Concave Hull of a Set of Freeform Closed Surfaces in R^3

A. V. Vishwanath and M. Ramanathan

Indian Institute of Technology Madras

ABSTRACT

Bounding hulls, such as *convex hulls*, have been shown to be useful in many application areas. Ŭ-hull, a generalization of convex hull, has been predominantly employed in reconstruction. Other hulls such as *concave hull*, which generates non-convex polygons poly hull, r-shape and s-shape etc. have also been shown to be useful. Most algorithms for bounding hulls deal with point-set as input. Recently, we approached the question of bounding hull to a set of close planar freeform curves and proposed an algorithm for concave hull by defining it (definition was lacking even for a point-set). In this paper, we extend it to concave hull of a set of freeform closed surfaces in R^3 of genus 0. Surfaces used are represented as NURBS (non-uniform rational B-splines). The concept of concave hull is then extended for genus > 0 and showed that the hull will consists of lower dimensional elements and topological disks. Based on this observation, a conjecture is proposed for elements in the concave hull of freeform closed objects in n dimensions.

Keywords: convex/concave hull, enclosing surface, freeform surfaces.
DOI: 10.3722/cadaps.2012.857-868

1 INTRODUCTION

In the domain of point sets in R^3, convex hull [7, 21] of a set of points, defined as the convex combination of a set of points is a prominent one and algorithms exist for computing it [21]. Similar to convex hull, the generalized version named alpha hull (Ŭ -hull) [9] (whose discrete counterpart is □-shape [10]) has also been defined in R^3 as well. The alpha shape uses a real parameter Ŭ, variations of which leads to a family of shapes. The output of the alpha shape need not necessarily be convex nor connected.

Applications of convex hull range from interference checking [16] to shape matching [6]. However, one of the disadvantages of the convex hull is that, at times it does not best represent the area occupied by the input set. Recently, *Concave hull*, which appears to have been introduced in [14] (they call it as non-convex footprints) and developed further in [1, 19], is an enclosure for the given set that represents the area occupied by the points by generating non-convex polygons. Figure 1(a) shows a convex hull for a set of points in R^2, whereas the concave hull of the same set of points, shown in Figure 1(b). A tighter enclosure can be achieved using concave hull than using convex hull. For concave hull of set of points, a user-controlled parameter, called as tuning parameter is used to smooth the concave hull.

Concave hull, so far, has been employed only in two-dimensions (2D) and no known extensions are available in three-dimension (3D). Other shapes such as r-shape and s-shape [5], □-shape [8], A-shape [18] etc. have been shown only for 2D, though they could be extended for 3D as most of them use Delaunay triangulation as the basis for computing the shape.

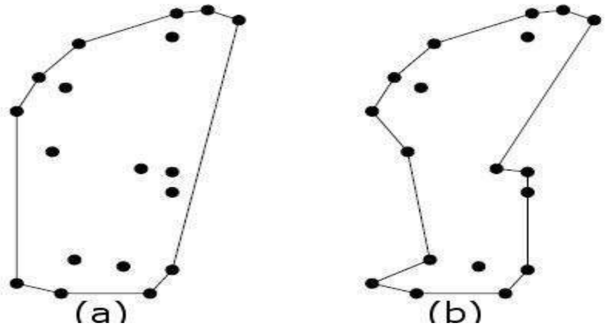


Fig. 1: Convex hull vs. concave hull [2].

Concave hull for a set of points, even in R^2, so far, does not seem to have a precise definition [19,26]. In particular, convex hull is the minimum perimeter as well as minimum area convex enclosure of the set of points. However, for non-convex enclosures, such objectives often conflict each other i.e., minimizing area and perimeter is not possible simultaneously [2] and one has to find a common ground, which leads to non-unique solutions called *Poly hulls*.

Concave hulls are ideally suited for fencing applications. Imagine a set of large number of trees and houses surrounding the trees. In case of fencing of trees, the bounding area will be greatly reduced when it is applied around the concave hull (since the area of concave hull is going to be much smaller than that of convex hull, in general). Concave hulls can also be employed in otherô ôareas such as de-featuring [17], molecular shape matching, geographic information processing, imageô ôprocessing, pattern recognition, and feature detection [3].

When the domain involves curves and surfaces, algorithm for computing convex hull has been presented [13, 23]. Recently, algorithmôs efficiency in [13] was improved using biarc approximation in [15]. For set of straight line and circular segments, an algorithm for computing convex hull has been presented in [25].

In this paper, the input surfaces are parametric freeform surfaces represented using NURBS [22], and are closed C^1-continuous, non self-intersecting that bound a closed volume in R^3. All the objects are assumed to be having genus 0 and also simply-connected and no portions of any two surfaces overlap each other. Moreover, the input surfaces are used as such, i.e. without sampling them. Figure 2 shows the input surfaces used in this paper.

Defnition 1 Concave hull of a set of surfaces is the enclosing concave surface with smallest volume.

Definition 1 is extended from the one that was defined for a set of curves [24]. This definition also indicates that the boundary of a concave hull will be simply-connected, very similar to the fact that a convex hull is a simply-connected one. Typically, concave hull implies the boundary along with the underlying volume, though while computing we do compute the boundary only (in a way similar to most of the algorithms for convex hull). Following are the major contributions of the paper:

- The definition for concave hull has been extended to R^3.
- For genus 0 objects, as in curves in R^2, in R^3, concave hull can be obtained similar to a boolean union operation and for non-intersection surfaces, antipodal points are identified and a minimum spanning tree approach has been proposed.

- For genus >0, the concave hull will be composed of lower-dimensional entities and topological disks.

For objects in R^n with hyper surfaces in R^n , a conjecture is proposed on the elements that constitute the concave hull.

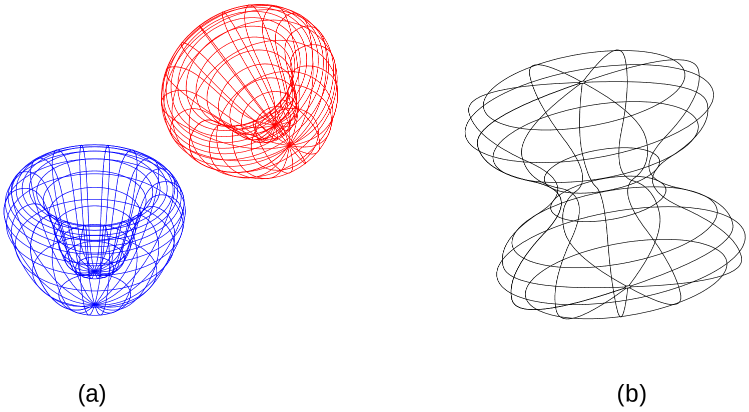


Fig. 2: Examples of simply connected surfaces: (a) two simply connected surfaces (b) another example.

Remainder of this paper is structured as follows: Section 2 describes the approach for determining the concave hull when only two curves are present. Section 3 extends the approach presented in section 2 for a given set of curves and discussed in section 4. Section 5 concludes the paper.

2 CONCAVE HULL OF TWO SURFACES

For a single surface, concave hull is the surface itself. Similar to set of curves, determining concave hull is started by investigating the following cases for two surfaces:

- one surface lying completely outside the other;
- Surfaces that intersect each other.
- One surface lying completely inside the other;

2.1 One Surface is Completely Outside the Other

Consider two surfaces that do not intersect and are outside each other (Figure 3(a)). We also assume that the surface contain no plane portions. Imagine an elastic cover that is tightly enclosing them, such as the convex hull between the two (Figure 3(b). As one keeps pushing the convex hull and still enclosing them (Figure 3(c)), it will reach a stage where the volume of the enclosure between them tends to become null. The initial surface and the zero volume line between them will form the minimum volume concave enclosure (Figure 3(d)). Though one can find different lines that can amount to zero volume, we employ the following approach as it facilitates a computational methodology. The minimal distance occurs when the points on the respective surfaces are antipodal to each other (Lemma 1).

Lemma 1 *The distance between two closed C^1 non-intersecting surfaces is minimum only when the normals of the corresponding points are opposite to each other (i.e., antipodal).*

2.1.1 *Antipodal constraint of two surfaces in R^3*

Consider two closed C^2-continuous surfaces $S_1(u1, v1)$ and $S_2(u2,v2)$ (Figure 4). Then, each of the antipodality constraint is represented by two equations [20].

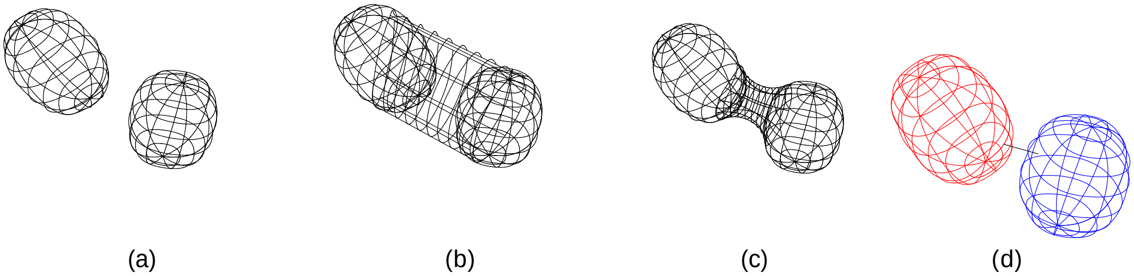


Fig. 3: Elastic cover analogy to get concave hull for two non-intersecting surfaces: (a) a pair of surfaces outside each other, (b) convex hull of surfaces, (c) pushing the convex hull of the surfaces, (d) corresponding concave hull.

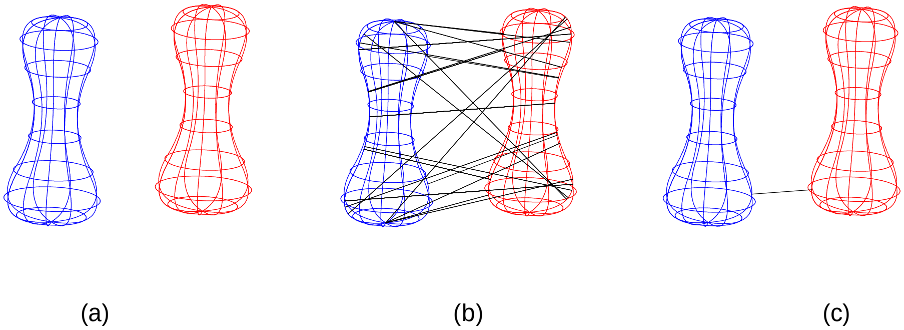
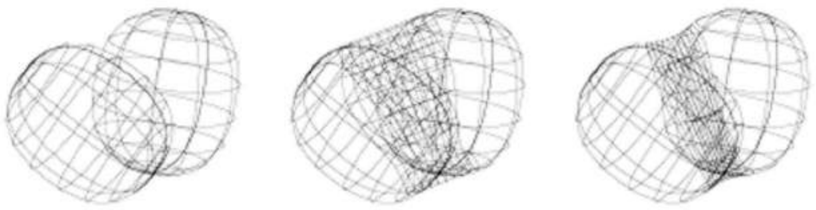


Fig. 4: Antipodality constraint for two free-form surfaces: (a) A pair of surfaces outside each other, (b) Output showing all antipodal lines, (c) Output showing MAL.

$$
\begin{aligned}
\left\langle \frac{\partial S_1(u_1,v_1)}{\partial u_1}, S_1(u_1,v_1) - \frac{S_1(u_1, v_1 + S_2(u_2,v_2)}{2} \right\rangle &= 0,\\
\left\langle \frac{\partial S_1(u_1,v_1)}{\partial v_1}, S_1(u_1,v_1) - \frac{S_1(u_1, v_1 + S_2(u_2,v_2)}{2} \right\rangle &= 0,\\
\left\langle \frac{\partial S_2(u_2,v_2)}{\partial u_2}, S_2(u_2,v_2) - \frac{S_1(u_1, v_1 + S_2(u_2,v_2)}{2} \right\rangle &= 0,\\
\left\langle \frac{\partial S_2(u_2,v_2)}{\partial v_2}, S_2(u_2,v_2) - \frac{S_1(u_1, v_1 + S_2(u_2,v_2)}{2} \right\rangle &= 0,
\end{aligned}
\tag{1}
$$

Thus, in (1) we have four equations in four unknowns (*u*1, *v*1, *u*2 and *v*2). Figure 4 illustrates the antipodal constraint for two surfaces. In practice, the solution of the constraint equations (Equations in 1)) results in a finite set of candidates (an infinite set of candidates is possible in certain degeneracies, such as when surfaces have planar portions. They are not considered in this paper). From this set (which gives a set of (*u*1*, v1*) and corresponding (*u*2*, v*2) values), evaluate the points on the surfaces and then find the minimum distance. Assuming that there is only one set of points contributing to minimum distance between the two surfaces, concave hull is then the two surfaces and the line between the minimum distance points.



(a) (b) (c)

Fig. 5: Elastic enclosure analogy to get concave hull for two curves that intersect: (a) a pair of surfaces that intersect each other, (b) convex hull of the surfaces, (c) pushing that convex hull of the curves.

Definition 2 The line connecting the two antipodal points that is minimum in distance is called *minimum antipodal line (MAL)* and the points as minimum antipodal points (MAP).

2.2 Two Intersecting Surfaces

Consider two surfaces that intersect each other (Figure 5(a)). Imagine an elastic enclosure that is tightly enclosing them, such as the convex hull between the two (Figure 5(b). As one keeps pushing the convex hull and still enclosing the surfaces (Figure 5(c)), it will reach a point where the enclosure cannot be moved beyond the points where the intersection of curves happen, which will then the yield minimum volume concave enclosure.

When two surfaces are intersecting (Figure 6(a)), the minimum distance between them is zero. A quick check to determine whether the surfaces intersect or not, is to employ the convex hull of the control polyhedrons of the two surfaces. One can use the òleftó predicate (Chapter 1, [21]) to do this operation. If they do not intersect, the curves for sure do not intersect each other. However, if they intersect, the surfaces are then processed for intersection, which typically amounts to polynomial root finding of the two surfaces. Do note that the antipodal condition need not be satisfied at the points of intersection. As the surfaces are intersecting, the intersection curve (Figure 6(b)) between the two surfaces is then identified.

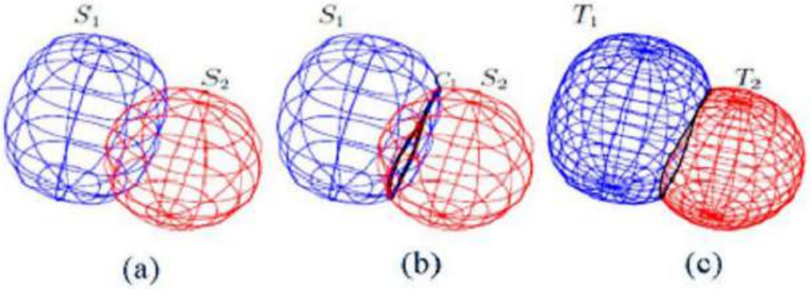


Fig. 6: (a) two intersecting surfaces S_1 and S_2, (b) intersection curve as trimming curve, (c) concave hull.

The intersection curve is considered in the parametric domain and the trimming operation is performed by Boolean subtraction (Figure 7). The trimming operation removes portion of the surface enclosed by the trimming curve. This essentially removes the portions of surface lying inside another. The resultant will be a set of surfaces forming the concave hull (Figure 6(c)). For genus = 0 surfaces, please note that the process of computing concave hull of two intersecting surfaces emulates Boolean union of two surfaces. As Boolean union yields a unique result, so will be the concave hull, for intersecting surfaces.

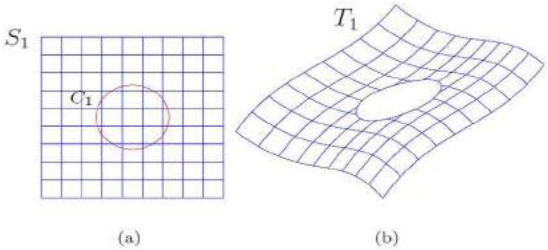


Fig. 7: (a) Surface S_1 and the trimming curve C_1 in the parametric space (b) trimmed surface.

It is to be noted that the intersection between two simply-connected surfaces can result in the following output;

- Simply-connected object with genus 0
- Simply-connected object with genus > 0.
- Multiply-connected object.

Figure 8(a) shows to surfaces of dumbbell-like shape and their intersection curves in Figure 8(b). The Boolean union between the two is shown in Figure 8(c) and the output results in genus > 0. If one can image an enclosure between two dumbbells and start pushing it, there will be a surface formation along with the Boolean output. The surface is yet again a zero-volume one. In this paper, to find such a surface, we connect two points of the intersecting curve via geodesic curves (Figure 8(d)). Considering the two curves, a ruled surface is fit between the two curves, resulting in a degenerate ruled surface patch (DRSP). Output concave hull is shown in Figure 8(e).

When two simply-connected surfaces are employed, the resultant output can be multiply-connected as well. For example, for the two surfaces in Figure 9(a), the Boolean output is multiply-connected (Figure 9(b)). The inner surface is then removed to obtain the concave hull (Figure 9(c)).

Proposition 1 In R^3, assuming DRSPsô does not intersect any surface in the set, the concave hull will consist of lower dimensional entities (straight lines), topological discs (DRSPôs) and surfaces in R^3.

Proposition 1 is based on the fact that the DRSPsô, which are degenerate ruled surface patches are equivalent to topological discs (i.e. a DRSP can be deformed into a disc, applying the valid topological operations such as bending, shearing etc. but no tearing, and punching holes). Boolean union results in objects in the same dimension or less. Hence the proposition. Based on Proposition 1 and findings in R^2 [24], the following conjecture is proposed (the term ôhyperô is used to denote higher dimensions).

Conjecture 1 For a set of hyper surfaces in R^n, the concave hull will consist of lower dimensional entities (straight lines), equivalent to topological hyper discs in R^{n-1} (such as DRSPôs in R^3 equivalent to topological disc (which is in $R2$) and hyper surfaces in R^n

Conjecture 1 is again under the assumption that the topological hyper discs do not intersect the input set.

2.3 One surface is completely inside the other

For this case, it is obvious that the surface which is completely outside is the concave hull between the two surfaces. To determine whether one surface is completely inside the other, first it is required to detect that there is no intersection. A ray tracing approach can then be used to determine which surface lies inside.

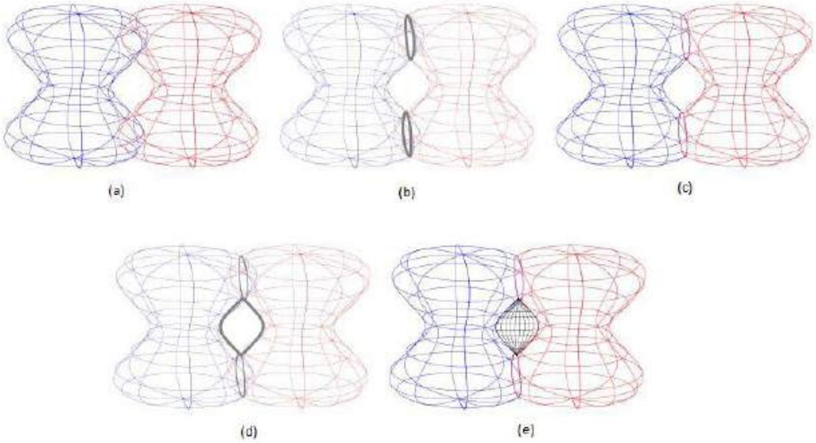
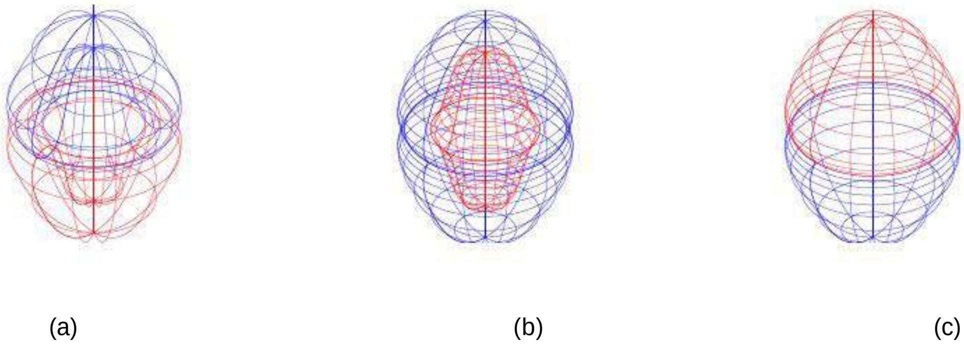


Fig. 8: Concave hull for output with genus >0: (a) Pair of dumbbell-like simply connected surfaces, (b) Intersection curves of two dumbbell shapes, (c) Boolean union of the shapes, (d) Geodesic curve connecting the intersection curves, (e) Zero volume degenerate ruled surface.



(a) (b) (c)

Fig. 9: Concave hull when intersection results in multiply-connected surfaces: (a) Two simply connected surfaces intersecting, (b) Boolean union resulting in multiply connected surfaces, (c) output concave hull.

3 CONCAVE HULL FOR A SET OF SURFACES

In this section, we extend the approach given in Section 2 to obtain the concave hull for a set of surfaces, assuming that there is only one MAL between two surfaces. We also assume that degenerate ruled surface patch (DRSP) in multiply-connected output of a Boolean union does not intersect with other surfaces. Initially, all surfaces that are completely inside another are eliminated, as they do not contribute to concave hull. This process will result in a set of simply-connected surfaces that either intersect or lie completely outside.

Intersection curves are identified for the intersecting surfaces. We apply the principle stated earlier to find the concave hull for a pair of surfaces (Section 2.2), resulting in concave hull for the pair. Note that the DRSP is fit when the output is of genus > 0. We keep repeating this step till all the intersecting surfaces are processed. This leaves us with a set of surfaces (some obtained using Boolean). For example, for the set of surfaces in Figure 10(a), the resultant is as shown in Figure 10(b), which will be set of surfaces lying outside one another.

For a surface (say $S1$), compute the MAP between $S1$ and all other surfaces (using Equation (1)). Repeating this process for all the surfaces that are outside results in a complete graph of all the MAL‟s (Figure 10(c)). Using the minimum spanning tree (MST) [4] computed with nodes as set of surfaces that are outside and the length as the distance between MAP for the set of surfaces lying outside each other, concave hull is computed. We assume that each distance is different, so that MST returns a unique result and thereby giving a unique concave hull (Figure 10(d)). It should be noted that, a computed MAL between a pair of surfaces is not included while computing minimum distance between other curves as well as the DRSP. Algorithm (1) explains the concave hull finding procedure. It should be noted that, a computed MAL between a pair of surfaces is not included while computing minimum distance between other curves as well as the DRSP.

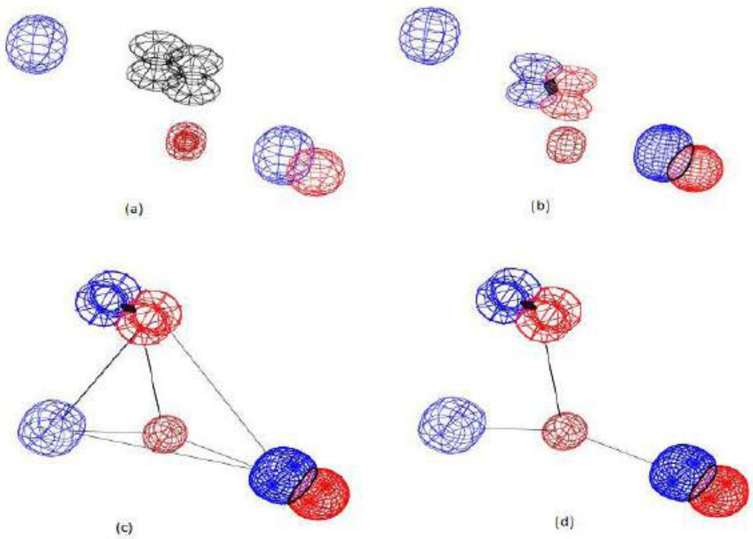
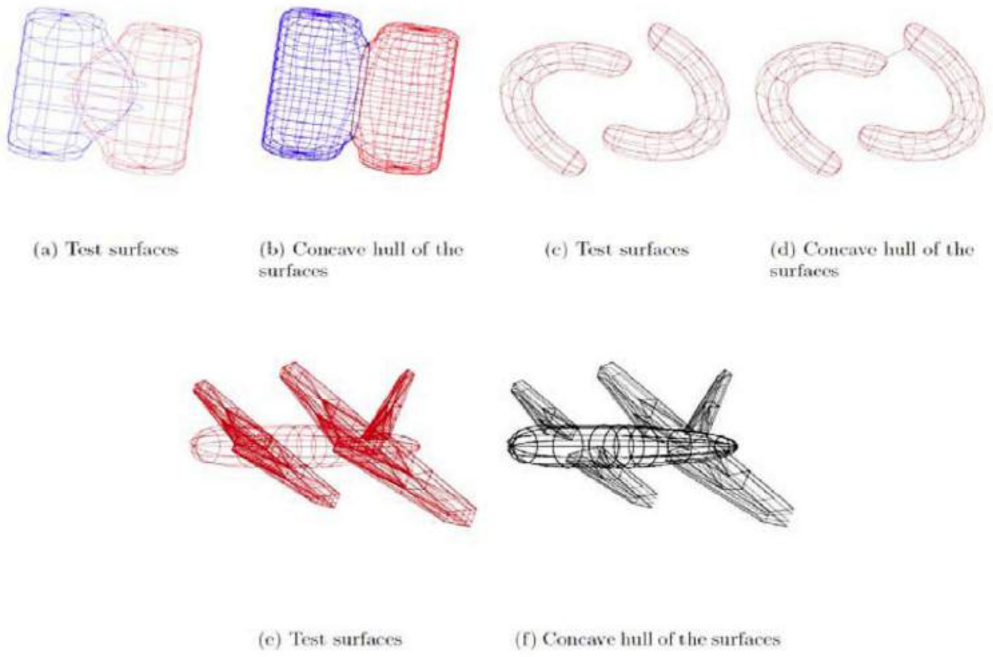


Fig. 10: Illustration of the algorithm: (a) Initial set of surfaces, (b) after processing surfaces that intersect, (c) MALs for all set of surfaces, (d) concave hull for the set.

Algorithm 1 $CONCAVEHULL(S = S_1, ..., S_n)$

```
Determine surfaces that intersect each other.
Eliminate surfaces that lie completely inside.
Find sets of surfaces that intersect. Let $T_i$ denote each set, and each $T_i$ will consist of surfaces from $S$.
for each set $T_i$ of intersecting surfaces do
  Perform Boolean union.
  Represent each Boolean union as a single surfaces (say $BU_i$).
  if $BU_i$ is simply connected with genus $>0$ then
    Fit a degenerate ruled surface patch.
  end if
  if Multiply connected then
    Eliminate interior regions.
  end if
end for
Let $S' = \{\{BU_i\} \bigcup \{S - \{T_i\}\}\}$.
for each surface in $S'$ do
  Compute MAP to all other surfaces in the set.
end for
if All surfaces in $S'$ are outside each other then
  Use surfaces as nodes and MAPs' as distances, find the minimum spanning tree (MST).
  Return MST as concave hull.
else
  Return $S'$ as concave hull.
end if
```

Figure 11 shows more example results of the algorithm. All the implementation have been carried out using IRIT [11] geometric kernel and its constraint solver [12]. Figure 11(b) shows the result for the test object in Figure 11(a). In Figure 11(c), MAP computation lead to the computation of the concave hull in Figure 11(d).



(a) Test surfaces

(b) Concave hull of the surfaces

(c) Test surfaces

(d) Concave hull of the surfaces

(e) Test surfaces

(f) Concave hull of the surfaces

Fig. 11: Concave hull for a set of freeform planar surfaces in R^3.

4 DISCUSSION

4.1 Algorithm Analysis

Assuming intersection and union are constant time operations, the major step in the algorithm is computing MST, which can be shown to be O(E log E), where E is the number of edges [4].

4.2 Limitations

Algorithm described in Section 3 returns an unique concave hull only under the assumptions that MAP between two surfaces is unique and no distances computed using MAP are same across different surfaces as MST will then not be unique. We have also assumed that the intermediate surface DRSP does not intersect with other surfaces, which may not be true in general. Moreover, DRSP is currently identified only using two curves, which may not hold in general as in the set of surfaces (which for genus > 0 after Boolean union) shown in Figure 12.

4.3 Future Work

The algorithm can be extended for C^0-continuous surfaces and possibly for open surfaces. The application areas are being explored at present. Various properties that the computed concave hull satisfies are also being explored. The restriction that a surface should not contain planar portions that will result in infinite solution while computing MAP is also being looked at currently.

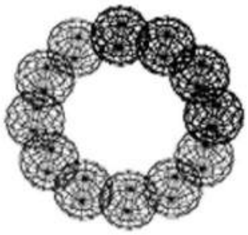


Fig. 12: Trimming patch to be employed to construct DRSP.

5 CONCLUSION

In this paper, an algorithm for computing concave hull of freeform surfaces, with an appropriate definition has been presented. It is shown that the concave hull can be computed using Boolean, antipodal calculations and degenerate ruled surface patches. Under certain conditions, the concave hull consists of antipodal lines, topological disks and the input surfaces in R^3 and a conjecture is proposed for R^n, based on the observations in R^3. Few results have been presented and discussed.

REFERENCES

[1] http://ubicomp.algoritmi.uminho.pt/local/concavehull.html.

[2] http://latingeo.net/datos latingeo/noticias doc/108Polyhulls-Galton.pdf..

[3] http://www.concavehull.com/.

[4] Aho, A. V.; Hopcroft, J. E.; Ullman, J. D.: Data Structures and Algorithms, Addison- Wesley, 1983.

[5] Chaudhuri, A. R.; Chaudhuri, B. B.; Parui, S. K.: A novel approach to computation of the shape of a dot pattern and extraction of its perceptual border, Comput. Vis. Image Underst. 68, 1997, 57ð275, http://dx.doi.org/10.1006/cviu.1997.0550

[6] Corney, J.; Rea, H.; Clark, D.; Pritchard, J.; Breaks, M.; MacLeod, R.: Coarse filters for shape matching, IEEE Computer Graphics and Applications, 22, 2002, 65ð74, 10.1109/MCG.2002.999789

[7] De Berg, M.; Cheong, O.; van Kreveld, M.; Overmars, M.: Computational geometry: algorithms and applications, Springer, 2008.

[8] Duckham, M.; Kulik, L.; Worboys, M.; Galton, A.: Efficient generation of simple polygons for characterizing the shape of a set of points in the plane, Pattern Recognition, 41(10), 2008, 3224 ð 3236, http://dx.doi.org/10.1016/j.patcog.2008.03.023

[9] Edelsbrunner, H.; Kirkpatrick, D.; Seidel, R.: On the shape of a set of points in the plane, IEEE Transactions on Information Theory, 29(4), 1983, 551 ð 559.

[10] Edelsbrunner, H.; Mucke, E. P.: Three-dimensional alpha shapes, ACM Trans. Graph., 13(1), 1994, 43ð72, 10.1145/174462.156635

[11] Elber, G.: IRIT 10.0 User's Manual, The Technionñ Israel Institute of Technology, Haifa, Israel, 2009.

[12] Elber, G.; Kim, M.-S.: Geometric constraint solver using multivariate rational spline functions, In SMA '01: Proceedings of the sixth ACM symposium on Solid modeling and applications, New York, NY, USA, 2001, 1ð10, 10.1145/376957.376958

[13] Elber, G.; Kim, M.-S.; Heo, H.-S.: The convex hull of rational plane curves, Graph Models, 63(3), 2001, 151ð162, 10.1006/gmod.2001.0546

[14] Galton, A.; Duckham, M.: What is the region occupied by a set of points?, In GIScience, 2006, 81ð 98. 10.1007/11863939_6

[15] Kim, Y.-J.; Lee, J.; Kim, M.-S.; Elber, G.: Efficient convex hull computation for planar freeform curves, Computers & Graphics, 35(3), 2011, 698 ð 705. Shape Modeling International (SMI) Conference 2011. http://dx.doi.org/10.1016/j.cag.2011.03.028

[16] Lee, Y.-S.; Chang, T.-C.: 2-phase approach to global tool interference avoidance in 5-axis machining, Computer-Aided Design, 27(10), 1995, 715 ð 729. http://dx.doi.org/10.1016/0010-4485(94)00021-5

[17] Li, B.; Liu, J.: Detail feature recognition and decomposition in solid model, Computer-Aided Design, 34(5), 2002, 405 ð 414. http://dx.doi.org/10.1016/S0010-4485(01)00118-X

[18] Melkemi, M.; Djebali, M.: Computing the shape of a planar points set, Pattern Recognition, 33(9), 2000, 1423 ð 1436. http://dx.doi.org/10.1016/S0031-3203(99)00124-7

[19] Moreira, A. J. C.; Santos, M. Y.: Concave hull: A k-nearest neighbors approach for the computation of the region occupied by a set of point, In GRAPP (GM/R), 2007, 61ð68.

[20] Muthuganapathy, R.; Elber, G.; Barequet, G.; Kim, M.-S.: Computing the minimum enclosing sphere of free-form hypersurfaces in arbitrary dimensions, Computer-Aided Design, 43, 2011, 247ð257. http://dx.doi.org/10.1016/j.cad.2010.12.007

[21] O'Rourke, J.: Computational Geometry in C, Cambridge University Press, New York, NY, USA, 1998.

[22] Piegl, L.; Tiller, W.: The NURBS book (2nd ed.). Springer-Verlag, New York, NY, USA, 1997.

[23] Seong, J.-K.; Elber, G.; Johnstone, J. K.; Kim, M.-S.: The convex hull of freeform surfaces, Computing, 72(1-2), 2004, 171ð183. 10.1007/s00607-003-0055-x

[24] Vishwanath, A. V.; Srivatsan, R. A.; Ramanathan, M.: Concave and alpha hulls of a set of freeform planar closed curves, In as abstract and presented in SIAM Geometric Design/Solid and physical modeling (GD/SPM11), Orlando, Florida, USA, 2011.

[25] Yue, Y.; Murray, J.; Corney, J.; Clarkr, D.: Convex hull of a planar set of straight and circular line segments, Engineering Computations, 16(8), 1999, 858 ð 875. 10.1108/02644409910304086

[26] Junyi, X.; Yanping, F.; Zuoya, Z.; Xihong, Q.: A Concave Hull algorithm for scattered data and its applications, Image and Signal Processing (CISP), 3rd International Congress on, 5, 2010, 2430-2433, doi: 10.1109/CISP.2010.5648277