2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

An FPGA Implementation of Insect-Inspired Motion Detector for High-Speed Vision Systems

Tianguang Zhang[1], Haiyan Wu[1], Alexander Borst[2],
Kolja Kühnlenz[1] and Martin Buss[1]

[1]Institute of Automatic Control Engineering (LSR)
Technische Universität München
D-80290 München, Germany

[2]Department of Systems and Computational Neurobiology
Max Planck Institute of Neurobiology
Am Klopferspitz 18, D-82152 Martinsried, Germany

Email: {tg.zhang, kolja.kuehnlenz, m.buss}@ieee.org,
haiyan.wu@tum.de, borst@neuro.mpg.de

Abstract— **In this paper, an array of biologically inspired elementary motion detectors (EMDs) is implemented on an FPGA (Field Programmable Gate Array) platform. The well-known Reichardt-type EMD, modeling the insect's visual signal processing system, is very sensitive to motion direction and has low computational cost. A modified structure of EMD is used to detect local optical flow. Six templates of receptive fields, according to the fly's vision system, are designed for simple ego-motion estimation. The results of several typical experiments demonstrate local detection of optical flow and simple motion estimation under specific backgrounds. The performance of the real-time implementation is sufficient to deal with a video frame rate of 350 fps at 256 x 256 pixels resolution. The execution of the motion detection algorithm and the resulting time delay is only 0.25 μs. This hardware is suited for obstacle detection, motion estimation and UAV/MAV attitude control.**

I. INTRODUCTION

Highly accurate real-time stabilization and navigation of humanoids and vehicles is a major research focus of robotics and automation. An important aspect is the use of high-speed visual servoing control loops running at framerates of several 100Hz controlling and stabilizing the motion of the system. This paper contributes an implementation of a high-speed motion estimation model.

A fly's panoramic vision system comprises at its front end several thousand photoreceptors feeding into a 2D array of motion detecting neurons which the animal uses for dynamic visuomotor pose and gaze stabilization and navigation in 6 degrees of freedom. The Reichardt detector [1] [2] [3] [4] [5] [6] is a well-known model which describes, at an algorithmic level, the process of local motion detection in the fly, leading from non-directional input to a direction selective output. In a structure of the fly brain called 'lobula plate' large neurons are found which integrate these local motion signals and additionally form extensive connections amongst themselves [7][8]. These neurons have large receptive fields and respond best to particular flow-fields such as occurring during certain maneuvers of the fly in free flight [9] [10]. In engineering applications such as robotics, driver assistance systems or surveillance systems, a camera system is usually used as a sensor to gather information about the environment. Motion perception based on the fly's vision system is computationally cheap and, thus, particularly suited for real-time applications.

In addition to optimizing the motion estimation algorithm, it is also required to select a suitable hardware. In [11] a new EMD circuit implemented on micro-air vehicles (MAV) has been designed by using Field Programmable Analog Array (FPAA). Besides, several EMDs-based models were developed based on Very-Large-Scale-Integrated (VLSI) circuits. In [12], a low-power VLSI chip was described which consists of a one-dimensional array of EMDs to perform motion computation. In [6], a biologically inspired VLSI system for measurement of self-motion was introduced. Later on, Harrison designed and tested a single-chip analog VLSI sensor that could detect imminent collisions [13]. Recently, FPGAs are favoured by engineers to implement EMDs. In [14], a real time algorithm for estimating motion vectors has been implemented. In [15], a FPGA implementation of a bio-inspired visual sensor is introduced. However, these implementations perform motion estimation with relatively low resolution and low frame rate.

In this paper, an array of biologically inspired EMDs is implemented on an FPGA platform exploiting its major advantages: the execution of massive truly parallel computations in only one processing cycle and the pipeline structure in data processing. Based on [16] two new simple templates of receptive fields for rotation detection are proposed to facilitate specialized motion detection and cover all types of movement. The performance of the implementation is sufficient to deal with video frame rates of 350 fps or above for a frame size of 256 x 256.

The remainder of this paper is organized as follows: firstly, in Section II, the basic and elaborated EMD models are introduced. In Section III, we added two templates of recep-

tive fields based on [16] and simulations are conducted. The implementation of the elaborated EMD model and receptive fields on an FPGA platform is introduced in Section IV. Experimental results are shown in Section V. Conclusions are given in Section VI.

II. MOTION DETECTION WITH EMDS

Motion detection is one of the most basic tasks a visual system has to perform. Motion information is not explicitly represented at the output level of retinal photoreceptors. Instead, it has to be computed from the changing retinal images by the nervous system [17]. Using an elaborated motion detector model and some receptive fields, some properties of the biological visual system have been converted into a computational model for engineering applications to estimate ego-motion.

A. The Reichardt Model

The earliest and probably the most famous model of motion detection inspired by biological systems was developed by Reichardt and Hassenstein in 1956 [1]. Fig. 1 presents a simplified version of the correlator model.

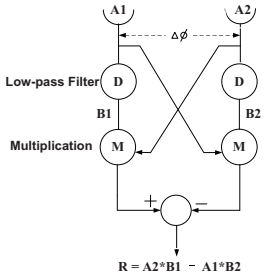

Fig. 1: The simple Reichardt detector [7]

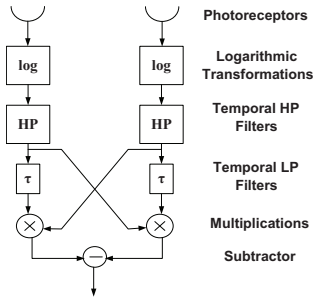

Fig. 2: The elaborated EMD

$A1$ and $A2$ are two photoreceptors. Their outputs are temporally delayed by a low-pass filter D. With $A1(t)$ and $A2(t)$ representing the input signals at the left and right inputs, and $B1(t)$ and $B2(t)$ representing the corresponding filtered signals, one obtains the output $R(t)$ of a motion detector:

$$R(t) = A2(t) \cdot B1(t) - A1(t) \cdot B2(t). \quad (1)$$

The detector generates a direction sensitive response because of the subtraction between the two symmetric detector halves. As shown in Fig. 3, a positive peak-like signal is generated for the motion to right and a negative peak-like signal is generated for the motion to left. The response is zero when no motion exists. However, the response of the Reichardt detector is not always as simple as a peak. For example, as shown in Fig. 3, the sign of the response in f) does not directly indicate motion direction.

B. The Elaborated EMDs

The simple Reichardt detector has two major drawbacks: i) its response is sensitive to edge contrast, reducing robustness

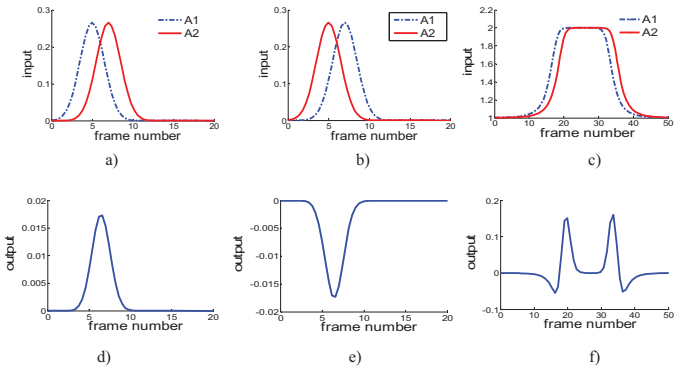

Fig. 3: Response of the Reichardt detector to a moving peak and to a moving pulse [16]; a) a peak moving to right ; b) a peak moving to left; c) a pulse moving to right; d) response to a); e) response to b); f) response to c).

to lighting conditions; ii)its response to step edges is complicated, making scene interpretation difficult. Therefore, the simple detector has been improved and two preprocessors are added:

- logarithmic transformation is applied in order to reduce the sensitivity to lighting conditions directly after the receptors;
- this is followed by a temporal high pass filter in order to obtain a simple response to the most common edge type as step edge in natural images.

The elaborated EMD is shown in Fig. 2, which is similar to the one proposed by [16]. The moving pulse signal c) in Fig. 3 (or equivalently a moving step edge) is transformed by the temporal HP filter into a moving peak as shown on the left of Fig. 4, leading to a simple response as shown on the right of Fig. 4.

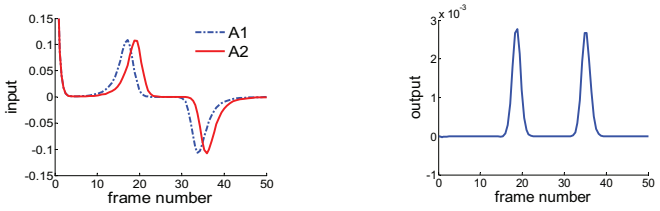

Fig. 4: Response of the elaborated motion detector to a pulse moving to right [16].

C. Two-Dimensional Motion Detection

As previously explained, an EMD can only detect one-dimensional motions along the line connecting its receptors. In [14] one simple algorithm is implemented to realize two-dimensional motion detection.

In order to detect two-dimensional motions, a pair of EMDs are combined as shown in Fig. 5. The vertical motion vector component is observed by receptors P_V and P_C while the horizontal motion vector component is observed by receptors P_H and P_C. The outputs of receptor P_C are labeled R_V and R_H in the right picture of Fig. 5. Following [18], this EMDs pair is a combination of two types of EMDs:

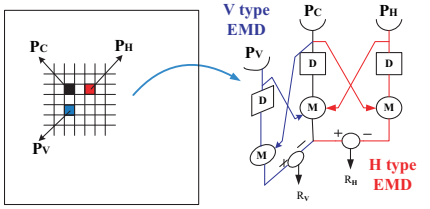

Fig. 5: Two-dimensionally placed receptors

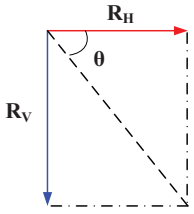

Fig. 6: Motion direction

- H-type EMD responding to local horizontal motion.
- V-type EMD responding to local vertical motion.

Motion direction can be estimated by computing the ratio of R_V and R_H as illustrated in Fig. 6 and formulated as:

$$\theta = \arctan(\frac{R_V}{R_H}). \quad (2)$$

D. The Receptive Field for Motion Estimation

In neurobiology, the receptive field of a sensory neuron is a region of space in which the presence of a stimulus will alter the firing of that neuron. The receptive fields of the so-called motion-sensitive wide-field neurons in the fly brain are targeted in this project. Motion sensitive neurons are cells in a fly's visual system and they have been found to be involved in optical flow (obtained by the proposed EMDs) processing. These neurons can be classified by their sensitivities to different kinds of motions. For example, some of them are more sensitive to horizontal motion, while some others are more sensitive to vertical motion in their receptive field. Then, these neurons integrate the signals of EMDs spatially in their receptive fields, where each single EMD only analyzes the local motion along its sensitive direction.

Four simple templates (a, b, d and e in Fig. 7) have been proposed in [16]. a) and d) are used for translation detection and b) and e) are used for expansion detection.

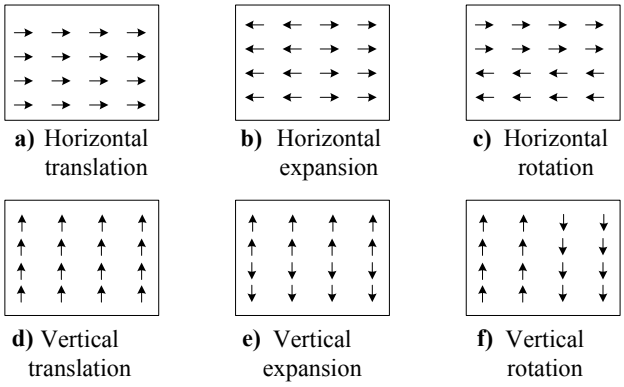

Fig. 7: The six simple templates of receptive fields

III. Development of Receptive Field Templates and Simulation Results

A. Development of Receptive Field Templates

Based on [16], two additional templates of receptive fields for rotation detection are proposed in this paper: c) is sensitive to the rotation of vertical edges and f) for horizontal edges. By adding these two templates, the system is supposed to cover all types of movements. We can use the templates to detect motion in images. Each template demonstrates a receptive field which is sensitive to a certain motion.

B. Simulation Results

To validate our designed templates of receptive fields for rotation detection, a simple simulation with Interactive Data Language (IDL, CREASO) is conducted. The input of this simulation is a series of pictures captured by a camera which was fixed on the end effector of a robot arm (Stäubli) and moved in front of a static background composed of black and white squares. Through the operation of the robot arm various movements of the camera can be executed, such as translation and rotation. Fig. 8 shows the results of rotation.

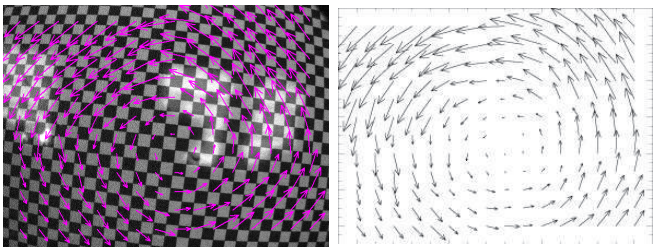

Fig. 8: Optical flow of rotation. Left: real picture with background; right: optimized result without background.

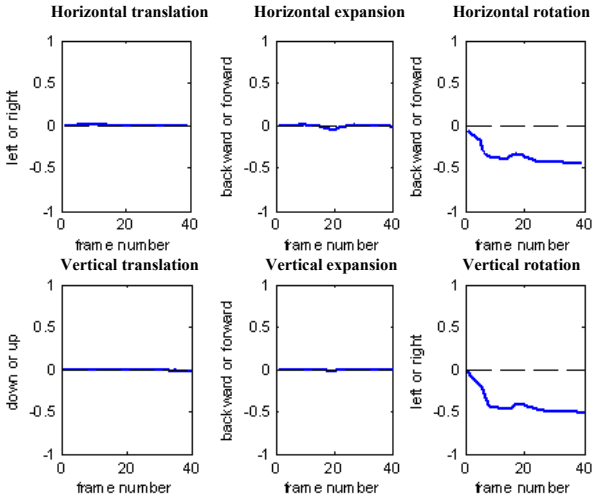

Fig. 9: Responses of the templets of the respective field of Fig. 7 to the rotation of a camera.

Fig. 9 shows the outputs of the six templates. The outputs of the templates of the horizontal rotation and the vertical rotation are more obvious than those from the other four templates. The negative values indicate that the camera moves in an anticlockwise direction. Due to the symmetry of the background, the responses of both receptive fields for the rotation detection are almost equal. But in the natural world, they are sensitive to horizontal and vertical edges. This simulation shows that these proposed six templates can estimate simple translations and rotations successfully.

IV. IMPLEMENTATION ON FPGA

This section deals with the implementation of the algorithm mentioned above on an FPGA platform.

A. Development Environment

The system structure is shown in Fig. 10. Firstly, the image is continuously captured by a High speed camera (MC1311, Microtron). Then, the image is transmitted to the Tsunami FPGA platform (SBS technologies) with Altera® Stratix® EP1S40 FPGA processors. The elaborated motion detector and algorithm of the receptive fields are implemented on the FPGA and the image is processed here. After processing of the image, the results of the EMDs and related outputs of the receptive fields are sent to host PC (AMD Opteron242 with 2GB RAM).

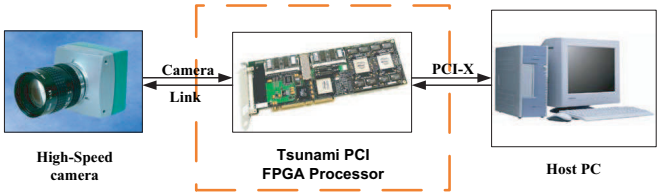

Fig. 10: Hardware platform

Fig. 11 shows the overall physical system hierarchy illustrating the communication between different modules. The main focus of this study, namely FPGA implementation of the model and algorithm in VHDL (Very High Speed Integrated Circuit Hardware Description Language), is highlighted by the red dashed rectangle of Fig. 11.

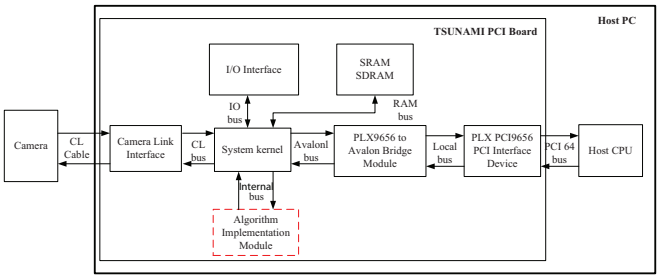

Fig. 11: Hardware system hierarchy [19]

B. Architecture of the VHDL Program

The VHDL program architecture is shown in Fig. 12. All the image processing is performed along with the data flow. Firstly, the image data (8-bit iDATA) is provided by the camera. Secondly, the 8-bit iDATA is fed into the EMDs module, which performs the optical flow calculation in both horizontal and vertical directions. The other inputs of the EMDs module are the previous outputs of the filters that are saved in the on board M-RAM. Then, the outputs of the EMDs are connected to two modules: the "data insert module" and the "receptive field calculation module". Finally, all of the results are written back to original data flow which is transferred to host PC subsequently. All of these modules will be introduced briefly in the next part.

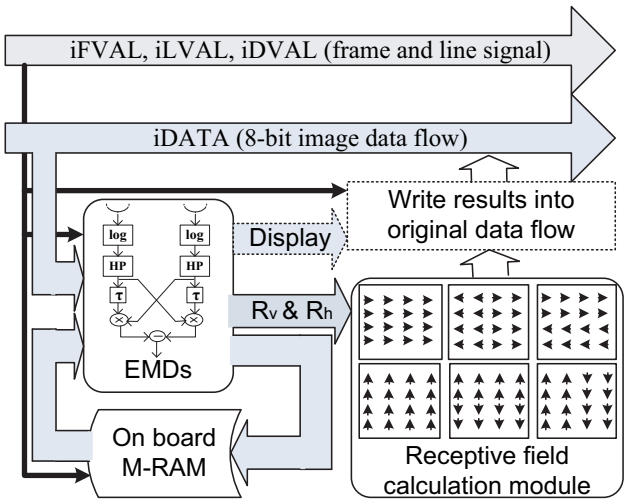

Fig. 12: VHDL program architecture of the elaborated model

Besides, there are three signals relevant to images: frame signal - iFVAL, line signal - iLVAL and data signal - iDVAL. These signals going with the images data flow will also be sent to all control modules in order to synchronize the image processing.

C. Components Design

The elaborated motion detector proposed in Section II has mainly five components that should be programmed in VHDL: logarithmic transformation, high-pass filter, low-pass filter, multiplication and subtractor. The design of a recursive single pole temporal high-pass filter is based on a temporal low-pass filter, in this part the structures of them are introduced together. The designs of multiplication and subtractor are relatively simple and neglected in this section.

The algorithm of the six receptive fields is also implemented on FPGA. Besides, a special module, called *data insert module*, is designed to write the calculated results back to the original data flow. The structures of these modules are also briefly introduced here.

1) Logarithmic transformation: In this project, the module of logarithmic transformation is realized through a look-up table. The output of the photoreceptor is an 8-bit value and has a range between 0 and 255. For more precision, the values are extended to 16 bits in the following form: the highest bit indicates the value being either positive or negative, bits 14 to 6 belong to the integer part, and the last 6 bits are the fraction part.

2) High-pass filter: The high-pass filter selected for this project is a recursive single pole temporal high-pass filter. Fig. 13 shows the expression and structure of the filter. Firstly, a single pole temporal low-pass filter is designed which is defined by $y_i = x_i/\tau_L + (1 - 1/\tau_L)y_{i-1}$. The parameter τ_L is changed to τ_H when it is assumed to be a high-pass filter. As a high-pass filter and a low-pass filter are running at the same time in the model, there are two parameters τ_H and τ_L. For this project, these two parameters are designed to be configurable dynamically in order to adapt to different situations. When given the result of the low-pass

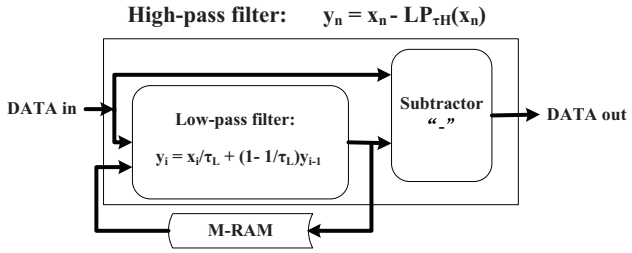

Fig. 13: The architecture of the high-pass filter

filter, subtracting the input signal by the output value of the low-pass filter is the result of the high-pass filter. The current output of the low-pass filter is saved in the on-board memory.

3) M-RAM controller: In Fig. 12, an on-board M-RAM is required to save the output values of filters. An M-RAM controller is designed to distribute the read address, write address and write enable signals.

It has to be mentioned that, because using less line pixels can not increase the image capture frequency, the input frame from the camera is set up to the resolution of 1280 x 256 pixels, but only 256 x 256 pixels will be processed in our approach. The write enable signal is used to perform this selection and a pixel counter (calculates the pixel number in a line) is needed. When the pixel counter is between 512 and 767 , the output values of the filters are stored with the M-RAM controller.

4) Receptive field calculation module: As illustrated in subsection III-A, six templates of the receptive field (Fig. 7) are designed for motion estimation. The output of each template for each image is designed to be a signed 40-bit value (5 bytes). The highest bit indicates that the value is positive or negative. The total output values of the six templates have 30 bytes.

5) Data insert module: In order to analyze and display the results on a Host PC, a data insert module is designed to write the results of both H-type EMD and V-type EMD in the selected 256 x 256 range back to the original data flow. The image data which are not in the region of interest are replaced by the results of EMDs as well as the outputs of the six templates. When the data flow is sent to the host PC, the results can be read from it and thus the local motion vector is drawn over the original frame. The results of the receptive field template are also read out and plotted in realtime.

The main frequency of the FPGA system is 160 Mhz. For each image (1280 x 256) the transfer from the camera to FPGA takes approximately 2*ms*. Only 40 system-clocks (about 0.25μs) are totally charged additionally for the whole implementation on FPGA. Thus, this system can work ideally at more than 450fps. In this respect, our system is more efficient than some already existing solutions, such as the VLSI sensor in [13] (132-pixels image, 30pfs) and the FPGA implementation in [14] (88 x 88 pixels, 29fps).

V. REAL-TIME EXPERIMENTAL RESULTS

The system is successfully tested in several experiments. Different motion manners are detected under various backgrounds. In our experiments, the backgrounds were always fixed in front of the camera while the camera was mounted on the end-effector of a 6 DOF Stäubli robot arm moving along some simple trajectories.

A. Horizontal and Vertical Translations

The left picture of Fig. 14 shows the optical flow on the background when the camera translates horizontally. The arrows (green line with yellow head) represent the local optical flow detected by the elaborated EMDs. Almost all of them are pointing right. Since the output of the EMD is influenced by several parameters, such as velocity, luminance and contrast, the lengths of the arrows are not identical throughout the whole picture. In addition, the spherical effect of the wide-angle camera also influences the result to some extent.

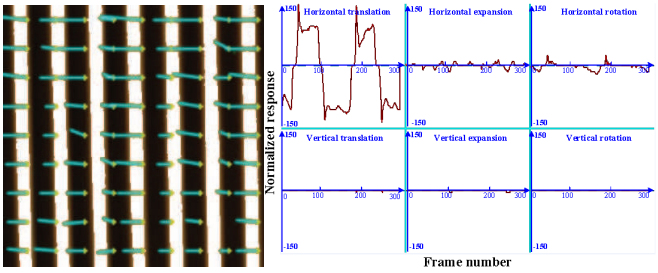

Fig. 14: Optical flow and receptive field results for horizontal translations with a pattern of vertical bars.

The right picture of Fig. 14 shows the results of the six receptive field templates. They are normalized to the range between -150 and 150. Ideally, when the camera moves horizontally, only the horizontal translation template will have an output. And the outputs of the other five templates should always be zero. The result in this experiment is almost coincident with the expectation. The positive/negative value denotes that, the camera moves horizontally to the right/left. The horizontal expansion and horizontal rotation have very small outputs, which are probably caused by the asymmetry between the left and right half-pictures as well as the top and bottom half-pictures. The three vertical templates' outputs are almost zero. Combining the results of these six templates, a conclusion can be drawn: the camera is moving horizontally right or left during this experiment.

B. Expansion and Contraction

The elaborated motion detector can also be used to detect optical flow field induced by the camera moving backwards or forwards. A diffused or contracted optic flow field can be observed. In this experiment, the background of some concentric rings is applied. The circle center is set to be aligned with the optical axis of the camera.

By moving the camera backwards away from the background, a contracted optic flow field can be observed, as shown in the left picture of Fig. 15. In the right picture, the outputs of both horizontal expansion and vertical expansion templates are obvious. Moreover, the horizontal translation and vertical translation templates have relatively small outputs. The main reason is that the center of the template is not coincident with the optical axis of the camera.

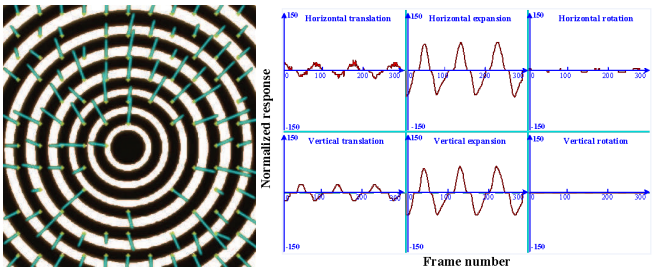

Fig. 15: Optical flow and receptive field results for expansion using the background of concentric rings.

C. Rotation

The left picture of Fig. 16 shows the optical flow of clockwise camera rotation with a pattern of radial lines. The right picture is the result of the receptive field for rotation. In the experiment, the camera rotated alternately clockwise (positive values) and anticlockwise (negative values).

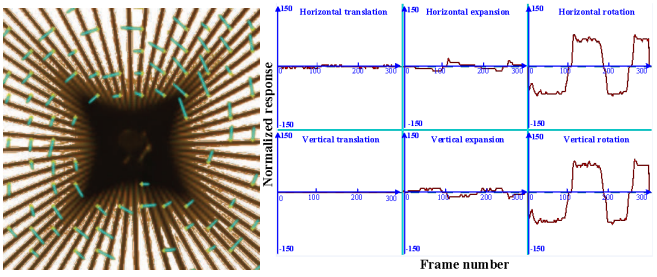

Fig. 16: Optical flow and receptive field results for rotation with the background of radial lines.

According to the experimental results and discussions above, the system is demonstrated to be efficient to detect local optic flow induced by different motion manners of the camera. The elaborated motion detector and the receptive field templates implemented on FPGA can detect motion direction fast and correctly. The system is successfully tested at a frame rate of 350 fps with the resolution of 256 x 256. The system delay caused by computations is approximately only 0.25 μs. This system can be used for motion estimation and obstacle avoidance for humanoids, ground and aerial vehicles. It will also be suitable for UAV/MAV attitude control.

VI. Conclusions

In this paper, an array of EMDs has been implemented on an FPGA platform. The EMDs are interconnected realizing various receptive fields for motion detection. It is successfully tested at a high frame rate (350 fps) with a resolution of 256x256 vectors. The time delay of the computation is only 0,25 μs. The results of the simulation and the experiments demonstrate that this elaborated model enables local optical flow detection and simple global motion estimation under specific backgrounds. However, if the model is applied in realistic engineering system, where the background is certainly more irregular than the backgrounds used in the experiments, data fusion must be done in the future. And in order to find out the relationship between the output of EMD and motion velocity clearly more parameters should be taken into consideration, such as the luminance, contrast, etc.

VII. Acknowledgments

This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys*, see also `www.cotesys.org` and the Bernstein Center for Computational Neuroscience Munich, see also `www.bccn-munich.de`.

References

[1] B. Hassenstein and W. Reichardt, "Systemtheoretische Analyse der Zeit-Reihenfolgen, und Vorzeihenauswertung bei der Bewegungsperzeption des Ruesselkaefers," *Naturforch*, vol. 11b, p. 513?524, 1956.

[2] M. Egelhaaf and A. Borst, "A look into the cockpit of the fly: Visual orientation, algorithms, and identified neurons," *The Journal of Neuroscience*, vol. 13, pp. 4563–4574, 1993.

[3] W. Reichardt and M. Egelhaaf, "Properties of individual movement detectors as derived from behavioural experiments on the visual system of the fly," *Biological Cybernetics*, vol. 58, pp. 287–294, 1988.

[4] A. Borst and M. Egelhaaf, "Principles of visual motion detection," *Trends Neurosci*, vol. 12, pp. 297–306, 1989.

[5] T. Neumann and H. Bulthoff, "Insect inspired visual control of translatory flight," *Advances in Artificial Life. 6th European Conference, ECAL 2001. Proceedings*, vol. 2159, pp. 627–636, 2001.

[6] C. M. Higgins and S. A. Shams, "A biologically inspired modular VLSI system for visual measurement of self-motion," *IEEE SENSORS JOURNAL*, vol. 2, pp. 508–528, 2002.

[7] A. Borst and J. Haag, "Neural networks in the cockpit of the fly," *Journal of Comparative Physiology A*, vol. 188, pp. 419–437, 2002.

[8] J. Haag and A. Borst, "Neural mechanism underlying complex receptive field properties of motion-sensitive interneurons," *Nature Neurosci*, vol. 7, pp. 628–634, 2004.

[9] H. Krapp and R. Hengstenberg, "Estimation of self - motion by optic flow processing in single visual interneurons," *Nature*, vol. 384, pp. 463–466, 1996.

[10] H. Cuntz, J. Haag, F. Foerstner, I. Segev, and A. Borst, "Robust coding of flow-field parameters by axo-axonal gap junctions between fly visual interneurons," *PNAS*, vol. 104, pp. 10 229–10 233., 2007.

[11] F. Ruffier, S. Viollet, S. Amic, and N. Franceschini, "Bio-inspired optical flow circuits for the visual guidance of micro-air vehicles," in *Proceeding of the IEEE International Symposium on Circuits And Systems*, 2003.

[12] S.-C. Liu, "A neuromorphic aVLSI model of global motion processing in the fly," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS*, vol. 47, pp. 1458–1467, 2000.

[13] R. Harrison, "A biologically inspired analog ic for visual collision detection," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, pp. 2308–2318, 2005.

[14] E. Nakamura, S. Asami, T. Takahashi, and K. Sawada, "Real time parameter optimization for elementary motion detectors," in *2006 IEEE International Conference on Image Processing*, 2006, pp. 1065–1068.

[15] F. Aubepart, M. El Farji, and N. Franceschini, "FPGA implementation of elementary motion detectors for the visual guidance of micro-air-vehicles," *Industrial Electronics, 2004 IEEE International Symposium on*, vol. 1, pp. 71–76, 2004.

[16] F. Faille, A. Borst, and G. Faerber, "Biologically inspired motion detector for video interpretation," biological Cybernetics, 2007, under review.

[17] A. Borst, C. Reisenman, and J. Haag, "Adaptation of response transients in fly motion vision. II: Model studies," *Vision Research*, vol. 43, pp. 1311–1324, 2003.

[18] E. Nakamura, M. Ichimura, and K. Sawada, "Fast global motion estimation algorithm based on elementary motion detectors," *International Conference on Image Processing*, vol. 2, pp. 297–300, 2002.

[19] *Wave FPGA Tool Kit Documentations, Tsunami PCI Technical Data Sheet, SBS Technologies.*