Centering behavior with a mobile robot using monocular foveated vision

Marc Ebner[1] and Andreas Zell

Eberhard-Karls-Universität Tübingen, Wilhelm-Schickard-Institut für Informatik,
Arbeitsbereich Rechnerarchitektur, Köstlinstraße 6, 72074 Tübingen, Germany

Abstract

A system for corridor following based on properties of the human visual system is presented. The robot extracts image features using an interest operator to compute sparse optical flow induced by the translatory motion of the robot. The available status information from the robot is used to compensate for the known rotatory movement of the image. Control of the robot is done by transforming the optical flow to ego-motion complex log space. The difference between the median flow extracted from the left and right peripheral visual areas is used to control the heading of the robot. Forward velocity is controlled by trying to keep the perceived optical flow constant.

Key words: Visual Robot Control; Centering Behavior; Complex Log Mapping

1 Motivation

Control of a mobile robot may be achieved by constructing a complete three dimensional model of the world, planning a path and then executing the required steps to move the robot along the path. Of course it is desirable to know as much as possible about the environment. However, this strategy requires much computing power which may be used for other, higher level tasks if a simple control strategy can be employed at the same time. Horswill [17] has pointed out that it is unnecessary to construct a three dimensional model of the environment simply to reduce it to a single number such as heading inside a corridor.

[1] Present address: Universität Würzburg, Lehrstuhl für Informatik II, Programmiersprachen und Programmiermethodik, Am Hubland, 97074 Würzburg, Germany. ebner@informatik.uni-wuerzburg.de

Aloimonos [1] introduced purposive vision which only uses the data required to solve the particular task, thereby saving valuable computer resources. This makes it possible to close the loop and achieve vision during action [31]. Bekey [2] offered constructive proofs that biologically inspired robot control can succeed where traditional approaches are difficult or impossible. Like Sandini et al. [31] we are working in the field of visual robot control with an emphasis on vision during action. In this context much may be learnt by looking at how insect vision works [15,16]. Some of this knowledge might be transferable to the field of robot vision [12].

In particular, researchers have been inspired by insects like bees for robot control [6,31,33,34,32]. Bees are unable to measure distances larger than a few centimeters using stereo vision because of the small distance between their eyes. However, bees are able to infer the range of an object from the perceived angular velocity provided that the ego-motion of the eye and the bearing of the object are known [36]. Bees are performing centering behavior in a corridor by balancing the speeds of the retinal images in the two eyes [37,36]. The higher the speed of the retinal image, the smaller the distance to the wall. Thus the bee is located exactly in the center of a corridor if the image speeds of the left and right eye are equal. Bees seem to be able to perform this centering behavior independent of the contrast or spatial-frequency content of gratings along the walls.

However, humans have a more advanced visual system which is different from the visual system of the bee. An introduction to the human visual system is given by Tovée [41]. In the human visual system the axis of fixation is facing forward with a rather small field of view [4]. The spatial resolution is much higher in the center (fovea) than in the periphery of the retina. Humans perceive the world as stationary even though the eye may be moving. Information about the desired eye movements is used to compensate for the motion in the image induced by the ego-motion of the eye [42]. If the eye is moved passively by using a finger or one tries to change the gaze with a mechanically fixated eye the world seems to move. Ego-motion of the head is determined by the vestibular apparatus [4]. Information from the vestibular apparatus could be used to improve the perceived image. In this respect our work has been more inspired by the human visual system than the visual system of the bee.

The paper is organized as follows. Section 2 reviews relevant research in the field of visual robot control in general and corridor following in particular. Section 3 reviews the complex logarithmic mapping whose properties are important for our work. Section 4 describes our control algorithm in detail. The computation of the ego-motion of the camera from the available status information supplied by our mobile robot is also described. Section 5 describes some of the experiments performed with our mobile robot. The paper ends with a conclusion in section 6.

2 Background

Much research has been done in the field of visual obstacle avoidance and navigation. In this section we briefly review some of these approaches. Research related to the task of corridor following using optical flow is discussed in more detail below.

2.1 Visual robot navigation

Moravec [25] used a mobile robot with a sliding camera inspired by the motion of lizards to determine a three dimensional map of interesting points in an environment. A path is planned though the environment and subsequently updated. The robot moves in discrete steps of 1m.

Brady and Wang [3] determine scene structure in the environment of a mobile robot by extracting corner points. For each point depth is computed using a stereo camera or a structure from motion approach. Brady and Wang suggest that it might not be necessary to explicitly compute depth. Useful information for robot control may be derived directly from optical flow, disparity or rate of change of disparity. Indeed, Tomasi and Kanade [40] have shown that it is possible to extract shape and motion from image sequences without calculating camera-centered depth.

Fossa et al. [11] developed a vision based navigation system using three cameras. Two cameras are used for obstacle detection and avoidance and the third camera is used for self-localization.

Franceschini et al. [12] developed an artificial compound eye with facets which was inspired by the fly's eye. Although control is done continuously the movement of the robot is separated into a purely translational step followed by a rotational step. Franceschini et al. use this type of motion to determine the translatory component of the optical flow.

Horswill [17] developed a simple, cheap and robust visual navigation system. Both the rotational and the translational velocity are under computer control. Corridor following is done by estimating the heading of the robot and calculating a measure for the distance to the left and right walls from visual data. Control of the robot is done in a closed loop except for turning at corners which is performed open-loop. Horswill emphasizes that only the required data needed for the task at hand has to be extracted from the visual data, e.g. it is not necessary to construct a complete 3D model of the environment just to achieve a corridor following behavior.

Crespi et al. [8] followed a memory-based approach to navigation inside a corridor. First, images labelled with known lateral displacement and heading of the robot inside the corridor are used during a learning phase. Lateral displacement is discretized into three values and heading into three or four values. During robot control a probability distribution is calculated for the classes which specifies the new translational and rotational velocity.

Sobey [38] developed a zigzag motion strategy for a monocular robot to avoid collisions. First an orientation movement is performed to determine the distance to the objects 60° on both sides of the goal. All subsequent movements are made in a direction for which a safe distance to travel is known using a potential field method. A change of direction between 20° and 120° is made for every move which results in a zigzag motion. Range values are computed only for a horizontal strip of the image. Sobey has used an action-perception-planning cycle which acquires 16 images for every translatory motion.

Vogelgesang et al. [43] developed an algorithm for depth perception from radial optical flow fields. They propose to use their algorithm with a mobile robot. However if the heading of the robot changes the algorithm has to be reset.

Jochem and Pomerleau [21] developed a very successful vision-based system for driving on outdoor roads which is also able to perform tactical maneuvers such as lane changing. The system is based on an artificial neural net trained by watching a human teacher.

Košecká [22] represented the environment as a place graph and used visual servoing to move from one place to the next. It is assumed that world coordinates of the landmarks used for navigation are known.

2.2 Centering behavior

Coombs and Roberts [6] used two cameras to achieve a centering behavior inspired by the honeybee. The cameras are located at an angle of 60° from the front of the robot. Coombs and Roberts determine gradient-parallel optical flow at full spatial resolution for each of the cameras. The maximum optical flow is determined by a histogram method. Both maxima are compared to calculate the desired heading of the robot. Coombs and Roberts note, that this approach works reliable as long as the focus of expansion does not enter the image. The desired heading is saturated to prevent this from happening. The gaze of the camera is stabilized as the robot rotates due to the architecture of the robot. An alternate strategy proposed by Coombs and Roberts would be to subtract the optical flow, which is observed around the focus of expansion, from the peripheral optical flow.

Coombs et al. [5] use peripheral normal flow obtained from a camera with a wide lens to achieve centering behavior. Another camera with a narrow lens is used to estimate time-to-contact from flow divergence to detect imminent collisions. This information is used to turn the robot around in a “dead end”. The cameras are rotationally stabilized by rotating them in the direction opposite to the rotation of the robot’s body. A saccade is made if the camera orientation differs too much from the heading of the robot to align them again.

Santos-Victor et al. [33,34] equipped a mobile robot with two laterally pointing cameras to achieve a centering reflex also motivated by the behavior of honeybees. This basic centering reflex was extended with a sustaining mechanism which allows the robot to navigate whenever optical flow is only available from one of the cameras. Due to their camera setup Santos-Victor et al. assume that only horizontal optical flow can occur which simplifies calculation of optical flow. Average optical flow from the left and right camera is used to control the robot. Santos-Victor et al. derived constraints that have to be met to control the robot during rotations. By choosing the minimum translational velocity, the maximum rotational velocity and the setup of the cameras appropriately these constraints may be met. Santos-Victor et al. also note the possibility of suppressing robot control during times when the constraints are not met. Both possibilities worked successfully with their robot. They control the rotational velocity of the robot with a PID controller. Translational velocity is controlled by keeping the desired optical flow at a rate of 2 pixels/frame.

Neven and Schöner [27] developed a behavior based vision-guided approach to homing and obstacle avoidance. Time to contact is estimated from optical flow which is used to control heading direction and velocity. In the absence of reliable visual information the robot uses information from dead reckoning. Neven and Schöner compute optical flow and time to contact from coarsely sampled images with two cameras (left and right). Neven and Schöner separate translatory and rotatory motions of the robot. Both forward velocity and angular velocity are controlled by the estimated time to contact information. Time to contact estimates close to the center of the image are excluded because the peripheral information is more reliable. Small rotatory motions that may occur even if the robot is supposed to perform only a translatory motion along the optical axis are removed by subtracting the average flow along the horizontal and vertical meridian.

The approach presented here works with a single camera with the focus of expansion in view. Data is gathered in a purposive way. Control is done continuously in a closed loop. Our approach differs from the previous ones by Coombs and Roberts, Coombs et al., Santos-Victor et al., and Neven and Schöner. Information about the ego-motion of the camera obtained from the robot’s status information is used to compensate for the rotatory motion of the camera. By using sensed velocities instead of the desired velocities a sep-

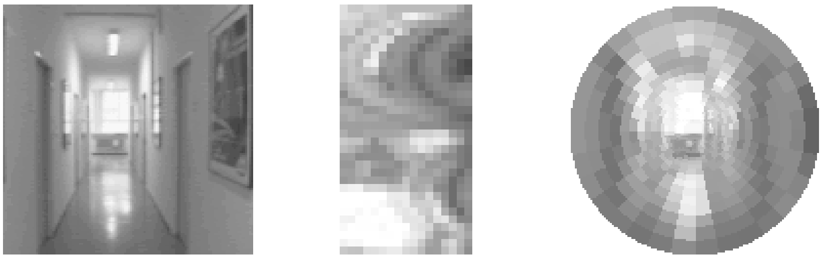

Figure 1. Image, image in complex log space and its inverse.

arate algorithm may be used to control the gaze of the camera. Obviously a centering behavior cannot be achieved with one camera pointing sideways. Nevertheless it should be possible to achieve centering behavior as long as both walls are visible and sufficient visual information is present in the visual areas which are used to control the robot.

3 Ego-motion complex logarithmic mapping

Since we are working with a camera with the focus of expansion in view we use the complex logarithmic mapping to transform the optical flow calculated in image coordinates to complex log space. This transformation simplifies the comparison of radial optical flow. The complex logarithmic mapping has already been extensively studied by several researchers and has been used in a number of different areas such as character recognition [28], template matching [44], extraction of moving objects [18,13], motion stereo [19] and object detection and centering [29]. Complex logarithmic mapping is pseudo-invariant to size, rotation, and projection scaling [35]. For a fixated object, changes of size or rotation lead to a linear shift of an invariant image. The complex logarithmic mapping performs a conformal mapping of the log polar plane to a Cartesian plane [39]. The ego-motion complex logarithmic mapping is defined as [20]:

$$r(x,y) = \log\sqrt{(x - x_{\mathrm{FOE}})^2 + (y - y_{\mathrm{FOE}})^2} \tag{1}$$

$$\theta(x,y) = \tan^{-1}\left(\frac{y - y_{\mathrm{FOE}}}{x - x_{\mathrm{FOE}}}\right) \tag{2}$$

An image transformed with this mapping is shown in figure 1. The ego-motion complex logarithmic mapping differs from the complex logarithmic mapping in that it is not taken about the origin. Instead it is taken about the focus of expansion $(x_{\mathrm{FOE}}, y_{\mathrm{FOE}})$. For a forward moving observer stationary objects in the field of view only produce horizontal flow in ego-motion complex log space [18]. Jain et al. [19] have shown that (assuming a focal length $f = 1$):

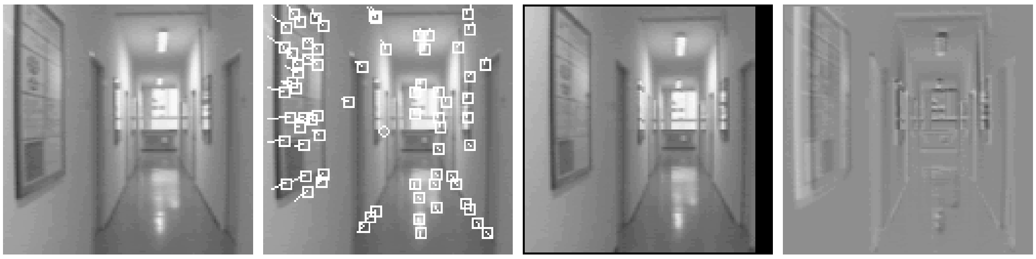

Figure 2. Image, image with sparse optical flow, previous image compensated for rotatory camera movement, difference picture between image and previous image compensated for rotatory camera movement.

$$\frac{dr}{dZ} = -\frac{1}{Z} \qquad \text{and} \qquad \frac{d\theta}{dZ} = 0 \tag{3}$$

Thus the optical flow in ego-motion complex log space is a direct measure for the distance Z to the objects in view for a known translatory motion.

Tistarelli and Sandini [39] also analyzed the polar and log polar mapping and determined several advantages that may be gained by using a polar or log polar mapping. In addition to the already mentioned invariance property to scaling and rotations the complex log mapping also performs a data reduction due to the non-uniform sampling of the image. Motion equations which relate ego-motion to the optical flow are simplified and time to impact may be easily calculated using a complex log mapping.

Schwartz [35] has shown that the retino-striate mappings of the rhesus, squirrel and owl monkey may be described by a complex logarithmic mapping. With the direction of the optical axis facing forward humans could be using the complex logarithmic mapping to achieve obstacle avoidance or centering behavior. We are not suggesting that human centering behavior is actually achieved the way we constructed our algorithm. We are investigating if it is indeed possible to achieve centering behavior with monocular vision and the focus of expansion directly in view for robot control using the complex logarithmic mapping.

4 Control of a mobile robot using foveated vision

The control algorithm is divided into three parts. First relevant information is extracted from the images by locating interesting points with an interest operator. These points are then used to compute a sparse optical flow field which is transformed into complex log space. An image with a sparse optical flow field in image coordinates is shown in figure 2. Finally, the difference between the left and right optical flows is used to control the robot. The input to the algorithm consists of an image $I(t_2)$ taken at time t_2. The algorithm works on two images at any given time. The data from the previous iteration of the control algorithm, image $I(t_1)$, is stored internally.

4.1 Feature extraction

Interesting points are located in the input images which simplify the computation of optical flow. Let the interesting points of image $I(t_2)$ be $F(t_2)$. Because of the aperture problem [20] dense optical flow describing the actual motion is difficult to compute. Most methods compute optical flow in an iterative way (e.g. Horn and Schunck [14]). Due to temporal Gaussian smoothing, several images are often needed [34] which introduces an additional delay because optical flow is only computed for the center image. Santos-Victor et al. [34] therefore separate image acquisition and evaluation by grabbing a set of five images at video rate and then analyzing the data. In contrast to the work of Santos-Victor et al. we only grab one image for each iteration of the control algorithm.

To detect interesting points in the image, we are using the Moravec interest operator [24,25]. It is a simple operator which computes the local maxima of the minimum variance of local image intensities in the horizontal, vertical and both diagonals. Computation of optical flow for a set of interesting points is easier than the computation of full optical flow due to the aperture problem [20]. The type of features relevant to control the robot obviously depends on the environment of the robot. Using genetic programming [23] (a class of evolutionary algorithms which can be used to evolve variable sized, hierarchical individuals) we are evolving interest operators which are optimal according to some measure. This would make the first stage of the algorithm, extraction of interesting points for the calculation of optical flow, adaptive. We previously evolved edge and interest operators [9,10] by directly incorporating the output of the desired operator into the fitness calculation and are now working towards adaptive feature extraction where only the desired qualities of the operators are specified.

4.2 Sparse optical flow in ego-motion complex log space

The ego-motion complex log mapping is only defined for a translatory movement of the observer. However to control an arbitrary moving mobile robot we compensate for any rotatory movements of the camera. A difference image between the current image and the previous image, compensated for rotatory movements of the camera is shown in figure 2. This image shows how accurate the compensation of rotatory ego-motion is performed. We compensate for the rotatory movement of the camera by transforming the image to subtract this

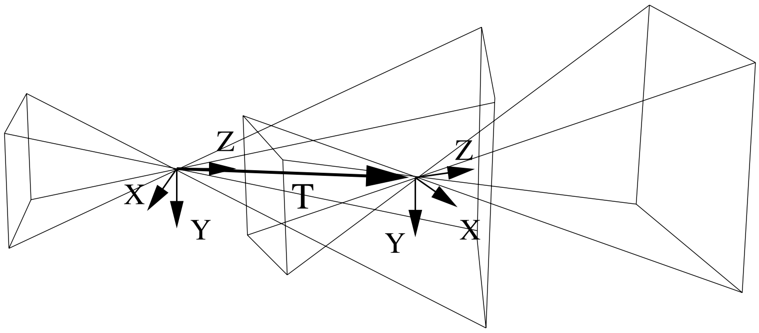

Figure 3. Movement of the camera.

component from the optical flow. This leaves only the translatory component of the optical flow which is relevant to the centering behavior. We are using the robot's status information (translational and rotational velocity, pan and tilt angles) to predict the image $\tilde{I}(t_2)$ at time t_2 from Image $I(t_1)$ due to the rotatory motion of the camera. It is assumed that the robot moves exactly as is derived from the status information. Indeed, the human visual system makes a similar assumption as has already been described above.

In the following text we are using the notation of Craig [7] to describe coordinate transformations. Let ${}^{C(t_1)}\mathbf{P} = [X(t_1), Y(t_1), Z(t_1), 1]$ be a point in the camera frame $\{C(t_1)\}$ at time t_1. Then the coordinates of this point are ${}^{C(t_2)}\mathbf{P} = [X(t_2), Y(t_2), Z(t_2), 1]$ in the camera frame $\{C(t_2)\}$ at time t_2. Let ${}^{C(t_2)}_{C(t_1)}\mathbf{T}$ be the homogeneous transformation from the camera frame $\{C(t_1)\}$ to the camera frame $\{C(t_2)\}$ (figure 3). Thus we have

$$ {}^{C(t_2)}\mathbf{P} = {}^{C(t_2)}_{C(t_1)}\mathbf{T} \cdot {}^{C(t_1)}\mathbf{P} \tag{4} $$

$$ \begin{bmatrix} X(t_2) \\ Y(t_2) \\ Z(t_2) \\ 1 \end{bmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} X(t_1) \\ Y(t_1) \\ Z(t_1) \\ 1 \end{bmatrix} \tag{5} $$

Using perspective projection $x(t) = f\frac{X(t)}{Z(t)}$ and $y(t) = f\frac{Y(t)}{Z(t)}$ with focal length f this gives the following transformation for every image point for a rotatory camera motion [26]:

$$ x(t_2) = f\frac{r_{11}x(t_1) + r_{12}y(t_1) + fr_{13}}{r_{31}x(t_1) + r_{32}y(t_1) + fr_{33}} \tag{6} $$

$$ y(t_2) = f\frac{r_{21}x(t_1) + r_{22}y(t_1) + fr_{23}}{r_{31}x(t_1) + r_{32}y(t_1) + fr_{33}} \tag{7} $$

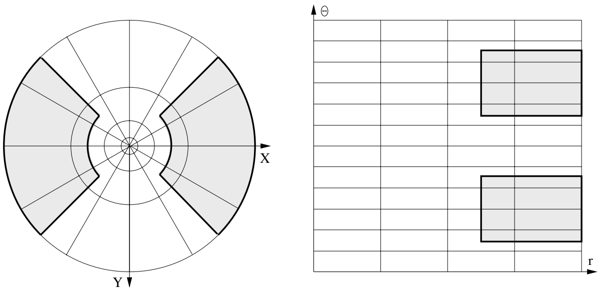

Figure 4. Areas of complex logarithmic space used to control the robot.

Sparse optical flow is computed by matching image intensities in a local area around the interesting points. We are matching interesting points $F(t_2)$ from the new image at time t_2 with the interesting points $\tilde{F}(t_2)$ from the image at time t_2 which has been compensated for camera rotations. The search is constrained to those points that radiate outward. The quality of the match (squared differences in a local area around the points) is scaled with the distance of the match. Thus preference is given to a close match. After the interesting points in the predicted image $\tilde{I}(t_2)$ and the actual image $I(t_2)$ have been matched, sparse optical flow has been determined in the original image.

The optical flow radiates outward from the focus of expansion for a forward motion of the camera. Provided that peripheral optical flow is available from the left and right sides, these flow vectors may be used to achieve a centering behavior with the focus of expansion in view. The focus of expansion is calculated from the status information of the robot.

$$\begin{bmatrix} x_{\text{FOE}} \\ y_{\text{FOE}} \end{bmatrix} = f \begin{bmatrix} t_x/t_z \\ t_y/t_z \end{bmatrix} \tag{8}$$

The sparse optical flow is then transformed to complex log space according to

$$r = \frac{r_{\max}}{\log r_{\text{src}}} \log \sqrt{(x - x_{\text{FOE}})^2 + (y - y_{\text{FOE}})^2} \tag{9}$$

$$\theta = \frac{\theta_{\max}}{2\pi} \left(\pi + \tan^{-1} \left(-\frac{x - x_{\text{FOE}}}{y - y_{\text{FOE}}} \right) \right) \tag{10}$$

where r and θ are the coordinates in ego-motion complex log space and r_{src} is the maximum radius in coordinates of the source image. The number of pixels in complex log space used for the radial direction is specified by $r_{\max}$ and $\theta_{\max}$ is the number of pixels used for the angular direction. Thus we have $r_{\min} \leq r \leq r_{\max}$ and $0 \leq \theta \leq \theta_{\max}$ with $r_{\min} = \frac{r_{\max}}{\log r_{\text{src}}} \log \frac{\theta_{\max}}{2\pi}$. Optical flow is computed first for the acquired image and then transformed to complex log space because the untransformed images have a much larger resolution than the transformed ones. Of course optical flow could be computed directly for

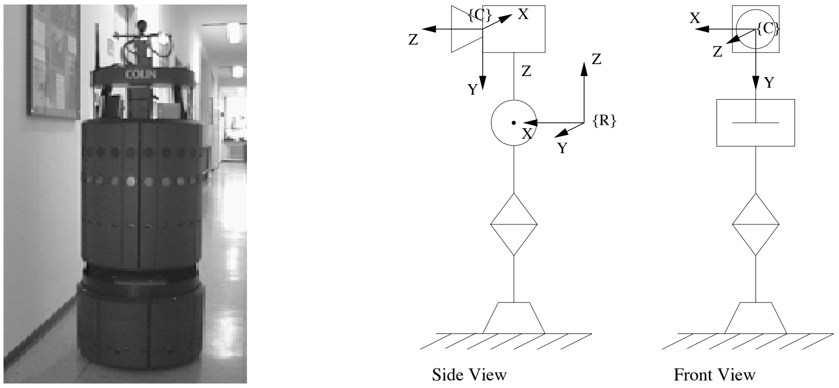

Figure 5. Colin, a Real World Interface B21 robot and the camera model with attached frames.

the already transformed images. However due to the reduced image size, the number of interesting points that can be extracted from the transformed image is reduced. We experimented with several different methods, e.g. extraction of features from the transformed image, calculation of full optical flow for the original images and the transformed images. The approach described here was found to be the one that worked best.

We only use the optical flow located in the area $\theta_{\mathrm{lmin}} \leq \theta \leq \theta_{\mathrm{lmax}}$, $\theta_{\mathrm{rmin}} \leq \theta \leq \theta_{\mathrm{rmax}}$ and $r'_{\mathrm{min}} \leq r \leq r_{\mathrm{max}}$ as shown in figure 4. For our experiments described below we chose the values $\theta_{\mathrm{lmin}} = 45°$, $\theta_{\mathrm{lmax}} = 135°$, $\theta_{\mathrm{rmin}} = 225°$ and $\theta_{\mathrm{rmax}} = 315°$. This effectively excludes the optical flow from the ceiling and the floor which cannot be used to control the robot. Also we excluded the area directly around the focus of expansion since the optical flow is usually small in this areas and might be further disturbed due to incorrect compensation for the rotatory motion of the camera. Since it is impossible to determine exact data from sensor information we exclude this area because its information is too unreliable. For our experiments we have chosen $r'_{\mathrm{min}} = 0.75(r_{\mathrm{max}} - r_{\mathrm{min}}) + r_{\mathrm{min}}$. Let f_l and f_r be the median of the optical flow extracted from the left and right peripheral areas respectively.

4.3 Computation of camera motion

The homogeneous transformation of the camera movement may be calculated directly from the status information of the robot. Let $\{C\}$ be the camera coordinate system and $\{R\}$ the coordinate system of the robot base. The coordinate frames for our camera setup are shown in figure 5. We split the camera transformation into three parts. The first describes the location of the camera relative to the robot's base at t_2, the next describes the movement of the robot and the last describes the location of the camera relative to the robot at time t_1. Then the homogeneous transform ${}^{C(t_2)}_{C(t_1)}\mathbf{T}$ which transforms coordinates in the camera frame $\{C(t_1)\}$ to coordinates in the camera frame $\{C(t_2)\}$ may be calculated as

$$ {}^{C(t_2)}_{C(t_1)}\mathbf{T} = {}^{C(t_2)}_{R(t_2)}\mathbf{T} \cdot {}^{R(t_2)}_{R(t_1)}\mathbf{T} \cdot {}^{R(t_1)}_{C(t_1)}\mathbf{T} \tag{11} $$

where ${}^{C(t_2)}_{R(t_2)}\mathbf{T}$ is the transformation from the coordinate system of the robot $\{R(t_2)\}$ to the coordinate system of the camera $\{C(t_2)\}$ at time t_2, ${}^{R(t_2)}_{R(t_1)}\mathbf{T}$ describes the robot movement and ${}^{R(t_1)}_{C(t_1)}\mathbf{T}$ transforms the coordinate system of the camera $\{C(t_1)\}$ to the coordinate system of the robot $\{R(t_1)\}$ at time t_1. ${}^{R(t_2)}_{R(t_1)}\mathbf{T}$ is calculated from the status information of the robot as

$$ {}^{R(t_2)}_{R(t_1)}\mathbf{T} = \mathbf{R}_Z(-\omega(t_2 - t_1)) \cdot \mathbf{D}_X(-v(t_2 - t_1)) \tag{12} $$

where $\mathbf{D}_X$ describes the translatory motion of the robot along the X-axis with translational velocity v and $\mathbf{R}_Z$ describes the rotatory motion of the robot where ω is the rotational velocity. The matrixes relating the camera coordinate system to the base coordinate system are computed using standard manipulator kinematics.

4.4 Robot control

A PID controller [7] was implemented to control the rotational velocity of the robot. The gains have been determined experimentally and were set to $k_p = 5.0$, $k_i = 0.1$ and $k_d = 0.005$ during the experiments described here. The control law is given by

$$ \omega_d(t) = k_p e(t) + k_i \int e(\tau) d\tau + k_d \dot{e}(t) \tag{13} $$

where $e(t) = f_l - f_r$ is the difference between the extracted flow on the left and right peripheral visual areas in complex log space and ω_d specifies the desired rotational velocity of the robot. In case no interesting points can be determined on one side, the robot moves towards the other side with a constant velocity. This behavior is inspired by the work of Sobey [38]. It is safer to move into a direction for which the distance to the nearest obstacle is known. If the robot turns to the other wall it will be repelled by it as soon as some of the interesting points of the other wall enter the peripheral area.

The translational velocity v_d of the robot is set according to the control law

$$v_d = \begin{cases} v_{\max} & \text{if } \frac{f_d}{f_a} v_{\max} > v_{\max}, \\ \frac{f_d}{f_a} v_{\max} & \text{if } v_{\min} \leq \frac{f_d}{f_a} v_{\max} \leq v_{\max}, \\ v_{\min} & \text{if } \frac{f_d}{f_a} v_{\max} < v_{\min} \end{cases} \tag{14}$$

where $v_{\min}$ and $v_{\max}$ are the minimum and maximum translational velocities, f_d is the desired radial optical flow in the original image and f_a is computed from the maximum of f_l and f_r which is transformed back into the original image. The control law could also have been formulated by transforming the desired optical flow to complex log space. This control law tries to keep the optical flow constant at about f_d by slowing the robot down as it approaches a wall. Speed of the robot is increased again as the actual optical flow decreases. Santos-Victor et al. [34] also used velocity control to keep the measured optical flow constant using a sigmoid function for the velocity-control loop.

A distributed client server architecture is used to control the robot. Therefore to obtain accurate status information from the robot a delay of 150ms is used after a change of velocity and the old image is discarded. This allows us to achieve accurate compensation of the rotatory motion of the camera. The PID controller produces a smooth trajectory with only small rotations. Therefore we used a much simpler control strategy for most of our experiments. This strategy turns the robot to the left with constant velocity if the optical flow in the peripheral area is larger on the right side than on the left and to the right if the optical flow on the left is larger than the flow on the right. Only in the rare case where both flows would be exactly equal we stop the rotatory movement. However this almost never happens. This control strategy produces an oscillatory behavior where the robot is constantly turning to one side or the other. Because we want to perform robot control during rotatory motions this is a suitable strategy to test our robot.

5 Experiments

For our experiments we used Colin, a Real World Interface B21 mobile robot (figure 5) with a Directed Perception pan-tilt unit. We only used one camera for the experiments described here. The camera setup can be seen in figure 5. Images were acquired with a resolution of 128×128 pixels and the transformation to complex log space was done with $r_{\max} = 32$, $r_{\text{src}} = 64$ and $\theta_{\max} = 32$. The experiments were done in the corridor at our lab. The width of the corridor was approximately 1.41m. The environment was not modified by attaching structured wall paper which would have simplified the computation of optical flow. However the environment did contain several posters mounted on the walls along the corridor.

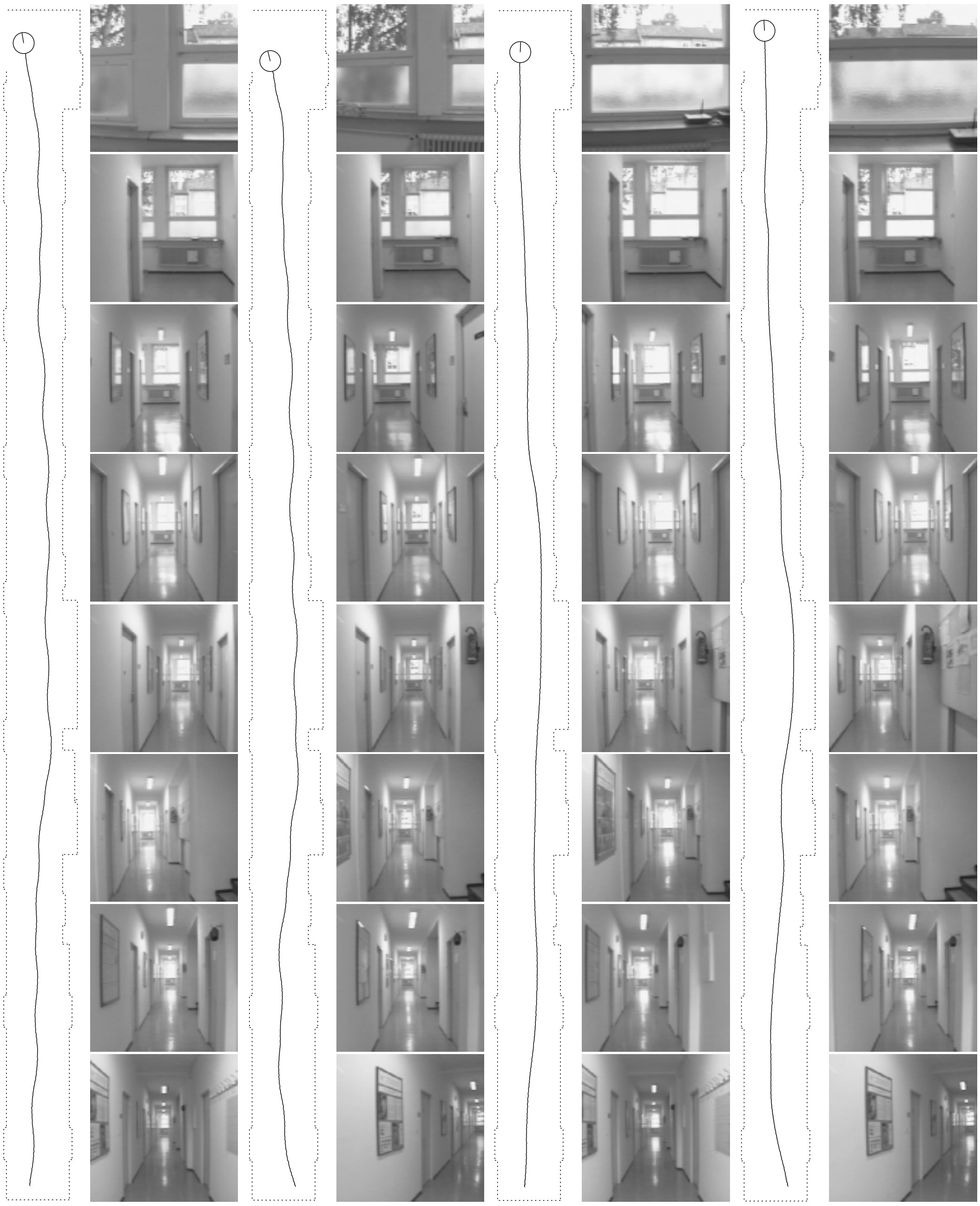

Figure 6. Path of the robot recorded from odometry data for different starting positions. The two on the left were recorded using the simple controller and the two on the right were recorded with the PID controller. The images along the path are a subset of the images that were recorded during execution of the algorithm. The path is overlayed on a map of the environment.

With this setup we conducted a series of experiments with the simple controller and with the PID controller. During the experiments with the simple controller the robot was moving with a constant forward velocity of $40\frac{\mathrm{cm}}{\mathrm{s}}$. During the experiments with the PID controller minimum translational velocity $v_{\min}$ was set to $20\frac{\mathrm{cm}}{\mathrm{s}}$ and the maximum translational velocity $v_{\max}$ was set to $55\frac{\mathrm{cm}}{\mathrm{s}}$. The desired optical flow in the original image was set to 3 pixels. Figure 6 shows the results for the simple controller and for the PID controller. We recorded the path travelled by the robot using the robot's odometry data. The images shown are a subset of the same images which were used to control the robot during the runs.

As can be seen in the images after a distance of approximately 11m the corridor is 0.35m wider than the remainder of the corridor. This part of the corridor is approximately 3.21m long. Just before this part of the corridor there is a staircase located on the right side of the corridor. In these areas the path of the robot is slightly offset to the right side of the corridor. The path recorded with the PID controller is much smoother than the one using the simple controller. During the runs which started in the middle of the corridor 126 images (1.77 frames per second) were recorded with the simple controller and 125 images (1.73 frames per second) with the PID controller. The robot travelled 27.1m respectively 26.9m during the runs. Distance was measured using the robot's odometry.

6 Conclusion and ongoing research

We have demonstrated that centering behavior may also be achieved with a single camera with the focus of expansion in view. Comparison of the optical flow for the left and right peripheral areas is done in ego-motion complex log space. Since only the optical flow due to translation of the robot may be used for the centering behavior we compensate for the image movement which is due to the rotatory component of the camera motion. To compensate for the image movement we use the status information which is directly available from the robot. Using the available status information may speed up robot control by eliminating the need to derive the ego-motion of the robot from the images.

Acknowledgement

This work was supported in part by a scholarship according to the Landesgraduiertenförderungsgesetz to Marc Ebner.

For image processing we have used the Vista software environment [30].

References

[1] Y. Aloimonos, editor. *Active Perception.* Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, 1993.

[2] G. A. Bekey. Biologically inspired control of autonomous robots. *Robotics and Autonomous Systems*, 18:21–31, 1996.

[3] M. Brady and H. Wang. Vision for mobile robots. *Phil. Trans. R. Soc. Lond. B*, 337:341–350, 1992.

[4] R. H. S. Carpenter. *Movements of the Eyes.* Pion Limited, London, second edition, 1988.

[5] D. Coombs, M. Herman, T.-H. Hong, and M. Nashman. Real-time obstacle avoidance using central flow divergence, and peripheral flow. *IEEE Transactions on Robotics and Automation*, 14(1):49–59, February 1998.

[6] D. Coombs and K. Roberts. “bee-bot”: using peripheral optical flow to avoid obstacles. In D. Casasent, editor, *Intelligent Robots and Computer Vision XI, Proceedings of the Society of Photo-Optical Instrumentation Engineers*, pages 714–721, 1992.

[7] J. J. Craig. *Introduction to Robotics: Mechanics and Control.* Addison-Wesley Publishing Company, Reading, Massachusetts, second edition, 1989.

[8] B. Crespi, C. Furlanello, and L. Stringa. A memory-based approach to navigation. *Biological Cybernetics*, 69:385–393, 1993.

[9] M. Ebner. On the evolution of edge detectors for robot vision using genetic programming. In H.-M. Groß, editor, *Workshop SOAVE ’97 - Selbstorganisation von Adaptivem Verhalten 1997*, pages 127–134. VDI Verlag, Düsseldorf, 1997.

[10] M. Ebner. On the evolution of interest operators using genetic programming. In R. Poli, W. B. Langdon, M. Schoenauer, T. Fogarty, and W. Banzhaf, editors, *Late Breaking Papers at EuroGP’98: the First European Workshop on Genetic Programming*, pages 6–10, Paris, France, 14-15 April 1998. The University of Birmingham, UK.

[11] M. Fossa, E. Grosso, F. Ferrari, M. Magrassi, G. Sandini, and M. Zapendouski. A visually guided mobile robot acting in indoor environments. In *Proceedings of the IEEE Workshop on Applications of Computer Vision, Palm Springs, USA, 1992*, pages 308–316. IEEE, 1992.

[12] N. Franceschini, J. M. Pichon, and C. Blanes. From insect vision to robot vision. *Phil. Trans. R. Soc. Lond. B*, 337:283–294, 1992.

[13] J. Frazier and R. Nevatia. Detecting moving objects from a moving platform. In *Proceedings of the DARPA Image Understanding Workshop, Pittsburgh, PA*, pages 348–355, 1990.

[14] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[15] G. A. Horridge. A theory of insect vision: velocity parallax. *Proc. R. Soc. Lond. B*, 229:13–27, 1981.

[16] G. A. Horridge. What can engineers learn from insect vision? *Phil. Trans. R. Soc. Lond. B*, 337:271–282, 1992.

[17] I. Horswill. A simple, cheap, and robust visual navigation system. In J.-A. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *From animals to animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior, Honolulu, Hawaii, 1992*, pages 129–136. The MIT Press, 1993.

[18] R. Jain. Segmenation of frame sequences obtained by a moving observer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(5):624–629, September 1987.

[19] R. Jain, S. L. Bartlett, and N. O'Brian. Motion stereo using ego-motion complex logarithmic mapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(3):356–369, 1987.

[20] R. Jain, R. Kasturi, and B. G. Schunck. *Machine Vision.* McGraw-Hill, Inc., New York, 1995.

[21] T. Jochem and D. Pomerleau. Life in the fast lane: The evolution of an adaptive vehicle control system. *AI Magazine*, 17(2):11–50, 1996.

[22] J. Košecká. Visually guided navigation. *Robotics and Autonomous Systems*, 21:37–50, 1997.

[23] J. R. Koza. *Genetic Programming, On the Programming of Computers by Means of Natural Selection.* The MIT Press, Cambridge, Massachusetts, 1992.

[24] H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Vision–1: p. 584, 1977.

[25] H. P. Moravec. *Ostacle Avoidance and Navigation in the Real World by a Seeing Robot Rover.* PhD thesis, Computer Science Department, Stanford University, No. STAN-CS-80-813 and AIM-340, September 1980.

[26] D. Murray and A. Basu. Motion tracking with an active camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):449–459, May 1994.

[27] H. Neven and G. Schöner. Dynamics parametrically controlled by image correlations organize robot navigation. *Biological Cybernetics*, 75:293–307, 1996.

[28] P.-W. Ong, R. S. Wallace, and E. L. Schwartz. Space-Variant Optical Character Recognition. In *11th International Conference on Pattern Recognition*, pages 504–507. IEEE, 1992.

[29] R. A. Peters II and M. Bishay. Centering peripheral features in an indoor environment using a binocular log-polar 4 dof camera head. *Robotics and Autonomous Systems*, 18:271–281, 1996.

[30] A. R. Pope and D. G. Lowe. Vista: A software environment for computer vision research. In *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 768–772. IEEE, 1994.

[31] G. Sandini, F. Gandolfo, E. Grosso, and M. Tistarelli. *Vision During Action*, In Y. Aloimonos, editor, *Active Perception*, pages 151–190. Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, 1993.

[32] J. Santos-Victor and G. Sandini. Embedded visual behaviors for navigation. *Robotics and Autonomous Systems*, 19:299–313, 1997.

[33] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo for robot navigation: Learning from bees. In *Proceedings of Computer Vision and Pattern Recognition, New York*, pages 434–439, 1993.

[34] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo in autonomous navigation: From bees to robots. *International Journal of Computer Vision*, 14:159–177, 1995.

[35] E. L. Schwartz. Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding. *Vision Research*, 20:645–669, 1980.

[36] M. V. Srinivasan. Distance perception in insects. *Current Directions in Psychological Science*, 1(1):22–26, February 1992.

[37] M. V. Srinivasan. How bees exploit optic flow: behavioural experiments and neural models. *Phil. Trans. R. Soc. Lond. B*, 337:253–259, 1992.

[38] P. J. Sobey. Active navigation with a monocular robot. *Biological Cybernetics*, 71:433–440, 1994.

[39] 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 Transactions on Pattern Analysis and Machine Intelligence*, 15(4):401–410, April 1993.

[40] C. Tomasi and T. Kanade. Factoring image sequences into shape and motion. In *Proceedings of the IEEE Workshop on Visual Motion, October 7-9, 1991, Nassau Inn, Priceton, New Jersey*, pages 21–28. IEEE Computer Society Press, 1991.

[41] M. J. Tovée. *An introduction to the visual system.* Cambridge University Press, Cambridge, 1996.

[42] E. von Holst and H. Mittelstaedt. Das Reafferenzprinzip. (Wechselwirkung zwischen Zentralnervensystem und Peripherie. *Die Naturwissenschaften*, 37(20):464–476, 1950.

[43] J. Vogelgesang, A. Cozzi, and F. Wörgötter. A parallel algorithm for depth perception from radial optical flow fields. In C. von der Malsburg, J. C. Vorbrüggen, W. von Seelen, and B. Sendhoff, editors, *Artificial Neural Networks: 6th international conference; proceedings/ICANN 96*, pages 721–725. Springer-Verlag, 1996.

[44] R. S. Wallace, P.-W. Ong, B. B. Bederson, and E. L. Schwartz. Space variant image processing. *International Journal of Computer Vision*, 13(1):71–90, 1994.

Marc Ebner was born in Stuttgart, Germany, in 1969. He received the M.S. degree in computer science from New York University, NY, in 1994, the degree Dipl.-Inform. from the Universität Stuttgart, Germany, in 1996 and the degree Dr. rer. nat. from the Universität Tübingen, Germany, in 1999. At present he is a research assistant at the Universität Würzburg, Germany. His research interests include biologically inspired systems, evolutionary algorithms in computer vision, evolutionary robotics, evolutionary algorithms, computer vision and robotics.

Andreas Zell received the Diploma in computer science from the Univ. Kaiserslautern, Germany, in 1986, the M.S. degree from Stanford University in 1987, the Ph.D. in 1989 and the Habilitation (venia legendi) in 1994 from the Univ. Stuttgart, Germany, all in computer science. From 1990-95 he has been assistant professor at the Univ. Stuttgart, Institute for parallel and distributed high-performance systems. 1991 he won the German University Software Prize for the development of the Stuttgart Neural Network Simulator (SNNS) and received an Int. MasPar Challenge Prize in 1992. Since 1995 he has been appointed as full professor to the chair of computer architecture at the University of Tübingen, Wilhelm-Schickard-Institute for Computer Science. His research interests include artificial neural networks, evolutionary algorithms, artificial life, mobile robotics and bioinformatics. He directs the mobile robotics lab at the University of Tübingen whose team was finalist in the 1998 RoboCup middle size competition at Paris. He participates in a number of joint research projects with industry. He also coordinates the new bioinformatics curriculum at the University of Tübingen.