Convex Hulls

Let me know
what you think

The algorithm described in this section can construct the convex hull of any finite set of points or cells of a cell complex which are given by their Cartesian coordinates. However, in the case of a set of voxels, which consists of a few connected components, the algorithm can be essentially accelerated. The idea of this acceleration is based on the fact that the convex hull of the whole set is equal to the convex hull of a relatively small subset of "distinguished" cells or of their "distinguished" vertices.

Definition

We define the convex hull of a finite set S of points as the smallest convex polyhedron PH containing S. The polyhedron PH is convex if the coordinates of all its vertices satisfy a set of linear inequalities $H_i(x, y, z) \geq 0$; $(i=1, 2, ..., m)$; where m is the number of faces of PH. Distinguished are points which are no convex combinations of other points of S. We call these points *local corners*. It is possible to take either the principal cells (voxels or pixels, see "a" in the figure below) or their 0-dimensional vertices ("b").


a   b
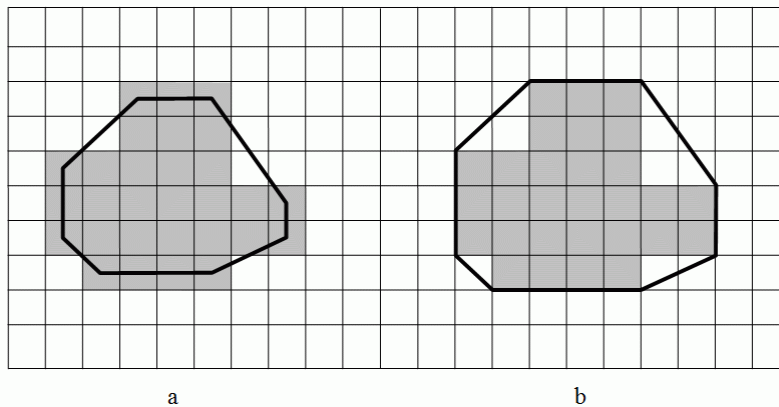
A point P is a convex combination of the points x_j ($j=1, 2, ..., n$), if $P=$ $\sum_{j=1}^{n} t_j x_j$ where the sum of the non-negative values t_i is equal to 1.
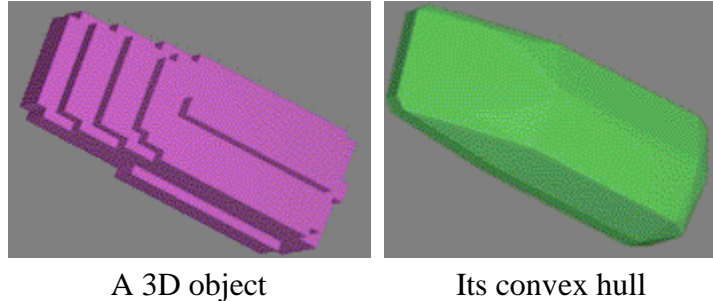
Idea of the Algorithm

We describe the surface of the 3D convex hull by a 2D cell list (see the section "Block Complexes and Cell Lists" above). The algorithm starts with producing a list L of all local corners. It chooses four non-coplanar points of L and produces the cell list of the spanning tetrahedron. This is the "current hull" CH which will be completed step by step to the convex hull of L. The algorithm tests each local corner of L while finding all faces of CH which are

"visible" from the local corner. The face F of a convex polyhedron is visible from a point P if P lies in the outer open half-space bounded by F, i.e. if the scalar product (N, W) of the outer normal N of the face F and the vector W pointing from an arbitrary point Q in F to P, is positive. If the scalar product is negative, then F is said to be invisible from P. If the scalar product is equal to zero, then F is said to be coplanar with P. If one or more faces are visible, then the polyhedron is being extended by the point P and some new faces. Each new face connects P with one of the edges of the boundary of the set of visible faces. A new face is a triangle having P as its vertex and one of the edges of the said boundary as its base. All triangles are included in the cell list of the current hull, while all visible faces are removed from the list. Also each edge incident to two visible faces is removed. When all local corners are processed in this way then the convex hull is ready. The figure below explains the idea of the algorithm by a 2D example. The starting triangle is drawn in black lines. Polyhedrons in red and blue are following.
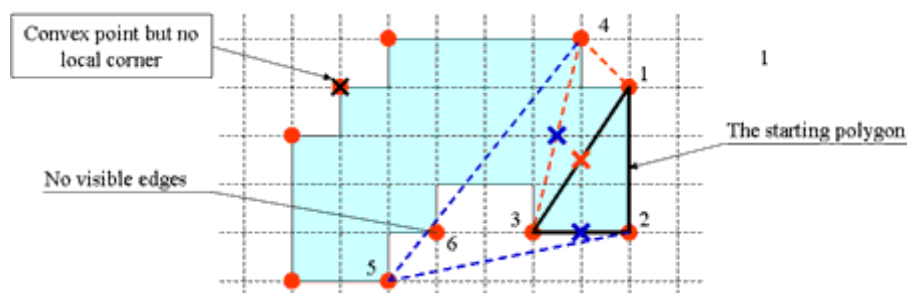
The following figure shows an example of a 3D body and its convex hull.

A 3D object   Its convex hull

The Algorithm for the 3D Case

1. Make the list of all local corners.
2. Find four local corners which are not coplanar and construct the cell list of the tetrahedron. This is the current hull.
3. For each local corner P find all visible faces of the current hull. If there are no such faces then P lies inside the hull and should be skipped. Otherwise connect P with all edges of the visible area, put P and the new triangles into the list and delete the visible area.
4. Repeat item 3 for all local corners.
5. If there are some coplanar triangles merge them to polygons.
End of algorithm.

If you are interested in details see the references [1] and [2].

References

[1] V. Kovalevsky and H. Schulz, *Convex Hulls in a 3-dimensional Space*, In: Klette, R. and Zunic, J. (Eds): Lecture Notes in Computer Science, Vol. 3322, (2004), pp. 175-191...

[2] V. Kovalevsky: *Geometry of Locally Finite Spaces*, Monograph, Berlin 2008.