whether the passage is wide enough for the robot. Thus, with respect to these navigational tasks, the virtual corridor is an example of a purposive representation [1], since it only represents the information about the environment necessary for the purpose of executing the navigational tasks.

According to Gibson's [7] theory of ecological perception, task-related representations should be directly extracted from sensory information. In section 3 we show that the quantities making up the virtual corridor can be easily extracted from the optical flow field. An even more direct "pick-up" [7] of the information is made possible by the introduction of the polar sector map, which we introduce in section 4.

A further property of the set of behaviors is that most of them do not require absolute obstacle distances. It is sufficient if they know an ordinal representation of these distances within each image sector, since this would not affect which object is considered closest. The only exception is the free-space behavior which needs to be able to compare the obstacles' distances to the robot's own width. We address this issue in section 5.4.

2.3 Image partition

In our system, the virtual corridor is extracted from visual information captured by a forward-facing camera rigidly attached to the robot. The partition of robot-centric space into five regions is paralleled by a partition of the image into five sectors : a central sector, and four peripheral sectors (left, right, lower, and upper). This image partition is illustrated in Figure 4. In each image sector the optical flow is computed, and a representative flow vector corresponding to the closest obstacle in the associated region of space is chosen. It is these five flow representatives that define the virtual corridor.

3 Optical Flow

3.1 General Motion

Consider a robot moving in a stationary environment with translational velocity $\mathbf{T}$ and instantaneous rotational velocity $\boldsymbol{\omega}$. Let a camera be rigidly mounted on the robot platform. We place the origin of the camera coordinate system at the optical center of the camera, align the Z-axis with the optical axis,

coordinates in the camera coordinate frame, moves with velocity $\dot{\mathbf{R}} = -\mathbf{T} - \boldsymbol{\omega} \times \mathbf{R}$ relative to the robot. Let $\mathbf{r} = (x, y)^T = (X/Z,\ Y/Z)^T$ be the image projection of $\mathbf{R}$ according to the pin-hole camera model. The optical flow (or *image motion*) $\dot{\mathbf{r}} = (\dot{x}, \dot{y})^T$ (obtained by differentiating $\mathbf{r}$ with respect to time) is given by [10] :

$$\dot{x} = \frac{1}{Z}(xV_z - V_x) + xy\omega_x + (1 + x^2)\omega_y + y\omega_z \tag{1}$$

$$\dot{y} = \frac{1}{Z}(yV_z - V_y) + (1 + y^2)\omega_x + xy\omega_y - x\omega_z \tag{2}$$

3.2 Ground-plane motion

For ground-based vehicles the "ground-plane motion" model, in which translation is limited to the XZ-plane and rotation is only around the Y-axis, is especially relevant. Typically, the dominant motion of such a robot is forward, in which case the robot looks in the heading direction (the optical axis of its camera is aligned with its translational direction) and does not rotate. Its motion parameters are then given by $\mathbf{T} = (0, 0, V)^T$ and $\boldsymbol{\omega} = \mathbf{0}$. Since in this case $\dot{Z} = -V$, the image flow becomes :

$$\dot{\mathbf{r}} = (\dot{x},\ \dot{y})^T = \frac{V}{Z}\ (x,\ y)^T = \frac{V}{Z}\ \mathbf{r} \tag{3}$$

This form of the flow equation highlights the main characteristics of image flow due to translational motion [8] :

1. its direction is always *radial*,
2. its magnitude increases linearly with eccentricity $r = \|\mathbf{r}\|$, thus flow is *small in the center* and *large in the periphery*,
3. its magnitude scales linearly with robot forward velocity V,
4. if facing a perpendicular wall, i.e. for $Z(x, y) = Z_0$, the flow is proportional to $1/\tau$, where $\tau = Z_0/V$ is the "time-to-collision" (TTC) [9, 8, 22, 18].

In practice, the assumptions about perfectly translational robot motion do not strictly hold. Although the heading direction can generally be kept parallel to the optical axis, i.e. $\mathbf{T} = (0, 0, V)^T$, non-negligible rotational motion components (principally ω_x and ω_y) are caused by uneven ground and turning maneuvers. These components must first be eliminated (by estimating them, and by subtracting the associated flow components from the flow field) before the flow field can be interpreted according to the simplified eq. (3). Since flow vectors in the central region are small, a precise compensation of the rotational component is necessary if the central flow is to be evaluated for collision detection.

3.3 3D Structure from motion

Assuming that the robot moves straight ahead without rotating, the 3D structure of the environment can be reconstructed if the velocity V of the robot is known. If it is not known, the 3D structure can only be reconstructed up to scale. This represents the well-known "velocity-scale" ambiguity of the structure-from-motion problem. The reconstruction up to scale is given by :

$$\frac{1}{V}(X,\ Y,\ Z) = (\frac{x^2}{\dot{x}},\ \frac{y^2}{\dot{y}},\ \frac{\|\mathbf{r}\|}{\|\dot{\mathbf{r}}\|}) \tag{4}$$

which can be verified by substituting eq. (3) into the above equation. The equation shows that velocity-scaled distances along the different coordinate axes can be extracted from the image flow. As noted in section 2.2, this provides the information necessary for the CA, CE, and AS behaviors.

The last section showed that the optical flow field contains the necessary information to build the virtual corridor. However, real-time execution constraints make a reduction of the input data necessary. Besides this computational constraint, there are other constraints on the design of a space-variant transformation that we would like to impose. In this section we analyze the CLM model with respect to these constraints, and then introduce the PSM which provides the desired ordinal representation of the periphery.

4.1 Design constraints

The optical flow of the homogeneously resolved image does not present the sought-for information in a directly accessible way. Firstly, the magnitude of the optical flow depends on eccentricity, as evidenced by the x^2, y^2, and $\|\mathbf{r}\|$ factors in eq. (4). Secondly, the flow has non-zero components along *both* x- and y-axes, although it is intrinsically one-dimensional. We would like the space-variant transformation to take into account these functional/structural constraints. In addition, the following behavioral constraints need to be accommodated. (1) In the central region the resolution must be high, since during forward motion the resulting flow vectors are small. A reduction of the resolution in the center would impair the ability of the robot to detect obstacles and avoid collisions with them. (2) In the periphery the resolution can be lower. This is acceptable for two reasons : (a) peripheral flow vectors are larger than central ones and could still be detected reliably even if compressed; (b) the robot does not move towards the objects in the periphery, and therefore the danger of collision is not as high as with objects in the heading direction. These considerations motivate the introduction of a space-variant computational map with high resolution at the center and gradually decreasing resolution towards the periphery.

4.2 The Complex-Logarithmic Map (CLM)

The CLM was first proposed by [16] as a model of the retino-cortical mapping in primates. See also [13]. It was later picked up by the computational vision community [23, 8, 18], and is also known under the name "log-polar mapping". In this model, the image plane is identified with the complex plane :

$$z = x + i\ y = re^{i\phi} \tag{5}$$

The complex-logarithmic mapping separates the angular component from the log-scaled radial component :

$$\zeta = \log z = \log r + i\ \phi \tag{6}$$

Since under forward motion the original image flow is given by $\dot{z} = (V/Z)\ z$, the flow in the transformed image is simply :

$$\dot{\zeta} = \frac{1}{z}\ \dot{z} = \frac{V}{Z} \tag{7}$$

The CLM has some advantages for optical flow processing, since (1) the optic flow is restricted to one coordinate (no flow along the angular component), and (2) the flow is equal to the reciprocal of the time-to-contact, and thus directly encodes the relevant quantity for the collision-avoidance behavior. One disadvantage of the CLM is the singularity of the logarithm at the origin. In a practical implementation, this problem is circumvented by cutting out a disk around the origin of the image [23, 13, 18].

4.3 Ordinal representation of the periphery

Recall from the introduction that the space-variant transformation should be defined in such a way that the peripheral optical flow field yields an ordinal representation of the surroundings. To understand the importance of this requirement, consider a column alley through which the robot travels at constant speed, as shown in Fig. 2. In such a situation, it is desirable that each column generate a constant flow over time as the robot travels through the alley. In other words, the optical flow for the columns forming the alley should be a monotonic function of the side-ways distance d only, and should not depend on the changing distance $Z(t)$ along the heading direction of the robot. Note that the image flow in both the original and

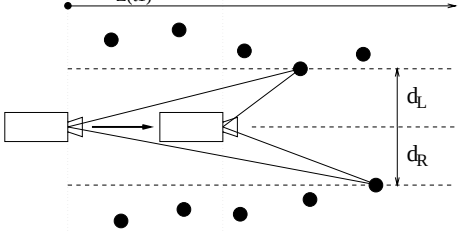

Figure 2: Robot in column alley. Peripheral optical flow in the PSM is a function of the side-ways distance d, not of the changing distance $Z(t)$ in the heading direction.

the CLM-transformed image does not satisfy this requirement, since it varies with eccentricity. In the first case it depends quadratically on eccentricity (see eq. (4)), in the second case linearly, as is seen by substituting $x = X/Z$ in eq. (7) :

$$\dot{\zeta} = \frac{V}{Z} = \frac{V}{X}\, x \tag{8}$$

The main advantage of an ordinal representation is that the robot can directly interpret it to decide how to adjust its lateral position within the hallway, which would not be possible without additional computations with an eccentricity-dependent flow field.

4.4 Definition of the polar sector map (PSM)

The polar sector map is a compound map consisting of a different space-variant transformation for each of five image sectors. Figure 3 shows an example of the PSM transformation. This transformation satisfies

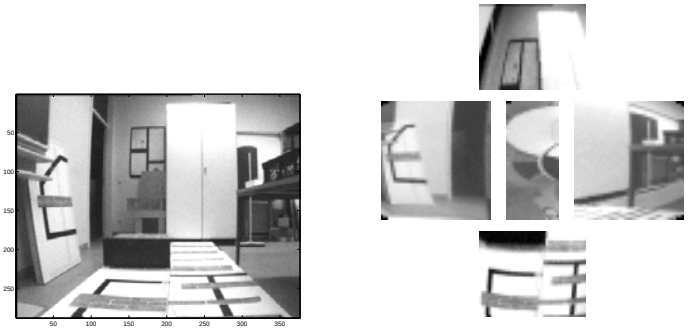

Figure 3: Image transformation by the PSM. Left : Original camera image. Right : PSM-transformed image. The data reduction is 10:1 in each of the peripheral sectors and 11:1 in the central CLM.

the following criteria :

1. One coordinate of the PSM transformation $(x, y) \mapsto (u, v)$ is a monotonic function of the **polar angle** : $v = f(\phi)$, where $\phi = \arctan(y/x)$, with the additional constraint $f(0) = 0$.

2. Different "radial" mapping functions $u(\xi)$ $(\xi \in \{r, x, y\})$ are applied **along radial lines** in each image sector. For comparison, the CLM $u = \log r$ defines a *global* mapping, and may, therefore, be considered as a special case of the PSM in which for all sectors the same function for u has been defined.

3. The radial mappings are chosen in such a way that **flow magnitude is a monotonic function of obstacle distance** along the associated coordinate axis (see section 4.5). For comparison, in the CLM $\dot{u} = V/Z$ holds globally, i.e. even in the periphery.

Central	$a \log(r/r_0)$	$\frac{1}{a} \cdot r$	$\frac{1}{a} \cdot r^2$	$a \cdot \frac{V}{Z}$
Right	$b\ x_0(1 - \frac{x_0}{x})$	$\frac{1}{bx_0^2} \cdot x^2$	$\frac{1}{bx_0^2} \cdot xr^2$	$bx_0^2 \cdot \frac{V}{X}$
Left	$b\ x_0(1 + \frac{x_0}{x})$	$\frac{1}{bx_0^2} \cdot x^2$	$\frac{1}{bx_0^2} \cdot xr^2$	$-bx_0^2 \cdot \frac{V}{X}$
Upper	$c\ y_0(1 - \frac{y_0}{y})$	$\frac{1}{cy_0^2} \cdot y^2$	$\frac{1}{cy_0^2} \cdot yr^2$	$cy_0^2 \cdot \frac{V}{Y}$
Lower	$c\ y_0(1 + \frac{y_0}{y})$	$\frac{1}{cy_0^2} \cdot y^2$	$\frac{1}{cy_0^2} \cdot yr^2$	$-cy_0^2 \cdot \frac{V}{Y}$

Table 1: Individual sector maps (radial component u), radial compression $(\frac{\partial u}{\partial \xi})^{-1}$, areal compression $(\det J)^{-1}$, and optical flow $\dot{u}$.

Definitions of the individual sector maps are given in Table 1. (Parameters of the mapping are illustrated in Figure 4.) In the central sector we have a plain CLM given by $u(r) = a \log(r/r_0)$, where $a \in (0, 1]$ is a constant controlling the desired resolution and r_0 is the radius of the cut-out disk. In the right sector the mapping is given by $u(x) = bx_0(1 - \frac{x_0}{x})$, where $b \in (0, 1]$ is a constant controlling the desired resolution and x_0 is the horizontal half-width of the central sector. The other peripheral mappings are obtained from the right sector map by symmetry. It turns out that the lower sector map is almost identical[2] with Mallot's inverse perspective map (IPM) [11].

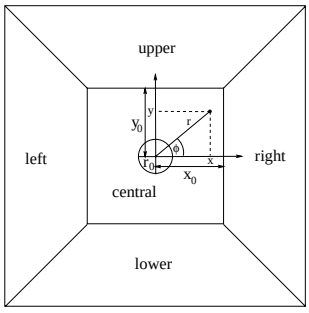

Figure 4: Parameters of the PSM (x_0, y_0 : half-width and -height of central sector, r_0 : radius of cut-out disk to avoid singularity at origin).

The different mapping functions result in different compression functions for each image sector. The *radial compression* of the original image by the PSM is given by $(\frac{\partial u}{\partial \xi})^{-1}$, i.e. by the reciprocal of the derivative of u with respect to the radial coordinate $\xi \in \{r, x, y\}$. As shown in Table 1, the compression is linear for the central sector and quadratic for the peripheral sectors. The *areal compression* is given by the reciprocal of the Jacobian $J = \frac{\partial(u,v)}{\partial(x,y)}$ of the mapping. It is quadratic in the central and cubic in the peripheral sectors. Thus, a significant data reduction is achieved by representing peripheral areas more coarsely.

4.5 Optical Flow in the PSM

Since the v-coordinate of the PSM is a function of the polar angle only, the optical flow along this coordinate is zero under forward motion : $\dot{v} = 0$. This applies to the entire PSM, not just to the central log-polar mapping. Optical flow along the u-coordinate is given in Table 1, and is obtained by the chain rule : $\dot{u} = \frac{\partial u}{\partial \xi}\ \dot{\xi}$. Note that the flow magnitude in each sector is indeed a monotonic function of the distance along the associated coordinate axis. Figure 5 illustrates this feature of the PSM. Two images were taken from a simple simulated world in which a wall parallel to the YZ-plane and a column in front of it are visible in the right image sector. Whereas the flow in the original image shows a wide distribution, the right PSM flow histogram has two relatively sharp peaks, one indicating a vertical wall parallel to the heading direction

[2]Identity is achieved when v is chosen to be y/x.

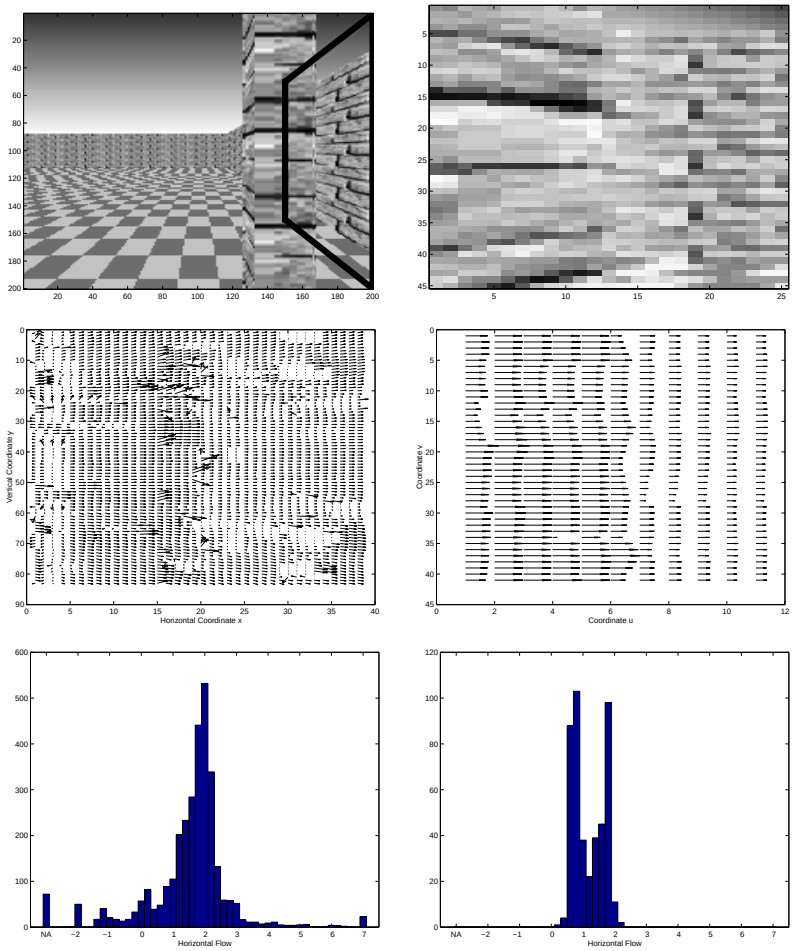

Figure 5: Optical flow for a vertical column in front of a vertical wall parallel to the heading direction (simulated environment). Top image pair shows view of the environment (right sector highlighted in original image), middle row shows optical flow in right image sector, and bottom row shows histograms of horizontal flow. Left column : original image. Right column : right sector of PSM.

and one slightly broader peak corresponding to the column[3]. The reason why the wall and the column are not separated in the flow histogram for the original image is that the optical flow in the original image is scaled by eccentricity.

4.6 Implementation Issues

Let w_x and w_y be the number of horizontal and vertical pixels of the input image. For the experiments reported in the following sections, we have chosen the central sector to be square ($x_0 = y_0$) and half as wide or high (whichever was smaller) as the input image. We always picked $a = b = c = 1$, which is the least possible compression of the radial dimension. (Choosing these parameters greater than one would lead to super-resolved images.) Table 2 shows image dimensions and PSM parameter settings used in our

Image	w	$x_0(y_0)$	FOV	FOV_c	n_u	n_v	R_y
Sim (x)	200	50	45°	27°	26	45	6.7
Sim (y)	200	50	45°	27°	26	45	6.7
Real (x)	384	72	42°	19°	46	45	11.4
Real (y)	288	72	34°	14°	37	45	9.6

Table 2: Image and PSM parameter settings used for simulated and real images.

simulated environment and with real camera images grabbed by the robot. The parameters are : image dimension (w), half-width (x_0), resp. half-height (y_0) of central sector, half field of view (FOV), half field of view of central sector (FOV_c), peripheral PSM sector dimensions (n_u, n_v), and data reduction factor for peripheral sectors (R). For the angular dimension (v-coordinate) of each peripheral sector, we used the function $v = \phi/2$, yielding a resolution of 45 pixels, each pixel corresponding to 2° polar angle. The data reduction factor R is defined here as the number of pixels in the original image sector divided by the number of pixels in the corresponding PSM sector.

The mapping from original image to the PSM was performed by averaging for each PSM pixel the original image pixels corresponding to its preimage. A fast implementation was realized using look-up tables[19]. Optical flow was computed in the PSM using a correlation-based method using zero-mean normalized cross-correlation as a similarity measure [5], with a correlation patch size of 5 pixels, followed by subpixel localization using quadratic interpolation of the correlation surface. In the peripheral sectors search was performed along the u-direction only with a search range from -1 to 6 pixels. The same method was used for the central sector in the simulated environment. On the real robot, however, we performed a 2D search in the central sector of the original image. This yields more accurate flow estimates which are necessary for (1) reliable TTC estimation, and (2) for the compensation of rotational motion components present in real vehicles (see section 3.2).

5 Purposive statistics and control of behavior

5.1 Extraction of the virtual corridor

According to the definition of the virtual corridor in section 2.2 the distance of each virtual wall is determined by the closest obstacle in the corresponding region of space. Since flow in each PSM sector is inversely proportional to distance, one would ideally like to extract the flow with the largest magnitude in each sector. However, because of image noise and mismatches in the flow detection procedure computing the maximal flow does not provide very reliable results. Instead, we apply more robust statistics to extract the 5 flow representatives $\Phi_{left}, \Phi_{right}, \Phi_{lower}, \Phi_{upper}, \Phi_{center}$ forming the virtual corridor :

[3]The broader peak for the column is due to the fact that a column generates a *range* of flow magnitudes corresponding to the range in depth (here measured along the X-axis) it occupies.

this yields the largest flow vector after discarding the largest 20%, which are presumably due to noise and mismatches.

- In the **central** sector the **median** (= **0.5-quantile**) of the flow vectors is computed. The rationale for this choice is the following :
  - The median is more robust than higher quantiles, since central flow vectors are smaller and less reliably estimated than peripheral flow vectors.
  - Although the median overestimates the distance of the closest obstacle, the likelihood of detection increases for the next frame. This follows from the fact that a stationary obstacle in heading direction will be even closer in the next frame; its image will therefore also be larger, and more likely to be detected.

Referring back to Figure 5, the flow histogram of the right PSM sector illustrates the importance of using the 0.8-quantile of the flows in the peripheral sectors. In that situation, the right flow representative Φ_{right} would be given by the position of the peak corresponding to the column. As a result, Φ_{right} would correctly encode the column as being the closest obstacle, a property that is crucial for avoiding collisions. This is in contrast to the mean or the median, both of which would encode an incorrect distance. The mean would encode some intermediate distance between the column and the wall, and the median would encode that of the wall.

5.2 The centering behavior

For the CE behavior, which controls the robot's heading direction, we have implemented a leaky integrator controller with output thresholding. The input to the controller at time t is the difference between left and right flow representatives : $\Delta\Phi^t = \Phi^t_{left} - \Phi^t_{right}$. The state variable I is updated as follows :

$$I^{t+1} = \alpha(I^t + \Delta\Phi^t), \tag{9}$$

where $0 < \alpha < 1$ is a forgetting constant currently set at 0.9. The output is one of three discrete actions : keep the current heading direction if $|I^{t+1}| < \tau$, turn by 9° if $I^{t+1} \geq \tau$, or turn by -9° if $I^{t+1} \leq -\tau$, where τ is the fixed output threshold. Initially, and after each turning action I is reset to 0. The turns are ballistic movements, executed with high speed. While the robot is turning, optical flow is not evaluated. Figure 6 shows an externally recorded sequence in which the robot navigates through the middle of a hallway using the CE behavior.

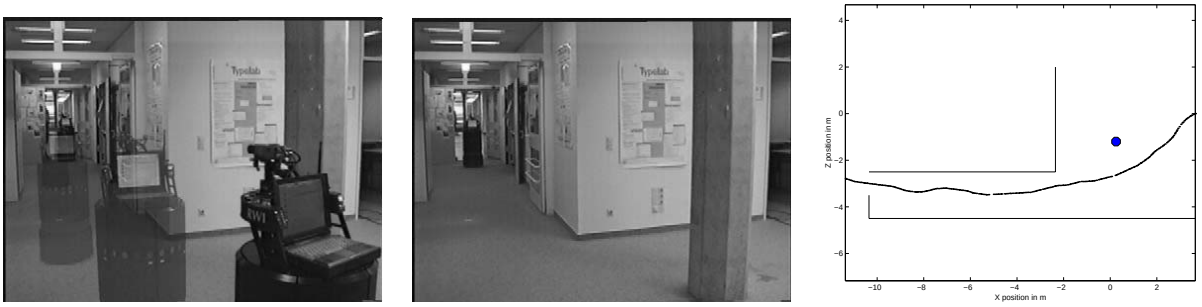

Figure 6: Experiment showing successful navigation down the middle of a hallway. Left : superposition of three snapshots from the beginning of the sequence. Middle : snapshot towards end of sequence with robot at end of hallway. Right : top view of the robot trajectory (from odometry) in the hallway (sketch based on laser scan).

If an obstacle is high enough to appear in the image center over a period of several frames, then a collision with the obstacle can be computed by evaluating the TTC from the central image flow. In our implementation of the CA behavior, the robot turns away from an obstacle by rotating 110° around its axis whenever the TTC remains below a fixed threshold for three consecutive frames. Currently, our system moves at a speed of 200mm/sec and operates at a frame rate of 3 Hz. Figure 7 shows an externally recorded sequence in which the robot performs a TTC-based avoidance maneuver.

Figure 7: Experiment showing successful collision avoidance. The obstacle (wall) is detected from the TTC values in the central sector. Left : Two superimposed snapshots before the turn. Middle : Two superimposed snapshots after the turn. Right : Robot trajectory from odometry (dotted) and environment (from laser scan)

5.4 Detection of low-lying obstacles

Figure 8 shows an experiment with a real image sequence recorded by the "Soccerbot" robot approaching a box. A sufficiently high obstacle would be detected by its small TTC in the central sector. However, since this box is fairly low its image quickly wanders out of the central sector without being detected there. It must therefore be detected in the lower sector. At first, no flow due to the box is present in the lower sector, only flow due to the ground plane. By the 21st frame the box occupies the entire left half of the lower sector, but only now has the textured upper side of the box entered it. The top of the box is detected in the flow field, which is also reflected in the small additional peak in the flow histogram. In the 26th frame, the well-textured top of the box accounts for the majority of the flow vectors, and the peak due to the box dominates the flow histogram. As shown in Figure 9, the trace of the 0.8-quantile of the flow starts rapidly increasing at the 21st frame, indicating an obstacle above ground. In contrast, the 0.2-quantile maintains a value around 1.5 pixels/frame throughout the sequence, corresponding to the motion due to the ground floor. This measure is much more robust than the mean, which is strongly affected by outliers (not shown), or even the median, which breaks down when the obstacles makes up more than half of the flow vectors in the lower sector (as is the case in frame 26). If the floor is always sufficiently textured, then thresholding the difference of the 0.8 and 0.2 quantiles represents a simple method for detecting above-ground obstacles.

5.5 Behavior integration

We have implemented the first three behaviors, namely CA, CE, and AS, in a virtual simulation environment embedded in MATLAB. The AS behavior controls the forward speed of the robot by keeping $\Phi_{left} + \Phi_{right}$ constant using a simple proportional controller. So far, we have ported and integrated the CA and the CE behaviors onto the two real robot platforms "Ulm Sparrow Soccerbot" (local development) and "B21" (developed by *RWI Inc.*). Experiments such as the ones depicted in Figures 6 and 7 show that the robot is able to successfully navigate through hallways containing columns or other obstacles and avoid driving into walls. We have implemented by not yet integrated the parts of the architecture which rely on the knowledge of the robot velocity. This concerns the detection of low above-ground obstacles

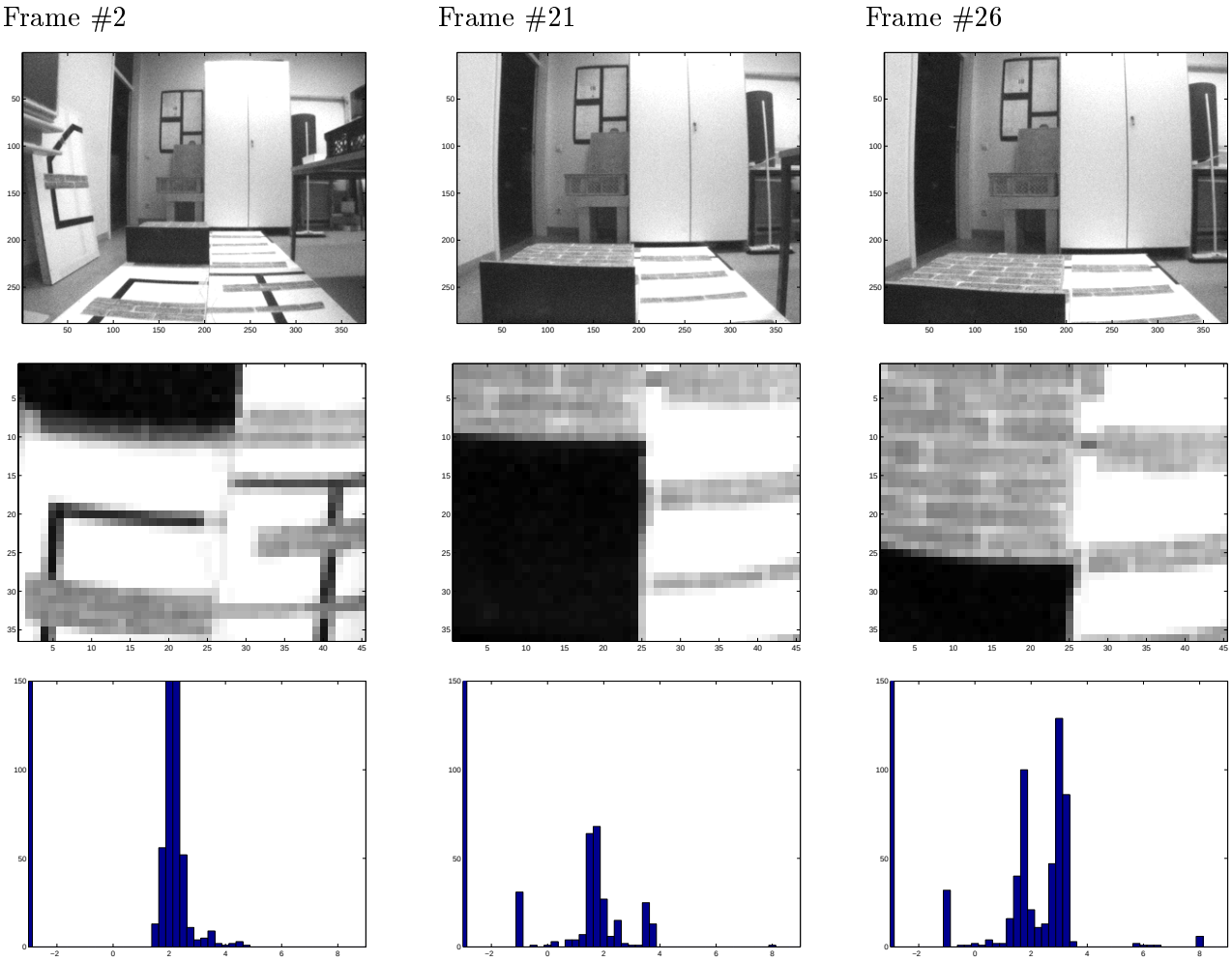

Figure 8: A few frames (2, 21, 26) from a real image sequence. Top row : original images. Second row : lower PSM sector. Third row : histograms of lower sector flow.

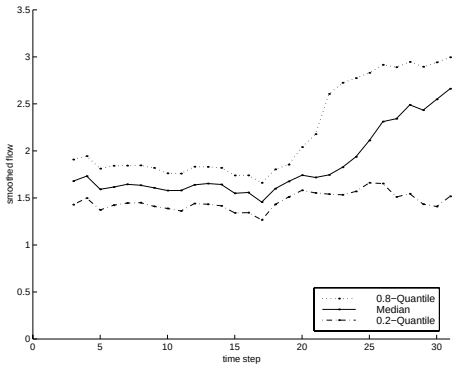

Figure 9: Traces of 0.2-/0.5-/0.8-quantiles for real image sequence.

measurements from the central sector by the CA behavior. The robot velocity is also required by the FS behavior for determining the width of the passage from the left and right flow representatives Φ_{left} and Φ_{right}. The simplest way to obtain the robot velocity is to simply fix it to some known value. Indeed, this is the approach taken in other systems for collision detection based on obstacle height above ground [12]. Note, however, that this precludes the adaptive speed behavior from being used. Another way to obtain the robot velocity is to use the odometric information provided by the robot base. Alternatively, the velocity-distance ambiguity can be resolved by estimating the flow field due to the ground floor [14]. The data presented in section 5.4 show, however, that robust methods – such as the proposed evaluation of the 0.2-quantile – are necessary. We are currently adapting the simulation environment and the robot implementation to include the free-space behavior in the robot's repertoire. For the implementation on the real robot platforms we are planning to use a sensor-fusion approach, where the robot velocity is determined from combined odometric and flow-based information.

6 Discussion and Conclusions

We have presented a processing architecture for a mobile robot navigating in an unstructured environment based on information from the optical flow. Biology has inspired us at two levels : (1) at the image representation level, where we were led to consider space-variant transformations similar to the first stage of the visual processing architecture of primates, and (2) at the behavioral level, where we implemented simple optical flow based behaviors modeled after the flight behavior of bees. Clearly, the obtained model architecture is neither that of a bee nor that of a primate. In fact, bees have two laterally looking eyes, not one forward looking one, and they do not perform a space-variant processing of the image. With these constraints, a more appropriate solution to collision detection might consist in using "zig-zag" motions observed in flies[20], rather than the TTC-based approach implemented here, since the latter requires a high resolution in the central visual field. Still, in our view it is not a contradiction to combine aspects of primates' and bees' processing architecture when the goal is not to model a particular animal, but to make judicious use of biological principles to solve the navigation and collision avoidance tasks for a mobile robot.

A central aspect of our work is the use of space-variant transformations, which represent a powerful mechanism for optimizing the processing of sensory data in view of a particular set of behavioral tasks[12]. Starting out with a set of four robot behaviors, we derived a purposive representation of the environment in terms of a virtual corridor, and designed a space-variant transformation, the PSM, which subserves all four behaviors in a single cortical image format. The advantages of the PSM are multi-faceted : computational (data reduction, real-time execution), functional (direct signaling of TTC or side-ways distance), behavioral (wide field of view, high resolution in center for collision detection), and structural (flow along one dimension only). In particular, a novel aspect of this transformation is the remapping of the peripheral image regions, which is defined in such a way that the peripheral optical flow yields an ordinal representation of the environment which is invariant under translational forward motion. This invariance allows efficient extraction of the virtual corridor and quick action determination.

Acknowledgments

This research was performed in the collaborative research center on the "*Integration of symbolic and sub-symbolic information processing in adaptive sensory-motor systems*" (SFB-527) at the University of Ulm, funded by the German Science Foundation (DFG). The authors would also like to thank the anonymous reviewers for their insightful comments.

References

[1] Y. Aloimonos. Purposive and qualitative active vision. In *Proc. Image Understanding Workshop*, pages 816–828, 1990.

Systems, 9(6):843–858, 1992.
[3] D. Coombs, M. Herman, T. Hong, and M. Nashman. Real-time obstacle avoidance using central flow divergence and peripheral flow. *IEEE Trans. on Robotics and Automation*, 14(1):49–59, 1998.
[4] D. Coombs and K. Roberts. Centering behavior using peripheral vision. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'93)*, pages 440–445, New York, NY, 1993.
[5] O. Faugeras, P. Fua, B. Hotz, R. Ma, L. Robert, M. Thonnat, and Z. Zhang. Quantitative and qualitative comparison of some area and feature based stereo algorithms. In W. Förstner and S. Ruwiedel, editors, *Robust Computer Vision*. Wichmann, 1992.
[6] C. Fermüller and Y. Aloimonos. Ordinal representations of visual space. In *Proc. Image Understanding Workshop*, pages 897–903, February 1996.
[7] J. J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, Boston, MA, 1950.
[8] R. Jain, S. L. Bartlett, and N. O'Brian. Motion stereo using ego-motion complex logarithmic mapping. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9:356–369, 1987.
[9] D. Lee. Plummeting gannets. *Perception*, 5:437–459, 1976.
[10] H. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. In *Proc. Royal Society of London B*, volume 208, pages 385–397, 1980.
[11] H. Mallot, H. Bülthoff, J. Little, and S. Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biol. Cyb.*, 64:177–185, 1991.
[12] H. Mallot, W. v. Seelen, and F. Giannakopoulos. Neural mapping and space-variant image processing. *Neural Networks*, 3:245–263, 1990.
[13] G. Sandini and V. Tagliasco. An anthropomorphic retina-like structure for scene analysis. *CVGIP*, 14:365–372, 1980.
[14] J. Santos-Victor and G. Sandini. Embedded visual behaviors for navigation. *Robotics and Autonomous Systems*, 19:299–313, 1997.
[15] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo in autonomous navigation : From bees to robots. *Int. J. Comp. Vision*, 14:159–177, 1995.
[16] E. Schwartz. Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception. *Biol. Cyb.*, 25:181–194, 1977.
[17] M. Srinivasan. How bees exploit optical flow: Behavioral experiments and neural models. *Phil. Trans. of the Royal Society of London B*, 337:253–259, 1992.
[18] M. Tistarelli and G. Sandini. On the advantages of polar and log-polar mapping for direct estimation of time-to-impact from optical flow. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15:401–410, 1992.
[19] C. Toepfer, M. Wende, G. Baratoff, and H. Neumann. Robot navigation by combining central and peripheral optical flow detection on a space-variant map. In *14th Int. Conf. on Pattern Recognition (ICPR'98)*, pages 1804–1807, Brisbane, Australia, 1998.
[20] H. Wagner. Flight performance and visual control of flight of the free-flying housefly. iii. interactions between angular movement induced by wide- and smallfield stimuli. *Phil. Trans. Roy. Soc. Lond, B*, 312:581–595, 1986.
[21] K. Weber, S. Venkatesh, and D. Kieronska. Insect based navigation and its application to the autonomous control of mobile robots. In *Proc. 3rd Int. Conf. on Automation, Robotics, and Computer Vision*, pages 1228–1231, Singapore, 1994. Nanyang University Press.
[22] C. Weiman. Log-polar vision for mobile robot navigation. In *Electronics Imaging: International Electronic Imaging Exposition and Conference*, pages 382–386, 1990.
[23] C. Weiman and G. Chaikin. Logarithmic spiral grids for image processing and display. *Computer Graphics and Image Processing*, 11:197–226, 1979.
[24] M. Wright and A. Johnston. Visual motion and cortical magnification. In D. Rose and V. Dobson, editors, *Models of the Visual Cortex*, pages 233–241. John Wiley & Sons Ltd, 1985.