Real-Time Estimation of Heading Direction [*]

Christoph Herwig
FB 3 Informatik
Zentrum für Kognitionswissenschaften
Universität Bremen
Postfach 330440
28334 Bremen, Germany
herwig@informatik.uni-bremen.de

Hans-Otto Carmesin
FB 1 Physik
Zentrum für Kognitionswissenschaften
Universität Bremen
Postfach 330440
28334 Bremen, Germany
carmesin@physik.uni-bremen.de

Fred Hamker
FG Neuroinformatik
Fakultät für Informatik und Automatisierung
TU Ilmenau
Postfach 327
98684 Ilmenau, Germany
fred@informatik.tu-ilmenau.de

Dirk Wandtke
FG Neuroinformatik
Fakultät für Informatik und Automatisierung
TU Ilmenau
Postfach 327
98684 Ilmenau, Germany

September 30, 1995

Running Title: Heading Direction Estimation

Abstract

This research demonstrates the usefulness of visual flow field information for the robotic navigation task of heading direction estimation. Egomotion parameters are directly computed on the basis of normalized normal flow fields derived from image sequences. Advantages over the traditional half-plane approach constitute the computation of unique minima and the applicability over a larger motion state space. At the same time the method preserves the desired properties of *robustness* towards statistical and systematic errors as well as *real-time* processing capability. The performance is evaluated with synthetic as well as real image sequences.

[*]

1 Introduction

The perception and control of motion has long been an exciting interdisciplinary field of research. It draws on many fields including psychology, biology, computer science and engineering [1]. The perception of motion may be separated into three categories: First, the movement of objects relative to the stationary observer, second, the movement of the observer relative to the stationary scene, and third, the independent motion of the observer relative to the moving objects in the scene. Clearly, the third category is the most general one and its solution would include the first two categories as special cases. The ambitious goal pursued during the 1980s was to develop such a general purpose motion module. However, the unsatisfactory performance of a plethora of algorithms has led some researchers to rethink the role of motion vision.

It has been postulated that vision should be looked at from a *purposive* point of view [2, 3, 4]. This branch of motion research develops a repertoire of visual algorithms that enables the computation of *partial* information in a simple way from *partial* measurements. People sometimes criticize such algorithms because they are ”incomplete” and ”qualitative”. Some robotic engineers however have come to prefer simple but *fast* and *working* systems. Not seldom they are influenced by biological navigation experts such as the bee and the fly [5, 6]. It has been postulated that biological vision systems utilize *different* strategies for the mentioned different motion categories also related to different tasks currently of interest to the insect [7, 8]. In context of active computer vision, Fermüller also distinguished between the first two separate categories by giving two algorithmic strategies [9] rather than viewing them in the global context of the third category.

Here we are interested in a fast and robust solution to the second situation, especially for the application in real-time robotic tasks. We develop a model for the recovery of the heading direction of a camera in motion that is important in mobile robot applications and hand-eye coordination tasks of stationary robots. We distinguish between 1. the robot heading direction (RHD) estimation and 2. the camera heading direction (CHD) estimation. No doubt: the RHD estimation is important. On one hand it may be more or less precisely available via the encoder readings of the robot motors. On the other hand if the CHD is known we may transform this to the robot coordinate system to yield the RHD (more or less precisely). What is the motivation to estimate the CHD first? Besides the possible subsequent estimation of the RHD, the CHD estimation is justified by serving at least two further purposes: the estimation of relative depth (structure, distance, time-to-contact) and the independent motion detection (segmentation and estimation). Both applications may build upon the CHD computation. The exclusive processing in the *camera* measurement space is highly preferred since otherwise erroneous transformations (camera calibration) become necessary. The CHD estimation can therefore be viewed as only an intermediate step towards important tasks related to the survival of the systems, i.e. breaking in time or obstacle avoidance.

We derive the dynamic information via visual flow field analysis. We allow the camera motion to be purely translational. The scene independent rotational motion may be available via gyro sensors. We follow a qualitative approach in that we only use the normalized normal flow vectors as input to the estimation process and avoid the intermediate optical flow field estimation.

The paper is organized as follows: Section 2 reviews relevant equations for motion processing. In section 3 we develop the cost function. The behavior of the algorithm in the context of several error cases is investigated in section 4. Section 5 provides insight into the performance of the algorithm in real imagery situations and we conclude with a summary and future work in section 6.

2 Review on motion processing

The goal of this module is to provide estimates on egomotion (direction of translation) in the case of a stationary scene. We first look at the geometry and then show how the flow fields are being derived from the images.

2.1 Differential geometry

When objects move in front of the camera or the camera moves through the environment, then corresponding changes are induced in the images. A *relative* motion between camera and object motion is registered in the images. According to a classical result in the kinematic theory every rigid body motion may be decomposed into six components (Fig. 1).

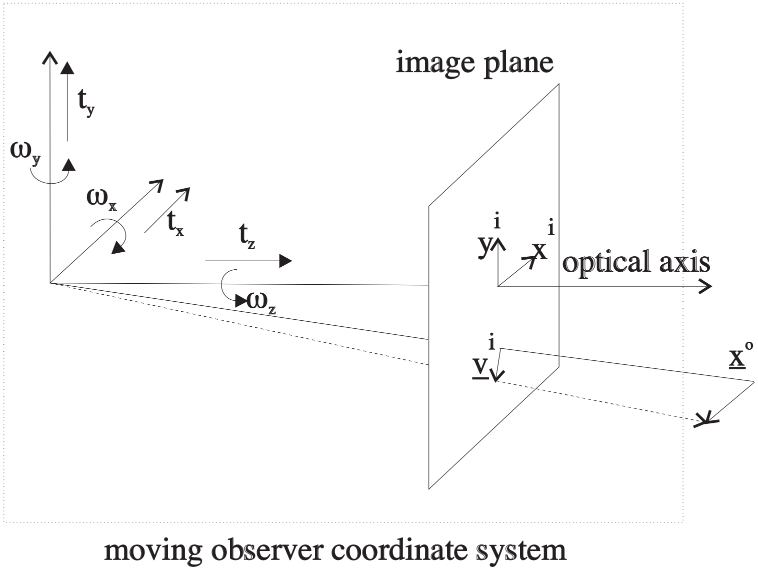


Figure 1: Imaging geometry of the object point ($\underline{x}^o$) – image point ($\underline{x}^i$) as well as of the object velocity ($\underline{v}^o$) – image velocity ($\underline{v}^i$) transformation; the motion parameters $\underline{t}$ and $\underline{\omega}$ are indicated. Optical axis - image plane intersection point $= (0, 0, f)^T$.

The 2D motion field on the image plane due to the 3D relative motion between the observer and the scene separated into translation and rotation terms is given by the *motion equations* [10]:

$$\begin{array}{rcl} v_x^i & = & \left(\frac{-f t_x + x^i t_z}{z^o}\right) + \left(\frac{x^i y^i}{f}\omega_x - \left(f + \frac{(x^i)^2}{f}\right)\omega_y + y^i\omega_z\right) \\ v_y^i & = & \left(\frac{-f t_y + y^i t_z}{z^o}\right) + \left(\left(f + \frac{(y^i)^2}{f}\right)\omega_x - \frac{x^i y^i}{f}\omega_y - x^i\omega_z\right) \quad , \end{array} \tag{1}$$

or in short form

$$v_x^i = v_x^{i,t} + v_x^{i,\omega}, \quad v_y^i = v_y^{i,t} + v_y^{i,\omega} \quad \text{that is} \quad \underline{v}^i = \underline{v}^{i,t} + \underline{v}^{i,\omega}. \tag{2}$$

During flow field analysis, our system operates in a *pure translational* mode, i.e., $\underline{\omega} = (0, 0, 0)^T$. The equations then simplify to:

$$\boxed{\begin{array}{rcccl} v_x^i & = & \left(\frac{-f t_x + x^i t_z}{z^o}\right) & = & (x^i - x^f)\frac{t_z}{z^o} \\ v_y^i & = & \left(\frac{-f t_y + y^i t_z}{z^o}\right) & = & (y^i - y^f)\frac{t_z}{z^o} \end{array}} \tag{3}$$

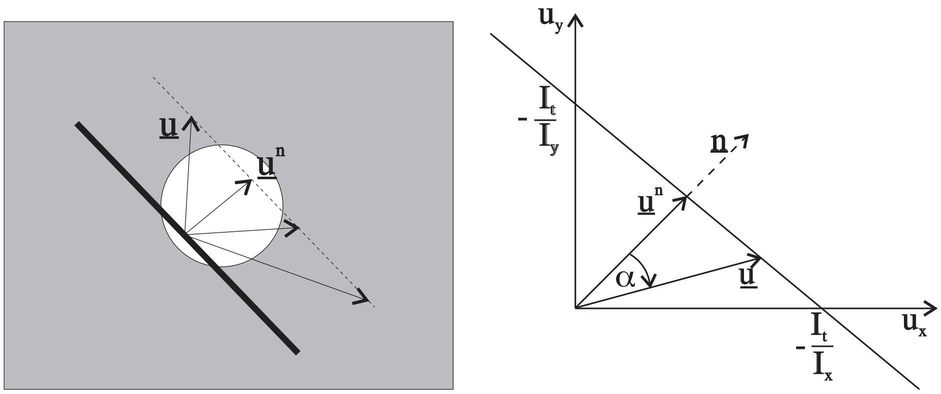
with $(x^f, y^f)^T$ as the egomotion parameters to be estimated by this module. They describe the direction of translation, i.e., the intersection of the vector originating in the camera nodal point with the infinite plane containing the image plane. In the pure translation case, all motion field vectors originate at this location [11] and it is commonly called the *focus of expansion* (FOE).

2.2 Normal flow fields

Differential methods of flow field computation in general provide qualitative measures which can be exploited for robot navigation tasks. Only the speed perpendicular to a moving edge, i.e., the normal component $\underline{u}^n$ can be recovered exactly. With $\underline{n} = \frac{\nabla I}{||\nabla I||}$, the projection of $\underline{u}$ onto the gradient direction $\underline{n}$ may be expressed in terms of the spatiotemporal derivatives

$$\underline{u}^n = (\underline{u} \cdot \underline{n}) \cdot \underline{n} = -\frac{I_t}{I_x^2 + I_y^2} \begin{pmatrix} I_x \\ I_y \end{pmatrix} \tag{4}$$

with a magnitude of $||\underline{u}^n|| = \frac{||I_t||}{\sqrt{I_x^2+I_y^2}}$. It is the *normal flow field*, i.e., the optical flow [11] projected onto the direction of the intensity gradient. This field can be quickly computed and we will therefore base our analysis on the use of this normal flow field. It approximates the geometrically given normal *motion* field for the case of large spatial intensity gradients [12]:



(a) Only the perpendicular component $\underline{n}^u$ of a translating local edge can be recovered. The knowledge about the true motion $\underline{u}$ is lost.

(b) The problem in (u_x, u_y)–space. The true motion is constrained to lie along a line. The normal flow component $\underline{u}^n$ has an associated magnitude along the gradient direction.

Figure 2: The aperture problem.

$$||\underline{v}_i^n|| - ||\underline{u}^n|| = \frac{1}{||\nabla I||}\frac{dI}{dt} = 0 \quad . \tag{5}$$

Note that the formulation of the normal flow field originates in the constancy of object intensity. This so-called "brightness constancy" (BC) is typically violated for non-lambertian surfaces, apparent motion situations due to reflecting surfaces and camera gain variations between successive frames.

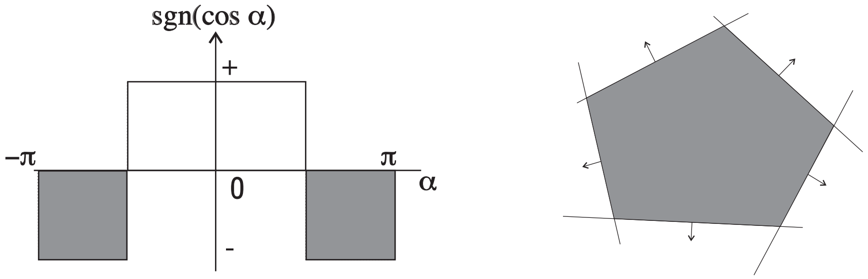
3 Cost function

3.1 General formulation

The starting point are Eqns.(3). One may consider a two–dimensional egomotion state $\epsilon := (x^f, y^f)$. This ϵ along with the scalar field $\zeta := \frac{t_z}{z^o(x^i,y^i)}$ (scaled depth field) yields a vector field (motion field) $\underline{v}^i = (v^i_x, v^i_y)^T$ on the (x^i, y^i) image plane (Fig. 1); that is, (ϵ, ζ) is mapped to $\underline{v}^i$ with Eqns.(3).

The inverse problem is treated, namely, to find (ϵ, ζ) from $\underline{v}^i$. It is specifically explicated as follows: First, the normal flow field $\underline{u}^n$ is estimated from the spatio–temporal derivatives (I_x, I_y, I_t) of the intensity image $I(x^i, y^i, t)$ (see Eq. (4)). Its normalized version is denoted by $\underline{n}^u$. Second, a necessary condition for a good $\underline{v}^i(\epsilon, \zeta)$ is that it has a *positive* scalar product $(\underline{v}^i(\epsilon, \zeta) \cdot \underline{n}^u)$ (half plane constraint).

Traditionally, *voting* mechanisms have been proposed for solving for the FOE by intersecting the half planes corresponding to various confident normal flow vectors (Fig. 3) [12]. It has been argued that this can be done in real-time.



(a) Step function depending on optical flow vector deviation angle α. Dark areas indicate angles for vectors pointing into the half plane containing the FOE.

(b) Intersection of half planes corresponding to selected normal flow vectors. Dark area indicates the region containing the FOE.

Figure 3: Utilizing the half plane constraint.

In the following, we propose an advantageous *convergent* algorithm for computing the FOE. In order to motivate our approach we argue that the positive scalar product can be used for a heuristic search procedure as follows:

```
1. Pick prospective $(x^f, y^f)^T$ location.
2. ∀ image points $(x^i, y^i)$
   if $(\underline{v}^i(\underline{\epsilon}, \zeta) \cdot \underline{n}^u) < 0$  Error=Error+1
   else Success=Success+1
3. Evaluate the FOE choice on the basis of Error
4. Goto 1 or Exit
```

As an improvement, the positive scalar product can also be formalized with the angle α between $\underline{v}^i(\epsilon, \zeta)$ and $\underline{n}^u$, the cosine and signum functions as follows

$$\operatorname{sgn}(\cos\alpha) > 0, \tag{6}$$

As opposed to the discrete matching we propose to use the continuous criterion

$$\cos\alpha \text{ is } \text{ maximal.} \tag{7}$$

This enables us to convert the undesired search into a minimization task of a cost function. This is equivalent to

$$\frac{1}{2}(1-\cos\alpha) \text{ is } \text{ minimal } \text{ and } \text{ positive.} \tag{8}$$

and by multiplying with $||\underline{v}^i||$

$$\frac{1}{2}\left(||\underline{v}^i|| - ||\underline{v}^i||\cos\alpha\right) \text{ is } \text{ minimal } \text{ and } \text{ positive.} \tag{9}$$

The term in the above equation is expressed using a scalar product

$$\frac{1}{2}\left(||\underline{v}^i(\epsilon,\zeta)|| - \underline{v}^i(\epsilon,\zeta)\cdot\underline{n}^u)\right) \ , \tag{10}$$

i.e. the projection of the optical flow direction onto the normalized normal flow vector. The above function of (ϵ,ζ) is denoted as the elementary cost measure for a pixel location $\underline{x}^i$

$$C^e(\epsilon,\zeta,\underline{x}^i) = \frac{1}{2}\left(||\underline{v}^i(\epsilon,\zeta)|| - (\underline{v}^i(\epsilon,\zeta)\cdot\underline{n}^u)\right). \tag{11}$$

In order to obtain a global estimation of (ϵ,ζ), we first propose to minimize the sum of the elementary cost measures

$$C'(\epsilon,\zeta) = \sum_{\underline{x}^i} C^e(\epsilon,\zeta,\underline{x}^i). \tag{12}$$

3.2 Normalization

The costs C^e contribute to C' depending on the angle between the motion field vector $\underline{v}^i$ and its projection onto the normal flow direction $\underline{n}^u$. More precisely, the vector length (magnitude) difference however is computed. Besides the desired dependence on the angle α, the magnitude difference involves a dependence on the *absolute* vector magnitude: for the same angle, higher magnitudes contribute a higher portion to the error measure. In order to eliminate this property, we propose the *normalized version* of C' as the final cost C.

$$C(\epsilon,\zeta) = \frac{1}{2}\sum_{\underline{x}^i}\left(\frac{1}{||\underline{v}^i(\epsilon,\zeta)||}C^e(\epsilon,\zeta,\underline{x}^i)\right). \tag{13}$$

3.3 Detailed formulation

From Eq. (11) and Eq. (13) we obtain

$$C(\epsilon,\zeta,\underline{x}^i) = \frac{1}{2}\sum_{\underline{x}^i}\left(1 - \frac{(\underline{v}^i)\cdot\underline{n}^u)}{||\underline{v}^i||}\right). \tag{14}$$

Next we express $\underline{v}^i$ in further detail (Eq. (3)) as

$$C(\epsilon,\zeta,\underline{x}^i) = \frac{1}{2}\sum_{\underline{x}^i}\left(1 - \frac{\zeta\cdot(\underline{\xi}\cdot\underline{n}^u)}{\zeta\cdot||\underline{\xi}||}\right) \tag{15}$$

$$\boxed{C(\epsilon,\underline{x}^i) = \tfrac{1}{2}\sum_{\underline{x}^i}\left(1 - \tfrac{(\underline{\xi}\cdot\underline{n}^u)}{||\underline{\xi}||}\right)} \tag{16}$$

with $\underline{\xi} = (x^i - x^f, y^i - y^f)^T$, i.e. a linear function of the FOE variables x^f and y^f. A side effect of the normalization is the achievement of environmental invariance (independence of depth).

4 Error analysis

4.1 Error measure

For the evaluation of the developed algorithm a suitable error measure was needed. Although the FOE constitutes a location on the image plane it seems appropriate to measure the deviation *angle* between true and estimated translation direction. This angle δ is given by

$$\begin{aligned}\delta &= \angle(\underline{\hat{x}}^f, \underline{x}^f) \\ &= \angle\left(\begin{pmatrix}\hat{x}^f \\ \hat{y}^f \\ f\end{pmatrix}, \begin{pmatrix}x^f \\ y^f \\ f\end{pmatrix}\right) \\ &= \text{arc } \cos\frac{\underline{\hat{x}}^f \cdot \underline{x}^f}{||\underline{\hat{x}}^f||||\underline{x}^f||}\end{aligned} \tag{17}$$

4.2 Cost function evaluation

Here we present experiments with a `Simon` [13] simulator image sequence in order to illustrate the quality of the cost functions. The synthetic sequence is computed by moving the camera towards a static scene comprised of a single planar object. In order to introduce environmental randomness the intensity distribution is chosen to be a superposition of gaussians with peaks at random locations on the object plane. Intensity derivatives are computed analytically.

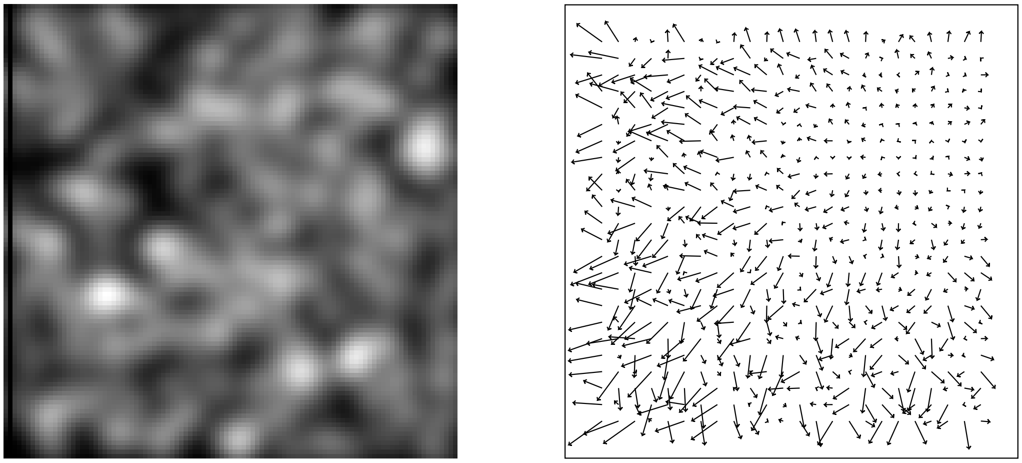
In the sequence considered here (Fig. 4a) the camera moves in direction of the center of the first quadrant, i.e, to the top right. The FOE is at location $\underline{x}^f = (-25, 25)^T$, where $(0,0)^T$ denotes the image center, x (y) is counted positive leftwards (upwards) and the image is of size $(100, 100)$. The normal flow field (Fig. 4b) indicates a broad representation of vector orientations. The dependence of flow vector magnitude on the distance from the FOE is clearly visible. The cost plots $C(x^f, y^f)$ (Fig. 5) indicate the the FOE as the location with minimum error. In this case it is found to within 1 pixel from the true location. The error surface does not introduce any local minima, where optimization algorithms could get stuck. For further clarification we plot the sign of the x^f- and y^f- derivative of the error function in Fig. 6. We identify a single line separating the positive and negative monotonic regions in both derivative images. We can locate the FOE where the superposed lines of both images intersect (Fig. 6c).

4.3 Statistical noise

The first experiment evaluates the sensitivity of the algorithm on *statistical* noise. Gaussian noise of varying degree of variance and zero mean was added to the intensity derivatives I_x, I_y and I_t. This includes the various sources of statistical noise up to this processing stage, e.g. intensity quantization errors or errors in discrete spatiotemporal derivative computation. The signal to noise ratio is computed to be

$$\frac{S}{N} = \log_{10}\frac{\sum_{image\ pixels}(|I_x| + |I_y| + |I_t|)}{\sum_{image\ pixels}\left(|I_x^n| + |I_y^n| + |I_t^n|\right)} \tag{18}$$

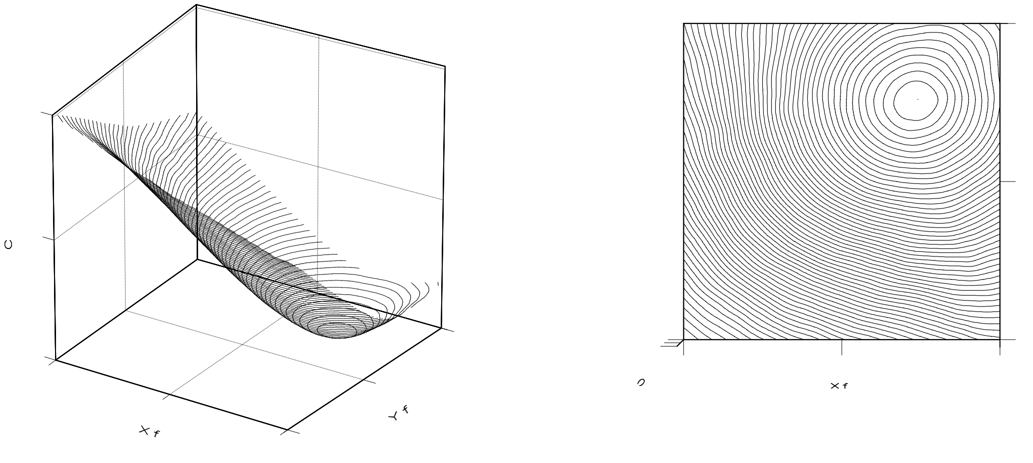
with I_x^n,I_y^n,I_t^n as the additive gaussian noise components on the derivatives. Fig.7 presents the results. We observe a breakpoint at $\log_{10}\frac{S}{N} = 0$ or $\frac{S}{N} = 1$. Hence, additive noise up to an amount equivalent to the signal component is tolerable for error angles in heading direction of $\delta = 0.1$, i.e. about 6 degrees.



(a) Camera view (Resolution 100x100).

(b) Normal flow field (subsampled by 4).

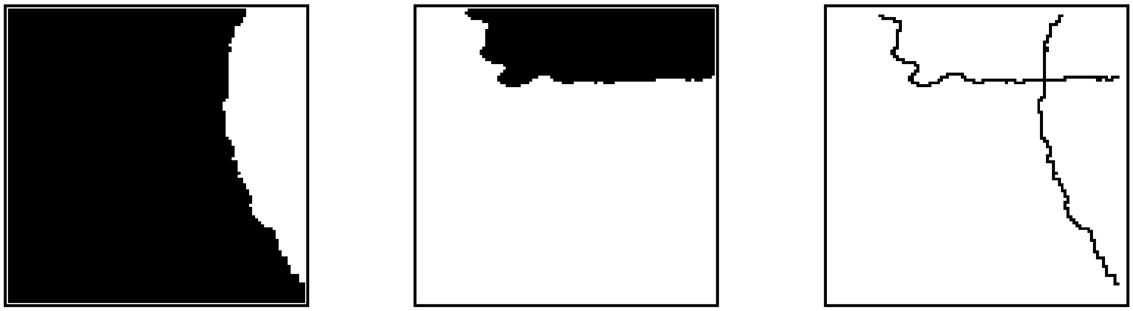
Figure 4: The synthetic test sequence.



(a) 3D cost function contour plot $C(x^f, y^f)$.

(b) 2D cost function contour plot $C(x^f, y^f)$.

Figure 5: The cost function surfaces. The cost function minimum is clearly visible and can be exactly retrieved by gradient descent techniques.



(a) Sign of the cost function x-derivative.

(b) Sign of the cost function y-derivative.

(c) Intersection indicates the FOE location.

Figure 6: Sign of the cost function derivatives.

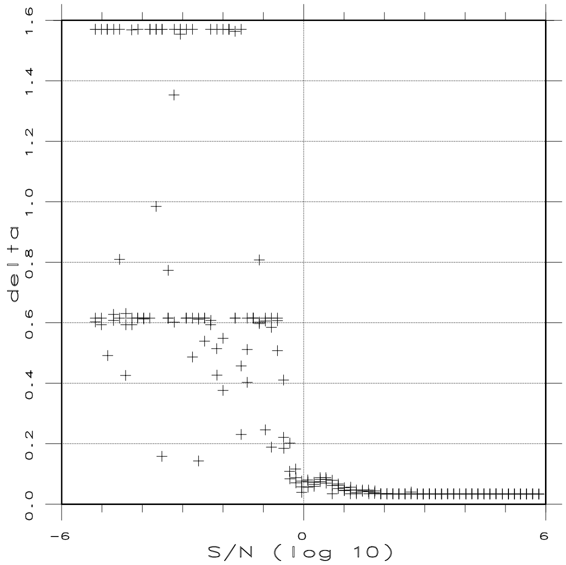


Figure 7: Error in translation direction estimation δ as a function of the signal to noise ratio $\frac{S}{N}$ on the spatiotemporal intensity derivatives.

4.4 Direction of translation

This experiment checks the performance for different heading directions. Possible directions on the half sphere corresponding to positive z values are parameterized in spherical coordinates (Fig.8) by the *elevation angle* α bounded by $[\frac{-\pi}{2}, \frac{\pi}{2}]$ counting from the optical axis z the *azimuth angle* β describing the rotation around the optical axis bounded by $[\frac{-\pi}{2}, \frac{\pi}{2}]$ counting from the y-axis and the *translation magnitude* $||\underline{t}||$. The varying egomotion parameters are expressed in cartesian coordinates as

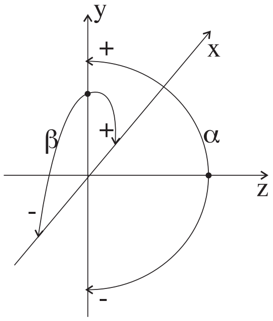


Figure 8: Convention for the spherical coordinate system angles (the poles lie on the z-axis): elevation angle α rotates around the (-x)-axis with origin on the z-axis. The azimuth angle β rotates around the (-z)-axis with origin on the y-axis.

$$\begin{aligned} t_x &= ||\underline{t}|| \sin\alpha \sin\beta \\ t_y &= ||\underline{t}|| \sin\alpha \cos\beta \\ t_z &= ||\underline{t}|| \cos\alpha \end{aligned} \tag{19}$$

For some $||\underline{t}||$ and $\frac{-\pi}{3} < \alpha < \frac{\pi}{3}$ and $\frac{-\pi}{2} < \beta < \frac{\pi}{2}$ the δ error is plotted in Fig.9a. The high frequency content is due to the specific underlying sequence. Clearly visible are breakdown locations on the infinite image plane for FOE locations approximately $\frac{1}{2}$ the image size outside the image plane.

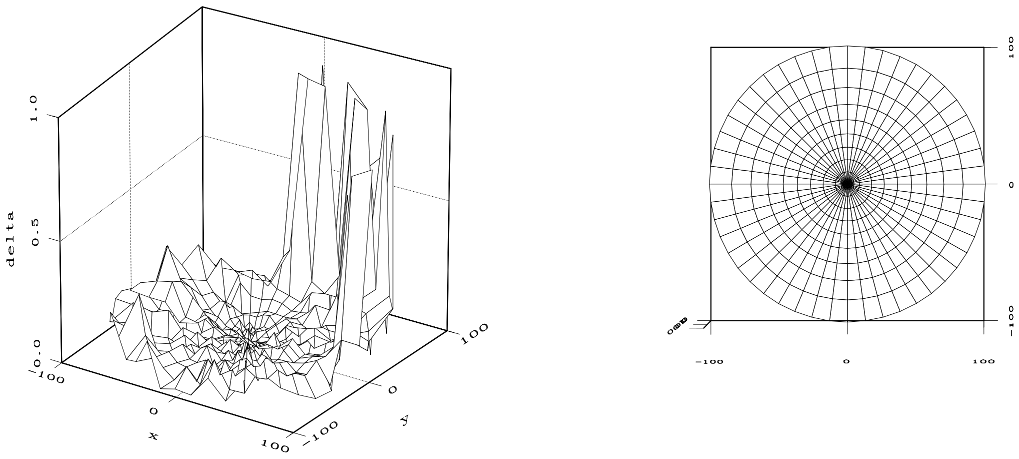
This result demonstrates a major improvement over the heuristic search procedure introduced in section 3.1. Whereas conventional voting algorithms *fail* for FOEs outside the image region, this algorithm reaches its breakpoint further outside the image. It also relates to findings by Waxman [14] describing the benefits of large field of views.

4.5 Rotational noise

Although assuming translational motion in order to gain maximal information we expect remaining rotational components. Aloimonos and Duric [15] investigated the effect of this *systematic* noise on the extend of the voting region S due to rotational motion components. They have shown that it depends on the angle between the direction of translation and the axis of rotation as well as rotation-to-translation ratio. Here we provide the corresponding results for our algorithm. Two experiments seem suitable.

4.5.1 Rotation axis

This experiment investigates the influence of the location of the rotation axis on the FOE estimation. Fig. 10 plots the error for varying directions of the rotation axis $\left(\frac{\omega_x}{\omega_z}, \frac{\omega_y}{\omega_z}\right)^T$ on the half sphere. As the rotation axis deviates more and more from the translation axis, the



(a) Error surface plotted for a 200x200 support region. The center 100x100 region corresponds to the image plane.

(b) Distribution of measurements: homogeneous sampling on the half sphere corresponding to positive z values projected onto the image plane.

Figure 9: Error in translation direction estimation δ as a function of the FOE location $(x^f, y^f)^T$.

vector magnitudes, i.e. the heading direction estimation errors, increase. Most important, *no singularity* can be found in this smoothly varying vector field.

4.5.2 Rotation magnitude

Here, the influence of the rotation magnitude is tested. Fig. 11 illustrates the error with varying magnitude due to a certain rotation. For small rotation components, we observe a linear increase in estimation error. For high rotation magnitudes the error converges towards the maximum error, i.e. $\frac{\pi}{2}$ or 90 degrees.

5 Performance evaluation

5.1 Real imagery

Various real sequence evaluations have been completed to demonstrate the performance of the algorithm. As examples we chose to use the benchmark NASA-Ames sequence (Fig.13a) [1] as well a sequence recorded at Ilmenau using the mobile robot MILVA (Fig.12). The robot (0.58m height, 0.69m width and 1m long) is an autonomous mobile platform with air tires. During translational motion images are digitized and transferred to the on-board computer. For our detailed investigations the images were downloaded to a host computer via Ethernet. For the MILVA sequence (Fig. 14a). one image was shot every centimeter of robot motion. The

[1] This calibrated motion sequence was made public at the 1991 IEEE Workshop on Visual Motion.

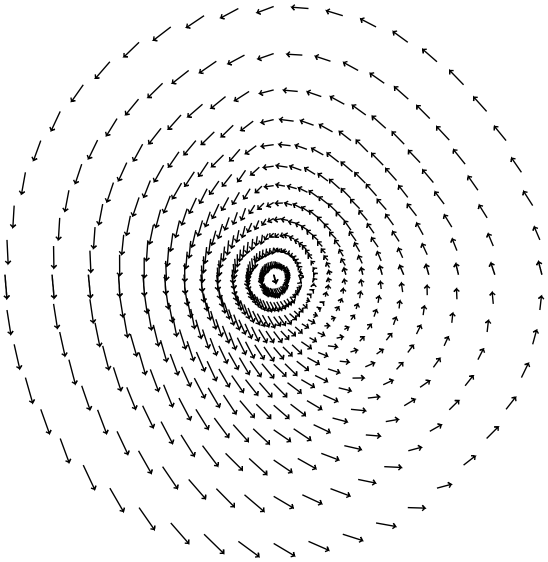


Figure 10: Vector field description of error in translation direction estimation δ as a function of the location of the rotation axis intersecting the image plane. The vector origins specify these intersection points. The vectors themselves indicate the offset in magnitude and direction between the true FOE location and the estimated FOE location. The examples here is given for $||\underline{\omega}|| = 0.01$ and an FOE at $(0,0)^T$.

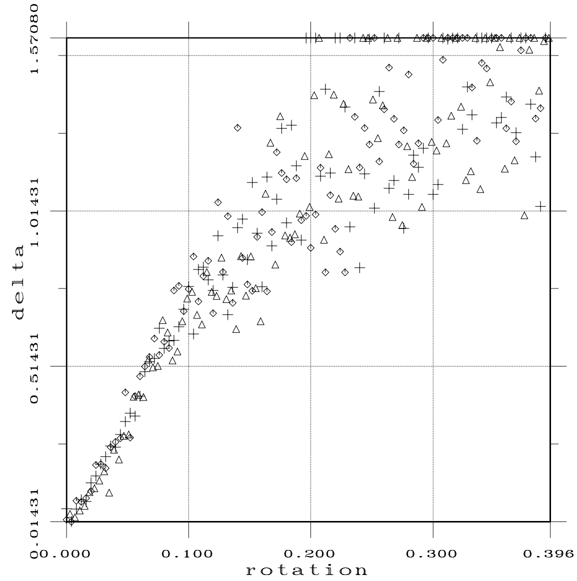


Figure 11: Error in translation direction estimation δ as a function of absolute value of rotation $||\underline{\omega}||$. Three runs for different scenes are distinguished by different markers. Examples for $\underline{t} = (0,0,100)^T$, $\alpha = 0.7$ and $\beta = -0.8$.

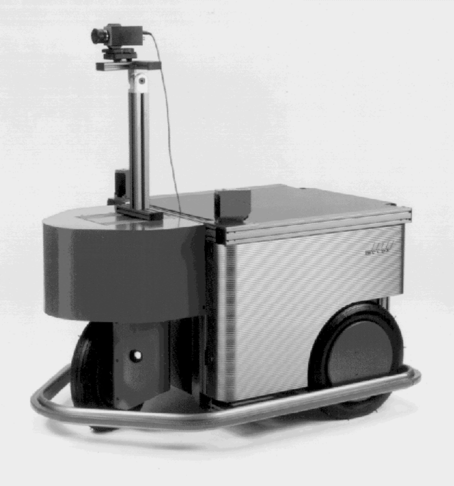


Figure 12: The mobile robot MILVA.

spatiotemporal derivatives are computed by binomial filters. The normal flow fields were checked for high confidence, i.e. very low and very high magnitude entries were marked non-confident.

5.2 Cost functions

5.2.1 Shape of the error function within image region

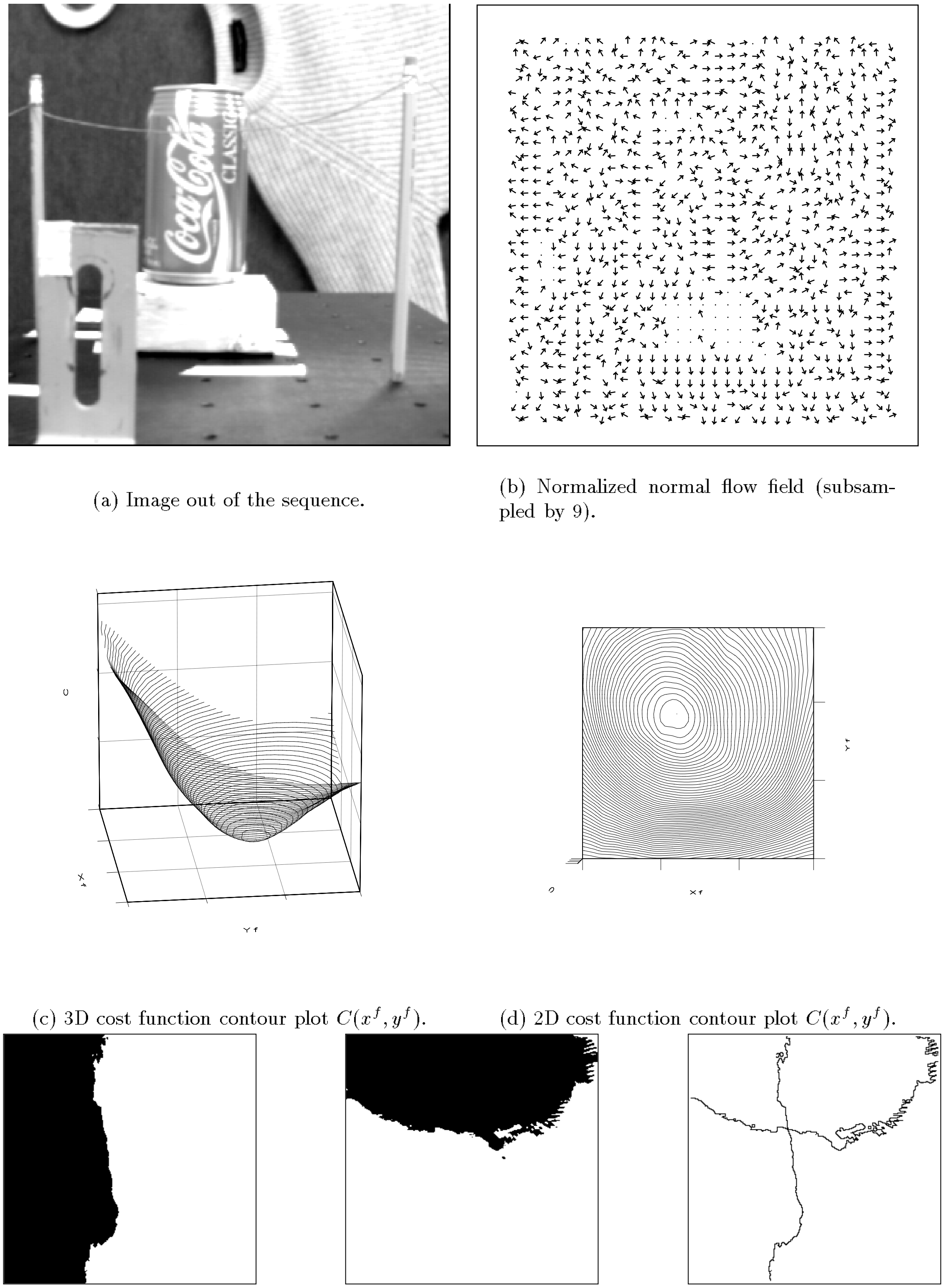
Of particular interest is the *shape* of the error function for real imagery cases. We demonstrate results for the two sequences: the benchmark NASA-Ames sequence (Fig.13) and the MILVA sequence (Fig. 14). In the $(x^f, y^f)^T$ - plane, $C(\epsilon, \underline{x}^i)$ is positive, has a single minimum and is monotonically increasing with distance from the FOE. Accordingly, we determine the focus of expansion $(\hat{x}^f, \hat{y}^f)^T$ with minimal cost C by gradient descent (see section 5.3).

5.2.2 Cost function in larger areas

This examination demonstrates the shape of the error function also in areas outside the image support region. We used an area of 9x9 images (Fig. 15). Overall we find that the shape of the function is considerably influenced by the FOE's location, the textural information and the motion state. The error function rises symmetrically in all directions if the FOE lies at $(0, 0)$. For the common case of off-centered translational motion, the available normal flow vector directions are biased towards one direction. This results in a shape distortion with a shape symmetry axis through $(0, 0)^T$ and $(x^f, y^f)^T$. In all of our tested real sequences we observed only a single minimum.

5.3 Cost minimization by gradient descent

In principle, different methods are suited for discovering the global minimum. It proved best to use the Polak Ribiere conjugate gradient descent method [16, ch.10]. In this section we provide insight into the performance of this minimization technique.



(a) Image out of the sequence.

(b) Normalized normal flow field (subsampled by 9).

(c) 3D cost function contour plot $C(x^f, y^f)$.

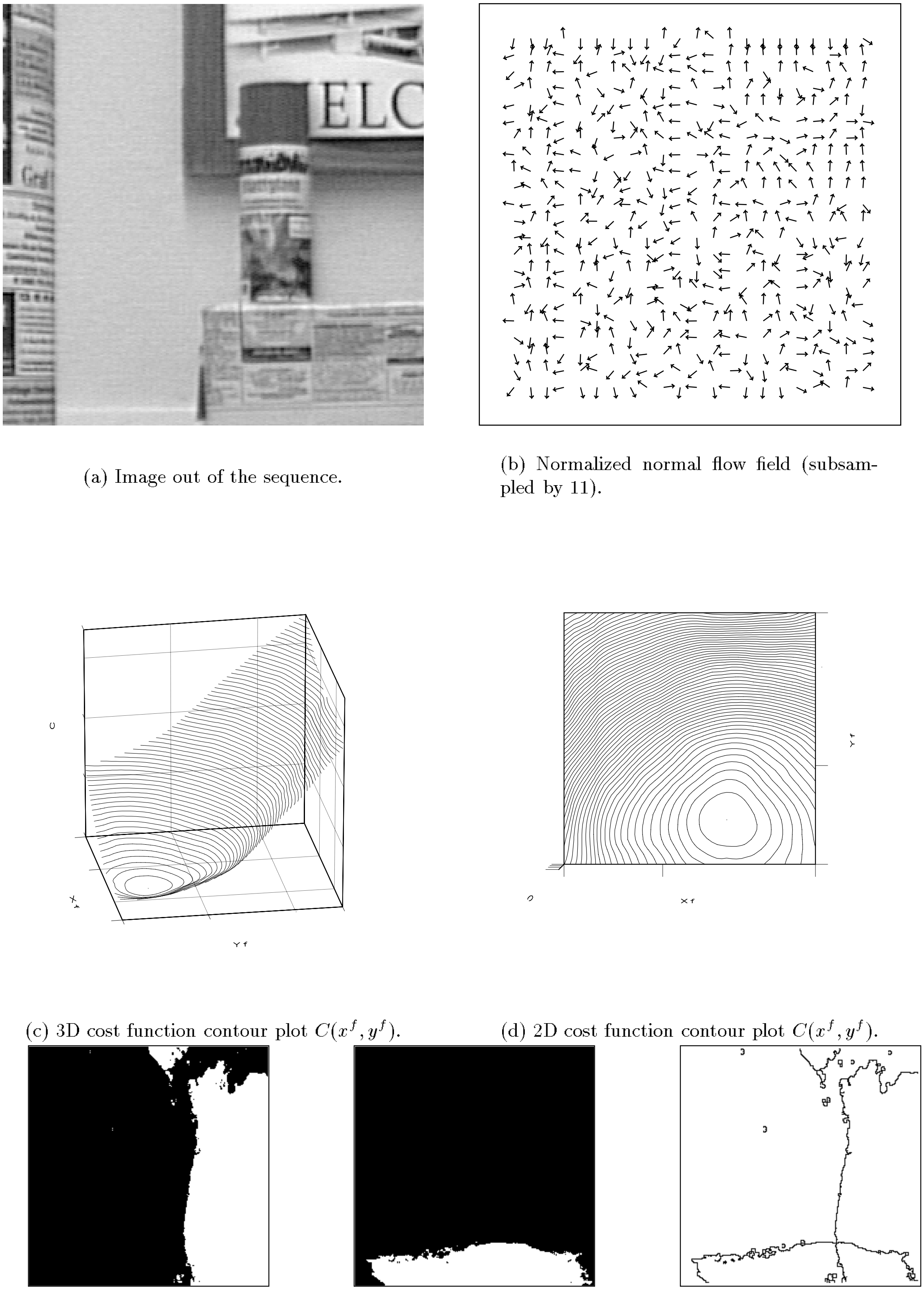
(d) 2D cost function contour plot $C(x^f, y^f)$.

(e) Sign of cost function x-derivative.

(f) Sign of cost function y-derivative.

(g) Intersection indicates the FOE location.

Figure 13: The NASA-Ames real sequence.



(a) Image out of the sequence.

(b) Normalized normal flow field (subsampled by 11).

(c) 3D cost function contour plot $C(x^f, y^f)$.

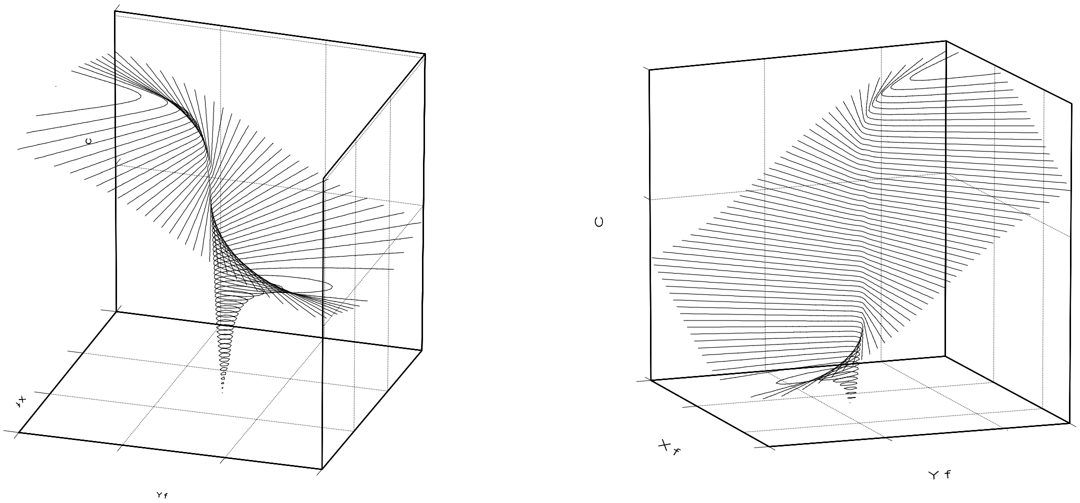
(d) 2D cost function contour plot $C(x^f, y^f)$.

(e) Sign of cost function x-derivative.

(f) Sign of cost function y-derivative.

(g) Intersection indicates the FOE location.

Figure 14: The MILVA real sequence.



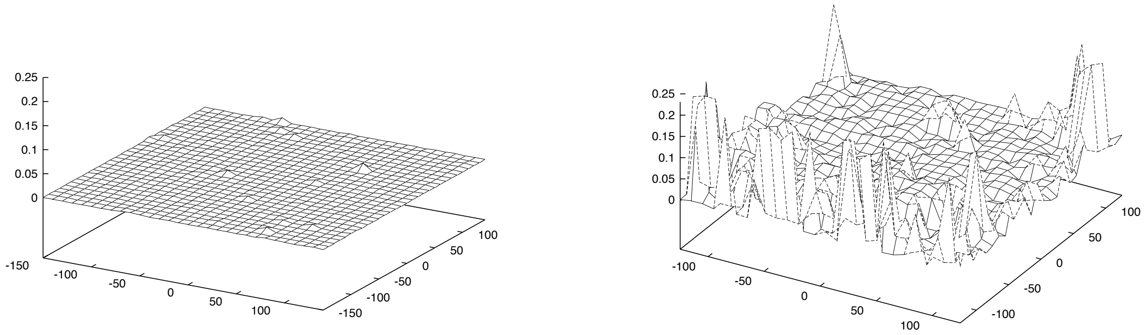
(a) 3D cost function contour plot $C(x^f, y^f)$ of the NASA sequence.

(b) 3D cost function contour plot $C(x^f, y^f)$ of the MILVA sequence.

Figure 15: Extended cost function surfaces: The cost was computed for FOE locations $(-4.5N^{cols} < x^f < 4.5N^{cols}, -4.5N^{rows} < y^f < 4.5N^{rows})$ with N^{cols} as the number of columns in the image, N^{rows} as the number of rows in the image and with the origin at the image center. Hence, a 9x9 image region was computed for prospective FOE locations. The cost functions still are monotonically decreasing. The global minima are clearly visible and can be exactly retrieved by gradient descent techniques.

5.3.1 Dependence on starting point

Does the starting point (gradient descent initialization) influence the performance? One image out of every sequence is selected and every 10th pixel location is used as a starting point for gradient descent. In Fig. 16 the angle between the estimated FOE of the gradient descent and the minimum of the error function (compare Eq. (17)) is presented. For the NASA sequence no false estimations occur. For the MILVA sequence the algorithm has difficulties at some starting points in long distances from the minimum despite the fact that only one minimum exists. Overall, the success is based on the shape of the error function and the termination criterion. From the shape of the cost functions (see section 5.2.2) and the results here we conclude that it is best to commence the gradient descent at $(0, 0)^T$.



(a) Angle for the NASA sequence.

(b) Angle for the MILVA sequence.

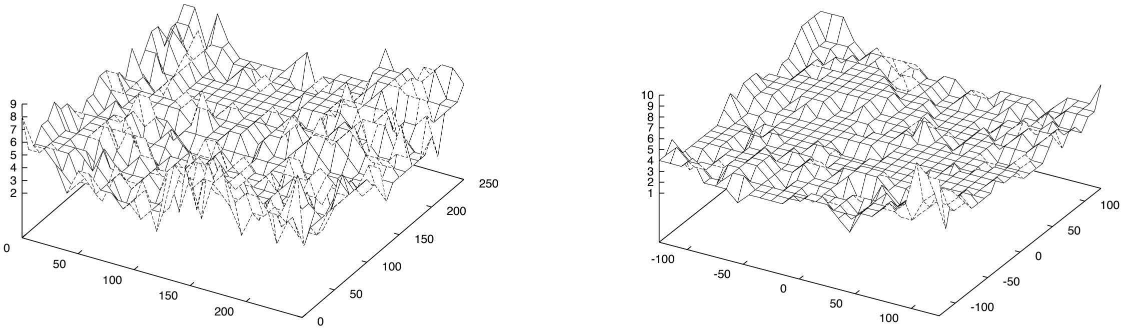
Figure 16: Angle between the estimated FOE of the gradient descent and the minimum of the error function for different starting points.

5.3.2 Number of iterations

Fig. 17 shows the number of iterations required. The absolute value depends on the value of the termination criterion. It is therefore not of too much interest. Nevertheless, the iteration count is very low overall. Interesting is the distribution of the iteration count. It remains almost constant for the varying starting points considered.

5.4 Tracking FOE through time

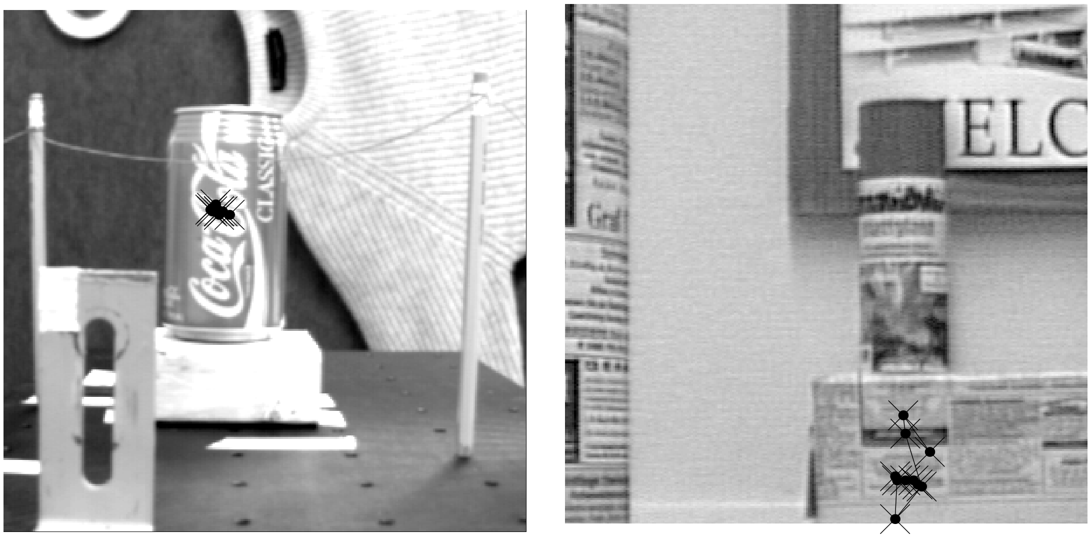
The tracking of the FOE over time indicates the robustness of our approach. We calculated the error functions of 10 successive images and plotted the minima onto an image out of each sequence (Fig. 18). Because of our real image shooting conditions in a real environment, all potential error influences accumulate. Nevertheless the MILVA sequence displays an almost constant FOE for the sequence. For the NASA-Ames sequence the true FOE is known to be at (25, 32). The gradient descent determined the minimum around (26, 34) hence a deviation of about 2 pixels for the complete sequence.



(a) Iterations for the NASA sequence.

(b) Iterations for the MILVA sequence.

Figure 17: Number of required iterations for different starting points.



(a) Image of the NASA sequence.

(b) Image of the original MILVA sequence.

Figure 18: Tracking the FOE through time in real sequences.

6 Conclusion

We have presented a robust and fast way to compute the camera heading direction. The algorithm was extensively tested with synthetic as well as with real image sequences. The main results are:

- The developed cost function exhibits only single minima in all sequences tested. The function is of monotonic shape for large viewing angles.
- Statistical noise equal to the signal value is tolerable for the achievement of a few degrees of arc accuracy in heading direction estimation. Cutting [17] showed that humans require an accuracy of about 1^o of visual arc for the judgement of heading in order to avoid obstacles successfully while running and driving.
- Heading directions with FOE's outside the image region can be estimated reliably up until a breakpoint location.
- Rotational noise continually degrades the estimation process; no singularities have been detected.
- Gradient descent performance indicates real-time capability and a definite FOE initialization suggestion at $(0, 0)^T$.

The beneficial characteristics of this approach are that only the direction of normal flow vectors are used as input (*qualitative* information) and no assumptions about the scene are made (*environmental invariance*).

The good results listed in section 4.4 suggest future experiments towards tiling the image and computing relative FOE's for each (possibly overlapping) subimage. This segmentation could then allow for the coarse detection of independently moving objects in the scene. The results of the FOE estimation process will be further used to compute depth maps from 'time-to-contact' information derived from Eqns. (3) and (4) in order to allow for obstacle avoidance behavior.

Acknowledgements

Ch. Herwig acknowledges the support by a DFG (German Research Council) and City of Bremen Graduate Scholarship (Graduiertenkolleg “Raumorientierung und Handlungsorganisation Autonomer Systeme”).

References

[1] R. Warren and A.H. Wertheim, Eds., *Perception and Control of Self-Motion*, Resources for Ecological Psychology. Lawrence Erlbaum, Hillsdale, New Jersey, 1990.

[2] R. Bajcsy, “Active Perception”, *Proc. of the IEEE*, vol. 76, no. 8, pp. 996–1005, Aug. 1988.

[3] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, “Active Vision”, *Int. J. of Computer Vision*, vol. 1, pp. 333–356, 1988.

[4] D.H. Ballard, “Animate Vision”, *Artificial Intelligence*, vol. 48, pp. 57–86, 1991.

[5] N. Franceschini, J.M. Pichon, and C. Blanes, “From Insect Vision to Robot Vision”, *Phil. Trans. R. Soc. Lond.*, vol. B 337, pp. 283–294, 1992.

[6] G.A. Horridge, "What can Engineers learn from insect vision?", *Phil. Trans. R. Soc. Lond.*, vol. B 337, pp. 271–282, 1992.

[7] M. Lehrer, "Spatial Vision in the Honeybee: the Use of Different Cues in Different Tasks", *Vision Research*, vol. 34, no. 18, pp. 2363–2385, 1994.

[8] M.V. Srinivasan, M. Lehrer, W.H. Kirchner, and S.W. Zhang, "Range perception through apparent image speed in freely flying honeybees", *Visual Neuroscience*, vol. 6, pp. 519–535, 1991.

[9] C. Fermüller, "Navigational Preliminaries", in *Active Perception*, Y. Aloimonos, Ed., chapter 3, pp. 103–150. Lawrence Erlbaum, Hillsdale, 1993.

[10] H.C. Longuet-Higgins and K. Prazdny, "The Interpretation of a Moving Retinal Image", *Proc. R. Soc. Lond.*, vol. B 208, pp. 385–397, 1980.

[11] J.J. Gibson, *The Senses Considered as Perceptual Systems*, Houghton Mifflin, Boston, 1966.

[12] Y. Aloimonos, E. Rivlin, and L. Huang, "Designing Visual Systems: Purposive Navigation", in *Active Perception*, Y. Aloimonos, Ed., chapter 2, pp. 47–102. Lawrence Erlbaum, 1993.

[13] H.-O. Carmesin and Ch. Herwig, "Structure from Motion by Linear Transformations", *Proc. Workshop "Kognitive Robotik", B. Krieg-Brückner, Ch. Herwig (eds.), ZKW–Bericht 3/95, Zentrum für Kognitionswissenschaften, Universität Bremen*, Mar. 1995.

[14] A.M. Waxman and K. Wohn, "Contour Evolution, Neighborhood Deformation, and Global Image Flow: Planar Surfaces in Motion", *Int. J. of Robotics Research*, vol. 4, no. 3, pp. 95–108, Fall 1985.

[15] Y. Aloimonos and Z. Duric, "Active Egomotion Estimation: A Qualitative Approach", in *Proc. 2nd European Conf. on Computer Vision (ECCV)*, G. Sandini, Ed. 1992, Lecture Notes in Computer Science 588, pp. 497–510, Springer Verlag.

[16] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, Cambridge, 1990.

[17] J.E. Cutting, *Perception with an Eye for Motion*, MIT Press, 1986.