Algorithms for Determining the Convex Hull of a 3D Finite Set

Problem: We are given a finite set $\mathcal{X} = \{x_i \mid i \in \mathcal{I}\} \subset \mathbb{R}^3$ satisfying $\dim(\mathcal{X}) = 3$. To find is the set $\mathcal{F}$ of all faces of the convex hull $\mathrm{conv}(\mathcal{X})$. Let us characterize a face $F \in \mathcal{F}$ by $F = [i_1, i_2, \ldots, i_j]$, which means that the set of its vertexes $\mathcal{V}_F$ and the set of its edges $\mathcal{E}_F$ are given by

$$\mathcal{V}_F = \{x_{i_1}, x_{i_2}, \ldots, x_{i_j}\},$$
$$\mathcal{E}_F = \{[x_{i_1}, x_{i_2}], [x_{i_2}, x_{i_3}], \ldots, [x_{i_j}, x_{i_1}]\}.$$

Note that $[x_i, x_{i'}] = [x_{i'}, x_i]$.

It is possible that $x_i = x_{i'}$ for $i \neq i'$, i.e., there are points with different names but with the same vector value. This situation happens often when generating random vectors with integer values.

As result, all vertexes or edges or faces with the same values are identified.

Main ideas:

- Transformation $\mathcal{X} \subset \mathbb{R}^3 \to \mathcal{Y} \subset \mathbb{R}^2$;
- The idea of the method of orienting curves;
- Replacing the evaluation of determinants through the evaluation of linear functions;
- Compute determinants and linear functions in two steps;
- Edge indexing.

Storing faces and edges

For the original data $\mathcal{X} = \{x_i \mid i \in \mathcal{I}\} \subset \mathbb{R}^3$, correspondingly to every $i \in \mathcal{I}$, the coordinates of $x_i = (x_{i1}, x_{i2}, x_{i3})$ are stored. After fixing these data, for any edge, we only need to store the index pair $[i_1, i_2]$ instead of $[x_{i_1}, x_{i_2}]$. Since $[x_{i_1}, x_{i_2}] = [x_{i_2}, x_{i_1}]$, there holds $[i_1, i_2] = [i_2, i_1]$, too. Similarly, for any face, we store the index tuple $[i_1, i_2, \ldots, i_j]$ instead of $[x_{i_1}, x_{i_2}, \ldots, x_{i_j}]$, and there holds

$$[x_{i_1}, x_{i_2} \ldots, x_{i_{j'}}, x_{i_{j'+1}}, \ldots, x_{i_j}] = [x_{i_{j'+1}}, x_{i_1} \ldots, x_{i_j}, x_{i_1}, \ldots, x_{i_{j'}}],$$
$$[i_1, i_2 \ldots, i_{j'}, i_{j'+1}, \ldots, i_j] = [i_{j'+1}, i_1 \ldots, i_j, i_1, \ldots, i_{j'}].$$

We can write in short

$$[x_{i_1}, x_{i_2}] = x_{[i_1, i_2]}, \quad [x_{i_1}, x_{i_2}, \ldots, x_{i_j}] = x_{[i_1, i_2, \ldots, i_j]}.$$

Finding edges of faces

(Needed for all convex hull algorithms finding proper faces and edges)

Theorem: Let $\{d_1, d_2, d_3\} \subset \mathbb{R}^3$ be an orthogonal system with $d_1 \neq 0$, $d_2 \neq 0$, $d_3 \neq 0$, and $\mathcal{X}_I = \{x_i \mid i \in I\} \subset \mathbb{R}^3$. Assume

$$\dim(\mathcal{X}_I) = 2 \qquad (ee21?)$$

and

$$\begin{aligned}&\text{there exist } x_{i_1}, x_{i_2}, x_{i_3} \in \mathcal{X}_I \text{ such that} \\ &x_{i_2} - x_{i_1}, x_{i_3} - x_{i_1}, \text{ and } d_3 \text{ are linearly independent.}\end{aligned} \qquad (1)$$

Let

$$\mathcal{Y}_I = \{y_i \mid i \in I\} \subset \mathbb{R}^2, \quad \text{where } y_i = \begin{pmatrix} y_{i1} \\ y_{i2} \end{pmatrix} = \begin{pmatrix} d_1^T x_i \\ d_2^T x_i \end{pmatrix}.$$

Then x_i is a vertex of $\operatorname{conv}(\mathcal{X}_I)$ iff y_i is a vertex of $\operatorname{conv}(\mathcal{Y}_I)$.

If (ee21?) holds true, then (ee22?) is satisfied at least for $d_3 = (1,0,0)^T$ or $d_3 = (0,1,0)^T$ or $d_3 = (0,0,1)^T$. Hence, we can choose

$$\{d_1, d_2\} = \{(1,0,0)^T, (0,1,0)^T, (0,0,1)^T\} \setminus d_3.$$

Then no operation must be done for the projection $x_i \mapsto y_i$. Hence, in order to determine the convex hull of some 2-dimensional subset $\mathcal{X}_I$ of $\mathbb{R}^3$, we only have to choose two suitable coordinates of x_i to get the corresponding $\mathcal{Y}_I \subset \mathbb{R}^2$ and to determine the convex hull of $\mathcal{Y}_I$.

(ee22?) is fulfilled if

$$\det \begin{pmatrix} x_{i_2 1} - x_{i_1 1} & x_{i_2 2} - x_{i_1 2} & x_{i_2 3} - x_{i_1 3} \\ x_{i_3 1} - x_{i_1 1} & x_{i_3 2} - x_{i_1 2} & x_{i_3 3} - x_{i_1 3} \\ d_{31} & d_{32} & d_{33} \end{pmatrix} \neq 0.$$

Concretely, we may choose

$$d_3 = \begin{cases} (0,0,1)^T & \text{if } \det \begin{pmatrix} x_{i_2 1} - x_{i_1 1} & x_{i_2 2} - x_{i_1 2} \\ x_{i_3 1} - x_{i_1 1} & x_{i_3 2} - x_{i_1 2} \end{pmatrix} \neq 0, \\ (0,1,0)^T & \text{if } \det \begin{pmatrix} x_{i_2 1} - x_{i_1 1} & x_{i_2 3} - x_{i_1 3} \\ x_{i_3 1} - x_{i_1 1} & x_{i_3 3} - x_{i_1 3} \end{pmatrix} \neq 0, \\ (1,0,0)^T & \text{if } \det \begin{pmatrix} x_{i_2 2} - x_{i_1 2} & x_{i_2 3} - x_{i_1 3} \\ x_{i_3 2} - x_{i_1 2} & x_{i_3 3} - x_{i_1 3} \end{pmatrix} \neq 0. \end{cases}$$

Note that, as $x_{i_2} - x_{i_1}$ and $x_{i_3} - x_{i_1}$ are linearly independent, at least one of the three just mentioned determinants is nonzero, and we choose only one of them if two or three determinants are nonzero.

Projective wrapping algorithm for finding 3D convex hull
(16.2.2011)

Main idea: Projection

1. Introduction

Let

$$\mathcal{X} = \{x_i \mid i \in \mathcal{I}\} \subset \mathbb{R}^3, \quad \text{where } \mathcal{I} = \{1, 2, \ldots, n\} \text{ and } \dim(\mathcal{X}) = 3.$$

Our aim is to find the 3D convex hull of $\mathcal{X}$, which is denoted by $\operatorname{conv}(\mathcal{X})$.

There are several algorithms for finding convex hull in three dimensions. Some basic algorithms are gift wrapping algorithm, which was discovered by Chand and Kapur in 1970 [CK70] and Jarvis in 1973 [J73], incremental algorithm published in1984 by Kallay [K84], and quickhull algorithm, which was discovered by Eddy in 1977 [E77] and by Bykat in 1978 [B78]. Some other algorithms use such basic algorithms for finding convex hulls of subsets and then combine them together, e.g., divide-and-conquer algorithm, which was presented in 1977 by Preparata and Hong [PH77], and Chan's algorithm, which was introduced in 1996 by Chan [C96].

In this paper, we are particularly interested in gift wrapping algorithm, which was developed further by many authors, e.g., in [GCF09], [S94], [S85], [B97], etc.

The main idea of this paper to use suitable projections to replace the problem of finding faces in three dimensions by the problem of finding edges in two dimensions, which makes the evaluation of the point position much cheaper. By such a way, we get the so-called projective wrapping algorithm, which works essentially faster than conventional gift wrapping algorithm for 3D convex hull.

In contrary to many other papers on convex hull algorithms, we do not require $x_i \neq x_{i'}$ for $i \neq i'$, and do not exclude collinarity and coplanarity, i.e., $\mathcal{X}$ may have three different points on a line and four different points on a plane. This release is necessary for doing with more general point sets, in particular, with relatively large point sets, which are generated by computer.

When allowing collinarity and coplanarity, the problem of finding edges of two-dimensional faces becomes much more complicated and expensive, and must be treated extra in addition. But this is not a subject of this paper. We assume that this subproblem is solved sufficiently somewhere else, and refer, for instance, to [???].

As usual, in order to denote the components of $x_i \in \mathbb{R}^3$ and $y_i \in \mathbb{R}^2$, we write

$$x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \end{pmatrix}, \quad y_i = \begin{pmatrix} y_{i1} \\ y_{i2} \end{pmatrix}.$$

By storing the coordinates of all points $x_i \in \mathcal{X}$ according to their indexes $i \in \mathcal{I}$, we can use corresponding indexes to indicate points. Thus, the index pair $[i_1, i_2]$ is stored and mentioned briefly for the edge $[x_{i_1}, x_{i_2}]$ of $\operatorname{conv}(\mathcal{X})$. Similarly, if a face F of $\operatorname{conv}(\mathcal{X})$ is described by $[x_{i_1}, x_{i_2}, \ldots, x_{i_j}]$, where $x_{i_1}, x_{i_2}, \ldots, x_{i_j}$ are its sequentially ordered vertexes, then F is stored and mentioned by the index tuple $[i_1, i_2, \ldots, i_j]$. Although the face F is the whole convex hull $\operatorname{conv}\{x_{i_1}, x_{i_2}, \ldots, x_{i_j}\}$, we abbreviate it by $F = [i_1, i_2, \ldots, i_j]$.

2. Algorithms

Algorithm 10b: Let $\mathcal{F}$ be the set of found faces and $\mathcal{E}^1$ be the set of edges which belong to exactly one found face. Begin with $\mathcal{X}' = \mathcal{X}$.

Step 1. Find the first face F of $\operatorname{conv}(\mathcal{X}')$. Set $\mathcal{F} = \{F\}$ and $\mathcal{E}^1 = \mathcal{E}_F$, where $\mathcal{E}_F$ is the set of all edges of the face F.

Step 2. While $\mathcal{E}^1 \neq \emptyset$ do the following task cycle:

- Take an (arbitrary) $E \in \mathcal{E}^1$, which is an edge of some already found face $F^{\text{old}} \in \mathcal{F}$, i.e., $E \in \mathcal{E}_{F^{\text{old}}}$, and find the other face $F^{\text{new}} \neq F^{\text{old}}$ of $\operatorname{conv}(\mathcal{X}')$, which shares the edge E, i.e., $E \in \mathcal{E}_{F^{\text{new}}}$.

- Replace $\mathcal{F} = \mathcal{F} \cup \{F^{\text{new}}\}$ and $\mathcal{E}_1 = (\mathcal{E}_1 \cup \mathcal{E}_{F^{\text{new}}}) \setminus (\mathcal{E}_1 \cap \mathcal{E}_{F^{\text{new}}})$.

- Delete from $\mathcal{X}'$ those points in the face F^{new}, which are no vertexes, and among all vertexes with the same vector value, keep only one of them.

Step 3. Output $\mathcal{F}$. $\square$

In this first level, Algorithm 10b has almost the same scheme as conventional gift wrapping algorithm. The main difference takes place in the second level, where the still open problem of finding the face F^{new} is treated. Besides, the act of deleting points, which turn out to not be vertexes, is meaningful only if collinarity and coplanarity are allowed and if the problem of finding edges of two-dimensional faces is solved sufficiently well.

Theorem: $\mathcal{F}$ obtained by Algorithm 10b is the set of all faces of the convex hull $\operatorname{conv}(\mathcal{X})$.

In order to simplify the three-dimensional problem, we choose some orthogonal system of nonzero vectors $\{d_1, d_2, d_3\} \subset \mathbb{R}^3$, i.e.,

$$d_1 \neq 0,\ d_2 \neq 0,\ d_3 \neq 0,\ d_1^T d_2 = d_1^T d_3 = d_2^T d_3 = 0, \qquad (e2e5?)$$

and consider the following two-dimensional projection

$$\mathcal{Y} = \{y_i \mid i \in \mathcal{I}\} \subset \mathbb{R}^2, \quad \text{where } y_i = \begin{pmatrix} y_{i1} \\ y_{i2} \end{pmatrix} = \begin{pmatrix} d_1^T x_i \\ d_2^T x_i \end{pmatrix}. \qquad (e2e6?)$$

As usual in many convex hull algorithms, we use determinants for evaluating the position of points. Denote

$$D_{\mathcal{Y}}(y_{i_1}, y_{i_2}, y_{i_3}) = \det \begin{pmatrix} 1 & y_{i_1 1} & y_{i_1 2} \\ 1 & y_{i_2 1} & y_{i_2 2} \\ 1 & y_{i_3 1} & y_{i_3 2} \end{pmatrix}. \qquad (e2e7?)$$

Algorithm 11b: To find is the first face of $\operatorname{conv}(\mathcal{X})$.

Step 1. Take the orthogonal system $\{d_1, d_2, d_3\}$ defined by

$$d_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad d_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad d_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Step 2. Consider the set $\mathcal{Y} = \{y_i \mid i \in \mathcal{I}\} \subset \mathbb{R}^2$ defined by (e2e6?). Determine an $i^* \in \mathcal{I}$ satisfying

$$y_{i^*1} = \max\{y_{j1} \mid j \in \mathcal{I},\ y_{j2} = \min_{i \in \mathcal{I}} y_{i2}\}$$

and an $i^{**} \in \mathcal{I}$ satisfying

$$y_{i^{**}} \neq y_{i^*}, \quad D_{\mathcal{Y}}(y_{i^*}, y_{i^{**}}, y_i) \geq 0 \text{ for all } i \in \mathcal{I}.$$

Let

$$I^* = \{i \in \mathcal{I} \mid D_{\mathcal{Y}}(y_{i^*}, y_{i^{**}}, y_i) = 0\}.$$

Step 3. If $\dim\{x_i \mid i \in I^*\} = 1$, then choose another orthogonal system of nonzero vectors $\{d_1, d_2, d_3\}$ with $d_3 = y_{i^{**}} - y_{i^*}$, and go back to Step 2. Otherwise, if $\dim\{x_i \mid i \in I^*\} = 2$, then $\mathrm{conv}\{x_i \mid i \in I^*\}$ is a face F of $\mathrm{conv}(\mathcal{X})$. Determine the set $\mathcal{E}_F$ of all edges of F. □

Theorem: The convex hull $\mathrm{conv}\{x_i \mid i \in I^*\}$ yielded by Algorithm 11b is a face the convex hull $\mathrm{conv}(\mathcal{X})$.

In Algorithm 11b, the projection $x_i \mapsto y_i$ is applied for finding the first face of $\mathrm{conv}(\mathcal{X})$. In order to use such a projection to find further faces, we base on the following property.

Theorem: Suppose that $[x_{i_1}, x_{i_2}]$ is an edge of the convex hull $\mathrm{conv}(\mathcal{X})$, the system $\{d_1, d_2, d_3\} \subset \mathbb{R}^3$ satisfies

$$d_1 \neq 0,\ d_2 \neq 0,\ d_3 = x_{i_2} - x_{i_1} \neq 0,\ d_1^T d_2 = d_1^T d_3 = d_2^T d_3 = 0, \tag{e2e8?}$$

and $\mathcal{Y}$ is defined by (e2e6?). If F_1 and F_2 are two different faces of $\mathrm{conv}(\mathcal{X})$, which share the common edge $[x_{i_1}, x_{i_2}]$, and

$$\mathcal{Y}_{F_1} = \{y_i \mid i \in \mathcal{I}, x_i \in F_1\}, \quad \mathcal{Y}_{F_2} = \{y_i \mid i \in \mathcal{I}, x_i \in F_2\},$$

then $\mathrm{conv}(\mathcal{Y}_{F_1})$ and $\mathrm{conv}(\mathcal{Y}_{F_2})$ are two different edges of $\mathrm{conv}(\mathcal{Y})$, which share the common vertex y_{i_1}. On the other hand, if E_1 and E_2 are two different edges of $\mathrm{conv}(\mathcal{Y})$, which share the common vertex y_{i_1}, and

$$\mathcal{X}_{E_1} = \{x_i \mid i \in \mathcal{I}, y_i \in E_1\}, \quad \mathcal{X}_{E_2} = \{x_i \mid i \in \mathcal{I}, y_i \in E_2\},$$

then $\mathrm{conv}(\mathcal{X}_{E_1})$ and $\mathrm{conv}(\mathcal{X}_{E_2})$ are two different edges of $\mathrm{conv}(\mathcal{X})$, which share the common edge $[x_{i_1}, x_{i_2}]$.

(e2e8?) implies that, for all $i \in \mathcal{I}$, there holds

$$\begin{aligned} y_{i_2 1} - y_{i_1 1} &= d_1^T x_{i_2} - d_1^T x_{i_1} = d_1^T (x_{i_2} - x_{i_1}) = d_1^T d_3 = 0, \\ y_{i_2 2} - y_{i_1 2} &= d_2^T x_{i_2} - d_2^T x_{i_1} = d_2^T (x_{i_2} - x_{i_1}) = d_2^T d_3 = 0. \end{aligned}$$

In the following, we assume that the vertexes of the face $F^{\mathrm{old}} = [i_1, i_2, \ldots, i_j]$ are ordered counterclockwise when viewing from the interior of the convex hull $\mathrm{conv}(\mathcal{X})$, i.e.,

$$D_{\mathcal{X}}(x_{i_1}, x_{i_2}, x_{i_3}, x_i) > 0 \quad \text{for some } i \in \mathcal{I}, \tag{ee1?}$$

where

$$D_{\mathcal{X}}(x_{i_1}, x_{i_2}, x_{i_3}, x_i) = \det \begin{pmatrix} 1 & x_{i_1 1} & x_{i_1 2} & x_{i_1 3} \\ 1 & x_{i_2 1} & x_{i_2 2} & x_{i_2 3} \\ 1 & x_{i_3 1} & x_{i_3 2} & x_{i_3 3} \\ 1 & x_{i1} & x_{i2} & x_{i3} \end{pmatrix}. \qquad (e2e9?)$$

Algorithm 13b: Let $F^{\text{old}} = [i_1, i_2, \ldots, i_j]$ be an already found face of $\text{conv}(\mathcal{X})$ and $[x_{i_1}, x_{i_2}]$ be an edge of F^{old}. Suppose (ee1?). Our aim is to find the other face $F^{\text{new}} \neq F^{\text{old}}$ of $\text{conv}(\mathcal{X})$, which shares with F^{old} the common edge $[x_{i_1}, x_{i_2}]$.

Step 1. Choose an orthogonal system $\{d_1, d_2, d_3\} \subset \mathbb{R}^3$ satisfying (e2e8?) and determine $\mathcal{Y}$ according to (e2e6?).

Step 2. Find the set K of all $k \in \mathcal{I}$ such that

$$y_k \neq y_{i_1}, \quad D_{\mathcal{Y}}(y_{i_1}, y_k, y_i) \geq 0 \ \text{ for all } i \in \mathcal{I}$$

as follows. Begin with

$$K = \{k\}, \quad i = k,$$

where k is the first index from $\mathcal{I}$ satisfying $y_k \neq y_{i_1}$.

Step 2.1. If $\mathcal{I}' = \{i' \in \mathcal{I} \mid i < i' \leq n, \ y_{i'} \neq y_{i_1}\}$ is empty, then go to Step 3. Otherwise replace i by $\min \mathcal{I}'$.

Step 2.2. Do the following:

$$\begin{array}{llll} \text{if} & D_{\mathcal{Y}}(y_{i_1}, y_k, y_i) = 0 & \text{then} & K = K \cup \{i\}, \\ \text{if} & D_{\mathcal{Y}}(y_{i_1}, y_k, y_i) < 0 & \text{then} & K = \{i\}, \ k = i, \\ \multicolumn{4}{l}{\text{go to Step 2.1.}} \end{array}$$

Step 3. The convex hull $\text{conv}\{x_k \mid k \in K \cup \{i_1, i_2\}\}$ is the new face F^{new} of $\text{conv}(\mathcal{X})$, which we look for. Determine the set $\mathcal{E}_{F^{\text{new}}}$ of all edges of F^{new}. □

Theorem: $\mathcal{F}$ obtained by Algorithm 10b is the set of all faces of the convex hull $\text{conv}(\mathcal{X})$.

In order to simplify the three-dimensional problem, we choose some orthogonal system of nonzero vectors $\{d_1, d_2, d_3\} \subset \mathbb{R}^3$, i.e.,

$$d_1 \neq 0, \ d_2 \neq 0, \ d_3 \neq 0, \ d_1^T d_2 = d_1^T d_3 = d_2^T d_3 = 0, \qquad (e2e5?)$$

and consider the following two-dimensional projection

$$\mathcal{Y} = \{y_i \mid i \in \mathcal{I}\} \subset \mathbb{R}^2, \quad \text{where } y_i = \begin{pmatrix} y_{i1} \\ y_{i2} \end{pmatrix} = \begin{pmatrix} d_1^T x_i \\ d_2^T x_i \end{pmatrix} \qquad (e2e6?)$$

As usual in many convex hull algorithms, we use determinants for evaluating the position of points. Denote

$$D_{\mathcal{Y}}(y_{i_1}, y_{i_2}, y_{i_3}) = \det \begin{pmatrix} 1 & y_{i_1 1} & y_{i_1 2} \\ 1 & y_{i_2 1} & y_{i_2 2} \\ 1 & y_{i_3 1} & y_{i_3 2} \end{pmatrix}. \qquad (e2e7?)$$

Algorithm 11b: To find is the first face of $\text{conv}(\mathcal{X})$.

Step 1. Take the orthogonal system $\{d_1, d_2, d_3\}$ defined by

$$d_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad d_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad d_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Step 2. Consider the set $\mathcal{Y} = \{y_i \mid i \in \mathcal{I}\} \subset \mathbb{R}^2$ defined by (e2e6?). Determine an $i^* \in \mathcal{I}$ satisfying

$$y_{i^*1} = \max\{y_{j1} \mid j \in \mathcal{I},\ y_{j2} = \min_{i\in\mathcal{I}} y_{i2}\}$$

and an $i^{**} \in \mathcal{I}$ satisfying

$$y_{i^{**}} \neq y_{i^*}, \quad D_{\mathcal{Y}}(y_{i^*}, y_{i^{**}}, y_i) \geq 0 \ \text{ for all } i \in \mathcal{I}.$$

Let

$$I^* = \{i \in \mathcal{I} \mid D_{\mathcal{Y}}(y_{i^*}, y_{i^{**}}, y_i) = 0\}.$$

Step 3. If $\dim\{x_i \mid i \in I^*\} = 1$, then choose another orthogonal system of nonzero vectors $\{d_1, d_2, d_3\}$ with $d_3 = y_{i^{**}} - y_{i^*}$, and go back to Step 2. Otherwise, if $\dim\{x_i \mid i \in I^*\} = 2$, then $\mathrm{conv}\{x_i \mid i \in I^*\}$ is a face F of $\mathrm{conv}(\mathcal{X})$. Determine the set $\mathcal{E}_F$ of all edges of F. □

Theorem: The convex hull $\mathrm{conv}\{x_i \mid i \in I^*\}$ yielded by Algorithm 11b is a face the convex hull $\mathrm{conv}(\mathcal{X})$.

In Algorithm 11b, the projection $x_i \mapsto y_i$ is applied for finding the first face of $\mathrm{conv}(\mathcal{X})$. In order to use such a projection to find further faces, we base on the following property.

Theorem: Suppose that $[x_{i_1}, x_{i_2}]$ is an edge of the convex hull $\mathrm{conv}(\mathcal{X})$, the system $\{d_1, d_2, d_3\} \subset \mathbb{R}^3$ satisfies

$$d_1 \neq 0,\ d_2 \neq 0,\ d_3 = x_{i_2} - x_{i_1} \neq 0,\ d_1^T d_2 = d_1^T d_3 = d_2^T d_3 = 0, \qquad (e2e8?)$$

and $\mathcal{Y}$ is defined by (e2e6?). If F_1 and F_2 are two different faces of $\mathrm{conv}(\mathcal{X})$, which share the common edge $[x_{i_1}, x_{i_2}]$, and

$$\mathcal{Y}_{F_1} = \{y_i \mid i \in \mathcal{I}, x_i \in F_1\}, \quad \mathcal{Y}_{F_2} = \{y_i \mid i \in \mathcal{I}, x_i \in F_2\},$$

then $\mathrm{conv}(\mathcal{Y}_{F_1})$ and $\mathrm{conv}(\mathcal{Y}_{F_2})$ are two different edges of $\mathrm{conv}(\mathcal{Y})$, which share the common vertex y_{i_1}. On the other hand, if E_1 and E_2 are two different edges of $\mathrm{conv}(\mathcal{Y})$, which share the common vertex y_{i_1}, and

$$\mathcal{X}_{E_1} = \{x_i \mid i \in \mathcal{I}, y_i \in E_1\}, \quad \mathcal{X}_{E_2} = \{x_i \mid i \in \mathcal{I}, y_i \in E_2\},$$

then $\mathrm{conv}(\mathcal{X}_{E_1})$ and $\mathrm{conv}(\mathcal{X}_{E_2})$ are two different edges of $\mathrm{conv}(\mathcal{X})$, which share the common edge $[x_{i_1}, x_{i_2}]$.

(e2e8?) implies that, for all $i \in \mathcal{I}$, there holds

$$\begin{aligned} y_{i_21} - y_{i_11} &= d_1^T x_{i_2} - d_1^T x_{i_1} = d_1^T (x_{i_2} - x_{i_1}) = d_1^T d_3 = 0, \\ y_{i_22} - y_{i_12} &= d_2^T x_{i_2} - d_2^T x_{i_1} = d_2^T (x_{i_2} - x_{i_1}) = d_2^T d_3 = 0. \end{aligned}$$

In the following, we assume that the vertexes of the face $F^{\text{old}} = [i_1, i_2, \dots, i_j]$ are ordered counterclockwise when viewing from the interior of the convex hull $\text{conv}(\mathcal{X})$, i.e.,

$$D_{\mathcal{X}}(x_{i_1}, x_{i_2}, x_{i_3}, x_i) > 0 \quad \text{for some } i \in \mathcal{I}, \tag{ee1?}$$

where

$$D_{\mathcal{X}}(x_{i_1}, x_{i_2}, x_{i_3}, x_i) = \det \begin{pmatrix} 1 & x_{i_1 1} & x_{i_1 2} & x_{i_1 3} \\ 1 & x_{i_2 1} & x_{i_2 2} & x_{i_2 3} \\ 1 & x_{i_3 1} & x_{i_3 2} & x_{i_3 3} \\ 1 & x_{i1} & x_{i2} & x_{i3} \end{pmatrix}. \tag{e2e9?}$$

Algorithm 13b: Let $F^{\text{old}} = [i_1, i_2, \dots, i_j]$ be an already found face of $\text{conv}(\mathcal{X})$ and $[x_{i_1}, x_{i_2}]$ be an edge of F^{old}. Suppose (ee1?). Our aim is to find the other face $F^{\text{new}} \neq F^{\text{old}}$ of $\text{conv}(\mathcal{X})$, which shares with F^{old} the common edge $[x_{i_1}, x_{i_2}]$.

Step 1. Choose an orthogonal system $\{d_1, d_2, d_3\} \subset \mathbb{R}^3$ satisfying (e2e8?) and determine $\mathcal{Y}$ according to (e2e6?).

Step 2. Find the set K of all $k \in \mathcal{I}$ such that

$$y_k \neq y_{i_1}, \quad D_{\mathcal{Y}}(y_{i_1}, y_k, y_i) \geq 0 \ \text{ for all } i \in \mathcal{I}$$

as follows. Begin with

$$K = \{k\}, \quad i = k,$$

where k is the first index from $\mathcal{I}$ satisfying $y_k \neq y_{i_1}$.

Step 2.1. If $\mathcal{I}' = \{i' \in \mathcal{I} \mid i < i' \leq n, \ y_{i'} \neq y_{i_1}\}$ is empty, then go to Step 3. Otherwise replace i by $\min \mathcal{I}'$.

Step 2.2. Do the following:

if $\ D_{\mathcal{Y}}(y_{i_1}, y_k, y_i) = 0 \ $ then $\ K = K \cup \{i\}$, if $\ D_{\mathcal{Y}}(y_{i_1}, y_k, y_i) < 0 \ $ then $\ K = \{i\}, \ k = i$, go to Step 2.1.

Step 3. The convex hull $\text{conv}\{x_k \mid k \in K \cup \{i_1, i_2\}\}$ is the new face F^{new} of $\text{conv}(\mathcal{X})$, which we look for. Determine the set $\mathcal{E}_{F^{\text{new}}}$ of all edges of F^{new}. □

Theorem: The convex hull $\text{conv}\{x_k \mid k \in K \cup \{i_1, i_2\}\}$ yielded by Algorithm 13b is a face of $\text{conv}(\mathcal{X})$, which is different from the given face F^{old} and has $[x_{i_1}, x_{i_2}]$ as an edge.

3. Efficiency of suitable projection

In the previous section, we need a vector system $d_1, d_2, d_3 \in \mathbb{R}^3$ satisfying (e2e5?) and $d_3 = y_{i^{**}} - y_{i^*}$ (in Algorithm 11b) or $d_3 = x_{i_2} - x_{i_1}$ (in Algorithm 13b). For given $d_3 = (\delta_1, \delta_2, \delta_3)^T \in \mathbb{R}^3$, we have to choose special vectors $d_1, d_2 \in \mathbb{R}^3$ in order to decrease the number of necessary operations when computing $y_{i1} = d_1^T x_i$ and $y_{i2} = d_2^T x_i$.

In general, if all components of d_1 and d_2 are nonzeros, then 4 additions and 6 multiplications are needed for each pair (y_{i1}, y_{i2}). But the number of operations be reduced by choosing d_1 and d_2 as follows.

If $\delta_2 = \delta_3 = 0$, then we choose

$$d_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad d_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

In this case, $y_{i1} = x_{i2}$ and $y_{i2} = x_{i3}$, i.e., no operation is needed.

If $\delta_2 = 0$ and $\delta_3 \neq 0$, then we choose

$$d_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad d_2 = \begin{pmatrix} 1 \\ 0 \\ -\delta_1/\delta_3 \end{pmatrix}.$$

In this case, $y_{i1} = x_{i2}$ and $y_{i2} = x_{i1} + d_{23}x_{i3}$, i.e., one addition and one multiplication are needed for computing each pair (y_{i1}, y_{i2}).

If $\delta_2 \neq 0$ and $\delta_3 = 0$, then we choose

$$d_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad d_2 = \begin{pmatrix} 1 \\ -\delta_1/\delta_2 \\ 0 \end{pmatrix}.$$

In this case, $y_{i1} = x_{i3}$ and $y_{i2} = x_{i1} + d_{22}x_{i2}$, i.e., one addition and one multiplication are needed for computing each pair (y_{i1}, y_{i2}).

If $\delta_2 \neq 0$ and $\delta_3 \neq 0$, then we choose

$$d_1 = \begin{pmatrix} 0 \\ -\delta_3/\delta_2 \\ 1 \end{pmatrix}, \quad d_2 = \begin{pmatrix} 1 \\ -\delta_1\delta_2/(\delta_2^2 + \delta_3^2) \\ -\delta_1\delta_3/(\delta_2^2 + \delta_3^2) \end{pmatrix}.$$

In this case, $y_{i1} = d_{12}x_{i2} + x_{i3}$ and $y_{i2} = x_{i1} + d_{22}x_{i2} + d_{23}x_{i3}$, i.e., 3 additions and 3 multiplications are needed for computing each pair (y_{i1}, y_{i2}).

By choosing d_1 and d_2 as above, we can reduce at least one of the 4 additions and 3 of the 6 multiplications. This advantage is rather high.

Note that if only integer coordinates are allowed, as often assumed in many papers on convex hull algorithms, then division must be avoided. In this case, when $\delta_2 \neq 0$ and $\delta_3 \neq 0$, we may choose

$$d_1 = \begin{pmatrix} 0 \\ \delta_3 \\ -\delta_2 \end{pmatrix}, \quad d_2 = \begin{pmatrix} -\delta_2^2 - \delta_3^2 \\ \delta_1\delta_2 \\ \delta_1\delta_3 \end{pmatrix}.$$

Then 3 additions and 5 multiplications are needed for computing each pair (y_{i1}, y_{i2}), i.e., at least one addition and one multiplication can be saved.

Now, let us explain the advantage of our convex hull algorithms using the projection $x_i \mapsto y_i$. Normally, in order to evaluate the position of the point $x_{i_4} \in \mathcal{X} \subset \mathbb{R}^3$ with respect to the three given points $x_{i_1}, x_{i_2}, x_{i_3} \in \mathcal{X}$, one uses the determinant

$$D_{\mathcal{X}}(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}) = \det \begin{pmatrix} 1 & x_{i_1 1} & x_{i_1 2} & x_{i_1 3} \\ 1 & x_{i_2 1} & x_{i_2 2} & x_{i_2 3} \\ 1 & x_{i_3 1} & x_{i_3 2} & x_{i_3 3} \\ 1 & x_{i_4 1} & x_{i_4 2} & x_{i_4 3} \end{pmatrix},$$

whose computation according to this original definition needs 23 additions and 36 multiplications. After projecting $\mathcal{X} \to \mathcal{Y}$, we works in two dimensions. In order to evaluate the position of the point $y_{i_3} \in \mathcal{Y} \subset \mathbb{R}^2$ with respect to the two given points $y_{i_1}, y_{i_2} \in \mathcal{Y}$, we use the determinant

$$D_{\mathcal{Y}}(y_{i_1}, y_{i_2}, y_{i_3}) = \det \begin{pmatrix} 1 & y_{i_1 1} & y_{i_1 2} \\ 1 & y_{i_2 1} & y_{i_2 2} \\ 1 & y_{i_3 1} & y_{i_3 2} \end{pmatrix},$$

whose computation according to this original definition needs 5 additions and 6 multiplications. Together with at most 3 additions and 3 multiplications needed for computing a pair (y_{i1}, y_{i2}), at most 8 additions and 9 multiplications are needed for evaluating a point $y_i \in \mathbb{R}^2$. That is very little in comparison to 23 additions and 36 multiplications for evaluating a point $x_i \in \mathbb{R}^3$.

A better way to compute $D_{\mathcal{X}}(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4})$ and $D_{\mathcal{Y}}(y_{i_1}, y_{i_2}, y_{i_3})$ is based on

$$\det \begin{pmatrix} 1 & x_{i_1 1} & x_{i_1 2} & x_{i_1 3} \\ 1 & x_{i_2 1} & x_{i_2 2} & x_{i_2 3} \\ 1 & x_{i_3 1} & x_{i_3 2} & x_{i_3 3} \\ 1 & x_{i_4 1} & x_{i_4 2} & x_{i_4 3} \end{pmatrix} = \det \begin{pmatrix} x_{i_2 1} - x_{i_1 1} & x_{i_2 2} - x_{i_1 2} & x_{i_2 3} - x_{i_1 3} \\ x_{i_3 1} - x_{i_1 1} & x_{i_3 2} - x_{i_1 2} & x_{i_3 3} - x_{i_1 3} \\ x_{i_4 1} - x_{i_1 1} & x_{i_4 2} - x_{i_1 2} & x_{i_4 3} - x_{i_1 3} \end{pmatrix},$$

and

$$\det \begin{pmatrix} 1 & y_{i_1 1} & y_{i_1 2} \\ 1 & y_{i_2 1} & y_{i_2 2} \\ 1 & y_{i_3 1} & y_{i_3 2} \end{pmatrix} = \det \begin{pmatrix} y_{i_2 1} - y_{i_1 1} & y_{i_2 2} - y_{i_1 2} \\ y_{i_3 1} - y_{i_1 1} & y_{i_3 2} - y_{i_1 2} \end{pmatrix}.$$

Then 14 additions and 9 multiplications are needed for computing $D_{\mathcal{X}}(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4})$, while 5 additions and 2 multiplications are needed for computing $D_{\mathcal{Y}}(y_{i_1}, y_{i_2}, y_{i_3})$. Together with at most 3 additions and 3 multiplications needed for computing a pair (y_{i1}, y_{i2}), we need at most 8 additions and 5 multiplications for evaluating a point $y_i \in \mathbb{R}^2$. That means that the evaluation of a point in $\mathcal{X} \subset \mathbb{R}^3$ by using $D_{\mathcal{X}}(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4})$ is at least almost 80% more expensive ($6/8 = 75\%$ additions and $4/5 = 80\%$ multiplications more) than the evaluation of a point in $\mathcal{Y} \subset \mathbb{R}^2$ by using $D_{\mathcal{Y}}(y_{i_1}, y_{i_2}, y_{i_3})$. In other words, for evaluating a point position, doing by projecting $x_i \mapsto y_i$ can save at least $6/14 \approx 42.9\%$ additions and $4/9 \approx 44.4\%$ multiplications. These ratios change a bit for the total evaluation when finding a face, since while the projection $x_i \mapsto y_i$ must be done for all n points x_i, $i \in \mathcal{I}$, only $n-3$ determinants must be calculated.

Note that the total number of operations can be reduced more if the determinant computation is divided into two steps. This and some further ideas will be presented in other papers.

References

[1] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa, The Quickhull algorithm for convex hulls, ACM Transactions on Mathematical Software, Vol. 22, pp. 469–483, 1996.

[2] K. H. Borgwardt, Average complexity of a gift-wrapping algorithm for determining the convex hull of randomly given points, Discrete & Computational Geometry, Vol. 17, pp. 79–109, 1997.

[3] A. Bykat, Convex hull of a finite set of points in two dimensions, Inf. Process. Lett., Vol. 7, pp. 296–298, 1978.

[4] Timothy M. Chan, Optimal output-sensitive convex hull algorithms in two and three dimensions, Discrete and Computational Geometry, Vol. 16, pp. 361-368, 1996.

[5] D. R. Chand, S. S. Kapur, An algorithm for convex polytopes, J. ACM, Vol. 7, pp. 78–86, 1970.

[6] B. Chazelle and J. Matousek, Derandomizing an output-sensitive convex hull algorithm in three dimensions, Computational Geometry 5 (1995) 27–32

[7] K.L. Clarkson and P.W. Shor Applications of random sampling in computational geometry, Discr. Comput. Geom. 4, 387–421, 1989.

[8] W. Eddy, A new convex hull algorithm for planar sets, ACM Trans. Math. Softw., Vol. 3, pp. 398–403, 1977.

[9] Y. Gerard, D. Coeurjolly, and F. Feschet, Gift-wrapping based preimage computation algorithm, Pattern Recognition, Vol. 42, pp. 2255–2264, 2009.

[10] R. A. Jarvis, On the identification of the convex hull of a finite set of points in the plane, Info. Proc. Letters, Vol. 2, pp. 18–21, 1973.

[11] M. Kallay, The Complexity of Incremental Convex Hull Algorithms in $\mathbb{R}^d$, Info. Proc. Letters 19, p. 197 (1984)

[12] F. P. Preparata and S. J. Hong, Convex hulls of finite sets of points in two and three dimensions, Communications of the ACM, Volume 20, pp. 87–93, 1977.

[13] K. Sugihara, Robust gift wrapping for the three-dimensional convex hull, JOURNAL OF COMPUTER AND SYSTEM SCIENCES 49, 391–407 (1994)

[14] G. Swart, Finding the convex hull facet by facet, JOURNAL OF ALGORITHMS 6,17-48 (1985)

3D algorithms:

Chan's algorithm

Divide-and-Conquer published in 1977 by Preparata and Hong

Incremental convex hull algorithm - O(n log n) Published in 1984 by Michael Kallay.

Randomized Incremental Algorithm This algorithm was first described by K.L. Clarkson and P.W. Shor in 1989 [CS89]

QuickHull Discovered independently in 1977 by W. Eddy and in 1978 by A. Bykat

----

Convex Hull Algorithms

Convex hull algorithms

Preparata, Shamos, Computational Geometry, Chapter ”Convex Hulls: Basic Algorithms”

Luc Devroye and Godfried Toussaint, ”A note on linear expected time algorithms for finding convex hulls,” Computing, Vol. 26, 1981, pp. 361-366.

Avis, David; Bremner, David; Seidel, Raimund (1997), ”How good are convex hull algorithms?”, Computational Geometry: Theory and Applications 7 (5-6): 265-301, doi:10.1016/S0925-7721(96)00023-5 .

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 33.3: Finding the convex hull, pp. 947-957.

Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf (2000). Computational Geometry (2nd revised edition ed.). Springer-Verlag. ISBN 3-540-65620-0. Section 1.1: An Example: Convex Hulls (describes classical algorithms for 2-dimensional convex hulls). Chapter 11: Convex Hulls: pp. 235-250 (describes a randomized algorithm for 3-dimensional convex hulls due to Clarkson and Shor).

Marriage-before-conquest – Kirkpatrick-Seidel algorithm

Kirkpatrick, David G.; Seidel, Raimund (1986). ”The ultimate planar convex hull algorithm”. SIAM Journal on Computing 15 (1): 287-299.

2D, 3D, and dD Convex Hull in CGAL, the Computational Geometry Algorithms Library