Scalable Computing: Practice and Experience
Volume 8, Number 4, pp. 387–394. http://www.scpe.org
ISSN 1895-1767

A COMPUTING ARCHITECTURE FOR CORRECTING PERSPECTIVE DISTORTION IN MOTION-DETECTION BASED VISUAL SYSTEMS*

SONIA MOTA†, EDUARDO ROS‡, AND FRANCISCO DE TORO§

Abstract. The projection of 3D scenarios onto 2D surfaces produces distortion on the resulting images that affects the accuracy of low-level motion primitives. Independently of the motion detection algorithm used, post-processing stages that use motion data are dominated by this distortion artefact. Therefore we need to devise a way of reducing the distortion effect in order to improve the post-processing capabilities of a vision system based on motion cues. In this paper we adopt a space-variant mapping strategy, and describe a computing architecture that finely pipelines all the processing operations to achieve high performance reliable processing. We validate the computing architecture in the framework of a real-world application, a vision-based system for assisting overtaking manoeuvres using motion information to segment approaching vehicles. The overtaking scene from the rear-view mirror is distorted due to perspective, therefore a space-variant mapping strategy to correct perspective distortion arterfaces becomes of high interest to arrive at reliable motion cues.

Key words. Real-time computing, high performance computing, fine grain pipeline, image processing.

1. Introduction. Animals and human beings have powerful tools for processing information. Recent advances in biological neural circuits and processing schemes is one of the reasons of a new tendency in engineering that emulates specific biological computation schemes, this is the research paradigm called *neuromorphic engineering*. The objective is to achieve more effective machines with a huge potential impact on industry and society [1, 2, 3, 4].

Vision is one of the most important senses for animals' survival. In particular, visual motion detection is the most important information source and constitutes a complex and accurate system. The long-medium term goal is to implement devices based on vertebrates' visual systems, because of their astonishing efficiency in analysing dynamic scenes. However, current vision models based on vertebrates require high computational cost while most real-time applications cannot be addressed with traditional computer vision strategies due to their complexity.

But adapting bio-inspired processing schemes on silicon is a complex task. The neural system has synaptic plasticity (the connection from neuron A to neuron B changes in order to stabilize specific neural activity patterns in the brain, for instance with neural adaptation strategies such as *Hebbian learning* [5]) that allows response to changes to different stimulus or environments. Furthermore the connectivity among neurons in biological tissues takes place in three dimensions. In contrast, the silicon systems allow only two-dimensional connectivity among computational threads and lack abilities such as local synaptic plasticity.

Biological systems use efficiently massive parallel processing to overcome the slow chemical-based computing that takes place in neurons. This advantage of biological systems is shared by current FPGA devices. Different researchers are working in this direction, i. e. bio-inspired visual systems implemented on FPGAs devices with massively parallel computation using fine grain processing architectures [6, 7, 8, 9]. This approach allows real-time image processing and represents a first step towards solutions to particular problems in a wide range of applications

However, even biological systems need to project 3D scene onto a 2D surface (for instance, a retina or a camera sensor) before extracting data. Due to the 2D projection the scene is distorted by perspective. This affects motion processing, a moving object, although moving at a constant speed, seems to accelerate and its size increases as it approaches the camera. This apparent enlargement adds an expanding motion to the translational one, and the perception of different velocities in different regions of an object.

Biological systems use low level stereo information or other visual modalities in higher level processing stages to deal with the perspective distortion. But uni-modal motion-based artificial systems require other strategies

*This work has been supported by the European Project DRIVSCO (IST-2001-35271) and the National Project DEPROVI (DPI2004-07032)

†Departamento Informática y Análisis Numérico, Universidad de Córdoba, Campus de Rabanales s/n, Edificio Albert Einstein, 14071, Córdoba, Spain (smota@uco.es). Questions, comments, or corrections to this document may be directed to that email address.

‡Departamento de Arquitectura y Tecnología de Computadores, Universidad de Granada, Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain (eros@atc.ugr.es).

§Departamento de Teoría de la Señal, Telemática y Comunicaciones, Universidad de Granada, Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain (ftoro@ugr.es).

to compensate this effect. We propose a scheme that corrects perspective distortion so that motion information can be used in a reliable manner: *Space-variant mapping* (SVM) method. It is possible to compensate for the effect of perspective by remapping the image before extracting motion. This processing unit can be connected to the whole motion detection system as a pre-processing stage of the image.

The rest of this paper is organized as follows: section 2 introduces the space variant mapping method; section 3 describes the hardware implementation and cost; and section 4 presents an example of perspective distortion correction in a real-world task, an overtaking monitoring system. Furthermore, the perspective distortion correction is described using two different methods: space variant mapping (SVM) and another bio-inspired method based on neural integration of information that we use in order to validate the results and compare the two different approaches.

2. Space-variant mapping method. The *space-variant-mapping* (SVM) method is the selected strategy for dealing with perspective distortion. The Space Variant Mapping [10, 11] is an affine coordinate transformation that aims at reversing the process of projection of a 3-D scene onto a 2-D surface. It is possible to invert the projection equations and to compensate the effect of perspective by remapping the original image. In this approach (a) parallel lines and equal distances in the real scene are remapped to parallel lines and equal distances in the processed (remapped) image and (b) it is assumed that the depth of the scene, i. e. distance to the camera projected on its optical axis, varies linearly.

Generally, distances closer to the image plane are projected onto larger segments. Using these assumptions the SVM approach re-samples the original image. We assume a specific camera configuration targeting the left vision field with respect to the optical axis. In this case, the required remapping is done by expanding the left-hand side of the image (corresponding to the part of the scene furthest away from the camera) and collapsing the right-hand side (corresponding to the part of the scene closest to the camera). The coordinates at the distorted space are transformed in new coordinates at the remapped space. The operations involved in the process are additions, multiplications, divisions and trigonometric operations (sine and tangent).

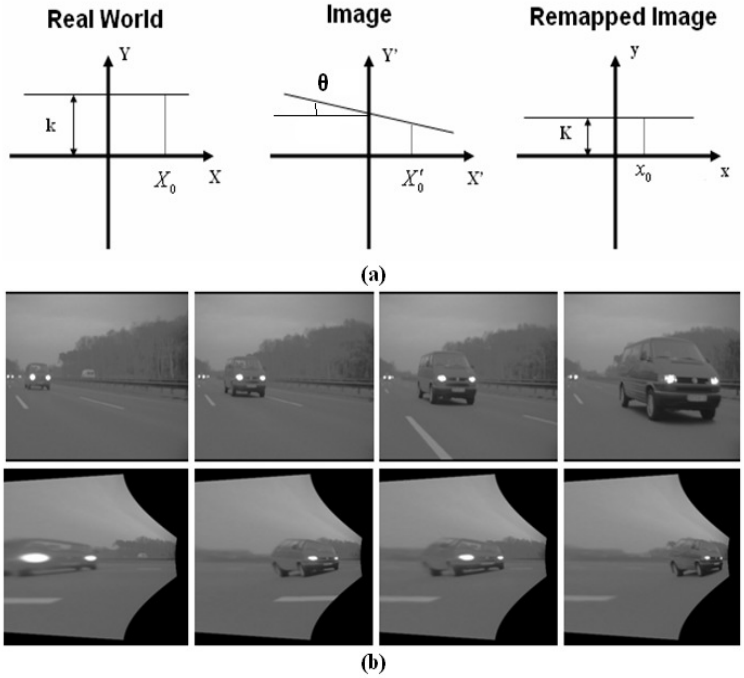

FIG. 2.1. ***(a)*** *Coordinates transformation;* ***(b)*** *Original and remapped image of an overtaking sequence.*

Figure 2.1a shows the coordinates transformation that is required in order to correct the perspective distortion due to the projection of the 3D scene onto a 2D surface. Figure 2.1b shows an example of a re-sampled image from the real-world application described in Section 4.

The blurred appearance of the left-hand side of the image is generated by the interpolation process necessary to resize a small portion of the original image into a larger area. The interpolation method used here is the

truncated Taylor expansion, known as *local Jet* [12]. In the remapped scenario the mean speed of a car that is actually overtaking at a constant relative speed is more constant along the sequence. Furthermore, each point that belongs to the rigid body moves approximately at the same speed (Figure 2.2). On the other hand, on the right-hand side of the remapped image, we are subsampling the original image, which means that aliasing effects may occur.

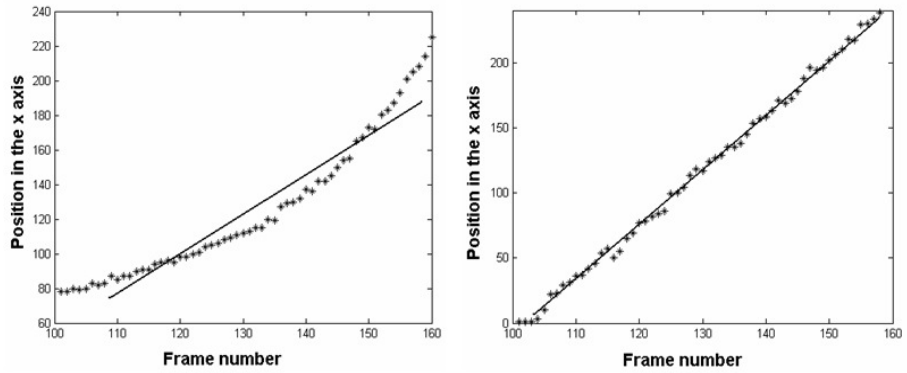

FIG. 2.2. *Space Variant Mapping makes stable the speed along the sequence. On the left plot we represent the x position of the centre of the car along the constant-speed overtaking sequence. We see that although the overtaking sequence speed is constant the curve is deformed (constant speed is represented by a line) due to the perspective distortion. On the other hand, the right plot shows the same result on a remapped sequence. In this case the obtained overtaking speed is constant (accurately approximated by a line with a slope that encodes the speed).*

The advantage of SVM is that the effect of perspective is compensated through the remapping scheme and the acceleration artefact is removed. In the real-time application described in Section 4, we have manually marked the overtaking car position along a scene, in this way it is easy to compute the centre of the marked area, i. e. the overtaking car, and its speed. Figure 2.2 shows the compensation effect on the speed of the centre of the overtaking car.

Furthermore SVM reduces the difference between the extracted speeds of the front and rear of an overtaking car. Finally, the remapped image is easier to interpret using motion estimation information.

3. Hardware implementation. We use conventional cameras that provide 30 frames per second and 256 gray levels. The processed image size is of 640 x 480 pixels. The prototyping computing platform has 2 SRAM banks and a Xilinx Virtex-II FPGA (XC2V1000 device) [13]. This device allocates 1 million system gates distributed in 5,120 slices and 40 embedded memory blocks of a total of 720 Kbits.

The whole system has been implemented on the FPGA device (see Figure 3.1). This system includes the processing stages (space variant mapping, motion detection algorithm [14] and specific circuits for packing and unpacking temporal data) and the interface elements (frame-grabber, memory management units and VGA output interface).

The complete system is designed with independent processing modules. The architecture design adopts a fine grain pipeline structure for all modules. Specific communication channels are used in order to connect the modules with each other.

In this way, space variant mapping (SVM) constitutes a pre-processing stage before the motion estimation module. The architecture of the whole system allows changing modules of the datapath if necessary, i. e. we can use different modules implementing diverse motion-detection algorithms with the same system.

SVM architecture is also implemented as a fine grain pipeline structure to ensure a successful connection with the motion extraction module at 1 pixel per clock cycle. Motion primitives are computed using a fine grain pipeline structure that consumes 1 cycle per stage. If necessary, it is possible to reduce the parallelism in the SVM module (and consequently its efficiency) to fit the processing performance of the motion-detection module requirements (if other motion estimation schemes are used). Alternatively, we can replicate the SVM module and split the image to send parts of the original image to the different SVM units increasing the processing velocity if further performance is required. Therefore the architecture is modular and scalable. Figure 3.1 shows the data flow of the integrated system.

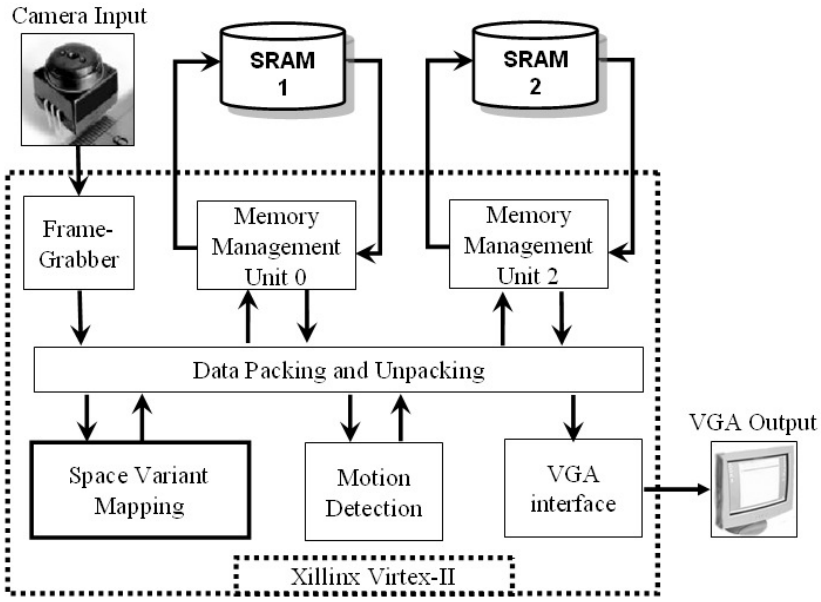

FIG. 3.1. *Complete system: motion detection after correcting the distortion by space variant mapping preprocessing.*

SVM uses several multiscalar units consistent with the goal. To transform each pixel coordinates the operations that take place are additions, multiplications, divisions and trigonometric operations (sines and tangents). To compute sine and tangent we use a look up tables, and to compute the divisions we use optimized cores customized for our application. Each core computes one division and consumes one cycle. We use two division cores. Figure 3.2 shows the pipeline structure of the modules related with perspective distortion correction. Rectangles represent multiscalar units. Rectangles on a column are working in parallel. Rectangles on a row represent different pipeline stages. Numbers in brackets are the number of micropipelined stages. The final block represents the motion estimation datapath.

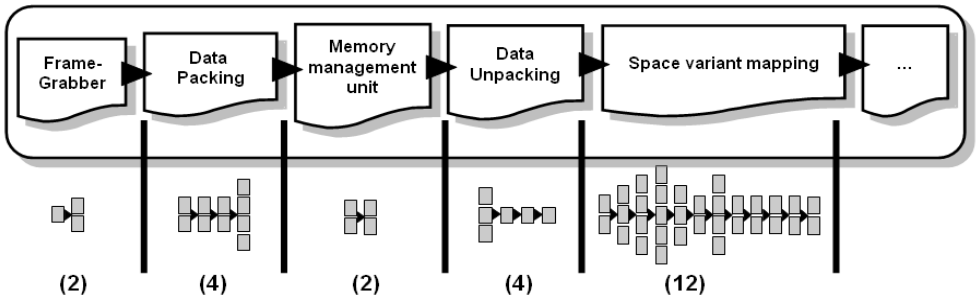

FIG. 3.2. *Data flow and pipelined structure of the perspective distortion correction datapath.*

The SVM module takes 12 pipeline stages, and only one division core that produces 28 clock cycles of latency.

Table 1 summarizes the main performance and hardware cost of the system implemented. The hardware costs in the table are estimates extracted from the ISE environment. Note that the maximum clock frequency advised by the ISE environment is limited to 36.1 MHz (Table 1). This is because we use a specific core for the division that limits the global frequency of the whole pipelined structure. However, the circuit frequency fully allows computation at camera frame-rate.

One of the important bottlenecks for FPGA processing capability is the external memory access. There are several reasons to use the external SRAM: first of all, conventional cameras interlace the image (they send even rows first and then odd rows of a scanned image). Therefore, in order to compute the image it is necessary to previously de-interlace the image, i. e. to arrange the rows in properly appearance order. Furthermore, SRAM

TABLE 3.1

Hardware cost of the different stages of the described system. The global clock of the design is running at 31.5 MHz, although the table includes the maximum frequency allowed by each stage. The data of the table has been extracted using the ISE environment.

Pipeline Stage	Number of Slices	% Device Occupation	Max. Fclk.(MHz)
Frame-Grabber	753	14	75.9
Memory Management Units	581	11	53.8
Space-Variant Mapping	838	16	36.1

access is shared by space variant mapping modules and motion detection modules. Finally, the synchronization among different modules related to different clock frequencies (frame-grabber, VGA, etc.) is done with external memories.

Using exclusively embedded memory blocks becomes not possible due to the image size. Therefore, the necessity of storing data in external SRAM banks forces us to design a module that allows the writing and reading to/from the SRAM banks as efficiently as possible. This process of storing/recover data is sequential and consumes 2 cycles per pixel (1 cycle is consumed in assigning the address and 1 cycle is consumed in transferring the data). The access control is carefully designed. We define different reading and writing ports using a double-buffer technique to avoid temporization problems. We use a micropipelined architecture to access two different ports. A state machine feeds the reading/writing ports sequentially, achieving a performance of one data per cycle. Furthermore, it is feasible to store several pixels at each memory address due to the memory word size. In this way we can reduce the number of external memory accesses. For this purpose we use specific packing and unpacking circuits in the pipelined architecture (see Figures 3 and 4).

4. Real-world application. One of the most dangerous operations in driving is to overtake another vehicle. The driver's attention is on the road, and sometimes he does not use the rear-view mirror or it is unhelpful when an overtaking car is at the blind spot. Therefore an automatic alarm system is of interest in these scenarios.

Systems based on vision would be very effective in driving assistance; in fact the driver himself uses vision and represents a good proof of the concept. We place a camera onto the real-view mirror to cover the blind spot area. If an overtaking vehicle approaches the host car it is detected as forward moving features, while the rest of the patterns in the camera visual field move backwards due to the ego-motion of the host vehicle. Therefore motion provides useful cues to achieve an efficient segmentation in this application framework. In this context, the sequences taken with a camera fixed onto the driver's rear-view mirror are strongly deformed by the perspective, and reducing the deformation effect is necessary in order to enhance the segmentation capabilities of a motion-based vision system.

We define two different methods to deal with the perspective distortion. On one side, we use space variant mapping method, and on the other side, for validation purposes we use an alternative bio-inspired method based on neural integration of information.

Many studies suggest that the integration of local information allows the discrimination of objects in a noisy background [15, 16, 17, 18]. The mechanism of this integration in biological systems is almost unknown. We define *velocity channels* based on motion patterns of the image that seem to correspond to independent moving objects (rigid bodies) [19]. Each velocity channel computes a population of features moving coherently (by sharing velocity and direction in a local area). The velocity channels are processed in a competitive manner and the one that integrates a maximum number of features moving coherently in an area becomes salient. In this way, low quality motion-detection estimations, i. e. errors, are filtered.

The system has been tested on real overtaking sequences in a wide speed range.

The *"centre of mass"* of obtained features (moving coherently) is used to validate the quality of moving features. We manually mark the overtaking car by drawing a rectangle (around it). We calculate the distance between the centre of mass and the centre of the rectangle. This distance is normalized by dividing it by the radius of the minimum circle containing the rectangle in each frame. This distance is what we call *Quality Measure* (QM). If the centre of mass falls into this circle this QM is below 1. In this case we assume that we are detecting the overtaking vehicle accurately. In other cases the QM is higher than 1, motion detection has dominant noisy patterns (motion detection is assumed to be of low quality) leading to incorrect estimations. Figure 4.1a shows the QM along the sequence when velocity channels method is adopted, and Figure 4.1b shows QM throughout the sequence when the space-variant mapping method is adopted.

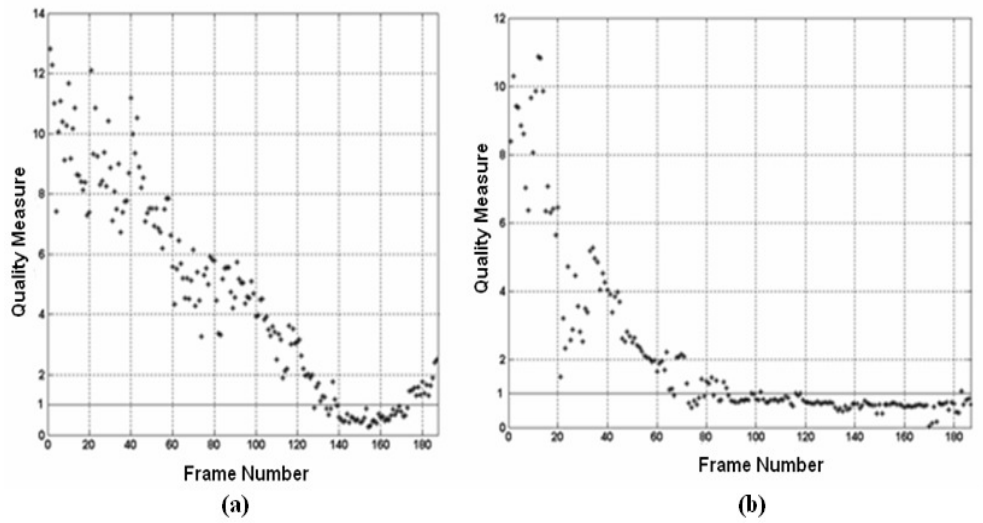

FIG. 4.1. *Quality measurement plot of a sequence adopting:* ***(a)*** *Velocity-channels method;* ***(b)*** *Space-variant-mapping method.*

When the velocity channels method (VC) is used, motion detection is accurate from frame 138 to 175, and when the space variant mapping (SVM) method is adopted, motion detection is accurate from frame 89 to the end of the sequence. In fact accurate detection occurs when the overtaking vehicle begins to be dangerously close (see Figure 4.1).

We used four sequences to test the space variant mapping scheme. The results are summarized in Figure 4.2. The first and the fourth sequences were taken with a CCD camera on a sunny day. In the first sequence the overtaking car approaches from the distance and in the fourth, it suddenly appears into our line of vision. The second and third sequences are HDR ones. The second one corresponds to a cloudy day with some mist and the other was taken in twilight conditions. These two sequences show overtaking processes by far-away cars with their lights on.

Figure 4.2 shows that motion detection is done properly from a vehicle size of 10660 pixels with the VC method and 3216 pixels with the SVM method. This size is only approximate, taken as it is from the size of the confidence rectangle used to calculate QM. The data in the next column represents the number of features detected moving rightwards, on which the estimation is based.

In the HDR sequences the cars have their lights on, and adverse weather conditions reduce noisy detection. The best detected features belong to the overtaking car lights and allow an early success in the tracking task with both methods.

SVM constitutes a good method for medium distances in all weather conditions, even when the cars have no lights on that facilitate their detection.

5. Conclusions. We have presented a perspective distortion correction for a vision-based segmentation system.

Using a real-world sequence of a car moving at constant speed we showed that the SVM considerably reduces the spurious acceleration effect due to perspective projection and improves motion estimation results.

We have compared the results of Space-variant mapping method with a bio-inspired one based on neural integration of information. Adopting space variant mapping method the results based on motion information are improved.

We have designed a pipelined computing architecture that takes full advantage of inherent parallelism of FPGA technology. In this way we achieve computing speeds of 36.1 Mpixels (for instance, around 30 frames per second with 1280x960 image resolution) that allow fully computation at camera frame-rate (25-30 frames per second).

The architecture is modular and scalable.

This contribution is a good case of study that illustrates how very diverse processing stages can be finely pipelined in order to achieve high performance.

Sequence Frame	Method used	1st. Frame in successful tracking	Vehicle Size (pixels)	Best results
	VC	139	10660	
	SVM	89	3216	√
	VC	1	899	√
	SVM	1	899	√
	VC	1	1813	√
	SVM	1	1813	√
	VC	36	32469	
	SVM	23	14385	√

FIG. 4.2. *Results of the two methods applied to four different sequences. "1st frame of successful tracking" represents image number in a sequence from which the motion detection is of high quality, i. e. QM is below 1. "Vehicle size" is the number of pixels inside the manually drawn rectangle that contains the overtaking car in the "1st frame of successful tracking".*

Finally the hardware resources of the system are not very high. Therefore, the presented approach can be considered a moderate cost module for the real world application of the overtaking car monitor.

REFERENCES

[1] E. ROS, E. M. ORTIGOSA, R. AGÍS, M. ARNOLD AND R. CARRILLO, *Real time computing platform for spiking neurons Real Time Spiking Neurons (RT-Spike)*, IEEE Trans. Neural Networks, 17(4) (2006), pp. 1050–1063.

[2] S. MOTA, E. ROS, E. M. ORTIGOSA AND F. J. PELAYO, *Bio-Inspired motion detection for blind spot overtaking monitor*, Int. Journal of Robotics and Automation, 19(4) (2004), pp. 190–196.

[3] A. BROGGI, P. CERRI AND P. C. ANTONELLO, *Multi-resolution vehicle detection using artificial vision.*, in Intelligent Vehicles Symposium, 2004, pp. 310–314.

[4] T. DELBRUCK, *Silicon retina with correlation-based, velocity-tuned pixels*, IEEE Trans. Neural Networks 4 (1993), pp. 529-541.

[5] D. O. HEBB, *The organization of behavior*, Wiley, New York, 1949.

[6] J. DÍAZ, E. ROS, F. PELAYO, E. M. ORTIGOSA, AND S. MOTA, *FPGA based real-time optical-flow system*, IEEE Trans. Circuits for Video Technology, 16(2) (2006), pp. 274–279.

[7] S. MOTA, E. ROS, J. DÍAZ AND F. TORO, *General purpose real-time image segmentation system*, Lecture Notes in Computer Science 3985 (2006), pp. 164-169.

[8] F. AUBÉPART AND N. FRANCESCHINI, *Bio-inspired optic flow sensors based on FPGA: Application to Micro-Air-Vehicles*, Microprocessors and Microsystems 31(6), (2007), pp. 408–419.

[9] P. CHALIMBAUD AND F. BERRY, *Embedded active vision system based on an FPGA architecture*, EURASIP Journal on Embedded Systems, volume 2007 (2007), Special Issue on Embedded Vision System.

[10] H. MALLOT, H. H. BULTHOFF, J. J. LITTLE AND S. BOHRER, *Inverse perspective mapping simplifies optical flow computation and obstacle detection*, Biol. Cybern., 64 (1991), pp. 177–185.

[11] S. TAN, J. DALE AND A. JOHNSTON, *Effects of Inverse Perspective Mapping on Optic Flow*, in ECOVISION Workshop, 2004, (Isle of Skye, Scotland, UK).

[12] L. FLORACK, B. TER HARR ROMENY, M. VIERGEVER AND J. KOENDERINK, *The Gaussian Scale-Space paradigm and the multiscale local Jet*, Int. J. Comp. Vis. 18 (1996), pp. 61–75.

[13] WWW.XILINX.COM

[14] S. MOTA, E. ROS, J. DÍAZ, R. RODRIGUEZ AND R. CARRILLO, *A space variant mapping architecture for reliable car segmentation*, Lecture Notes in Computer Science 4419 (2007), pp. 337–342.

[15] H. B. BARLOW, *The efficiency of detecting changes of intensity in random dot patterns*, Vision Research, 18(6) (1978), pp. 637–650.

[16] D. J. Field, A. Hayes and R. F. Hess, *Contour integration by the human visual system: evidence for local "association field"*, Vision Research, 33(2) (1993), pp. 173–193.
[17] J. Saarinen, D. Levi and B. Shen, *Integration of local pattern elements into a global shape in human vision*, in Proceeding of the National Academic of Sciences USA, Vol. 94, 1997, pp. 8267–8271.
[18] C. D. Gilbert and T. N. Wiesel, *Intrinsic connectivity and receptive field properties in visual cortex*, Vision Research, 22(2) (2005), pp. 125–177.

Edited by: Javier Díaz and Dorothy Bollman
Received: December 14th, 2007
Accepted: December 27, 2007