# Efficient curve detection using a Gauss-Newton method with applications in agriculture

R. Andreani[1], G. Cesar[1], R.M. Cesar-Jr[2], J. M. Martinez[1], P. J. S. Silva[2]

[1] Department of Applied Mathematics, IMECC-UNICAMP, State University of Campinas, Campinas SP, Brazil.
{andreani, giovane, martinez}@ime.unicamp.br
[2] Department of Computer Science, IME, University of São Paulo. São Paulo, Brazil.
{cesar, pjssilva}@ime.usp.br

**Abstract.** Agriculture is one of the most important economic activities in Brazil and much research effort has been concentrated to develop new computer vision techniques to analyze agricultural images. The present paper focus on the problem of detecting geometric primitives (straight lines and circles) in agricultural images, which is done by a model fitting approach. Model fitting plays a crucial role in many problems in computer vision and most state-of-art methods (e.g. RANSAC) rely on sampling to solve this problem. This paper presents a new approach based on Gauss-Newton methods that does not depend on sampling. Model fitting is expressed as an Order-Value Optimization (OVO) problem, which consists in the minimization of an order-value function. An efficient algorithm is then used to solve the OVO problem. The proposed approach is efficient, robust to noise and presents a very important property: both running time and memory costs increase linearly with the number of active points in the image. The performance of the proposed algorithm improves when compared to alternative methods as the number of model parameters increases. A curve detection method based on the OVO approach is here described. Methods for the detection of straight lines and circles are provided. Extensive comparative performance assessment using synthetic data has been carried out considering the following methods: Hough transform, RANSAC, QMDPE and LKS. The obtained results indicate that the assessed methods are comparable regarding precision and robustness to outliers. The introduced method is faster than the others in the case of circle detection, as the number of model parameters is increased. Preliminary experimental results using real agricultural images are provided to illustrate the proposed approach.

## 1 Introduction

The analysis of biological forms has been a key shape analysis application for more than 40 years [15]. There has been a great variety of computer vision techniques developed aiming at the analysis of images from life sciences from micro to macro scales. For instance, bioinformatics became a very popular field mainly during the 90's with the sequencing techniques and projects that produced large amounts of genome information. After this initial phase, research attention started to be focused on the relation

between genotype and phenotype, i.e. how the genome information available can be explored to better understand how genotype generates phenotype. In this context, image analysis will play a key role since it is a natural approach to capture and to quantify phenotype information. Among the important computer vision applications to analyze images from life sciences, it is worth mentioning agricultural images. Agriculture is one of the most important economic activities in Brazil and much research effort has been concentrated for the development of agricultural technology[3]. Examples of computer vision applications in Brazilian agriculture include unmanned aircrafts for precision faming [20], the analysis of raindrops [14] and volume calculation of fruits [12], to name but a few. Curve fitting is a particular important issue that has many applications in agriculture imaging such as the detection of seeds, fruits, tree tops and soil pores, for instance. The present paper addresses this problem by introducing a new methodology and comparing it to standard techniques available from the literature.

Even though there are some curve detection methods that are applied to gray-scale images, most rely on a binary image representing points of interest (e.g. edges after edge detection). Moreover, simple line fitting methods are not in general suitable because it is necessary to identify which image points are associated to each curve (i.e. fitting a curve to *all* image edge points is not a solution in general as few discrepant points are already sufficient to induce poor results). In other words, the image points that do not actually belong to a given line may be viewed as outliers to be ignored by the method. The most well-known solution for the curve detection problem is the Hough transform [11]. The original idea has been proposed as a patent in the early 60's [10], put into the pattern recognition context by Rosenfeld, Duda & Hart [8, 19], and popularized in Ballard's well known paper [4]. Since then, its performance has been improved by much research based on important ideas like backmapping [6], gradient information [17], digital straight line models [7], constraints [17, 18], probabilistic approaches [13] or resampling [5, 9]. The advantages of the Hough transform include its generality and the fact that it is more robust to noise. However, there are some key problems that limit its adoption for general curves, as its computational cost increases substantially with the number of curve parameters. Curve detection may also be solved by parametric model fitting methods like RANSAC, for instance. Some key issues related to such methods are robustness to outliers, running time, memory requirements and quality of the final solution. Most state-of-art methods explore random sampling to perform well regarding these issues. In particular, coping with outliers is a very important topic in computer vision, which explains the popularity of the so called robust techniques in the field [16, 21]. A different approach, which does not rely on sampling, is proposed in the present paper: a new model fitting-based curve detection method which is expressed as an Order-Value Optimization (OVO) problem [2].

The paper is organized as follows. Section 2 describes the OVO problem and the Gauss-Newton algorithm that solves it. Section 3 models the curve detection task as an OVO problem, detailing the required steps to explore the Gauss-Newton method to detect parametric curves in images. Experimental results are presented and discussed in Section 4. The paper is concluded with some remarks on our ongoing research in Section 5.

[3] See e.g. `http://www.embrapa.br/`

## 2 Order-value optimization problem

Assume that the data is composed of points $P_1, \ldots, P_m$ and that the model to be fitted is defined by $n$ parameters $x = (x_1, \ldots, x_n) \in \Omega \subset \mathbb{R}^n$. For each point $P_i$, $f_i(x)$ is defined as the error of the attribution of the point $P_i$ to the model defined by the parameters in $x$. For each $x \in \Omega$, let $i_1(x), \ldots, i_m(x)$ be a permutation of the indices that sorts the error, i.e. $f_{i_1(x)}(x) \leq ... \leq f_{i_m(x)}(x)$. The problem consists of minimizing $F_p(x) \equiv \sum_{k=1}^{p} f_{i_k(x)}(x)$, where $p \leq m$ represents the minimum number of points that are supposed to belong to the fitted model. This $p$ parameter is very important as it distinguishes the suggested model from simple model fitting where all the data points enter in the error expression. In particular, it is the fact that only the closest points to the model play a role in the error measurement that gives the model its robustness to noise and outliers. This advantage does not come without its costs though: the optimization problem associated to the OVO model is nonsmooth and nonconvex, hence much more difficult to solve than smooth and convex problems that usually appear in simple model fitting.

Nevertheless, efficient algorithms for solving OVO problems have been recently proposed in [1, 2]. Gradient methods have been initially proposed. More recently, Newton variations have also emerged: specially quasi-Newton and Gauss-Newton variants [1]. The theory ensures that limit points generated by many OVO algorithms are stationary and almost certainly local minimizers. This is a typical optimization problem where local minimizers are important and deserve to be detected, besides the global minimum. In fact, one is interested in discovering as many meaningful models as possible, not only the one that best fits $p$ points. Fortunately, the theory supports the conjecture that, starting from well distributed initial points, convergence to close local minimizers will occur.

The OVO algorithms are iterative and the convergence of the Newton variants is, usually, super-linear [1]. The complexity of a single iteration depends on the Hessian approximation adopted. The Gauss-Newton approach used in this paper needs, at each iteration, the computation of a matrix $A \in \mathbb{R}^{p \times n}$, the matricial products $A^T A$ and $A^T v$ and the resolution of an $n \times n$ linear system. The computation of $F_p(x)$ involves, at most, $O(m)$ flops and is dominated by the selection of the $p$ smallest errors. Therefore, roughly speaking, an iteration of OVO may be completed using $O(m + pn + n^2 p + np + n^3)$ time cycles. As usually $p$ and $n$ are much smaller than $m$, this complexity is, roughly, linear in the number of points in the data. This value is quite moderate, but it must be stressed that several iterations (typically $10$) are needed for convergence of the Gauss-Newton method. Moreover, different initial points are necessary to detect many local minimizers. Remarkably, the memory requirement of the proposed algorithm is also roughly linear, $O(mn + n^2)$.

A curve in $\mathbb{R}^2$ depending on the parameters $x_1, \ldots, x_n$ is defined by $\varphi(x, z) = 0$, where $z = (z_1, z_2)$. For example, circles are defined by $\varphi(x_1, x_2, x_3, z_1, z_2) \equiv (z_1 - x_1)^2 + (z_2 - x_2)^2 - x_3^2 = 0$. Many times, one is interested only on a subset of the family $\varphi(x, z) = 0$, represented by constraints on the parameters $x$. In this case, we denote by $\Omega$ the constraint set and we write $x \in \Omega$.

Given the set $P_1, \ldots, P_m$ of target points, we define

$$r_i(x) = \varphi(x, P_i) \text{ and } f_i(x) = r_i(x)^2.$$

Therefore, $f_i(x)$ measures the error implicit in the statement "$P_i$ belongs to the curve defined by $x$". The integer $p \leq m$ is a problem-dependent parameter that, roughly speaking, represents the number of target points that must belong to a curve so that it can be recognized in the image. Typically $p$ is much smaller than $m$.

The indices $i_1(x), \ldots, i_m(x)$ are an index permutation that sorts the errors, as defined in Section 1. The OVO objective function is:

$$F_p(x) = \sum_{k=1}^{p} f_{i_k(x)}(x).$$

In order to minimize $F_p$ we consider the following algorithm [2]:
**Algorithm 1.** Let $x^0$ be an initial set of curve parameters. Let $x^k$ be the approximation of the curve parameters at the beginning of iteration $k$. We compute $x^{k+1}$ as follows:
**Step 1.** Compute

$$\nabla r(x^k) = (\nabla r_{i_1(x^k)}(x^k), \ldots, \nabla r_{i_p(x^k)}(x^k))$$

If $\|\nabla r(x^k) r(x^k)\|$ is small enough, stop.
**Step 2.** Solve the Gauss-Newton system

$$\nabla r(x^k) \nabla r(x^k)^T d = -\nabla r(x^k) r(x^k).$$

(If the matrix is nearly singular, or the direction obtained is almost orthogonal with the gradient, use a diagonal perturbation to force positive definiteness.)
**Step 3.** Starting with $t = 1$, reduce $t$ using safeguarded quadratic interpolation, until the following (Armijo) inequality holds:

$$F_p(x^k + td) \leq F_p(x^k) + td^T \nabla r(x^k) r(x^k).$$

if the test succeeds with $t = 1$ then it is increased until failure.
**Step 4.** Define $x^{k+1} = x^k + td$.

Limit points of sequences generated by this algorithm are stationary points in the sense of [1]. This means that, if $x^*$ is such an stationary point, we can define $i_1(x^*), \ldots, i_p(x^*)$ in such a way that $\nabla F_p(x^*) = 0$. In other words, $x^*$ is a putative local minimizer.

## 3 Curve detection

### 3.1 Method Overview

Curve detection is performed in three steps: (1) detection of the image points of interest (e.g. edge detection, image segmentation followed by contour extraction, etc.); (2) selection of initial solutions for each curve candidate; (3) application of Algorithm

1 (Section 2). In the results reported in this paper, edge detection is performed by the Canny edge detector, thus producing a set of 2D points that are used as input to the curve detection algorithm. These points are referred as *active points*. An initial solution for Algorithm 1 (i.e. $x^0$) is specified as the parameters that possibly converge to those of the desired curve in the image. The selection of initial solutions depend on the curve to be detected. Although the introduced framework applies to any parametric curve, we discuss the cases for straight line and circle detection in the present paper. These are detailed below.

### 3.2 Straight Lines

We have used the normal line parameterization, where each line is represented by an angle $\theta$ and a drift $\rho$. A point $(x, y)$ belongs to the line if $x\cos(\theta) + y\sin(\theta) = \rho$.

The initial solutions were chosen to form a regular grid in the $(\rho, \theta)$ parameter space. We first divide the range of angles that define lines in the picture, $[0, \pi]$, in `ntheta` equally spaced arcs. We then compute the range of drifts $\rho$ that corresponds to lines that intersect the image window and divide this range in `nrho` intervals. The chosen $\rho$ are the intervals midpoints. Usually, `ntheta` is larger than `nrho` as the function that has to be minimized by Algorithm 1 is highly nonlinear in $\theta$, and merely quadratic in $\rho$. See for example Figure 1(a).

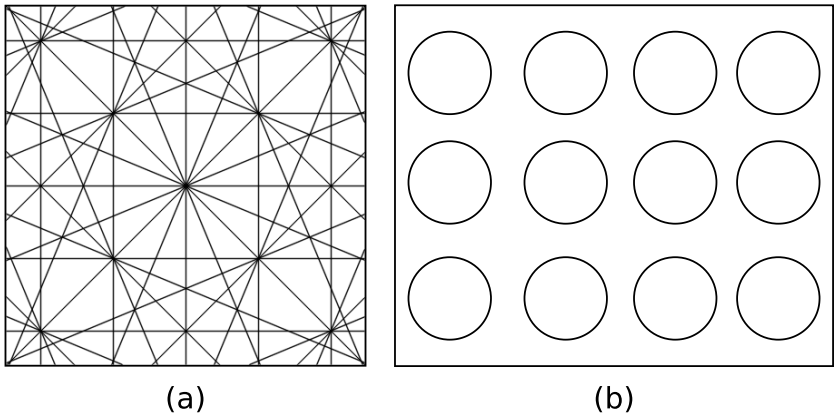


**Fig. 1.** Schemes for producing initial solutions for (a) straight lines; and (b) circles.

Once a solution $(\rho^*, \theta^*)$ of the OVO problem is reached, some tests verify if a valid straight line has been detected:

- The number of active points that satisfy $|r_i(\rho^*, \theta^*)| \leq \epsilon$ (usually $\epsilon = 0.5$) must be larger than an *a priori* threshold $M$. See Figure 2(a).
- The median and variance (relative to the median) of the detected points are calculated. The variance divided by the number of detected points must be lower than an *a priori* threshold $\lambda$.

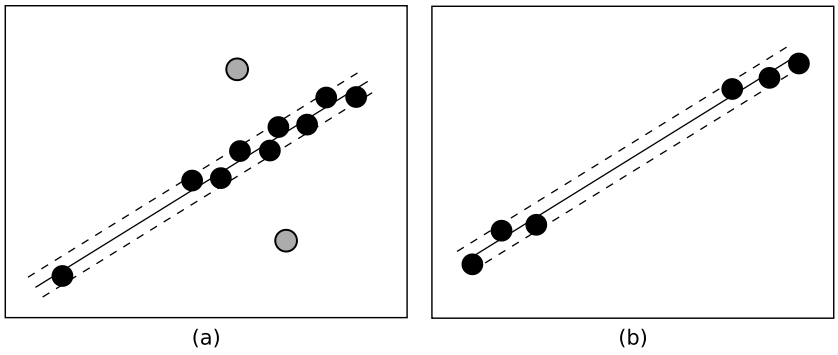


**Fig. 2.** (a) Active points counted on lines. (b) Dispersion test.

Using these tests, we exclude solutions that correspond to "lines" composed by few active points or many dispersed points, like in Figure 2(b).

If a solution succeeds both tests, it corresponds to a real line in the image and the points identified in the first test are excluded from the active points set. It is worth noting that the image points that actually belong to the detected curve are recognized (i.e. those active points that satisfy the first condition above), which is important in different situations, e.g. grouping and edge filling. In case of straight lines, there is no need for searching for the end points of each detected line (which is the case for the Hough transform, for instance). The algorithm proceeds with the next initial solution and remaining active points.

### 3.3 Circles

The image is divided in $r \times s$ points which are used as circle centers. Each center, together with an *a priori* defined radius, is used as an initial solution for circle detection. See Figure 1(b). Depending on the specific application, different strategies may be adopted to decrease the number of initial circles (e.g. we may eliminate regions without edge points). Since this is not a central issue in the present paper, no such strategies have been designed.

The initialization and application of Algorithm 1 to detect circles are analogous to the straight line case, explained above. The main differences are in tests to accept a solution computed by the OVO model:

- Like in the straight line case, the number of active points near the solution is compared to a threshold. See Figure 3;
- The number of active points identified above is divided by the area of the respective circular tolerance ring defined around the solution. The computed value is compared to a threshold.

The second test ensures that circles with large radius have points enough to be regarded as a valid circle arc.

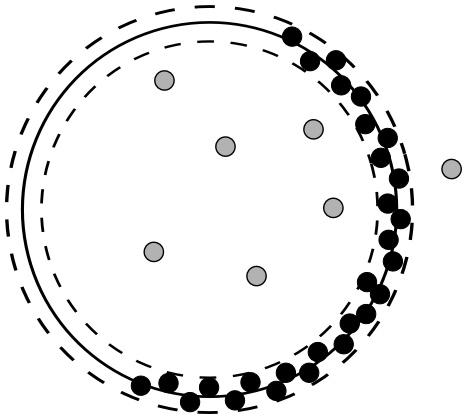

**Fig. 3.** Active points counted on circles.

After the tests, the method proceeds as in the straight line case. We exclude the active points close to a circle (i.e. within the tolerance ring), that were identified in the first test, and restart the process with the next initial solution.

## 4 Experimental results

In order to evaluate and compare the proposed method, comparative extensive performance evaluation tests have been carried out based on the experiments described in [21], which presents a comparison among the following curve detection methods: QMDPE, RESC, ALKS, LMedS, RANSAC and Hough transform. In the present paper, we have improved the evaluation protocol in some important aspects and compare our method to QMDPE, RANSAC, Hough transform, which presented better performance in [21], as well as LKS, a natural upgrade of LMedS [3]. The image sizes have been increased from $100 \times 100$ to $400 \times 400$ for lines and from $150 \times 150$ to $300 \times 300$ for circles to allow accurate processing time measurement in modern computers.

The following methods have been implemented and compared: HT (Hough transform), RANSAC, QMDPE [21], LKS [3], and OVO. These methods have been chosen because they presented the best performance in [21] and may be considered state-of-art algorithms for model fitting. Six patterns of noisy data have been generated: step, three step, roof, six lines, one circle and five circles, as shown in Figure 4. Each experiment corresponding to straight line detection, cases (a)-(d), has been carried out 1000 times (i.e. 1000 different realizations of the noisy image). The number of samples for the sampling methods have been computed to ensure that, with at least $99\%$ of confidence, a real line would be found. More specifically, the number of samples were $2000$ for step, $4000$ for three step, $8000$ for roof, $4500$ for six lines. In the OVO method, we used `ntheta`$= 16$ and `nrho`$= 10$, as described in Section 3.2. The experiments for circle detection, cases (e)-(f), have been repeated 100 times. The number of samples for the sampling methods was $2000$ for one circle, and $75000$ in five circles. The number of

initial circles for OVO was $16$ in one circle and $400$ in five circles. In both cases the initial radius was $30$. Gaussian noise with zero mean has been adopted in all cases. For each experiment, the time for a full run of each method has been measured and the best line or circle found according to the respective criterion was computed. For OVO, we did not use the tests to detect good curves described in Sections 3.2 and 3.3 since they are not well suited for such synthetic tests. We used then the RANSAC criterion: the best curve is the one with the largest number of active points close to it. The parameter $p$ for OVO is taken as a rough upper bound of the visual length of the object (line or circle) instead of the exact length to ensure realistic results for the synthetic test.

The parameters of the detected line or circle have been used to verify if a valid curve has been detected. Table 1 summarize the obtained results. Each cell of the table shows the percentage of correct matches and the average running time in seconds. As it can be seen, all methods present a similar performance regarding precision, i.e. number of correct matches. As far as running time is concerned, HT presented the best results for the straight line detection experiments except for the step case, where RANSAC outperformed the others in running time. On the other hand, for the circle detection experiments, OVO was significantly faster than all others. The reason is that circle fitting involves a larger number of parameters.

| Method | Step | | 3 step | | roof | | 6 lines | | 1 circle | | 5 circles | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HT | 100 | 0.15 | 100 | **0.14** | 99.9 | **0.16** | 100 | **0.10** | 100 | 12.09 | 100 | 7.18 |
| RANSAC | 99.6 | **0.10** | 99.9 | 0.19 | 99.2 | 0.44 | 100 | 0.13 | 100 | 0.14 | 93 | 5.19 |
| QMDPE | 99.8 | 0.72 | 100 | 1.32 | 99.4 | 3.06 | 100 | 0.85 | 98 | 0.55 | 68 | 17.40 |
| LKS | 100 | 0.29 | 100 | 0.54 | 99.9 | 1.24 | 100 | 0.39 | 99 | 0.20 | 100 | 7.60 |
| OVO | 100 | 0.49 | 99.8 | 0.46 | 99.6 | 0.49 | 99.6 | 0.32 | 100 | **0.06** | 100 | **0.64** |

**Table 1.** Precision results and running time for the experiments. Each cell shows the percentage of correct matches and the average running time in seconds (best times are shown in bold).

Another important parameter to evaluate a model fitting method is its robustness to outliers, which may be measured by the breakdown point [21]. The experimental setup described in [21] used noisy realizations of an image with one line to measure the robustness of the method as the number of line points decreases while the noisy points (i.e. outliers) increases. We have repeated this experiment but also evaluated one case containing one single line and a cluster of points, which is known to degrade the performance of some line fitting methods. The plots in Figure 5 shows the model fitting error (in terms of the normal parameters $\rho$ and $\theta$ of the line) as a function of the increasing noise. As it can be seen, all methods present a similar performance regarding precision and breakdown point, being robust up to more than $95\%$ of outliers.

The introduced approach has been applied to agricultural images and some preliminary results are shown in Figure 6. The original images are shown in the left column while the corresponding edge points are shown in the right column. The detected curves

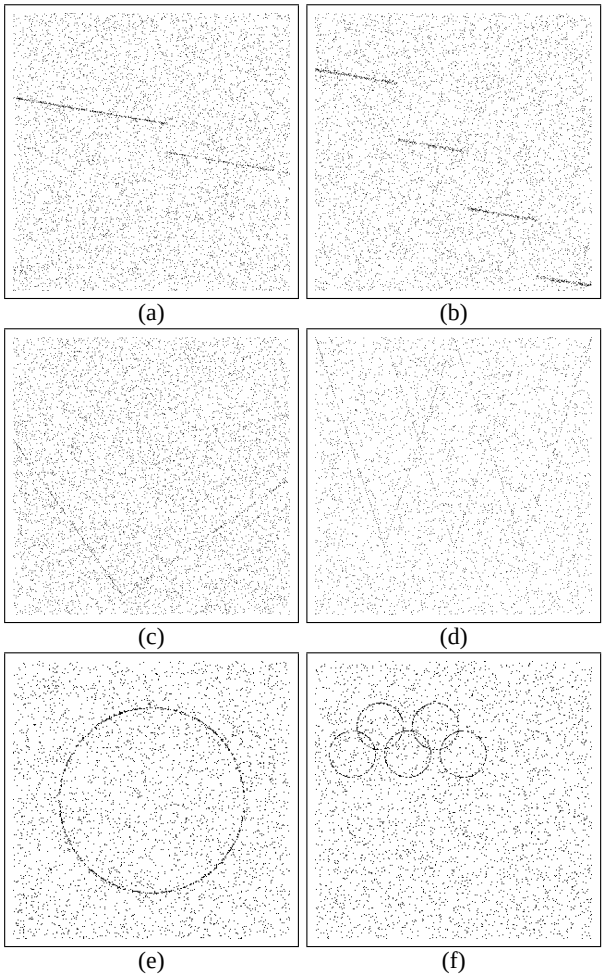


**Fig. 4.** Example of each type of noisy data used in the performance evaluation: (a) step, (b) three step, (c) roof, (d) six lines, (e) one circle, (f) five circles.

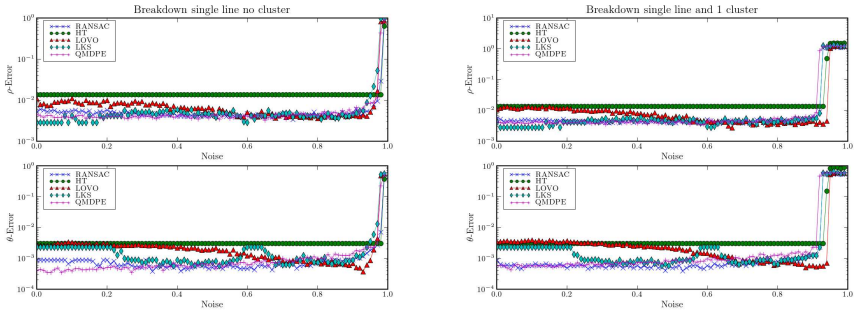


**Fig. 5.** Breakdown-point plots for two analyzed cases. The error is presented in log scale. The plots for each case show the model fitting error in terms of the normal parameters $\rho$ and $\theta$ of the line as a function of the increasing noise.

are shown superimposed to both images. As it can be seen, the methods have been successfully able to detect the target objects (trees, fruits and tree tops from aerial images).

## 5 Concluding remarks

Computer vision plays an important role in many situations for agricultural image analysis. One of such problems is the detection and fitting of geometric parametric primitives such as straight lines and circles. A new model fitting approach has been proposed as an OVO problem. An efficient algorithm is used to minimize an order-value function and thus detect the desired curves in an image. The proposed approach has been implemented to detect straight lines and circles (including circle arcs) and experimental results have been shown. The method is efficient, robust to noise and presents a very important property: both running time and memory costs increase linearly with the number of active points in the image. Other parametric curves can also be detected by the proposed approach, which is expected to perform well when more parameters are involved, e.g. for ellipses. A very important issue is how to decide a good parameterization for the OVO problem, which has important implications in the computational performance of the method. Also, a strategy for generating the initial solution should be designed for each different type of curve. It is interesting to compare the OVO computational costs to those associated to the Hough transform. OVO complexity is basically the number of initial solutions times the time to run the Gauss-Newton method that is linear in the number of active points in the image and cubic in the number of curve parameters $n$. Now, let $q$ represent the number of voting cells for each parameter (i.e. the parameter space discretization size). The Hough transform uses $O(q^n)$ in both running time and memory. This difference may indicate that, in practice, the OVO approach is not only suitable for detecting straight lines ($n = 2$), but it may be considered specially appealing to shapes like circles ($n = 3$), ellipses ($n = 5$) and other parametric curves that involve many parameters. The OVO performance is more attractive as the number

of model parameters increase and for large images that would require large $q$ to keep good precision of the Hough approach.

Our ongoing work includes the development of a detector for ellipses, which are much more commonly found in agricultural images due to 3D projection and the application of the developed techniques in agricultural studies. Such advances will be reported in due time.

*Acknowledgements* The authors are grateful to CNPq (proc. 300722/98-2, 52.1097/01-0 and 468413/00-6), FAPESP (proc. 05/00587-5, 01/09401-0 and 04/03967-0) and CAPES for financial help. The authors are grateful to Lúcio A. Jorge (EMBRAPA) for the test images and discussion about the agricultural images application.

## References

[1] R. Andreani, J. M. Martínez, L. Martínez, and F. Yano. Order-value optimization and applications. `http://www.optimization-online.org/DB_FILE/2005/10/1234.pdf`, 2005.

[2] R. Andreani, J. M. Martínez, M. Salvatierra, and F. Yano. *L. Liberti and N. Maculan (Eds.), Global Optimization: from Theory to Implementation*, chapter Global Order-Value Optimization by means of a multistart harmonic oscillator tunneling strategy, pages 379–404. Kluwer, 2006.

[3] A. Bab-Hadiashar and D. Suter. Robust segmentation of visual data using ranked unbiased scale estimate. *Robotica*, 17(6):649–660, 1999.

[4] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[5] William A. Barrett and Kevin D. Petersen. Houghing the hough: Peak collection for detection of corners, junctions and line intersections. In *CVPR*, pages 302–309, 2001.

[6] L. F. Costa and R.M. Cesar-Jr. *Shape Analysis and Classification: Theory and Practice*. CRC Press, 2001.

[7] L. da F. Costa and M.B. Sandler. Effective detection of bar segments with hough transform. *CVGIP: Image Understanding*, 55(3):180–191, 1993.

[8] R. O. Duda and P. E. Hart. Use of the hough transform to detect lines and curves in pictures. *Communications of ACM*, 15(1):11–15, 1972.

[9] C. Galambos, J. Matas, and J. Kittler. Progressive probabilistic hough transform for line detection. In *CVPR*, pages 1554–1560, 1999.

[10] P.V.C. Hough. Method and means for recognizing complex patterns. *United States Patent Office*, 13:Patent 3,069654, 1962.

[11] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision, Graphics, and Image Processing*, 44:87–116, 1988.

[12] L. A. C. Jorge, R. Minghim, L. G. Nonato, C. I. Biscegli, C. Pedrino, V. O. Roda, and M. S. V. Paiva. 3d reconstruction, visualization and volume calculation of fruits. In *Proc. Computer Graphics International - CGI 2007*, 2007.

[13] N. Kiryati, Y. Eldar, and A. M. Bruckstein. A probabilistic hough transform. *Pattern Recognition*, 53:213–222, 1991.

[14] L. V. Koenigkan. Desenvolvimento de método para a determinação do volume de gotas de chuva artificial baseado em processamento de imagens com análise de curvatura (in portuguese). Master's thesis, EESC - USP, 2003.

[15] R.S. Ledley. High-speed automatic analysis of biomedical pictures. *Science*, 146:216–223, 1964.

[16] P. Meer. *Robust Techniques for Computer Vision*, chapter 4. IMSC Press Multimedia Series. Prentice Hall, 1st edition, July 2004.

[17] C. F. Olson. Constrained hough transforms for curve detection. *Image and Vision Computing*, 73(3):329–345, 1999.

[18] C. F. Olson. A general method for geometric feature matching and model extraction. *International Journal of Computer Vision*, 45(1):39–54, 2001.

[19] A. Rosenfeld. *Picture Processing by Computer*. Academic Press, 1969.

[20] O. Trindade-Jr., L.A.C. Jorge, and J.G.B. Aguiar. Field of dreams - using uavs for precision farming. *Unmanned Systems*, 22:35 – 39, 2004.

[21] H. Wang and D. Suter. Mdpe: A very robust estimator for model fitting and range image segmentation. *International Journal of Computer Vision*, 59(2):139–166, 2004.

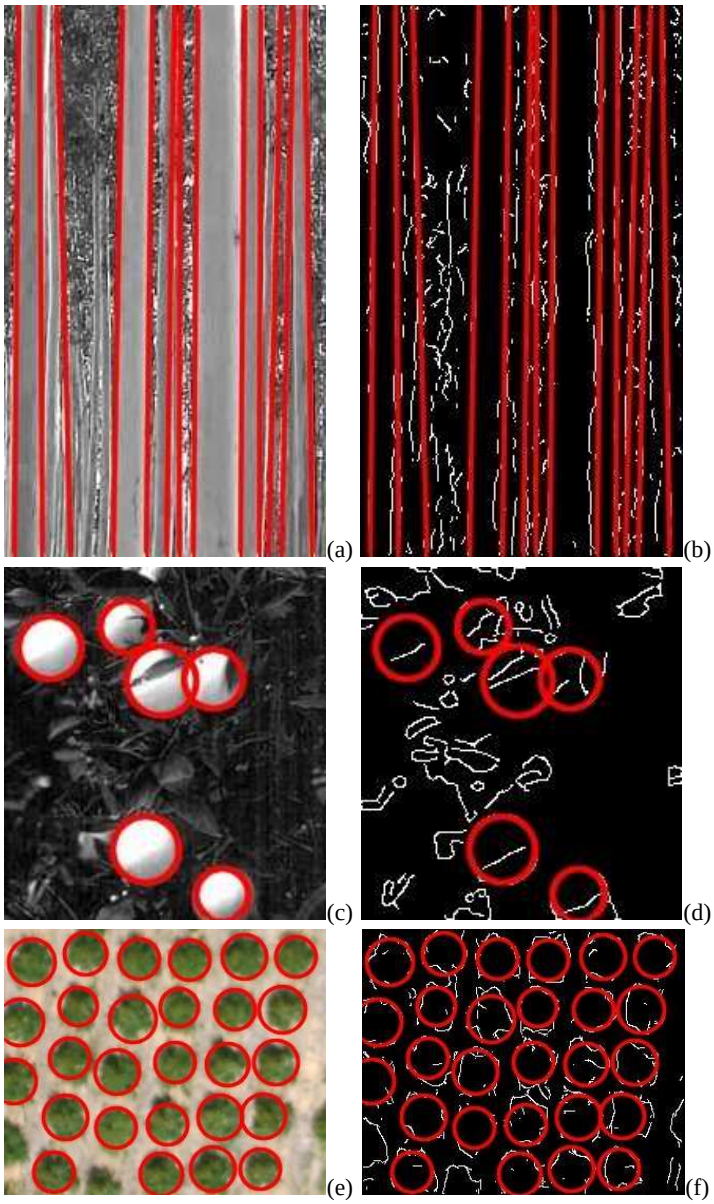


**Fig. 6.** Detected lines in agricultural images using the experimental setup of Section 4 and edge detection.