Motion Detection by Correlation and Voting*

Stefan Bohrer[1], *Heinrich H. Bülthoff*[2], *and*
Hanspeter A. Mallot[1]

1. Institut für Neuroinformatik, Ruhr–Universität, D–4630 Bochum, FRG
2. Brown University, Dept. of Brain and Cognitive Sciences, Providence, RI, USA

We discuss the properties of a motion detection algorithm published recently by Bülthoff *et al.* [1]. Analytical results show that the algorithm can correctly recover the projected motion in most interesting cases and thus solves the so called weak aperture problem. An implementation on a serial computer is presented that processes image frames within resonable time. Numerical results from this application are included to illustrate the algorithm's performance.

1 INTRODUCTION

Optical flow is generated on the retina of an observer by objects moving relative to the observer. The true motion field W is a 3D vector field whose projection on the image plane is denoted by $W'(\mathbf{x})$ where $\mathbf{x}$ is a 2D vector in the image plane. Unfortunately, the measurement of this 2D field of image velocities from changes of intensity in subsequent images, $E_t(\mathbf{x})$, $E_{t+\Delta t}(\mathbf{x})$, is not possible in general [5]. It is, however, possible to compute suitable optical flows $V(\mathbf{x})$ that are qualitatively similar to the true velocity field in most cases. 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 [1, 3].

2 PARALLEL MOTION ALGORITHM

The algorithm can be described by the following three steps:
Shift and Compare: The expected motion displacements are characterized by an 2D interval $D_\delta := [-\delta, \delta] \times [-\delta, \delta]$. For each node $\mathbf{x}$ and permissible displacement $\mathbf{d} \in D_\delta$, a comparison function $\phi(a, b)$ is evaluated. (Here, a, b denote either greylevels or intensities of preprocessed images.) The output of this step is a matching strength for each node and displacement, $m(\mathbf{x}, \mathbf{d}) = \phi(E_t(\mathbf{x}), E_{t+\Delta t}(\mathbf{x} + \mathbf{d}))$.
Local Summation: At each pixel $\mathbf{x}$ the matching strength for corresponding displacements from the pixels in a neighborhood $P_\nu(\mathbf{x})$ are accumulated. The output of this step is a combined matching strength which, again, is a point wise function:

$$M_\mathbf{x}(\mathbf{d}) := \sum_{\mathbf{y} \in P_\nu(\mathbf{x})} m(\mathbf{y}, \mathbf{d}). \tag{1}$$

*Supported by the German Federal Department of Research and Technology (BMFT), Grant No. ITR8800K4

Winner-Take-All: To each pixel $\mathbf{x}$, the displacement that received the highest matching strength M is assigned as its velocity value $V(\mathbf{x})$ by a winner-take-all scheme. That is, V is selected to satisfy the condition

$$M_{\mathbf{x}}(V(\mathbf{x})\Delta t) \;=\; \max_{\mathbf{d}\in D_\delta} M_{\mathbf{x}}(\mathbf{d}). \tag{2}$$

A large vote for one particular displacement is expected if the motion field is locally constant.

3 TRUE AND RECOVERED MOTION

In this Section, we prove an important Theorem for the continuous approximation of the algorithm:

Theorem: Let $E_1(\mathbf{x})$ and $E_2(\mathbf{x})$ be two image frames with $E_2(\mathbf{x}) = E_1(\mathbf{x}+\mathbf{v})$ for some $\mathbf{v} \in D_\delta$. Let further $\phi : \mathbf{R}^2 \mapsto \mathbf{R}$ be a comparison function in the above sense. Then, if $\phi(a,b)$ is of the form $\psi(a-b)$ with $\psi'(0) = 0$, the true displacement vector (u,v) maximizes the combined matching strength M.

Proof: In the continuous approximation, we have:

$$M_{\mathbf{x}}(\mathbf{d}) = \int_{P_\nu(\mathbf{x})} \phi(E_1(\mathbf{y}), E_2(\mathbf{y}+\mathbf{d}))d\mathbf{y}, \tag{3}$$

Where the integral is taken over a 2D range. In the maximum, the gradient of M, $(\partial M/\partial d_1, \partial M/\partial d_2)$, vanishes. That is, we have to show that $\mathrm{grad}M(\mathbf{v}) = 0$.

$$\mathrm{grad}M_{\mathbf{x}}(\mathbf{v}) \;=\; \int_{P_\nu(\mathbf{x})} \frac{\partial\phi}{\partial b}(E_1(\mathbf{y}), E_2(\mathbf{y}+\mathbf{v}))\ \mathrm{grad}E_2(\mathbf{y}+\mathbf{v})d\mathbf{y} \tag{4}$$

$$\;=\; \int_{P_\nu(\mathbf{x})} \frac{\partial\phi}{\partial b}(E_1(\mathbf{y}), E_1(\mathbf{y}))\ \mathrm{grad}E_1(\mathbf{y})d\mathbf{y}. \tag{5}$$

We now use the relation $\phi(a,b) = \psi(a-b)$ and obtain:

$$\mathrm{grad}M_{\mathbf{x}}(\mathbf{v}) \;=\; \int_{P_\nu(\mathbf{x})} -\psi'(0)\ \mathrm{grad}E_1(\mathbf{y})d\mathbf{y} = 0. \qquad \square \tag{6}$$

Suitable choices of the comparison function that satisfy the condition $\psi'(0) = 0$ are of the type:

$$\phi(a,b) \;=\; |a-b|^n. \tag{7}$$

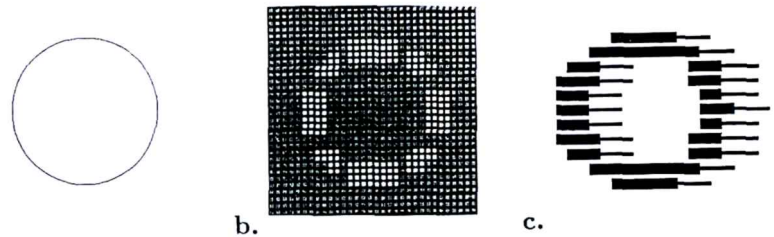
It is easy to show that functions of the type $\phi(a,b) = ab$ lead to systematic errors in the algorithm.

The above theorem shows that the voting algorithm is not subject to the weak aperture problem. As was argued earlier by Reichardt *et al.* [4], the aperture problem seems to be much less severe than was previously thought (for review see [2]). An illustration of how the algorithm deals with the aperture problem is given in Fig. 1.

A circle moves horizontally within the image plane and its motion is correctly recovered by the motion scheme. Fig. 1b shows the region in which motion was detected and Fig. 1c illustrates the calculated velocity field.

4 IMPLEMENTATION AND EXPERIMENTAL RESULTS

The algorithm is serially implemented on a SUN 4/110 workstation. In contrast to the parallel implementation [1], the computational cost is of the order $\delta^2 \times \nu^2$. In general, there is a trade–off between computation time and memory requirements. We have developed two different kinds of implementation with differing advantages:



a. **b.** **c.**

Figure 1: Performance of the algorithm for a synthetic image pair, i.e., a shifted circle. **a.** Input image (only one frame shown). **b.** Regions of detected motion. **c.** Needle–plot. In algorithms subjected to the aperture problem, only the motion components orthogonal to the contour can be recovered. The voting scheme yields correct results, even if only small voting domains ($\delta = 2$) are used.

Minimal Storage: For each pixel $\mathbf{x}_{ij}$, only those matching strength $m(\mathbf{x},\mathbf{d})$ are computed and stored that have not been computed previously. Note that $m(\mathbf{x}_1, \mathbf{x}_2 - \mathbf{x}_1) = m(\mathbf{x}_2, \mathbf{x}_2 - \mathbf{x}_1)$. *(Overlap–Scheme)*

Maximal Speed: For each pixel $\mathbf{x}_{ij}$, all required values of m are stored. *(Nested Loops–Scheme)*

A number of experiments with different kinds of circular and rectangular displacement- and voting-windows was made and various voting rules with a considerable level of complexity were tested against each other. Nevertheless, the simple rule of choosing the maximal excitation by a fixed, space–invariant threshold yielded the best results. This is consistent with the results of Sect. 3, where the winner–take–all mechanism was shown to give correct motion estimates.

A surprising result is the fact that very small voting regions (no more than 7×7 pixel) are sufficient to recover the projected velocity field correctly. Due to this fact, natural image sequences can be processed in a reasonable time. A typical example of a natural scene is shown in Fig. 2.

Fig. 2 shows a sequence of two frames. The resolution is 128×256 pixels in noninterlaced mode. A rectangular displacement area of 27×21 pixels and a relatively small rectangular voting area of 7×7 pixels was chosen. Computation time was roughly 25 minutes with the nested–loop–version and some 90 minutes for the overlap-scheme on a SUN 4/110 workstation.

5 FUTURE WORK

- Parallel implementation on a MIMD architecture (Transputer).
- Adaptive adjustment of voting– and displacement–domains. Expected space-variance (e.g. from optical flow, cf. [3]) can thus be represented by a map of local magnification factors given by the relative sizes of the displacement window.
- Combination of a 1D version of the algorithm with inverse perspective mapping for obstacle avoidance [3].
- Development of a looming- (i.e., "blasting-bomb-") detector based on the local histograms of the $m(\mathbf{x},\mathbf{d})$.

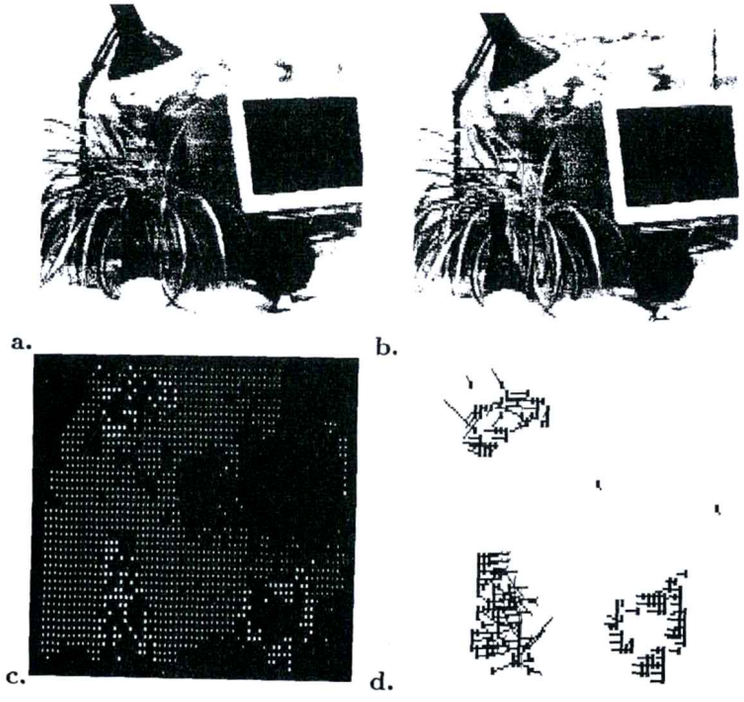


Figure 2: Performance for a natural scene. **a.**, **b.** Motion sequence: the lamp is rotated upwards while the bottle and the bomb are shifted towards each other. **c.** Intensity-coded length of motion displacement. The image is segmented into regions of coherent motion. **d.** Needle-plot.

References

[1] H. H. Bülthoff, J. J. Little, and T. Poggio. A parallel algorithm for real–time computation of optical flow. *Nature*, 337:549, 1989.

[2] E. C. Hildreth and C. Koch. The analysis of visual motion: From computational theory to neuronal mechanisms. *Ann. Rev. Neurosci.*, 10:477 –533, 1987.

[3] H. A. Mallot, H. H. Bülthoff, and J. J. Little. Neural architecture for optical flow computation. Artif. Intell. Lab. Memo. 1067, Massachusetts Institute of Technology, 1989.

[4] W. Reichardt and R. W. Schlögl. A two dimensional field theory for motion computation. First order approximation; translatory motion of rigid patterns. *Biol. Cybern.*, 60:23 – 35, 1988.

[5] A. Verri and T. Poggio. Motion field and optical flow: Quantitative properties. *IEEE Trans. Pattern Analysis and Machine Intell.*, 11:490 – 498, 1989.