

Footpoint distance as a measure of distance computation between curves and surfaces

Anonymous submission

Abstract

In automotive domain, CAD models and its assemblies are validated for conformance to certain design requirements. Most of these design requirements can be modeled as geometric queries, such as distance to edge, planarity, gap, interference and parallelism. Traditionally these queries are made in discrete domain such as a faceted model, inducing approximation. Thus, there is a need for modeling and solving these queries in the continuous domain without discretizing the original geometry. In particular, this work presents an approach for distance queries of curves and surfaces, typically represented using NURBS.

Typical distance problems that have been solved for curves/surfaces are the minimum distance and Hausdorff distance. However, the focus in the current work is on computing corresponding portions (patches) between surfaces (or between a curve and a set of surfaces) that satisfy a distance query. Initially, it was shown that the footpoint of the bisector function between two curves can be used as a distance measure between them, establishing points of correspondence. Curve portions that are in correspondence are identified using the antipodal points. It is also identified that the minimum distance in a corresponding pair is bound by the respective antipodal points. Using the established footpoint distance function, the distance between two surfaces was approached. For a query distance, sets of points satisfying the distance measure are identified. The boundary of the surface patch that satisfies the distance is computed using the α -shape in the parametric space of the surface. Islands contributing to the distance query are also then computed. A similar approach is then employed for the distance between a curve and a set of surfaces. Initially, the minimum footpoint distance function for a curve to a surface is computed and repeated for all other surfaces. A lower envelope then gives the portions of the curves where the distance is more than the query.

Keywords: distance computation, freeform curves, freeform surfaces, lower envelope

1. Introduction

Freeform objects such as curves/surface are predominantly used in automotive domain. The CAD models of body parts are validated for conformance to certain design requirements. Geometric queries, such as distance to edge, planarity, gap, interference and parallelism are typically used for modeling design requirements. These properties are usually computed on the approximated models, using discrete approaches such as faceting (or meshes) rather than on the freeform representation itself. There are certain disadvantages associated with this approach, such as

- Approximation: Since faceted models are only approximate representation of original geometry, the queries made on them are also approximate.
- Computational complexity: Accuracy of results

depends on the quality of faceting. There is a tradeoff as computational expense increases with densely faceted models.

- Result remapping: Mapping of results from faceted representation to original geometry adds to the approximation.

Though commercial CAD packages offer some capability to make geometric queries from original geometry (exact computation), they are elementary in nature and difficult to scale. For example, given two geometric entities one can query corresponding points where the distance is minimum, but one cannot query for corresponding regions where distance is less than a threshold value. For spot welding operation one may identify regions where faces of welded parts are within a given proximity of each other. Such regions can be defined as manufacturing workspace within which valid joint enti-

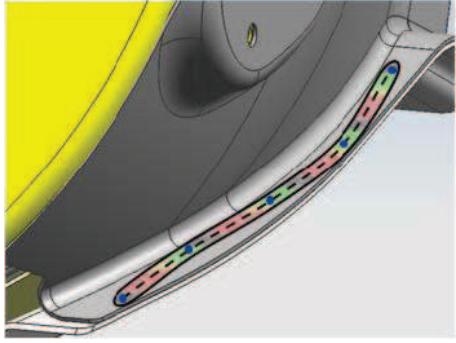


Figure 1: Manufacturing workspace for creating valid joint entities.

ties, such as weld spots, laser weld curves and adhesive bonding strips can be created (Figure 1). Thus, there is a need to develop an efficient method for modeling and solving these queries accurately. Rather than making the queries through discretized models, in the current work, the objective is to model distance queries without approximating the inputs by a faceted model.

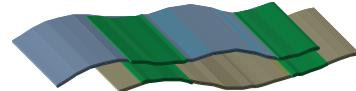
1.1. Problem statement

In this work, the following are the problems that are looked at;

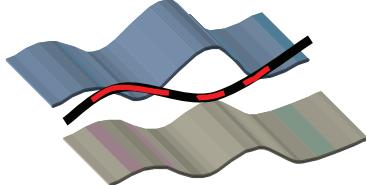
- Given two freeform surfaces, compute regions on each surface, such that, for any point (P) in a region on one surface there lies a corresponding point (P') on the other surface at a distance less than a threshold value (Figure 2(a)).
- Given a freeform curve and a set of freeform surfaces, compute segments of the curve where the minimum distance between the curve and any of the surfaces is more than a threshold value (Figure 2(b)).

1.2. Related work

Precise computation of freeform curves and surfaces without discretizing them is now becoming prominent, as the accuracy of such computations is provenly better. One of the prominent approaches is to formulate the problem in terms of parametric representation and then solved in parametric space [1], employing a rational solver [2]. This approach has been demonstrated for various algorithms such as bisectors [3], Voronoi cell [4], medial axis [5], minimum enclosing sphere [6] etc. Another approach for global analysis of freeform geometry is dimensional lifting scheme, where the problem is solved in higher dimensions and has been demonstrated



(a) Two surfaces, where the corresponding patches (in green) satisfy a distance threshold.



(b) A curve and a set of surfaces, where the red segments of the curve satisfy a distance threshold with respect to all the surfaces.

Figure 2: Distance for surface-surface or curve and a set of surfaces.

for visibility problems [7], minimum distance query [8], and critical point analysis [9].

In the precise computation of distance function of curves and surfaces, most of the work has focussed on computing the minimum distance between a point and a curve/surface. For a pairs of curves, only [10] deal with distance between two curves, albeit Hausdorff distance, which does not appear to be useful for the problems at our hand.

1.3. Contributions

To the best of our knowledge, no work seems to exist that compute corresponding patches of curves/surfaces satisfying above or below a certain distance value, which is the focus of this work. Perhaps this is the first time that such a distance query has been addressed. Moreover, the curves and surfaces are not discretized into mesh. The following are the major contributions:

- Footpoint distance measure has been proposed as a measure for distance computation.
- Points of correspondence through footpoints were explored in the case of curve-curve case and found to be an useful tool.
- Corresponding surface patches for the surface-surface case are identified using footpoint distance. α -shape has been used to detect boundaries including island regions.
- A lower-envelope based approach has been proposed and demonstrated for the distance query between a curve and a set of surfaces.

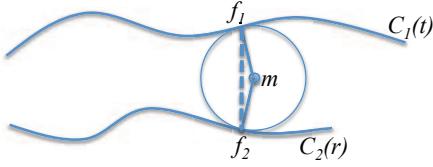


Figure 3: Illustration of footpoint distance function. f_1 and f_2 are the footpoints corresponding to the bisector point m . Distance between the footpoints is called footpoint distance.

2. Establishing footpoint distance as a measure

To solve the stated problems in Section 1.1, we started solving ‘corresponding curve portions between two given curves’. In this section, computing distance between two curves is explained, along with how to employ them to get the distance bounds for the curve portions. This then served as motivation to solve the problems delineated in Section 1.1.

Definition 1. An Entity is a freeform curve or surface.

Definition 2. Bisector point is a point that is equidistant from two different points on the two entities respectively. Set of bisector points is then called as bisector (which could be a curve or a surface).

Definition 3. Footpoint is a point on an entity corresponding to its bisector point. Footpoint distance (FD) is the Euclidean distance between the footpoints corresponding to a bisector point.

For example, in Figure 3, footpoints f_1 and f_2 correspond to a bisector point m . The circle with centre m and radius $\|m - f_1\|$ is a disc passing through the footpoints f_1 and f_2 . Distance between footpoints f_1 and f_2 is termed as *footpoint distance*.

Let $C_1(t)$ and $C_2(r)$ be the two planar curves with parameters t and r respectively. Elber and Kim [11] showed that the bisector for a pair of rational curves can be implicitly represented symbolically in the parametric space. The bisector so generated can be represented in an implicit form that can be used for further processing. Moreover, the bisectors can be accurately represented up to machine precision.

The bisector provides an inherent correspondence between different points on the boundary curve as opposed to devising a separate approach for correspondence. Instead of measuring distance between two arbitrary points, in this approach, distance between the footpoints corresponding to a point on the bisector has been used. FD is the distance between footpoints f_1 and f_2 .

2.1. Distance function and establishing correspondence

Let $D(t,r)$ be the footpoint distance function. Since D is Euclidean, physically it is the length of the line joining points of the given curves. To find the extremum, following partial differential equations are used.

$$\begin{aligned} \frac{dD}{dt} &= 0, \\ \frac{dD}{dr} &= 0 \end{aligned} \quad (1)$$

In fact, it is shown in [6] that the distance is extremum only if the normals are at opposing directions to each other, termed as ‘antipodal’. The constraint equations to find antipodality based on $D(t, r)$ is as follows:

$$\langle D(t, r), C'_1(t) \rangle = 0, \quad (2)$$

$$\langle D(t, r), C'_2(r) \rangle = 0 \quad (3)$$

The above equation can be rewritten as follows:

$$\begin{aligned} \left\langle C'_1(t), C_1(t) - \frac{C_1(t) + C_2(r)}{2} \right\rangle &= 0, \\ \left\langle C'_2(r), C_2(r) - \frac{C_1(t) + C_2(r)}{2} \right\rangle &= 0. \end{aligned} \quad (4)$$

These equations are solved to find the corresponding antipodal points. Now the search is restricted to finite number of antipodal solutions as opposed to the earlier tr parametric space. Using the extrema, we can obtain the minimum by second differential. This translates to relative curvature being concave.

$$\langle k(t) - k(r), C_1(t) - C_2(r) \rangle > 0 \quad (5)$$

where k represents radius of curvature vector.

Antipodal points establish correspondence between two curves which can then be used to identify distance bounds.

2.1.1. Distance bounds

It is desired to find the exact bounds on the distance for various pairs of sections of the curves $C_1(t)$ and $C_2(r)$ that correspond. The global minima of the footpoint distances (which is minimum of the antipodal distances) is certainly a part of the bounds obtained (d_1 in Figure 4 shows the global minima of the antipodal A_1P_1). Let d_2 be the subsequent global minima (of the antipodal A_2P_2) obtained by splitting the curves using the first antipodal A_1P_1 . Hence, the distance for these segments of the two curves $C_1(t)$ and $C_2(r)$ is bound by the distances d_1 and d_2 . This process can be repeated to identify distance bounds of the two curves. Given a distance value, the portions of the curves can be identified in a straight forward manner. For example if d_1

¹⁷⁰ is 5 units and d_2 is 6 units, for a given distance query
¹⁷¹ of 5.8, then the curve segments between the antipodal
¹⁷² points A_1P_1 and A_2P_2 will play a role. This establishes that
¹⁷³ the correspondence obtained using footpoints can then
¹⁷⁴ be used to identify distance bounds and subsequently to
¹⁷⁵ find segments of curves satisfying a query distance. In
¹⁷⁶ general, correspondence based on footpoints forms the
¹⁷⁷ basis for this work.

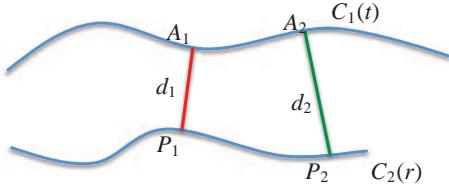


Figure 4: Figure showing global minima (d_1) of the antipodal distances. d_2 is the subsequent global minima.

¹⁷⁸ 3. Surface-surface distance function

¹⁷⁹ Armed with the finding that the footpoint distance is a
¹⁸⁰ good measure for computing distance between freeform
¹⁸¹ entities such as curves, we approach the problem of
¹⁸² ‘Given a distance query, find the corresponding surface
¹⁸³ patches’ in a similar manner.

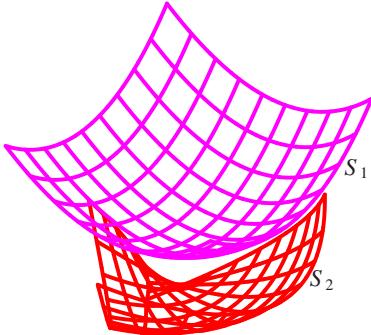


Figure 5: Input surfaces.

¹⁸⁴ Let $S_1(u_1, v_1)$ and $S_2(u_2, v_2)$ be the surfaces (Figure
¹⁸⁵ 5) and $D_1(u_1, v_1, u_2, v_2)$ be the distance function. The
¹⁸⁶ basic partial differential equations for extremum are

$$\partial D_1 / \partial u_1 = 0, \partial D_1 / \partial v_1 = 0, \partial D_1 / \partial u_2 = 0, \partial D_1 / \partial v_2 = 0.$$

¹⁸⁷ Since D_1 is Euclidian, it is the length of the line joining
¹⁸⁸ points of the given curves. Hence the conditions trans-
¹⁸⁹ late to the antipodal line constraint equations [6]. (Fig-
¹⁹⁰ ure 6(a) shows a couple of antipodal lines). Using the
¹⁹¹ antipodal equations the extrema of the distance func-
¹⁹² tion are obtained. It should be noted that the bisector

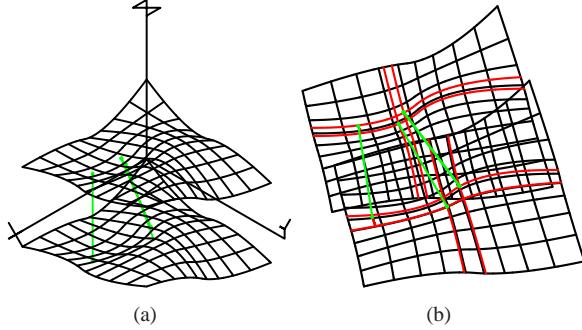


Figure 6: (a) Antipodal lines (in green) between two surfaces. (b) Only isoparametric patches can be computed using antipodal lines.

¹⁹³ between a pair of freeform curves are not rational, in
¹⁹⁴ general [12]. A symbolic representation was shown to
¹⁹⁵ be possible only for curve-curve case [11]. However,
¹⁹⁶ such a representation for the bisector of a pair of sur-
¹⁹⁷ faces [13] and subsequently for $D_1(u_1, v_1, u_2, v_2)$ has not
¹⁹⁸ been shown to be possible yet. Figure 6(a) shows the
¹⁹⁹ antipodal for two surfaces. The major bottleneck in
²⁰⁰ this approach is that it can generate only iso-parametric
²⁰¹ patches as has been shown in Figure 6(b). Hence, us-
²⁰² ing antipodal points as the start looks infeasible and this
²⁰³ motivated us to directly work on the footpoint distance,
²⁰⁴ given a query distance.

²⁰⁵ 3.1. Using footpoint distance and α -shape

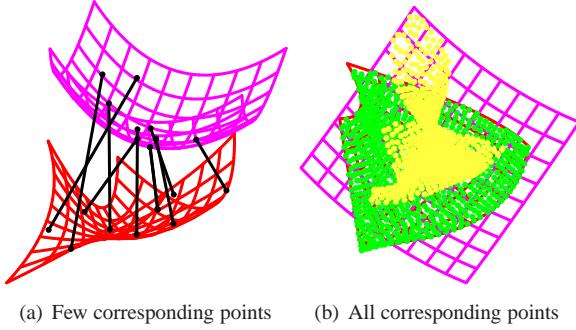
²⁰⁶ It is desired to find corresponding patches on the
²⁰⁷ surfaces $S_1(u_1, v_1)$ and $S_2(u_2, v_2)$ for various distance
²⁰⁸ query values. The bisector surface is used to establish
²⁰⁹ correspondence between points on each surface. Bisec-
²¹⁰ tor surface is computed using the following equations
²¹¹ [13];

$$S_1(u_1, v_1) + \alpha N_1(u_1, v_1) = S_2(u_2, v_2) + \beta N_2(u_2, v_2), \\ \|B(u_1, v_1, u_2, v_2) - S_1(u_1, v_1)\| = \|B(u_1, v_1, u_2, v_2) - S_2(u_2, v_2)\| \quad (6)$$

²¹² where normal of two surfaces are

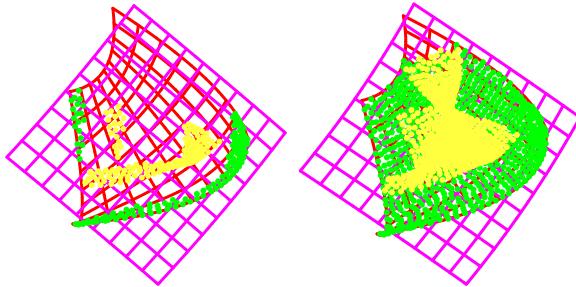
$$N_1(u_1, v_1) = \frac{\partial S_1(u_1, v_1)}{\partial u_1} X \frac{\partial S_1(u_1, v_1)}{\partial v_1}, \\ N_2(u_2, v_2) = \frac{\partial S_2(u_2, v_2)}{\partial u_2} X \frac{\partial S_2(u_2, v_2)}{\partial v_2} \quad (7)$$

²¹³ Once the equations (Equations (6)) are solved for the
²¹⁴ given surfaces we get a relation between the points on
²¹⁵ the two surfaces (footpoints) which is the correspon-
²¹⁶ dence established between the two surfaces. Figure 7(a)
²¹⁷ shows a certain number of points which correspond to
²¹⁸ each other via footpoints of the bisector surface. Fi-
²¹⁹ gure 7(b) shows all the corresponding points on both
²²⁰ surfaces.



(a) Few corresponding points (b) All corresponding points

Figure 7: Footpoint correspondence.



(a) Points on both surfaces for query (0.7 units) (b) Points on both surfaces for query (1.9 units)

Figure 8: Footpoints for different distance queries, in brackets.

221 3.2. Solving distance query

222 The next step is to solve for the distance query i.e find
223 regions on the surfaces where the footpoint distance is
224 less than the distance query. Corresponding footpoints
225 that are less than or equal to the distance query on both
226 surfaces are identified. Figure 8 shows the footpoints on
227 the surfaces for various distance queries. This approach
228 also has the advantage of computing the corresponding
229 points only once and then use it to identify points satis-
230 fying distance query. Now as a result, the patches on the
231 surfaces were obtained as point sets on the two surfaces.

232 3.3. Boundary detection using α -shape

233 To compute the surface patches corresponding to a
234 distance query, the boundary of the point-sets satisfying
235 the distance query has to be detected. Using the bound-
236 ary as a trimmed curve, patches in each of the original
237 surfaces are identified. As α -shape is one of the pre-
238 mier approaches for reconstruction, we also employ it
239 to detect the boundaries.

240 Let $P = \{p_1, p_2, \dots, p_n\}$ be a set of points. The α -hull
241 for the set P can be defined in the following manner
242 [14].

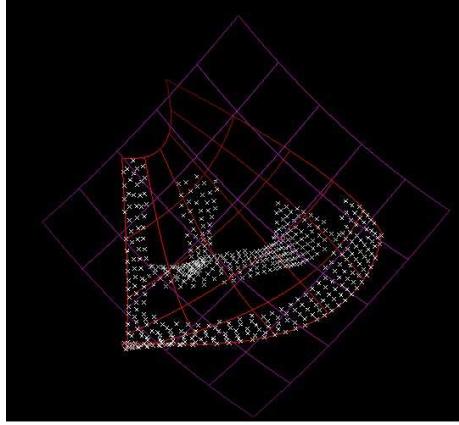


Figure 9: Patches in the form of point sets on both surfaces for $D_q = 0.8$.

243 **Definition 4.** Let α be a sufficiently small but otherwise
244 arbitrary positive real. α -hull is defined as the inter-
245 section of all closed complements of discs (where these
246 discs have radii α) that contain all the points.

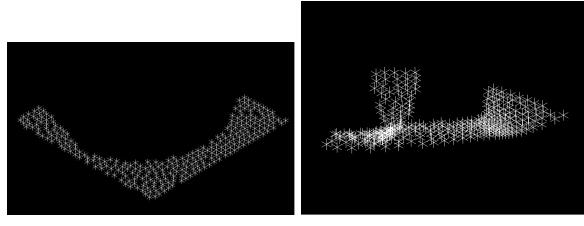
247 **Definition 5.** α -shape is obtained by replacing the disc
248 with a straight line between α -neighbours of the α -hull.

249 To detect the boundaries of the surface patches, the
250 point-sets were taken to the parametric space. One of
251 the major factors in the α -shape is to find a value of α
252 for which the boundary is reasonably constructed. After
253 much experimentation, the value for α was arrived at as
254 0.04.

255 The method is explained for a single value of the dis-
256 tance query: $D_q = 0.8$. The point sets for distance \leq
257 0.8 in each of the surfaces are shown in Figure 9. The
258 point sets in the respective parametric space are shown
259 in Figures 10(a) and 10(b). These point sets (in para-
260 metric space) are fitted with alpha shapes as shown in
261 Figure 11. Then these boundaries were used to trim the
262 surfaces as shown in Figure 12(a). Similar outputs are
263 shown for $D_q = 1.1$ and $D_q = 1.4$ in Figures 12(b) and
264 12(c) respectively.

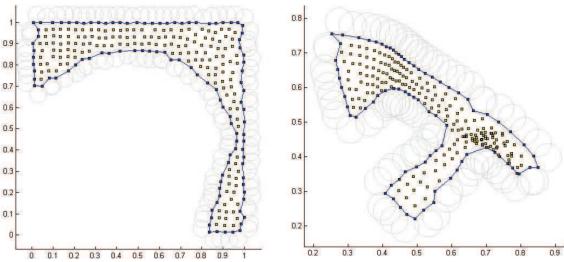
265 3.4. Boundary identification for islands

266 Numerical solving gives point sets where the points
267 are adequately close and evenly spaced within the sets.
268 However, there need not be a single patch for all the
269 points. There exists a possibility of multiple patches on
270 the same surface satisfying distance query. α -shape has
271 the ability to inherently divide point sets into islands.
272 Thus we generated the islands on the surfaces by using
273 α -shapes. Figure 13 shows the identified island regions



(a) Point set of surface 1 in para- (b) Point set of surface 2 in para-
metric space metric space

Figure 10: Points Sets in parametric space ($D_q = 0.8$)



(a) α -shape for point set of S_1 . (b) α -shape for point set of S_2 .

Figure 11: α -shapes for points in parametric space ($D_q = 0.8$).

for a point-set in the parametric space and their respective surface patches in Figure 14.

Figures 15 and 16 show the results for the distance queries 0.7 and 0.75 respectively. For a distance query of 0.7, Figures 15(a) and 15(b) show the points in the parametric space of each of the surfaces and their respective α -shapes. Figure 15(c) shows the points corresponding to the distance query on the respective surface faces and their surface patches in Figure 15(d). Similar results for a distance query of 0.75 are shown in Figure 16. It can be observed that Figure 16 contains all points pertaining to a distance query of 0.7 as well (refer Figure 15) showing the correctness of the results obtained.

Figure 17(a) shows an airplane model from which surfaces are picked for testing the algorithm. It should

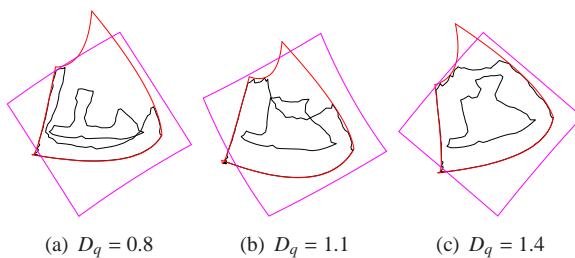


Figure 12: Surface patches for various distance queries.

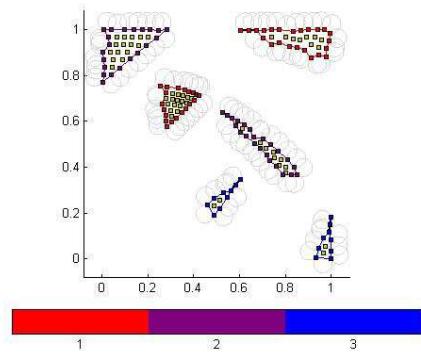


Figure 13: Island boundaries identified in parametric space

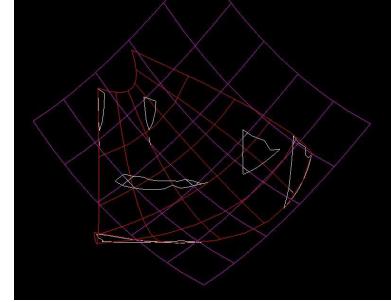
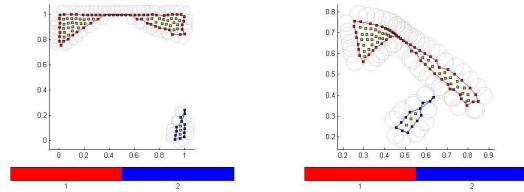


Figure 14: Regions on the surfaces for the identified boundaries



(a) α -shape in the parametric space of S_1 . (b) α -shape in the parametric space of S_2 .

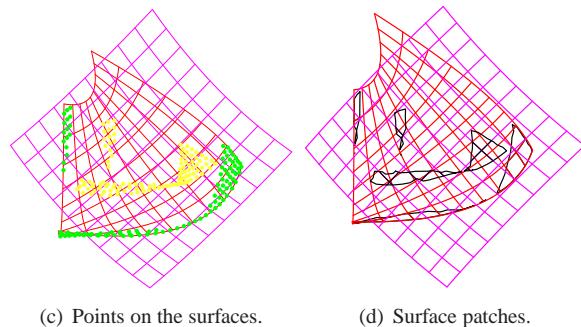


Figure 15: Results for $D_q = 0.7$

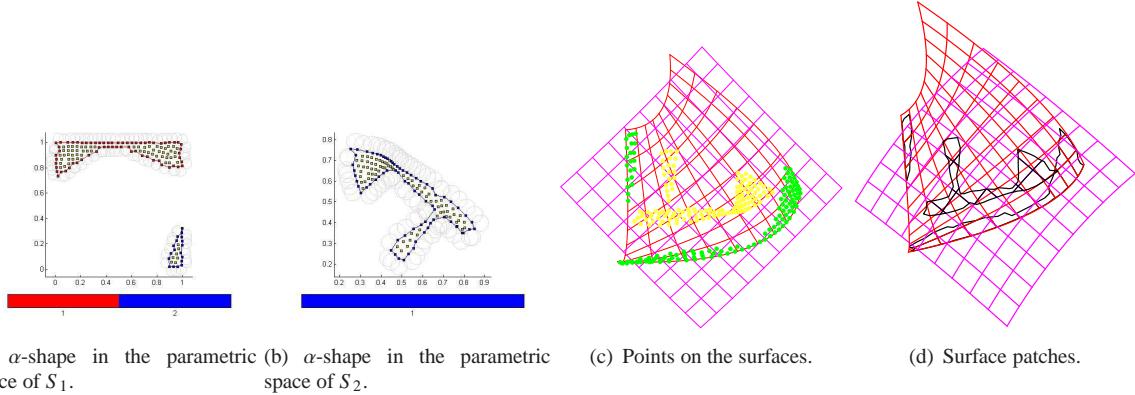


Figure 16: Results for $D_q = 0.75$

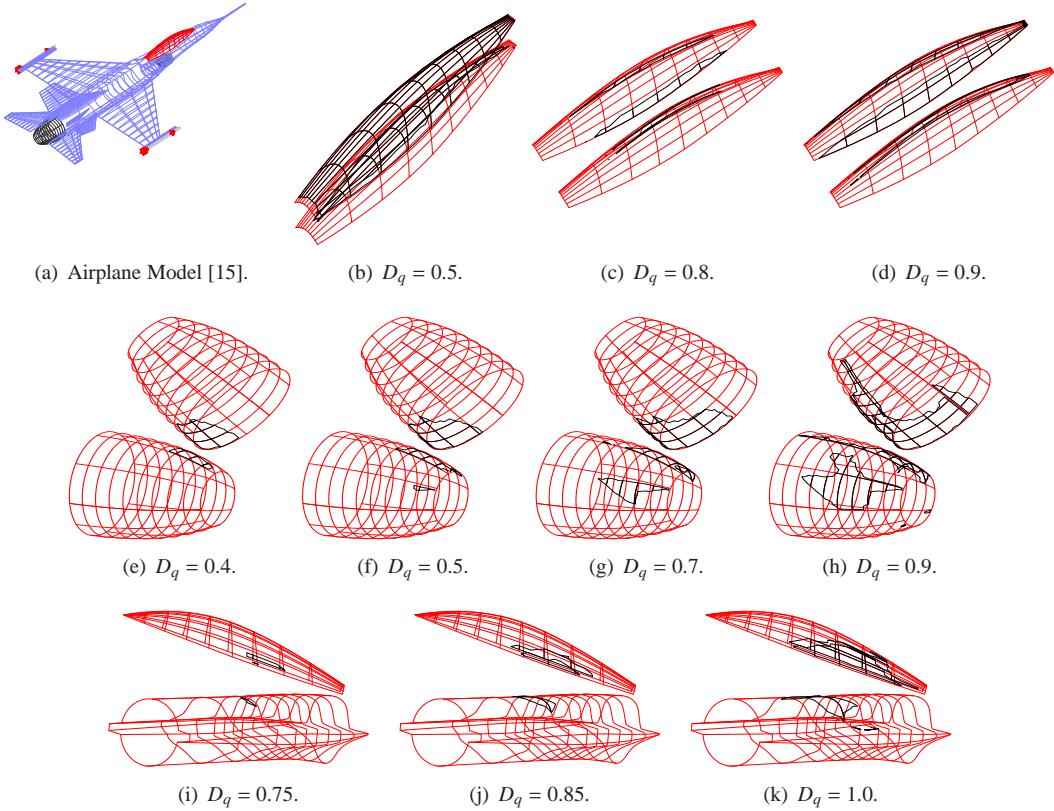


Figure 17: Patches (in black) show the corresponding ones for various distance queries. 1st row - (a) Airplane model (b),(c), and (d) patches for various distances for cockpits. 2nd row - results for fuse burner from the airplane model. Last row - results for a pair of different parts.

be noted that a model such as airplane is usually modeled as set of surfaces representing each parts and then assembling them, enabling us to pick parts. Figures 17(b), 17(c) and 17(d) use the ‘cockpit’ patch for testing (distance query value for each of them has also been

mentioned). The second and third row in Figure 17 show results for ‘fuse burner’ and for a pair of different parts from the same airplane model respectively. Another result that uses two car bumper models is shown in Figure 18.

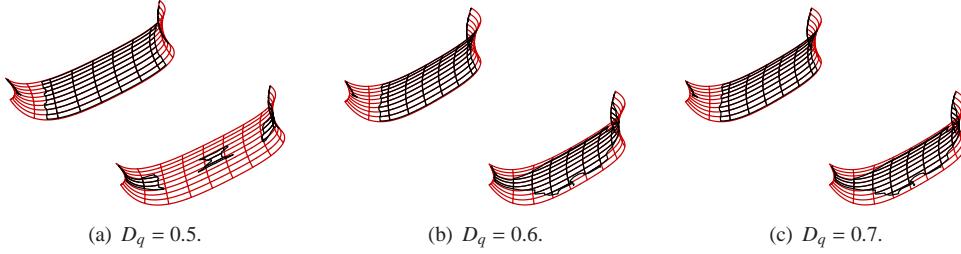


Figure 18: Results for a car bumper model.

299 4. Distance between a curve and a set of surfaces

300 In this section, an approach to the problem of ‘Given
301 a distance query, find the corresponding segments of the
302 curve where the minimum distance between the curve
303 and any of the surfaces is more than the query’ has been
304 presented.

305 4.1. Normal plane at a point on a curve

306 Let $C(t)$ be the given parametric curve in R^3 , $C'(t)$, n
307 and b represent the tangent, normal and binormal ($b =$
308 $C'(t)Xn$) at any point on the curve $C(t)$. Then,

309 **Definition 6.** *The plane determined by the tangent and
310 normal is called osculating plane. Normal plane is the
311 plane formed by the normal and binormal.*

312 It should be noted that, only a ‘single normal’ is
313 present at any point in the case of a plane curve. Un-
314 like this, at any point on a curve in R^3 , any vector lying
315 in the normal plane and passing through that point is
316 perpendicular to the tangent is a normal vector.

317 4.2. Procedure to compute the distance between a curve
318 and set of freeform surfaces

319 Figure 19 shows a curve and a few surfaces for which
320 the portions of the curve has to be identified which is
321 above a distance threshold.

322 4.2.1. Minimum footpoint distance from the given curve

323 We initially find all the bisector points B between a
324 curve $C(t)$ and a surface $S = S(u, v)$, which can be iden-
325 tified by solving the following equations [13]:

$$\langle B - C(t), C'(t) \rangle = 0, \quad (8)$$

$$\langle B - S, \frac{\partial S(u, v)}{\partial v} \rangle = 0, \quad (9)$$

$$\langle B - S, \frac{\partial S(u, v)}{\partial u} \rangle = 0, \quad (10)$$

$$\langle B - C(t), B - C(t) \rangle = \langle B - S, B - S \rangle \quad (11)$$

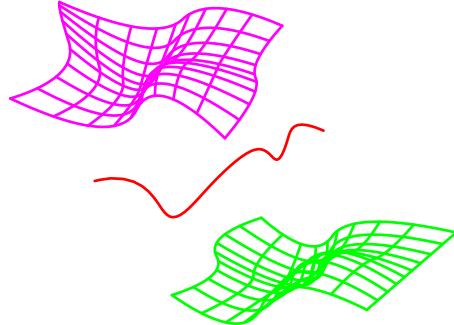


Figure 19: Input Curve and Surfaces

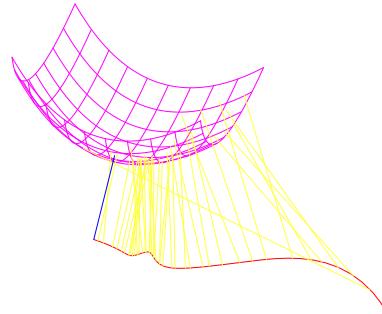


Figure 20: Lines joining minimum distance footpoints.

326 For a point on the curve, there are several footpoints
327 on the surface, because of the existence of normal plane
328 at a point on the curve in R^3 . Out of all the points, we
329 take the minimum distance footpoint ($MinF$). Figure 20
330 shows the minimum distance footpoints on the curve to
331 the surface. We now plot a graph with parameter t vs
332 $MinF$. Figure 21 shows $MinFs$ for two surfaces plotted
333 against t in the x-axis. This is then used to find segments
334 which satisfy distance query.

335 4.2.2. Splitting $MinFs$

336 It is to be noted that a $MinF$ does not have self-
337 intersection as each t will have only one value for $MinF$.
338 Next we split each minimum footpoint distance func-

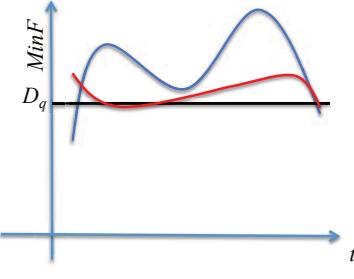


Figure 21: MinF for two surfaces (t vs $MinFs$).

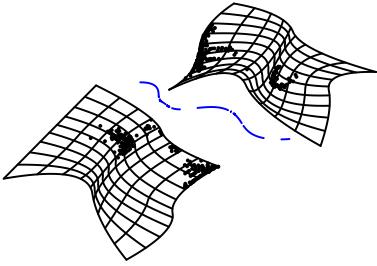


Figure 22: Curve segments for or $D_q = 0.55$.

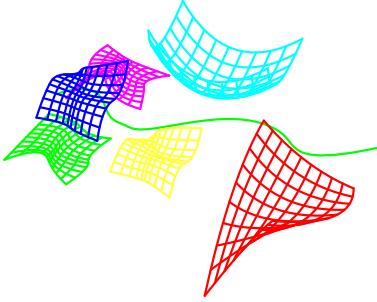


Figure 23: Input Curve and 6 Surfaces

tion according to the given distance query and get the corresponding segments where the minimum distance between the curve and the set of surfaces is more than the given value (D_q to split $MinFs$ in Figure 21).

4.2.3. Lower envelope

We repeat the same procedure described in Section 4.2.2 for all the other given surfaces with the same curve and then we take lower envelope [4] of the $MinFs$. This gives the segments of the curve where the minimum distance between curve and any of the given surfaces is more than the distance query value.

4.3. Results

Figure 22 shows the segments of curves which satisfy more than the distance query of 0.55 units. Figure

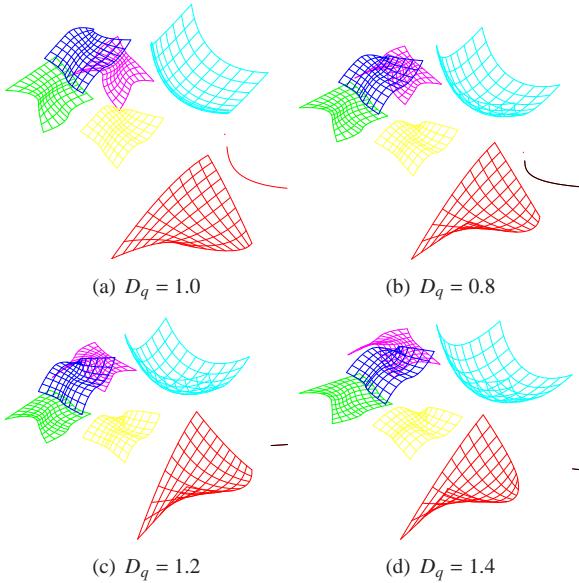


Figure 24: Curve segments for various distance queries.

23 shows a curve and a set of six surfaces. The curve segments that satisfy the distance criteria $D_q = 1.0$ units is shown in the Figure 24(a). Figures 24(b), 24(c), and 24(d) show results for distance queries 0.8, 1.2, and 1.4 respectively.

All the algorithms developed in this work has been implemented using the IRIT geometric kernel [15] and its solver [16]. In this work, few APIs have been built on top of the existing ones.

4.3.1. Discussion

One major advantage of this approach is that, the footprint correspondences are done only once, which can then be used for different distance queries. Also, the footprint distances can be captured precisely. For a surface-surface case, the approach also works for multiple-patch correspondences through island identifications rather than capturing only a pair of patches having correspondence. Segments of curves satisfying a distance query can also be identified.

As has been already mentioned, so far, the distance computation for curves and surfaces focusses mainly on computing the minimum distance. However, there appear to be not many works that address such computations precisely (see [10] which focusses on Hausdorff distance). In addition to that, there seems to be no work that computes patches of surfaces or segments of curves satisfying a distance query precisely. Perhaps this is the first time such a computation has been addressed as it could prove useful for automotive applications.

382 **5. Conclusions**

383 In this work, algorithms for computing distance be-
 384 tween curves and surfaces that satisfy a distance input
 385 value has been proposed and implemented. Footpoint
 386 distance has been shown to be an appropriate distance
 387 measure for the intended problems. It has also been
 388 demonstrated that it can be achieved without discretiz-
 389 ing the surface into an approximate mesh. Results indi-
 390 cate that the methodologies proposed are very amenable
 391 for implementation. IRIT geometric kernel has been
 392 used for computation purposes.

393 *5.1. Future work*

- 394 • A higher dimensional symbolic or implicit repre-
 395 sentation can be formulated for curve-surface and
 396 surface-surface bisector.
- 397 • As α -shape requires a value for α , it is usually
 398 identified through experimentation. An automatic
 399 mechanism based on the knowledge from the dis-
 400 tance query could be developed.

- 436 [9] J.-K. Seong, D. E. Johnson, G. Elber, E. Cohen, Critical
 437 point analysis using domain lifting for fast geom-
 438 etry queries, Comput. Aided Des. 42 (7) (2010) 613–624.
 439 doi:10.1016/j.cad.2010.03.004.
- 440 [10] Y.-J. Kim, Y.-T. Oh, S.-H. Yoon, M.-S. Kim, G. Elber, Precise
 441 hausdorff distance computation for planar freeform curves using
 442 biarcs and depth buffer, Vis. Comput. 26 (6-8) (2010) 1007–
 443 1016. doi:10.1007/s00371-010-0477-3.
- 444 [11] G. Elber, M.-S. Kim, Bisector curves for planar rational curves,
 445 Computer-Aided Design 30 (14) (1998) 1089–1096.
- 446 [12] R. Farouki, J. Johnstone, The bisector of a point and a plane
 447 parametric curve, Computer Aided Geometric Design 11 (2)
 448 (1994) 117–151.
- 449 [13] G. Elber, M.-S. Kim, A computational model for nonrational bi-
 450 sector surfaces : curve-surface and surface-surface bisectors, in:
 451 TR-CIS9908, Center for Intelligent Systems technical reports,
 452 Technion-Israel Institute of Technology, Department of Com-
 453 puter Science, 1999.
- 454 [14] H. Edelsbrunner, D. Kirkpatrick, R. Seidel, On the shape of a set
 455 of points in the plane, Information Theory, IEEE Transactions
 456 on 29 (4) (1983) 551 – 559. doi:10.1109/TIT.1983.1056714.
- 457 [15] G. Elber, IRIT 10.0 User's Manual, The Technion—Israel Insti-
 458 tute of Technology, Haifa, Israel (2009).
- 459 [16] G. Elber, M.-S. Kim, Geometric constraint solver using multi-
 460 variate rational spline functions, in: SMA '01: Proceed-
 461 ings of the sixth ACM symposium on Solid modeling and
 462 applications, ACM, New York, NY, USA, 2001, pp. 1–10.
 463 doi:htp://doi.acm.org/10.1145/376957.376958.

401 **References**

- 402 [1] M.-S. Kim, G. Elber, J.-K. Seong, Geometric computations in
 403 parameter space, in: Proceedings of the 21st spring conference
 404 on Computer graphics, SCGG '05, ACM, New York, NY, USA,
 405 2005, pp. 27–32. doi:10.1145/1090122.1090127.
- 406 [2] G. Elber, T. Grandine, An efficient solution to systems of mul-
 407 tivariate polynomial using expression trees, IEEE Transactions
 408 on Visualization and Computer Graphics 15 (2009) 596–604.
 409 doi:htp://doi.ieeecomputersociety.org/10.1109/TVCG.2009.42.
- 410 [3] G. Elber, M.-S. Kim, Bisector curves of planar rational
 411 curves, Computer-Aided Design 30 (14) (1998) 1089 – 1096.
 412 doi:10.1016/S0010-4485(98)00065-7.
- 413 [4] I. Hanniel, R. Muthuganapathy, G. Elber, M.-S. Kim,
 414 Precise Voronoi cell extraction of free-form rational plan-
 415 nar closed curves, in: SPM '05: Proceedings of the
 416 2005 ACM symposium on Solid and physical model-
 417 ing, ACM, New York, NY, USA, 2005, pp. 51–59.
 418 doi:htp://doi.acm.org/10.1145/1060244.1060251.
- 419 [5] M. Ramanathan, B. Gurumoorthy, Interior medial axis trans-
 420 form computation of 3d objects bound by free-form sur-
 421 faces, Computer-Aided Design 42 (12) (2010) 1217 – 1231.
 422 doi:10.1016/j.cad.2010.08.006.
- 423 [6] R. Muthuganapathy, G. Elber, G. Barequet, M.-S. Kim, Com-
 424 puting the minimum enclosing sphere of free-form hypersur-
 425 faces in arbitrary dimensions, Comput. Aided Des. 43 (2011)
 426 247–257. doi:htp://dx.doi.org/10.1016/j.cad.2010.12.007.
- 427 [7] G. Elber, Global curve analysis via a dimensionality lifting
 428 scheme, in: Proceedings of the 11th IMA international con-
 429 ference on Mathematics of Surfaces, IMA'05, Springer-Verlag,
 430 Berlin, Heidelberg, 2005, pp. 184–200.
- 431 [8] J.-K. Seong, D. E. Johnson, E. Cohen, A higher dimensional
 432 formulation for robust and interactive distance queries, in: Pro-
 433 ceedings of the 2006 ACM symposium on Solid and physical
 434 modeling, SPM '06, ACM, New York, NY, USA, 2006, pp. 197–
 435 205. doi:10.1145/1128888.1128916.