# High-speed movement analysis from log-polar images using dynamic feature extraction and correlation

Esther de Ves, Fernando Pardo, Jose A. Boluda

Departamento de Informática - Universidad de Valencia

Avda. Vicente Andrés Estellés s/n, 46950, Burjassot, Valencia, SPAIN

Esther.DeVes@uv.es, Fernando.Pardo@uv.es, Jose.A.Boluda@uv.es

## Abstract

There are several methods to measure movement in front of a mobile vehicle (robot) equipped with a camera. Some methods detect movement from the analysis of the optical flow, while other methods detect movement from the displacement of objects or part of the objects (corners, edges, etc.) Those methods based on the optical flow are suitable for high speed analysis (say 25 images per second) but they are not very accurate and treat the image as a whole, being it difficult to separate different objects in the scene. Those methods based on image feature extraction are good for object recognition and clustering, that can be more precise than other methods, but they usually require lot of computations to yield a result, making it difficult to implement these methods in a navigation system of a robot or mobile vehicle. In this article we present a technique that allows high-speed movement analysis using the accurate displacement measure of the feature extraction and correlation method.

## 1. Introduction

Most of 2D feature detectors employed in image processing are computationally expensive and are not suitable for high speed image analysis (25 frames/s). It is possible to increase the computation speed by loosing accuracy or employing powerful computer systems that are not adequate for mobile robots. There is a different approach to successfully employ this feature extraction on mobile robots, and it is to decrease the global visual data to be processed. This reduction is accomplished by means of the log-polar mapping that concentrates pixels in the image center (the most interesting part) and decreases resolution toward the periphery (the least interesting part) [1]. The log-polar mapping has the advantage of selective image data reduction, but also has some mathematical invariance properties (scaling and rotation) that are especially useful for image processing, particularly for a robot moving ahead.

Figure 1 shows the transformation of the scaling of an approaching object into a linear displacement. The focal plane (camera) and the computational plane (array in the computer memory) are shown in this figure. The original object (black ring) is a centered ring in the focal plane, but it is converted to a straight line in the computational plane after the log-polar transformation. The scaling produced by the camera approaching the ring is converted in just a displacement in one of the orthogonal axis. This interesting property can be exploited to simplify computations of such approaching movements, commonly found in the movement of robots toward an objective.

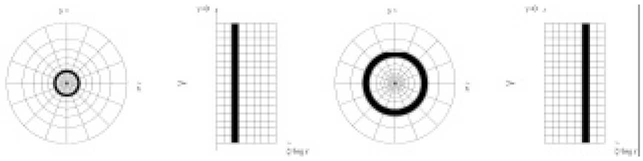

***Fig. 1.*** *Camera approaching to an object (black ring) in log-polar coordinates*

The log-polar transformation has also the advantage of the selective reduction of information. The special log-polar pixel distribution has more resolution in the interesting parts of the scene (center) reducing the number of pixels toward the periphery; the view field is kept while the total pixel count is reduced.

We use a resolution of 76 rings by 128 pixels per ring in our experiments. This makes a total of 9728 pixels to be processed. Comparing this data (roughly 10 K), with the data of a standard 512x512 Cartesian image (256 K) the difference is quite significant. The difference in the number of pixels to be processed has a direct impact on the rates at which the images can be processed. Depending on the image analysis, this save of time can be of several orders of magnitude while the precision is still kept at acceptable values.

## 2. Feature extraction

Not many feature extraction methods can be employed in the log-polar domain due to the special mathematic characteristics of this mapping. One of them is the distortion suffered by objects after transformation (the shape of any object is not constant and depends on its position in the log-polar plane). But there is a characteristic of the log-polar mapping that makes it possible to employ object detection methods: since the log-polar is a conformal transform, angles in Cartesian coordinates are preserved in the log-polar coordinate system. We therefore employed a feature extraction based on corner and junction detection that allows the measurement of relevant point movement and object detection despite the fact that the object changes its shape as it moves (or the robot moves).

We have chosen a 2D gray-level detector based on a statistical analysis of the gradient orientations in a circular neighborhood of the point considered as a possible 2D feature, since this method is not computationally expensive and is more robust, especially for the log-polar mapping, than other corner detectors.

According to this approach [2], a corner point *p* can be defined as a point in the image whose gradient is not null and for which the orientations of the edges that converge in it are grouped around two (or more) different modes. Thus, it is proposed the hypothesis that a corner, where *n* edges converge, can be modeled as a mixture of *n* von Mises distributions as follows [3]:

$$h(q) = \sum_{i=1}^{n} \frac{p_i}{2\Pi I_0(k_i)} \exp(k_i \cos(q - m_i)) \qquad (1)$$

where ì$_i$ and $k_i$ are defined for each population and $p_i$ is the mixing proportion of the *ith* population. Therefore, the method to detect 2D features is to study the distribution of the orientations and to test the null hypothesis: the hypothesis that the distribution of orientations constitutes a mixture of two von Mises distributions. If the hypothesis can not be rejected, the point is assumed to be a corner where two edges converge. The test used for this purpose is the Watson-Stephens test.

Two important remarks about the method: the neighborhood analyzed in each point should be large enough so as to contain a sufficient number of points for the test to be applied and non-maxima suppression is done to keep only those points which have the highest significance of its corresponding hypothesis in a fixed neighborhood. This is done to avoid duplications of the same corner point.

The selection of this interest point detector is due to its simplicity (it is easier to implement than others in literature) and it is quite insensitive to contrast.

## 3. Object tracking

There are several methods to measure motion in front of a mobile vehicle (robot) equipped with a camera. Some methods are based on computing the optical flow whereas others detect motion based on the displacement of some objects or part of the objects.

As we explained before, in this work we have adopted the second approach for motion analysis: tracking some objects in the scene. The objects to track are a set of 2D features extracted in each single image of the sequence using the corner detector described above.

This point detector procedure is followed by a matching process, which looks for correspondences between these interest points. Assuming that interest points have been located in all images of a sequence, the correspondence between points in consecutive images can be found by using the assumption of maximum velocity. This assumption implies that a point in the image will correspond to the closest point in the next image.

Using this simple idea we have constructed an algorithm to track corner points in the scene: for each point of a frame we find the closest point in next image.

It is possible that new points appear and others disappear; a new point appears when there is not any point close enough in previous images and a point disappears when there is not a near point in next frame. Our algorithm is able to manage these events using information from previous frames.

In order to avoid that several point are candidates to correspond with a given interest point, some parameters of the algorithms should be selected carefully: the maximum velocity parameter and the size of the neighbourhood for maxima suppression of interest points (in the 2D point detector). The maximum velocity (in pixels) should be smaller than the size of the chosen neighbourhood.

Figure 2 shows four images of the sequence employed for the experiments. This sequence has a total of 280 images and corresponds to the camera approaching toward two objects located at difference distances. The closest object is around 150 images far and the latest is 280 images away; it takes the whole sequence to impact this last object. The log-polar computational plane is shown along the real Cartesian image.

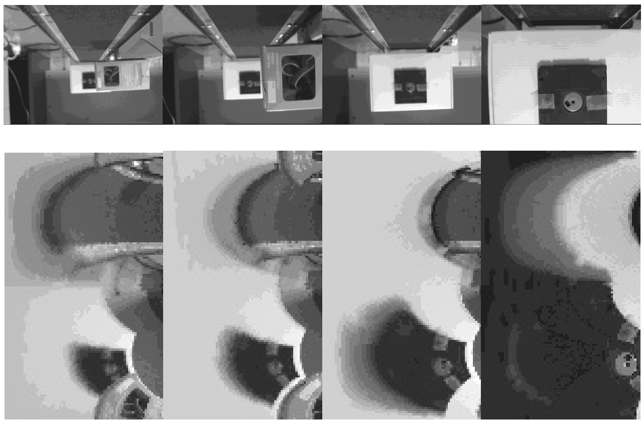

***Fig. 2.*** *Four images (30,100,170,140) of the total sequence in Cartesian (up) and log-polar (down) representations*

Figure 3 shows the plot of the trajectories of some detected corner points in an instance of the sequence of figure 2 using the tracking algorithm. The tracking of some corners are not correct due to mismatches obtained by the proposed algorithm. However, most of the points are tracked correctly and their trajectories are as expected (rectilinear movement from left to right).

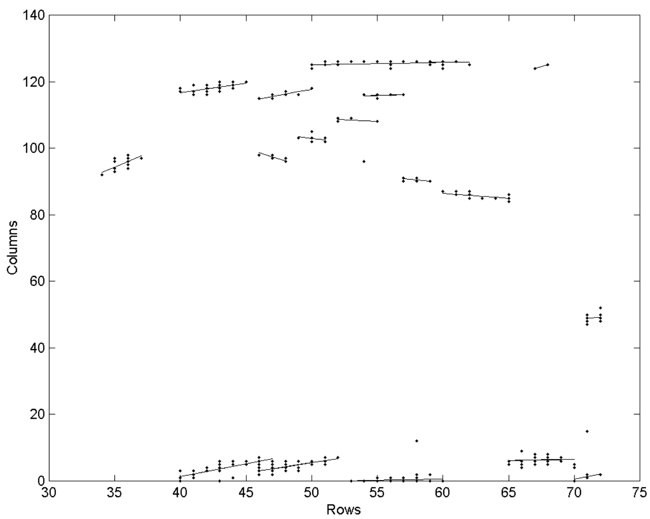


***Fig. 3.*** *Trajectories of the tracked points along the sequence in the log-polar domain*

## 4. Time to impact computation

The time to impact of an approaching object is the time required by an object to reach the camera. If the camera moves, this time to impact is the time employed by the camera to reach the first object in its trajectory. This time to impact can be calculated from the measurement of object speeds at the sensor plane. The use of log-polar coordinates simplifies this calculation as it is shown next.

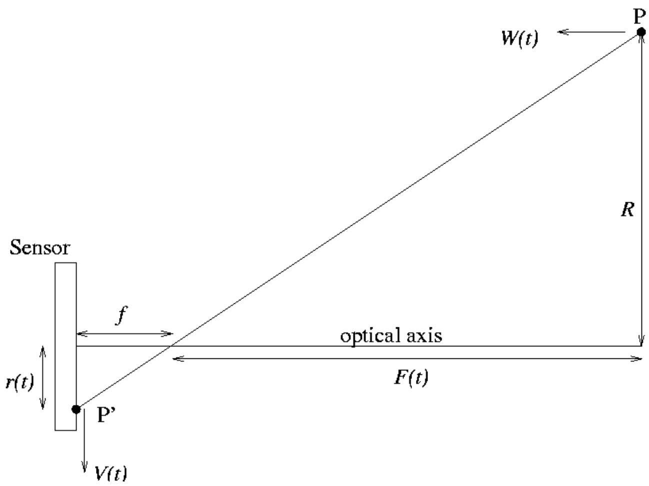


***Fig. 4.*** *Pin-hole model of an object approaching the camera*

Figure 4 shows the approaching object (P) projection on the camera sensor plane (P') following the pin-hole camera model. From this figure the following relation is straight forward:

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

where *F(t)* is the object (P) distance to the camera focus, *R* is the distance of the object to the camera optical axis, *f* is the camera focus distance and *r(t)* is the distance of the object projection (P') to the optical axis. *F(t)* and *r(t)* depend on the time since the object (or the camera) is moving. The speed at which the object P is approaching the camera is *W(t)*=d*F(t)*/d*t*, and the resulting speed of P' in the sensor plane is *V(t)*=d*r(t)*/d*t*. It is possible to obtain a relation between these two speeds deriving equation (2) with respect to time:

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

In the other hand, the time to impact ($t$) supposing a constant approaching speed *W(t)*, can be evaluated by:

$$t = -\frac{F(t)}{W(t)} \tag{4}$$

Now we want an expression for the time to impact that only takes into account camera or image parameters. For this objective we can take equation (2) to obtain an expression for *F(t)* and equation (3) to obtain an expression for *W(t)* and then substitute in equation (4) to obtain:

$$t = \frac{r(t)}{V(t)} \tag{5}$$

This is to say that the time to impact computation of an approaching object can be calculated as the division of the radius of the object projection (image) and the object projection speed. Both measurements can be obtained directly from image analysis.

Both magnitudes, *r(t)* and *V(t)*, may have any spatial orientation depending on the approaching object position. It means that we have to take into account the velocity components in *X* and *Y* directions supposing a Cartesian representation.

Now we can consider a camera with a log-polar sensor which transformation follows this equation:

$$r(t) = Ae^{Bx(t)} \tag{6}$$

where *x(t)* is one of the orthogonal components of the log-polar computational plane as seen on Figure 1. We can derive this expression to obtain the velocity in the log-polar domain:

$$V_x(t) = \frac{1}{B}\frac{V(t)}{r(t)} \tag{7}$$

Substituting this expression in equation (5) we finally obtain the equation for the time to impact computation in log-polar coordinates:

$$t = \frac{1}{B}\frac{1}{V_x(t)} \tag{8}$$

where B is the constant exponential growth factor of the log-polar transformation and $V_\xi(t)$ is the object projection speed in the radial direction measured in the log-polar computational plane.

There are two advantages of the log-polar transformation compared to the Cartesian representation [4]: first, there is no need of knowing the position of the object in the image, since it does not appear in the equation; and second, only one component of the velocity must be calculated. This last feature is especially important since it simplifies a lot the amount of calculus to be performed.

## 5. Time to impact results

The theory involving time to impact computation is very simple, but the real implementation of such a theory if not so simple [5]. The discrete nature of the images and the acquisition itself produce errors with similar magnitude as the parameters to be measured.

A large change in the image (say a displacement of several pixels) is necessary to obtain accurate speed measurement and thus time to impact. But such a displacement means that the object is moving too fast or the image rate is too low. In both cases there is no use of calculating the time to impact since it could crash immediately after measurement. Time to impact should be measured accurately with some time in advance.

It is necessary the use of statistical analysis over a large amount of images to obtain an accurate time to impact calculation long before the approaching object become dangerous.

We have employed two techniques to obtain accurate time to impact. In the first technique we calculate object speed over a large number of object positions obtained from different images of the sequence. This method consists on storing the last coordinates of an object over time and then calculating its speed just performing a least-squares linear function fitting. The second technique consists of just a calculation of the mean value of the object speeds to obtain a global time to impact for each image of the sequence.

The first technique has the problem of filtering: if the least-square fitting is performed over a large amount of images, the precision increases, but only for far objects with slow motion, since nearby objects move so fast that the speed is not constant and they could even impact before the calculation is done. In the other hand, if we take few images to calculate velocity, there are fewer points for the measure and we loose precision, and it is not possible to even detect movement if the object is far.

We calculate the speed in two passes to avoid the filter effect of fixing a certain number of images for the least-squares fitting. In the first pass we use the last few coordinates of the object, and in the second pass we use more points to detect long distance object motion.

Figure 5 represents the time to impact mean value at every image of the sequence of figure 2. Dotted points are these time to impact values, while straight lines represents the real time to impact of the two approaching objects of the scene.

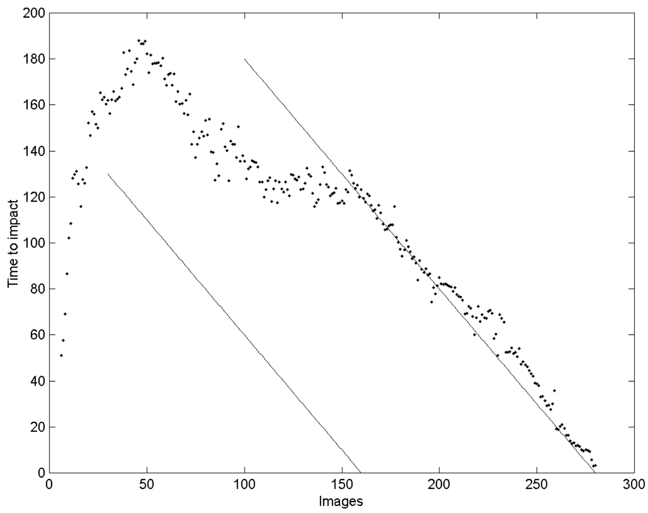


***Fig. 5.*** *Calculated time to impact for the whole image*

It is easy to see in this figure that the closest object theoretical time to impacts (left straight line) do not match at all the calculated empirical values, while the matching of the second object is almost total at the end, just when the other object disappeared from the scene.

The lack of matching of both objects while both are present in the scene is quite logical: the experimental values that we see in the figure correspond to the mean value between the time to impact of both objects, that is roughly the points between the two straight lines.

Just when the closest object disappears from the scene (after first impact) the matching between the calculated and theoretical times to impact is complete. This is because there is only one moving object. A better analysis consists of detecting the two mean values of the measured velocities.

First 50 calculated times to impacts are not correct since the algorithm needs these previous images to stabilize and to have enough points for statistical accurate calculations. Once the "pipeline" has started all subsequent points are correctly calculated as seen in the figure.

## 6. Conclusions

In this paper we have presented the feature extraction method as a high speed technique for movement analysis using log-polar images. Current processing speed is around 8 images per second in a Pentium III, 800 MHz personal computer. Higher image rates can be easily reached with more up to date computers. Also the algorithm can be further refined for higher speed computation.

Feature extraction performs well for object tracking but it needs additional statistical treatment for time to impact computation. The high image rates employed allows us to have large amounts of data to obtain statistical sound values for the time to impact and other dynamic parameters interesting for motion analysis.

## 7. Acknowledgements

This research has been supported by project TIC2001-3546 of the Spanish "Ministerio de Ciencia y Tecnología" partially funded by the European FEDER.

## References

[1] F. Pardo, B. Dierickx, D. Scheffer. "Space-Variant Non-Orthogonal Structure CMOS Image Sensor Design", *IEEE Journal of Solid-State Circuits*, Vol. 33(6), pp. 842-849, 1998.

[2] M.E. Díaz, J. Domingo, G. Ayala, "A grey-level 2D feature detector using circular statistics". *Patter Recognition Letters*, 18 (1997) 1083-1089.

[3] K. Mardia, "Statistics of Directional Data". *Academic Press*, New York. 1972.

[4] M. Tistarelli, 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*, Vol 15(4), pp. 401-410, 1991.

[5] F. Pardo, J.A. Boluda, I. Coma, "High speed log-polar time to crash calculation of mobile vehicles", *Proc. SERVICEROB 2001*. Santorini, Greece. June 2001.