A Parallel Algorithm for Depth Perception from Radial Optical Flow Fields

Jens Vogelgesang, Alex Cozzi, Florentin Wörgötter

worgott@neurop.ruhr-uni-bochum.de
Inst. für Neurophysiologie, Ruhr-Universität, Bochum

Abstract. While optical flow has been often proposed for guiding a moving robot, its computational complexity has mostly prevented its actual use in real applications. We describe a restricted form of optical flow algorithm, which can be parallelized on chain-like neuronal structures, combining simplicity and speed. In addition, this algorithm makes use of predicted motion trajectories in order to remove noise from the input images.

1 Introduction

Optical Flow has been proposed as a possible cue to guide robot navigation[1, 2, 3]. Unfortunately general optical flow algorithms are unsuitable for hardware implementation or real-time processing as a consequence of their complexity. In this paper we describe a specialized obstacle detector algorithm that is easily implementable in digital hardware and performs at frame-rate speed.

We simplify the problem introducing two constraints, which cover many generic situations:
1. The robot is supposed to move forward, along the optical axis of the camera-system.
2. The scene is supposed to contain only static objects.
Under these hypotheses the optical flow-field has a pure radial structure, with the center of expansion in the middle of the image.

The distance of the objects contained in the scene is computable from the apparent motion of their projection on the image plane and from the camera's characteristics (lens' focal length and CCD size). In our case, exploiting the radial structure of the optical flow, the problem is further simplified from the 2D structure of a generic flow field to the 1D structure along a radius.

2 Simulated Retina and the Explanation of the Technique

We devised an algorithm to implement the approach in the form of a network of idealized neurons. The network can be interpreted as an artificial retina, with each neuron directly connected to a photo-receptor as in Fig. 1.

The computation takes place by neurons exchanging information with each other in the network. Each neuron in our algorithm needs to be connected only to its two directly adjacent neighbors on the same radius. We call the set of neurons on a radius a *neuron chain*.

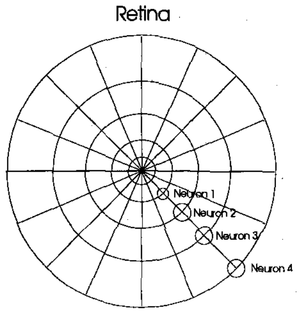

Fig. 1. artificial retina

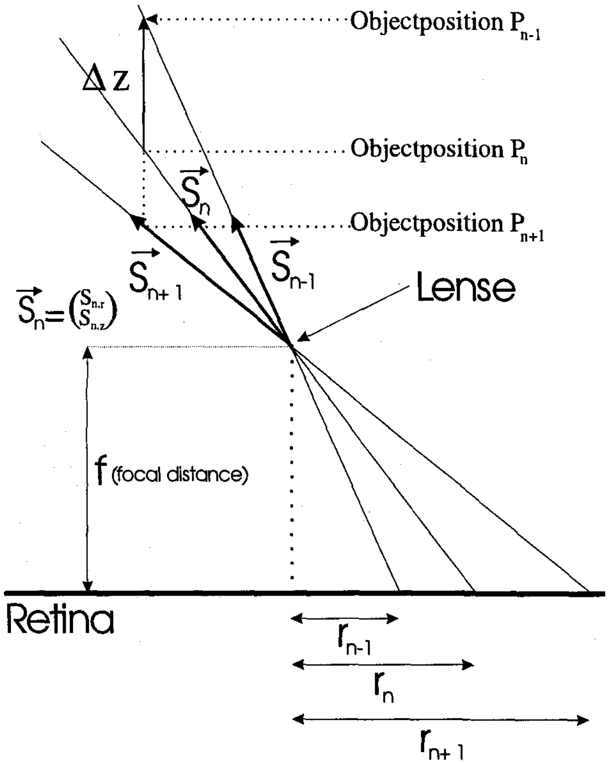

Fig. 2. geometry of position estimation

Let us assume an object position P_{n-1} at time t_{n-1} projected onto neuron r_{n-1} on the retina. Then the projection of this object will be displaced on the retina to r_n due to the robot motion of which covers a distance of $-\Delta z$ between two camera frames such that the real position of the object will be P_n at time t_n. The equations of a pin-hole camera can be solved in our case to compute the position of an object in world coordinates: (Fig. 2) from the system of geometric equations

$$k \cdot \overrightarrow{s_n} + \overrightarrow{\Delta z} = l \cdot \overrightarrow{s_{n-1}} \tag{1}$$

where k and l are scalar factors for the vectors $\vec{s}$. As solution we obtain the object position (X, Y, Z):

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \left(\frac{\Delta z}{s_{n,z} \left(\left(\frac{s_{n,r}}{s_{n-1,r}} \right) - 1 \right)} \right) \begin{pmatrix} s_{n,r} \cos \arctan \left(\frac{s_{n,y}}{s_{n,x}} \right) \\ s_{n,r} \sin \arctan \left(\frac{s_{n,y}}{s_{n,x}} \right) \\ s_{n,z} \end{pmatrix} = \Delta z \cdot \overrightarrow{P_n} \tag{2}$$

All terms of Eq.2, excepted Δz, which is measured between two camera frames, are constants at each given neuron with:

$$\overrightarrow{P_n} = \frac{1}{s_{n,z} \cdot \left(\left(\frac{s_{n,r}}{s_{n-1,r}}\right) - 1\right)} \cdot \begin{pmatrix} s_{n,r} \cdot \cos\arctan\left(\frac{s_{n,y}}{s_{n,x}}\right) \\ s_{n,r} \cdot \sin\arctan\left(\frac{s_{n,y}}{s_{n,x}}\right) \\ s_{n,z} \end{pmatrix} \quad (3)$$

Due to the constant terms, the depth of an object can be determined at the cost of a single multiplication. Solving Eq. 2 is the first step of the algorithm. In order to deal with possible errors introduced by noise or false detections we introduce a second computational step: using the computed position and knowing the actual position of the robot, we can predict where the object will be detected the next time. After having computed the object position (X, Y, Z) at neuron n, we compute the robot position Δz_{pred}, where the neuron $n+1$ is expected to be activated by this object using Eq. 4. If this excitation does not occur, the old object position was incorrect. If it occurs the old position is confirmed and the object becomes "reliable".

$$\Delta z_{pred} = \sqrt{X^2 + Y^2} \cdot \left(\frac{s_{n,z}}{s_{n,r}} - \frac{s_{n+1,z}}{s_{n+1,r}}\right) \quad (4)$$

Each neuron of our artificial retina is composed of three subcomponents: a photoreceptor, a processing unit and a data storage. The processing unit receives information from the photoreceptor and from the adjacent inner neuron, it sends information to the adjacent outer neuron and can write and read data from the storage and perform simple arithmetic operations as specified by the equations.

3 Defining the Algorithm for a Neuron Chain

Let us suppose that our artificial retina is implemented in a camera, moving forward along its optical axis, in a world containing only one black spot somewhere in the direction of gaze. The computational at process along the neuron chain on which the black spot is projected, can be defined by Fig. 3

1. At position P_0 the neuron 1 detects an object. It communicates the event to neuron 2. (Fig. 3,left)
2. At position P_1 the neuron 2 detects the same object (Fig. 3 middle). Since its memory is not empty it can compute a tentative position (X, Y, Z) for the detected object from the position difference $\Delta Z = 8$ and the known position of the photo-receptors (Eq 2). With this information neuron 2 predicts the distance expected to be covered by the camera motion before the object will be detected by the next neuron ($\Delta Z_{pred} = 6$), and communicates the predicted value and the actual position information to neuron 3.
3. When neuron 3 actually detects the object ($\Delta z = 5$), the prediction can be checked: if the precision is satisfactory (e.g., $\Delta z = \Delta z_{pred} \pm 1$) the computed position of the object is updated and marked as reliable, and all informations

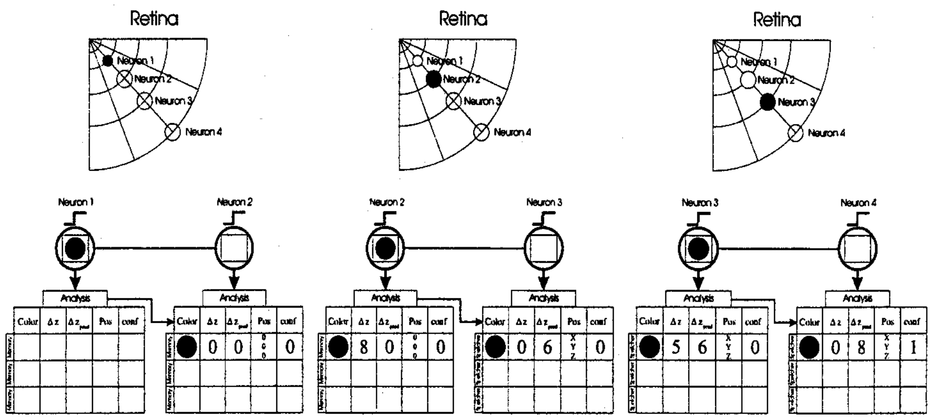

Fig. 3. ideal excitation cycle

are communicated to the next neuron with a reliability value of 1. Otherwise the detected object is considered a new object, and the computation resumes as in 1.

4 Results

Figs. 4-7 show the results obtained from a noisy scene containing three objects at different depths (Fig. 4). Fig. 5 represents the very noisy retina image. Fig. 6 shows the depth information without making use of the predictions while in Fig. 7 only reliable pixels ($conf \geq 3$) are shown and the rough outline of the three objects appears. Depth is coded by gray-scale.

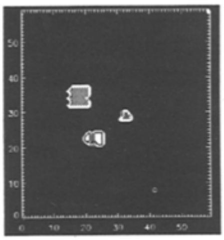

Fig. 4. *first retina image without noise*

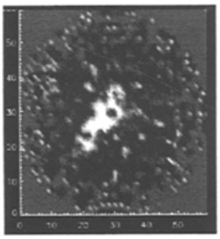

Fig. 5. retina image with noise, step 50

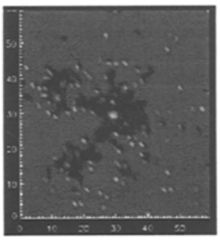

Fig. 6. depth map without prediction, step 50

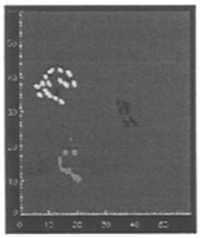

Fig. 7. depth map with prediction, step 50

5 Discussion

The algorithm proposed here can be parallelized with simple computational units on an artificial retina because only a single multiplication is necessary to retrieve the depth information. While this has been suggested before[1] the novel aspect of this study is the use of prediction values to improve the depth maps. Even with such a prediction the arithmetic remains rather simple. The price to pay for this simplicity is the restriction to radial flow-fields. Thus, when changing the direction of motion the algorithm has to be reset. To cover these situations the algorithm should be used in a modular system, where other computationally more expensive depth algorithms[4] are applied to refine the first shot depth maps obtained with our very fast processing scheme. We a currently implementing the algorithm on the robot RHINO of the university of Bonn in cooperation with the computer science department (J. Buhmann and V. Gerdes) and preliminary results indicate that the algorithm will indeed be applicable in real word situations.

6 Acknowledgements

F.W. acknowledges the support of the European community grant ESPRIT BRA 8503 as well as from the Deutsche Forschungsgemeinschaft (Wo 388/5-2).

References

1. Poggio T. & Ancona N., Optical Flow from 1D Correlation: Application to a simple Time-To-Crash Detector /it Massachusetts Institute of Technology (1993).
2. Koenderink J.J. Optic Flow *Vision Res.* **26**, 161-180 (1986).
3. Parado F., Martinuzzi E. Hardware environment for a retinal CCD visual sensor, *Department of Artificial Intelligence, University of Edinburgh, 5 Forest Hill, Edinburg EH1 2QL, Scotland* (1995)
4. Fleet, D., Jepson, A. & Jenkin, M. Phase-based disparity measurement *Comp. Vision, Graphic and Image Proc.*, **53,2**, 198-210 (1991).