1998 International Computer Symposium
Workshop on Algorithms
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

EXTRACTION OF QUADRIC SURFACES FROM WIRE-FRAME MODELS

M. H. Kuo

Graduate School of National Defense Information,
National Defense Management College,
P.O. Box 90046-17, Chung-Ho, Taipei, Taiwan, R.O.C.
Email:mhk@rs590.ndmc.edu.tw

ABSTRACT

The automatic conversion of 3D wire-frame models to boundary representation solids is very important for one-off conversion of line drawings to solid modeling system. In this paper, a minimum internal angle (MIA) algorithm that efficiently finds all quadric surfaces in a wire-frame model is addressed. It requires considerably less searching time than depth-first searches that could grow exponentially in complexity. In addition, the proposed method is benefit for easily description the geometry of the traced surfaces when compared to other methods.

1. INTRODUCTION

In CAD, designer uses a graphic system to simply draw a wire-frame model on screen to describe the shape of an object continues to be popular for early conceptual stages of a design. However, the main role for such concept is the ability to automatically convert the wire-frame models to boundary representation solids, i.e. the extraction of accurate surface information from a 3D wire-frame model.

Previous attempts at automatic surface extraction from a wire-frame model are generally grouped into topological and geometric approach [1]. The topological approaches treat the search as a graph-connecting problem. It uses concepts from graph theory to derive the face topology and avoids the geometric reasoning about the features of a wire-frame. Based on such technique, Hanrahan [2], Dutton [3] presented work that started with an adjacency list structure of an undirected graph. A planarity algorithm [4] was used to create a directed graph. The faces of a wire-frame model thus could be found by constructing a new adjacency list structure, which was based on the embedding of the directed graph for the original undirected graph. Their algorithm was linear in the number of vertices in time and space. However, it was constrained to wire-frame with 3-connected planar surfaces with no holes.

Later, Ganter [5], Courter [6] and Hojnicki [7] also proposed purely topological techniques, based on Paton's algorithm [8], that generated a set of fundamental cycles from a wire-frame. Then, the set of cycles was reduced to the set of candidate faces according to two heuristic observations:

1. Any cycle that is a true face in a given object has a minimum number of edges in common with any other cycle.
2. The sum of all edges in all cycles is a minimum when the cycles are the faces of an object.

The geometric approach is based on the pure geometric information of the wire-frame to identify three dimensional face loops. It assumes that the types of two adjacent edges can define a surface and its equation. For example, two straight lines define a planar face; a straight line and a circular arc define a cylinder; two circular arcs define a sphere or a torus [9-12]. Once the type of a surface is defined, a depth first search method searches all edges on this surface. Since a face loop is closed, the search is completed when it returns to the start vertex, providing none of the edges intersect each other.

From the above discussion, the topological approach provides a good framework for dealing with the arbitrary surface type of the wire-frame model in automatic face loop identification. However, this approach has its inherent drawbacks. Firstly, it cannot describe the type of a face, i.e. the face equation. This is because it relies on the relative connectivity of the graph elements to generate the desired edge cycles. Thus, the cycles are nothing but a set of circuits. Secondly, using a reduce-fundamental-cycle method [6] for wire-frame conversion can be very expensive. For some highly symmetric wire-frame models, the size of the established fundamental cycle matrix could be very large. Finally, this approach has no guarantee that it can always establish correct face loops for objects whose genus greater than zero, nor even for 3-connected planar wire-frames. The problem is that these models do not have unique graph embedding [2].

In the geometric approach, it is clear that tracing planar loops is very straight forward by using the vector product of two adjacent straight edges to define an assume face normal. However, the identification of a curved surface equation from the types of two adjacent edges is ill defined without severely restricting the scope of models that can be handled. For example, a straight edge and a circular edge could define a planar, cylindrical, conical or other higher order curved surface. On the other hand, a conical surface could be defined by two of any kind of coinc edges. In this approach, an exhaustive depth-first search method is

usually used to find a close loop that is very expensive. In addition, it could also produce face loop with inner edge when the starting vertex was not a convex hull vertex [13]

In this paper, a method that efficiently extracts all quadric faces in a wire-frame model is addressed. Unlike previous mentioned approaches; this method finds quadric face loops that have no inner edges subdividing them and avoids finding the unwanted outer loops. It requires considerably less searching time than depth-first searches that could grow exponentially in complexity, and it simplifies the subsequent face type identification.

2. THE MIA ALGORITHM

The method used to automatically trace three dimensional face loops in a wire-frame model is called the minimum internal angle (MIA) method. This includes five steps as follows (see figure 1):

Step 1. *Choose a convex hull vertex v_1 that has not been processed yet as the tracing-start vertex. This can be, for example, the vertex with minimum x-coordinate.*

This step ensures that the edge loop will lie on one side of v_1, so that an orientation can be assumed that avoids producing edge loops with inner edges subdividing them.

Step 2. *Choose a pair of edges $e_1(v_0, v_1)$ and $e_2(v_1, v_2)$ adjacent to v_1, that have not paired together before, and a third edge $e_3(v_2, v_3)$ to define the surface equation.*

The general implicit quadric surface can be represented as follows:

$$f(x,y,z) = Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fzx + Gx + Hy + Jz + K = 0 \quad (1)$$

To solve (1), we could choose nine different vertices on e_1, e_2 and e_3 to work out the ten homogeneous coefficients.

Step 3. *Choose e_2 as the tracing-start edge and define the turning direction. This includes two sub-steps*:

(a) Calculate the surface gradient ∇_1 at v_1 to define the auxiliary plane P_1 common to ∇_1 and e_2 where

$$\nabla_1 = \frac{\partial f(v_1)}{\partial x} i + \frac{\partial f(v_1)}{\partial y} j + \frac{\partial f(v_1)}{\partial z} k \quad (2)$$

$$P(v) = (\nabla_1 \times (v_1' - v_1)) \bullet (v - v_1) = 0 \quad (3)$$

(b) The turning direction T_1 is assumed to be clockwise when $P(v_0') \geq 0$, otherwise, counterclockwise.

Step 4. *Choose a next edge, e_3, adjacents to v_2 at the far end of e_2 that satisfies the surface equation. If there is a choice, choose the edge with minimum internal turning angle.*

The internal angle, θ, between e_2 and e_3 is calculated using two intersection vertices v_2', v_3'. When turning direction at e_3 about their auxiliary plane is T_2, then

$$\theta = \begin{cases} \pi - \cos^{-1}\left(\dfrac{\vec{d}_2 \bullet \vec{d}_3}{|\vec{d}_2||\vec{d}_3|}\right) & when\ T_2 = T_1 \\ \pi + \cos^{-1}\left(\dfrac{\vec{d}_2 \bullet \vec{d}_3}{|\vec{d}_2||\vec{d}_3|}\right) & otherwise \end{cases} \quad (4)$$

where $\vec{d}_2 = (v_2' - v_2)$ and $\vec{d}_3 = (v_3' - v_2)$

The calculations of inverse function, $\cos^{-1}()$ in equation (4) involve finding the convergence of its power series, which is very time consuming. This can also be implemented using comparisons of the dot and cross products, so that square-root and inverse-cosine function are avoided, as follows:

Let

$\vec{d}_2 = (x_2, y_2, z_2)$, $\vec{d}_3 = (x_3, y_3, z_3)$ and

$$\omega = \frac{\vec{d}_2 \bullet \vec{d}_3}{|\vec{d}_2||\vec{d}_3|} = \frac{(x_2x_3 + y_2y_3 + z_2z_3)}{(x_2^2 + y_2^2 + z_2^2)^{1/2}(x_3^2 + y_3^2 + z_3^2)^{1/2}}$$

where $-1 \leq \omega \leq 1$

The target of the minimum internal angle method is to choose the minimum internal angle edge as the next tracing edge. In other words, it is to find the edge with the smallest internal angle in a set of co-surface candidate edges that are adjacent to the current tracing vertex. This allows us to subtract π on both sides of (4), and rewrite it as

$$\theta' = \begin{cases} -\cos^{-1}(\omega) & when\ T_2 = T_1 \\ \cos^{-1}(\omega) & otherwise \end{cases} \quad (5)$$

where $\theta' = \theta - \pi$

The *inverse cosine* is a monotonically decreasing function between $\omega = -1$ and $\omega = 1$. Therefore, we may just depend on the ω value in (5) to decide the edge's internal angle as follows

$$\theta'' = \begin{cases} \omega & when\ T_2 = T_1 \\ 2 - \omega & otherwise \end{cases} \quad (6)$$

Let $\delta = \omega^2$, (6) can be divided into four cases as

$$\theta''' = \begin{cases} \delta & if\ \vec{d}_2 \bullet \vec{d}_3 \geq 0\ and\ T_2 = T_1 \\ -\delta & if\ \vec{d}_2 \bullet \vec{d}_3 < 0\ and\ T_2 = T_1 \\ 2 - \delta & if\ \vec{d}_2 \bullet \vec{d}_3 \geq 0\ and\ T_2 \neq T_1 \\ 2 + \delta & if\ \vec{d}_2 \bullet \vec{d}_3 < 0\ and\ T_2 \neq T_1 \end{cases} \quad (7)$$

where $\delta = \dfrac{(x_2x_3 + y_2y_3 + z_2z_3)^2}{(x_2^2 + y_2^2 + z_2^2)(x_3^2 + y_3^2 + z_3^2)}$

It should be noted that (7) uses one more multiplication and two more comparisons, but one less square roots and one less inverse-cosine calculations, when compared to (4).

Step 5. *When e_1 is retraced, a face-loop has been found, but when no more edges fit, abandon the search. Otherwise go to step 1.and treat v_2 as v_1.*

Application of the above steps to all vertices in the wire-frame results in a set of quadric faces.

The reason for choosing a convex hull vertex, as the tracing-start vertex in step 1 is that it avoids producing edge loops with inner edges subdividing them and it avoids the outermost edge loop. For example, in figure 2 there are six coplanar edges and three face loops. If the face tracing algorithm had started from v_4 with the normal direction for face f_1 set to be $e(v_4,v_3)\times e(v_4,v_5)$, then a face loop $f(v_4,v_3,v_5)$ would be traced. But the second face loop would then be $f(v_1,v_2,v_3,v_5)$ with two inner edges $e(v_4,v_3)$ and $e(v_4,v_5)$. This is the unwanted outer edge loop. On the other hand, if the tracing-start vertex is always a convex hull vertex, v_1 for example, then the two inner face loops, $f(v_1,v_2,v_3,v_4,v_5)$ and $f(v_3,v_5,v_4)$ are correctly found. Step 2 and step 3 assumed a quadric surface and an orientation of the surface that is used to choose the next tracing edge. Step 4 applies (7) to choose a minimum internal angle as the next tracing edge. Finally, step 5 iteratively assigns the current tracing vertex and edge. Application of the above steps to all vertices in the wire-frame results in a set of quadric faces.

Note that in the MIA algorithm, once a pair of edges is chosen to define an assumption surface equation, the choice of next tracing edge is forced. When there is more than one edge on the assumption surface, the algorithm always selects an edge with minimum internal angle. Therefore, to trace a candidate face, the complexity is linear to the vertex degree.

3. CONTAINMENT TEST

Sometimes a 3D object includes holes whose loops are contained in another loop. We need a containment test to decide the "hole face" (not a true face). There are three well-known methods for performing such a test. These are half-plane containment test, sum-of-angle containment test, and count-ray-crossing containment test [15].

The half-plane method is computationally attractive. However, this method works only for convex polygons. The sum-of-angle method works for both convex and concave polygons. However, it may not work in the case of loop with curved edges. In addition, this method turns out to be very slow as it sums the signed angles formed at the test point by each edge's end points. The count-ray-crossing method is based on the *Jordan curve theorem* [15]. This theorem is always true when the points in a given region are 2-connected. Since there is no intersection between two faces (with the same face equation), either one face completely surrounds the other or they do not overlay at all. Therefore, the count-ray-crossing algorithm is the most useful one for the containment test.

To test two 3D face loops, we project back to their 2D views (orthographic projections). Then we apply the count-ray-crossing method to determine the containment relationship between two loops. Theoretically, to complete the containment test we repeat the test procedure for each point of a testing loop with each edge of the other loop. However, according to the MIA method, a face loop is either totally inside or outside the other loop. Therefore, we just need a point to complete the test. This saves much of the testing time.

In general, faces many contain each other in a hierarchical manner. Once the status of one face has been established, the remaining can be deduced. They are alternately face and hole, i.e., $f_i(\text{face}) \supset f_{i+1}(\text{hole}) \supset f_{i+2}(\text{face}) \cdots$ where f_i is a true face when i is even, and a hole when i is odd.

4. TYPE OF THE SURFACES

The type of quadric surface can be determined by translating the general quadric form into a central quadric form with its center at the origin. However, the axes of the quadric can be made to align with the coordinate axes by three rotation, whether it is central or non-central [14].

When the quadric is central, the result of the translation and rotation gives a standard form equation as in (8).

$$f(x,y,z) = \alpha x^2 + \beta y^2 + \gamma z^2 + \kappa = 0 \qquad (8)$$

The type of the quadric surface is usually classified according to the signs and real values of the coefficients that appear in the standard form.

(1) If $\alpha = \beta = \gamma > 0$ and $\kappa > 0$, the quadric surface is spherical.

(2) If $\alpha \neq \beta \neq \gamma > 0$ and $\kappa > 0$, the quadric surface is elliptic.

(3) If $\alpha = \beta > 0$, $\gamma = 0$ and $\kappa > 0$, the quadric surface is cylindrical. The axis is aligned with the z-coordinate. Similarly, if $\alpha = \gamma > 0$, $\beta = 0$ the axis is aligned with the y-coordinate. If $\beta = \gamma > 0$ and $\alpha = 0$ the axis is aligned with the x-coordinate. When two of the non-zero coefficients are unequal, an elliptical cylindrical surface results.

(4) If $\kappa = 0$, $\alpha = \beta > 0$ and $\gamma < 0$, the quadric surface is circularly conical with its axis aligned with the z-coordinate. Actually, the axis of a cone is associated with the coordinate of the negative constant. As with a cylinder, if $\alpha \neq \beta > 0$ the surface is on an elliptic cone.

(5) if $\kappa \neq 0$, $\alpha, \beta > 0$ and $\gamma < 0$, the surface is on a hyperboloid with one sheet whose axis is the z-coordinate axis.

(6) if $\kappa \neq 0$, $\alpha, \beta < 0$ and $\gamma < 0$, the surface is on a hyperboloid with two sheet. The axis is associated with the positive constant.

When the quadric is non-central, the linear terms in equation (1) cannot be eliminated. The standard form is then represented as

$$\alpha x^2 + \beta y^2 + \gamma z = 0 \qquad (9)$$

Similarly, the type of the non-central surface is defined by the coefficients of (9) as

(1) If $\alpha, \beta > 0$ and $\gamma > 0$, the quadric surface is on a downward elliptic paraboloid where z-coordinate is the axis. If $\gamma < 0$ the paraboloid is upward, i.e. the surface opens along the positive z-axis.

(2) If $\alpha < 0$, $\beta > 0$ and $\gamma > 0$, the quadric surface is on a downward hyperbolic paraboloid in which the focal axis is the x-axis, the saddle downward. If $\gamma < 0$, the saddle is upward. When $\alpha > 0$ and $\beta < 0$ the focal axis is the y-axis, the saddle's open direction is decided by the sign of γ.

(3) If $\alpha = 0$ or $\beta = 0$ the surface lies on a parabolic cylinder.

The above discussion of quadric surface type can be summarized as in Table 1.

5. EXPERIMENTAL RESULTS

The MIA method presented above has been applied as part of a complete system (implemented in C++ on a SUN4 UNIX system) for the automatic interpretation of three-view engineering drawing [17]. Two examples are given, in figure 3 and 4, to demonstrate the power of the MIA method. In figure 3(a), a wire-frame model with some redundant faces is reconstructed from 2D, 3-view engineering drawings using a Junction-Match method [16]. There are 18 planar face loops are traced by the MIA method as shown in figure 3(b). Then, 7 true faces are extracted, and a surface shading solid is displayed as in figure 3(c).

The wire-frame model shown in figure 4 includes three types of natural quadric surfaces, the spherical, conical and cylindrical surfaces.

6. CONCLUSION

Now a day, in geometric lofting and the design of cars, ships and aircrafts are mainly performed by CAD systems. However, in many cases these systems provide only wire-frame models. In this case, a method applied to automatically generate solid boundary representations from the wherefore model is very important. Especially, this is important for one-off conversion of drawings to solid modeling systems.

Previous attempts at automatic surface extraction from a wire-frame model are generally grouped into topological and geometric approach. The topological approach is good for dealing with arbitrary face type of wire-frame models in automatic face loop identification. However, this approach is difficult in determining the type of extracted faces.

The geometric approach is very straightforward by using the vector product of two adjacent straight edges to define an assumed face normal. However, the identification of a curved surface equation from the types of two adjacent edges is ill defined without severely restricting the scope of models that can be handled. In this approach, an exhaustive depth-first search method is usually used to find a close loop that is very expensive. In addition, it could also produce face loop with inner edge when the starting vertex was not a convex hull vertex [13].

This research proposes to develop the automatic capability of extraction all quadric faces from a wire-frame model. Unlike previous mentioned approaches; this method finds the quadric face loops that have no inner edges subdividing them and avoids finding the unwanted outer loops. It requires considerably less searching time than depth-first searches that could grow exponentially in complexity, and it simplifies the subsequent face type identification. As a quadric surface is traced, its surface type is uniquely defined by two adjacent edges.

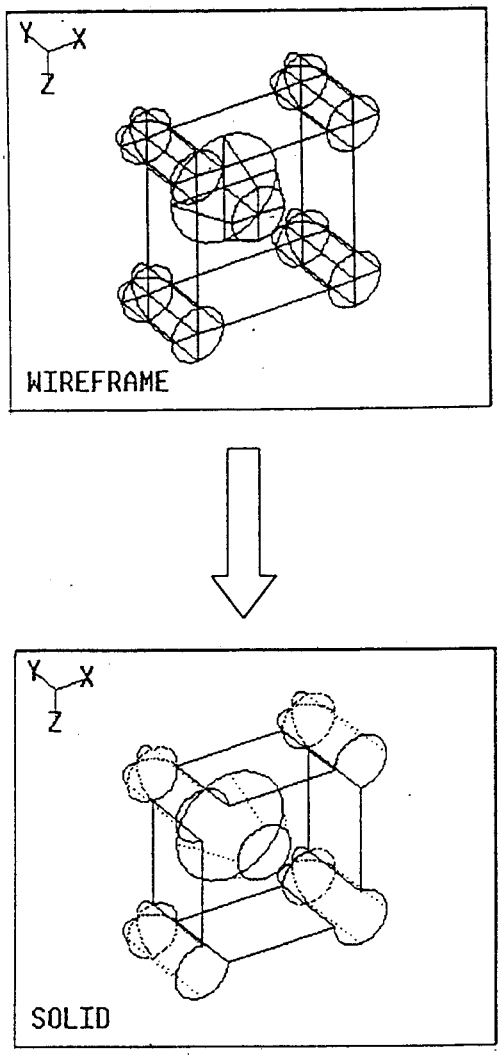


Figure 4. An object with planar, spherical, conical and cylindrical surfaces

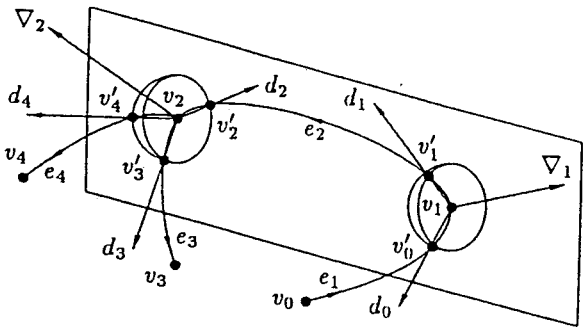


Figure 1. The MIA method

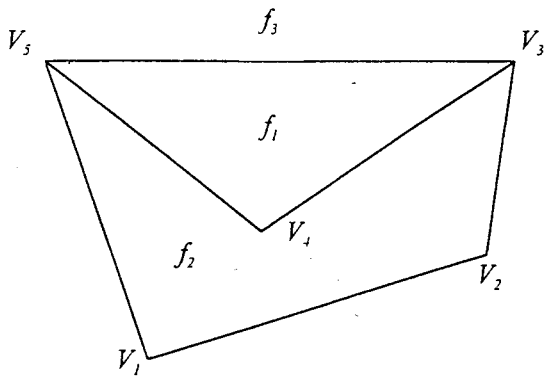
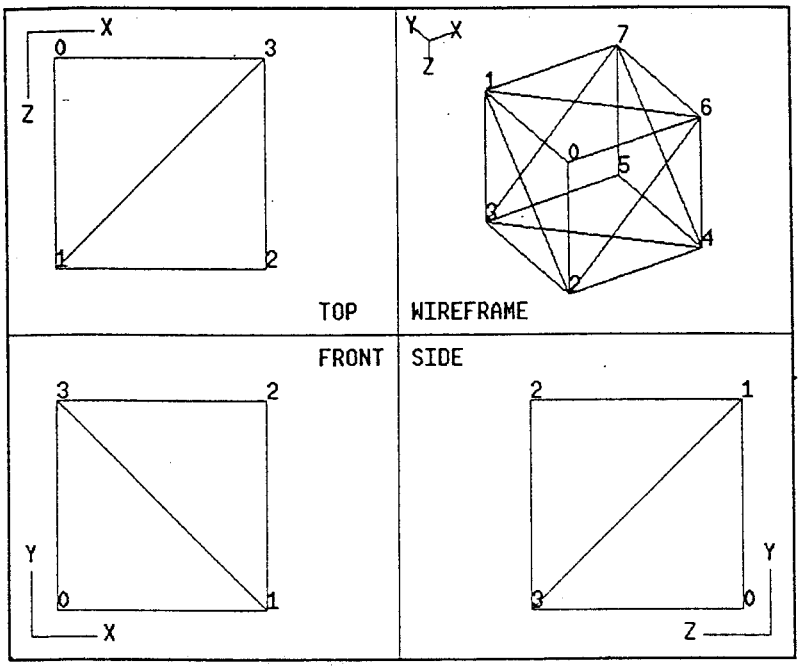


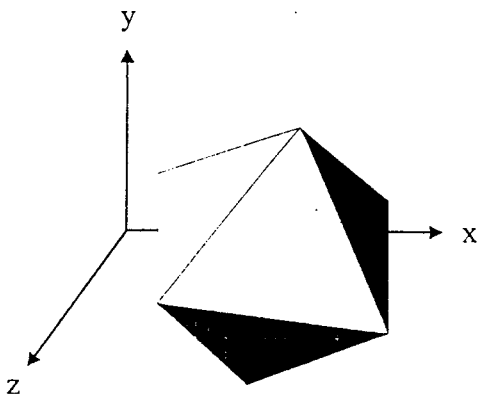
Figure 2. Six coplanar edges form three face loops



(a)

```
[U]E 0:( 0- 1)-f0-f1
[U]E 1:( 0- 2)-f0-f2
[U]E 2:( 0- 6)-f1-f2
[U]E 3:( 1- 2)-f0-f3-f4-f5
[U]E 4:( 1- 3)-f3-f6-f7
[U]E 5:( 1- 6)-f1-f4-f6-f8
[U]E 6:( 1- 7)-f5-f7-f8
[U]E 7:( 2- 3)-f3-f9-f10
[U]E 8:( 2- 4)-f5-f9-f11
[U]E 9:( 2- 6)-f2-f4-f10-f11
[U]E10:( 3- 4)-f6-f9-f12-f13
[U]E11:( 3- 5)-f12-f14
[U]E12:( 3- 7)-f7-f10-f13-f14
[U]E13:( 4- 5)-f12-f15
[U]E14:( 4- 6)-f6-f11-f16
[U]E15:( 4- 7)-f5-f13-f15-f16
[U]E16:( 5- 7)-f14-f15
[U]E17:( 6- 7)-f8-f10-f16
```

(b)



(c)

Figure 3. (a) a 3D wire-frame model (b) 18 face loops are extracted (c) the surface shading solid with 7 true faces

Table 1: Geometry of quadric surfaces

Standard form / Surface type	Central Quadric $\alpha x^2+\beta y^2+\gamma z^2+\kappa=0$	Non-central Quadric $\alpha x^2+\beta y^2+\gamma z=0$
Plane	$\alpha=\beta=\gamma=0$	
Ellipsoid*	$\alpha\neq\beta\neq\gamma>0,\ \kappa>0$	
Cylinder	$\alpha=\beta>0,\gamma=0,\kappa>0$ or $\beta=\gamma>0,\alpha=0,\kappa>0$ or $\gamma=\alpha>0,\beta=0,\kappa>0$	
Cone	$\alpha=\beta>0,\gamma<0,\kappa=0$ or $\beta=\gamma>0,\alpha<0,\kappa=0$ or $\gamma=\alpha>0,\beta<0,\kappa=0$	
Hyperbola	$\alpha,\beta>0,\ \gamma<0$ or $\beta,\gamma>0,\ \alpha<0$ or $\alpha,\gamma>0,\ \beta<0$	
Elliptic paraboloid		$\alpha,\beta>0$
Hyperbolic paraboloid		$\alpha<0,\ \beta>0$
Parabolic cylinder		$\alpha=0$ or $\beta=0$

*Sphere is a special case of ellipsoid

REFERENCE

[1] S. C. Agarwal, W. N. Waggenspack Jr, "Decomposition method for extracting face topologies from wire-frame models," *Computer-Aided Design*, Vol. 24, No. 3, pp.123-140, 1992

[2] P. M. Hanrahan, "Creating volume models from edge-vertex graphs," *ACM Computer Graphics*, Vol. 16, No. 3, pp. 77-84,1982

[3] R. D. Dutton, R. C. Brigham, "Efficiency identifying the faces of a solid," *Computers and Graphics in Mechanical Engineering*, Vol. 7, No. 2, pp. 143-147, 1983

[4] J. Hopcroft, R. Tarjan, "Efficient planarity testing," *J. Assoc. Computing Machinery*, Vol. 21, pp. 549-568, 1974

[5] M. A. Ganter, J. J. Uicjer, "From wire-frame to solid geometric: automated conversion of data representation," Computer in Mechanical Engineering, Vol. 2, No. 2, pp. 40-45, 1983

[6] S. M. Courter, J. A. Brewer, "Automated conversion of curvilinear wire-frame models to surface boundary models; a topological approach," *ACM, SIGGRAPH'86*, Vol. 20, No. 4, pp.171-178,1986

[7] J. S. Hojnicki, P. R. White, "Converting CAD wire-frame data to surfaced representation," *Computers in Mechanical Engineering*, pp. 19-25, 1988

[8] K. Paton, "An algorithm for finding a fundamental set of cycles of a graph," Comm. ACM, Vol. 12, No. 9, pp. 514-518, 1969

[9] U. G. Gujar, I. V. Nagendra, "Construction of 3D solid objects from orthographic views," *Comput. & Graphics*, Vol. 13, No.4, pp. 1-18, 1989

[10] Kuo-Chin Fan, Chia-Yuan Chang, "Surface extracting from line drawings of a polyhedron, *Pattern Recognition Letters*, Vol. 12, pp. 627-633,1991

[11] Kaining Gu, Zesheng Tang, Jiaguang Sun, "Reconstruction of 3D objects from orthographic projections," *Computer Graphics Forum*, Vol. 10, No. 5, pp. 317-324, 1986

[12] Remi Lequette, "Automatic construction of curvilinear solids from wire-frame views," *Computer-Aided Design*, Vol. 20, No. 4, pp.171-179,1988

[13] R. E. Marston, M. H. Kuo, "Efficiently identifying the faces of a wire-frame reconstructed from 3-view engineering drawings," SDAIR'95, pp. 249-259, USA, 1995

[14] David F. Rogers, J. Alan Adams, *Mathematical elements for computer graphics*, McGraw-Hill, Inc., 1990

[15] Paul S. Heckbert, *Graphics Gems IV*, Academic press, Inc., 1994

[16] R. E. Marston and M. H. Kuo, "Reconstruction of 3D objects from three orthographic projections using a decision-chaining method," *IAPR MVA'94*, pp.423-426, Japan, 1994

[17] M. H. Kuo, *Reconstruction of quadric surface solids from three orthographic views*, PhD thesis, Department of Computer Science, University of Nottingham, UK, 1996.