Robustly Computing Intersection Curves of Two Canal Surfaces with Quadric Decomposition

Jinyuan Jia[1], Ajay Joneja[2], and Kai Tang[2]

[1] Zhuhai College of Jilin University,
Jinwan District, Zhuhai, P. R. China, 519041
csjyjia@yahoo.com.cn
[2] The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong, P. R. China
Joneja@ust.hk, mektang@ust.hk

Abstract. This paper revisits the intersection problems of two canal surfaces with a new quadric decomposition we proposed for canal surfaces. It reduces computing intersection curves of two canal surfaces to computing intersection curves of two revolute quadrics. Furthermore, Bounding Cylinder Clipping is proposed for efficient intersection determination. Compared to the existing method, our method can (*i*) run more robustly and efficiently; (*ii*) represent the final intersection curves as a piecewise closed-form RQIC; and (*iii*) give a simple shape analysis.

1 Introduction

Surface intersection is a fundamental issue in CAGD and geometric modeling. Robustness, accuracy and efficiency are used to evaluate surface intersection algorithms. Several important algorithms that were developed over few decades have been summarized in [16]. To design accurate, robust and efficient algorithms of computing intersection curves of two surfaces, even two special surfaces, e.g. quadrics, cyclides and canal surfaces still remains an open challenge. Although methods for general surface intersections can be applied to special surface intersection problems, they are inefficient. Considering that special surfaces usually have good geometric properties, it desirable to develop more efficient intersection algorithms for them. Therefore many papers have addressed specific intersection problems for CSG primitives, e.g. plane, sphere, cylinder, cone, quadric and tori [1, 2, 3, 5, 13, 14, 17, 18, 19] and some potential geometric primitives e.g. cyclides, surfaces of revolution, ruled surfaces and ringed surfaces [4, 6, 7, 8, 9, 10, 11, 12, 17].

Canal surfaces are one of important geometric primitives in solid modeling, VR, CG, CAD and CAM. Examples of canal surfaces include natural quadrics, revolute quadrics, tori, Dupin cyclides, surfaces of revolution and pipe surfaces. Canal surfaces are very useful in representing long thin objects, for instance, pipes, poles, 3D fonts, brass instruments, internal body organs, and a variety of filleted surfaces. Therefore, it is essential to devise robust and efficient intersection algorithms for canal surfaces.

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part II, LNCS 3992, pp. 342 – 349, 2006.

2 Related Works

Subdivision is an important approach to solve surface intersection problems. Heo *et al* use *circle decomposition* to solve intersection problem of two canal surfaces in [6], which subdivides a canal surface into a dense set of characteristic circles, and reduces intersection problem of two canal surfaces to a zero-set searching problem of a bivariant function $f(u,v) = 0$, that is much simpler than the original intersection problem. However, (*i*) the numerical behavior of zero-set searching of $f(u,v) = 0$ is both time and memory consuming at high precision; (*ii*) it outputs the intersection curves with a set of discrete sampling points, that is not easy to concatenate; and (*iii*) no shape analysis is performed on the intersection curves, e.g. loops and singularity.

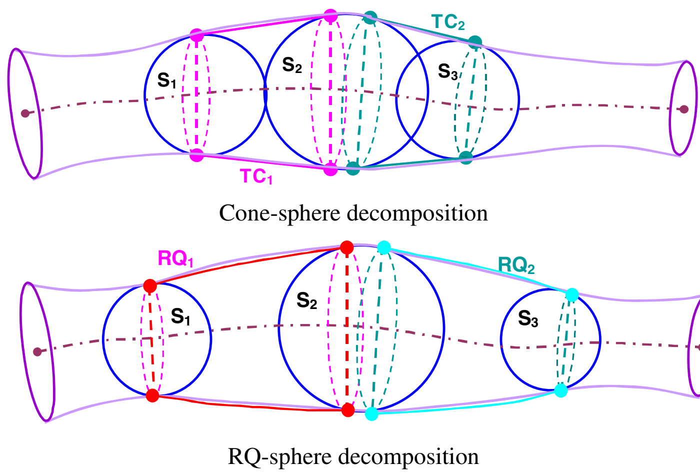


Fig. 1. Two quadric decomposition schemes of same canal surface: (a) Cone-sphere decomposition; (b) RQ-sphere decomposition

We revisit this problem by proposing more suitable subdivision scheme for canal surfaces. Except for circle decomposition, there are three other subdivision schemes for canal surfaces, *cyclide* [20], *cone-sphere* [15] and *RQ-sphere* [11]. The first one approximates a canal surface with a set of G^1 truncated cyclides [20]. The second one approximates a canal surface with a series of sampling spheres and associated tangential truncated cones (see Fig. 1(a)). However, the density of cone-spheres increases for good approximation quality at high precision and easily causes self-intersection of two neighboring truncated cones when the spine curve has high curvature. The last one approximates canal surfaces with a series of sampling spheres and associated tangential revolute quadrics (see Fig. 1(b)), instead of truncated cones. Apparently, not only less RQ-sphere pairs are required than cone-sphere scheme for same approximation quality, but the self-intersection problem can be avoided as well. In fact, cone-sphere is a special case of RQ-sphere, *cyclide* decomposition reduces canal/canal intersection computing to cyclide/cyclide intersection, that has to solve order 8 polynomial equation numerically. Both *cone-sphere* and *RQ-sphere* can reduce computing canal/canal intersection curves to computing RQ/RQ intersection curves, that has closed-form solutions [1, 2, 5, 16, 17, 19, 20]. However, cone-sphere may yield incorrect intersection curve if it has self-intersection. So we decided to employ RQ-sphere decomposition to solve canal/canal intersection problem. In this paper, we propose a new conception, *canal valid intersection intervals* (CVII), a good hierarchical data structure, *cylindrical bounding volume* (BCT), and a new method, *bounding cylinder clipping* for efficient intersection determination in Section 3. The rough idea of canal/canal intersection algorithm is described in Section 4. Experimental examples given in Section 5 show the robustness and efficiency of our method. Conclusions and future work are presented in Section 6.

3 Bounding Cylinder for Canal Surfaces

Compared to traditional bounding volumes, bounding box and bounding sphere, *bounding cylinder* BC is chosen for canal/canal intersections because (*i*) it can enclose canal surfaces more closely; (*ii*) it can be constructed rather easily; (*iii*) more important, intersections of two BCs can be computed geometrically [4].

3.1 Hierarchical Construction of Bounding Cylinder Tree

It is uneasy to construct a smallest BC for canal surfaces theoretically. In practice, it is acceptable to construct a near BC for canal surfaces, as long as it can reasonably close to the smallest BC.

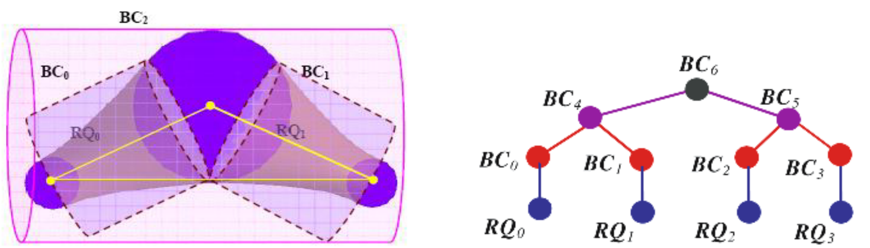


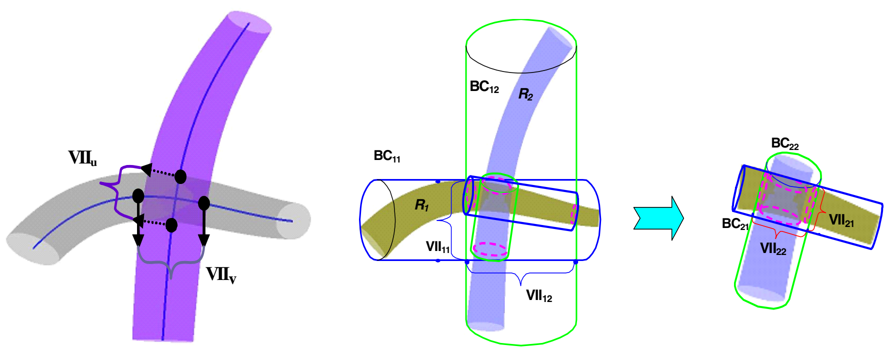
Fig. 2. A binary bounding cylinder tree BCT for a canal surface

Considering that canal surfaces have good geometric properties, they should have simpler but more efficient BC construction methods. We use a hierarchical BC construction method for canal surfaces. In our *RQ-sphere* decomposition, a canal surface is subdivided as a set of n G^1 *RQ-spheres*. It is easy to construct a bounding cylinder BC for each *RQ*. For two neighboring RQ_0 and RQ_1, we can have two bounding cylinders BC_0 and BC_1 in a straightforward way, and as shown in Fig. 2, a bigger bounding cylinder BC_2 can be constructed geometrically to enclose BC_1 and BC_2. In the same manner, a binary tree of bounding cylinders HBCT can be organized hierarchically.

3.2 Canal Valid Intersection Intervals (CVII)

Suppose that two canal surfaces are subdivided into n and m *RQ-sphere* pairs respectively. A brute force method for canal/canal intersection would require invoking $n*m$

RQ/*RQ* intersection computations. However, it becomes quite inefficient when two canal surfaces only intersect each other within a small overlapping region (see Fig. 3(1)), in this case, only few *RQ*/*RQ* pairs of the two canal surfaces within their overlapping regions need to be examined. Therefore, by filtering out those *RQ*/*RQ* pairs outside the overlapping regions, the expected performance of the algorithms should improve. To describe it conceptually, *canal valid intersection intervals* CVII is defined as the interval pairs (*u-interval, v-interval*) corresponding to the overlapping regions of the two canal surfaces, hereafter denoted by ($CVII_u$, $CVII_v$). The idea, *bounding cylinders clipping*, hereafter denoted by *BC clipping*, is also proposed to detect ($CVII_u$, $CVII_v$), or to find all those potential intersecting *RQ*/*RQ* candidates for more efficient intersection determination.



(1) (2)

Fig. 3. The valid intersection intervals of two canal surfaces: (1) Definition of VII of two overlapping canal surfaces; (2) Two consecutive rounds of BC clipping

3.3 Clipping of Two Bounding Cylinders

It is difficult to determine the exact interval ($CVII_u$, $CVII_v$) of two canal surfaces R_1 and R_2 directly by solving complicated equations numerically. Therefore, it is acceptable to estimate ($CVII_u$, $CVII_v$) of R_1 and R_2 only approximately but more efficiently.

Similar to computing RVII of two surfaces of revolution [9, 10], we propose *BC clipping* to estimate ($CVII_u$, $CVII_v$) of two canal surfaces approximately by computing the overlapping regions of their respective bounding cylinders $\mathbf{BC_{11}}$ and $\mathbf{BC_{12}}$, then refining the regions recursively as shown in the right part of Fig. 3(b). The first round of *BC clipping* for $\mathbf{BC_{11}}$ and $\mathbf{BC_{12}}$ yields the initial intersection interval ($CVII_{11}$, $CVII_{12}$). It is a very rough approximation to the real CVII, within which there are still some *RQ-sphere* pairs of R_1 and R_2 that have no intersection, since $\mathbf{BC_{11}}$ and $\mathbf{BC_{12}}$ enclose R_1 and R_2 rather loosely. Furthermore, two smaller bounding cylinders $\mathbf{BC_{21}}$ and $\mathbf{BC_{22}}$ are constructed respectively for those *RQ-sphere* pairs within $CVII_{11}$, and $CVII_{12}$, then, the second round *BC clipping* is taken on $\mathbf{BC_{21}}$ and $\mathbf{BC_{22}}$ as shown in the left part of Fig. 3 (b) (amplified version of second round of 2^{nd} *BC clipping*), giving a smaller intersection interval pair ($CVII_{21}$, $CVII_{22}$). Usually, very few rounds of such *BC clipping* output very close to the real ($CVII_u$, $CVII_v$) of two canal surfaces.

4 Computing Intersections of Two Canal Surfaces

The intersection curve of two canal surfaces, hereafter denoted *CSIC*, is computed in three steps: (*i*) computing CVII ($CVII_u$, $CVII_v$) of the two canal surfaces; (*ii*) computing all the intersection curve segments of all potential *RQ*/*RQ* pairs, RQIC, within ($CVII_u$, $CVII_v$) by using Goldman's method [5]; (*iii*) concatenating all the individual *RQIC* into *CSIC* as a set of independent components (open branches or closed loops).

4.1 Computing Intersection Curves of Two Canal Surfaces

The procedure, Find_CVII (BCT_1, BCT_2), is to compute ($CVII_{11}$, $CVII_{12}$) of two canal surfaces. There are three possible cases which should be treated differently:

```
procedure FindCVII (BCT1, BCT2)
   begin
      If both BCT1 and BCT2 are only two bounding cylin-
      ders (0-level BCT) [Case 1]
         compute their CVII (CVII11, CVII12);
         return (CVII11, CVII12);
      If one of them is a bounding cylinder [Case 2]
         assume it be BC1;
      If both of them are BCT [Case 3]
         assume the lower one of them be BC1;
      if BC1 and the root BC of BCT2 do not overlap
         return an empty CVII;
      else
         [check if BC1 and two children BCT21 and BCT22
         overlap recursively]
         Find_CVII(BC1, BCT21);
         Find_CVII(BC1, BCT21);
         add all the individual sub-CVIIs to CVII;
      end if;
      return (CVIIu, CVIIv);
   end.
```

The main idea of our computing intersection curves of two canal surfaces *CSIC* based on *RQ-sphere* decomposition can be sketched as follow:

```
program ComputeCSIC (BCT1, BCT2)
   begin
      (CVIIu CVIIv) = FindCVII (BCT1, BCT2);
      assume CVIIu be shorter one;
      for each RQui (BCui) within CVIIv
         construct BCTvj for all the RQs within CVIIv;
         CVIIij = Find_CVII (BCui, BCTvj);
         for each RQvj within CVIIij
            if RQui and RQvi overlap
               RQICij = ComputeRQIC (RQui, RQvi);
               if RQICij is a closed loop itself
                  output it as a new loop;
               else
```

```
                    Check coincidence of RQIC_ij and prior
                    RQIC_i-1,j along their common bounding
                    circle C_i-1;
                    Check coincidence of RQIC_ij and prior
                    RQIC_i,j-1 along their common bounding
                    circle C_j-1;
                    if no coincidence between them
                      output it as a new branch;
                    else
                       concatenate RQIC_i,j with RQIC_i-1,j and
                       RQIC_i,j-1 respectively;
                       if a closed loop is formed,
                         output it as a new closed loop;
                       end if;
                    end if;
               end if;
end.
```

4.2 Computing Self-intersections of a Single Canal Surface

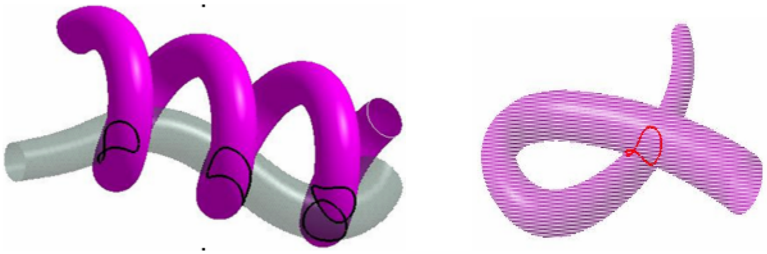
Similarly, self-intersection of a single canal surface *R* can be solved. Assume that the bounding cylinder tree of *R* is BCT_0, *R* is decomposed *n RQ-sphere* pairs RQS_i ($i = 1, 2, \ldots, n$), BCT_0 has two sub-trees BCT_{Left} (enclosing the first half RQS_i, $i = 1, 2, \ldots, [n/2]$) and BCT_{Right} (enclosing the other half RQS_i, $i = [n/2], [n/2]+1, \ldots, n$). If BCT_{Left} and BCT_{Right} overlap, then, *R* may intersect itself and its self-intersection curves can be computed by calling $CSIC(BCT_{Left}, BCT_{Right})$. Otherwise, conquer each half (sub-tree) recursively. This idea is sketched roughly as follows.

```
procedure Canal_Self_Intersection (BCT_0)
  begin
    if BCT_left and BCT_right, two children of BCT_0 overlap
         ComputingCSIC(BCT_Left, BCT_Right);
    else
         Canal_Self_Intersection(BCT_Left);
         Canal_Self_Intersection(BCT_Right);
    Output all the open branches or loops;
  end.
```

5 Illustrative Examples

The proposed algorithms have been implemented with *C++* and *OpenGL* under Windows *XP* and PC (Pentium III, 512M RAM, 512M HZ). Two examples are given in Fig. 4, one is for computing general *CSIC*, the other is for self-intersection. Their spine curves and radii are represented with cubic Beizer form. Both of them are computed within one second.

Comparing determination method of global self-intersection for a single pipe surface in [16], it is simpler algorithmically, easier for implementation and also more robust, since both fundamental intersection computations of BC/BC and *RQ*/*RQ* have closed form solutions.



(1) (2)

Fig. 4. Intersection Curves on canal surfaces: (1) The intersection curves of two canal surfaces; (2) Self-intersection curves of a single canal surface

6 Conclusion

Robustness is one of most important factors for surface intersection algorithms. We have shown and analyzed the instability of the method [6] in our previous work [9]. Our *RQ-sphere* decomposition based method reduces computing intersections of two canals to computing intersections of two *RQ*s, which can be solved by Goldman's method [5] robustly and efficiently. Further, *BC clipping* makes computing the *CSIC* relatively efficient. In fact, the efficiency, accuracy and robustness of two revolute quadrics *RQ*/*RQ* intersection can be further enhanced by more recent algorithms [1, 2, 18, 19]. Also, *RQ-sphere* decomposition facilitates tracing the intersection curves because it is easy to recognize the closed loops and singular points on the intersection curves. Therefore, both theoretical analysis and practical implementation show the robustness and efficiency of our proposed method.

The *RQ-sphere* decomposition of canal surfaces also can be extended to solving other geometric problems of canal surfaces, e.g. collision detection, isophotes, silhouette, bisector, distance computing and so on.

References

1. Dupont L., Lazard D., Lazard S., Petitjean S.: Towards The Robust Intersection of Implicit Quadrics. In Proc. of Workshop on Uncertainty in Geometric Computations, Sheffield, UK, (2001).
2. Dupont L., Lazard D., Lazard S., Petitjean S.: Near-Optimal Parameterization of the Intersection of Quadrics. In *ACM Symposium on Computational Geometry*, San Diego, USA, (2003).
3. Elber G., IRIT Software, http://www.cs.technion.ac.il/~irit/, The Technion-IIT, Haifa, Israel, (1993).
4. Eberly David: Intersection of two cylinders. Magic Software Company. http://www.magic -software. com/ Intersection.html (2000).
5. Goldman R. N.: Quadrics of Revolution. IEEE Computer Graphics and Applications. 3(2), (1983) 68-76.
6. Heo H. S., Hong S. J., Seong J. K., Kim M. S.: The Intersection of Two Ringed Surfaces and Some Related Problems. Graphical Model. 63(4), (2001) 228-244.

7. Johnstone J.: A New Intersection Algorithm for Cyclides and Swept Surfaces Using Circle Decomposition. CAGD, 10(1). (1993) 1-24.
8. Jia J., Tang K., Joneja A, Kwok K. W.: New Algorithm of Computing Planar Sections of Surfaces of Revolution Based on Revolute Quadric Subdivision". Proc. of Shape Modeling and Applications. Genova. Italy. (2004).
9. Jia J., Baciu G., Kwok K. W.: Revolute Quadric Decomposition for Computing Intersection Curves of Surfaces of Revolution. Graphical Model. 55(5). (2004) 363-383.
10. Jia J., Tang K, Joneja A.: Bi-Conic Subdivision for Surfaces of Revolution and Its Applications to Intersection Problems. The Visual Computer. 20(7), (2004) 457-478.
11. Jia J., Joneja A. and Tang K., "Revolute Quadric Decomposition of Canal Surfaces and Its Application". LNCS. Vol. 3514. Springer-Verlag. (2005) 213-222.
12. Kim M. S.: Intersecting Surfaces of Special Types. Proc. of Shape Modeling and Processing. University of Aizu, Japan. (1999) 122-128.
13. Levin J. Z.: A Parametric Algorithm for Drawing Pictures of Solid Objects Composed of Quadrics. Communications of the ACM. 19(10). (1976) 555-563.
14. Levin J. Z..: Mathematical Models for Determining The Intersections of Quadric Surfaces. Computer Graphics and Image processing. Vol. 1, (1979) 73-87.
15. Nelson Max: Cone-Spheres. ACM SIGGRAPH Computer Graphics, Vol. 24, (1990) 59-62.
16. Patrikalakis N. M., Maekawa T.: Shape Interrogation in Computer Aided Design and Manufacturing. Springer Verlag. (2002).
17. Seong J. K., Kim K. J., Kim M. S., Elber G.: Intersecting a Freeform Surface with a Ruled or a Ringed Surface. Proceedings of Geometric Modeling and Processing. Beijing, (2004) 38-45.
18. Wang W., Goldman R., Tu C.: Enhancing Levin's Method for Computing Quadric-Surface Intersections. CAGD, 20(7). (2003) 401-422.
19. Wang W., Joe B., Goldman R.: Computing Quadric Surface Intersections Based on An Analysis on Plane Cubic Curves. Graphical Model. 64(6), (2003) 335-367.
20. Paluszny, M., Bühler, K.: Canal Surfaces and Inversive Geometry. Mathematical Methods for Curves and Surfaces II. In Daehlen M., Lyche T., Schumaker L. (eds.), (1998) 367-375.