High-Speed Log-Polar Time to Crash Calculation of Mobile Vehicles[*]

Fernando Pardo Francisco Micó José A. Boluda Inmaculada Coma
Departamento de Informática - Universitat de València
Avda. Vicente Andrés Estellés s/n, 46100 Burjassot, Valencia, SPAIN
Fernando.Pardo@uv.es, Francisco.M.Mico@uv.es, Jose.A.Boluda@uv.es, Inmaculada.Coma@uv.es

Abstract

Time to impact computation is one of the applications of the image optical flow. It is useful in vehicle crash detection or robotic navigation. A high-speed image acquisition and computation rate is necessary in most of these applications. The main problem of time to impact computation from optical flow is the accuracy; accurate results usually need complex and slow computations not affordable for the fast time reaction of a vehicle or robot. Using log-polar images reduces the amount of data to be processed thus increasing speed and accuracy of results. Most optical flow techniques are inadequate to obtain high rate time to impact measurements, since they rely on complex static calculations on few images of the sequence. In our approach the time to impact is calculated using simple but fast algorithms over a high amount of images to have accurate results.

1 Introduction

It is possible to calculate a vehicle time to crash using the time to impact algorithm. This algorithm uses as input the image sequence of the vehicle front view. The time to impact algorithm generates a map of time to contact for every pixel of every image of the sequence. This map is useful to detect objects, collision hazards, obstacle avoidance, etc. It also may be employed in 3D scene reconstruction.

The simplest time to impact algorithm just takes few pixels and at least two images to calculate spatial and temporal derivatives. This simple algorithm can be computed at very high speed, though precision is not very high. This lack of precision can be reduced increasing the image acquisition speed.

With normal Cartesian images, high-speed image acquisition produces amounts of data to be processed in constrained time intervals, requiring high performance computing systems. These systems are rare in autonomous vehicles where space, weight and power consumption are limited. There must be a way to accelerate the time to impact computation without reducing the image acquisition speed and quality [1].

One of the ways of reducing image data is by the use of foveated images, where the interesting part of the image (usually the center) has more resolution than other less interesting areas (periphery). One foveated approach, with interesting mathematical properties, is the log-polar mapping. This log-polar pixel distribution resembles the human eye where photoreceptors are concentrated at the retina center or fovea, while still there is enough resolution at the periphery.

The log-polar mapping also has interesting mathematical properties specially suited for the time to impact computation problem. Since the vehicle movement on a road usually follows the camera optical axis, as in our case study, the optical flow computation in log-polar coordinates simplifies a lot, since only the radial component of the optical flow must be computed. The logarithmic part of the mapping is also useful because it simplifies the expression of the time to impact from optical flow, since time to impact in log-polar coordinates does not depend on the radius.

2 Log-polar mapping

The log-polar mapping is one of the possibilities of having a space variant image acquisition system [2]. The focal plane of this camera has two areas with analogous names as the human eye: The retina is the outer part and occupies most of the sensor area, in this part the distance of pixels to the sensor center increases exponentially thus decreasing pixel resolution toward the periphery. The fovea is the small central part with the highest resolution; it follows the same polar pixel distribution though pixel distance to the center increases linearly instead of exponentially. This log-polar transformation can be shown in figure 1 where the circle represents the focal or retinal plane, and the Cartesian represents

[*] This research work has been supported by the Generalitat Valenciana project ref. GV99-116-1-14

the computation or cortical plane. The computation plane is the transformation of the focal plane to the computer memory; this computation plane is what the computer *sees* of the environment.

The use of this kind of log-polar images reduces the amount of data to be processed allowing higher image processing rates. Selective data reduction is one of the most interesting properties of this mapping, but its special polar structure has also interesting mathematical properties as the rotation invariance. Also the exponential grow of pixel radius have interesting mathematical properties that simplifies some calculations, specially the optical flow and time to impact [3].

An approaching object following the optical axis of the retinal sensor will experiment an apparent scaling and its progress will be viewed as an expansion motion. Consequently, its movement can be represented by a radial optical flow from the center of the sensor towards its edges. Calculations of such a movement are simplified by means of a log-polar representation. The following equation shows the transformation from Cartesian to log-polar coordinates:

$$\begin{cases} \xi = \log r \\ \gamma = \theta \end{cases} \tag{1}$$

Scalings as a result of approaching objects become simple translations as shown in figure 1, where a ring going to the camera has been represented.

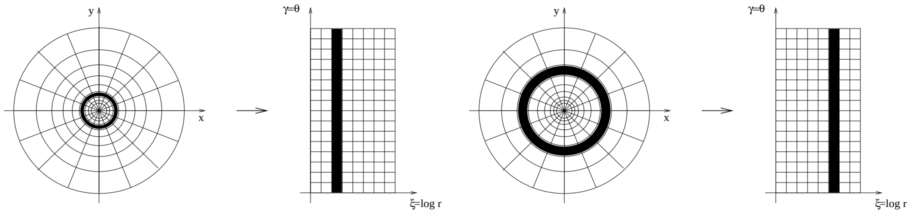

Figure 1: Approaching object in a log-polar mapping.

The simplest way of working with this kind of images is to perform a soft conversion from Cartesian (standard cameras) to Log-polar. This conversion consumes processing time though it can be performed at very high speed. Other approaches consist of directly use of a log-polar camera that include a log-polar sensor [4]. In this last case is when more speed and higher resolution can be achieved.

3 Time to impact computation in log-polar coordinates

It is possible to establish a relationship between the speed of an approaching object and its apparent sensor radial speed, since objects approaching to a camera produce a radial optical flow in the image plane [5]. Let us suppose a P point that is coming to the objective of the sensor in a direction parallel to its optical axis, with speed $W(t)$ as shown in figure 2.

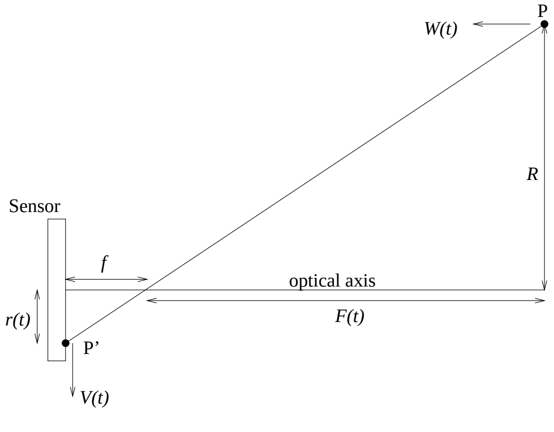

Figure 2: Object approaching the camera.

Being f the objective focal, R the distance from P to the optical axis (and therefore constant), $F(t)$ the distance from the projection point of P on the optical axis to the objective focus, and lastly, $r(t)$ the distance from the optical axis to P', the image of P on the sensor, then the following simple relationship is accomplished:

$$\frac{f}{r(t)} = \frac{F(t)}{R} \tag{2}$$

It is also straightforward to conclude that $V(t)$ (radial speed of P') is related to $W(t)$ (P object speed) by:

$$V(t)\frac{f}{r^2(t)} = \frac{W(t)}{R} \tag{3}$$

Combining these two equations, it is possible to calculate the *time to impact* τ; it is to say, the time the object will invest to collide with the focus of the system:

$$\tau = \frac{F(t)}{W(t)} = \frac{r(t)}{V(t)} \tag{4}$$

Thus, the time to impact can be expressed as the relation between the distance of P' to the center of the sensor and the radial component of the speed in the image plane. As a consequence, it is possible to trace an impact time map, simply dividing the radio of each image point between the optical flow in that point.

When these equations are translated to the log-polar plane, equation (4) becomes even simpler since the term $r(t)$ disappears (cancelled by the coordinate change in $V(t)$).

The $V(t)$ magnitude is the *optical flow* and there are several methods for its computation [6]. In Cartesian coordinates it becomes a difficult task since the speed $V(t)$ has two components V_x and V_y. In log-polar coordinates it also has two components (V_ξ,V_γ), but as far as we are just focusing the problem of approaching objects, the polar optical flow component V_γ is always cero. Having just one component to calculate simplifies the problem since the Horn equation [7] can be directly employed to calculate the optical flow and its inverse, that gives the time to impact as shown in the following equation:

$$\tau = -\frac{1}{B}\frac{\frac{\partial I}{\partial \xi}}{\frac{\partial I}{\partial t}} = -\frac{1}{B}\frac{I_\xi}{I_t} \tag{5}$$

where B is a constant that depends on the sensor geometry, I is the image intensities (image), $\frac{\partial I}{\partial \xi}$ is the image derivative along the ξ axis (spatial derivative) and $\frac{\partial I}{\partial t}$ is the image derivative with respect to the time (time derivative). These two derivatives are the only calculations to be made on the image, and they are as simple as subtractions. The use of log-polar coordinates and its application to the special problem treated here, allow this great simplification respect to the same problem solved in Cartesian coordinates.

4 Time to Impact practical implementation

The formula for the time to impact computation is simple as it has been shown in the last section. Nevertheless it is impractical to directly implement equation (5) for obtaining accurate time to impact from real moving images. The reason is that most of the main working constraints of the algorithm are not accomplished.

There are two conditions that an image sequence must follow for the time to impact algorithm to properly work. First, the two dimensional function, that describes the image grey level, must be continuous and differentiable at any real point. Second, the grey level evolution as a function of time must be also continuous and differentiable. Since image acquisition is clearly a discrete task, the grey level function is not continuous or differentiable, neither with respect to time nor space.

The solution usually adopted is to suppose that the image grey level function is continuous in time and space. This assumption is of course false and it is the main reason for the inaccuracy obtained by these algorithms. Despite the error, this approach is still useful and may be employed in most image processing algorithms, though it is necessary to perform some previous processing for converting the discrete grey level function in a continuous function to accomplish with the algorithm requirements.

4.1 Derivative constancy and spatial-temporal smoothing

The most common method for making an image function differentiable is probably to perform an image smoothing. This method is especially interesting for the time to impact, or optical flow, computation. The reason is that the algorithm can be applied only on those parts where there is some grey level gradient. Objects usually do not have any gradient in real scenes, since most objects have the same colour or reflectivity. The larger gradients are usually present at object borders where there is a sharp discontinuity. Precisely in these points, where the gradient is large enough, is where the algorithm conditions are not met due to sharpness of the grey level function. Performing a smoothing of these points converts these sharp changes in a smooth gradient very suitable for differential computations. Smoothing is therefore necessary for differential methods to work.

A simple spatial smoothing can be obtained applying a simple 3x3 weighed mask. This simple mask gives good results and it is not much computational demanding. Along with this spatial smoothing it is also interesting to perform a temporal smoothing with the same goal: to have the grey levels as a continuous differentiable function of time. In this case we can consider that the grey level function has tree coordinates and the mask should be applied to a tree dimensional cube formed by the 3x3 planes of the three images. Thus, in the case of a spatial-temporal smoothing at least a 3x3x3 convolution mask should be applied to every point in every image of the sequence. Including the temporal smoothing increases three times the calculus to be performed, while the obtained precision increasing is not so high.

Smoothing makes functions more differentiable and also removes some image noise, but it is not enough to obtain good time to impact calculations. Other problem that arises in real scenes is the grey level gradient constancy. For the algorithm to work it is necessary that the grey level gradient or slope is constant from one image to the next; only in those parts of the image where the spatial gradient is constant, the calculated time to impact will be reliable. For this reason it is interesting to have the grey level function differentiable and also to have a constant gradient present in relatively large areas of the image. Smoothing helps on having constant gradient on local areas of the image.

4.2 Taking advantage of high image rate

Most image acquisition systems usually yield up to 25 images per second. This high rate is often more than enough for movement detection and reaction. In fact, it is very difficult to extract movement from images that are very close in time. This is one of the reasons why many times only few images per second are really employed for image processing and movement analysis. Also, the high amount of data produced by normal Cartesian images makes it difficult to process such a high image rate.

Using log-polar images reduces the amount of data to be processed allowing processing of 25 images per second or even more. The problem at these rates is that the differences from one image to the next is very small and it is not possible to extract any movement information; only in fast movements there will be a significant difference among images. Thus, if we need images separated in time to extract movement, why using image high rates? There are at least two reasons for using image high rates: having more images per second allows the system to calculate more accurate results, and the algorithms can be adapted to detect fast and slow movements at the same time.

As stated before it is necessary to wait some time between two images to calculate the time to impact. This time between two images depends on the movement that we want to measure; if the movement in front of the camera is fast, the time among images must be short, while if the movement is slow this time must be large enough to detect differences among images.

To have a time to impact calculation rate equal to the image acquisition rate, we have calculated the time to impact using images of the sequence that are distanced enough in time. Figure 3 shows the image sequence and the images taken to calculate the time to impact. In this way it is possible to calculate the time to impact keeping the original image rate.

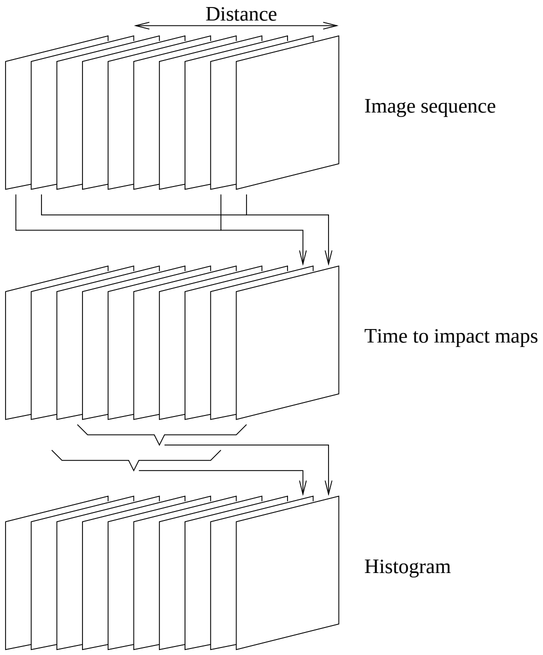

Figure 3: Algorithm implementation diagram.

The sequence of time to impacts obtained in this way is not very accurate since just using two images to calculate the time to impact is not enough. We use several images to obtain a statistical more accurate time to impact map just calculating the medium value over several images. It is possible to do it this way since the image rate is fastest than the movement to be analysed.

Finally it is interesting to create a histogram of the calculated time to impacts to detect the medium, or most probable, value that is the time to impact of the main object in the scene.

The procedure explained here depends on the speed of the object to be studied. Experiments show that, depending on the distance between the two images used to calculate the time to impact, some movements are detected better than others. The algorithm works like a movement band-pass filter, where the distance between the two images for the time to impact computation marks the center of the band. If movements fall into this band it will be possible to detect them, otherwise we need a mechanism to enlarge the band of the algorithm.

An algorithm has been implemented to enlarge the band of the time to impact calculation. It consists of an accumulative calculation of the time to impact map for several differences between the two images employed for the time to impact computation. For example, figure 3 shows the $Distance$ between the two images employed for the time to impact computation. In the implemented algorithm this distance is in fact the maximum distance taken, since first it starts at a distance of just one image, then two, and so on. Merging all time to impact mappings obtained for every distance gives a time to impact map where fast and slow movements are included.

Some experiments show that it is better not to use every distance to calculate the time to impact; it is better to use higher jumps (5-10) for every distance. For example, it is better to calculate the time to impact maps at distances of 1, 6, 11..., than 1, 2, 3..., since it gives better results and takes less time to execute for the same band-width.

4.3 Other implementation issues

The time to impact algorithm is very sensible to noise and other inaccuracy sources. To avoid wrong and erroneous time to impact calculations it is interesting to discard those results that are probably wrong.

One way of discarding probably erroneous results is to calculate the statistical relative error of those results from equation (5):

$$\frac{\Delta\tau}{\tau} = \frac{1}{B}\left(\frac{\Delta I_\xi}{|I_\xi|} + \frac{\Delta I_t}{|I_t|}\right) \tag{6}$$

where $\Delta\tau$, ΔI_ξ and ΔI_t are the absolute errors of the time to impact, spatial derivative and temporal derivative respectivelly. It is possible to assume that ΔI_ξ and ΔI_t are around one due to the discrete nature of these quantities. Therefore, to keep the time to impact error to the minimum we should discard those results where I_ξ or I_t are small. We would like to choose a large value for I_ξ and I_t, but this is usually not possible and discarding lower values means also to cut detection of small time to impacts that, by the way, are the most dangerous.

Experiments show that it is interesting to discard those spatial derivatives below 10 or 15. The threshold for the time derivative does not need to be so high, since it is located at the denominator of the time to impact expression. If the time derivative is very small it will yield a high time to impact, but this is not a real problem since abnormally high time to impacts are not critical. For this reason, the time derivative may have a threshold as small as 3 or 4, or even lower in situations where there is a need of higher number of time to impact points.

Another important issue when detecting reliable data to calculate the time to impact is the spatial gradient constancy. The time to impact works if the gradient is constant over a local region but also among images. It is not a good idea to calculate and use the temporal derivative between two images if the spatial gradient differs at that point between the images. For this reason we have also discarded those points where there were a difference, larger than a certain threshold, between the spatial gradients of the two images employed to calculate the time derivative.

5 Experimental results

All issues presented at previous section have been implemented in a program that calculates the time to impact maps and histograms from sequences of log-polar images. Several experiments have been carried out for sequences taken at a rate of 10 images per second.

A camera has been mounted in a mobile platform having a constant rectilinear movement approximately along the camera optical axis. The speed of the vehicle is around 5 cm/s. Two experiments where performed, in the first we put an obstacle at 150 cm., while in the second, we left the same obstacle and put another one at approximately half that distance. Since the image rate is 10 images per second, it takes around 300 images to impact the last obstacle.

The first experiment results are shown in figure 4. There are 300 images for the total sequence; the four images shown in the figure are the number 100, 150, 200 and 250 from left to right. The first row of the figure represents the time to impact histogram for each elected moment of the sequence. The vertical line represents the real time to impact measured from the camera to the object. The second row is the original log-polar image. The third row is the transformation of the

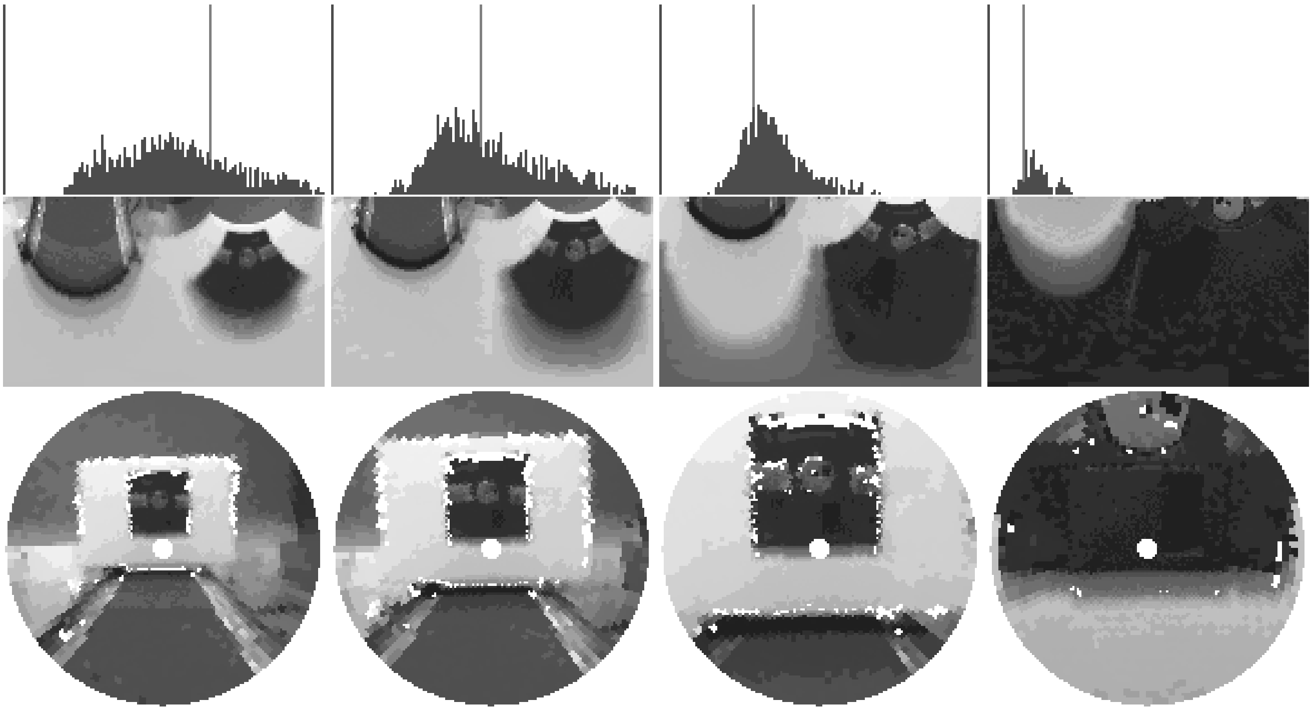

Figure 4: First experiment: First row (top) represents the time to impact histogram of each image. Second row (middle) represents the log-polar images taken by the camera. Third row (bottom) shows the standard representation of the log-polar images.

log-polar image to the Cartesian standard representation; in these images we can see a diskette attached to a white box forming the object to impact. The white points in the Cartesian images show the places where "reliable" time to impact has been calculated.

It is possible to extract some conclusions from the histograms of figure 4. The first histogram (left) corresponds to the moment where the object is still far from the camera. This histogram shows a kind of gaussian with a center different than the real value expected; this is not very strange since large time to impacts are difficult to calculate. The gaussian width for this first histogram is also large indicating a not very reliable medium value. The second histogram shows a better result, since the gaussian most probable value is very close to the real value. Finally, in the third histogram, the top of the gaussian perfectly matches the real time to impact. In the forth and last histogram the matching is also complete being the width of the gaussian small. These last two histograms show that the time to impact calculation is more accurate when the camera is close to the impact instant. It is worth noting that at the third image there are still 100 images before impact and at the last one there are still 50 images to impact.

The second experiment results are shown in figure 5. The difference with the first experiment is that we added an extra obstacle between the final obstacle and the camera. This new obstacle is only visible at the left of the first image shown in the figure. This obstacle is not present at the other images. The images shown in the figure are taken at similar intervals as in the first experiment.

The first histogram shows the time to impact for the first object, but it is shifted because of the influence of the second far object. In the second histogram there is also a mismatch and now is the near object which changes the calculated time to impact; this first object is not anymore in front of the camera, but the time to impact employs past images for the algorithm and also there is a medium value over a range of images, this is why the gaussian center comes before the real time to impact. At the third histogram the gaussian center matches the real value, and at fourth there is also a perfect matching though the number of reliable points is not very high.

6 Conclusions

We have presented a fast algorithm for the time to impact computation. The algorithm itself is not new since it is based in a very well known differential method, though our implementation present specific characteristics that make it interesting for high-speed time to impact computation. One of these implementation issues is the use of different spaced image of the sequence to obtain a better time to impact result, since depending on the time difference among images the algorithm

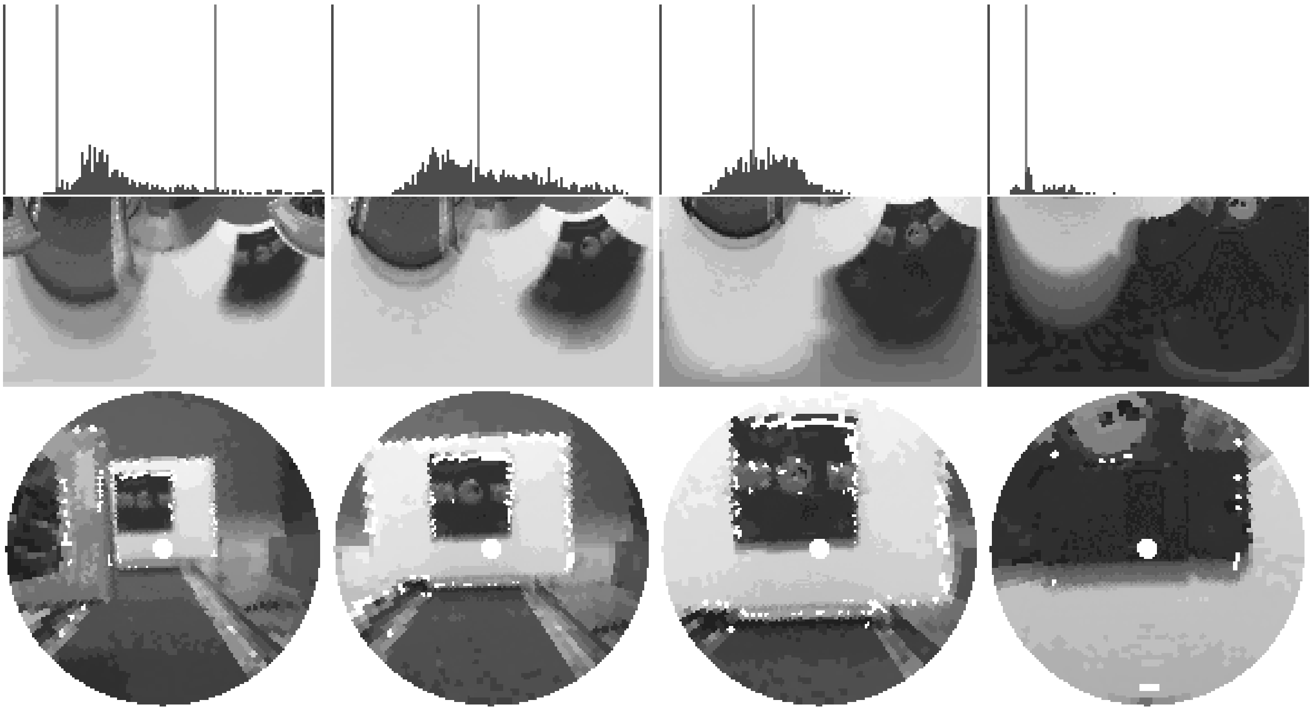

Figure 5: Second experiment: Two obstacles are shown at the first image (left).

acts like a filter where some speeds are discarded; having different measures for different time intervals and merging them together produce more reliable results at different speeds. We also have employed log-polar images to achieve real-time constraints and algorithm simplifications. Image smoothing has shown also to be of special importance when implementing differential algorithms. In our case, we have noticed better results when this smoothing is performed in three dimensions: two for the space and one for the time.

References

[1] P. Questa and G. Sandini. Time to contact computation with a space-variant retina-like CMOS sensor. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'96,* Osaka, Japan, November 1996.

[2] A.S. Rojer and E.L. Schwartz. Design considerations for a space variant visual sensor with complex logarithmic geometry. In *Proc. Int. Conf. on Pattern Recognition,* Philadelphia, PA, 1990.

[3] 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 PAMI,* PAMI-15, No. 4:401–410, 1991.

[4] F. Pardo, B. Dierickx, and D. Scheffer. Space-variant non-orthogonal structure CMOS image sensor design. *IEEE Journal of Solid State Circuits,* 33(6):842–849, June 1998.

[5] F. Pardo, I. Llorens, F. Micó, and J.A. Boluda. Space variant vision and pipelined architecture for time to impact computation. In *International Workshop on Computer Architectures for Machine Perception, CAMP 2000,* Padova, Italy, September 2000.

[6] K. Daniilidis. Optical flow computation in the log-polar plane. In *Int. Conf. on Computer analysis of images and patterns, CAIP'95,* pages 65–72, 1995.

[7] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. Technical Report RPL-TR-9107, Dpto. of Computing and Information Science, Queen's University, Kingston, Ontario, July 1993.