Ellipse Detection in Very Noisy Environment

Pierre Zoppitelli
Pierre.Zoppitelli@univ-amu.fr

Sébastien Mavromatis
Sebastien.Mavromatis@univ-amu.fr

Jean Sequeira
Jean.Sequeira@univ-amu.fr

Aix-Marseille University
Polytech Sud - LSIS - Case 925
163, avenue de Luminy
13009 Marseille, France

ABSTRACT

Ellipse detection is a major issue of image analysis because circles are transformed into ellipses by projective transformations, and most of 3D scenes contain circles that are significant for understanding them (mechanical parts, man-made objects, interior decoration,). Several algorithms have been designed and published, some of them very recently, to characterize ellipses within images and they are very efficient in most classical situations. But they all fail in some specific cases that regularly happen as, for example, when the noise in the image is such as it produces dashed ellipses, or when they are very flat, or when only small parts of them are visible. We propose an algorithm that brings a solution in such cases, even if it is not more efficient than the other ones in classical situations. Hence, it can be used as a complement of other algorithms when we want to detect ellipses in a robust way, i.e. in all situations. This algorithm takes advantage of a property of ellipses related to their tangent lines, without any assumption on edge connectivity: primitives are designed to characterize the possibility for a point and an orientation to locally represent an edge; these primitives are not connected and their global analysis enables to obtain the center location and the three other parameters of ellipses that can be drawn through this set of primitives, i.e. that go through some of these points and that have the corresponding tangent lines. A set of tests has been used to measure its robustness.

Keywords
Ellipse detection, Hough Transform.

1 INTRODUCTION

Image understanding requires to extract relevant elements from images and to make them match with the corresponding potential items of a scene model; we also could say that it consists in extracting information from images and associating it with a knowledge representation. Knowledge can be related to color, texture, shape and many other factors, in relation to the problem we want to study. When considering shapes, we can notice that circles are often part of knowledge representation. For example, many road signs are circular, so that a lot of man-made devices that can be seen either inside (boxes, lampshades, clocks,) or outside (traffic circles, pipes,).

Circles are transformed into ellipses by homographies (projective transformations) and it is especially the case of images produces by video cameras. Hence, several ellipses are present in such images and characterizing them is crucial in the frame of robotics, video watching and UAV (Unmanned Aerial Vehicles) applications, for example.

2 STATE OF THE ART

Ellipse properties have been widely studied since the Greek Antiquity, with a lot of interesting results in the seventeenth to nineteenth centuries, especially when considering them in frame of the projective geometry using complex numbers. A lot of papers have been published during the last thirty years on ellipse detection in images, but this topic is particularly crucial and thus, several important papers have been published very recently on ellipse characterization. Let us describe the most recent and significant papers about ellipse detection. Three types of approaches enable to characterize an ellipse within an image, by making an ellipse evolve in the image under constraints, by using a Hough Transform [Dud72], or by linking edges or parts of ellipses.

Using the Hough Transform requires to describe and then to manage a homogeneous 5-dimension space (because an ellipse is defined by 5 parameters), and this is a very complex problem. An alternative solution, based on this idea, has been proposed in 2005 by Basca et al [Bas05]. Using an ellipse that evolves under constraints is a very classical idea, which is similar to an active contour approach: in 2013, Prasad et al published an improved version of this approach [Pra13]. Finally, some works as those described in Libuda et al [Lib07] propose a method that connects small edges until it produces an ellipse.

Fornaciari et al give an evaluation of these approaches and similar ones in [For14]. In this paper, they propose a new algorithm that improves the "connecting edges" approach by using a Hough Space in which they proceed to a parameter separation (in order to avoid a 5-dimensional space).

Other very recent works have been published (it shows that it is really important to find an efficient solution to this problem), and we mention the algorithm proposed by Lu [Lu15] and that uses a polygonal curve and a likelihood ratio test to stabilize the ellipse evolution.

At the end of 2016, Jia et al propose an original approach [Jia16], based on the "connecting edges" approach but with a major improvement that uses a very powerful property of projective space. In this paper, they compare the performance of this new algorithm with all those algorithms mentioned before and they demonstrate they obtain better results. Thus, we will give a short description of this algorithm in the next lines.

Jia et al use a projective property that is a generalization of the concept of cross-ratio on a polygon (in the case of the cross-ratio, the polygon is reduced to two points) and that enables to provide a Characteristic Number (CN) that is specific of the polygon because it is invariant by projective transformation; in the case of six points located on an ellipse, the CN is +1. In such a way, we can find very easily and very quickly if six points belong to an ellipse or not. They first look for edges (by using a Canny operator [Can86] and a Freeman encoding) and they assemble these edges when they belong to a same ellipse (by using this projective property). Although this algorithm is very efficient compared to the other ones, it fails in some situations that happen frequently. We identified at least, three types of situations in which it is the case: when points on the ellipse are sparsely located and thus do not enable to produce edges (e.g. a road sign partly hidden by branches or the end of pipe partly hidden by a grid), when having a very short arc of ellipse, and when the ellipse is very flat.

We propose an algorithm that takes in charges such situations and that find ellipses, even in very tough conditions. This algorithm is not proven to be better than other ones in classical situations, but it can be used in parallel (as a complement) in the frame of a more robust and reliable detection approach.

3 THE PROPOSED ALGORITHM

Let us show two images illustrating a case in which the algorithms mentioned before often fail.

Figure 1: The road sign is partly hidden by branches

Figure 2: The end of the cylindrical pipe is hidden by a grid

The main reason is that these algorithms require the detection of edges that have a minimum length, and it is not the case in these images. In order to overcome this problem, we decided not to use edges (as sets of connected points) but to base our algorithm on an ellipse local property that is integrated in a global scheme. Let us see in next subsection what is this property.

3.1 A property of ellipse tangent line

Let us consider P_1 and P_2, two points of an ellipse and their corresponding tangent lines T_1 and T_2 in P_1 and P_2; let us also consider I_{12} the middle of $[P_1P_2]$ and J_{12} the intersection point of T_1 and T_2; then, I_{12}, J_{12} and C (center of the ellipse) are on the same line D_{12}.

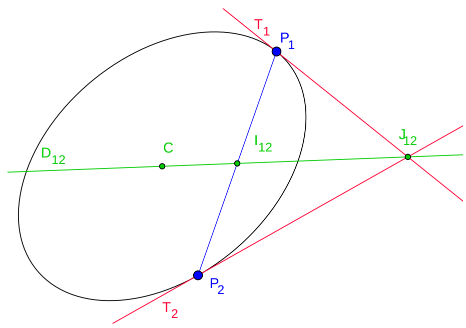

Figure 3: C, I_{12} and J_{12} are on the same line

This property enables to skip from one point to another on the ellipse without any need of point connectivity.

3.2 Primitive definition

We define a primitive that enables to take advantage of this property for characterizing ellipses. Such a primitive consists of a point and a tangent vector. Practically, we prefer to use a representation that is a bit redundant with a point (x and y coordinates) and the normalized coefficients of the tangent line equation (a, b and c, where $a^2+b^2=1$).

Finally: $P=(x,y,a,b,c)$

$(b,-a)$ is a normalized tangent vector, but $(-b,a)$ is another one: the choice of the sign gives us the curvature orientation. The c value is not really necessary but there should be to calculate it at each time we use the tangent line, and thus, it is easier like that.

3.3 The principle of the algorithm

The fundamental idea of our approach is the transposition of a classical reasoning method in mathematics. We first express a necessary condition that drastically reduces the set of potential solutions; then, we look within this set of potential solutions if some of them are effective solutions.

An ellipse can be defined by its center (two parameters) and three other parameters (e.g. the angle of its axis and the lengths of its two main axes). Our method consists in three steps:

1. We look for the approximate location of the center of the potential ellipse (i.e. if an ellipse is ”drawn” in the image, its center is necessarily this one)
2. Once a center has been found, we look for the approximate location of a reference ellipse
3. The neighborhood of this reference ellipse is the support for a deeper and strict research of the ellipse, if it exists.

These considerations can be derived with several centers (e.g. several road signs) and for several ellipses for a center (e.g. the inner and the outer ellipses of the outer red band of forbidding road signs).

Within each step, and especially for finding the center of a potential ellipse, we use embedded processes that progressively eliminate wrong solutions and that make the center converging to a quite accurate and reliable location.

3.4 Process homogenization

Several parameters must be adjusted in relation with the size of the ellipses we have to detect, and there may be big ellipses as well as small ones. Adjusting such parameters should be too complex because some of them are correlated. Let’s give some parameters that needs to be set : the maximum distance between triplets during the accumulation (if it’s too small a big ellipse will not have a distinct peak at its center, if it’s too big then it will add a lot of noise and hide small ellipses), the size of the masks during the tangent calculation phase (this mask has to encompass an edge, if it’s too big then it will capture too many points outside of the edge, if it’s too small then it will be too sensitive to noise), the parameters to compute the spatial connected components (angle between two consecutive edges, distance between primitives, etc...).

Thus, instead of changing their values, we change the size of the ellipses we search, and we keep an optimized set of parameters for a given size of image. Practically, we only look for ellipses that are in a range of 1 to 3, e.g. whose diameter (i.e. the greatest distance between two of its points) is between the $1/2$ and the $1/6$ of the image width.

We first define a strategy for scanning an image with sub-images and we then apply the global process after resizing each of these sub-images to a predefined size. For example, we can apply it to the initial image (to detect ”big ellipses”), then to four sub-images (dividing it in each dimension by two), then to nine ones (dividing the initial image in each dimension by three), and so on. Or, it can be driven by a global strategy that only considers some parts of the image that should be relevant.

3.5 Primitive characterization

We can work on color images, on grey-level images and even on binary images (in which edge points are sparsely distributed, and that can be noisy).

In the case we work on a color image, we transform it into a grey level one (e.g. using a linear combination of R, G and B values). Then, we apply a Nagao filter to eliminate a part of the noise (the advantage of this filter is that it replaces “a priori wrong values” by predominant values in their neighborhood). Finally, we apply a High-Pass filter to enhance the contours, and we use a threshold to produce a binary image in which points with value ”1” are on the edges.

The next step consists in extracting the ”relevant points” from this binary image: these points are those that should to ”support” a primitive, i.e. a location at which other points in the neighborhood are roughly along a line (that could be considered as a very local part of a curve).

We define a neighborhood and a set of masks, each of them being associated with an orientation. A mask is a thick segment of a given orientation. For each point of the binary image, we count the number of points covered by each mask and we keep the maximum of these numbers. If it is greater than a given threshold, we keep the corresponding point as a support for a primitive, else we do not keep it. The purpose of this step is to eliminate points that clearly do not belong to an edge. We obtain a subset of the initial set of points (initial binary image).

For each of these points, we compute the gradient at the corresponding location in the grey level image by applying a Gaussian Blur and a Sobel Operator. In the case we started with the binary image, we obtain an estimation of the tangent vector by using PCA (Principal Component Analysis) on the subset of points covered by the selected mask (it allows a more precise evaluation of the local tangent than by only using the direction of the mask). The drawback of this method is that it's less precise than using the Sobel Operator when the edges are net and precise, however it's more robust to local noise, and thus, might be used even on color image in a very noisy context.

We sort these points by decreasing values of their associated gradient measure; then, we scan this sorted list and we keep only those points whose distance to previous ones (that have already been selected) is greater than a given minimum distance d_{min}. It provides a sparse set of couples "point + gradient" in which two elements are not closer than d_{min}.

Then, we compute a normalized equation of the tangent line at each point from its location and the value of the associated gradient. But, as we mentioned it before, there are two possibilities for this normalized equation, one with (a,b,c) coefficients and the other one with the opposite values $(-a,-b,-c)$ for the coefficients.

We finally choose the set of coefficients that provides a positive result when replacing x and y by the coordinates of the curvature center: we do not know where is the curvature center but we have a reliable information on which side it is, by evaluating of the variance on both sides of the tangent line (the side of the curvature center locally contains pixels of both sides of the curve, and its variance must be greater than on the other side).

3.6 Locating a potential ellipse center

Our goal is not to directly find an ellipse but to focus on specific and structured areas in which the probability of finding an ellipse is different from zero. Such an area could be called an "elliptic band" and is defined by the dilation of a reference ellipse: it looks like a "thick ellipse" but, in some cases (when the reference ellipse is flat), it may look like a "filled ellipse" (even if it is not, mathematically).

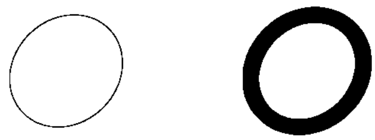

Figure 4: An elliptic band (on the right) is defined by the dilatation of a reference ellipse (on the left)

Figure 5: Elliptic band on a flat ellipse

Because this area is structured and does not contain too many pixels, we can then use a dedicated process to efficiently detect an ellipse (or ellipses) if it exists (they exist) in this area.

The first step is to locate the center of this reference ellipse, and we do it by applying the property of tangent lines (mentioned before) to the set of primitives we obtained previously.

Let us assume that an ellipse $\mathcal{E}$ can be associated with a subset $\mathcal{P}_{\mathcal{E}}$ of $\mathcal{P}$. Then any triplet of primitives (P_i, P_j, P_k) of $\mathcal{P}_{\mathcal{E}}$ produces a triplet of lines (D_{ij}, D_{jk}, D_{ki}) that intersect at the center C of $\mathcal{E}$.

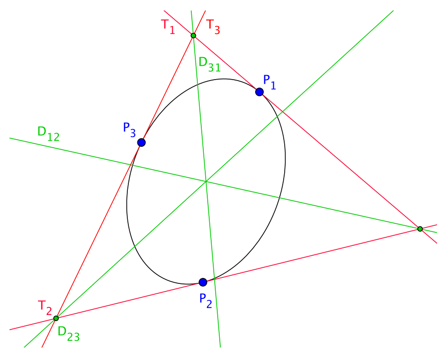

Figure 6: D_{12}, D_{23} and D_{31} intersect in C

Practically, because the primitives cannot be accurately computed, D_{ij}, D_{jk} and D_{ki} do not intersect perfectly in one point and thus, they produce a small triangle $\mathcal{T}r_{ijk}$ that covers the center location.

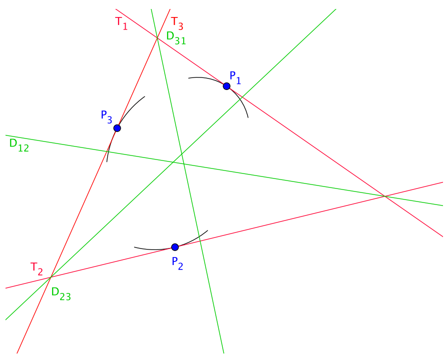

Figure 7: D_{12}, D_{23} and D_{31} do not intersect perfectly

We then use a Hough-like approach in order to characterize the existence of an ellipse and to approximately evaluate its center location. We consider a Hough Space that represents the image at a lower resolution (30x20): we draw triplets of $\mathcal{P}$ (not all the triplets, and under given rules of proximity); when the triangle $\mathcal{T}r_{ijk}$ is small enough (it happens when the three primitives belong to an ellipse $\mathcal{E}$), we evaluate its center in the Hough Space and we increment the corresponding cell. If a relevant maximum appears in the Hough Space, it means that, with a high probability, there is an ellipse $\mathcal{E}$ in the image, we obtain approximately its center C, and we characterize $\mathcal{P}_{\mathcal{E}}$ by keeping those primitives that "voted" to C.

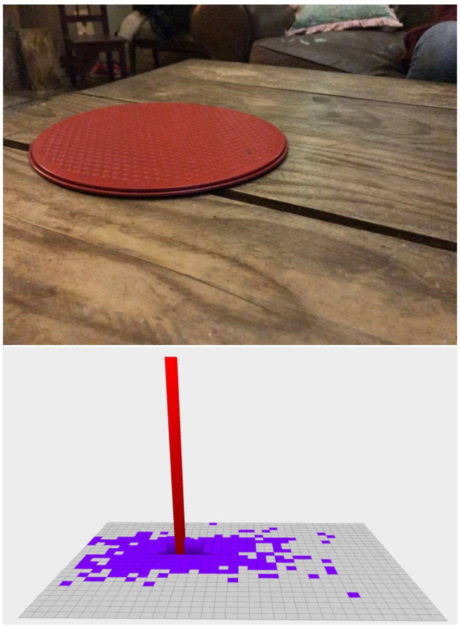

Figure 8: A single ellipse and the Hough Space

We may have to face to complex situations: for example, if there is an elliptic band (and not a single ellipse), then we will have to detect two concentric ellipses (the two boundaries of the band), associated with two subsets of primitives $\mathcal{P}_{\mathcal{E}1}$ and $\mathcal{P}_{\mathcal{E}2}$ that both vote for the same center C. In such a case, the approach described before still works but the resulting center does not appear so clearly, and we had to improve our algorithm in order to be able to provide a correct detection in all cases.

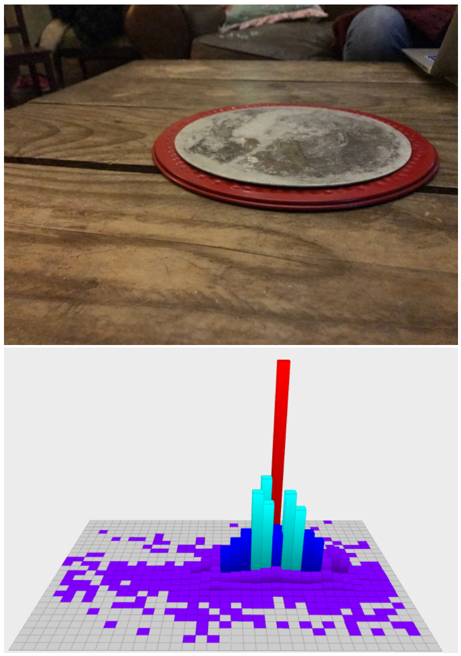

Figure 9: Two ellipses and the Hough center

We have developed an algorithm that enables to separate these two ellipses. At the end of the previous step, we obtain a subset $\mathcal{P}_{\mathcal{E}}$ and a center C, and we would like to split $\mathcal{P}_{\mathcal{E}}$ into $\mathcal{P}_{\mathcal{E}1}$ and $\mathcal{P}_{\mathcal{E}2}$. We define a graph – we call it the "Coherence Graph" – whose vertices are the primitives of $\mathcal{P}_{\mathcal{E}}$ and whose edges joining any couple of primitives have a weight that measures their involvement in finding C (in other terms, the weight of $\mathcal{E}d_{ij}$ is the number of other points P_k that participated with P_i and P_j to the vote for C): the "strong" connected components of this graph produce a link between the primitives of the same ellipses (it is important to notice that the primitives of such a connected component do not necessarily correspond to spatially connected components: it can be a sparse set of points, or two arcs of the same ellipse at opposite sides). We also use a more classical approach that consists in spatially connecting primitives using their proximity and their tangent line coherence.

3.7 Reference ellipse characterization

The Hough Transform provides an ellipse center: we do not directly associate it with the center of the cell that contains the maximum vote, but we use a weighted average with its neighbors. We also obtain a subset of primitives (or more) that participated to the characterization of this center (by means of the "Coherence

Graph" and with the help the spatially connected components). After eliminating outliers and taking care of the relative relevance of the resulting arcs of ellipse, we can characterize a "Reference Ellipse" that is an ellipse globally located close to all the relevant extracted elements.

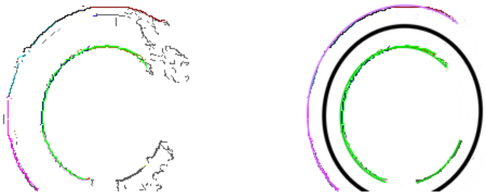

Figure 10: Connected components and reference ellipse

This reference ellipse structures the area in which we will perform a specific and accurate research of an ellipse (or ellipses) located in its neighborhood. This area is generated by the dilation of the reference ellipse, and if there are no topology changes (i.e. if it provides an homotopic transformation), the reference ellipse is its skeleton and is used to drive a dedicated research within this area.

3.8 Validation and configuration analysis

The information extracted all along this process gives us a robust basis to produce an efficient and very fast analysis that reliably and accurately characterizes the situation.

Until now, the algorithm was general and adapted to all cases, independently of the situation (color, texture, single ellipse or elliptic band whose edges are two ellipses,).

Now, we need to introduce knowledge (related to the problem we want to solve) in order to detect what we want to. Typically, we progress along the reference ellipse and we analyze an orthogonal cross section of the corresponding area: the distribution of values along this cross section provides results that can be analyzed globally. It enables to obtain the accurate location of the ellipse or the elliptic band, to characterize the hidden parts, and to refine the analysis inside the ellipse (color,pattern,).

Figure 11: The research area (with dimmed colors) and the extracted arcs of ellipses

4 RESULTS

We did obtain convincing results in some complex situations in which classical algorithms fail, especially for dashed ellipses, for single short arcs of ellipses, and for very flat ellipses.

Let us explain briefly and then illustrate why the algorithm we propose works in these cases when the other ones do not.

About dashed ellipses, the reason is that we do not require any continuity to assemble edges under the constraint of belonging to a same ellipse: we only need a sparse set of primitives that simply indicate a location and an associated orientation.

Most approaches described in the state of the art requires to have parts of ellipses with different orientations, and thus, in most of the cases to have at least more than the half of an ellipse to detect it (in the presentation of the results on different images and the comparison of various algorithms, Jia and al[Jia16] clearly show this problem). Our algorithm does not require this constraint and works even if only a small arc is visible.

In the case of flat ellipses, it is interesting to give some more scientific details. It should seem difficult, and even impossible, to compute the intersection of tangent lines when they are parallel or quite parallel. In fact, it is not the case because all the geometrical elements are considered in the homogeneous space (projective geometry) and thus, there is absolutely no problem to compute the intersection of two parallel lines, and no problem of accuracy when computing the intersection of two lines that are almost parallel.

In the first subsection, we illustrate how the algorithm works in some difficult cases. Then, because we wanted to quantify its limits, we developed a program to test its robustness when evolving toward limit cases, and this is illustrated in the second subsection.

4.1 Some results in critical cases

The results shown in this section were obtained with this set of parameters : 350 pixels for the width of the sub-image, 150 pixels for the maximum distance between points of a same triplet, 31 pixels for the size of the mask, 40degrees for the max angle between two consecutive tangents of a connected component, 7 pixels for the maximum distance between two consecutive points. These values were chosen to detect ellipses whose size is comprised between one half and one sixth of the sub-image. The following examples present the detected ellipse (or arc of ellipse) in several cases: different sizes, party hidden, different point of view.

Figure 12: Ellipses of different sizes and partly hidden

Figure 13: Three arcs of a unique ellipse

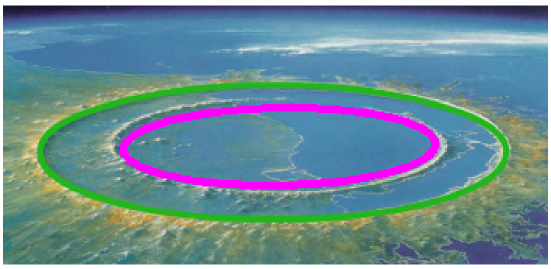

Figure 14: Two very flat ellipse with noisy edges

4.2 Exploring the limits of this algorithm

Several parameters can be studied to provide an extensive evaluation of the algorithm robustness; In particular, we can produce different types of noise, we can consider different sizes of arcs (in terms of ratio of the whole ellipse), different numbers and locations of very small arcs, ... but it should be too long and it is not the goal of this communication. So we decided to consider the variation of only a very few but relevant parameters in order to show how it works in limit cases.

In order to do that, we have simplified the protocol by directly starting with the binary image (we note that, in this case, the tangent line evaluation is less accurate than when using the grey level image).

We produce binary images that are based on the projection of an 3D circle viewed under a varying angle, with a given level of discretization (we do not visualize a polygon that fits an ellipse but a set of points distributed on this ellipse), and with a varying amount of noise.

On the next figures, we display images from our testing program. The first one (figure 14) is the original image; then (figure 15), we visualize the points supporting the primitives; and finally (figure 16), the Hough space shows a very good detection of the ellipse center.

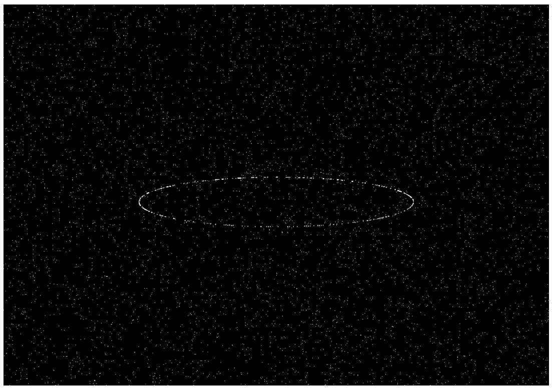

Figure 15: The original binary image (partly hidden ellipse and noise)

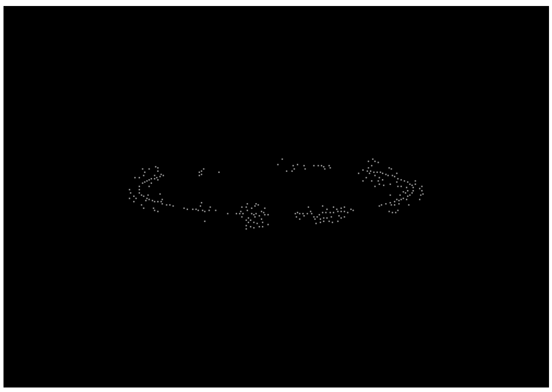

Figure 16: Points supporting primitives

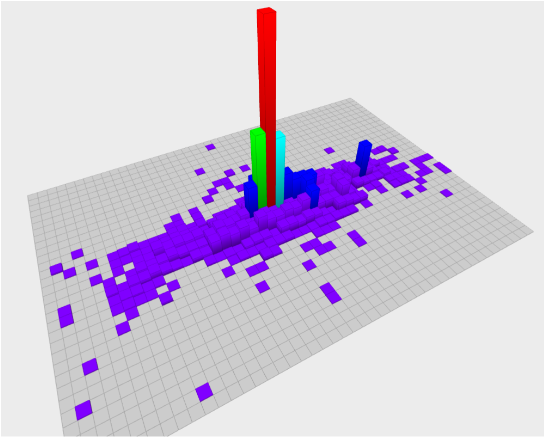

Figure 17: The corresponding Hough Space

The next figures provide a synthetic visualization of the robustness of the algorithm for various angles.

Each diagram is the result for different angles: in these cases, the angles are of 75 degrees, 80 degrees and 88 degrees (they measure the angle between an orthogonal vector to the ellipse plane and the viewing direction; e.g. in the last case – 88 degrees – it means that the viewpoint is almost in the plane of the ellipse – 2 degrees over it – and so the ellipse is very flat).

In each diagram, we put on the right (where it is written ”degradation”) the ratio of points of the ellipse that have been removed (0% means we fully display the ellipse, and 100% means we do not display any point of the ellipse). And we put on the left (where it is written ”Bruit”) the ratio of noise (this ratio is the number of points added randomly compared to the number of points of the fully displayed ellipse – e.g. 200 means there are twice more points). For example, at the cell ”Bruit = 300 and Degradation = 50”, we have only 50% of the points of the ellipse (50% have been randomly removed) and 6 times more points added randomly. These perturbations also deteriorates the tangent lines computed with the set of mask.

For each cell, we ran the detection program 1000 times (after 1000 times, the ratio is stabilized) with different sets of random values (under the constraints of ratios of the corresponding cell) and we used the following color code:

- green is for less than 10% of failure
- yellow is between 10% and 25% of failure
- orange is between 25% and 50% of failure
- red is for more than 50% of failure

These diagrams are very interesting because they drove us in the choice of the algorithm parameters. For example, we can see that for angles close to 90 degrees (flat ellipses); the sensitivity of the detection is more important for noise than for the number of points of the ellipse (with only 20% of points and a noise that is not too important, we detect such flat ellipses; thus, it should be better to threshold more drastically the grey-level image of contours in order to provide a better detection of flat ellipses.

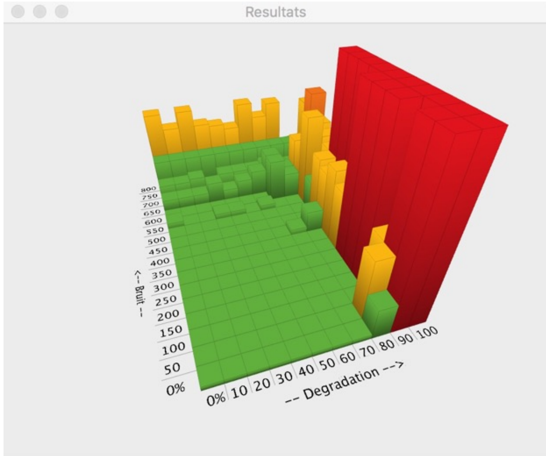

Figure 18: Angle value is 75 degrees

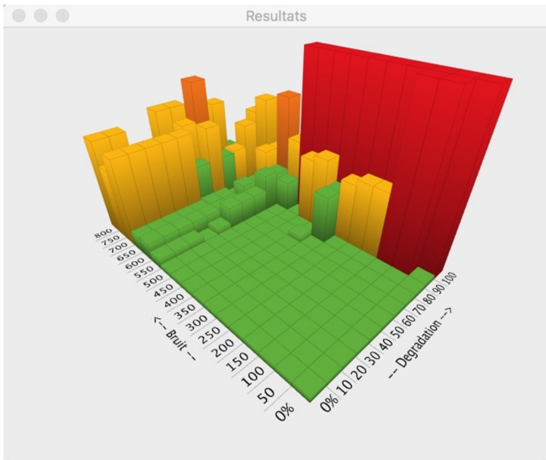

Figure 19: Angle value is 80 degrees

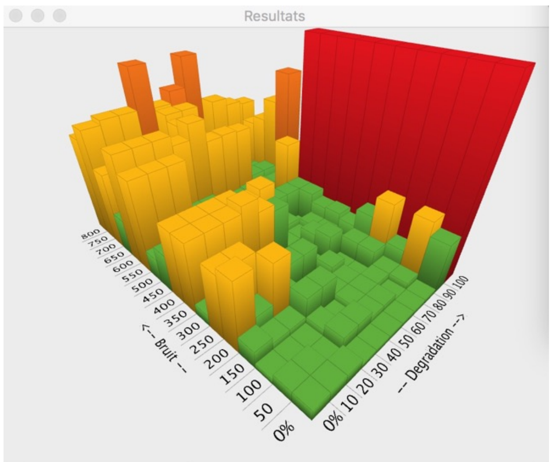

Figure 20: Angle value is 88 degrees

5 CONCLUSION

The algorithm proposed in this communication provides a solution to the ellipse detection problem when the ellipse is dashed, partly visible or very flat, all these situations being difficult to be solved with classical algorithms.

As it was mentioned before, this algorithm has to be used in complement to another one in order to avoid false negative, knowing that it does produces a very low amount of false positive (thanks to the validation step), when we need to have a complete and robust solution to ellipse detection.

6 REFERENCES

[Bas05] C. A. Basca, M. Talos and R. Brad, Randomized Hough Transform for Ellipse Detection with Result Clustering, EUROCON 2005 - The International Conference on ”Computer as a Tool”, Belgrade, 2005, pp. 1397-1400.

[Can86] J. Canny, A Computational Approach to Edge Detection, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8-6, pp. 679-698, 1986.

[Dud72] Duda, Richard O. and Hart, Peter E., Use of the Hough Transformation to Detect Lines and Curves in Pictures, in Commun. ACM, vol. 15-1, pp. 11-15, 1972.

[For14] Michele Fornaciari, Andrea Prati, Rita Cucchiara, A fast and effective ellipse detector for embedded vision applications, Pattern Recognition, Volume 47, Issue 11, November 2014, Pages 3693-3708, ISSN 0031-3203, http://dx.doi.org/10.1016/j.patcog.2014.05.012.

[Jia16] Qi Jia and Xin Fan and Zhongxuan Luo and Lianbo Song and Tie Qiu; A Fast Ellipse Detector Using Projective Invariant Pruning, in CoRR, vol. abs/1608.07470, 2016.

[Lib07] Libuda Lars and Grothues Ingo and Kraiss Karl-Friedrich, Ellipse Detection in Digital Image Data Using Geometric Features, in Advances in Computer Graphics and Computer Vision: International Conferences VISAPP and GRAPP 2006, Setúbal, Portugal, February 25-28, 2006.

[Lu15] T. Lu, W. Hu, C. Liu and D. Yang, ”Effective ellipse detector with polygonal curve and likelihood ratio test,” in IET Computer Vision, vol. 9, no. 6, pp. 914-925, 12 2015. doi: 10.1049/iet-cvi.2014.0347

[Pra13] Dilip K. Prasad, Maylor K.H. Leung, Chai Quek, ElliFit: An unconstrained, non-iterative, least squares based geometric Ellipse Fitting method, Pattern Recognition, Volume 46, Issue 5, May 2013, Pages 1449-1465, ISSN 0031-3203, http://dx.doi.org/10.1016/j.patcog.2012.11.007.