

Reconstruction of 3D Curvilinear Wireframe Model from 2D Orthographic Views

Aijun Zhang¹, Yong Xue^{1,2*}, Xiaosong Sun¹, Yincui Hu¹, Ying Luo¹,
Yanguang Wang¹, Shaobo Zhong¹, Jianqin Wang¹, Jiakui Tang¹, and Guoyin Cai¹

¹Laboratory of Remote Sensing Information Sciences, Institute of Remote Sensing Applications, Chinese Academy of Sciences, P. Box 9718, Beijing 100101, China,
aj.zhang@263.net

²Department of Computing, London Metropolitan University, 166-220 Holloway Road,
London N7 8DB, UK
y.xue@londonmet.ac.uk

Abstract. An approach for reconstructing wireframe models of curvilinear objects from three orthographic views is discussed. Our main stress is on the method of generating three-dimensional (3D) conic edges from two-dimensional (2D) projection conic curves, which is the pivotal work for reconstructing curvilinear objects from three orthographic views. In order to generate 3D conic edges, a five-point method is firstly utilized to obtain the algebraic representations of all 2D projection curves in each view, and then all algebraic forms are converted to the corresponding geometric forms analytically. Thus the locus of a 3D conic edge can be derived from the geometric forms of the relevant conic curves in three views. Finally, the wireframe model is created after eliminating all redundant elements generated in previous reconstruction process. The approach extends the range of objects to be reconstructed and imposes no restriction on the axis of the quadric surface.

1 Introduction

Automatic conversion from 2D engineering drawings to solid models which allows existing engineering drawings to be fully used for newer designs is an important research topic in computer graphics and CAD. Much work has already been done on automatically reconstructing solid models from orthographic views to date[1-6]. However, the existing approaches have some limitations, which hinder the work from developing further. One of the major limitations is the narrow range of solid objects which can be generated from 2D views. The earlier work was able to generate planar polyhedral objects whose projections contain only lines[1,2]. Most of the later research has concerned with extending the range of objects to be reconstructed. Sakurai[3], Gu[4], Lequette[5], and Kuo[6] extended the earlier method to deal with curved surfaces. However, they restricted the orientation of the quadric surface to be parallel to one of the coordinate axes or projection planes.

In this paper, we propose a wireframe-oriented approach that can handle a wider variety of manifold objects with curved surfaces and conic edges than existing

* Corresponding author

methods, and impose no restrictions on the orientation of the quadric surface. A key idea of our work is to utilize five-point method to generate 3D conic edges, which was employed by Kuo. Nevertheless, our method differs from that of Kuo's somewhat. The difference will be discussed in section 3.2.

2 Preprocessing of the Input Drawing

The input drawing consists of three orthographic views of the objects. i.e., front view, top view, and side view. Only the necessary geometric elements in the drawing are taken into consideration, and three views have been separated and identified[6]. In addition, auxiliary lines must be added to the views as the corresponding projections of tangent edges and/or silhouette edges for curved objects.

2D points and segments in each view are stored in $P_list(view)$ and $S_list(view)$, respectively, where $view$ is front, top, or side. Each item of $P_list(view)$ holds coordinate value and type. Each item of $S_list(view)$ holds indices of its two endpoints and type.

3 Generation of Wireframe

Generating wireframe is the early stage of reconstruction using wireframe-oriented approach, during which all the possible 3D elements constituting the wireframe are recovered from the 2D elements in three views. We call the 3D elements candidate vertices, candidate edges, and candidate faces during the reconstruction, written as c-vertices, c-edges, and c-faces, respectively.

3.1 Generation of 3D Candidate Vertices

We use the method to generate 3D c-vertices which has been detailed in previous work [1-6]. The c-vertices are stored in $V_list()$, each item of which holds coordinate values, corresponding points in each views, associated 3D edges, and type.

3.2 Generation of 3D Candidate Edges

In this section, we focus on the method for generating 3D conic edges, which is based on some important properties of the conic section under orthographic projection.

3.2.1 Construction and Representation of Projection Conics

Conic curves are defined in two ways in general: implicitly and parametrically. In solid modeling, a common way of representing a conic is by its more intuitive geometric parameters[7], so this representation form, unlike the implicit one, requires a different definition for each type of conic. In most cases, however, the type of a projection curve is not directly available from the views, for there are no explicit

notations for it. Hence it is much difficult to derive the straightforward geometric representation for each type of conic curve. In our approach, therefore, the five-point method is employed to obtain the algebraic equation of a 2D projection conic curve firstly, and then the algebraic form is converted to the geometric representation analytically, seeing that it is more easy to obtain the algebraic equation of a conic curve and this form can be conveniently converted to the geometric representation. While in Kuo’s approach, the geometric representation of 2D projection conic curves are obtained directly, which needs more complex geometric computation because no more definite conic types are available from the views.

We begin with describing the general equation for a 2D conic curve in the projection plane coordinates. Basically, the 2D coordinate systems associated with the three projection planes are denoted as x - y for top view, x - z for front view, and y - z for side view, respectively. Without loss of generality, suppose $c_i(p_1, p_2)$ is a curve in top view, in which $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$ are the endpoints of the curve. We regard it as a conic on the basis of the theorems of the conics under orthographic projection [8].

The conic curve c_i in top view can be described by an algebraic expression:

$$a_{11} x^2 + 2a_{12} xy + a_{22} y^2 + 2a_{13} x + 2a_{23} y + a_{33} = 0 \tag{1}$$

In order to construct this conic curve, we choose arbitrary three points excluding two endpoints p_1 and p_2 , say $p_i(x_i, y_i)$, $i=3,4,5$. The five points, $p_i(x_i, y_i)$, $i=1,2,\dots,5$, in which no three points are collinear, uniquely determine the projection conic. The conic coefficients in Eq.(1) are the solution to

$$\mathbf{P} \begin{bmatrix} a_{11} & a_{12} & a_{22} & a_{13} & a_{23} & a_{33} \end{bmatrix}^T = 0$$

where

$$\mathbf{P} = \begin{bmatrix} x_1^2 & 2x_1 y_1 & y_1^2 & 2x_1 & 2y_1 & 1 \\ x_2^2 & 2x_2 y_2 & y_2^2 & 2x_2 & 2y_2 & 1 \\ x_3^2 & 2x_3 y_3 & y_3^2 & 2x_3 & 2y_3 & 1 \\ x_4^2 & 2x_4 y_4 & y_4^2 & 2x_4 & 2y_4 & 1 \\ x_5^2 & 2x_5 y_5 & y_5^2 & 2x_5 & 2y_5 & 1 \end{bmatrix}$$

3.2.2 Classification of Projection Conics

The type of a conic expressed in the form of Eq.(1) is determined easily by being converted to the geometric representation according to analytic geometry[7]. The geometric parameters for the conic are derived as follows:

(1) The orientation θ of conics

$$\theta = \frac{1}{2} \text{ctg}^{-1} \frac{a_{11} - a_{22}}{2a_{12}}$$

(2) The geometric parameters for central conics

$$x_0 = \frac{a_{12} a_{23} - a_{22} a_{13}}{a_{11} a_{22} - a_{12}^2} \quad y_0 = \frac{a_{12} a_{13} - a_{11} a_{23}}{a_{11} a_{22} - a_{12}^2}$$

$$\{a, b\} = \begin{cases} \left\{ \sqrt{-a'_{33}/a'_{11}}, \sqrt{-a'_{33}/a'_{22}} \right\} & \text{for an ellipse} \\ \left\{ \sqrt{-a'_{33}/a'_{11}}, \sqrt{a'_{33}/a'_{22}} \right\} & \text{for a hyperbola} \end{cases}$$

where

$$\begin{aligned} a'_{11} &= \frac{a_{11} + a_{22}}{2} + \frac{a_{11} - a_{22}}{2} \cos 2\theta + a_{12} \sin 2\theta \\ a'_{22} &= \frac{a_{11} + a_{22}}{2} - \frac{a_{11} - a_{22}}{2} \cos 2\theta - a_{12} \sin 2\theta \\ a'_{33} &= (a_{13} x_0 + a_{23} y_0) + a_{33} \end{aligned}$$

(3) The geometric parameters for a parabola

$$\begin{aligned} p &= \sqrt{-I_3/I_1^3} \\ x_0 &= \frac{c^2 - 2a_{23}c + a_{22}a_{33}}{2(a_{12}a_{23} - a_{13}a_{22})} \\ y_0 &= \frac{a_{11}c^2 - 2a_{12}a_{13}c + a_{12}^2a_{33}}{2a_{12}(a_{13}a_{22} - a_{12}a_{23})} \end{aligned}$$

where

$$I_1 = a_{11} + a_{22} \quad I_3 = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{vmatrix} \quad c = \frac{a_{12}a_{13} + a_{22}a_{23}}{a_{11} + a_{22}}$$

From the above derivations, we could obtain the parameter equations of commonly utilized conic curves, e.g., elliptical curves, hyperbolic curves, and parabolic curves, easily [7].

3.2.3 Reconstruction of 3D Conics

We now consider a point matching method to determine the relationship between the three projection conic curves of a 3D conic. Let v_1 and v_2 be two different vertices and their corresponding points in three views be (pt_1, pf_1, ps_1) and (pt_2, pf_2, ps_2) , respectively. A conic edge, $e(v_1, v_2)$, was found when there existed a conic connection between the two corresponding points of v_1 and v_2 in each view, viz. $c_i(pt_1, pt_2)$ between pt_1 and pt_2 in top view, $c_i(pf_1, pf_2)$ between pf_1 and pf_2 in front view, $c_i(ps_1, ps_2)$ between ps_1 and ps_2 in side view, and it satisfied Eq.(2), providing there was no internal vertices on the edge, in between v_1 and

$$|pt_i(x) - pf_i(x)| \leq \varepsilon \quad \text{and} \quad |pt_i(y) - ps_i(y)| \leq \varepsilon \quad \text{and} \quad |pf_i(z) - ps_i(z)| \leq \varepsilon \quad (2)$$

where $pt_i \in c_i(pt_1, pt_2)$, $pf_i \in c_i(pf_1, pf_2)$, $ps_i \in c_i(ps_1, ps_2)$, and $i=3,4,5$, ϵ is a tolerance introduced to allow for an inexact matching problem, considering the fact that the input data may not give an exact alignment of coordinates between each view. According to the point matching method, when the expression of each corresponding projection conic curve in three views is generated, it is easy to derive the 3D conic.

Without loss of generality, let

$$\begin{cases} x = \phi_1(\alpha) \\ z = \phi_2(\alpha) \end{cases} \quad \alpha_1 \leq \alpha \leq \alpha_2 \tag{3}$$

and

$$\begin{cases} x = \phi_1(\beta) \\ y = \phi_2(\beta) \end{cases} \quad \beta_1 \leq \beta \leq \beta_2 \tag{4}$$

be two corresponding projections of a 3D conic edge, in the front and top views, respectively. Eq.(3) and (4) share the x-axis and have the same x-coordinate. Therefore we can derive that the combination of Eq.(3) and (4) associates with the locus of a 3D conic edge.

Observe Eq.(3) and (4), in order to eliminate one of the two variants(e.g. α), it is necessary to review the algebraic form of the projection conic in front view

$$f(x, z) = a_{11}x^2 + 2a_{12}xz + a_{22}z^2 + 2a_{13}x + 2a_{23}z + a_{33} = 0 \tag{5}$$

solving the z-coordinate value, then the explicit form of Eq.(5) is expressed as

$$z = \frac{-(a_{12}x + a_{23}) \pm \sqrt{(a_{12}x + a_{23})^2 - a_{22}(a_{11}x^2 + 2a_{13}x + a_{33})}}{a_{22}} \tag{6}$$

We can rewrite Eq.(6), substituting x value derived from Eq.(4) for the original x value in Eq.(3), as the following simplified form:

$$z = \phi_3(\beta) \quad \beta_1 \leq \beta \leq \beta_2 \tag{7}$$

It follows that the corresponding 3D conic is the locus which satisfies

$$\begin{cases} x = \phi_1(\beta) \\ y = \phi_2(\beta) \\ z = \phi_3(\beta) \end{cases} \quad \beta_1 \leq \beta \leq \beta_2 \tag{8}$$

Thus, in accordance with the types of the relevant 2D projection conic curves in three views, we can get the corresponding 3D conic depicted in the form of Eq.(8).

3.2.4 3D c-Edges Reconstruction Algorithm

In previous algorithms, the process to generate c-edges is obviously time consuming[1-6]. In this section, an accelerated method is introduced to decrease the processing time. The major steps of 3D c-edges generation procedure are as follows:

Step1. Select a 2D projection conic segment arbitrarily in one view, e.g., $c_i(pt_1, pt_2)$ in top view, where $pt_1, pt_2 \in P_list(Top)$, $c_i \in S_list(Top)$;

Step2. Get two 3D vertices, v_1 and v_2 , from the $V_list()$, whose corresponding projections in top view are pt_1 and pt_2 , respectively. Then search for the corresponding 2D projections of v_1 and v_2 in another two views, i.e., the front and side views, accordingly $pf_1, pf_2 \in P_list(Front)$, $ps_1, ps_2 \in P_list(Side)$, respectively;

Step3. Examine each pair of 2D points, (pf_1, pf_2) and (ps_1, ps_2) , to determine whether or not there exists a 2D curve segment connecting the two points of each pair. If it is true, it is certain that at least one 3D edge between v_1 and v_2 ;

Step4. In accordance with the type of $c_i(pt_1, pt_2)$ in top view, choose the corresponding 2D curve segments, $c_f(pf_1, pf_2)$ and $c_s(ps_1, ps_2)$, respectively, from the remaining two views. A 3D c-edge between v_1 and v_2 then can be generated by applying the method for constructing a 3D conic discussed in previous sections.

During the above steps, each 2D segment is labeled as *examined* after being examined. In this case, the performance of constructing 3D c-edges is finished until all segments in three views are labeled as *examined*.

All 3D c-edges are stored in $E_list()$, each item of which holds two endpoints, corresponding conic segments in each of three views, parametric equation of the 3D conic containing the edge, and type.

3.3 Construction of Wireframe

There may exist some redundant elements in $V_list()$ and $E_list()$, on account of the fact that the reconstruction is a process of recovering information from low dimensions to high dimensions. In the stage of wireframe generation, redundant elements generally involve overlapping edges and pathological elements[2], which may not only increase the complexity of the computation, but also introduce ambiguities in the wireframe as well as solid model generation process. Therefore, they must be eliminated from $V_list()$ and $E_list()$, respectively.

Thus, we can establish the wireframe with the $V_list()$ and $E_list()$ containing the information of 3D vertices and edges, whose reprojections are identical with the input drawings, and that each element, i.e., vertex or edge, in the wireframe satisfies the following topological conditions:

- (1) $\forall v_j \in V_list(), \exists \rho(v_j) \geq 3$;
- (2) $\forall e_j \in E_list(), \exists v_j, v_{j+1} \in V_list()$, where v_j, v_{j+1} are two endpoints of e_j ;
- (3) $\forall e_j, e_k \in E_list(), \exists e_i \cap e_j = \{v_k\}$ and $v_k \in V_list()$, or $e_i \cap e_j = \emptyset$.

Where $\rho(v)$ denotes the edge connectivity of the c-vertex v .

4 Implementation

Based on the method described above, the prototype implementation of reconstruction is realized in C++. Figure 1 demonstrates the case that can be handled by our method, the implementation is restricted to three orthographic views and curvilinear objects. Figure 1(b) shows the wireframe reconstructed from a three-view engineering drawing with straight lines, circular and elliptical arcs in Figure 1(a).

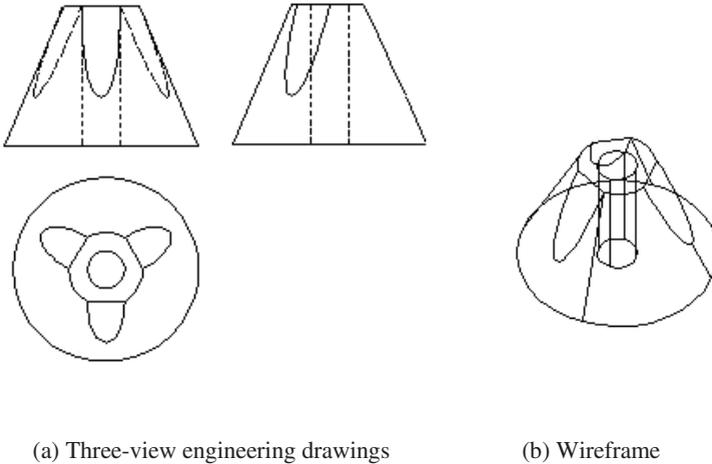


Fig. 1. The wireframe of an object reconstructed from a three-view engineering drawing with straight lines, circular and elliptical arcs

5 Conclusion

A wireframe-oriented approach to reconstruct 3D wireframes of curvilinear solids from three orthographic views is presented, in which the method to generate conic edges is emphasized. In order to obtain 3D conic edges, the five-point method is firstly applied to obtain the geometric representations of 2D projection conic curves in each view in two steps, and then 3D conic edges described in the form of coordinate locus are accordingly derived from corresponding 2D projections afterwards by using point matching method. In addition, an accelerated algorithm to generate 3D edges is introduced to decrease the processing time. At last, the wireframe is established when the redundant elements are removed from candidate vertices and edges. Our approach extends the range of objects to be reconstructed, i.e., an object may include straight lines, circular arcs, elliptical arcs, parabolic arcs, and hyperbolic arcs, and imposes no restriction on the axis of the quadric surface.

Acknowledgement. This publication is an output from the research projects "CAS Hundred Talents Program", "Digital Earth" (KZCX2-312) funded by Chinese Academy of Sciences and "Dynamic Monitoring of Beijing Olympic Environment Using Remote Sensing" (2002BA904B07-2) funded by the Ministry of Science and Technology, China.

References

1. Wesley MA, Markowsky G. Fleshing out projection. *IBM Journal of Research and Development* 1981; 25(6): 934-954.
2. Yan QW, Philip CL, Tang Z. Efficient algorithm for the reconstruction of 3-D objects from orthographic projections. *Computer-aided Design* 1994; 26(9): 699-717.
3. Sakurai H, Gossard DC. Solid model input through orthographic views. *Computer Graphics* 1983; 17(3): 243-25.
4. Gu K, Tang Z, Sun J. Reconstruction of 3D solid objects from orthographic projections. *Computer Graphics Forum* 1986; 5(4): 317-324.
5. Remi Lequette. Automatic construction of curvilinear solids from wireframe views. *Computer-aided Design* 1988; 20(4): 171-179.
6. Kuo MH. Reconstruction of quadric surface solids from three-view engineering drawings. *Computer-aided Design* 1998; 30(7): 517-527.
7. Wilson PR. Conic representations for shape description. *IEEE Computer Graphics and Applications* 1987; 7(4): 23-30.
8. Nalwa VS. Line-drawing interpretation: straight lines and conic sections. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1988; 10(4): 514-529.