# FitEllipsoid: a fast supervised ellipsoid segmentation plugin

Jérôme Fehrenbach, Bastien Kovac & Pierre Weiss

*ITAV and IMT, CNRS and Université de Toulouse, France.*

**Abstract**

We propose a supervised segmentation algorithm dedicated to fitting ellipsoids in 3D images with a minimal amount of user inputs. Its computational core is based on an original approach to fit ellipsoids to point clouds in an affine invariant manner. The algorithm estimates the fitting ellipsoid in a small number of cheap iterations. The code is distributed as an open-source plugin to be used within the image analysis software Icy. Computational experiments show that collections of nuclei in fluorescence microscopy images can be segmented much faster than with more traditional 2D delineation approaches.

*Keywords:* Supervised segmentation, ellipsoid fitting, Icy plugin, open-source software, fast convex optimization
*PACS:* 07.05.Pj, 87.57.nm, 87.85.Ng
*2000 MSC:* 49M29, 90C30, 65D18

## 1. Introduction

*Motivation.* Segmenting ellipsoidal structures in 2D or 3D images is a very common problem in biomedical imaging. The results can be used to understand the geometry of organs (e.g. lungs, kidneys), tumors, cells or nuclei (Lockett et al., 1998; Okada et al., 2005; Mahdavi and Salcudean, 2008), or serve as an initialization for more advanced algorithms such as active contours (Dufour et al., 2005; Thevenaz et al., 2011; Cuingnet et al., 2012; Delgado-Gonzalo et al., 2013).

While fully automatic detection algorithms (Olivo-Marin, 2002; Jaqaman et al., 2008; Soubies et al., 2013; Zhang et al., 2016) are probably the ideal tool to reduce human work and mistakes, our recent experience suggests that existing strategies are not sufficient to provide convincing segmentation results when images suffer from strong degradations (e.g. blur, noise, low resolution) or contain densely packed objects. In that case, humans are often superior to machines. In addition, automatic methods usually require tuning a few parameters, which may be more time consuming than a simple supervised segmentation algorithm. Finally, generating learning databases or gold standards to test and compare existing segmentation algorithms still requires the use of supervised algorithms. Unfortunately, to the best of our knowledge, there currently exists no such freely available tool, which would benefit different communities.

*Contributions.* These few considerations motivated us developing two simple plugins for the Icy image analysis software (De Chaumont et al., 2012). They allow users fitting ellipses or ellipsoids in 2D or 3D images. The objective of this paper is describe the plugin for 3D ellipsoids, and present the algorithm that was implemented and compare it with other approaches.

For the specific task of fitting ellipsoids, the proposed method is about 5 times faster than the best rivals we found based on slice-by-slice delineation. This efficiency is obtained thanks to an intuitive user interaction and to the fact that our algorithm focuses on finding ellipsoids and not more complex surfaces.

The core of the algorithm consists of solving the well studied problem of ellipsoid fitting from point clouds. This is a notoriously difficult problem that attracted the attention of researchers from different fields such as computer vision, statistics or numerical analysis, to name a few (Gander et al., 1994; Fitzgibbon et al., 1999; Nievergelt, 2001; Ahn et al., 2001; Calafiore, 2002; Ahn et al., 2002; Markovsky et al., 2004; Li and Griffiths, 2004; Chernov and Lesort, 2005; Kleinsteuber and Hüper, 2010; Ying et al., 2012; Saunderson et al., 2012; Kanatani et al., 2016; Lin and Huang, 2016). We propose an original, lightweight and robust computational algorithm that shares the same spirit as a recent work (Lin and Huang, 2016), but significantly outperforms it when the ellipsoids are not centered or anisotropic. An

important feature of the proposed algorithm is affine invariance: the point cloud is centered and reduced prior to computation, ensuring a robust behavior whatever the shape and location of the point cloud.

The proposed algorithm is shared not only within the Fiji plugin, but also through a set of Matlab codes delivered on a Github repository `https://github.com/pierre-weiss/FitEllipsoid`. To the best of our knowledge, this is the first open-source toolbox that allows fitting ellipsoids and not more general quadrics (i.e. hyperboloids).

*Structure of the paper.* The rest of the paper is organized as follows. We first describe in Section 2 the plugin to segment 3D ellipsoids. Then we present in Section 3 the algorithmic approach that was adopted to solve the ellipsoid fitting problem. In Section 4 we propose a series of numerical experiments that compare the proposed approach with other algorithms.

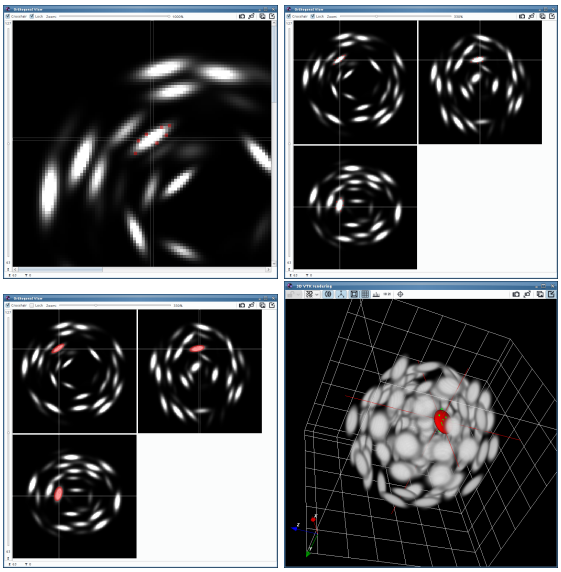

Figure 1: Top: Selecting points (at least 9) in the orthogonal views. Bottom: result of the fitting for 15 points.

## 2. The plugin

A video tutorial describing the main features of the plugin is provided at the following address: `https://youtu.be/MjotgTZi6RQ`. Below, we explain the technical choices and illustrate how the user can input the point cloud and retrieve the segmented ellipsoid.

### *2.1. Specifications*

The main objective of this plugin is to segmentation of ellipsoidal objects, while satisfying the following constraints:

- Permit 3D visualization to allow for visual inspection of the segmentation or illustrate some results.
- Deliver a free and open-source software.
- Minimize the time required to obtain the results. This is particularly important in biology where some applications require to segment hundreds or thousands of cells.
- Export the results as files that other programs can use for further processing.

### *2.2. Description*

The need for a free software dedicated to biomedical imaging, allowing 3D visualization and user interfaction, oriented us to the recently developed imaging tool icy (De Chaumont et al., 2012). It is based on VTK (Visualization ToolKit) (Schroeder and Martin, 1996), allowing for nice 3D rendering using graphics cards. The user interaction can take advantage of 3 orthogonal views that can be intuitively displaced in space.

An ellipsoid can be mathematically represented in different ways:

- A center (3 parameters), three angles of rotations and the length of each axis (3 parameters).
- A center (3 parameters), the three axes (9 parameters linked through orthogonality relationships) and the length of each axis (3 parameters).
- A center (3 parameters) and a positive symmetric definite matrix (6 coefficients).

Unfortunately, none of these representations can be easily used by a human. For instance, finding the center of the ellipsoid precisely by just looking at the image would result in inaccurate estimation.

The interface of the plugin asks the user to select points in 3D on the object's boundary and then create an ellipsoid that passes through them approximately.

In theory, it is possible to reconstruct an ellipsoid perfectly when knowing as little as 9 points in generic position lying on its surface (see Proposition 1). The estimation with just 9 points may be unstable to noise, which cannot be avoided due to imperfect pixel selection by the user. We therefore let the user select as many points on the boundary as desired.

In order to select points on the object boundary, the user is asked to select points on 3 orthogonal 2D views

(see Fig. 1, top). Orthogonal views are probably the easiest way to interact with a 3D environment and is very common in biomedical imaging (see e.g. (Heller et al., 2002)). The user first selects a point in 3D space to define the 3 planes of interest and then locks the views to click on a few points on each plane. The operation may be repeated on multiple orthogonal views to sample the object surface more uniformly. When enough points have been selected, an algorithm described in Section 3 fits an ellipsoid to the points cloud. The operation can be repeated in case multiple ellipsoids have to be fitted. The result obtained by the points selection is displayed in Fig. 1, bottom.

Apart from the 3D visualization, the ellipsoids parameters (center, axes orientations and length of axes) can be exported in a CSV file. In addition, it is possible to save a 3D binary image indicating the interior of each segmented ellipsoid.

### *2.3. Experiments*

The initial motivation for this plugin is the segmentation of nuclei in 3D tumor spheroids, images with fluorescence microscopes. To the best of our knowledge, the only attempt to automatically segment 3D ellipsoids specifically was proposed in Soubies et al. (2013). This algorithm provided too many false positives on the 3D stacks we obtained. The main reason is that Selective Plane Illumination Microscope images suffer from sever degradations in the $z$-direction, related to diffraction and scattering.

On the side of supervised segmentation algorithms, the closest rival we found is the segmentation editor from the commercial software Amira. This algorithm allows delineating a few 2D slices and then renders a surface passing through the curves approximately. The surface can then be approximated with an ellipsoid. The users feedback is that the proposed plugin allows segmenting nuclei about 5 times faster than the segmentation editor. In practice, segmenting a nuclei in a complex, large scale image with FitEllipsoid requires between 30 seconds and a minute.

## 3. Fast and robust ellipsoid fitting

### *3.1. Mathematical description*

Given a set of $n$ points $X = (x_i)_{1\leq i\leq n}$ in $\mathbb{R}^d$, where $d = 2$ or 3, the objective of this section is to describe a fast and robust algorithm to fit an ellipsoid to those points. This is a longstanding problem studied in more than 40 journal papers. We refer to the book (Kanatani et al., 2016) for a more comprehensive overview. Two main approaches have been proposed to solve it.

*The geometric approach.* This method was proposed in (Gander et al., 1994; Ahn et al., 2001; Kleinsteuber and Hüper, 2010). It consists of finding an ellipsoid $E$ that minimizes the following least squares problem:

$$F(E) = \sum_{i=1}^{n} \mathrm{dist}(x_i, E)^2, \tag{1}$$

where $\mathrm{dist}(x_i, E) = \inf_{x\in E} \|x - x_i\|$ is the Euclidean distance from the point $x_i$ to the ellipsoid $E$. While this formulation has a simple geometrical meaning, it suffers from being highly nonconvex. Designing global minimization methods is therefore heavy from a computational point of view.

*The algebraic approach.* This method is the one adopted in this paper. An ellipsoid $E$ can be represented by a triplet $(A, b, c)$ through the implicit equation:

$$E = \left\{x \in \mathbb{R}^d, \langle x, Ax\rangle + \langle b, x\rangle + c = 0\right\}, \tag{2}$$

where $A \in \mathbb{R}^{d\times d}$ is a symmetric positive definite matrix, $b \in \mathbb{R}^d$ is a vector and $c \in \mathbb{R}$ is a scalar. Assuming that the eigenvalue decomposition of $A$ reads $A = U\Sigma U^T$, where $U = [u_1, \ldots, u_d]$ is an orthogonal matrix and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_d)$, equation (2) can be rewritten as

$$E = \left\{x \in \mathbb{R}^d, \|\Sigma^{1/2}U^T(x - z)\|^2 = r^2\right\}, \tag{3}$$

where the ellipse center $z$ and the parameter $r$ are related to $b$ and $c$ via the following equations:

$$b = -2Az \text{ and } c = \langle Az, z\rangle - r^2. \tag{4}$$

The ellipsoid's $i$-th axis is characterized by its length $\frac{r}{\sqrt{\sigma_i}}$, and its direction is given by $u_i$.

The algebraic approach consists of minimizing the following residual:

$$G(X, A, b, c) = \sum_{i=1}^{n} (\langle x_i, Ax_i\rangle + \langle b, x_i\rangle + c)^2, \tag{5}$$

over a set $\mathcal{M}$ of admissible triplets $(A, b, c)$. The sole positive definiteness condition $A \succ 0$ is not sufficient since the infimum of $G$ over the set of positive semi-definite matrices is reached for $(A, b, c) = (0, 0, 0)$. It is therefore required to add a normalization condition to avoid this trivial solution. Various possibilities have been considered in the literature. The most popular approach lately was proposed in (Calafiore, 2002) and consists of imposing $\mathrm{Tr}(A) = 1$. This choice has the advantage of leading to a convex constraint, allowing to

design efficient numerical algorithms. Overall, the optimization problem considered herein reads:

$$\min_{(A,b,c)\in\mathcal{M}} G(X,A,b,c), \tag{6}$$

where $\mathcal{M} = \{(A,b,c) \in \mathbb{R}^{d\times d}\times\mathbb{R}^d\times\mathbb{R}, A \succeq 0, \mathrm{Tr}(A) = 1\}$.

From a numerical point of view, the interest of this specific formulation lies in Proposition 1 below, which states that the solution is unique when enough points are provided.

**Proposition 1.** *Problem* (6) *is convex. It admits at least one minimizer. Define*

$$m = d(d+1)/2 + d + 1. \tag{7}$$

*If $n \leq m-2$, the minimizer is non unique. If the points are in "generic" position and $n \geq m-1$, then the solution is unique.*

Figure 2 illustrates the fact that in 2D there are an infinity of ellipses passing through 4 given points.

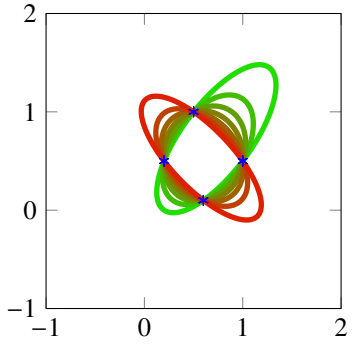


Figure 2: A family of ellipses passing through 4 points. In dimension $d = 2$, we have $m = 6$, hence the minimum number of points required for uniqueness is $n = 5$ according to Proposition 1.

*Proof.* We will use of the equivalent formulation (15) instead of (6).

Problem (15) is convex since $\mathcal{Q}$ and $q \mapsto \|D^T q\|^2$ are both convex. It consists of a projection problem on $\mathcal{Q}$ with the possibly degenerate metric $q \mapsto \|D^T q\|^2$. Standard convex analysis results (Bertsekas et al., 2003, Prop. 2.3.4) state that the solution exists and it is unique when the metric is non degenerate.

If $n \leq m-2$, notice that $\dim(\ker(D^T)) \geq 2$ and $\dim(\{q \in \mathbb{R}^m, \mathrm{Tr}(\mathcal{A}(q)) = 1\}) = m-1$. Hence, there is a subspace $V$ of dimension at least 1 such that for all $q \in V$, $D^T q = 0$ and $\mathrm{Tr}(\mathcal{A}(q)) = 1$. All vectors $q$ in this subspace satisfy $\|D^T q\|_2^2 = 0$, hence they are solution of (15). They all describe ellipsoids passing perfectly through the set of points.

If $n \geq m-1$, the genericity hypothesis in proposition 1 amounts to assume that $\ker(D^T)$ is of dimension less or equal than 1 and is transverse to $\{q \in \mathbb{R}^m, \mathrm{Tr}(\mathcal{A}(q)) = 1\}$. Hence, the function $q \mapsto \frac{1}{2}\|D^T q\|_2^2$ is strictly convex over the hyperplane $\{q \in \mathbb{R}^m, \mathrm{Tr}(\mathcal{A}(q)) = 1\}$ ensuring uniqueness of the minimizer. □

**Remark 1.** *An ellipsoid should satisfy $A \succ 0$ and not $A \succeq 0$. However, it is important to work over closed sets to ensure existence of a minimizer. The minimizer of* (6) *can therefore in principle be the equation of a degenerate ellipsoid such as a line in 2D or a plane in 3D. This situation never happened in our numerical experiments.*

The following proposition states an isometric invariance of the problem.

**Proposition 2.** *The minimizer $(\hat{A},\hat{b},\hat{c})$ is covariant to translation and rotation of the input point locations $X$. More precisely, let $\hat{E}$ denote the ellipsoid solution of* (6) *and $\hat{E}'$ denote the ellipsoid obtained by solving* (6) *with input coordinates $X' = (x'_i)_{1\leq i\leq n}$, where $x'_i = Rx_i + t$, $R \in \mathbb{R}^{d\times d}$ is an orthogonal matrix and $t \in \mathbb{R}^d$ is a translation vector. Then $\hat{E}' = R\hat{E} + t$.*

*Proof.* Let $E$ denote an ellipsoid defined through the triplet $(A,b,c) \in \mathcal{M}$. Now, let $E' = RE + t$. A change of variable shows that $E'$ is defined through $(A',b',c')$ with

$$A' = RAR^T \ , \ b' = Rb - 2A't \text{ and } c' = c - \langle t, A't + b'\rangle. \tag{8}$$

In addition, note that $(A',b',c') \in \mathcal{M}$ since the trace and eigenvalues are invariant to isometries. Straightforward calculus and the relationship $R^TR = RR^T = Id$ show that

$$G(X,A,b,c) = G(X',A',b',c'). \tag{9}$$

Hence the minimizers $(\hat{A},\hat{b},\hat{c})$ of $G(X,A,b,c)$ over $\mathcal{M}$ and the minimizer $(\hat{A}',\hat{b}',\hat{c}')$ of $G(X',A',b',c')$ are related through (8). □

**Remark 2.** *The proof uses the orthogonality identity $R^TR = Id$, and the result is no more true when $R$ is not orthogonal. The solution of* (6) *is not covariant to affine changes.*

### *3.2. A numerical algorithm*

In (Calafiore, 2002), Calafiore suggested reformulating (6) as a semi-definite program and used interior point type methods to solve it. This type of algorithm is known to be robust and reliable but is rather elaborate

to implement. Moreover, common primal-dual interior point methods (Ying et al., 2012) have a complexity that does not scale well with the number of input data points $n$. Based on this observation, Lin and Huang (Lin and Huang, 2016) designed a method based on the alternating direction method of multipliers (ADMM) to solve problem (6). While the per-iteration complexity of this approach is lower than that of interior point methods, the number of iterations is hard to control from a theoretical point of view, and we will show through numerical experiments that it can be very long to reach satisfactory solutions (10,000 to 100,000 iterations). We propose a more robust approach in what follows, and in practice our plugin converges in 200 iterations.

In 2D, the fact that point $x_i$ belongs to the ellipse represented by $(A, b, c)$ reads:

$$a_{1,1}x_i[1]^2+a_{2,2}x_i[2]^2+2a_{1,2}x_i[1]x_i[2]+b_1x_i[1]+b_2x_i[2]+c = 0. \tag{10}$$

By stacking the coefficients in a vector

$$q = (a_{1,1}, a_{2,2}, \sqrt{2}a_{1,2}, b_1, b_2, c)^T,$$

equation (10) can be rewritten in the compact form (see e.g. (Gander et al., 1994))

$$\langle d_i, q\rangle = 0,$$

where

$$d_i = (x_i[1]^2, x_i[2]^2, \sqrt{2}x_i[1]x_i[2], x_i[1], x_i[2], 1)^T. \tag{11}$$

Now, letting $D = [d_1, \dots, d_n]$, function $G$ can be rewritten as

$$G(X, q) = \|D^T q\|^2. \tag{12}$$

In 3D, a similar decomposition can be performed. Following (Gander et al., 1994) we used the following conventions in our codes

$$q = (a_{1,1}, a_{2,2}, a_{3,3}, \sqrt{2}a_{1,2}, \sqrt{2}a_{1,3}, \sqrt{2}a_{2,3}, b_1, b_2, b_3, c)^T,$$

$$\begin{aligned} d_i = (&x_i[1]^2, x_i[2]^2, x_i[3]^2,\\ &\sqrt{2}x_i[1]x_i[2], \sqrt{2}x_i[1]x_i[3], \sqrt{2}x_i[2]x_i[3],\\ &x_i[1], x_i[2], x_i[3], 1)^T. \end{aligned} \tag{13}$$

Let $m = d(d+1)/2 + d + 1$ denote the number of parameters in $q$. The set of admissible vectors $\mathcal{Q}$ is then defined as:

$$\mathcal{Q} = \{q \in \mathbb{R}^m, \mathrm{Tr}(\mathcal{A}(q)) = 1, \mathcal{A}(q) \succeq 0\}, \tag{14}$$

where $\mathcal{A} : \mathbb{R}^m \to \mathbb{R}^{d\times d}$ is the linear mapping that associates matrix $A$ to vector $q$. Similarly, we let $\mathcal{B} : \mathbb{R}^m \to \mathbb{R}^d$ and $C : \mathbb{R}^m \to \mathbb{R}$ denote the mapping that to $q$ associate $b$ and $c$ respectively. With the proposed notation, problem (6) simplifies to the following convex problem:

$$\min_{q\in\mathcal{Q}} \|D^T q\|^2. \tag{15}$$

We solve (15) using the Douglas-Rachford algorithm, see (Lions and Mercier, 1979). It is an algorithm designed to solve problems of the following type:

$$\min_{q\in\mathbb{R}^m} f_1(q) + f_2(q), \tag{16}$$

where $f_1 : \mathbb{R}^m \to \mathbb{R}\cup\{+\infty\}$ and $f_2 : \mathbb{R}^m \to \mathbb{R}\cup\{+\infty\}$ are extended real-valued convex closed functions such that $f_1(x) + f_2(x) \to +\infty$ as $\|x\| \to +\infty$. It is described in Algorithm 1, while Proposition 3 states its convergence properties. We remind that the proximal operator of a function $f$ is defined by:

$$\mathrm{Prox}_{\gamma f}(z) = \underset{x\in\mathbb{R}^n}{\mathrm{argmin}}\ \gamma f(x) + \frac{1}{2}\|x - z\|^2.$$

**Algorithm 1** Douglas-Rachford algorithm to solve (16)

**Input:** Initial guess $q^{(1/2)} \in \mathbb{R}^m$, number of iterations $Nit$, parameter $\gamma > 0$.
**Output:** $q^{(Nit)}$ an approximate solution of (16).
**for** $k = 1$ to $Nit$ **do**

$q^{(k)} := \mathrm{Prox}_{\gamma f_2}(q^{(k-1/2)})$
$q^{(k+1/2)} := q^{(k-1/2)} - q^{(k)} + \mathrm{Prox}_{\gamma f_1}(2q^{(k)} - q^{(k-1/2)})$.

**end for**

**Proposition 3** (Convergence (Lions and Mercier, 1979; Combettes and Pesquet, 2011))**.** *The sequence $(q^{(k)})_{k\in\mathbb{N}}$ generated by Algorithm 1 converges to a solution of* (16)*.*

To apply Algorithm 1 to solve (15), we set

$$f_1(q) = \|D^T q\|^2 \tag{17}$$

and

$$f_2(q) = \begin{cases} 0 & \text{if } q \in \mathcal{Q},\\ +\infty & \text{otherwise.}\end{cases}$$

It remains to evaluate the proximal operators of $f_1$ and $f_2$. They are given in proposition 4 and 5 below.

**Proposition 4.** *The proximal operator of $f_1$ is given by*

$$\mathrm{Prox}_{\gamma f_1}(z) = (\gamma DD^T + \mathrm{Id})^{-1}(z). \tag{18}$$

**Proposition 5.** *Let $z \in \mathbb{R}^m$. Assume that $\mathcal{A}(z)$ is diagonalized as $\mathcal{A}(z) = U\Sigma U^T$, where $\Sigma = \mathrm{diag}(\sigma)$ and $U$ is orthogonal. Let $\sigma_+$ denote the projection of $\sigma$ on the unit simplex and define $A_+ = U\mathrm{diag}(\sigma_+)U^T$. Then $\mathrm{Prox}_{\gamma f_2}(z)$ is obtained by changing the first $d(d+1)/2$ components of $z$ by $\mathcal{A}^{-1}(A_+)$ and leaving the others unchanged.*

*Proof.* It suffices to note that:

$$\begin{aligned}
&\min_{\mathcal{A}(q)\geq 0, \mathrm{Tr}(\mathcal{A}(q))=1} \frac{1}{2}\|q - z\|^2 \\
&= \min_{\mathcal{A}(q)\geq 0, \mathrm{Tr}(\mathcal{A}(q))=1} \frac{1}{2}\|\mathcal{A}(q) - \mathcal{A}(z)\|_F^2 \\
&= \min_{S\geq 0, \mathrm{Tr}(S)=1} \frac{1}{2}\|S - \Sigma\|_F^2.
\end{aligned}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, which is invariant by unitary transforms. Projecting a vector on the unit simplex of $\mathbb{R}^d$ is a standard issue met in matrix completion problems, see e.g. (Condat, 2016). It can be solved exactly in $O(d)$ operations. □

**Remark 3.** *In (Lin and Huang, 2016), the authors proposed using the ADMM algorithm (Glowinski and Marroco, 1975), which can be seen as the Douglas-Rachford algorithm applied to the dual of* (16)*. Their implementation relies on the fact that $\mathcal{Q} = \mathcal{Q}_1 \cap \mathcal{Q}_2$, where $\mathcal{Q}_1$ describes the set of symmetric positive semi-definite matrices and $\mathcal{Q}_2$ describes the set of matrices with trace equal to* 1*. They then propose to split the problem in three terms (one for $\mathcal{Q}_1$, one for $\mathcal{Q}_2$ and one for $f_1$) while our decomposition uses only two terms. This simplifies the algorithm by reducing the number of parameters to tune to* 1*: the value of $\gamma$.*

The final algorithm to solve the ellipsoid fitting problem (6) is given in Algorithm 2. It depends on two extra parameters: the number of iterations *Nit* and a value $\gamma > 0$. The number of iterations to achieve a reasonable result strongly depends on normalizing conditions described in the next section. The convergence is also very sensitive to the value of $\gamma$. However, with the normalization proposed in the next section, it can be tuned once for all. In all our numerical experiments, we use $\gamma = 10$.

### *3.3. Making the algorithm invariant to affine transforms*

#### *3.3.1. Non invariance of Algorithm 2*

As stated in Proposition 2, the minimizers of (6) are invariant to isometries. However, it is easily seen that the algorithm is not, since the basis of polynomials (11) is not adapted to the dataset. As a result, we observed that the algorithm performance strongly depends on the points locations. This is illustrated in Fig. 3 for coordinate shifts and in Fig. 4 for the dilation of one axis. In addition, the solutions of (6) are not invariant to affine transforms, which is a desirable property. We propose to address both issues below. Similar ideas were proposed in (Nievergelt, 2001) for the specific case of spheres.

**Algorithm 2** An algorithm to solve (6)

**Input:** Data points $X = [x_1, \ldots, x_n]$, number of iterations *Nit*, parameter $\gamma > 0$.
**Output:** An ellipsoid $E$.
Compute matrix $D$ using equation (11) in 2D or (13) in 3D.
Set $c_i := \mathrm{mean}(X(:, i))$.
Set $r^2 := \mathrm{mean}(\sum_{i=1}^{d}(X(:, i) - c_i)^2)$.
**if** $d = 2$ **then**
  Set $q^{(1/2)} := [0.5, 0.5, 0, -c_1, -c_2, (c_1^2 + c_2^2 - r^2)/2]$.
**else if** $d = 3$ **then**
  Set $q^{(1/2)} := [0.5, 0.5, 0.5, 0, 0, 0, -c_1, -c_2, -c_3, (c_1^2 + c_2^2 + c_3^2 - r^2)/2]$.
**end if**
Call Algorithm 1.
Transform vector $q^{(Nit)}$ as an ellipsoid $E$ using equation (3).

#### *3.3.2. Ensuring invariance using the SVD*

We propose to ensure invariance of the algorithm by changing the coordinate system in order to obtain a point cloud that is centered with covariance matrix equal to the identity. Then we use Algorithm 2 to fit an ellipsoid in the modified system, and finally map it back to the original coordinates. This can be achieved using a singular value decomposition of a $d \times d$ matrix, as explained below.

Let $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d\times n}$. We are looking for a linear transform $P \in \mathbb{R}^{d\times d}$ and a translation vector $t \in \mathbb{R}^d$, such that the vectors

$$y_i = P(x_i - t) \tag{19}$$

have mean 0 and covariance matrix Id. Letting $Y = [y_1, \ldots, y_n]$, this means that $YY^T = \mathrm{Id}$.

For centering $X$, we use

$$t = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{20}$$

Now, let $\bar{X} = [x_1 - t, \ldots, x_n - t]$ denote the set of centered vectors. The eigenvalue decomposition of $\bar{X}\bar{X}^T$ reads

$$\bar{X}\bar{X}^T = \bar{U}\bar{\Sigma}\bar{U}^T. \tag{21}$$

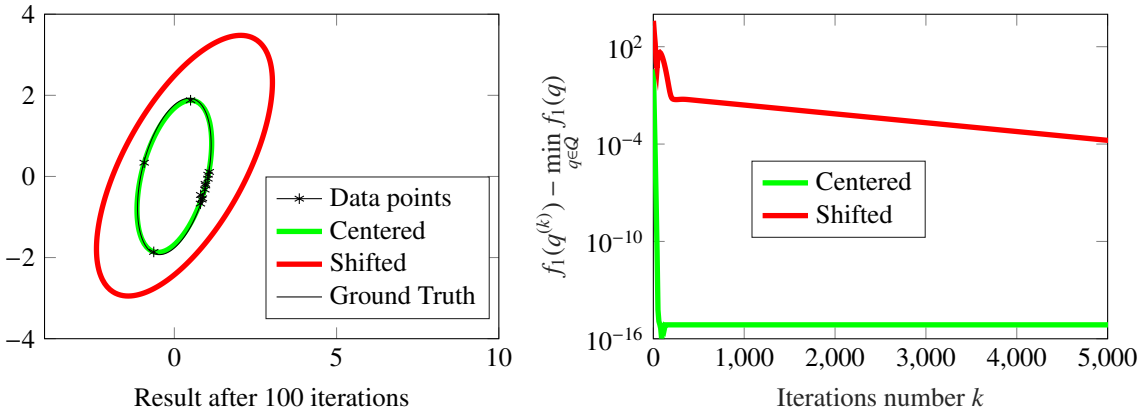


Figure 3: Shift experiment: in this experiment, we fit the set of points indicated with asterisks using 100 iterations of Douglas-Rachford algorithm 2, or the same set shifted by 10 units horizontally. We then plot the result at the same origin. As can be seen, the green ellipsoid (non shifted points) is well retrieved, while the one associated to shifted points is unsatisfactory. The same behavior can be observed for the cost function. The convergence curve associated to the non shifted points shows that machine precision is reached after about 50 iterations. On its side, the convergence curve associated to the shifted points converges linearly with a very slow rate. It reaches the maximal reachable precision ($10^{-10}$) in about 20,000 iterations.

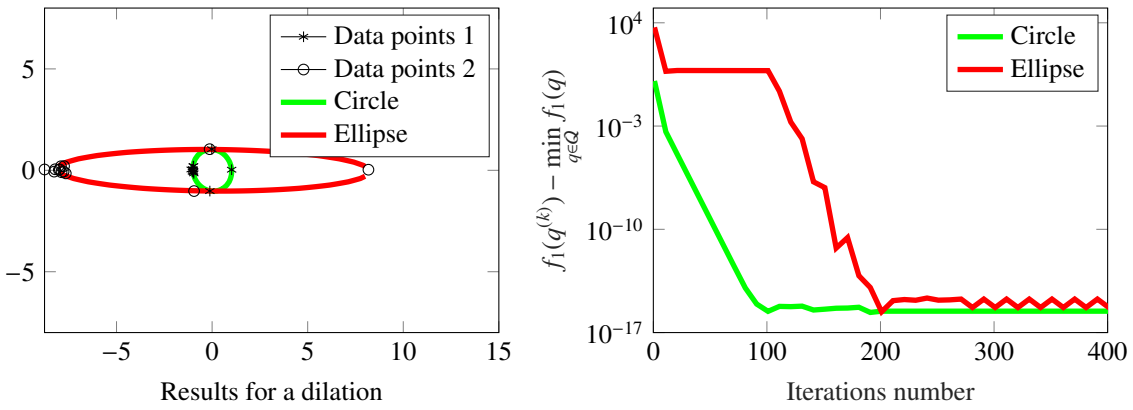


Figure 4: Linear transform experiment. Left: we fit the set of points indicated with asterisks using 400 iterations of Douglas-Rachford algorithm 2, or the same set dilated by a factor 8 along the $x$-axis. Right: the convergence curve shows a different behavior: while the curve associated to the circle shows a favorable linear behavior right from the start, the curve associated to the ellipse has a plateau for the first 110 iterations and then then has the same favorable linear behavior. This phenomenon illustrates recent results about the Douglas-Rachford algorithm behavior (Liang et al., 2016): the algorithm converges to a manifold in a finite number of iterations and then shows a linear convergence rate.

By letting

$$P = \Sigma^{-1/2} U^T, \tag{22}$$

we obtain the desired result.

Algorithm 3 summarizes the proposed idea. It strongly improves the algorithm's convergence. In practice, this approach never required more than 200 iterations to reach machine precision, while the unnormalized method can require arbitrarily large computing times depending on the points location.

**Algorithm 3** Ellipsoid fitting using SVD normalization

**Input:** Data points $X$.
**Output:** An ellipsoid $E$.
Evaluate $t$ and $P$ using (20), (21) and (22).
Construct $y_i = P(x_i - t)$.
Apply Algorithm 2 with input $Y = [y_1, \dots, y_n]$, $Nit = 100$ and $\gamma = 10$.
Set $E = P^{-1}E + t$.

## 4. Experiments

### *4.1. Comparisons in 2D*

We illustrate here the behavior of different algorithms on a 2D example. We compare 4 algorithms both in terms of computing times and robustness of the results with respect to i) noise and ii) non-uniform sampling. In all experiments, the same ellipse is used. Its center is set to $z = (4, 5)$ and the lengths of its axes are 4 and 1.

1. The first algorithm denoted DR is Algorithm 2.
2. The second algorithm denoted DR-SVD is Algorithm 3.
3. The third algorithm denoted LLS is the standard Linear Least Square approach (Fitzgibbon et al., 1999). It consists in replacing the normalization constraint $\mathrm{Tr}(\mathcal{A}(q)) = 1$ in equation (14) by the constraint choosing $\|q\| = 1$, and to skip the constraint $\mathcal{A}(q) \succeq 0$. The problem becomes

$$\min_{\|q\|=1} \|D^T q\|^2, \tag{23}$$

   which amounts to finding the smallest eigenvector of $K = DD^T$. This vector $\tilde{q}$ describes a conic, which is not necessarily an ellipse.
4. The fourth algorithm denoted LLS-SVD consists of applying LLS to the dataset after a change of coordinates, similarly to what is described in Section 3.3.2. This algorithm is affine invariant.

The results are presented in Fig. 5, where the solutions of the different algorithms are shown for two different points cloud. The results are presented after 4000 iterations, which is necessary for DR to converge, while DR-SVD only requires 30.

### *4.2. 3D experiments*

In this section, we perform a few experiments to challenge the algorithm implemented in the Icy plugin. To do so, we draw points uniformly at random on the boundary of an ellipsoid on 3 orthogonal planes. We then add a random perturbation of normal distribution with variance $\sigma^2$ within each plane. This way, we simulate what a user does by clicking on points in 3 orthogonal views. Note that the orthogonal planes orientation do not necessarily coincide with the ellipsoid axes.

The objective is to illustrate the minimal number of points required and the stability to noise. As can be seen in this example, 20 points are enough to provide an accurate result despite a significant amount of noise.

## Conclusions

We proposed a supervised segmentation algorithm of ellipsoids within the open-source Icy imaging software. It provides results significantly faster than usual 2D delineation techniques, reducing human labor or allowing to segment much larger sets in a given amount of time. The results may be used to create gold standards or learning databse, or also as such for biomedical interpretations.

The algorithm is based on a novel fast and lightweight method to fit ellipsoids to point clouds in an affine invariant manner. It provides satisfactory results in a short time, whatever the points configuration. We show a much better behavior than recently developed approaches.

All the codes are open-source.

## Acknowledgments

The authors wish to thank Bernard Ducommun, Ludivine Guillaume and Valérie Lobjois for testing, giving their feedback advices to improve the plugin for collections of real fluorescence microscopy images. They thank Guillaume de Brito for developing the 2D plugin DrawEllipse and the StructureTensor plugins, which were the starting point of this work.

This work was supported by the MIMMOSA project, funded by Plan Cancer. It was also supported by the OPTIMUS project, funded by fondation Innabiosanté.

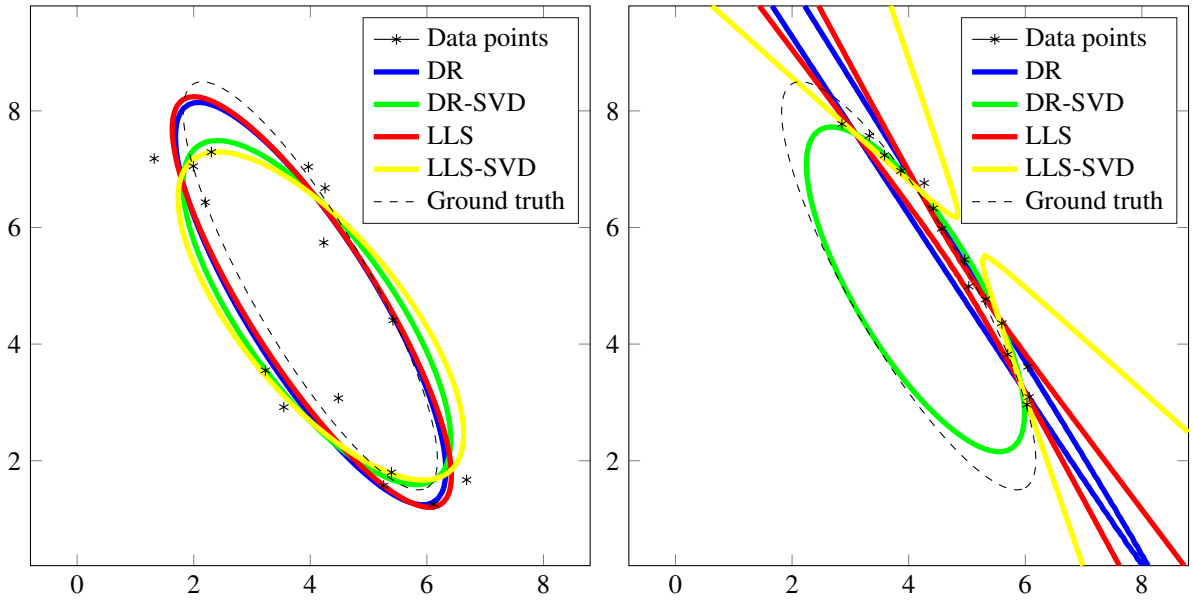


Figure 5: Comparisons of 4 different algorithms to fit an ellipse to a set of points. Left: all algorithms perform well for uniformly distributed points along an ellipse. The number of iterations necessary for DR to converge is 4000 and it is 30 for DR-SVD. Right: results for nonuniformly distributed points. The LLS and LLS-SVD algorithm produce hyperbolas instead of ellipses. In this specific case, DR-SVD produces a result closer to the ground truth than DR, but it is not the case for all noise realizations. Notice that the users of our plugins should avoid specifying configurations of points as shown in the right figure.


Ahn, S. J., Rauh, W., Cho, H. S., Warnecke, H.-J., 2002. Orthogonal distance fitting of implicit curves and surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5), 620–638.

Ahn, S. J., Rauh, W., Warnecke, H.-J., 2001. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. Pattern Recognition 34 (12), 2283–2303.

Bertsekas, D. P., Nedi, A., Ozdaglar, A. E., et al., 2003. Convex analysis and optimization. Athena Scientific.

Calafiore, G., 2002. Approximation of n-dimensional data using spherical and ellipsoidal primitives. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 32 (2), 269–278.

Chernov, N., Lesort, C., 2005. Least squares fitting of circles. Journal of Mathematical Imaging and Vision 23 (3), 239–252.

Combettes, P. L., Pesquet, J.-C., 2011. Proximal splitting methods in signal processing. In: Fixed-point algorithms for inverse problems in science and engineering. Springer, pp. 185–212.

Condat, L., 2016. Fast projection onto the simplex and the\ pmb {l} _\ mathbf {1} ball. Mathematical Programming 158 (1-2), 575–585.

Cuingnet, R., Prevost, R., Lesage, D., Cohen, L. D., Mory, B., Ardon, R., 2012. Automatic detection and segmentation of kidneys in 3d ct images using random forests. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 66–74.

De Chaumont, F., Dallongeville, S., Chenouard, N., Hervé, N., Pop, S., Provoost, T., Meas-Yedid, V., Pankajakshan, P., Lecomte, T., Le Montagner, Y., et al., 2012. Icy: an open bioimage informatics platform for extended reproducible research. Nature methods 9 (7), 690–696.

Delgado-Gonzalo, R., Chenouard, N., Unser, M., 2013. Spline-based deforming ellipsoids for interactive 3d bioimage segmentation. IEEE Transactions on Image Processing 22 (10), 3926–3940.

Dufour, A., Shinin, V., Tajbakhsh, S., Guillén-Aghion, N., Olivo-Marin, J.-C., Zimmer, C., 2005. Segmenting and tracking fluorescent cells in dynamic 3-d microscopy with coupled active surfaces. IEEE Transactions on Image Processing 14 (9), 1396–1410.

Fitzgibbon, A., Pilu, M., Fisher, R. B., 1999. Direct least square fitting of ellipses. IEEE Transactions on pattern analysis and machine intelligence 21 (5), 476–480.

Gander, W., Golub, G. H., Strebel, R., 1994. Least-squares fitting of circles and ellipses. BIT Numerical Mathematics 34 (4), 558–578.

Glowinski, R., Marroco, A., 1975. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique 9 (2), 41–76.

Heller, G. V., Cerqueira, M. D., Weissman, N. J., Dilsizian, V., Jacobs, A. K., Kaul, S., Laskey, W. K., Pennell, D. J., Rumberger, J. A., Ryan, T., et al., 2002. Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart: a statement for healthcare professionals from the cardiac imaging committee of the council on clinical cardiology of the american heart association. Journal of Nuclear Cardiology 9 (2), 240–245.

Jaqaman, K., Loerke, D., Mettlen, M., Kuwata, H., Grinstein, S., Schmid, S. L., Danuser, G., 2008. Robust single-particle tracking in live-cell time-lapse sequences. Nature methods 5 (8), 695–702.

Kanatani, K., Sugaya, Y., Kanazawa, Y., 2016. Ellipse fitting for computer vision: implementation and applications. Synthesis Lectures on Computer Vision 6 (1), 1–141.

Kleinsteuber, M., Hüper, K., 2010. Approximate geometric ellipsoid fitting: A cg-approach. In: Recent Advances in Optimization and its Applications in Engineering. Springer, pp. 73–82.

Li, Q., Griffiths, J. G., 2004. Least squares ellipsoid specific fitting. In: Geometric modeling and processing, 2004. proceedings. IEEE, pp. 335–340.

Liang, J., Fadili, J., Peyré, G., 2016. Convergence rates with inex-

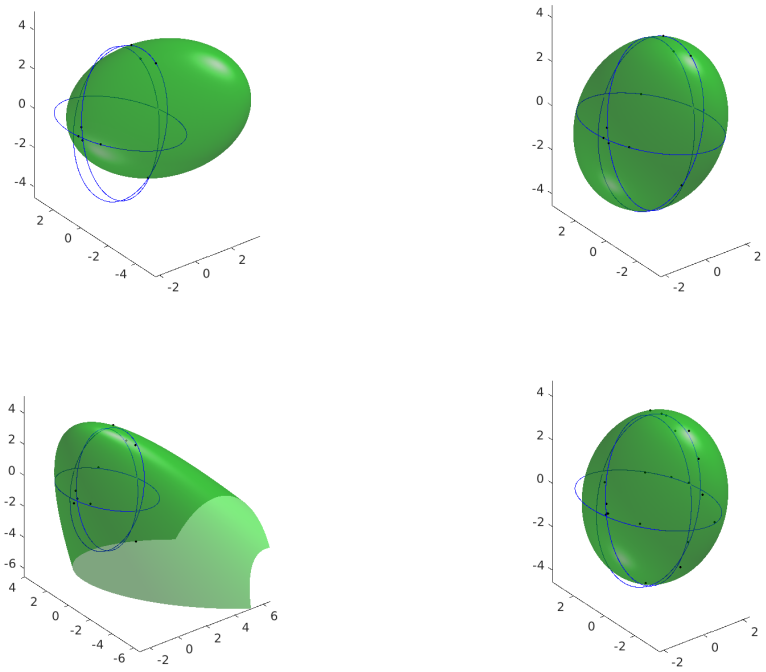

Figure 6: Fitting a 3D ellipsoid with semi-axes lengths $l_1 = 5$, $l_2 = 3$ and $l_3 = 1$. All results in this experiment were obtained with the proposed DR-SVD algorithm. The retrieved ellipsoid is displayed in green. The planar sections of the true underlying ellipsoid are displayed in blue. The sampled points are displayed in black. In each experiment, numerical precision was obtained in less than 200 iterations. Top-left ($n = 8$ points, $\sigma = 0$) : this is not sufficient to retrieve the ellipsoid. Top-right ($n = 9$, $\sigma = 0$): this is sufficient to perfectly recover the exact ellipsoid. These two results illustrate Proposition 1. Bottom-left ($n = 9$, $\sigma = 0.2$): the detection is inaccurate since $n$ is too small. Bottom-right ($n = 20$, $\sigma = 0.2$): increasing the number of points $n$ renders satisfactory results even in noisy settings.

act non-expansive operators. Mathematical Programming 159 (1-2), 403–434.

Lin, Z., Huang, Y., 2016. Fast multidimensional ellipsoid-specific fitting by alternating direction method of multipliers. IEEE transactions on pattern analysis and machine intelligence 38 (5), 1021–1026.

Lions, P.-L., Mercier, B., 1979. Splitting algorithms for the sum of two nonlinear operators. SIAM Journal on Numerical Analysis 16 (6), 964–979.

Lockett, S. J., Sudar, D., Thompson, C. T., Pinkel, D., Gray, J. W., 1998. Efficient, interactive, and three-dimensional segmentation of cell nuclei in thick tissue sections. Cytometry 31 (4), 275–286.

Mahdavi, S., Salcudean, S. E., 2008. 3d prostate segmentation based on ellipsoid fitting, image tapering and warping. In: Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE. IEEE, pp. 2988–2991.

Markovsky, I., Kukush, A., Van Huffel, S., 2004. Consistent least squares fitting of ellipsoids. Numerische Mathematik 98 (1), 177–194.

Nievergelt, Y., 2001. Hyperspheres and hyperplanes fitted seamlessly by algebraic constrained total least-squares. Linear Algebra and its Applications 331 (1), 43 – 59.

Okada, K., Comaniciu, D., Krishnan, A., 2005. Robust anisotropic gaussian fitting for volumetric characterization of pulmonary nodules in multislice ct. IEEE Transactions on Medical Imaging 24 (3), 409–423.

Olivo-Marin, J.-C., 2002. Extraction of spots in biological images using multiscale products. Pattern recognition 35 (9), 1989–1996.

Saunderson, J., Chandrasekaran, V., Parrilo, P. A., Willsky, A. S., 2012. Diagonal and low-rank matrix decompositions, correlation matrices, and ellipsoid fitting. SIAM Journal on Matrix Analysis and Applications 33 (4), 1395–1416.

Schroeder, W. J., Martin, K. M., 1996. The Visualization Toolkit-30. Elsevier Inc.

Soubies, E., Weiss, P., Descombes, X., 2013. A 3d segmentation algorithm for ellipsoidal shapes. application to nuclei extraction. In: ICPRAM-International Conference on Pattern Recognition Applications and Methods. pp. p–97.

Thevenaz, P., Delgado-Gonzalo, R., Unser, M., 2011. The ovuscule. IEEE transactions on pattern analysis and machine intelligence 33 (2), 382–393.

Ying, X., Yang, L., Zha, H., 2012. A fast algorithm for multidimensional ellipsoid-specific fitting by minimizing a new defined vector norm of residuals using semidefinite programming. IEEE transactions on pattern analysis and machine intelligence 34 (9), 1856–1863.

Zhang, W., Fehrenbach, J., Desmaison, A., Lobjois, V., Ducommun, B., Weiss, P., 2016. Structure tensor based analysis of cells and nuclei organization in tissues. IEEE transactions on medical imaging 35 (1), 294–306.