Biol. Cybern. 64, 177–185 (1991)

Biological Cybernetics
© Springer-Verlag 1991

Inverse perspective mapping simplifies optical flow computation and obstacle detection

H. A. Mallot [1], H. H. Bülthoff [2], J. J. Little [3], and S. Bohrer [1]

[1] Institut für Neuroinformatik, Ruhr-Universität, W-4630 Bochum, Federal Republic of Germany
[2] Brown University, Department of Cognitive and Linguistic Sciences, Providence, RI 02912, USA
[3] University of British Columbia, Department of Computer Science, Vancouver, BC, Canada

Received November 6, 1989/Accepted in revised form July 25, 1990

Abstract. We present a scheme for obstacle detection from optical flow which is based on strategies of biological information processing. Optical flow is established by a local "voting" (non-maximum suppression) over the outputs of correlation-type motion detectors similar to those found in the fly visual system. The computational theory of obstacle detection is discussed in terms of space-variances of the motion field. An efficient mechanism for the detection of disturbances in the expected motion field is based on "inverse perspective mapping", i.e., a coordinate transform or retinotopic mapping applied to the image. It turns out that besides obstacle detection, inverse perspective mapping has additional advantages for regularizing optical flow algorithms. Psychophysical evidence for body-scaled obstacle detection and related neurophysiological results are discussed.

1 Introduction

Biological neural networks that subserve different information processing tasks are distinguished (if at all) by their global architecture rather than by the properties of their single neurons. In this paper, we present implementations of basic tasks of visual information processing using structural adaptations that allegedly have evolved for these tasks. More specifically, we show how two structural principles of the mammalian visual cortex, *local space-invariant operations* within one cortical area and *retinotopic mapping* (cf. Mallot and von Seelen 1989), can be applied in one of the basic tasks of visually guided behavior, i.e. obstacle detection.

In Sect. 2, we describe a parallel motion detection scheme that uses local uniform operations and a correlation-based motion detector. While this algorithm produces highly satisfactory image flow data, it is too slow for real time applications on conventional computers. On the other hand, the complete image flow contains far more information than is needed for most tasks in visually guided behavior. For the problem of obstacle detection from translatory egomotion, we study exactly how much information is necessary. In Sect. 3, we present a coordinate transform, the *inverse perspective mapping* that reduces the 2D search in complete motion detection to a 1D problem and leads directly to the desired information on obstacles. Examples for the proposed minimal algorithm for obstacle detection using synthetic images are presented in Sect. 4. A formal analysis of retinotopic mapping and its interaction with local neural processing has been presented elsewhere (Mallot et al. 1990).

2 Parallel motion detection

Optical flow is generated on the retina of an observer by objects moving relative to the observer. The true motion field $W(x, y, z)$ is a 3D vector field whose projection on the image plane is denoted by $W_I(x', y')$. Unfortunately, the measurement of this 2D field of image velocities from changes of intensity in subsequent images, $E_t(x', y')$, $E_{t+\Delta t}(x', y')$, is not possible in general. It is, however, possible to compute suitable optical flows $V(x', y')$ that are qualitatively similar to the true velocity field in most cases. Rich qualitative information on the motion of objects and their boundaries is contained in singular points and discontinuities of the optical flow (Verri and Poggio 1989). We describe a simple, parallel algorithm that computes optical flow from sequences of real images. The algorithm is consistent with human psychophysics and electrophysiological data from cortical areas $V1$ and MT (Bülthoff et al. 1989).

2.1 Constraints

Consider a discrete motion displacement

$$V(x', y') =: (u(x', y'), v(x', y')) . \tag{1}$$

We assume throughout the sequel that displacements are confined to an interval of length $2\delta + 1$ pixels:

$$(u\,\Delta t, v\,\Delta t) \in [-\delta, \delta] \times [-\delta, \delta] . \tag{2}$$

For this interval and an $N = n \times n$ image, the number of possible vector fields is:

$$(2\delta + 1)^{2N} \tag{3}$$

In general, many of these optical flow fields will be consistent with the data obtained from some motion detector. The problem of recovering optical flow from subsequent image frames is ill-posed and additional constraints are needed to compute a satisfactory solution e.g. by regularization (cf. Horn and Schunk 1981; Hildreth 1984). In the algorithm presented here, we require *uniqueness*, each image point has a unique velocity, and *continuity*, the optical flow field is locally smooth.

Smoothness is a result of physical constraints on motion that limit the spatial variation of the optical flow field. First, the projected velocity field of a planar patch under arbitrary, rigid motion in space is quadratic in (x', y') (Waxman and Wohn 1988). A constraint that is true under more restrictive conditions is: the projected velocity field of a planar patch, translating parallel to the image plane, is constant. This is true only in limited cases but is may be a satisfactory *local* approximation in many cases (Little and Verri 1989).

In our algorithm, the smoothness constraint is accounted for by a *local support area*, P_v, within which the optical flow is assumed to be constant (cf. (5)). The diameter v of P_v is determined by the assumed size of planar surface patches projected on the image plane. Local support is overlapping from pixel to pixel. Each pixel, surrounded by its patch independently chooses the optical flow to maximize the match between its patch and a patch in the previous image. The continuity constraint is enforced by the overlap of the local support areas, the regularization parameter (relative importance of smoothness vs. data) being incorporated into the patch diameter v. When $v > 1$, the local support region (assumed to be square or a Gaussian weighted window) of each point shares $v - 1/v$ of its area with the support regions of its nearest neighbors in the grid. When v is one, the data used by each pixel is entirely independent of all other pixels; as v increases, the recovered flow field becomes smoother, because the support regions of adjoining pixels are essentially identical.

2.2 Comparing image frames

Our algorithm consists of patch-wise correlation between image features (such as the filtered intensity data or intensity edges) and then choice of the motion corresponding to the peak of the correlation over the patch around the pixel under consideration. It is thus based on a correlation-like motion detector that is motivated by the well studied motion detector of the house fly (Hassenstein and Reichardt 1956). In an edge-based variant of the algorithm, we use And-Not gates (logical correlation) that are similar to the motion detection scheme in the rabbit retina proposed by Barlow and Levick (1965).

Both edge-based and intensity based "correlations" are expressed by means of a *comparison function* ϕ which assigns a positive real number to a pair of pixels taken from the two image frames. If we denote the preprocessed image (after filtering and/or feature-extraction) by I, we can write a possible choice of ϕ as:

$$\phi(I_1(x_1', y_1'), I_2(x_2', y_2')) := |I_1(x_1', y_1') - I_2(x_2', y_2')| . \tag{4}$$

2.3 The algorithm

Our motion scheme uses the constraint of constant image flow in the local support areas P_v. Recall that the projected motion of a planar patch moving orthogonal to the viewing direction is constant over the projected area of the patch. The patch diameter v depends on the distance to objects in the scene and their expected size in the image. The magnitude of δ depends on the expected velocities of objects in the scene, their distances from the camera, and the time separation Δt between frames. In perspective images, the optimal choice for these parameters varies with the position in the image plane. This space-variance can be subsumed into one coordinate transform (inverse perspective mapping, Sect. 3, 4) such that expected velocities (δ) and smoothness domains (v) become uniform.

The algorithm consists of three steps which can naturally be performed in a layered structure (Fig. 1).

Shift and compare: For each node (x', y') and each permissible displacement $(d_1, d_2) \in [-\delta, \delta] \times [-\delta, \delta]$, the comparison function ϕ is evaluated. The output of this step is a matching strength for each node and displacement, $m(x', y'; d_1, d_2) = \phi(I_t(x', y'), I_{t+\Delta t}(x' + d_1, y' + d_2))$. Displacements correspond to discrete velocities. This computation can easily be performed in a stack of layers sensitive to different velocities.

Local summation: At each node (x', y') the matching strength for corresponding displacements from the nodes in a neighborhood ("local support area") $P_v(x', y')$ are accumulated. The output of this step is a combined matching strength:

$$M(x', y'; d_1, d_2) := \sum_{(u', v') \in P_v(x', y')} m(u', v'; d_1, d_2) . \tag{5}$$

In a layered structure, this corresponds to spatial averaging within each layer.

Winner-take-all: To each node (x', y'), the displacement that received the highest matching strength M is assigned as its velocity value $V(x', y')$ by a winner-take-all scheme (non-maximum suppression). A large vote for one particular displacement is expected if the motion field is locally constant.

With this winner-take-all scheme, the number of possible flow fields evaluated in the flow computation is:

$$(2\delta + 1)^2 N \tag{6}$$

The algorithm has been implemented on a massively parallel computer (Connection Machine^{TM}) and tested

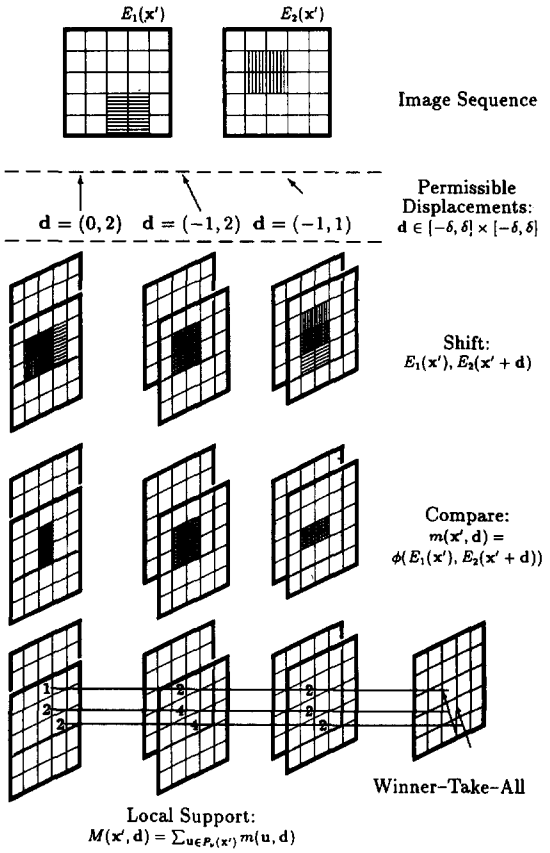


Fig. 1. Schematic of the algorithm. Consider a motion displacement by the pixel vector $(-1, 2)$. For each permissible displacement $d \in \{-\delta, \delta\} \times \{-\delta, \delta\}$, a shifted version of the second frame, $E_2(x' + d_1, y' + d_2)$ is superimposed to the first frame E_1. (Only three displacement vectors are shown in the figure). Next, the comparison function ϕ is evaluated for pairs of pixels that coincide after the shift operation; the result is the matching strength $m(x', y', d_1, d_2)$. Matches can only be obtained where the two bright regions overlap. In the next step, "local support", each processor in each layer sums the matching strengths obtained by the processors in the local support area P_v of the same layer. This sum is the "combined matching strength", $M(x', y', d_1, d_2)$; in the figure, it is indicated by numerals. Finally, for each pixel (x', y'), the displacement whose combined matching strength is largest, is selected as the motion displacement vector assigned to this pixel. Three locations are marked in the figure: for the two lower locations, the maximum of M occurs for the displacement vector $(-1, 2)$. For the third marked location, no unique maximum of M is found, indicating a motion discontinuity at this location

with natural and synthetic images (Bülthoff et al. 1989; Little et al. 1988). Recently, Bohrer et al. (1990) showed analytically that for comparison functions of the form (4), the combined matching strength is in fact maximized by the true displacement vector.

In the motion algorithm described, correspondences between image features have to be established across image frames. In contrast to the problem of stereo matching where correspondences are constrained to the epipolar lines, a complete two-dimensional search is required in motion detection. However, if optical flow is generated by translatory ego-motion, motion correspondences are in fact confined to a class of straight lines passing through the focus of expansion. The inverse perspective mapping introduced in the next section makes these lines parallel and thus transforms an important class of motion computations (translatory egomotion) into a one-dimensional problem. This makes it especially suitable for fast obstacle detection and avoidance in autonomous mobile robots.

Inverse perspective mapping

3.1 What is an obstacle?

Consider a mobile observer with two translatory and one rotatory degree of freedom to move. As is the case for a typical mammal, this means that movement is confined to a horizontal plane in which turns and translations can be performed. In this situation, an obstacle can be defined as anything rising above the plane, i.e., objects that the observer's path cannot cross. This is the minimal definition of an obstacle which does not require additional information about the nature of the object. It can be extended in the sense that any deviation from some expected surround is an obstacle, the expectation being in this casé a horizontal plane.

3.2 Inverse perspective mapping: intuitive description

Variations of optical flow can be due to both, perspective foreshortening or the 3D structure of the scene. Since we are only interested in the latter part, i.e., the presence of elevated points, we could try to eliminate the effects of perspective by means of a coordinate transform.

Figure 2 shows how this can be done. If we consider a point $\mathbf{p}$ in 3D space, perspective mapping means that

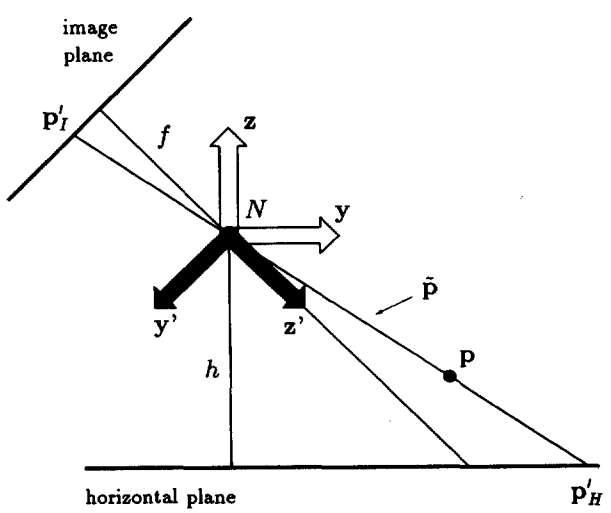


Fig. 2. Geometry of perspective mapping. N: center of projection (nodal point). $\mathbf{y}', \mathbf{z}'$ axes of the camera coordinate system. $\mathbf{y}, \mathbf{z}$ axes of the world coordinate systems. The horizontal axes $\mathbf{x}'$ (for the camera) and $\mathbf{x}$ (for the world system) are perpendicular to the paper plane and are omitted in the figure. $\mathbf{p}$: a point in 3D space; $\mathbf{p}'_I$, $\mathbf{p}'_H$: its projections into the image and the horizontal plane; $\tilde{\mathbf{p}}$: homogeneous representation of $\mathbf{p}'$. f is the focal length of the camera system and h the height of the camera nodal point above the ground plane. By φ, we denote the inclination angle of the optical axis, i.e., the angle between $\mathbf{y}$ and $\mathbf{z}'$

we draw a line through this point and the center of projection N and intersect it with the image plane to find the corresponding image point. In order to remove the distortions of the horizontal plane, we now want to undo this perspective map for points in the plane. We therefore define an "inverse perspective mapping" by the following procedure: for a point $\mathbf{p}'$ in the image plane, we trace the associated ray through N towards the horizontal plane. The intersection is the result of the inverse perspective mapping applied to the image point $\mathbf{p}'$. The term 'inverse perspective mapping' does not correspond to an actual inversion of perspective mapping which is of course mathematically impossible. Rather, it denotes an inversion under the additional constraint that inversely mapped points should lie in the horizontal plane. If we compose both perspective and inverse perspective, the horizontal plane is mapped onto itself, while elevated parts of the scene are distorted.

3.3 *Inverse perspective mapping: mathematics*

Mathematically, inverse perspective mapping is a perspective collineation (Coxeter 1987) which can be written as a linear mapping in homogeneous (or projective) coordinates (cf. Duda and Hart 1973). We state two definitions for a general coordinate system $B := \{a, b, c\}$ and an image plane at distance d from the origin parallel to a, b. A *perspective mapping* is described by

$$\mathscr{P}_B : \mathbf{R}^3 \mapsto \mathbf{R}^2$$

$$\mathbf{p} \mapsto \mathbf{p}'_B = \begin{pmatrix} p'_{B,1} \\ p'_{B,2} \end{pmatrix} := \frac{-d}{(\mathbf{p}\cdot\mathbf{c})} \cdot \begin{pmatrix} (\mathbf{p}\cdot\mathbf{a}) \\ (\mathbf{p}\cdot\mathbf{b}) \end{pmatrix}, \tag{7}$$

where $(\cdot)$ denotes the inner product. The 3D-line of all points projected to $\mathbf{p}'_B = (p'_{B,1}, p'_{B,2})^\top$ is given by $\tilde{\mathbf{p}}_B$:

$$\mathbf{p}'_B \mapsto \tilde{\mathbf{p}}_B := \begin{pmatrix} \lambda p'_{B,1} \\ \lambda p'_{B,2} \\ -\lambda d \end{pmatrix} \quad \text{for} \quad \lambda \in \mathbf{R}. \tag{8}$$

$\tilde{\mathbf{p}}_B$ is the *homogeneous representation* of $\mathbf{p}'_B$ in the coordinate frame B. By construction, we have $\mathscr{P}_B(\tilde{\mathbf{p}}_B) = \mathbf{p}'_B$ for all $\lambda \neq 0$.

Let us now introduce a world coordinate system $H := \{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ where $\mathbf{x}$ and $\mathbf{y}$ span the horizontal plane while $\mathbf{z}$ points in the upward direction. The camera model is described by a second coordinate system $I := \{\mathbf{x}', \mathbf{y}', \mathbf{z}'\}$ where $\mathbf{x}'$ and $\mathbf{y}'$ span the image plane and $\mathbf{z}'$ is the optical axis. Both frames share a common origin, the center of projection or nodal point, N, at distance h (height) and f (focal length) from the horizontal and image planes, respectively. The coordinate transform from the camera centered system to the world system is described by an orthogonal matrix Q which is composed out of the column vectors $\mathbf{x}'$, $\mathbf{y}'$ and $\mathbf{z}'$ (cf. Fig. 2).

In inverse perspective mapping, we start with a point $\mathbf{p}'_I$ in the image plane and seek the corresponding point $\mathbf{p}'_H$ in the horizontal plane, i.e., the $\mathbf{x}, \mathbf{y}$-plane of the world coordinate system H. In homogeneous coordinates, this is a linear mapping of a 'point' $\tilde{\mathbf{p}}_I$ to a 'point' $\tilde{\mathbf{p}}_H$, characterized by the orthogonal 3×3 matrix Q:

$$\tilde{\mathscr{Q}} : \mathbf{R}^3 \mapsto \mathbf{R}^3; \quad \tilde{\mathbf{p}}_H := Q \cdot \tilde{\mathbf{p}}_I$$

$$\begin{pmatrix} \mu p'_{H,1} \\ \mu p'_{H,2} \\ -\mu h \end{pmatrix} = \begin{pmatrix} x'_1 & y'_1 & z'_1 \\ x'_2 & y'_2 & z'_2 \\ x'_3 & y'_3 & z'_3 \end{pmatrix} \cdot \begin{pmatrix} \lambda p'_{I,1} \\ \lambda p'_{I,2} \\ -\lambda f \end{pmatrix}. \tag{9}$$

Here, $x'_1, \ldots, z'_3$ denote the components of the axes of the camera frame I expressed in world coordinates. We project $\tilde{\mathbf{p}}_H$ onto the horizontal plane (7) and obtain the inverse perspective mapping $\mathscr{Q}$:

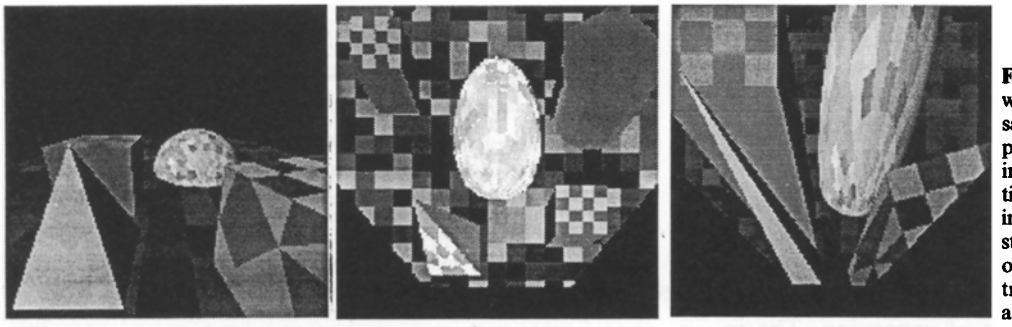
$$\mathscr{Q} : \mathbf{R}^2 \mapsto \mathbf{R}^2; \quad \mathbf{p}'_H = \mathscr{Q}(\mathbf{p}'_I)$$

$$\begin{pmatrix} p'_{H,1} \\ p'_{H,2} \end{pmatrix} := \frac{-h}{x'_3 p'_{I,1} + y'_3 p'_{I,2} - z'_3 f} \cdot \begin{pmatrix} x'_1 p'_{I,1} + y'_1 p'_{I,2} - z'_1 f \\ x'_2 p'_{I,1} + y'_2 p'_{I,2} - z'_2 f \end{pmatrix}. \tag{10}$$

By construction, the composition of perspective and inverse perspective mapping is identical to the projection through the camera nodal point onto the horizontal plane. It is only this projection $\mathscr{Q} \circ \mathscr{P}_I$ that we need to discuss in the applications. We state the central result of this section:

$$\mathscr{Q} \circ \mathscr{P}_I(\mathbf{p}) = \mathscr{P}_H(\mathbf{p})$$

$$\mathscr{Q} \circ \mathscr{P}_I : \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \mapsto -\frac{h}{p_3} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} p'_{H,1} \\ p'_{H,2} \end{pmatrix}. \tag{11}$$



a b c

Fig. 3. **a** Perspective view of a block world. **b** Bird's-eye eye view of the same scene under orthographic projection. **c** The same image after inverse perspective mapping. The tiltings of the floor are correctly imaged as squares but the elevated structures are distorted. Since the operation is a pure coordinate transform, parts of the scene that are occluded in image **a** are occluded likewise in image **c**

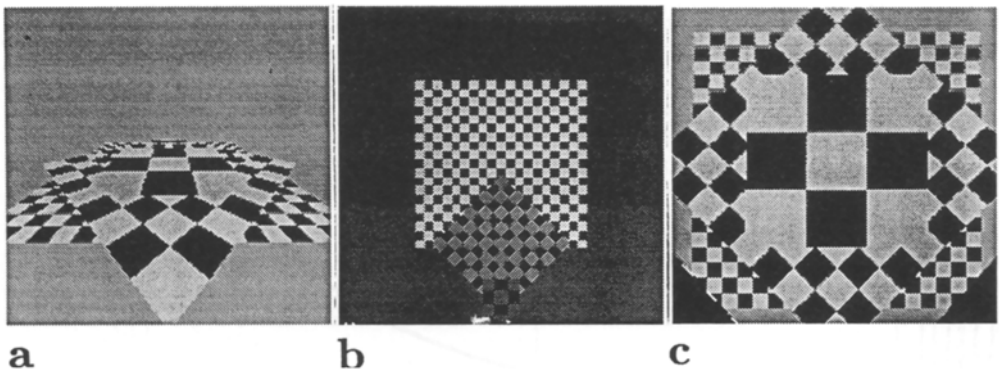


Fig. 4a–c. Inverse perspective mapping. **a** wide angle perspective view of three checker board planes at different heights. **b** bird's-eye eye view of the same scene under orthographic projection. **c** inverse perspective map of image **a**

Here, $p_1, \ldots, p_3$ denote the components of $\mathbf{p}$, expressed in the world coordinate system H.

In the final result, $\mathscr{P}_H$, the dependence on the camera coordinate system was removed by the inverse perspective map. In applications with natural images, errors in the estimated camera parameters may give rise to deviations between $\mathscr{Q} \circ \mathscr{P}_I$ and $\mathscr{P}_H$. Experimental results, however, show that the algorithm is robust against these effects (cf. Sect. 4.3, Figs. 7, 8).

4 Inverse perspective mapping and optical flow computation

4.1 Body-scaled obstacle avoidance

If the camera frame, i.e., the observer is moving in the horizontal plane at a pure constant translation $\mathbf{m}$, a stationary 3D point $\mathbf{p}$ will move relative to the camera frame with a motion vector $\mathrm{d}\mathbf{p}/\mathrm{d}t = -\mathbf{m}$. Thus, the 3D motion field $W(x, y, z)$ will be constant. In the image plane, the projected motion vector W'_I at the projection of $\mathbf{p}$ is determined by

$$W'_I(\mathscr{P}(\mathbf{p})) := \frac{\mathrm{d}\mathscr{P}_I(\mathbf{p})}{\mathrm{d}t} = -\mathbf{J}_{\mathscr{P}_I}(\mathbf{p}) \cdot \mathbf{m} \tag{12}$$

where $\mathbf{J}_{\mathscr{P}_I}$ denotes the Jacobian matrix of the perspective projection (7). Figure 5a shows the resulting optical flow field for the scene depicted in Fig. 4a and translatory egomotion. If we apply the inverse perspective mapping $\mathscr{Q}$ prior to the computation of the image flow, i.e., if we compute the image flow from the transformed image shown in Fig. 4c, the result is:

$$\begin{aligned} W'_H &= -\mathbf{J}_{\mathscr{Q} \circ \mathscr{P}_I}(\mathbf{p}) \cdot \mathbf{m} = -\mathbf{J}_{\mathscr{P}_H}(\mathbf{p}) \cdot \mathbf{m} \\ &= -h \begin{pmatrix} -1/p_3 & 0 & p_1/p_3^2 \\ 0 & -1/p_3 & p_2/p_3^2 \end{pmatrix} \cdot \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix}. \end{aligned} \tag{13}$$

The Jacobian was obtained by differentiation of (11).

Since egomotion is bound to the plane, we have $m_3 = 0$ in (13), hence

$$W'_H = \frac{h}{p_3} \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \tag{14}$$

$$\|W'_H\| = \left| \frac{h}{h - elev.} \right| \cdot \|\mathbf{m}\|, \tag{15}$$

where $elev. := h + p_3$ is the elevation of the point $\mathbf{p}$ above the ground plane, i.e., its importance as an obstacle. (Note that $p_3 < 0$ in typical cases). From here, it is easy to detect the obstacle by a local uniform operation such as the motion detector described in Sect. 2 or a unidirectional version thereof. Thresholding the result to cut off the egomotion vector itself, the obstacle can be made stand out clearly (Fig. 5c). No further information concerning the obstacle is required to trigger some 'avoidance behavior'.

The relation of the velocity of the mapped flow and the elevation of the obstacle (15) is shown in Fig. 6. The effect scales with the elevation, i.e., the importance of the obstacle, and approaches infinity for objects that are about to hit the observer's "eye".

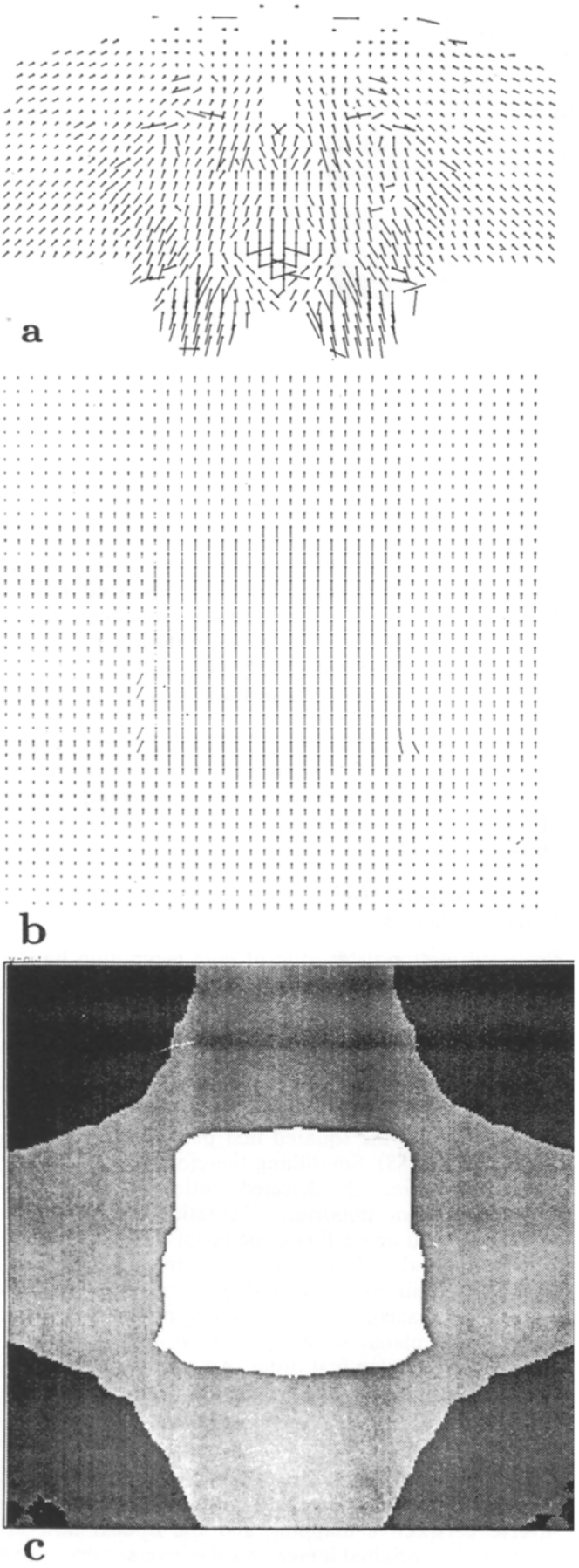
4.2 The mapped velocity field is of first order in image coordinates

Projected motion fields even of pure egomotion have a rather complicated structure. It is easy to show that the projected motion field W'_I is at least of second order in image coordinates. In regularizing approaches to the recovery of the optical flow from changing image intensities, this presents a problem: smoothness constraints tend to cancel second order terms since they assume that the sum of the squared first partial derivatives be small (Scott 1988). Smoothing therefore introduces systematic errors into the detected motion field.

We state an important theoretical result on inversely mapped image flows, the proof of which is given in the Appendix: For ego-motion in the horizontal plane (translations in $\mathbf{x}$ and $\mathbf{y}$ direction, rotations around the $\mathbf{z}$ axis), the inversely mapped motion field, $W'_H(x, y)$, of planar surface patches strictly satisfies the assumption of being first order. An equivalent formulation is that the motion is "pseudo-frontoparallel" after the inverse perspective mapping is performed.

4.3 Experimental results

Figures 7 and 8 show an experimental test of both the inverse perspective mapping and the optical flow algorithm. The original image and the inverse perspective



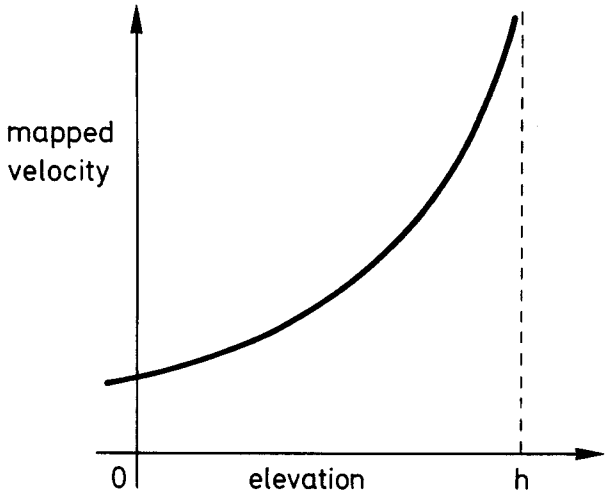


Fig. 6. Relation of projected velocity and elevation for an obstacle point under inverse perspective mapping. Image velocity depends on the relation of obstacle and observer height (body-scaled obstacle detection)

map appear in Fig. 7a, b. The angle of inclination of the optical axis against the floor was measured simply with a ruler (accuracy $\pm 2°$). The center of the camera target (i.e. its intersection with the optical axis) was known with an accuracy of ± 10 pixels. Nevertheless, the compensation of perspective foreshortening of the ground plane is almost complete. Inverse perspective mapping thus is a robust technique that can well be used in applications involving moving cameras.

The optical flow algorithm was applied to a pair of mapped images, one of which is shown in Fig. 7b. We used a local support area of 20×20 pixels ($P_v(x', y') = [x' - 10, x' + 10] \times [y' - 10, y' + 10]$). The computation took about one minute for a 256×256-image on a SUN Sparc Station 1. Figure 8a is the resulting needle-plot; the regular line-like structure reflects the high contrast along the edges of the checker-board pattern in the ground plane. After suitably thresholding the image, Fig. 8b is obtained, where only the displacement vectors in the obstacle region remain.

A related algorithm that uses inverse perspective mapping in stereopsis has been implemented on a custom mobile robot that runs as a transportation cart in warehouses or production plants (Mallot et al. 1989; von Seelen et al. 1988).

5 Discussion

5.1 Obstacle detection from optical flow

Optical flow analysis aims to compute either "generic" quantities such as the egomotion vector or the 3D-

Fig. 5a–c. Flow field for translatory egomotion in the scene depicted in Fig. 4a. **a** original image. **b** after correction by the inverse perspective map. **c** gray-level coding of the height of the planes as computed from the optical flow map in image **b**

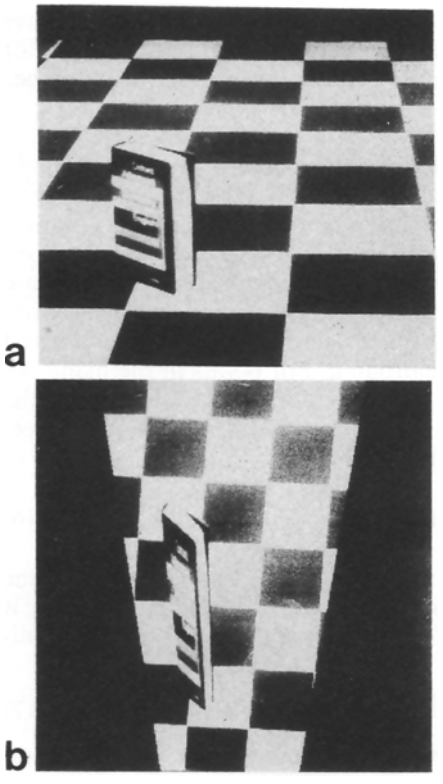


Fig. 7. **a** Natural scene with a textured ground plane and an obstacle. **b** Inverse perspective map of **a** obtained with rough estimates of the camera parameters

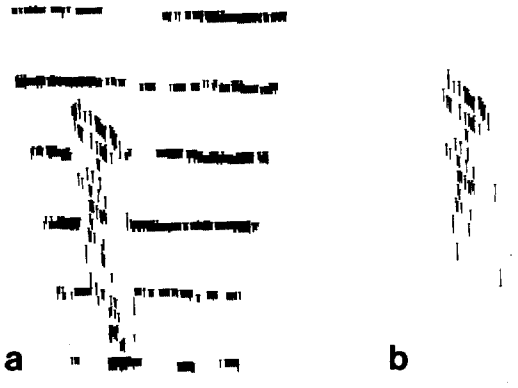


Fig. 8. **a** Motion displacements computed in a pair of mapped images including the frame shown in Fig. 7b. Local support area was 20 × 20 pixels. **b** Segmentation obtained by thresholding the motion vectors of frame **a**. The obstacle is clearly detected

structure of the environment, or "compound" quantities such as time-to-contact (looming) or the one proposed in this paper. While much work has been done to recover the "generic" properties of optical flow fields (e.g., Gibson 1950; Longuet-Higgins and Prażdny 1980; Horn and Weldon 1988), it seems quite clear that compound properties are much more useful for both animals and technical applications. Here we present a new measure that can be computed from the optical flow and specifies the relevance of an obstacle. The basic assumption underlying this measure can be stated as follows:

The observer has only three degrees of freedom to move: the translations forwards-backwards and left-right, and the rotation around the axis orthogonal on these two translations (which, for a normal, quadrupedal vertebrate is the dorso-ventral body-axis).

From this assumption, it follows that movement is confined to the horizontal plane defined by the two possible translations. Anything outside this plane is an obstacle, since the path of the observer cannot cross it. That is, we have a definition of an obstacle that does not require pattern recognition of any kind and, on the other hand, will not get confused by cast shadows or contours within the horizontal plane.

The proposed algorithm recovers the minimally required information for the detection of obstacles that are defined in the above sense. It is therefore well in line with the "ecological" idea that biological information processing should be understood from the overall behavior that it subserves (Gibson 1950) rather than from the point of view of an "ideal observer" who is able to recover whatever information is present in the input. A related approach in psychophysics is characterized by the phrase "from perception to action" (Warren and Whang 1987).

5.2 Inverse perspective mapping facilitates motion detection

Inverse perspective mapping transforms motion sequences obtained from an earthbound observer into a *pseudo-frontoparallel* motion. Frontoparallel motion has certain advantages for regularizing motion-algorithms that are in fact generated by the mapping step and exploited by our algorithm.

Due to the ill-posed nature of optical flow computation, regularization constraints are usually required to yield stable and unique solutions. Horn and Schunk (1981), for example, require that the norm of the gradient of the image flow, integrated over the entire image, be low. Hildreth (1984) uses a similar constraint along one-dimensional contours. It has been pointed out by Scott (1988) that regularization approaches of this type yield correct results only if the projection of the motion field is of first order. Systematic errors will result if higher order terms are present which, however, is usually the case. Inverse perspective mapping solves this problem for egomotion in a plane: if the image is projected onto a plane which is parallel to the direction of translatory egomotion or normal to the axis of rotation, the image flow resulting from planar patches is in fact of first order (cf. 20). In the plane of movement itself, image motion is uniform.

In the optical flow algorithm described in this paper, the advantages of inverse perspective mapping are more obvious: for best performance, an expected velocity range and a voting domain (smoothness domain) must be specified. In perspective images, however,

expected velocities vary markedly from regions close to the focus of expansion to the periphery and the expected smoothness domains depend on the distance of the imaged objects from the observer. Therefore, both parameters would have to be adjusted in a space-variant way. If the environment is a horizontal plane with some obstacles scattered around, this space variance takes a special form. In a sequence of a coordinate transform and the space-invariant motion detection algorithm, the space-variance can completely be included in the coordinate transform which turns out to be the inverse perspective map.

5.3 Inverse perspective in biology

While a wide variety of retinopic mappings has been described in the neurophysiological literature, most functional interpretations available so far apply to just one mapping function, i.e. the complex logarithm. Here, we present a new mapping function, i.e. the optimal coordinate transform for optical flow analysis in the plane, which accounts for a number of neurophysiological findings from the literature:

Epstein (1984) pointed out that the representation of the lower visual field in the cat's striate cortex (area 17) compensates for the perspective distortion introduced to the horizontal plane if the cat fixates a point about one meter in front of it. Johnston (1989) studied the reverse question of what particular spatial surface is represented equal-area after perspective and retinotopic mapping. For the primary visual cortex of various monkeys, he derives cone-shaped surfaces oriented along the line of sight. For a suitable inclination of the line of sight, this cone coincides with the horizontal plane along its vertical meridial line. Finally, in terms of (areal) magnification factors, inverse perspective mapping corresponds to a *visual streak*, since it magnifies the region close to the horizon while the lower (and upper) part of the visual field is compressed. Visual streaks are quite commonly found in many animals living in flat environments (cf. Hughes 1977; Zeil et al. 1989).

6 Conclusion

We propose biologically motivated algorithms for optical flow computation and obstacle detection. The combination of elementary motion detectors and local voting in the optical flow algorithm is a robust and reliable scheme for the detection of both image motion and motion discontinuities. Here, it is combined with a scheme for obstacle detection from optical flow which does not involve any kind of pattern recognition; it relies on the assumption that the movement of the observer is confined to a plane such that anything outside this plane is considered an obstacle. The detection of deviations in the motion field is implemented via a "retinotopic mapping" which transforms the original pattern of image motion into that of a frontoparallel movement (inverse perspective mapping). Besides the application to obstacle avoidance, inverse perspective mapping facilitates the detection of optical flow for both translatory and rotatory egomotion in the plane.

Appendix: the projected image flow is of first order

We prove the proposition made in Sect. 4.2 by computing the complete image flow in the inversely mapped plane. This is done by introducing 3D-surfaces whose motion vectors are then projected and mapped. Consider a plane in 3D space determined by a normal $\mathbf{n}$ and a distance from the origin d. For points in the plane, we have $(\mathbf{s}\cdot\mathbf{n})-d=0$. A point $\mathbf{r}'_H := (x, y, -h)^\top$ in the horizontal plane (i.e., the plane of the inverse perspective map) images the point

$$\mathbf{s}(x, y) = \frac{d}{(\mathbf{r}'_H\cdot\mathbf{n})}\cdot\mathbf{r}'_H \tag{16}$$

if $(\mathbf{r}'_H\cdot\mathbf{n})\neq 0$. (If the inner product vanishes, the plane is either not imaged at $\mathbf{r}'_H$, or, if additionally $d=0$, it views the edge of the planar patch). With this definition, we have:

$$W'_H(x, y) = \mathbf{J}_{\mathcal{Q}\circ\mathcal{P}_I}(\mathbf{s}(x, y))\cdot W(\mathbf{s}(x, y)) \tag{17}$$

The 3D-motion field for egomotion in the horizontal plane is given by

$$W(\mathbf{r}) = -\mathbf{m}+\omega\times\mathbf{r} = \begin{pmatrix} -m_1-\omega r_2 \\ -m_2+\omega r_1 \\ 0 \end{pmatrix}. \tag{18}$$

The latter equality is due to the confinement of the motion to the plane, i.e., $\mathbf{m}=(m_1, m_2, 0)^\top$ and $\omega=(0, 0, \omega)^\top$. We write $\mathbf{s}$ in components s_1, s_2, s_3 and insert the 3D motion field (18) into the mapping and projection rule (17):

$$W'_H(x, y) = h\begin{pmatrix} -1/s_3 & 0 & s_1/s_3^2 \\ 0 & -1/s_3 & s_2/s_3^2 \end{pmatrix}\begin{pmatrix} -m_1-\omega s_2 \\ -m_2+\omega s_1 \\ 0 \end{pmatrix}. \tag{19}$$

Substituting for s_i from (16) and writing the inner product in components, we obtain

$$W'_H(x, y) = \begin{pmatrix} -\frac{m_1}{d}(xn_1+yn_2-hn_3)-\omega y \\ -\frac{m_2}{d}(xn_1+yn_2-hn_3)+\omega x \end{pmatrix}. \tag{20}$$

We introduce the notations

$$W_1 = \begin{pmatrix} -n_1m_1/d & -\omega-n_2m_1/d \\ +\omega-n_1m_2/d & -n_2m_2/d \end{pmatrix}$$

and

$$\mathbf{w}_0 := \frac{hn_3}{d}\begin{pmatrix} m_1 \\ m_2 \end{pmatrix}, \tag{21}$$

and obtain

$$W'_H(x, y) = \mathbf{W}_1\cdot\begin{pmatrix} x \\ y \end{pmatrix}+\mathbf{w}_0. \tag{22}$$

Equation (22) shows that the projected motion field for egomotion in the horizontal plane is of first order. For purely translatory motion ($\omega = 0$), we have from (20):

$$W'_H(x', y') = -\frac{n_1 x' + n_2 y' - n_3 h}{d} \begin{pmatrix} m_1 \\ m_2 \end{pmatrix}, \tag{23}$$

i.e., all vectors of the projected motion field have the same orientation. If additionally $\mathbf{n} = (0, 0, 1)^T$, i.e., the imaged surface is a horizontal plane, the flow field becomes uniform, or zeroth order. This case has already been discussed in Sect. 4.1, (14).

Acknowledgements. This report describes research done in part within the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. Further support for this work was provided by the German Federal Department of Research and Technology (BMFT), Grant. No. ITR8800/K4 and the NATO Scientific Affairs Division.

References

Barlow HB, Levick RW (1965) The mechanism of directional selectivity in the rabbit's retina. J Physiol (London) 173:477–504

Bohrer S, Bülthoff HH, Mallot HA (1990) Motion detection by correlation and voting. In: Eckmiller R, Hartmann G, Hauske G (eds) Parallel processing in neural systems and computers, North-Holland, Amsterdam New York, pp 471–474

Bülthoff HH, Little JJ, Poggio T (1989) A parallel algorithm for real-time computation of optical flow. Nature 337:549

Coxeter HSM (1987) Projective geometry, 2nd edn. Springer, New York Berlin Heidelberg

Duda RO, Hart PE (1973) Pattern classification and scene analysis. Wiley, New York

Epstein LI (1984) An attempt to explain the differences between the upper and lower halves of the striate cortical map in the cat's field of view. Biol Cybern 49:175–177

Gibson JJ (1950) The perception of the visual world. Houghton Mifflin, Boston

Hassenstein B, Reichardt W (1956) Reihenfolgen-Vorzeichenauswertung bei der Bewegungsperzeption des Rüsselkäfers Chlorophanus. Z Naturforsch (B) 11:513–524

Hildreth EC (1984) Computations underlying the measurement of visual motion. Artif Intell 23:309–354

Horn BKP, Schunk BG (1981) Determining optical flow. Artif Intell 17:185–203

Horn BKP, Weldon Jr EL (1988) Direct methods for recovering motion. Int J Comput Vision 2:51–76

Hughes A (1977) The topography of vision in mammals of contrasting life style: comparative optics and retinal organisation. In: Crescitelli F (ed) The visual system of vertebrates. Handbook of Sensory Physiology, VIII/5. Springer, Berlin Heidelberg New York, pp 613–756

Johnston A (1989) The geometry of the topographic map. Vision Res 29:1493–1500

Little JJ, Verri A (1989) Analysis of differential and matching methods for optical flow. In: Proc. Workshop on Visual Motion. IEEE

Little JJ, Bülthoff HH, Poggio T (1988) Parallel optical flow using local voting. In 2. Int Conf Computer Vision (ICCV). IEEE

Longuet-Higgins HC, Prazdny K (1980) The interpretation of a moving retinal image. Proc R Soc London B 208:385–397

Mallot HA, von Seelen W (1989) Why cortices? Neural networks for visual information processing. In: Ewert J-P, Arbib MA (eds) Visuomotor integration: amphibians, comparisons, models, and robots. Plenum Press, New York pp 357–382

Mallot HA, Schulze E, Storjohann K (1989) Neural network strategies for robot navigation. In: Dreyfus G, Personnaz L (eds) Neural networks from models to applications. I.D.S.E.T. Paris, pp 560–569

Mallot HA, von Seelen W, Giannakopoulos F (1990) Neural mapping and space-variant image processing. Neural Networks 3:245–263

Scott GL (1988) Local and global interpretation of moving images. Pitman, London

Seelen W von, Storjohann K, Schulze E, Mallot HA (1988) Verfahren zum Segmentieren dreidimensionaler Szenen. European Patent Application No. 88115757.2, 1988

Verri A, Poggio T (1989) Motion field and optical flow: qualitative properties. IEEE Trans Pattern Anal Machine Intell 11:490–498

Warren WH Jr, Whang S (1987) Visual guidance of walking through apertures: body-scaled information for affordances. J Exp Psychol Human Percept Perf 13:371–383

Waxman AM, Wohn K (1988) Image flow theory: a framework for 3D inference from time-varying imagery. In: Brown C (ed) Advances in computer vision, vol 1. Erlbaum, Hillsdale, NJ pp 165–224

Zeil J, Nalbach G, Nalbach H-O (1989) Spatial vision in a flat world: optical and neural adaptation in arthropods. In Singh RN, Strausfeld NJ (eds) Neurobiology of sensory systems. Plenum Press, New York

Dr. Hanspeter A. Mallot
Institut für Neuroinformatik
Ruhr-Universität ND/04-174
Universitätsstrasse 150
W-4630 Bochum
Federal Republic of Germany