

A Robust Multi-Camera 3D Ellipse Fitting for Contactless Measurements

Filippo Bergamasco, Luca Cosmo, Andrea Albarelli and Andrea Torsello

Dipartimento di Scienze Ambientali, Informatica e Statistica

Università Ca' Foscari - Venice, Italy

Email: bergamasco@dais.unive.it lcosmo@dais.unive.it albarelli@unive.it torsello@dais.unive.it

Abstract

Ellipses are a widely used cue in many 2D and 3D object recognition pipelines. In fact, they exhibit a number of useful properties. First, they are naturally occurring in many man-made objects. Second, the projective invariance of the class of ellipses makes them detectable even without any knowledge of the acquisition parameters. Finally, they can be represented by a compact set of parameters that can be easily adopted within optimization tasks. While a large body of work exists in the literature about the localization of ellipses as 2D entities in images, less effort has been put in the direct localization of ellipses in 3D, exploiting images coming from a known camera network. In this paper we propose a novel technique for fitting elliptical shapes in 3D space, by performing an initial 2D guess on each image followed by a multi-camera optimization refining a 3D ellipse simultaneously on all the calibrated views. The proposed method is validated both with synthetic data and by measuring real objects captured by a specially crafted imaging head. Finally, to evaluate the feasibility of the approach within real-time industrial scenarios, we tested the performance of a GPU-based implementation of the algorithm.

1. Introduction

Among all the visual cues, ellipses offer several advantages that prompt their adoption within many machine vision tasks. To begin with, the class of ellipses is invariant to projective transformations, thus an elliptical shape remains so when it is captured from any viewpoint by a pinhole camera [4]. This property makes easy to recognize objects that contain ellipses [11, 8] or partially elliptical features [18]. When the parameters of one or more coplanar 3D ellipses that originated the projection are known, the class of homographies that make it orthonormal to the image plane can be retrieved. This is a useful step for many tasks, such as the recognition of fiducial markers [1, 13], orthonormalization of playfields [7], forensic analysis of organic stains [20] or any other planar metric rectification [2]. Furthermore, el-

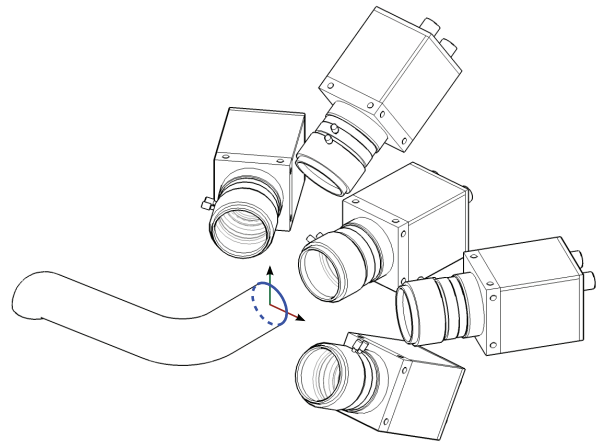


Figure 1. Schematic representation of a multi-camera system for industrial in-line pipes inspection.

lipses (including circles) are regular shapes that often appear in manufactured objects and can be used as optical landmarks for tracking and manipulation [22] or measured for accurate in-line quality assurance [16].

Because of their usefulness and broad range of applicability, it is not surprising that ellipse detection and fitting methods abound in the literature. In particular, when points belonging to the ellipse are known, they are often fitted through ellipse-specific least square methods [6]. In order to find co-elliptical points in images, traditional parameter-space search schemas, such as RANSAC or Hough Transform, can be employed. Unfortunately, the significantly high dimensionality of 2D ellipse parametrization (which counts 5 degrees of freedom) makes the direct application of those techniques not feasible. For this reason a lot of efficient variants have appeared. Some try to reduce the number of samples for a successful RANSAC selection [17, 21]. Others attempt to escape from the curse of dimensionality that plagues the Hough accumulator [12, 3]. If high accuracy is sought, point-fitted ellipses can be used as an initial guess to be refined through intensity-based methods. Those approaches allow to obtain a sub-pixel estimation by exploiting the raw gradient of the image [14] or by preserv-

ing quantities such as intensity moments and gradients [9]. Multiple view geometry has also been exploited to get a better 3D ellipse estimation. In [19], multiple cameras are used to track an elliptical feature on a glove to obtain the estimation of the hand pose. The ellipses fitted in the images are triangulated with the algorithm proposed in [15] and the best pair is selected. In [10], holes in metal plates and industrial components are captured by a couple of calibrated cameras and the resulting conics are then used to reconstruct the hole in the Euclidean space. Also in [5] the intersection of two independently extracted conics is obtained through a closed form. All these approaches, however, exploit 3D constraints in an indirect manner, as triangulation always happens on the basis of the ellipses fitted over 2D data.

In this paper we present a rather different technique that works directly in 3D space. Specifically, we adopt a parametric level-set approach, where the parameters of a single elliptical object that is observed by a calibrated network of multiple cameras (see Fig.1) are optimized with respect to an energy function that simultaneously accounts for each point of view. The goal of our method is to bind the 2D intensity and gradient-based energy maximization that happens within each image to a common 3D ellipse model. The performance of the solution has been assessed through both synthetic experiment and by applying it to a real world scenario. Finally, to make the approach feasible regardless of the high computational requirements, we propose a GPU implementation which performance has been compared with a well optimized CPU-based version.

2. Multiple Camera Ellipse Fitting

In our approach we are not seeking for independent optima over each image plane, as is the case with most ellipse fitting methods. Rather, our search domain is the parametrization of an ellipse in the 3D Euclidean space, and the optimum is sought with respect to its concurrent 2D reprojections over the captured images. In order to perform such optimization we need to sort out a number of issues. The first problem is the definition of a 3D ellipse parametrization that is well suitable for the task (that is, it makes easy to relate the parameters with the 2D projections). The second one, is the definition of an energy function that is robust and accounts for the usual cues for curve detection (namely the magnitude and direction of the intensity gradient). The last issue is the computation of the derivative of the energy function with respect to the 3D ellipse parameters to be able to perform a gradient descent.

2.1. Parameterization of the 3D Ellipse

In its general case, any 2-dimensional ellipse in the image plane is defined by 5 parameters, namely: the length of the two axes, the angle of rotation and a translation vector with respect to the origin.

In matrix form it can be expressed by the locus of points $\vec{x} = (x_1 \ x_2 \ 1)^T$ in homogeneous coordinates for which the equation $\vec{x}^T A \vec{x} = 0$ holds, for

$$A = \begin{pmatrix} a & b & d \\ b & c & f \\ d & f & g \end{pmatrix} \quad (1)$$

with $\det(A) < 0$ and $ac - b^2 > 0$.

In the 3-dimensional case it is subjected to 3 more degrees of freedom (i.e. rotation around two more axes and the z-component of the translation vector). More directly, we can define the ellipse by first defining the plane T it resides on and then defining the 2D equation of the ellipse on a parametrization of such plane. In particular, let $\vec{c} = (c_1, c_2, c_3, 1)^T \in T$ be the origin of the parametrization, and $\vec{u} = (u_1, u_2, u_3, 0)^T$, $\vec{v} = (v_1, v_2, v_3, 0)^T$ be the generators of the linear subspace defining T , then each point on the 3D ellipse will be of the form $\vec{\sigma} + \alpha\vec{u} + \beta\vec{v}$ with α and β satisfying the equation of an ellipse.

By setting the origin $\vec{\sigma}$ to be at the center of the ellipse and selecting the directions \vec{u} and \vec{v} appropriately, we can transform the equation of the ellipse on the plane coordinates in such a way that it will take the form of the equation of a circle. Hence, allowing the 3D ellipse to be fully defined by the parametrization of the plane on which the ellipse resides. However, this representation has still one more parameter than the actual degrees of freedom of the ellipse. To solve this we can, without any loss of generality, set $u_3 = 0$, thus, by defining the matrix

$$\mathbf{U}_c = \begin{pmatrix} u_1 & v_1 & c_1 \\ u_2 & v_2 & c_2 \\ 0 & v_3 & c_3 \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

and the vector $\vec{x} = (\alpha, \beta, 1)^T$, we can express any point p in the 3D ellipse as:

$$\vec{p} = \mathbf{U}_c \vec{x} \quad \text{subject to} \quad \vec{x}^T \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \vec{x} = 0. \quad (3)$$

Even if \mathbf{U}_c embeds all the parameters needed to describe any 3d ellipse, it is often the case that an explicit representation through center \vec{c} and axes $\vec{a}_1, \vec{a}_2 \in R^3$ is needed. Let \mathbf{U} be the 3×2 matrix composed by the first two columns of \mathbf{U}_c . The two axes \vec{a}_1, \vec{a}_2 can be extracted as the two columns of the matrix:

$$\mathbf{K} = \begin{pmatrix} | & | \\ \vec{a}_1 & \vec{a}_2 \\ | & | \end{pmatrix} = \mathbf{U} \phi^T$$

where ϕ^T is the matrix of left singular vectors of $\mathbf{U}^T \mathbf{U}$ computed via SVD decomposition. The vector \vec{c} is trivially composed by the parameters $(c_1 \ c_2 \ c_3)^T$.

Conversely, from two axes \vec{a}_1, \vec{a}_2 , the matrix \mathbf{U} can be expressed as:

$$\mathbf{U} = \mathbf{K} \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix}$$

by imposing that $\begin{cases} \alpha K_{31} + \beta K_{32} = 0 \\ \alpha^2 + \beta^2 = 1 \end{cases}$. Finally, once \mathbf{U} has been computed, the 3D ellipse matrix can be composed in the following way:

$$\mathbf{U}_c = \begin{pmatrix} \mathbf{U} & \vec{c} \\ \vec{0} & 1 \end{pmatrix}$$

Finally, with this parametrization it is very easy to obtain the equation of the ellipse projected onto any camera. Given a projection matrix \mathbf{P} , the matrix \mathbf{A}_P describing the 2-dimensional ellipse after the projection can be expressed as:

$$\mathbf{A}_P = (\mathbf{P}\mathbf{U}_c)^{-T} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} (\mathbf{P}\mathbf{U}_c)^{-1} \quad (4)$$

2.2. Energy Function over the Image

To estimate the equation of the 3D-ellipse we set-up a level-set based optimization schema that updates the ellipse matrix \mathbf{U}_c by simultaneously taking into account its re-projection in every camera of the network. The advantages of this approach are essentially threefold. First, the equation of the 3D ellipse estimated and the re-projection in all cameras are always consistent. Second, erroneous calibrations that affects the camera network itself can be effectively attenuated, as shown in the experimental section. Third, the ellipse can be partially occluded in one or more camera images without heavily hindering the fitting accuracy.

In order to evolve the 3D ellipse geometry to fit the observation, we need to define the level set functions $\varphi_i : R^2 \rightarrow R$ describing the shape of the ellipse \mathbf{U}_c re-projected to the i^{th} camera. Given each level set, we cast the multi-view fitting problem as the problem of maximizing the energy function:

$$E_{I_1 \dots I_n}(\mathbf{U}_c) = \sum_{i=1}^n E_{I_i}(\mathbf{U}_c) \quad (5)$$

Which sums the energy contributions of each camera:

$$E_{I_i}(\mathbf{U}_c) = \int_{R^2} \langle \nabla H(\varphi(\vec{x})), \nabla I_i(\vec{x}) \rangle^2 dx \quad (6)$$

$$= \int_{R^2} \langle H'(\varphi(\vec{x})) \nabla \varphi(\vec{x}), \nabla I_i(\vec{x}) \rangle^2 dx, \quad (7)$$

where H is a suitable relaxation of the Heavyside function. In our implementation, we used:

$$H(t) = \frac{1}{1 + e^{-\frac{t}{\sigma}}} \quad (8)$$

where parameter σ models the band size (in pixels) of the ellipse region to be considered. By varying σ we can manage the trade-off between the need of a regularization term in the energy function to handle noise in the image gradient and the estimation precision that has to be achieved.

The level set for a generic ellipse is rather complicated and cannot be easily expressed in closed form, however, since it appears only within the Heavyside function and its derivative, we only need to have a good analytic approximation in the boundary around the ellipse. We approximate the level set in the boundary region as:

$$\varphi_i(\vec{x}) \approx \frac{\vec{x}^T \mathbf{A}_i \vec{x}}{2\sqrt{\vec{x}^T \mathbf{A}_i^T \mathbf{I}_0 \mathbf{A}_i \vec{x}}} \quad (9)$$

Where $\mathbf{I}_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ and \mathbf{A}_i is the re-projection of the ellipse \mathbf{U}_c into the i^{th} camera computed using equation (4). The function has negative values outside the boundaries of the ellipse, positive values inside and is exactly 0 for each point $\{\vec{x} | \vec{x}^T \mathbf{U}_c \vec{x} = 0\}$.

The gradient of the level set function $\nabla \varphi : R^2 \rightarrow R^2$ can actually be defined exactly in closed form:

$$\nabla \varphi_i(\vec{x}) = \frac{\mathbf{A}_i \vec{x}}{\sqrt{\vec{x}^T \mathbf{A}_i^T \mathbf{I}_0 \mathbf{A}_i \vec{x}}} \quad (10)$$

Starting from an initial estimation, given by a simple triangulation of 2d-ellipses between just two cameras, we maximize the energy function (5) over the plane parameters \mathbf{U}_c by means of a gradient scheme.

2.3. Gradient of the Energy Function

The gradient of the energy function can be computed as a summation of the gradient of each energy term. This gradient can be obtained by analytically computing the partial derivatives of equation (6) with respect to the eight parameters $(p_1 \dots p_8) = (u_1, v_1, c_1, u_2, v_2, c_2, v_3, c_3)$:

$$\begin{aligned} \frac{\partial}{\partial p_i} E_{I_i}(\mathbf{U}_c) &= \frac{\partial}{\partial p_i} \int_{R^2} E_{I_i}(\mathbf{U}_c, \vec{x})^2 dx \\ &= \int_{R^2} 2E_{I_i}(\mathbf{U}_c, \vec{x}) \frac{\partial}{\partial p_i} E_{I_i}(\mathbf{U}_c, \vec{x}) dx \end{aligned}$$

Where:

$$E_{I_i}(\mathbf{U}_c, \vec{x}) = \langle H'(\varphi(\vec{x})) \nabla \varphi(\vec{x}), \nabla I_i(\vec{x}) \rangle$$

and

$$\begin{aligned} \frac{\partial}{\partial p_i} E_{I_i}(\mathbf{U}_c, \vec{x}) &= \left(\frac{\partial}{\partial p_i} H'(\varphi(\vec{x})) \right) \langle \nabla \varphi(\vec{x}), \nabla I_i(\vec{x}) \rangle + \\ &+ H'(\varphi(\vec{x})) \left\langle \left(\frac{\partial}{\partial p_i} \nabla \varphi(\vec{x}) \right), \nabla I_i(\vec{x}) \right\rangle. \end{aligned}$$

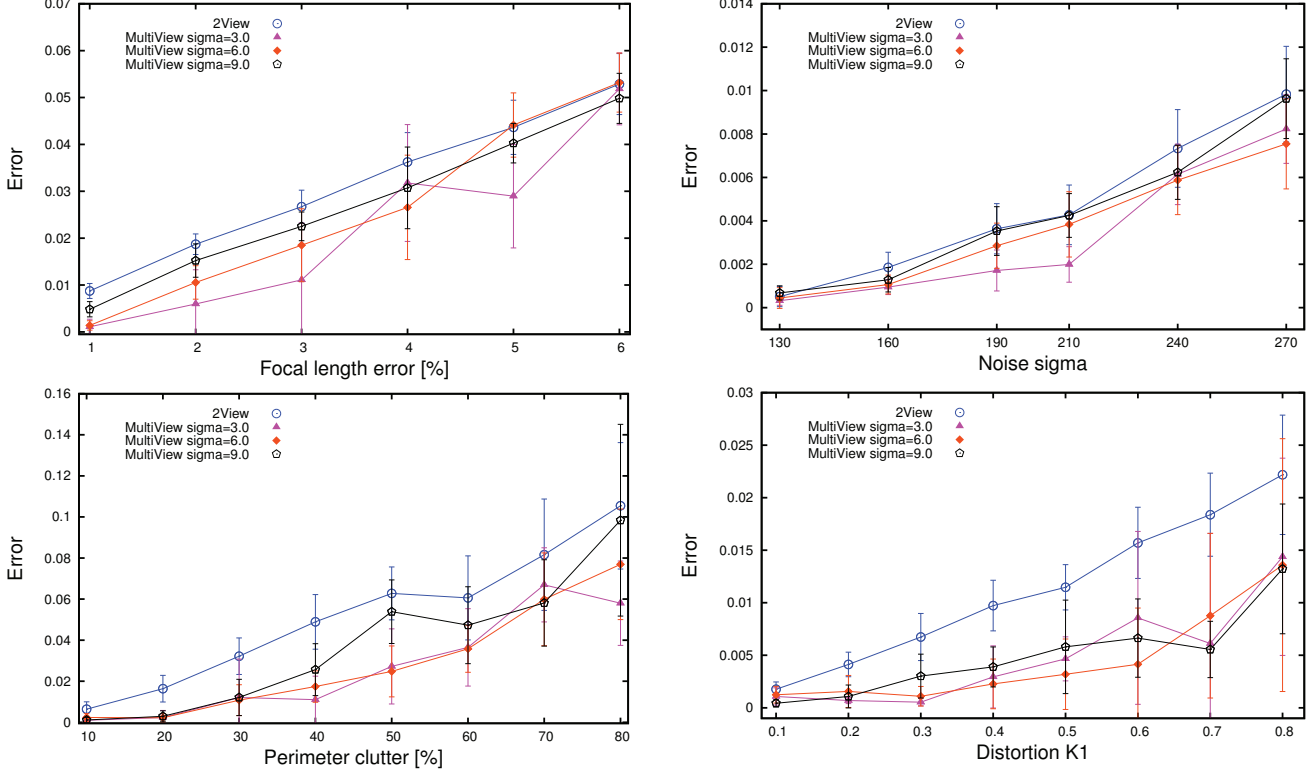


Figure 2. Evaluation of the accuracy of the proposed method with respect to different noise sources. The metric adopted is the relative error between the minor axis of the ground truth and of the fitted ellipse.

The derivatives of the parametric level set functions can be computed analytically. At the beginning of each iteration we compute the derivative of the projected ellipse matrices \mathbf{A}_i which are constant with respect to \vec{x} :

$$\frac{\partial}{\partial p_i} \mathbf{A}_i = \mathbf{T} + \mathbf{T}^T \quad (11)$$

where

$$\mathbf{T} = \left(\frac{\partial}{\partial p_i} [(\mathbf{P}_i \mathbf{U}_c)^{-1}] \right)^T \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} (\mathbf{P}_i \mathbf{U}_c)^{-1} \quad (12)$$

and

$$\frac{\partial}{\partial p_i} [(\mathbf{P}_i \mathbf{U}_c)^{-1}] = -(\mathbf{P}_i \mathbf{U}_c)^{-1} (\mathbf{P}_i \frac{\partial}{\partial p_i} \mathbf{U}_c) (\mathbf{P}_i \mathbf{U}_c)^{-1}. \quad (13)$$

Then, using (11), we can compute the level set derivatives for each pixel:

$$\frac{\partial}{\partial p_i} \nabla \varphi(\vec{x}) = \frac{\left(\frac{\partial}{\partial p_i} \mathbf{A}_i \right) \vec{x}}{\sqrt{\vec{x}^T \mathbf{A}_i^T \mathbf{I}_0 \mathbf{A}_i \vec{x}}} - \frac{\mathbf{A}_i \vec{x} (\vec{x}^T \left(\frac{\partial}{\partial p_i} \mathbf{A}_i \right)^T \mathbf{I}_0 \mathbf{A}_i \vec{x} + \vec{x}^T \mathbf{A}_i^T \mathbf{I}_0 \left(\frac{\partial}{\partial p_i} \mathbf{A}_i \right) \vec{x})}{2(\vec{x}^T \mathbf{A}_i^T \mathbf{I}_0 \mathbf{A}_i \vec{x})^{\frac{3}{2}}} \quad (14)$$

$$\frac{\partial}{\partial p_i} \varphi(\vec{x}) = \frac{1}{2} \langle \vec{x}, \frac{\partial}{\partial p_i} \nabla \varphi(\vec{x}) \rangle \quad (15)$$

$$\frac{\partial}{\partial p_i} H'(\varphi(\vec{x})) = H''(\varphi(\vec{x})) \frac{\partial}{\partial p_i} \varphi(\vec{x}). \quad (16)$$

By summing the derivative $\frac{\partial}{\partial p_i} E_{I_i}(\mathbf{U}_c, \vec{x})$ over all images and all pixels in the active band in each image, we obtain the gradient $\mathbf{G} = \nabla E_{I_1 \dots I_n}(\mathbf{U}_c)$. At this point, we update the 3D ellipse matrix \mathbf{U}_c through the gradient step

$$\mathbf{U}_c^{(t+1)} = \mathbf{U}_c^{(t)} + \eta \mathbf{G} \quad (17)$$

where η is a constant step size.

3. Experimental evaluation

We evaluated the proposed approach both on a set of synthetic tests and on a real world quality control task where we measure the diameter of a pipe with a calibrated multi-camera setup. In both cases, lacking a similar 3D based optimization framework, we compared the accuracy of our method with respect to the results obtained by triangulating ellipses optimally fitted over the single images. The rationale of the synthetic experiments is to be able to evaluate the accuracy of the measure with an exactly known ground truth (which is very difficult to obtain on real objects with very high accuracy). Further, the synthetically generated

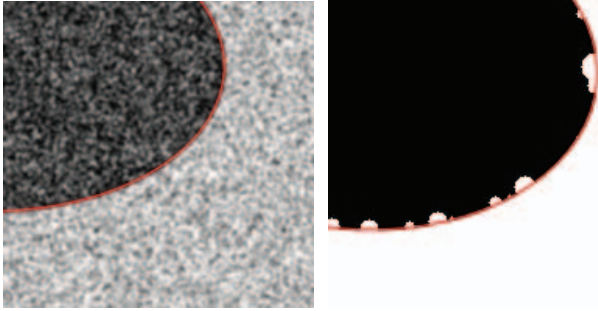


Figure 3. Examples of images with artificial noise added. Respectively additive Gaussian noise and blur in the left image and occlusion in the right image. The red line shows the fitted ellipse.

imagery permits us to control the exact nature and amount of noise, allowing for a separate and independent evaluation for each noise source. By contrast, the setup employing real cameras does not give an accurate control over the scene, nevertheless it is fundamental to assess the ability of the approach to deal with the complex set of distractors that arise from the imaging process (such as reflections, variable contrast, defects of the object, bad focusing and so on). In both cases the ellipse detection is performed by extracting horizontal and vertical image gradients with an oriented derivative of Gaussian filter. Edge pixels are then found by non-maxima suppression and by applying a very permissive threshold (no hysteresis is applied). The obtained edge pixels are thus grouped into contiguous curves, which are in turn fitted to find ellipses candidates. The candidate that exhibit the higher energy is selected and refined using [14]. The refined ellipses are then triangulated using the two images that score the lower triangulation error. The obtained 3D ellipse is finally used both as the result of the baseline method (labeled as *2view* in the following experiments) and as the initialization ellipse for our refinement process (labeled as *multiview*). All the experiments have been performed with 3Mp images and the processing is done with a modern 3.2 Ghz Intel Core i7 PC equipped with Windows 7 Operating System. The CPU implementation was written in C++ and the GPU implementation uses the CUDA library. The video card used was based on the Nvidia 670 chipset with 1344 CUDA cores.

3.1. Synthetic Experiments

For this set of experiments we chose to evaluate the effect of four different noise sources over the optimization process. Specifically, we investigated the sensitivity of the approach to errors on the estimation of the focal length and of the radial distortion parameters of the camera and the influence of Gaussian noise and clutter corrupting the images. In Fig. 3 examples of Gaussian noise and clutter are shown (note that these are details of the images, in the experiments the ellipse was viewed in full). For each test we created 5 synthetic snapshots of a black disc as seen from 5

different cameras looking at the disk from different points of view (see Fig. 1 and Fig. 4). The corruption by Gaussian noise has been produced by adding to each pixel a normal distributed additive error of variable value of σ , followed by a blurring of the image with a Gaussian kernel with $\sigma = 6$. The artificial clutter has been created by occluding the perimeter of the disc with a set of random white circles until a given percentage of the original border was corrupted. This simulates the effect of local imaging effect such as the presence of specular highlights that severely affect the edge detection process. The focal length error was obtained by changing the correct focal length of the central camera by a given percentage. Finally, the distortion error was introduced by adding an increasing amount to the correct radial distortion parameter $K1$. In Fig. 2 we show the results obtained using the baseline triangulation and our optimization with different values of the parameter σ used for the heavyside function (respectively 3, 6 and 9 pixels). As expected, in all the tests performed the relative error grows with the level of noise. In general, all the methods seem to be minimally sensitive to Gaussian noise, whereas the clutter has a big effect even at low percentages. The baseline method performs consistently worse and, among the multi-view configurations, the one with lower heavyside band appears to be the most robust for almost all noise levels. This is probably due to the fact that the images have already been smoothed by the gradient calculation step, and thus further smoothing is not required and, to some degree, leads to a more prominent signal displacement.

3.2. Real World Application

For the experiments with real images we built an imaging device that hold 5 PointGrey Flea3 3.2Mp Monochrome USB3 machine vision cameras (see Fig. 4). The 5 cameras were calibrated for both intrinsic and extrinsic parameters.



Figure 4. The experimental Multiple-camera imaging head.

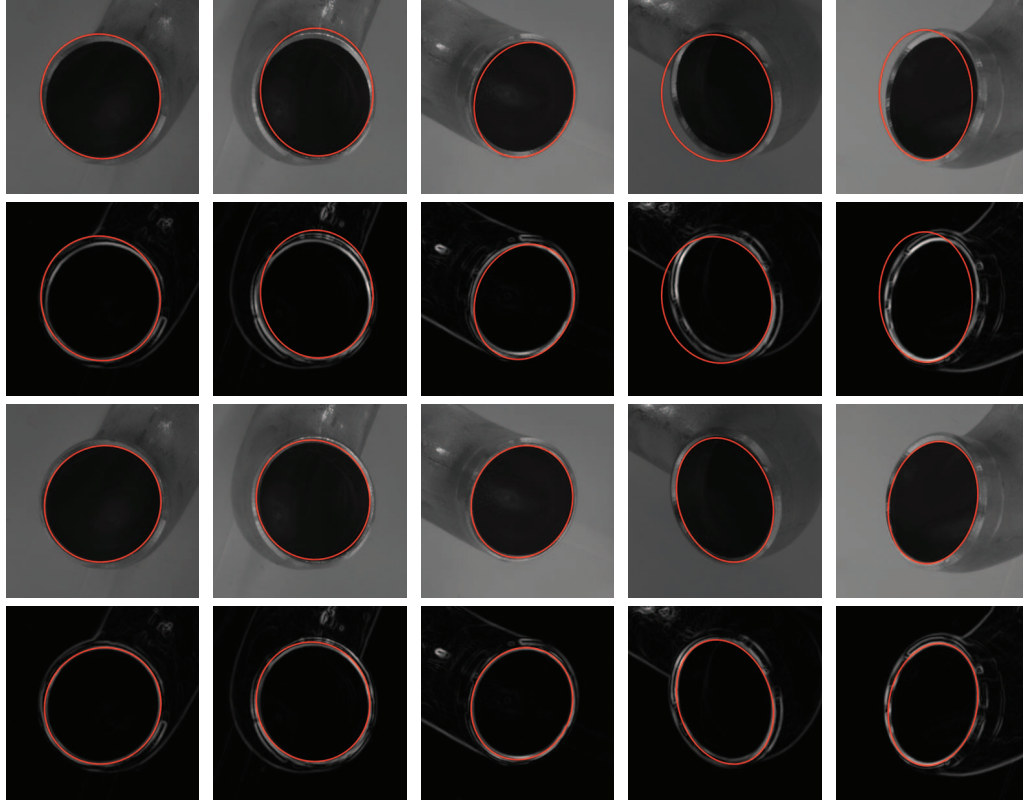


Figure 5. Comparison between the accuracy of the initial 2D fitting and the proposed 3D optimization.

We used an aluminium pipe for air conditioning system as the object to be measured, and the imaging head has been supplemented with four high power leds in order to get an even illumination of the rim. This is a typical scenario for in-line inspection in manufacturing lines. Additionally, the smooth and polished surface of the pipe offers especially challenging conditions for ellipse detection and refinement, since reflections and changes of contrast tend to create a lot of false elliptical sectors and some highly structured noise. If Fig. 5 a complete qualitative example of the refinement

process is shown. In the first two rows of the figure the re-projection of the initially estimated 3D ellipse is overlaid to both the original images and the intensity-coded gradient magnitude. In the remaining rows the re-projection of the optimized 3D ellipse is overlaid over the same images. The images used for the initial triangulation in this specific case were the first and the third. Specifically, the initial guess for the first image was a slightly off-center ellipse fitted in between the edge response produced by the inner and outer rims of the pipe opening (see the gradient image). As a matter of fact, it is immediate to note that these two images exhibits the lower reprojection error, especially for the central camera. However, the other reprojections are rather grossly misaligned with the remaining three points of view. By contrast, almost all the misalignment has been corrected after performing the 3D refinement procedure. While some degree of displacement is still visible in some images, we think that this is mainly due to miscalibration of the extrinsic parameters of the imaging head.

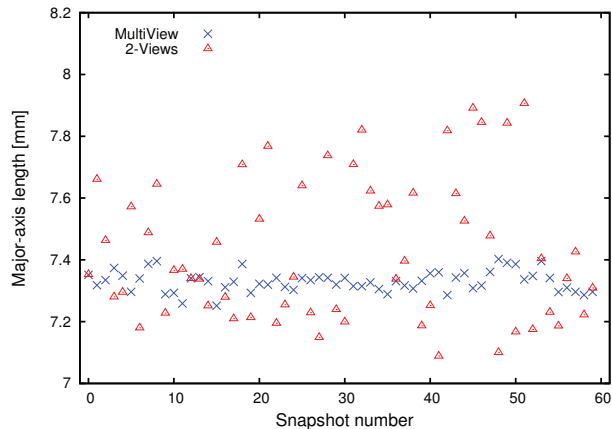


Figure 6. Quantitative assessment of the improvement in accuracy.

We manually measured the internal and external diameter of the pipe with a caliper (with $\pm 0.1\text{mm}$ accuracy) obtaining respectively 13.9 and 16.1 mm. However, since the optimization process aim to converge toward the middle of the two rims, it would make no sense to evaluate directly the measurement error committed. Still, the standard deviation of the data with respect to several subsequent measures of

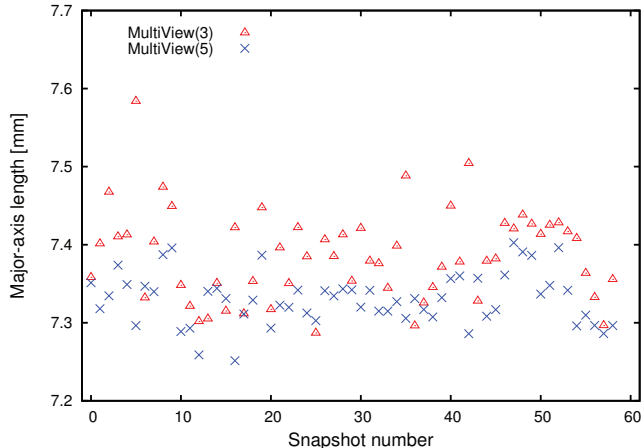


Figure 7. Effect of the number of views over the measure quality.

the same object from slightly different angles can be considered a good indication of measurement error. Indeed, even if the final measure can be affected by systematic errors, they can be estimated and corrected a-posteriori. In Fig. 6 we plotted the measured length of the major axis of the detected 3D ellipse for 60 repeated shots of the pipe opening. The improvement in uncertainty reduction after the refinement step is clearly noticeable as the variance of the measurements is strongly reduced. Indeed, the standard deviation went from 0.23 to 0.03.

All the refinements performed so far have been conducted using 5 points of view. In order to complete our tests it would have been interesting to evaluate if similar accuracy could be obtained using a smaller number of cameras. To this end we disabled two cameras and took further 60 shots of the pipe. The results are plotted in Fig. 7. While the dispersion of the measurements is a little higher using only three points of view, it is still noticeably smaller than the one obtained without the optimization step (note that the scales of Fig. 6 and Fig. 7 are different).

4. GPU-based Implementation

In a naive implementation, the optimization scheme proposed is quite intensive in terms of raw computing power. Especially for the gradient computation, which requires several matrix and vector multiplications that may easily sum up to an unacceptable total computation time.

However, the intrinsic structure of the problem leads naturally to an implementation in which every pixel in the area of interest defined by the Heaviside function are computed in parallel. After this computation, that can be performed with no required synchronization between each view, a reduction step is needed to aggregate all terms and obtain the final value of the energy and gradient in each iteration.

We implemented the algorithm in C++ with no additional external libraries except for OpenCV for image IO,

OpenMP for CPU parallelization and CUDA for GPU computing. Both the CPU and GPU based implementations are essentially the same, except for the fact that the latter can exploit the massive computing power of modern graphics cards. For every algorithm's iteration, a CPU-based function computes a list of pixel for each image that will be affected by the computation. This list is generated by considering a band around each 2d-ellipse reprojection with a thickness of 5σ pixels and is uploaded to the device memory, together with the optimized parameters and the pre-computed image gradient for each pixel in the list. Once the upload is completed, all available stream processors are used to compute the energy and the energy gradient terms. At the end of the computation steps, all threads are synchronized and 9 values are reduced (energy and the 8 terms of the gradient) to obtain the final values. The total energy is used to track the optimization status and trigger a termination criteria, the gradient is used to adjust the 3d ellipse that is being optimized, moving toward a local maxima.

We tested the execution time per iteration for both the CPU and GPU based implementation of our algorithm (see Fig.8) with respect to the average number of pixel processed. In both cases, the process is fast enough to handle a real-time optimization in 3 megapixels images with the fitted ellipse spanning into about 50% of the image. As expected, the GPU implementation performs better than the CPU and exhibits a more consistent running time throughout the tests. This is probably due to the fact that we are dealing with a dedicated hardware. Finally, the synchronization overhead caused by the reductions decreases the performance gap between the two implementations when a relatively low number of pixels are processed, which in turn becomes dramatic when an optimization involving more than 10^5 pixels is needed.

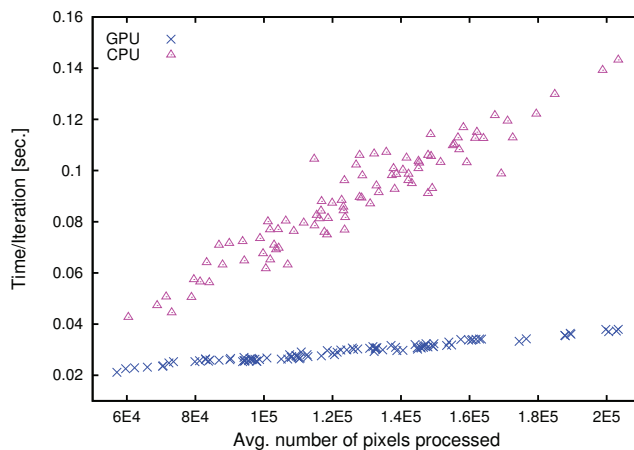


Figure 8. Comparison between the running time of the CPU and GPU-based implementations of the multiview algorithm. Times are plotted with respect to the number of pixels in the evaluated mask (i.e. size of the ellipse to be refined).

5. Conclusions

In this paper we presented a novel approach for ellipse fitting that exploits multiple simultaneous calibrated views of the same physical elliptical object. The proposed technique starts by obtaining an initial estimation of the 3D ellipse using 2D based fitting methods followed by a pairwise triangulation. This initial guess is then refined by moving its parameters according to an intensity-based level set approach that accounts for all the images simultaneously. To this end, a specially crafted parametrization and an apt energy function have been introduced. The effectiveness of the approach has been evaluated through an extensive experimental section that shows its resilience to a wide range of noise sources and compares the accuracy obtained with respect to a baseline approach. Finally, a running time analysis of our GPU-based implementation of the algorithm shows that the proposed method can be effectively used in real-time applications.

References

- [1] F. Bergamasco, A. Albarelli, E. Rodola, and A. Torsello. Rune-tag: A high accuracy fiducial marker with strong occlusion resilience. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '11*, pages 113–120, Washington, DC, USA, 2011. IEEE Computer Society.
- [2] Y. Chen and H. H. S. Ip. Planar metric rectification by algebraically estimating the image of the absolute conic. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4 - Volume 04, ICPR '04*, pages 88–91, Washington, DC, USA, 2004. IEEE Computer Society.
- [3] A. Chia, M. Leung, H.-L. Eng, and S. Rahardja. Ellipse detection with hough transform in one dimensional parametric space. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 5, pages V –333 –V –336, 16 2007-oct. 19 2007.
- [4] H. S. M. Coxeter. *Projective Geometry, 2nd ed.* Springer Verlag, 2003.
- [5] Y. Dufournaud, R. Horaud, and L. Quan. Robot Stereo-hand Coordination for Grasping Curved Parts. In J. N. Carter and M. S. Nixon, editors, *9th British Machine Vision Conference (BMVC '98)*, volume 2, pages 760–769, Southampton, Royaume-Uni, 1998. British Machine Vision Association.
- [6] A. Fitzgibbon, M. Pilu, and R. Fisher. Direct least square fitting of ellipses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):476–480, may 1999.
- [7] A. Gupta, J. Little, and R. Woodham. Using line and ellipse features for rectification of broadcast hockey video. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, pages 32–39, may 2011.
- [8] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(3):478–500, Mar. 2010.
- [9] J. Heikkila. Moment and curvature preserving technique for accurate ellipse boundary detection. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 1, pages 734–737 vol.1, aug 1998.
- [10] S. Malassiotis and M. Strintzis. Stereo vision system for precision dimensional inspection of 3d holes. *Machine Vision and Applications*, 15:101–113, 2003.
- [11] S. Martelli, R. Marzotto, A. Colombari, and V. Murino. Fpga-based robust ellipse estimation for circular road sign detection. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 53–60, june 2010.
- [12] R. A. McLaughlin. Randomized hough transform: Improved ellipse detection with comparison. *Pattern Recognition Letters*, 19(34):299–305, 1998.
- [13] L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality, ISMAR '02*, pages 27–, Washington, DC, USA, 2002. IEEE Computer Society.
- [14] J. Ouellet and P. Hebert. Precise ellipse estimation without contour point extraction. *Mach. Vision Appl.*, 21, 2009.
- [15] L. Quan. Conic reconstruction and correspondence from two views. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(2):151–160, feb 1996.
- [16] D. Schleicher and B. Zagar. Image processing to estimate the ellipticity of steel coils using a concentric ellipse fitting algorithm. In *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, pages 884–890, oct. 2008.
- [17] G. Song and H. Wang. A fast and robust ellipse detection algorithm based on pseudo-random sample consensus. In *Computer Analysis of Images and Patterns*, volume 4673 of *Lecture Notes in Computer Science*, pages 669–676. Springer Berlin / Heidelberg, 2007.
- [18] P. Srestasathien and A. Yilmaz. Planar shape representation and matching under projective transformation. *Computer Vision and Image Understanding*, 115(11):1525–1535, 2011.
- [19] J. Usabiaga, A. Erol, G. Bebis, R. Boyle, and X. Twombly. Global hand pose estimation by multiple camera ellipse tracking. *Machine Vision and Applications*, 21(1):1–15, 2009.
- [20] J. Wright, A. Wagner, S. Rao, and Y. Ma. Homography from coplanar ellipses with application to forensic blood splatter reconstruction. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1250–1257, june 2006.
- [21] Y. Xie and J. Ohya. Efficient detection of ellipses from an image by a guided modified ransac. In *Image Processing: Algorithms and Systems*, volume 7245 of *SPIE Proceedings*. SPIE, 2009.
- [22] Y. Yoon, G. DeSouza, and A. Kak. Real-time tracking and pose estimation for industrial objects using geometric features. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 3, pages 3473–3478 vol.3, sept. 2003.