# MINIMUM VOLUME ENCLOSING ELLIPSOIDS

NIMA MOSHTAGH
NIMA@GRASP.UPENN.EDU

Abstract. Two different methods for computing the covering ellipses of a set of points are presented. The first method finds the optimal ellipsoids with the minimum volume. The second method uses the first and second moments of the data points to compute the parameters of an ellipsoid that covers most of the points. A MATLAB software is written to verify the results.

## 1. Introduction

In this work we are going to find an approximation to the projected area of an object of interest in the image plane. After detecting the object in the image, one crude way of finding the projected area is counting the number of pixels belonging to the object. But this method is very sensitive to the noise in the image. Another more interesting method is to fit an ellipse or rectangle around the object and approximate its area by the area of the covering well-known shape.

The images that we work with are color images captured by a USB camera, which has a low resolution. The objects of interest are colored boxes and the goal is to find good approximations of the areas of the boxes. A simple color detection algorithm is used that distinguishes bright colors of the boxes from the darker colors of the background. A sample output image of the color detection software is shown in Figure 1.b. The color detection results are highly sensitive to the lighting changes in the environment. As the result, the method of counting the number of pixels in not a robust method.

Instead we try to find an ellipse with minimum volume that encloses all the pixels of a colored box. We approach the problem in two ways. In section 2 we formulate it as an optimization problem and minimize the volume of the covering ellipse. Sections 2, 3 and 4 are based on two papers [2] and [3] that make improvements to the work of [1] who first introduced an algorithm for finding the minimum volume ellipsoids. In section 6, we use the first and second moments of the data points to find an approximate ellipsoid that encloses as many pixels as possible. This part of the paper is based on the work presented in [4].


(a) Original Image (b) Color Detection output

FIGURE 1. The colored boxes.

## 2. MINIMUM VOLUME ENCLOSING ELLIPSES

Consider a set of $m$ point in $n$ dimensional space: $\mathcal{S} = \{x_1, x_2, \ldots, x_m\} \in \mathbb{R}^n$. Let us denote the minimum volume enclosing ellipsoid of the set $\mathcal{S}$ by $MVEE(\mathcal{S})$. In order to guarantee that any ellipsoid containing $\mathcal{S}$ has positive volume, we assume that the affine hull of the set $\mathcal{S}$ spans $\mathbb{R}^n$.

**Definition 2.1.** *An ellipsoid in* center form *is given by*

$$\mathcal{E} = \{x \in \mathbb{R}^n \mid (x - c)^T E (x - c) \leq 1\}$$

*where* $c \in \mathbb{R}^n$ *is the center of the ellipse* $\mathcal{E}$ *and* $E \in \mathbb{S}^n_{++}$.

Since we want that points $x_i$ to be inside $\mathcal{E}$, they must satisfy

$$(x_i - c)^T E (x_i - c) \leq 1 \qquad i = 1, \ldots, m \tag{1}$$

The volume of $\mathcal{E}$ is given by

$$\mathrm{Vol}(\mathcal{E}) = \frac{v_0}{\sqrt{\det(E)}} = v_0 \det(E^{-1})^{\frac{1}{2}}$$

where $v_0$ is the volume of the unit hypersphere in dimension $n$. Thus the problem of determining the ellipsoid of least volume containing the points of $\mathcal{S}$ is equivalent to finding a vector $c \in \mathbb{R}^n$ and an $n \times n$ positive definite symmetric matrix $E$ which minimizes $\det(E^{-1})$ subject to (1).

A natural formulation of the minimum volume enclosing ellipsoid (MVEE) problem is

$$\begin{aligned} &\text{minimize}_{E,c} && \det(E^{-1}) \\ &\text{subject to} && (x_i - c)^T E (x_i - c) \leq 1 \qquad i = 1, \ldots, m \\ & && E \succ 0 \end{aligned} \tag{2}$$

This is not a convex optimization problem. By a change of variable, we can define the ellipsoid as

$$\mathcal{E} = \{x \in \mathbb{R}^n \mid \|Ax - b\|_2 \leq 1\}$$

where $A = E^{1/2}$ and $b = E^{1/2}c$. Thus, the optimization problem (2) becomes

$$\begin{array}{rll} \text{minimize}_{A,b} & \log\det(A^{-1}) & \\ \text{subject to} & \|Ax_i - b\| \leq 1 & i = 1, \dots, m \\ & A > 0 & \end{array} \tag{3}$$

The norm constraints $\|Ax_i - b\| \leq 1$, which are just convex quadratic inequalities in the variables $A$ and $b$, can be expressed as LMIs

$$\begin{bmatrix} I & Ax_i - b \\ (Ax_i - b)^T & 1 \end{bmatrix} \geq 0.$$

Therefore, problem (3) is convex optimization problem in variables $A$ and $b$. However, solving this problem is difficult. Thus, we first modify the primal problem (2) and then find the dual problem. It turns out that the dual problem is easier to solve.

## 3. Dual Formulation and its Solution

We define a "lifting" of $\mathcal{S}$ to $\mathbb{R}^{n+1}$ via $\mathcal{S}' = \{\pm q_1, \dots, \pm q_m\}$ where $q_i^T = [x_i^T, 1]$ $i = 1, \dots, m$. By this definition each point $x_i$ is lifted to the hyperplane $H = \{(x, x_{n+1}) \in \mathbb{R}^{n+1} \mid x_{n+1} = 1\}$. Since $\mathcal{S}'$ is centrally symmetric, MVEE($\mathcal{S}'$) is centered at the origin. The minimum volume enclosing ellipsoid of the original problem is recovered as the intersection of $H$ with the MVEE containing the lifted points $q_i$:

$$MVEE(\mathcal{S}) = MVEE(\mathcal{S}') \cap H$$

The "lifted" primal problem becomes

$$\begin{array}{rll} \text{minimize}_{M} & \log\det(M^{-1}) & \\ \text{subject to} & q_i^T M q_i \leq 1 & i = 1, \dots, m \\ & M > 0 & \end{array} \tag{4}$$

where $M \in \mathbb{S}_{++}^{(n+1)}$ is the decision variable.

Let $P$ denote the $n \times m$ matrix whose columns are the vectors $p_i$, i.e. $P = [p_i, \dots, p_m]$, then the matrix $Q$ whose columns are the vectors $q_i$ is given by

$$Q = [q_1, \dots, q_m] = \begin{bmatrix} P \\ \mathbf{1}^T \end{bmatrix} \in \mathbb{R}^{(n+1)\times m}.$$

The Lagrangian dual problem is given by

$$\begin{array}{ll} \text{maximize}_z & \log\det V(z) \\ \text{subject to} & \mathbf{1}^T z = d+1 \\ & z \geq 0 \end{array}$$

where $V(z) = Q\text{diag}(z)Q^T$ and $z \in \mathbb{R}^m$ is the decision variable. The change of variable $z = (d+1)u$ results in the following dual problem:

$$\begin{array}{ll} \text{maximize}_u & \log\det V(u) \\ \text{subject to} & \mathbf{1}^T u = 1 \\ & u \geq 0 \end{array} \tag{5}$$

where $V(u) = QUQ^T$ and $U = \text{diag}(u) \in \mathbb{R}^{m\times m}$. Problem (5) is a concave optimization problem, so we use an *ascent method* to find the maximum.

A general ascent method is as follows. Let $f(u)$ be the objective function and $\bar{u}$ be a feasible point, $\bar{u} \in \text{dom}(f)$. The search direction in an ascent method must satisfy

$$\nabla f(\bar{u})^T \Delta\bar{u} > 0$$

where $\Delta\bar{u}$ is the *ascent direction* for $f$ at $\bar{u}$. The following algorithm outlines a general ascent method.

**Algorithm 3.1.** General Ascent Method
***given*** *a staring point* $u \in dom(f)$.
***repeat***

- *Determine an ascent direction* $\Delta u$
- *Line search. Choose a step size* $\alpha > 0$.
- *Update.* $u \leftarrow u + \alpha\Delta u$

***until*** *stopping criterion is satisfied.*

We are going to find the ascent direction $\Delta u$ and the step size $\alpha$. Let $g(u) = [g_1(u), \ldots, g_m(u)]^T$ be the gradient of the objective function in (5). Thus

$$g_i(u) = \frac{\partial \log\det V(u)}{\partial u_i} = q_i^T V(u)^{-1} q_i \ .$$

Let $\bar{u}$ be the current iterate value. At each iteration of the conditional gradient method, we compute the gradient $g(\bar{u})$ of the objective function of the dual and solve a *linearization* of the dual problem at $\bar{u}$:

$$\begin{array}{ll} \text{maximize}_u & g(\bar{u})^T u \\ \text{subject to} & \mathbf{1}^T u = 1 \\ & u \geq 0 \end{array} \tag{6}$$

The optimal solution of this problem is given by the $j$-th unit vector $e_j \in \mathbb{R}^m$, where

$$j = \arg\max_i g_i(\bar{u}) \ .$$

Therefore, the ascent direction is $\Delta u = e_j - \bar{u}$.

At the line search stage, we need to solve the problem

$$\max_{\alpha \in [0,1]} \log\det V(\bar{u} + \alpha(e_j - \bar{u}))$$

which was shown in [1] that has the closed form solution

$$\alpha = \frac{g_j(\bar{u}) - (n+1)}{(n+1)(g_j(\bar{u}) - 1)} \ .$$

This leads us to the following algorithm that iteratively solves the dual problem (5).

**Algorithm 3.2.** Conditional Gradient Ascent
***Initialization.*** *Let* $u \leftarrow (1/n)\mathbf{1}$.
***repeat***

- *Ascent Direction.*
  *Compute* $g_i(u) = q_i V(u)^{-1} q_i, \ i = 1, \dots, m.$
  *Set* $j = \arg\max_i g_i(u)$ .
  *Set* $\Delta u = e_j - u$.
- *Line search.*
  $$\alpha \leftarrow \frac{g_j(u) - (n+1)}{(n+1)(g_j(u) - 1)} \ .$$
- *Update.* $u \leftarrow u + \alpha \Delta u$

***until*** *stopping criterion is satisfied.*

In the next section we show how the solution of the dual problem is used to find the parameters of the covering ellipsoids.

## 4. Computing the Parameters of the Covering Ellipse

Consider the primal problem with the lifted points $q_i$

$$\begin{array}{lll} \text{minimize}_M & \log\det(M^{-1}) & \\ \text{subject to} & q_i^T M q_i \le 1 & i = 1, \dots, m \\ & M > 0 & \end{array} \tag{7}$$

The Lagrangian is given by

$$L(M, z) = -\log\det M + \sum_{i=1}^{m} z_i (q_i^T M q_i - 1).$$

By the KKT conditions for optimality we must have

$$\frac{\partial L}{\partial M} = -M^{-1} + \sum_{i=1}^{m} z_i q_i q_i^T = -M^{-1} + QZQ^T = 0$$

where $Z = \text{diag}(z)$ and $Q = [q_1, \ldots, q_m]$. This implies that when a positive definite matrix $M^* \in \mathbb{R}^{(n+1)\times(n+1)}$ is optimal for the primal (7) with the lagrangian multipliers $z^* \in \mathbb{R}^m$, then we have

$$V(z^*) = QZ^*Q^T = (M^*)^{-1} = (d+1)V(u^*)\ .$$

Given $q^T = [x^T\ 1]$, the equation of the ellipsoid is given by

$$\begin{aligned} MVEE(\mathcal{S}) &= \{x \in \mathbb{R}^n \mid q^T M^* q \leq 1\} \\ &= \{x \in \mathbb{R}^n \mid (\frac{1}{d+1})\ q^T V(u^*)^{-1} q \leq 1\}. \end{aligned}$$

Therefore, given the solution of the dual problem (5) we can find the equation of the ellipse. Note that we have

$$V(u) = QUQ^T = \begin{bmatrix} PUP^T & Pu \\ (Pu)^T & \mathbf{1}^T u \end{bmatrix}$$

which can be factorized as

$$V(u) = \begin{bmatrix} I & Pu \\ 0 & 1 \end{bmatrix} \begin{bmatrix} E^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ (Pu)^T & 1 \end{bmatrix}$$

where $E^{-1} = PUP^T - Pu(Pu)^T$. The inverse $V(u)^{-1}$ is given by

$$V(u)^{-1} = \begin{bmatrix} I & 0 \\ -(Pu)^T & 1 \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -Pu \\ 0 & 1 \end{bmatrix}.$$

Thus, we get

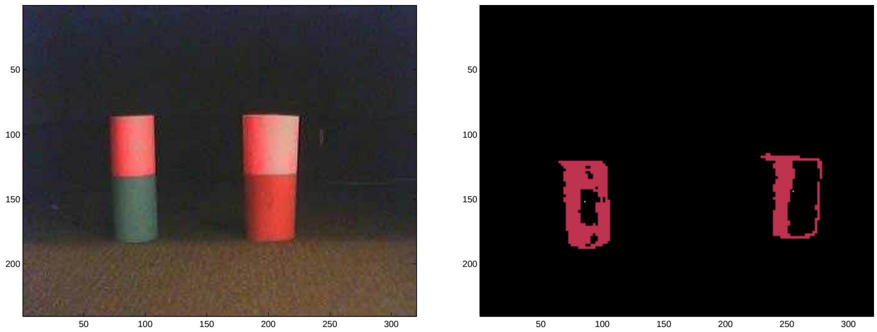
$$q^T V(u)^{-1} q = (x - Pu)^T E (x - Pu).$$

Therefore, for the dual optimal solution $u^*$ we have

$$\boxed{\begin{gathered} MVEE(\mathcal{S}) = \{x \in \mathbb{R}^n \mid (x - c^*)^T E^* (x - c^*) \leq 1\} \\ \text{where} \\ E^* = \frac{1}{d}(PU^*P^T - Pu^*(Pu^*)^T)^{-1}, \quad c^* = Pu^* \end{gathered}}$$

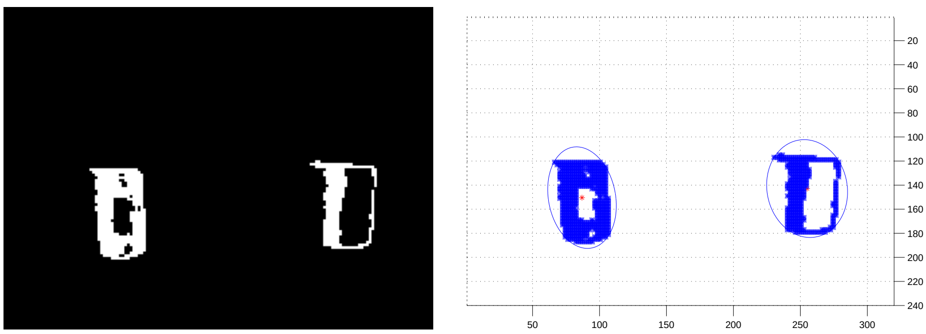
## 5. Implementation

Images are captured by an *IREZ* color USB camera. Figure 2.a shows a sample image with colored cylinders. A color detection code, written in C++, is applied on this image and Figure 2.b is the corresponding output. A MATLAB code was written to compute the minimum volume enclosing ellipse (MVEE). As the input, it takes a binary image with 1 at pixels with the desired color (see Figure 3.a). Algorithm (3.2) has been implemented in `MVE_dual_solver.m` to solve the dual problem (5) and the its solution was used by `MVE_from_dual.m` to compute the MVEE for each detected colored cylinder. The output of the MATLAB function `MVEE_segmentation.m` is shown in Figure 3.b. This function is capable of detecting multiple objects in an image and computing a MVEE for each one.




(a) Original Image (b) Color Detection output

Figure 2. The colored cylinders are detected in the image.




(a) Binary Image (b) MVEE overlaid on data points

Figure 3. The binary image with the detected cylinders and the MVEEs for the cones in the image.

## 6. Moment-Based Structuring - Nonoptimal

In this section we present a different method for covering the data points with ellipses, which uses the first and second moments of the data points. This result is introduced in [4].

Let $A$ be a binary image, and $p = (x, y)$ be a pixel of $A$. The $(p, q)$ moment of the binary image $A$ is defined as

$$m_{pq} = \sum_{A} x^p \ y^q$$

where the summation extends over all the pixels in $A$. $m_{00}$ is known as the zero-th moment. $m_{10}$ and $m_{01}$ are the first, and $m_{20}$ and $m_{02}$ are the second moments.

The goal is to find the equation of an ellipse that covers as many data points as possible in the binary image $A$. The ellipse can be identified by its centroid $c = (x_c, y_c)$, the major and minor axis $l, w$, the angle $\theta$ between the major-axis of the ellipse and the x-axis:

$$x_c = \frac{m_{10}}{m_{00}} \tag{8}$$

$$y_c = \frac{m_{01}}{m_{00}} \tag{9}$$

$$\text{Inclination angle:} \quad \theta = \frac{1}{2}\text{atan2}(b, a - c) \tag{10}$$

$$\text{minor axis:} \quad w = \sqrt{3(a + c - \sqrt{b^2 + (a - c)^2})} \tag{11}$$

$$\text{major axis:} \quad l = \sqrt{3(a + c + \sqrt{b^2 + (a - c)^2})} \tag{12}$$

where $a$, $b$ and $c$ are defined as:

$$a = \frac{m_{20}}{m_{00}} - x_c^2 \tag{13}$$

$$b = 2(\frac{m_{11}}{m_{00}} - x_c \ y_c) \tag{14}$$

$$c = \frac{m_{02}}{m_{00}} - y_c^2 \tag{15}$$

As it is shown in Figure 4, the ellipses that were computed in this way are good approximations to the minimum volume ellipses shown in Figure 3.b. The areas of the moment-based ellipses were slightly larger than that of the corresponding MVEEs. However, because computing a moment-based ellipse is much faster than finding the MVEE, the moment-based method is more suitable for real-time applications.

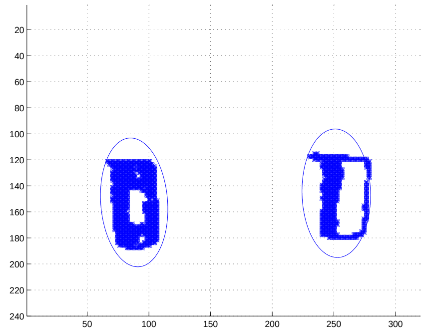



FIGURE 4. The covering ellipses are computed by using the moments of the data points.

## References

1. L. G. Khachiyan, *Rounding of polytopes in the real number model of computation*, Mathematics of Operations Research **21** (1996), 307–320.
2. P. Kumar and E. A. Yildirim, *Mimimum volume enclosing ellipsoids and core sets*, (2005).
3. Sun Peng and R. M. Freund, *Computation of minimum volume covering ellipsoids*, Operations Research **52** (2004), 690–706.
4. L.; Carvalho P.C.P. Carvalho Rocha, L.; Velho, *Image moments-based structuring and tracking of objects*, XV Brazilian Symposium on Computer Graphics and Image Processing, 2002.