Compressive Computation in Analog VLSI Motion Sensors

Rainer A. Deutschmann[1] and Oliver G. Wenisch[2]

[1] Walter Schottky Institute, 85748 Garching, Germany
[2] Technische Universität München, 85748 Garching, Germany
Rainer.Deutschmann@wsi.tu-muenchen.de

Abstract. We introduce several different focal plane analog VLSI motion sensors developed in the past. We show how their pixel-parallel architecture can be used to extract low-dimensional information from a higher dimensional data set. As an example we present an algorithm and corresponding experiments to compute the focus of expansion, focus of contraction and the axis of rotation from natural visual input. A fully integrated system for real-time computation of these quantities is proposed as well. In computer simulations it is shown that the direction of motion vector field is best suited to perform the algorithm even at high noise levels.

1 Analog VLSI Motion Sensors

In the past the computer vision community has invested much effort into developing motion detection algorithms; for a critical review see [BFB94]. Implementing these algorithms in real-time systems proved challenging for computational reasons. Additionally it has been realized that a motion vector field is useful mainly as starting point for further computation, such as segmentation from motion and determination of the focus of expansion.

We have developed and implemented several real-time motion sensors. Using different algorithms, our sensors share the following features: They are single-chip sensors, i.e. the photoreceptors and the motion computation circuitry sit in the focal plane. They are pixel-parallel implementations, i.e. motion computation is performed in synchrony by all pixels. No clock is required for motion computation and, in contrast to digital implementations, in our sensors transistors are used as analog computing elements (*analog* **V**ery **L**arge **S**cale **I**ntegrated). Previously we have proven that analog VLSI motion sensors can be used efficiently for measuring fast rotational velocity [BD97]. In this paper we experimentally show that our 2-D motion sensors can be used for estimating the focus of expansion (FOE), the focus of contraction (FOC), and the axis of rotation (AOR). We use an algorithm that ideally suits the pixel-parallel architecture of the sensors and effectively reduces the computational load from N^2 to $2\,\mathrm{N}$. We also propose a fully integrated 2-D system for on-chip estimation of the FOE, FOC, and AOR, which ultimately compresses the computational load from N^2 to 2. Through computer

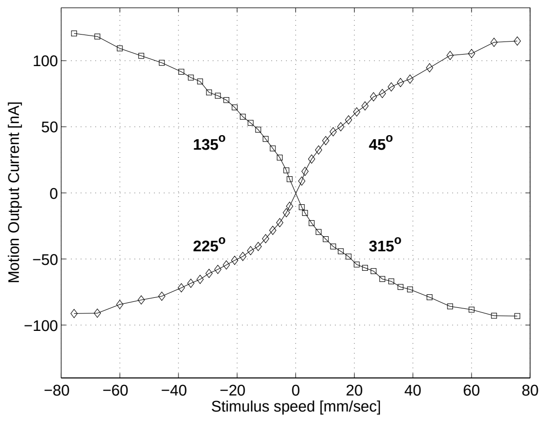

Fig. 1. Mean motion output u_x of one pixel of the Gradient2d sensor for different stimulus velocities.

simulations we have investigated the robustness of the proposed system to noise and low-contrast input.

Our motion sensors fall into two different categories:

1. **Feature-based sensors** generally look for features, such as edges, in the visual field and track them over time. Based on edge tracking for example the sensors reported in [DHK97] compute the direction of motion (DOM) vector field.
2. **Gradient-based sensors** instead use local temporal and spatial derivatives of the light intensity to compute motion. An implementation of the so-called gradient model, which yields velocity independent of spatial frequency and contrast, has been reported [DK98a]. The "Gradient2d" sensor implements an even simpler model for computing the motion vector field (u_x, u_y) [DK98b]:

$$u_x = u_o \frac{\partial I}{\partial t} \tanh\left(\frac{\partial I}{\partial x}\lambda\right) \tag{1}$$

$$u_y = u_o \frac{\partial I}{\partial t} \tanh\left(\frac{\partial I}{\partial y}\lambda\right) \tag{2}$$

where $I(x, y, t)$ is the light intensity distribution on the focal plane, $\partial I/\partial x$ and $\partial I/\partial y$ are both spatial derivatives, $\partial I/\partial t$ is the temporal derivative, u_o and λ are constants. The spatial derivatives are discretely approximated in the implementation. The sensor output u_x of one pixel for different stimulus velocities and orientations is shown in Figure 1. The motion output for higher velocities saturates due to low pass filtering in the photoreceptors and thus becomes direction of motion like.

The compressive computation algorithms of this work are valid for motion vector fields of both categories described above. We present experimental results obtained with a Gradient2d 15×15 pixel array.

2 Determining Axis of Rotation and Focus of Expansion

The FOE of a velocity flow field can be obtained analytically by solving an overdetermined system of $\mathcal{O}(\mathrm{N}^2)$ equations [HS93], where N×N is the array size. We extend the idea proposed in [Bor94] to general flow fields and reduce the computational load to $\mathcal{O}(\mathrm{N})$ by using row and column averages. The X component of the FOE is given by the position of the zero crossing (ZC) of the column average of u_x, the X component of the corresponding motion vectors. If there are multiple ZCs, the one with the maximal slope is to be taken. The FOE Y component and the AOR location are given accordingly:

$$x^{FOE} = ZC_x\left(\sum_y u_x(x,y)\right) \qquad y^{FOE} = ZC_y\left(\sum_x u_y(x,y)\right) \tag{3}$$

$$x^{AOR} = ZC_x\left(\sum_y u_y(x,y)\right) \qquad y^{AOR} = ZC_y\left(\sum_x u_x(x,y)\right) \tag{4}$$

Our motion sensors now can be used not only to compute the motion vector field $u(x,y)$ in real time, but also the row and column averages: The pixel array is addressed by one row and one column scanner, and the motion vector components are represented as bidirectional currents. Thus by gating out one entire row or column, the vector sum is obtained automatically. All N row sums and N column sums are read into a computer, where the ZCs and thus the FOE or AOR is determined. Since only 2N instead of N^2 operations are required, the FOE and AOR can be determined very fast. We achieve frame rates beyond 400Hz.

2.1 Axis of Rotation

We start with describing experimental results on estimating the AOR. We rotate different gray value images in the field of view of the sensor. With a digital computer we continously read out the motion vector field and the row and column averages as computed by the sensor. The vector field is simply read out and displayed for illustration. For AOR/FOE estimation only the averages are necessary. In Figure 2 we show a snapshot of the sensor data as a 'wagon wheel' stimulus was rotating in front of the sensor. In the lower right corner the stimulus as seen by the on-chip photoreceptors is shown. The solid curve in the horizontal graph represents the Y component of the column sum as computed by the sensor, the dotted line is obtained when the vector field is summed external of the sensor; correspondingly, the vertical graph represents the X component of the row averages. It can be seen that the motion vector field clearly reflects the motion of the stimulus. Most importantly, the zero crossing of the row and column average, as shown in Equation 4, obviously marks the location of the AOR.

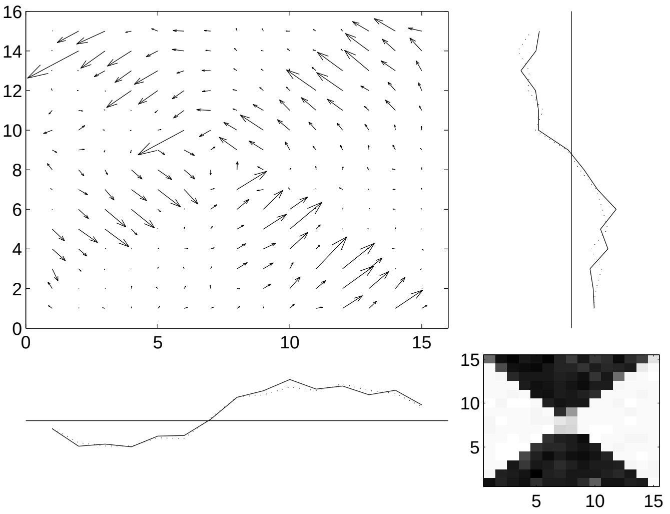

Fig. 2. Upper left: Motion vector field as computed by the sensor. Lower left: Column sum of the Y component, upper right: Row sum of the X component of the motion vectors. Lower right: Visual input as seen by the sensor.

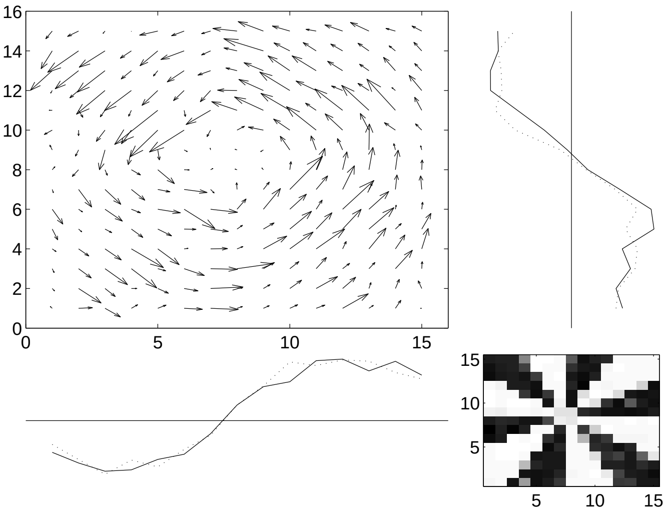

Fig. 3. Determining AOR from a dense flow field.

In Figure 3 a stimulus was used which produced an even denser motion vector field. Once again in the row and column averages computed by the sensor the AOR clearly stands out.

2.2 Focus of Expansion

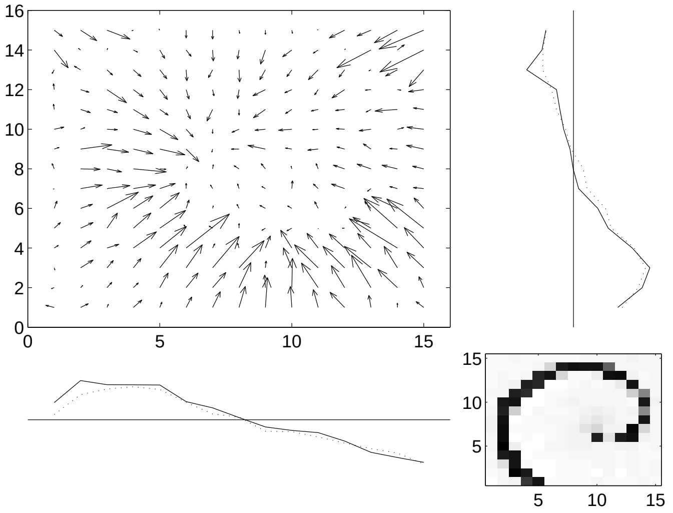

Fig. 4. Determining FOC. Horizontal and vertical graphs represent column and row averages of X and Y components of the motion vectors, respectively.

In order to simulate approaching or receding ego-motion we have rotated a spiral in front of the sensor. In Figure 4 the situation for receding ego-motion is captured: The motion flow field clearly is directed towards the FOC, and the ZCs of the row and column averages indicate its location correctly; see Equation 3. For approaching ego-motion the sensor computes an expanding motion flow field, and the corresponding row and column averages yield the FOE.

2.3 Accuracy

In order to determine how accurately the sensor could determine the AOR and FOE/FOC, we have recorded the row and column averages from the sensor chip several times while the stimulus was rotating at a fixed location. As described above, the location of the AOR was determined from the zero crossing of the appropriate component of the row and column averages as computed by the sensor. We find an absolute error for the X and Y component of 0.21 and 0.29 pixels respectively, the relative error with respect to the array size being 1.4% and 1.9%. These values are surprisingly low. We expect a reliable operation even under less ideal conditions.

2.4 Occlusions

We have tested how the estimate of the AOR would be influenced by partial occlusions of the visual field of the sensor. For that purpose we were using a

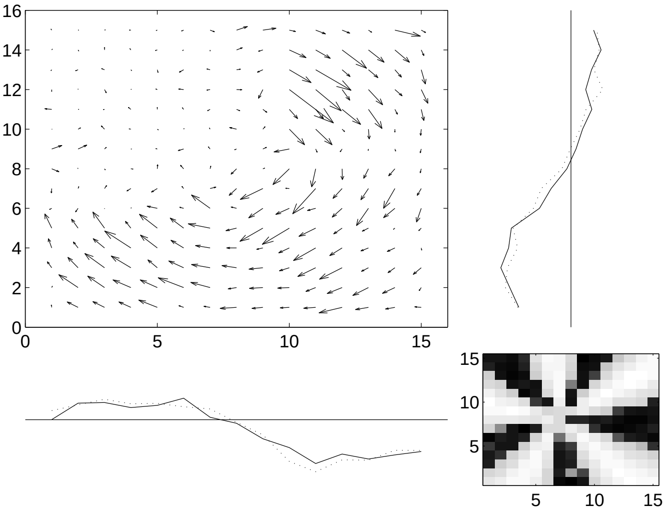

Fig. 5. Sensor output for occluded input. Lower right: Unoccluded input.

rotating sine wave stimulus and occluded about a quarter of it; cf. Figure 5. We have measured the AOR for several different occlusion scenarios, all of which only left three quarters of the full image information to the sensor. The surprising result is that the errors in estimating the AOR did not increase significantly as compared to the non-occluded situation: We find relative errors of 1.8% and 2.3% for the X and Y component, respectively. As can be seen, though, the averages obtained from the occluded areas are less strong, and the slope at the ZC becomes shallower. For that reason the estimate of the AOR becomes less reliable for stronger occlusions.

2.5 2-D FOE On-a-Chip

As we have shown that the compressive computation approach with our analog VLSI sensor is successful, we are now planning a single chip system that also takes over the task of finding the ZC from the row and column averages. The row and column averages are computed from the motion vector field and are then smoothed by a lateral resistive net. Similar to the 1-D design of Indiveri et. al. [IKK95] subsequently the zero crossing with the steepest slope is determined and its location is reported as a single 2-D vector. Thus AOR, FOE or FOC can directly be read off the sensor chip in true real time.

We have carried out computer simulations in order to determine the proposed systems' robustness against noise in the motion vector field, and to find out which motion detection algorithm performs best. We find three main results. First, there is a minimum in the mean absolute deviation of the computed AOR from the true AOR depending on the smoothing length σ; cf. Figure 6. This is

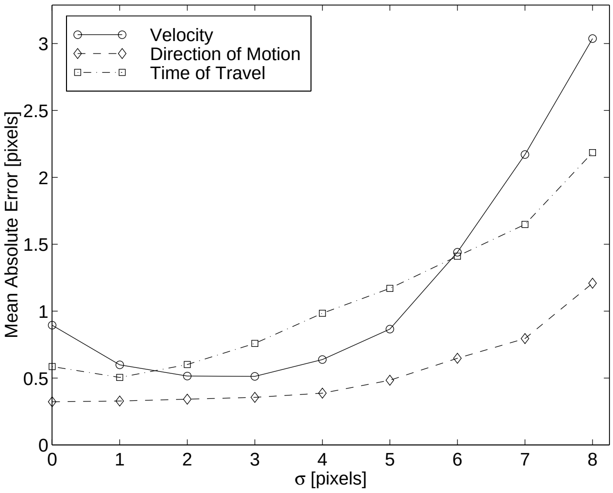

Fig. 6. Simulation result: Sensor performance for varying degree of smoothing.

because of the tradeoff between the positive effect of noise rejection with short-scale smoothing and the erroneous shift of the ZC at large smoothing lengths. Second, it has turned out that the vector field of type direction of motion is best suited for determining the FOE, FOC, and AOR when using our proposed algorithm. The mean absolute error of about 0.3 pixels corresponds well with the measurement results presented in Section 2. Third, Figure 7 reveals that the direction of motion sensor is most insensitive against noise. The saturation of the error curves is due to the finite size of the pixel array.

3 Summary

After briefly introducing different types of analog VLSI motion sensors we have described how their pixel-parallel architecture can favourably be used for computational tasks where from a higher dimensional data set lower dimensional information is to be extracted. As an example we described an algorithm for finding the FOE, FOC and AOR. Using a gradient based 2-D motion sensor we experimentally demonstrated that these quantities can robustly be determined from natural visual input. Finally we proposed a fully integrated 2-D system for real-time estimation of FOE, FOC and AOR and investigated in computer simulations its performance at the presence of noise. We find that direction of motion sensors perform most robust. Future work will concentrate on the implementation of the integrated system.

Acknowledgements: We would like to thank C. Koch for his support and J. L. van Hemmen for helpful discussions.

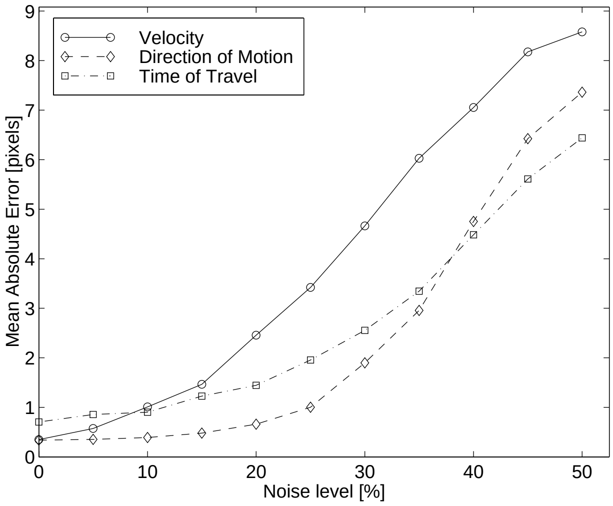

Fig. 7. Simulation result: Sensor performance for increasing noise level.

References

[BD97] C. Born and R. A. Deutschmann. Measurement of fast rotation by VLSI circuits. In *Proc. Deutsche Arbeitsgemeinschaft für Mustererkennung DAGM'97*, 1997.

[BFB94] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Systems and experiment: Performance of optical flow techniques. *Intern. J. Comput. Vis*, 12:43–77, 1994.

[Bor94] C. Born. Determining the focus of expansion by means of flowfield projections. In *Proc. Deutsche Arbeitsgemeinschaft für Mustererkennung DAGM'94*, pages 711–719, 1994.

[DHK97] R. A. Deutschmann, C. Higgins, and C. Koch. Real-time analog VLSI sensors for 2-D direction of motion. In *Proc. International Conference on Artificial Neural Networks ICANN'97*, volume 1327 of *Lecture Notes in Computer Science*, pages 1163–1168. Springer Verlag, 1997.

[DK98a] R. A. Deutschmann and C. Koch. An analog VLSI velocity sensor using the gradient method. In *Proc. International Symposium on Circuits and Systems ISCAS'98*, 1998.

[DK98b] R. A. Deutschmann and C. Koch. Compact real-time 2-D gradient based analog VLSI motion sensor. In *Proc. International Conference on Advanced Focal Plane Arrays and Electronic Cameras AFPAEC'98 Zurich*, 1998.

[HS93] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume II. Addison–Wesley, 1993.

[IKK95] G. Indiveri, J. Kramer, and C. Koch. Analog VLSI architecture for computing heading direction. *Proc. Intelligent Vehicles 1995*, 1995.