Tanmyat AL-Rafidain 84(28) 2006
P.P [9 - 21]

Geometrical Modeling Using Hidden Line Algorithm

Abdul-Aziz S. Khalil
Ass. Lecturer

Rawaa P. Polos
Ass. Lecturer

Nadia T. Saleh
Ass. Lecturer
Department of Computer Science
College of Computers and Mathematics Sciences
University of Mosul

ABSTRACT

Hidden line algorithms generate realistic scenes by identifying and removing parts of the displayed image or shape that are not visible from a chosen view position.

So, in order to achieve highly realistic view, Hidden Line Package (HLP) has been developed to draw many shapes of three-dimensional object using Depth-Sort hidden line algorithm.

This program uses one of the projection methods, either the perspective or the parallel method, to draw many types of three-dimensional shapes and then apply any transformation techniques (scaling, rotation, or translation), in addition, the program draws the shapes in transparent or solid modes. The resultant shape can be stored as an image file of BMP extension.

The software has been written by the authors using Microsoft Visual Basic 6.0.

نمذجة هندسية باستخدام خوارزميات الخط المخفي

عبدالعزيز سليمان خليل
مدرس مساعد- قسم علوم الحاسبات
كلية علوم الحاسبات والرياضيات- جامعة الموصل

رواء بطرس بولص
مدرس مساعد- قسم علوم الحاسبات
كلية علوم الحاسبات والرياضيات- جامعة الموصل

نادية طارق صالح
مدرس مساعد- قسم علوم الحاسبات
كلية علوم الحاسبات والرياضيات- جامعة الموصل

المستخلص

تُنتج خوارزميات الخط المخفي عرضاً واقعياً عن طريق تعريف وإلغاء أجزاء من الصورة أو الشكل المعروض والتي يكون من غير الممكن رؤيتها من زاوية النظر المختارة.

وبناءً عليه ومن أجل الحصول على رسوم ذات واقعية عالية فقد تم تطوير برنامج HLP **لرسم عدة أشكال ثلاثية الأبعاد باستخدام إحدى خوارزميات الخط المخفي وهي طريقة الترتيب بعمق.**

يستعمل هذا البرنامج إحدى طرق الإسقاط، إما المنظوري أو المتوازي، وذلك لرسم أنواع متعددة من الأشكال ثلاثية الأبعاد ثم تطبيق إحدى طرق التحويل، بالإضافة إلى رسم هذه الأشكال بأحد النمطين، الشفاف أو المعتم. كذلك يتيح البرنامج إمكانية خزن الرسم الناتج على شكل ملف صوري ذو امتداد BMP.

تم كتابة البرنامج باستخدام لغة فيجوال بيسك 6.0.

تاريخ التسلم 2005/6/26 ________________________________ **تاريخ قبول النشر2006/3/28**

Introduction

Many three-dimensional objects have surfaces that are planer polygons, for example, cubes, pyramids, primes, and so on. More complex objects such as houses are often built from these. So, if there is an ability to define an object or a scene using the planer surface, the displaying algorithm can take advantage of this fact and produce the display by just drawing the edges of the described polygons and removing all parts of the object that are not visible from a chosen viewing position [Foley, Dam V., Feiner, Hughes, 1996; Gauthier J., 2001].

Polygon tables:

Once a user has defined each polygon surface, input data must be organized into tables that will be used in processing and displaying the object. These data tables contain the geometrical representation of the object and are organized to facilitate processing. Geometrical data tables contain boundary coordinates and parameters to identify the spatial orientation of the polygon surfaces [Gauthier J., 2001; Hearn D., M. Baker, 1997].

A convenient method for storing coordinate information is by creating two lists: one for vertex table and the other for surface table. The vertex table contains coordinate values for each vertex in the object, flag for checking if the vertex has been selected as the far point during the hidden processing, and the distance of that vertex from the view plane see table (1). While the surface table would contain a list of its vertices and a flag for checking if the surface was drawn see table (2). This scheme is illustrated in figure (1)[Hearn D., M. Baker, 1997; Langbein F., 2001].

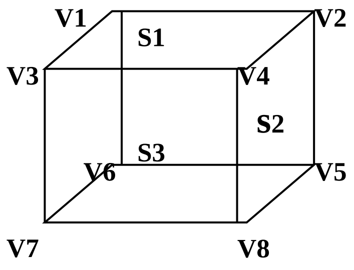


Figure 1
Polygon Geometrical Representation

Table 1
Vertex Table

Vertex	**Flag**	**Distance**
V1:x1,y1,z1	F=0	Dist=zv1
V2:x2,y2,z2	F=0	Dist=zv2
V3:x3,y3,z3	F=0	Dist=zv3
V4:x4,y4,z4	F=0	Dist=zv4
V5:x5,y5,z5	F=0	Dist=zv5
V6:x6,y6,z6	F=0	Dist=zv6
V7:x7,y7,z7	F=0	Dist=zv7
V8:x8,y8,z8	F=0	Dist=zv8

Table 2
Surface Table

Surface	**Flag**
S1: v1, v2, v3, v4	F=0
S2: v2, v4, v8, v5	F=0
S3: v3, v4, v8, v7	F=0
S4: v5, v6, v7, v8	F=0
S5: v1, v2, v5, v6	F=0
S6: v1, v3, v7, v6	F=0

Three-dimensional transformation:

Each Cartesian three-dimensional coordinate (x, y, z) can be represented with homogeneous coordinate quadruple [x, y, z, 1].

Basic three-dimensional transformations are represented with 4 by 4 transformation matrix [Deltenei J., 2000; Foley, Dam V., Feiner, Hughes, 1996].

- **Translation** :

$$[X`, Y`, Z`, 1]=[X, Y, Z, 1] \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ Tx & Ty & Tz & 0 \end{pmatrix} \qquad (1)$$

where Tx, Ty, Tz are the translated vector [Deltenei J., 2000; Foley, Dam V., Feiner, Hughes, 1996]. Figure (2) shows 3-D translation operation.

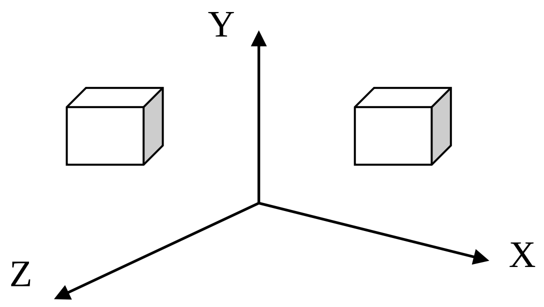


Figure 2
3-D Translation

- **Scaling**:

$$[X`, Y`, Z`, 1]=[X, Y, Z, 1]\begin{pmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (2)$$

where Sx scales objects in X direction, Sy scales objects in Y direction, and Sz scales objects in Z direction. Figure (3) shows 3-D Scaling operation.

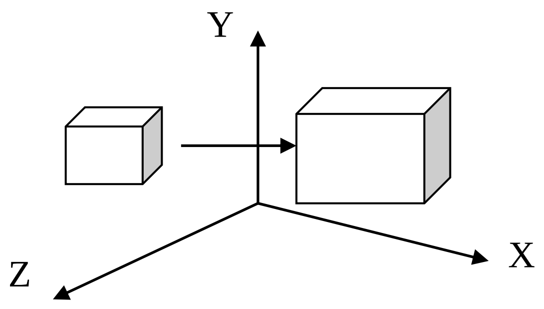


Figure 3
3-D Scaling

- **Rotation**:
  Rotation transformation for 3-D object requires designating an axis of rotation (about which the object is to be rotated) and the amount of angular rotation.

1. Rotation about X-axis. Figure (4) shows 3-D Rotation about X-axis operation.

$$[X`, Y`, Z`, 1]=[X, Y, Z, 1] \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

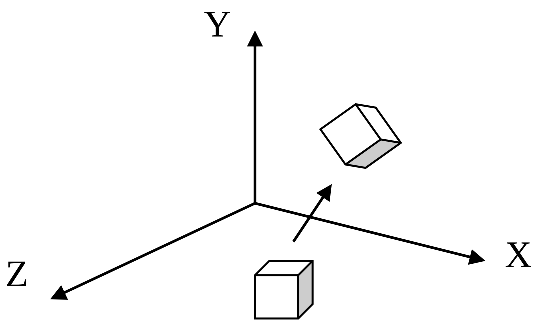


Figure 4
3-D Rotation about X-axis

2. Rotation about Y-axis. Figure (5) shows 3-D Rotation about Y-axis operation.

$$[X`, Y`, Z`, 1]=[X, Y, Z, 1] \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

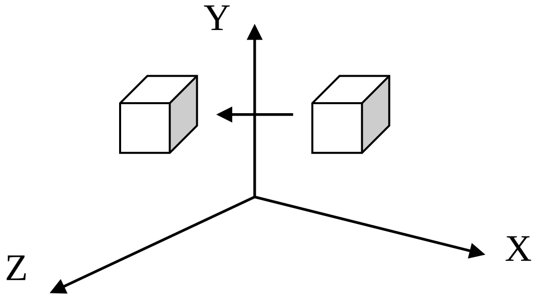


Figure 5
3-D Rotation about Y-axis

3. Rotation about Z-axis. Figure (6) shows 3-D Rotation about Z-axis operation :

$$[X`, Y`, Z`, 1]=[X, Y, Z, 1] \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ -\sin(\theta) & 1 & \cos(\theta) & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

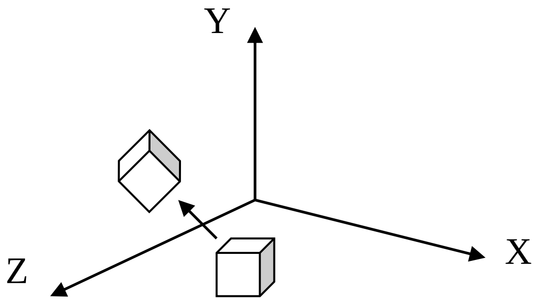


Figure 6
3-D Rotation about Z-axis

Projection:

Any 3-D object can be projected into 2-dimensional view plane using one of the following methods [Angell I., Griffith G., 1988; Hearn D., M. Baker, 1997]:

1. Parallel Projection:

Coordinate positions for an object are transformed to the view plane along parallel line. Parallel projection preserves relative proportion of object, but this does not give a realistic representation of the appearance of a 3-D object [3.Foley, Dam V., Feiner, Hughes, 1996; Langbein F., 2001].

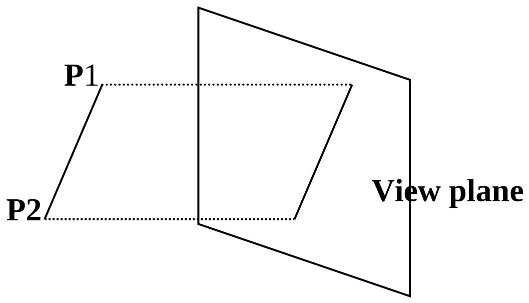


Figure 7
Parallel projection

(a) Orthogonal Parallel Projection:

When the projection line is perpendicular to the view plane.

The transformation equations for three dimensional point (x, y, z) are [Angell I., Griffith G., 1988; Deltenei J., 2000]:

Xp = x, Yp = y, Zp = 0.

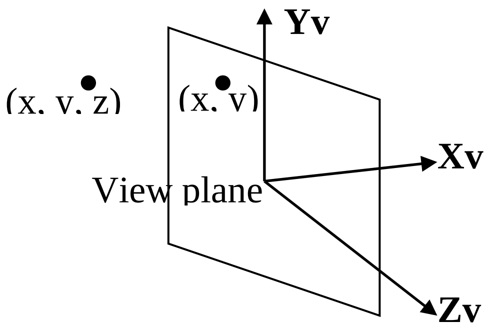


Figure 8
Orthogonal Parallel Projection

(b) Oblique Parallel Projection:

This type of projection can be obtained by projecting points along parallel lines that are not perpendicular to the projection plane [Deltenei J., 2000; Foley, Dam V., Feiner, Hughes, 1996]. An oblique vector is specified by two angle Ø and α where:

$$\left.\begin{array}{l} Xp = x + L\cos(Ø) \\ Yp = y + L\sin(Ø) \end{array}\right\} \quad (1)$$

$$\tan(\alpha) = z/L \quad (2)$$

$$\left.\begin{array}{l} L = z/\tan(\alpha) = zL1 \\ \text{L1 is the inverse of } \tan(\alpha). \\ Xp = x + z\,(L1\cos(Ø)) \quad \text{where } L1 = 1/\tan(\alpha) \end{array}\right\} \quad (3)$$

$$Yp = y + z\,(L1\sin(Ø)) \quad (4)$$

$$Zp = 0$$

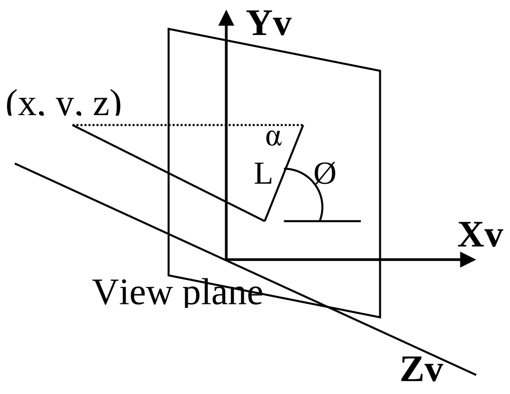


Figure 9
Oblique Parallel Projection

2. Perspective Projection:

Projection objects by this method can be obtained by transforming points along projection lines that meet at the center of projection [Deltenei J., 2000; Hearn D., M. Baker, 1997].

Perspective projection produce realistic views of the displayed objects but does not preserve relative proportion, so it changes the size of objects. Projections of distant objects are smaller than the projection of objects of the same size that are closer to the projection plane [1.Angell I., Griffith G., 1988; Hearn D., M. Baker, 1997].

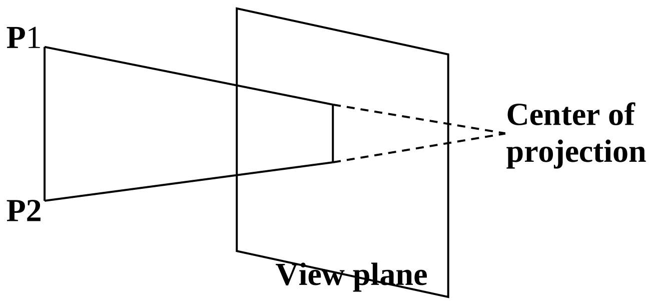


Figure 10
Perspective projection

The transformation equations are:

$$\left.\begin{aligned} x1 &= x - x \times u \\ y1 &= y - y \times u \\ z1 &= z - (z + d) \times u \\ 0 &<= u <= 1 \end{aligned}\right\} \quad (5)$$

$$u = z / (z + d) \quad (6)$$

$$\left.\begin{aligned} Xp &= x \times (d / (z + d)) \\ Yp &= y \times (d / (z + d)) \\ Zp &= 0 \end{aligned}\right\} \quad (7)$$

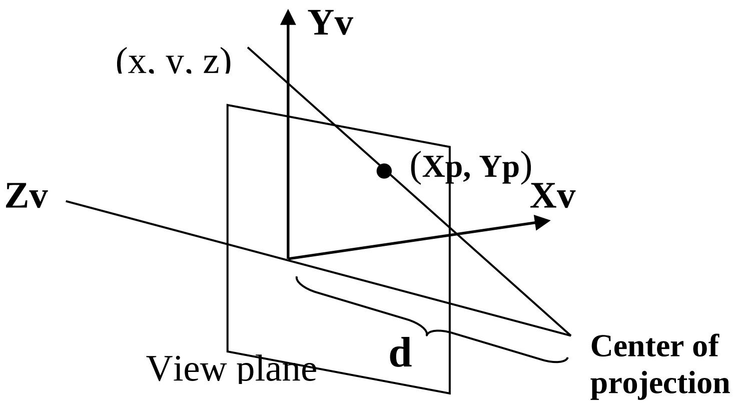


Figure 11
Perspective projection

Hidden Line algorithm:

The method that has been used for removing hidden lines depends on finding which surface lies behind other surface, so it requires to identify those surfaces that point away from the viewer and therefore will not seen and will be drawn first [Angell I., Griffith G., 1988; Nichewo, 2003]. This method is called Depth-sorting method (painter's algorithm) because polygons are drawn as an oil painter, the farthest one first [P. A. Egerton, W. S. Hall, 1998].

It is difficult and requires considerable computation to distinguish between visible and invisible surfaces of an object.

So, hidden line removal includes the following steps:

1. Perform checking process to find the furthest vertex with large distance from view plan, and then set the flag of that vertex to avoid choosing it again in the next time.
2. Find those surfaces in which these vertices belong to them then apply the selected transformation method (if any) and finally draw these surfaces using one of the projection methods, then set the flag of that surface.
3. Repeat steps 1 & 2 until the whole object has drawn.

Program review:

HLP (Hidden Line Program) program includes many forms, these forms are linked together. Starting from the first form which represents program main interface as shown in figure (12). This form contains:

- Name of shapes that one of them be chosen to be drawn later. These shapes are:

1.Cube
2.Cylinder
3.Pyramid
4.Cone
5.Tedrahedron

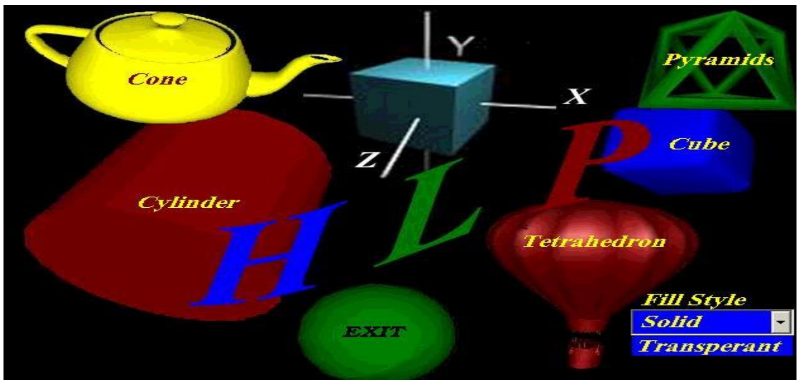
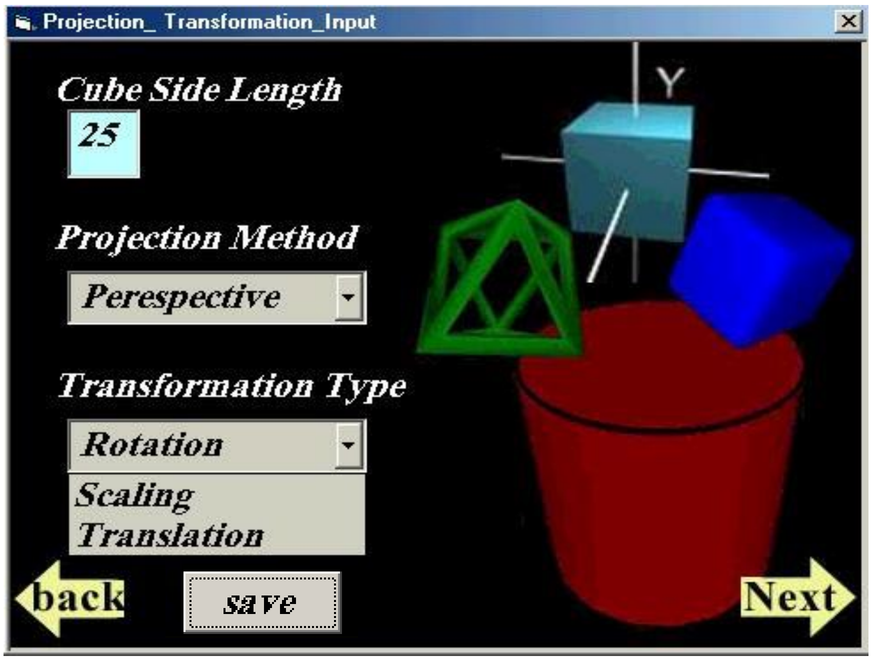


Figure 12
Program Main screen

- Fill Style: is a combo box that can be used to select one of the fill style type, either solid or transparency. First fill style must be selected and then select the shape to display the next form, which is called, Projection-Transformation-Input shown in figure (13).

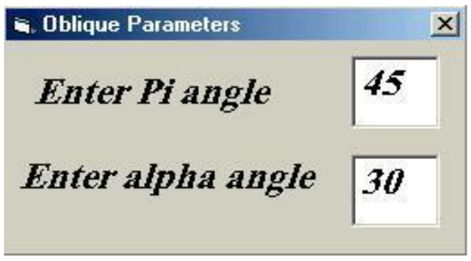


Cube Side Length: 25
Projection Method: Perespective
Transformation Type: Rotation
Scaling
Translation

Figure 13
Projection and Transformation form

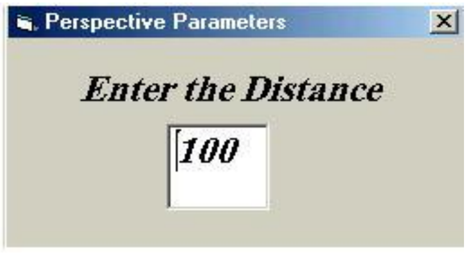
This form includes the following tools:

- Dimensions that will be needed to draw the shape such as cube side length if cube was selected.
- Transformation: this tool is another combo box that contain the basic three-dimensional operations to be applied to the object to be drawn. These transformations are:
  1.Scaling
  2.Rotation
  3.Translation
- Projection Method: is a combo box that will be used to select the projection methods, either one of parallel projection methods (orthogonal or oblique) or perspective projection. After this selection, an input box will appeared to enter the parameters needed to apply the projection process depending on the selected method (except orthogonal method) as shown in figure 14.


Oblique Parameters

Enter Pi angle 45

Enter alpha angle 30


Perspective Parameters

Enter the Distance 100

Figure 14
(Oblique & Perspective) Projection parameter form

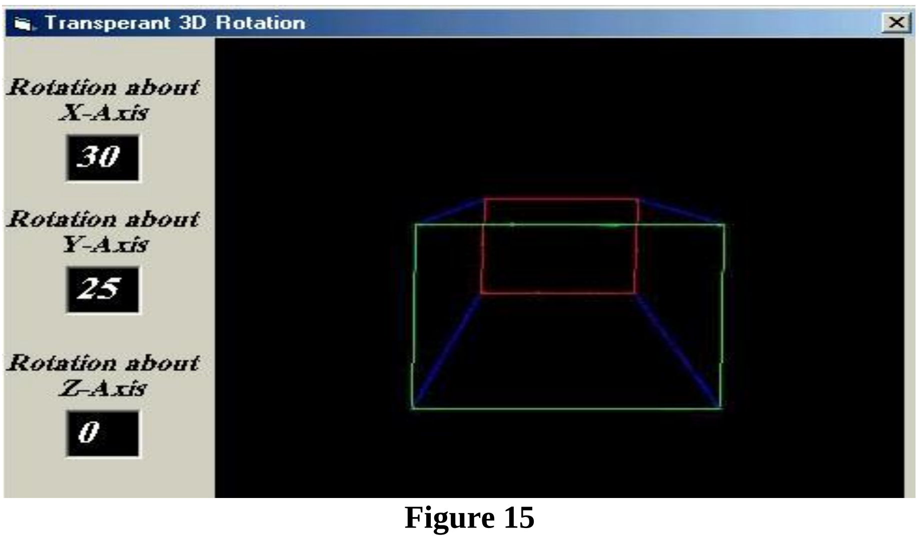
- Save: is a button used to save the resultant shape in BMP file by selecting path and name for the file when this button is pressed.

When the user selects any of the three transformations, next button must be pressed to display the final form that view selected shape drawn with the desired projection method as depicted in figure (15) if transparent fill style was selected and figure (16) if solid fill style was selected. In addition to that this form includes several text boxes which can be used to enter the parameters for the selected transformation operation as follows:

1.Scaling transformation needs Sx, Sy, Sz, which are the scaling values in three Cartesian axes.

2.Rotation transformation needs angle1, angle2, and angle3, which are the angles used to rotate the shape about the corresponding axis.

3.Translation transformation needs Tx, Ty, Tz, which is represented the translation vectors.


Transperant 3D Rotation

Rotation about X-Axis 30

Rotation about Y-Axis 25

Rotation about Z-Axis 0

Figure 15
shape Transparent Display Form

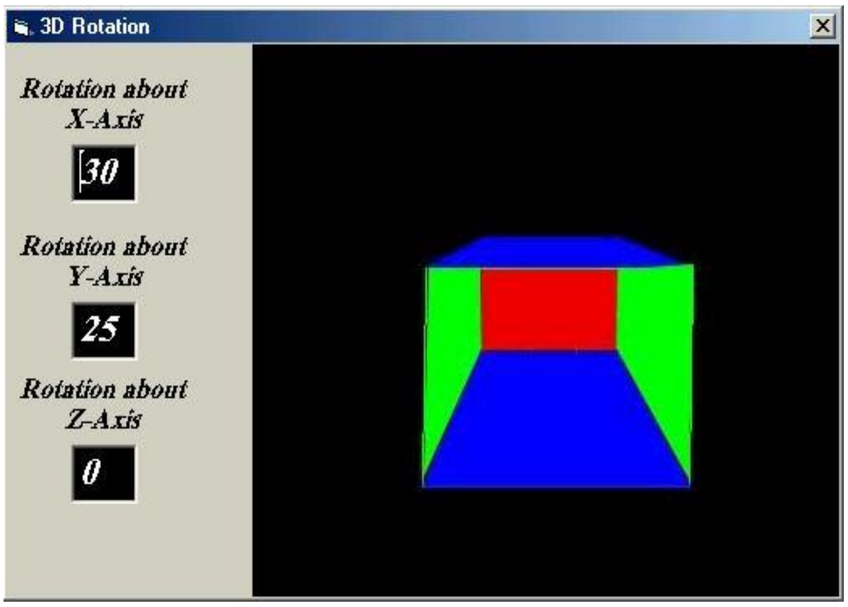


Figure 16
Shape Solid Display Form

Conclusions:

Hidden line algorithm really deserves special attention because it presents an important method for drawing many types of shape and generating object view as realistic as possible. This algorithm avoids the ambiguity drafting when display three-dimensional objects without removing the hidden lines. It is one of the efficient methods and gives superior time utilization but requires a suitable amount of memory to store the related shapes' tables that are used through the algorithm processing steps.

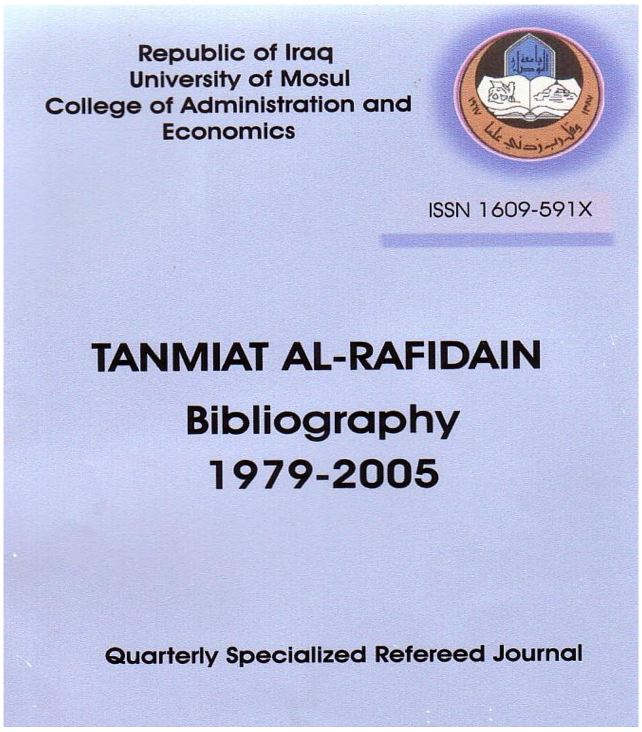
References:

1. Angell I., Griffith G., 1988, "High-resolution Computer graphics Using Pascal", Macmillan Education LTD.
2. Deltenei J., 2000, Inverse Reality, http://www.inversereality.org/tutorials/ graphics %20 programming / graphics programming.html.
3. Foley, Dam V., Feiner, Hughes, 1996, " Computer Graphics Principles and Practice", 2nd Edition in C, Addison-Wesley publishing company.
4. Gauthier J., 2001, Interactive 3D, http:// Interactive 3d/ Beginning Thoughts.html.
5. Hearn D., M. Baker, 1997, "Computer Graphics", Prentice-Hall International.
6. Langbein F., 2001, "Graphics Programming", 1st Edition, Cardiff University.
7. Nichewo, 2003, Graphics Languages Tutorial (guide to the modern CG Programming), http://www.nichewo.com/gltut.
8. P. A. Egerton, W. S. Hall, 1998, "Computer Graphics and Mathematical First Step", 1st Edition, Prentie Hall Europe.

Advertisement

Tanmiat AL-Rafidain journal will issue soon the bibliography of the published papers. This includes (V.1) (1979) to (V.28) (2006). The alphabetical and chronological order of the researchers will be regarded.

It is available for all institutions and academies.


Republic of Iraq
University of Mosul
College of Administration and Economics

ISSN 1609-591X

TANMIAT AL-RAFIDAIN
Bibliography
1979-2005

Quarterly Specialized Refereed Journal