A hardware structure for time-to-impact computation using log-polar images

Antonios Gasteratos,
Department of Production and Management Engineering
Democritus University of Thrace
GR-671 00, Xanthi, Greece
agaster@pme.duth.gr

Paschalis Gonidis and Ioannis Andreadis
Department of Electrical and Computer Engineering
Democritus University of Thrace
GR-671 00, Xanthi, Greece
pgonidis, andread@ee.duth.gr

***Abstract* - This paper describes the design of a new hardware system for the optical flow and time-to-impact computation with real-time response. This is achieved by estimating the optical flow of the object on the camera plane. In order to reduce the image size without losing significant information, the log-polar transformation is utilized. The system employs a sequence of log-polar images for the optical flow and time-to-impact computation of a moving object. The algorithm that has been implemented belongs to "differential techniques" category, which is suitable for parallel computation of the parameters. The structure of the implementation allows the processing of grey-level log-polar images of 45x60 pixels in real time (25 frames per second)**

Index Terms* - *Log-polar transformation, Optical Flow, Time-to-impact, FPGA.

I. INTRODUCTION

Optical flow is the estimation of the 2-D projection of the 3-D movement of an object in space [1, 2]. It can be used in computing motion detection, time-to-impact, structure, focus of expansion as well as in object segmentation [3-7]. Its applications vary from autonomous vehicles visual navigation to intelligent robotics and so forth. Unfortunately, most optical flow techniques have high computational cost and, thus, they are slow and most of them inappropriate for real-time computation. Notwithstanding, the computational burden can be overcame if the volume of the image data is reduced. In this case the optical flow measures would be inaccurate and sparse, unless a special topological image arrangement is used. Such an arrangement should be space-variant, as the log-polar one. The latter varies across the radius of the image, providing thus high resolution in the fixation point and low resolution in the peripheral zone [8].

The time-to-impact estimation is very important for the living beings. Their vision system must be able to detect any obstacle and to produce a quick and immediate estimation of the time to impact with these obstacles. Without this specific characteristic, the animals are not able to control their movements and to react in unforeseen events. In artificial vision, the time-to-impact is as crucial as in animal vision, because it allows real-time obstacle avoidance and protection of the system that bears the vision. In such systems, the computation of time-to-impact follows the computation of the optical flow.

In this paper, the structure of the digital circuitry for time-to-impact computation is presented. The log-polar transformation is explained, which is the first step of the implementation. The next step is the computation of optical flow and time-to-impact. This unit, which is the basic part, requires at least two log-polar images as an input, produced from the first step. The output is the time-to-impact, resulting from the optical flow computation, using a differential algorithm. The circuit was designed and implemented in Quartus II - Altera software environment, a tool for designing and simulating FPGAs.

II. THE LOG-POLAR TRANSFORMATION

Like most of birds and mammals the human vision system consists of receptors in the retina. These are the cones, involved in the diurnal vision and rods, responsible for the night one. The cones are placed in space with a growing density towards the center of the visual system called the 'fovea' and with a failing density towards the periphery, as shown in Figure 1. Consequently, the resolution also decreases, moving away from the fovea toward the periphery. The distribution has a radial symmetry, which can be approximated by a polar distribution. The aim of this phenomenon is the careful surveillance of a wide area with a detailed distinction of the objects at the fixation point.

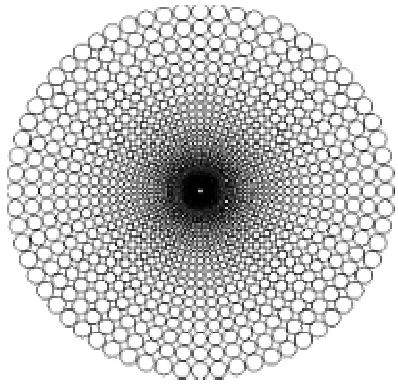

Fig. 1 Log-polar topological arrangement

A log-polar system, which imitates the function of the eye, reduces the processing data without losing the significant one [8]. Mathematically, the change of the co-ordinates according to the log-polar transformation can be done in two steps. In the first step the transformation from the Cartesian plane (x,y) (image plane) to the polar plane (ρ,θ) is performed according to eqns (1), and (2); in the second step the log-polar co-ordinates are computed from the polar ones, according to eqns (3) and (4).

Polar coordinates:

$$\rho = \sqrt{(x - x_c)^2 + (y - y_c)^2} \quad (1)$$

$$\theta = \tan^{-1}\left(\frac{y - y_c}{x - x_c}\right) \quad (2)$$

Log-polar co-ordinates:

$$\eta = q\theta \quad (3)$$

$$\xi = \log_a \frac{\rho}{\rho_o} \quad (4)$$

where x, y are the Cartesian co-orinates, (x_c, y_c) is the center of the image, ρ_o is the radius of the innermost circle and 1/q is the minimum angular resolution of the log-polar layout.

Figure 2 illustrates the log-polar layout as derived from equations (1), (2), (3) and (4). In particular, in the grid on the left represents a standard Cartesian image mapped according to equations (3) and (4). The plot on the right shows the corresponding log-polar (cortical) image.

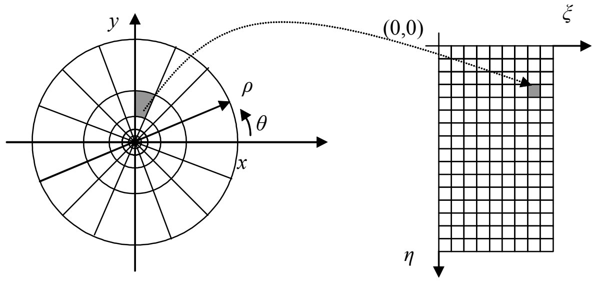

Fig. 2 The log-polar transformation.

In Figure 3 there is a real image as an example of the transformation: In Figure 3a a Cartesian gray-level image is depicted; applying the log-polar transformation on the previous image, the cortical one, presented in Figure 3b, is obtained. Figure 3c presents the remapped Cartesian image. The loose of information in the periphery is apparent, however the image resolution in the fovea is very high and no detail is wasted. On the other hand, the cortical image is 10 times smaller than the Cartesian one.

(a) (b) (c)

Fig. 3 (a) Cartesian image; (b) log-polar image and (c) reconstructed Cartesian from log-polar image.

III. Optical Flow and Time-to-impact estimation

A. Optical Flow

A fundamental problem in processing of image sequences is the measurement of optical flow or "image velocity", i.e. an approximation to the 2-D motion field of an image sequence. The 2-D image motion is the projection of the 3-D velocities of object in Cartesian space onto the image plane. Hence, this is the relative motion between the camera and the object. Several methods for the optical flow computation can be found in the literature, which are categorized and presented in [1]. In the proposed structure a differential one was implemented. This computes the optical flow from the spatiotemporal derivatives of the image intensities. The method was selected due to the fact that the amount of image data is reduced and parallel techniques can be implemented for the computation of the parameters.

Let us consider the image brightness at the point (η, ξ) in the image plane at time t be denoted by $I(\eta, \xi, t)$. Let us also consider the case that a pattern moves into the image plane. The brightness of a particular point in the pattern is constant, or $dI(\eta,\xi,t)=0$, and using the chain rule for differentiation:

$$\frac{dI}{\partial\eta}\frac{d\eta}{dt}+\frac{dI}{\partial\xi}\frac{d\xi}{dt}+\frac{dI}{dt}=0 \quad (5)$$

or

$$I_\eta u + I_\xi v + I_t = 0 \quad (6)$$

where $u=d\eta/dt$ and $v=d\xi/dt$ are the coordinates of the velocity vector and I_η, I_ξ and I_t are the partial derivatives of image brightness in log-polar space.

In order to compute the components of the velocity vector, further constraints are applied, i.e.:

$$\begin{bmatrix} I_{\eta\eta} & I_{\xi\eta} \\ I_{\eta\xi} & I_{\xi\xi} \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}+\begin{bmatrix} I_{t\eta} \\ I_{t\xi} \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (7)$$

Therefore, for the estimation of optical flow three steps have to be implemented. The first step is a spatial Gaussian smoothing of the input image sequence. For this purpose a weighted-mean linear filter is utilized. Linear filters possess excellent characteristics in removing noise of Gaussian distribution, as the one imposed to the images due to camera electronics. The weight values are given by a convolution mask, which convolves with the image. The mask of the applied simple smoothing filter, which is a rough approximation of a Gaussian filter, is:

$$\frac{1}{8}\mathbf{x}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

In the next step the computation of the spatial derivatives I_η and I_ξ and the temporal derivative I_t are implemented. The input to the derivative block is the smoothed image data from the previous step. The output of the block provides all the derivatives related to a single pixel location in parallel to the flow computation block. Assuming that the image data are already smoothed; the calculation of image derivatives can then be approximated by subtracting neighbouring intensity data in space and time. These operations can be performed simply by convolving the log-polar image with the masks: [-1 0 1] in case of the derivative I_η, and $[-1\ 0\ 1]^T$ in case of the derivative I_ξ. The temporal derivative I_t is the temporal difference between the current image and the precedent one. Similarly, the second order derivatives are obtained by convolving the first order ones with the same masks.

The final step is the computation of the optical flow vector:

$$\begin{bmatrix} u \\ v \end{bmatrix}=\begin{bmatrix} I_{\eta\eta} & I_{\xi\eta} \\ I_{\eta\xi} & I_{\xi\xi} \end{bmatrix}^{-1}-\begin{bmatrix} I_{t\eta} \\ I_{t\xi} \end{bmatrix} \quad (8)$$

B. Time-to-impact

After computing the optical flow, the time-to-impact computation is performed. In our application we benefit from the specific advantages of the log-polar transformation. Specifically, apart from the reduced data, the log-polar transformation separates optical flow into its expansion and rotation components. For an object centred in the visual field, a size increase is translation on one radial axis, whilst a rotation affects on the angular axis. However, in time-to-impact estimation we are only interested in the movement of an object towards the camera. This kind of motion is translated as an increase of the size of the projected object onto the camera plane. Consequently, we are interested only for the amount of the dilation of the image expressed from the radial component of the optical flow vector, i.e. the expansion.

The general expression of time-to-impact (T) derives from the eqn (9):

$$T=S/V \quad (9)$$

where V is the velocity of the moving object and S is the distance between the object and the camera. However, our aim is to estimate the time by using parameters that derive directly from the image and not from the trajectory information, such as the distance S. Thus, the equation 10 was adopted, that consists of the image velocity and its derivatives and not from the trajectory information [4]:

$$T = \left[\dot{\xi} \log a - \frac{\partial \dot{\xi}}{\partial \xi} + 2 \frac{\partial \dot{\eta}}{\partial \eta} \right]^{-1} \quad (10)$$

where $\dot{\xi} = d\xi / dt$, $\dot{\eta} = d\eta / dt$ are the velocity vector coordinates of a log-polar image sequence and a is the eccentricity (number of the rings) in log-polar mapping as illustrated in Figure 2. In the current implementation a = 45.

IV HARDWARE IMPLEMENTATION

The first part of the circuit is the log-polar transformation. The input is a gray-scale Cartesian image of 120x120 pixels, whereas the output is a gray-scale log-polar one, of 45x60-pixels resolution. The data flow is serial and consists of the values of the intensity in each pixel. The several units of this part as illustrated in Figure 4 and are as follows:

- **RAM_input_image:** A RAM of 16384 cells and 8-bit word length, that collects the data input.
- **ALU:** Arithmetic logical unit that performs the mapping of the log-polar transformation. This unit adds the values of the intensity data of the input image pixels that correspond to the same cell of the RAM_log_image and next it divides the sum with the number of those pixels determined from ROM_plithos (see bellow).
- **ROM_destination:** This is a look-up-table that actually performs the logpolar mapping. It consists of 16384 cells, each having a 12-bit word length. The ROM denotes the position that each pixel of the input image has in the log-polar image.
- **ROM_plithos:** This is a look-up-table of 16384 cells with 6-bit word length. It denotes the number of the pixels that correspond in the same cell of the RAM_log_image.
- **RAM_log_image:** A RAM of 4096 cells with 8-bit word length. It collects the data output (the intensities of the log-polar image).

The respective IO signals of the module are as follows:

- **Start:** A control signal to start of the process.
- **Data-in[7..0]:** The pixel-serial data input. The Cartesian image is padded into the unit pixel by pixel (8-bit resolution).
- **Clk:** The system clock.
- **Data-out[7..0]:** The pixel-serial data input. The Cartesian image is padded into the unit pixel by pixel

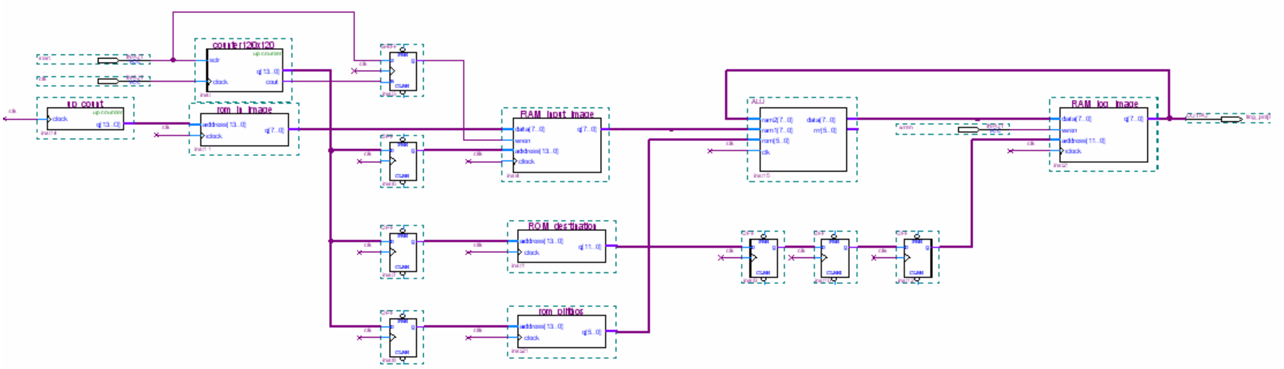

Fig. 4 The circuitry for the logpolar transformation

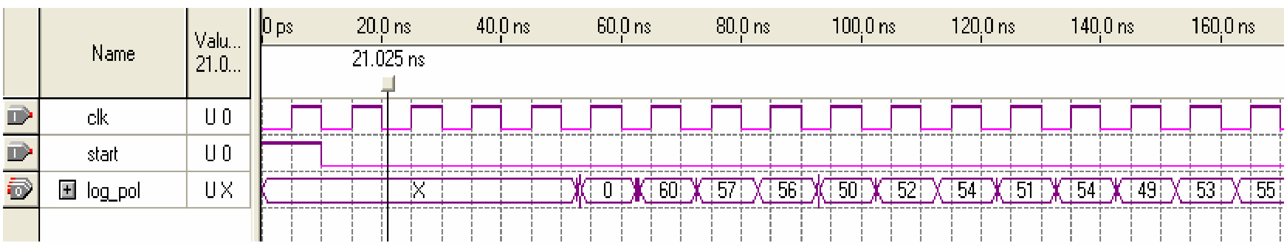

Fig. 5 The output data flow of the logpolar unit

(8-bit resolution).

The above output, which corresponds to the pixels of the cortical image, follows a serial data flow of 8-bit per cycle, as shown in Figure 5. This is collected in the RAM_log_image.

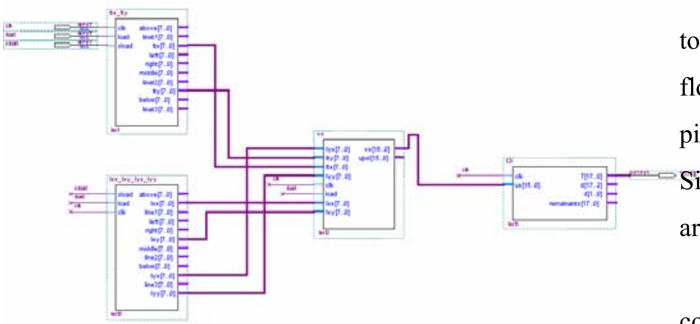

Fig. 6 Block diagram of the circuitry for the optical flow and time-to-impact computation

The second part of the circuit, which is illustrated in Figure 6, is the computation of the optical flow and the time-to-impact. First the smoothing operation is performed, by convolving the mask of the Gaussian filter, presented in section 3, throughout the whole image. In order to execute this operation the 4-conectivity neighbour pixels are preserved by means of register, as shown in Figure 7.

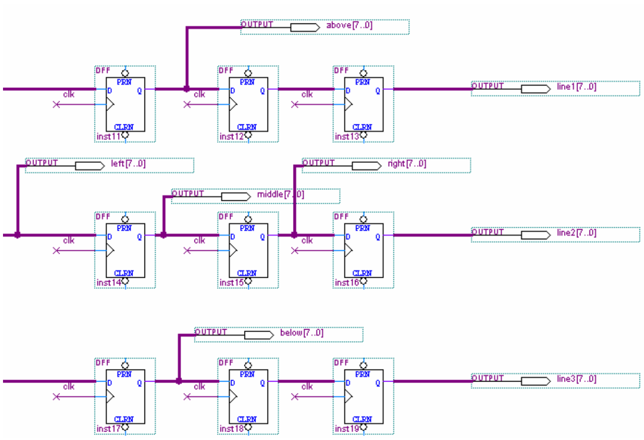

Fig. 7 Constructing the 4-conectivity neighborhood for the Gaussian filter operation.

The filter output is computed simply as:

$$I = \frac{4*middle[7..0]+above[7..0]+left[7..0]+right[7..0]+below[7..0]}{8}$$

where, multiplying by 4 is simply performed by shifting 2 places to the left and, similarly, dividing by 8 is done by shifting the result of the addition 3 places to the right.

The convolution with the masks 3x1 and 1x3 in order to produce I_x, I_y and I_t is also performed by means of flip-flops and by adding the complement of the corresponding pixels (wherever the parameter of the mask is -1). Similarly, with the same operation the second derivatives are produced (I_{xx}, I_{yx}, I_{xy}, I_{yy}, I_{tx}, I_{ty}).

In order to compute the optical flow, the *u* parameter is computed from equation 9. This is the only parameter needed for the estimation of the time-to-impact in the log-polar image plane, i.e.:

$$u = \frac{I_{yx} I_{ty} - I_{tx} I_{yy}}{I_{xx} I_{yy} - I_{yx} I_{xy}} \quad (11)$$

This is implemented easily by means of four multipliers, two subtracters and a divider.

The final block of the implementation is the time-to-impact computation. In our application the camera remains still without moving or rotating in its environment, i.e. without performing egomotions. This simplifies the implementation, since there is only one motion, i.e. the one of the object towards to camera, which is represented in the radial axis and depends only on the parameter u of image velocity. The simplified version of equation (10) is:

$$T = \frac{S}{V} = \frac{1}{\xi' \log_e a} \quad (10)$$

Therefore, the total circuit contains three inputs and an output as follows:

- **load** and **sload**: These are two control input signals, responsible for the image loaded to the registers.

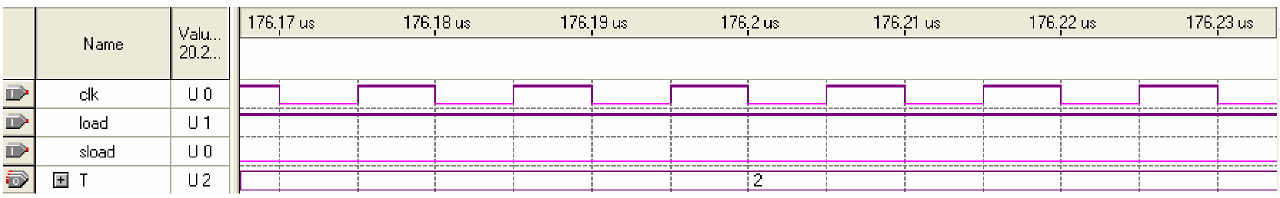

Fig. 8 Timing diagram of the time-to-impact estimation

- **Clk**: The system clock
- **T**: The estimated time-to-impact output of the circuit.

The function of the whole structure is the estimation of the time-to-impact of a moving object towards to the camera. A timing diagram expressing this estimation is illustrated in Figure 8. The simulation was performed in Altera Quartus II.

V CONCLUSIONS

The implementation of a dedicated hardware structure for time-to-impact estimation, using images from any common imaging sensor, has been presented in this paper. The Cartesian image was first transformed into a log-polar one. The main advantages obtained by utilizing the log-polar transformation are due to its space-variant sampling structure, which provides a variable resolution, though dense at the fixation point. Related to this specific structure, the volume data to be processed is considerably reduced, but without losing the high resolution in the part of the image corresponding to the focus of attention, where it is more necessary. The combination of log-polar transformation and the differential algorithm for the optical flow computation in addition with the pipelining technique produce direct, quick and economical estimation of time-to-impact. The maximum delay time of the circuitry is 25 μsec. The frequency of the system clock is 40 MHz. The total number of the clock cycles is 252, which are performed in 6,3 μsec. The circuit allows the processing of 58 fps, which permits real-time response for almost any commercial camera. The proposed system is intended to be used in navigation applications for both manned and unmanned vehicles as an extra option for risk management. Another target application is the mobile robot navigation for obstacle avoidance and impact prevention.

REFERENCES

[1] J.L. Barron, D.J. Fleet and S. Beauchemin, “Performance of optical flow techniques”, International Journal of Computer Vision (1994), 12(1):43-77.

[2] D.J. Fleet, “Measurement of image velocity”, Kluver Academic Publishers, Boston, 1992.

[3] P. Questa and G. Sandini, “Time to contact computation with a space-variant retina-line c-mos sensor”, in Proceedings of the International Conference on Intelligent Robots and Systems, Osaka, Japan, 1996.

[4] M. Tistarelli and G. Sandini. “On the advantages of polar and log-polar mapping for direct estimation of time-to-impact from optical flow”, IEEE Trans. on Pattern Analysis and Machine Intelligence (1993), 15(4):401-410.

[5] A. Khamene and S. Negahdaripour, “Motion and structure from multiple cues; image motion, shading flow, and stereo disparity”, Computer Vision and Image Understanding (2003), 90 (1): 99-127.

[6] I.S. McQuirk, B.K.P. Horn, H.S. Lee and J.L., Wyatt, “Estimating the focus of expansion in analog VLSI”, International Journal of Computer Vision (1998), 28(3):261-277.

[7] G. Metta , A. Gasteratos and G. Sandini, “Learning to track colored objects with log-polar vision”, Mechatronics (2004), 14(9):989-1006.

[8] G. Sandini and V. Tagliasco, “An anthropomorphic retina-like structure for scene analysis”, Computer Vision, Graphics and Image Processing (1980); 14 (3):365-372.