Ellipsoid and Super-Ellipsoid Fitting

Yuchen Lu* and Xu Zhuo*
School of Computer Science and Engineering
Southeast University
Nanjing, China
{yclu, zhuoxu}@seu.edu.cn

***Abstract*—Ellipsoid fitting is essential in geometric modeling. Unlike naive unconstrained methods solely depend on least squares, this study investigates Grammalidis et al.'s constrained method for ellipsoid fitting and CMA-ES optimizer for super-ellipsoid fitting, so as to enhance the understanding of related techniques for relevant researchers.**

***Index Terms*—Ellipsoid, Super-Ellipsoid, Fitting**

I. Introduction

Ellipsoid fitting is a fundamental problem in geometric modeling. Many applications, e.g., detection, segmentation and 3D point cloud reconstruction, require the technique. Naive methods include solving the least squares problem without forcing the quadratic to be an ellipsoid, which is not reliable in principle.

In this report, we investigate the principles of ellipsoid fitting from the very beginning. We introduce and implement the constraint method proposed by Grammalidis et al. [1], which uses a similar condition in ellipse fitting along with a post-check condition. We've also adopted the CMA-ES [2] optimizer for super-ellipsoid fitting that further improves the fitting performance on point cloud data.

Through this study, we hope to help those who are new to the basic knowledge of ellipsoid fitting techniques, and provide potential insights to benefit this domain. The source code will be released to https://github.com/z0gSh1u/ellipsoid-fit soon after the report is submitted.

*: Authors who contribute equally to this work and are ordered in Pinyin order.

II. Methods

A. Naive Solution for Ellipsoid Fitting

The standard form of ellipsoid equation under the Cartesian coordinate is

$$\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}+\frac{(z-z_0)^2}{c^2}=1 \tag{1}$$

where $\mathbf{c}=[x_0,y_0,z_0]^\top$ is its center and $a>0,b>0,c>0$ are lengths of semi-axes (radii). However, this form is not general; It actually assumes the ellipsoid is axis-aligned, i.e., three main axes are parallel to x,y,z axes.

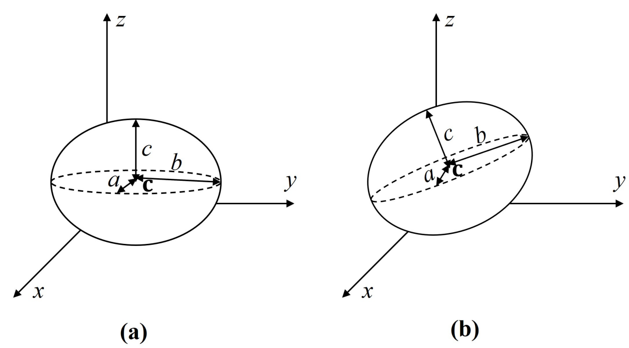

Fig. 1. (a) Axes-aligned ellipsoid and (b) General ellipsoid.

To describe an ellipsoid that tilts and is not necessarily centered at the origin, we need to write the equation into the general quadratic form

$$\begin{aligned} f(\mathbf{a},\mathbf{x}) &= \hat{\mathbf{x}}^\top\hat{\mathbf{A}}\hat{\mathbf{x}} \\ &= \mathbf{x}^\top\mathbf{A}\mathbf{x}+2\mathbf{x}^\top\mathbf{b}+J \\ &= Ax^2+By^2+Cz^2 \\ &+ \ Dxy+Exz+Fyz+Gx+Hy+Iz+J \\ &= \ 0 \end{aligned} \tag{2}$$

where $\mathbf{a} = [A, B, C, D, E, F, G, H, I, J]^\top$ is the parameter vector with 10 parameters, $\mathbf{x} = [x, y, z]^\top$ is the coordinate of the points on the ellipsoid and $\hat{\mathbf{x}} = [x, y, z, 1]^\top$ is the homogeneous coordinate. $\mathbf{A}$ is the quadratic matrix of 2nd order terms

$$\mathbf{A} = \begin{bmatrix} A & D/2 & E/2 \\ D/2 & B & F/2 \\ E/2 & F/2 & C \end{bmatrix} \tag{3}$$

while $\mathbf{b} = [G/2, H/2, I/2]^\top$ is the linear terms. The full quadratic coefficient matrix is then

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^\top & J \end{bmatrix} \tag{4}$$

in the block matrix form.

The Degree-of-Freedom (DoF) for an ellipsoid is 9: three for the center, three for the radii, and three for the inclination (Eular angle). So J is redundant and we usually normalize it to ± 1.

A straightforward naive way to fit the parameters with $n \geq 9$ data points $\mathbf{x}_i = [x_i, y_i, z_i]^\top$ by minimizing the mean squared error of the "algebraic distance" $\sum_i \|f(\mathbf{a}, \mathbf{x}_i)\|^2$ is to solve a least squared problem. From the data matrix

$$\mathbf{D}' = \begin{bmatrix} {x_1}^2 & {y_1}^2 & {z_1}^2 & x_1 y_1 & x_1 z_1 & y_1 z_1 & x_1 & y_1 & z_1 \\ & & & & \vdots & & & & \\ {x_n}^2 & {y_n}^2 & {z_n}^2 & x_n y_n & x_n z_n & y_n z_n & x_n & y_n & z_n \end{bmatrix}, \tag{5}$$

with the right-hand side vector $\mathbf{b}' = [1, \cdots, 1]^\top_{1\times n}$ and the reduced parameter vector $\mathbf{a}' = [A, B, C, D, E, F, G, H, I]^\top$ (assume $J = -1$), the solution in the sense of least squares is

$$\mathbf{a}' = \mathbf{D}'^{+} \mathbf{b}' \tag{6}$$

where $\mathbf{D}'^{+} = (\mathbf{D}'^\top \mathbf{D}')^{-1} \mathbf{D}'^\top$ is the pseudo inverse.

However, Equation (2) doesn't incorporate any condition to force itself to be an ellipsoid; it can be any quadratic, e.g., sheets, cylinders, etc.. Although with "good enough" data points (large amounts distributed all around the entire ellipsoid), the fitting result will tend to be an ellipsoid, it's anyhow not guaranteed.

B. Constrained Ellipsoid Fitting

To conquer the pitfall above, many constrained methods are proposed. Here we adopt Nikos Grammalidis and Michael G.Strintzis's method [1] as follows.

For a 2D conic ($C = E = F = I = 0$), the condition $4AB - D^2 > 0$ is sufficient to make the conic an ellipse. Similarly, migrating to 3D, they proposed the first condition, $4AB - D^2 > 0$. And since J is redundant, we can arbitrarily scale the parameters and pose $4AB - D^2 = 1$ instead. Nevertheless, this condition is necessary but not sufficient (obviously because we constraint only parameters of x and y without z when they are symmetrical in turn). Hence, they supplement a post-check condition $(A + B)|\mathbf{A}| > 0$. One can refer to [1] for detailed proof.

With the first equality condition, the constrained problem is to solve

$$\arg\min_{\mathbf{a}} \sum_i \|f(\mathbf{a}, \mathbf{x}_i)\|^2 \quad \text{s.t. } \mathbf{a}^\top \mathbf{C} \mathbf{a} = 1 \tag{7}$$

where $\mathbf{C}$ is the coefficient matrix of the constraint

$$\mathbf{C} = \begin{bmatrix} 0 & 2 & 0 & 0 & \cdots \\ 2 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & -1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}_{10\times 10} \tag{8}$$

(zeros at the ellipses entries). This problem can be solved using the Lagrange multiplier method. Introduce Lagrange multiplier λ, we get error function

$$\begin{aligned} \mathcal{E} &= \sum_i \|f(\mathbf{a}, \mathbf{x}_i)\|^2 + \lambda(1 - \mathbf{a}^\top \mathbf{C} \mathbf{a}) \\ &= \|\mathbf{D}\mathbf{a}\|^2 + \lambda(1 - \mathbf{a}^\top \mathbf{C} \mathbf{a}) \end{aligned} \tag{9}$$

where the data matrix

$$\mathbf{D} = \begin{bmatrix} {x_1}^2 & {y_1}^2 & {z_1}^2 & x_1 y_1 & x_1 z_1 & y_1 z_1 & x_1 & y_1 & z_1 & 1 \\ & & & & \vdots & & & & & \\ {x_n}^2 & {y_n}^2 & {z_n}^2 & x_n y_n & x_n z_n & y_n z_n & x_n & y_n & z_n & 1 \end{bmatrix}, \tag{10}$$

Differentiate $\mathcal{E}$ with respect to $\mathbf{a}$, we have

$$\frac{\partial \mathcal{E}}{\partial \mathbf{a}} = 2\mathbf{D}^\top \mathbf{D} \mathbf{a} - 2\lambda \mathbf{C} \mathbf{a} \tag{11}$$

Let it be zero, we get a closed-form solution

$$\mathbf{D}^\top \mathbf{D} \mathbf{a} = \lambda \mathbf{C} \mathbf{a} \tag{12}$$

This is similar to the common eigen decomposition but with matrix $\mathbf{C}$ on the right, which is called the *generalized* eigen decomposition problem. The result in the sense of least squares is from the above generalized eigendecomposition problem of $\mathbf{D}^\top\mathbf{D}$, which can be solved by MATLAB's *eig* or Scipy's *linalg.eig* function. Note that $\mathbf{C}$ is singular and has a rank of 3, so only 3 general eigenvalue-eigenvector pairs are valid. [1] has proved that exactly one eigenvalue is positive, and the corresponding eigenvector is the wanted answer for the parameters vector $\mathbf{a}$.

C. Retrieve the Center, Radii, and Tilt

The fitting result $\mathbf{a} = [A \sim J]^\top$ is suitable for formulating the problem in the matrix form, however, unfriendly for humans to know the figure of the ellipsoid. Here we provide the procedure to retrieve the center, radii, and tilt of the ellipsoid from the parameters without completing the square directly which might be challenging.

1) Center: The center $\mathbf{c}$ of the ellipsoid is given by

$$\mathbf{c} = -\mathbf{A}^{-1}\mathbf{b} \tag{13}$$

where $\mathbf{b} = [G/2, H/2, I/2]^\top$, the 1st order terms, describes the translation off the origin. The proof goes as follows.

Proof. The ellipsoid presented under the inhomogeneous coordinate is

$$\mathbf{x}^\top\mathbf{A}\mathbf{x} + 2\mathbf{x}^\top\mathbf{b} + J = 0 \tag{14}$$

as shown in Equation (2). When $\mathbf{x}^\top\mathbf{b} = 0$, the ellipsoid is centered at the origin, because putting $\mathbf{x}$ and the mirrored one $-\mathbf{x}$ in the equation produces the same result. Then, by translating every point with vector $\mathbf{c}$ (including the center), we have

$$\begin{aligned} & (\mathbf{x}-\mathbf{c})^\top\mathbf{A}(\mathbf{x}-\mathbf{c}) + J \\ = \; & \mathbf{x}^\top\mathbf{A}\mathbf{x} - 2\mathbf{x}^\top\mathbf{A}\mathbf{c} + \mathbf{c}^\top\mathbf{A}\mathbf{c} + J \\ = \; & 0 \end{aligned} \tag{15}$$

By comparing the coefficient of 1st order term with respect to $\mathbf{x}$ in Equation (14) and Equation (15), we obtain $-\mathbf{A}\mathbf{c} = \mathbf{b}$, which is the same as Equation (13).

2) Radii: With $\mathbf{c}$ known, we can translate the quadratic to be origin-centered

$$\mathring{\hat{\mathbf{A}}} = \mathbf{T}^\top\hat{\mathbf{A}}\mathbf{T} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & J' \end{bmatrix} \tag{16}$$

with the homogeneous translation matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix} \tag{17}$$

where $\mathbf{I}_n$ is $n \times n$ identity matrix. The "$\circ$" notation stands for "centered at the origin". $\mathring{\hat{\mathbf{A}}}$ is a matrix where entries for the 1st order terms are all zero and the last entry becomes a constant J'. Then we normalize J' to 1 by an assignment to cancel the rescaling

$$\mathring{\hat{\mathbf{A}}} \leftarrow \frac{1}{J'}\mathring{\hat{\mathbf{A}}} \tag{18}$$

and it become

$$\mathring{\hat{\mathbf{A}}} = \begin{bmatrix} \mathbf{A}' & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \tag{19}$$

which now represents the ellipsoid centered at the origin. Its radii (semi-axes lengths) are given by

$$\{a, b, c\} = \{\frac{1}{\sqrt{|\lambda_i|}}, \; i = 1, 2, 3\} \tag{20}$$

where λ_i are the 3 eigenvalues of $\mathbf{A}'$. Without loss of generality, we assume $\lambda_1, \lambda_2, \lambda_3$ corresponds with a, b, c and x, y, z, respectively.

This can be easily understood by knowing the geometrical meaning of the eigendecomposition. Imagine we treat the origin-centered ellipsoid as the result of applying an Affine Transform defined by $\mathbf{A}'$ to the unit ball ($x^2 + y^2 + z^2 = 1$) that "scales" and "rotates". The direction which the ball "grows" is the tilt and can be described by the eigenvectors, which will be introduced in the next subsection, while the "extent" the ball grows along those vectors is related to the eigenvalues. This theorem is given in [3], and here we provide the proof of it as follows.

Proof. Matrix $\mathbf{A}'$ is real and symmetric, then it's orthogonally diagonalizable, which means there exists an orthogonal (rotation) matrix $\mathbf{Q}$ ($\mathbf{Q}^\top = \mathbf{Q}^{-1}$), satisfying

$$\mathbf{Q}^\top\mathbf{A}'\mathbf{Q} = \mathrm{diag}\{\lambda_i\} \tag{21}$$

and the columns of $\mathbf{Q}$ are unit eigenvectors of $\mathbf{A}'$. The diagonalization process converts a quadratic from the general form to the standard form, i.e., $\mathbf{x} = \mathbf{Q}\mathbf{y}$

where $\mathbf{y} = [x', y', z']^\top$ is the coordinate under the standard form. This can be explained by rotating the tilted ellipsoid to axes-aligned so that the radii can be easier to retrieve. The quadratic now becomes

$$\begin{aligned}\mathbf{x}^\top \mathbf{A}'\mathbf{x} &= \mathbf{y}^\top \mathbf{Q}^\top \mathbf{A}'\mathbf{Q}\mathbf{y} \\ &= \mathbf{y}^\top \mathrm{diag}\{\lambda_i\}\mathbf{y} \\ &= \lambda_1 x'^2 + \lambda_2 y'^2 + \lambda_3 z'^2 \\ &= 1\end{aligned} \tag{22}$$

which develops into

$$\frac{x'^2}{\left(\frac{1}{\sqrt{\lambda_1}}\right)^2} + \frac{y'^2}{\left(\frac{1}{\sqrt{\lambda_2}}\right)^2} + \frac{z'^2}{\left(\frac{1}{\sqrt{\lambda_3}}\right)^2} = 1 \tag{23}$$

It follows the standard form as Equation (1), saying the radii correspond with Equation (20).

D. Tilt

The tilt (inclination) of the ellipsoid can be described by the eigenvectors of $\mathbf{A}$ or $\mathbf{A}'$ since the translation won't change the direction of eigenvectors.

Proof. Continued from the proof of radii's formula. Denote $\mathbf{Q} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$ where unit eigenvector $\mathbf{v}_i$ corresponds with λ_i, $i = 1, 2, 3$. Because $\mathbf{y} = \mathbf{Q}^\top\mathbf{x}$, we know $x' = \mathbf{v}_1^\top\mathbf{x}$ and $y' = \mathbf{v}_2^\top\mathbf{x}$ and $z' = \mathbf{v}_3^\top\mathbf{x}$. And they are orthogonal with each other, i.e., $\mathbf{v}_i^\top\mathbf{v}_j = 0, i \neq j$.

Take x-axis for example. From Equation (23), we know when $y' = z' = 0$, the equation becomes a parabola $x'^2 = \mathrm{const}$ that is perpendicular to the yz plane, and its projection is parallel with the radius corresponding with a. This is saying that by letting $\mathbf{v}_2^\top\mathbf{x} = \mathbf{v}_3^\top\mathbf{x} = 0$ we get the line that is collinear with one of the axes of the ellipsoid. According to the mutually orthogonal property of $\mathbf{v}_i$, we have $\mathbf{x} = k_1\mathbf{v}_1$ where $k_1 \neq 0$ is a constant, which leads to the fact that the axis is parallel with the eigenvector $\mathbf{v}_1$. And all similarly for y-axis and z-axis.

The "closer" $\mathbf{v}_1$ is to $[1, 0, 0]^\top$ (In other words, the cosine similarity $|\cos\theta_1| = |\mathbf{v}_1 \cdot [1, 0, 0]^\top|$ closer to 1 where "$\cdot$" is the inner product of vectors and θ_1 is the angle formed by x-axis and $\mathbf{v}_1$), the more "aligned" the ellipsoid is in x direction. And so on for y and z. In fact, $\mathbf{Q}$ is the rotation matrix that rotates the ellipsoid from axes-aligned to the actual one.

E. Super-Ellipsoid Fitting

The standard form equation for the origin-centered axes-aligned super-ellipsoid [4] is

$$\left(\left|\frac{x}{a}\right|^{2/\epsilon_2} + \left|\frac{y}{b}\right|^{2/\epsilon_2}\right)^{\epsilon_2/\epsilon_1} + \left|\frac{z}{c}\right|^{2/\epsilon_1} = 1 \tag{24}$$

where ϵ_1 and ϵ_2 describe its roundness (squareness) on different planes. It degenerates to the ellipsoid if $\epsilon_1 = \epsilon_2 = 1$. Since the coordinate system can be rotated and translated, we won't consider the general form here which is very complicated.

There is no simple analytical form of super-ellipsoid fitting in the sense of least squares, so we should use an automatic black-box optimizer to determine the parameters. Here, we inherit $\{a, b, c\}$ from the result of ellipsoid fitting and treat only ϵ_1 and ϵ_2 as free parameters, so that the search process is more robust.

The error function (metric) we choose to minimize is the root mean squared value (RMSE) of the distances from data points to the super-ellipsoid surface. The definition of the distance is given in the next subsection. The optimizer we use is the covariance matrix adaptation-evolution strategy (CMA-ES)[1], which is a stochastic and derivative-free method for non-linear continuous numerical optimization problems [2]. The implementation is provided by *pycma* library [5] that comes with an *fmin* function to minimize the error function. One can also use other methods like MATLAB's built-in *fminsearch* and there won't be a significant difference.

F. Distance from a Point to the Super-Ellipsoid Surface

The distance from a point to the super-ellipsoid surface is easier to compute under the parametric form. The points $\mathbf{x} = [x, y, z]^\top$ on the super-ellipsoid has a parametric representation [4], [6]

$$\begin{aligned}\mathbf{x} &= \mathbf{x}(\phi, \theta) \\ &= \begin{bmatrix} a\sin^{\epsilon_1}\theta\cos^{\epsilon_2}\phi \\ b\sin^{\epsilon_1}\theta\sin^{\epsilon_2}\phi \\ c\cos^{\epsilon_1}\theta \end{bmatrix}\end{aligned} \tag{25}$$

where parameters $0 \leq \phi < 2\pi$ is the azimuth angle and $0 \leq \theta \leq \pi$ is the polar angle.[2]

[1]Because we used it recently in one of our researches.

[2]This is slightly different in trigonometric functions from the handout due to the definition differences in pole angle and equator angle [6].

The distance is defined as the modulus from the data point $\mathbf{p} = [x_\mathbf{p}, y_\mathbf{p}, z_\mathbf{p}]^\top$ to the intersection point $\mathbf{x}$ of the connecting line from the center $\mathbf{c}$ to $\mathbf{p}$ and the super-ellipsoid as shown by d in Figure 2(a), i.e,

$$\begin{aligned} d &= |\mathbf{p} - \mathbf{x}| \\ &= \sqrt{(x - x_\mathbf{p})^2 + (y - y_\mathbf{p})^2 + (z - z_\mathbf{p})^2} \\ &\geq 0 \end{aligned} \tag{26}$$

where the first subtraction is in the vector sense. To get the coordinate of $\mathbf{x}$, since $\mathbf{c}$, $\mathbf{x}$ and $\mathbf{p}$ are collinear, we can use the parametric form whose parameters are determined by line $\overrightarrow{\mathbf{cp}}$. To properly simplify the question, we assume the super-ellipsoid has been translated to be origin-centered and rotated to be axes-aligned as before. The parameters are now

$$\begin{aligned} \theta &= \arccos \frac{z_\mathbf{p}}{|\mathbf{p}|} \\ \phi &= \arctan \frac{y_\mathbf{p}}{x_\mathbf{p}} \end{aligned} \tag{27}$$

as demonstrated in Figure 2(b). We can now substitute them into Equation (25) and compute the distance using Equation (26).

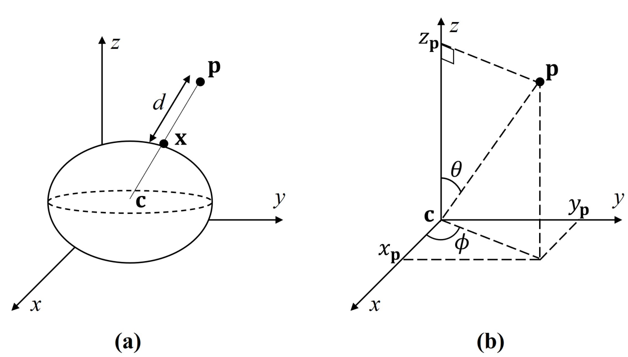

Fig. 2. (a) Definition of the distance. (b) The two parametric angles of the super-ellipsoid.

Note that for super-ellipsoid with some ϵ_1 and ϵ_2, we might suffer from numerical issues in computing the power term, e.g., for $\cos\phi = -1$ and $\epsilon_2 = 1.5$, we get $\cos^{\epsilon_2}\phi = -\mathrm{j}$, a complex number, resulting in unwanted Not-a-Number (NaN) that makes the optimization and distance calculation unable to continue. To solve this, in practice, for the origin-centered axes-aligned super-ellipsoid, we can mirror all points to Octant I $(+x, +y, +z)$ by taking the absolute value of $x_\mathbf{p}$, $y_\mathbf{p}$ and $z_\mathbf{p}$ before computing the angles, so that they are all positive. Since the super-ellipsoid is symmetric with respect to the origin, the distance won't change after the mirroring.

All the above procedures can be applied to the ellipsoid by letting $\epsilon_1 = \epsilon_2 = 1$. If the super-ellipsoid is not origin-centered, we should first compute the angle parameters on the center-translated version of the point, translate the intersection point back, and compute the distance with the original point.

G. Plot the Ellipsoid and Super-Ellipsoid

The origin-centered axes-aligned ellipsoid and super-ellipsoid can be easily plotted by enumerating all possible parameters θ and ϕ (with a certain interval on the computer, of course) in their definition domain to get a set of coordinates of $\mathbf{x}$. The *plot_surface* and *plot_wireframe* methods in the *matplotlib* library receives the coordinates on the ellipsoid and super-ellipsoid to draw the figure. We can then multiply all $\mathbf{x}$ by the rotation matrix $\mathbf{Q}$, and plus the translation defined by $\mathbf{c}$ for the general ellipsoid and super-ellipsoid if necessary.

Note that due to the same numerical problem in the previous subsection, we choose only to draw part of the super-ellipsoid in Octant I. In order to have the whole super-ellipsoid plotted, we can use a similar strategy that mirrors Octant I to infer the remaining parts by the symmetry property of the origin-centered super-ellipsoid that, if (x, y, z) is on the super-ellipsoid then so as all eight points $(\pm x, \pm y, \pm z)$ be.

III. Results

The experiment is performed on the provided point cloud of a prostate *ProstateSurfacePoints.xlsx*.

A. Fitting with the Ellipsoid

This subsection is corresponding with Questions 1, 2, and 3 in the handout. The result of the fitted parameter vector of the ellipsoid is

$$\mathbf{a} = \begin{bmatrix} 1.2907 \times 10^{-5} & = A \\ 2.3037 \times 10^{-5} & = B \\ 1.6659 \times 10^{-5} & = C \\ 4.3869 \times 10^{-7} & = D \\ 1.5389 \times 10^{-7} & = E \\ 1.7768 \times 10^{-7} & = F \\ -4.1111 \times 10^{-3} & = G \\ -7.8662 \times 10^{-3} & = H \\ -1.8920 \times 10^{-3} & = I \\ 9.9996 \times 10^{-1} & = J \end{bmatrix} \tag{28}$$

The quadratic is validated to be an ellipsoid using the post-check condition

$$(A + B)|\mathbf{A}| = 1.7802 \times 10^{-19} > 0 \tag{29}$$

The center of the ellipsoid is

$$\mathbf{c} = [156.0514, 169.0321, 55.1621]^\top \tag{30}$$

The rotation matrix formed by eigenvectors as columns is

$$\mathbf{Q} = \begin{bmatrix} 0.9996 & -0.0215 & -0.02 \\ -0.0217 & -0.9997 & -0.0142 \\ 0.0196 & -0.0146 & 0.9997 \end{bmatrix} \tag{31}$$

We can see each eigenvector is very close to the corresponding one in $\mathbf{I}_3$, so we treat the ellipsoid as axes-aligned in the following texts. The radii (semi-axis lengths) are

$$\begin{aligned} a &= 54.1369 \\ b &= 40.5078 \\ c &= 47.6402 \end{aligned} \tag{32}$$

The mean value of the distances from the data points to the ellipsoid is 4.5166, and the Root Mean Squared Error (RMSE) is 4.7232.

Note that during the fitting procedure, the datatype (precision) of decimals is important. We use the 64-bit (double) datatype for floating points throughout the experiment. The result will be incorrect if we only use 32-bit (single) precision datatype.

The plots of the ellipsoid and the data points are given in Figure 3 and Figure 4. We can see the ellipsoid is fitted well for the given data points.

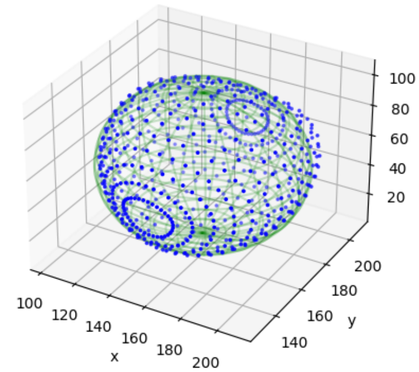

Fig. 3. The fitted ellipsoid (green wire-frame) and the data points (blue dots).

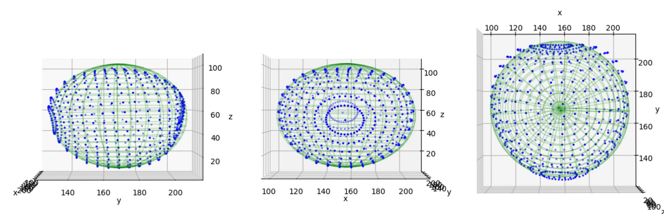

Fig. 4. Different views of the fitted ellipsoid and data points. (a) sagittal, (b) transverse, (c) coronal.

B. Fitting with the Super-Ellipsoid

This subsection is corresponding with Question 4 in the handout.

As the initial condition, the center and radii are inherited from the ellipsoid fitting result, and the tilt matrix is $\mathbf{I}_3$. The CMA-ES is run to minimize the mean value of the distances, with the default hyper-parameters. The optimization converges after 78 iterations, and the error function values during the process is given in Table I. Finally, we get the optimal prediction of the roundnesses

$$\begin{aligned} \epsilon_1 &= 1.0245 \\ \epsilon_2 &= 1.0636 \end{aligned} \tag{33}$$

which means the super-ellipsoid slightly "inflates" the ellipsoid. The mean value of the distances from the data points to the super-ellipsoid is 4.4166, and the Root Mean Squared Error (RMSE) is 4.6364, with a reduction by about 2% each compared with the ellipsoid. This is as expected since the super-ellipsoid has more freedom to better fit the data points.

TABLE I
THE ERROR FUNCTION VALUE DURING THE OPTIMIZATION

# Iteration	Error Function Value
1	5.1811
2	5.8151
3	5.6038
10	4.6569
20	4.6234
30	4.6227
40	4.6227
50	4.6227
60	4.6227
70	4.6227
78	4.6227

Some intermediate results of the optimization evolution is given in Figure 5.

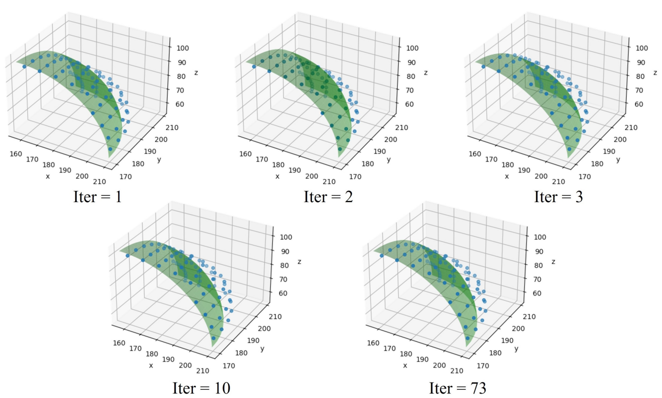

Fig. 5. The intermediate results of the optimization evolution process of different iterations (Octant I). Green: the super-ellipsoid surface; Blue: data points.

IV. Discussions and Conclusions

This report analyzes the ellipsoid fitting problem from scratch. The pitfalls of naive pseudo inverse methods are conquered by the Grammalidis' constraints to convert the fitting procedure into a generalized eigendecomposition problem. We've also shown and proved the way to retrieve important properties, center, radii and tilt of an arbitrary ellipsoid. To fit a super-ellipsoid, this work begins with a method for calculating the distance between the point and the ellipsoid's surface by using the parametric equation of the ellipsoid. The numerical problem introduced by the power term is also overcome utilizing the symmetry property. Then the CMA-ES optimizer is employed to minimize the RMSE. All procedures are validated on a real point cloud dataset.

There still exist some limitations in the study. First, it doesn't describe the detailed procedure to fit an super-ellipsoid which is not axes-aligned, although one can always transform the coordinate with $\mathbf{Q}$ to make it like that. Second, for super-ellipsoid fitting, the radii won't change during the optimization, which might limit the performance to some extent. Third, it's possible to obtain an even better fitting using other quadratics except the super-ellipsoid; for those situations, alternative analytic functions should be derived to model the objective optimization function. All these can be regarded as future research directions.

Acknowledgment

The authors would like to thank Prof. J. L. Dillenseger for providing this chance to dive deep into the ellipsoid fitting problem from which they have learnt a lot.

References

[1] N. Grammalidis, "Using 2-d and 3-d ellipsoid fitting for head and body segmentation and head tracking."
[2] Wikipedia contributors, "Cma-es — Wikipedia, the free encyclopedia," 2023, [Online; accessed 29-June-2023]. [Online]. Available: https://en.wikipedia.org/wiki/CMA-ES
[3] B. Bertoni, "Multi-dimensional ellipsoidal fitting."
[4] J.-L. Dillenseger, "Geometrical modeling (handout for int. joint master degree)."
[5] Y. A. Nikolaus Hansen and P. Baudis, "Cma-es/pycma on github," DOI:10.5281/zenodo.2559634.
[6] Wikipedia contributors, "Ellipsoid — Wikipedia, the free encyclopedia," 2023, [Online; accessed 28-June-2023]. [Online]. Available: https://en.wikipedia.org/wiki/Ellipsoid