CHAPTER 5

Two-Dimensional Viewing and Clipping

Much like what we see in real life through a small window on the wall or the viewfinder of a camera, a computer-generated image often depicts a partial view of a large scene. Objects are placed into the scene by modeling transformations to a master coordinate system, commonly referred to as the *world coordinate system* (WCS). A rectangular *window* with its edges parallel to the axes of the WCS is used to select the portion of the scene for which an image is to be generated (see Fig. 5-1). Sometimes an additional coordinate system called the *viewing coordinate system* is introduced to simulate the effect of moving and/or tilting the camera.

On the other hand, an image representing a view often becomes part of a larger image, like a photo on an album page, which models a computer monitor's display area. Since album pages vary and monitor sizes differ from one system to another, we want to introduce a device-independent tool to describe the display area. This tool is called the *normalized device coordinate system* (NDCS) in which a unit (1 × 1) square whose lower left corner is at the origin of the coordinate system defines the display area of a virtual display device. A rectangular *viewport* with its edges parallel to the axes of the NDCS is used to specify a sub-region of the display area that embodies the image.

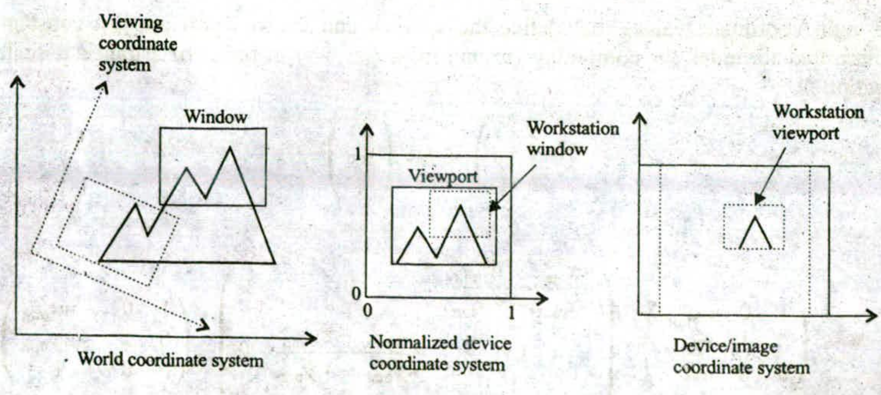


Fig. 5-1 Viewing transformation.

The process that converts object coordinates in WCS to normalized device coordinates is called *window-to-viewport mapping* or *normalization transformation*, which is the subject of Sect. 5.1. The process that maps normalized device coordinates to discrete device/image coordinates is called *workstation transformation*, which is essentially a second window-to-viewport mapping, with a workstation window in the normalized device coordinate system and a workstation viewport in the device coordinate system. Collectively, these two coordinate-mapping operations are referred to as *viewing transformation*.

Workstation transformation is dependent on the resolution of the display device/frame buffer. When the whole display area of the virtual device is mapped to a physical device that does not have a 1/1 aspect ratio, it may be mapped to a square sub-region (see Fig. 5-1) so as to avoid introducing unwanted geometric distortion.

Along with the convenience and flexibility of using a window to specify a localized view comes the need for *clipping*, since objects in the scene may be completely inside the window, completely outside the window, or partially visible through the window (e.g. the mountain-like polygon in Fig. 5-1). The clipping operation eliminates objects or portions of objects that are not visible through the window to ensure the proper construction of the corresponding image.

Note that clipping may occur in the world coordinate or viewing coordinate space, where the window is used to clip the objects; it may also occur in the normalized device coordinate space, where the viewport/workstation window is used to clip. In either case we refer to the window or the viewport/workstation window as the *clipping window*. We discuss point clipping, line clipping, and polygon clipping in Secs. 5.2, 5.3, and 5.4, respectively.

5.1 WINDOW-TO-VIEWPORT MAPPING

A window is specified by four world coordinates: $wx_{\min}$, $wx_{\max}$, $wy_{\min}$, and $wy_{\max}$ (see Fig. 5-2). Similarly, a viewport is described by four normalized device coordinates: $vx_{\min}$, $vx_{\max}$, $vy_{\min}$, and $vy_{\max}$. The objective of window-to-viewport mapping is to convert the world coordinates (wx, wy) of an arbitrary point to its corresponding normalized device coordinates (vx, vy). In order to maintain the same relative placement of the point in the viewport as in the window, we require:

$$\frac{wx - wx_{\min}}{wx_{\max} - wx_{\min}} = \frac{vx - vx_{\min}}{vx_{\max} - vx_{\min}} \quad \text{and} \quad \frac{wy - wy_{\min}}{wy_{\max} - wy_{\min}} = \frac{vy - vy_{\min}}{vy_{\max} - vy_{\min}}$$

Thus

$$\begin{cases} vx = \dfrac{vx_{\max} - vx_{\min}}{wx_{\max} - wx_{\min}}(wx - wx_{\min}) + vx_{\min} \\[2ex] vy = \dfrac{vy_{\max} - vy_{\min}}{wy_{\max} - wy_{\min}}(wy - wy_{\min}) + vy_{\min} \end{cases}$$

Since the eight coordinate values that define the window and the viewport are just constants, we can express these two formulas for computing (vx, vy) from (wx, wy) in terms of a translate–scale–translate transformation N

$$\begin{pmatrix} vx \\ vy \\ 1 \end{pmatrix} = N \cdot \begin{pmatrix} wx \\ wy \\ 1 \end{pmatrix}$$

where

$$N = \begin{pmatrix} 1 & 0 & vx_{\min} \\ 0 & 1 & vy_{\min} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \dfrac{vx_{\max} - vx_{\min}}{wx_{\max} - wx_{\min}} & 0 & 0 \\ 0 & \dfrac{vy_{\max} - vy_{\min}}{wy_{\max} - wy_{\min}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -wx_{\min} \\ 0 & 1 & -wy_{\min} \\ 0 & 0 & 1 \end{pmatrix}$$

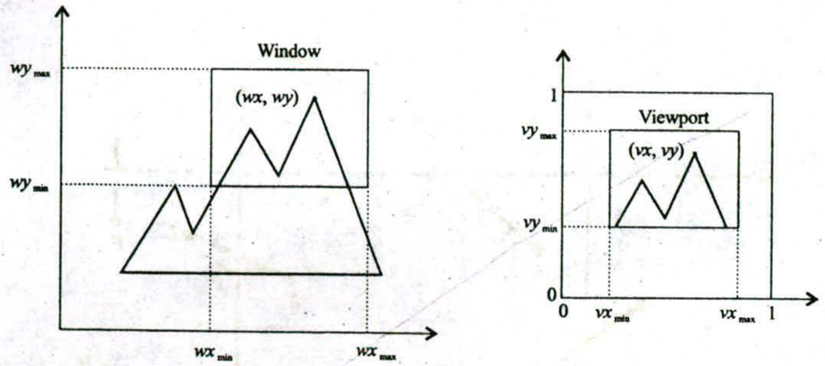


Fig. 5-2 Window-to-viewport mapping.

Note that geometric distortions occur (e.g. squares in the window become rectangles in the viewport) whenever the two scaling constants differ.

5.2 POINT CLIPPING

Point clipping is essentially the evaluation of the following inequalities:

$$x_{min} \leq x \leq x_{max} \quad \text{and} \quad y_{min} \leq y \leq y_{max}$$

where x_{min}, x_{max}, y_{min} and y_{max} define the clipping window. A point (x, y) is considered inside the window when the inequalities all evaluate to true.

5.3 LINE CLIPPING

Lines that do not intersect the clipping window are either completely inside the window or completely outside the window. On the other hand, a line that intersects the clipping window is divided by the intersection point(s) into segments that are either inside or outside the window. The following algorithms provide efficient ways to decide the spatial relationship between an arbitrary line and the clipping window and to find intersection point(s).

The Cohen–Sutherland Algorithm

In this algorithm we divide the line clipping process into two phases: (1) identify those lines which intersect the clipping window and so need to be clipped and (2) perform the clipping.

All lines fall into one of the following *clipping categories*:

1. *Visible*—both endpoints of the line lie within the window.
2. *Not visible*—the line definitely lies outside the window. This will occur if the line from (x_1, y_1) to (x_2, y_2) satisfies any one of the following four inequalities:

$$x_1, x_2 > x_{max} \qquad y_1, y_2 > y_{max}$$
$$x_1, x_2 < x_{min} \qquad y_1, y_2 < y_{min}$$

3. *Clipping candidate*—the line is in neither category 1 nor 2.

In Fig. 5-3, line *AB* is in category 1 (visible); lines *CD* and *EF* are in category 2 (not visible); and lines *GH*, *IJ*, and *KL* are in category 3 (clipping candidate).

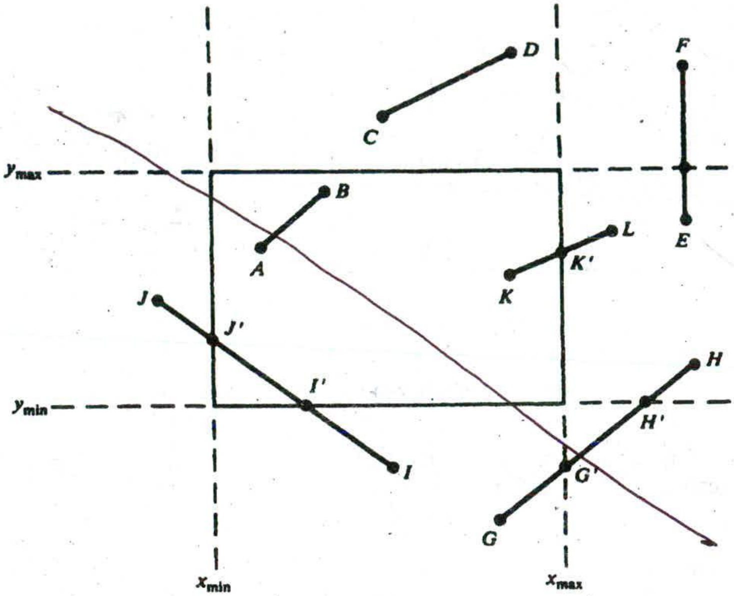
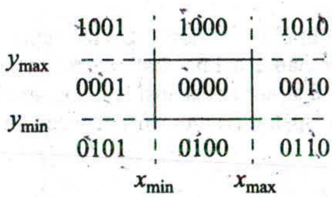


Fig. 5-3

The algorithm employs an efficient procedure for finding the category of a line. It proceeds in two steps:

1. Assign a 4-bit region code to each endpoint of the line. The code is determined according to which of the following nine regions of the plane the endpoint lies in



Starting from the leftmost bit, each bit of the code is set to true (1) or false (0) according to the scheme

Bit 1 ≡ endpoint is above the window = sign $(y - y_{max})$
Bit 2 ≡ endpoint is below the window = sign $(y_{min} - y)$
Bit 3 ≡ endpoint is to the right of the window = sign $(x - x_{max})$
Bit 4 ≡ endpoint is to the left of the window = sign $(x_{min} - x)$

We use the convention that sign$(a) = 1$ if a is positive, 0 otherwise. Of course, a point with code 0000 is inside the window.

2. The line is visible if both region codes are 0000, and not visible if the bitwise logical AND of the codes is not 0000, and a candidate for clipping if the bitwise logical AND of the region codes is 0000 (see Prob. 5.8).

For a line in category 3 we proceed to find the intersection point of the line with one of the boundaries of the clipping window, or to be exact, with the infinite extension of one of the boundaries (see Fig. 5-4). We choose an endpoint of the line, say (x_1, y_1), that is outside the window, i.e., whose region code is not

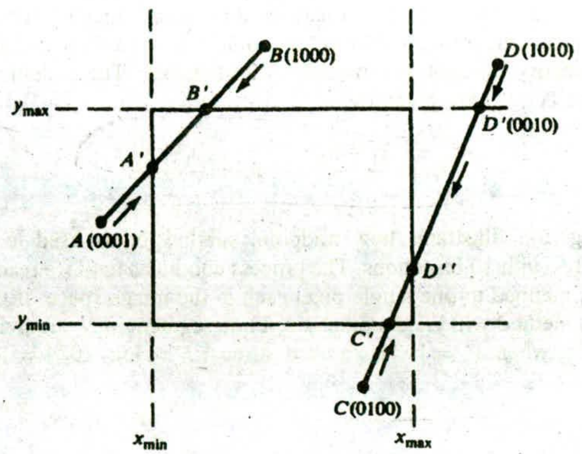


Fig. 5-4

0000. We then select an extended boundary line by observing that those boundary lines that are candidates for intersection are the ones for which the chosen endpoint must be "pushed across" so as to change a "1" in its code to a "0" (see Fig. 5-4). This means:

If bit 1 is 1, intersect with line $y = y_{max}$.
If bit 2 is 1, intersect with line $y = y_{min}$.
If bit 3 is 1, intersect with line $x = x_{max}$.
If bit 4 is 1, intersect with line $x = x_{min}$.

Consider line *CD* in Fig. 5-4. If endpoint *C* is chosen, then the bottom boundary line $y = y_{min}$ is selected for computing intersection. On the other hand, if endpoint *D* is chosen, then either the top boundary line $y = y_{max}$ or the right boundary line $x = x_{max}$ is used. The coordinates of the intersection point are

$$\begin{cases} x_i = x_{min} \text{ or } x_{max} \\ y_i = y_1 + m(x_i - x_1) \end{cases} \quad \text{if the boundary line is vertical}$$

or

$$\begin{cases} x_i = x_1 + (y_i - y_1)/m \\ y_i = y_{min} \text{ or } y_{max} \end{cases} \quad \text{if the boundary line is horizontal}$$

where $m = (y_2 - y_1)/(x_2 - x_1)$ is the slope of the line.

Now we replace endpoint (x_1, y_1) with the intersection point (x_i, y_i), effectively eliminating the portion of the original line that is on the outside of the selected window boundary. The new endpoint is then assigned an updated region code and the clipped line re-categorized and handled in the same way. This iterative process terminates when we finally reach a clipped line that belongs to either category 1 (visible) or category 2 (not visible).

Midpoint Subdivision

An alternative way to process a line in category 3 is based on binary search. The line is divided at its midpoint into two shorter line segments. The clipping categories of the two new line segments are then determined by their region codes. Each segment in category 3 is divided again into shorter segments and categorized. This bisection and categorization process continues until each line segment that spans across a window boundary (hence encompasses an intersection point) reaches a threshold for line size and all other segments are either in category 1 (visible) or in category 2 (invisible). The midpoint coordinates (x_m, y_m) of a line joining (x_1, y_1) and (x_2, y_2) are given by

$$x_m = \frac{x_1 + x_2}{2} \qquad y_m = \frac{y_1 + y_2}{2}$$

The example in Fig. 5-5 illustrates how midpoint subdivision is used to zoom in onto the two intersection points I_1 and I_2 with 10 bisections. The process continues until we reach two line segments that are, say, pixel-sized, i.e., mapped to one single pixel each in the image space. If the maximum number of pixels in a line is M, this method will yield a pixel-sized line segment in N subdivisions, where $2^N = M$ or $N = \log_2 M$. For instance, when $M = 1024$ we need at most $N = \log_2 1024 = 10$ subdivisions.

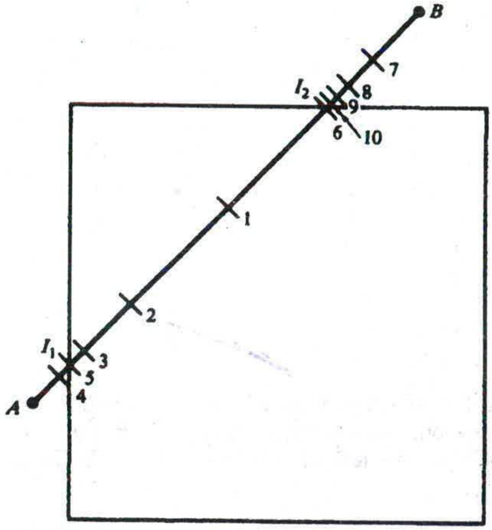


Fig. 5-5

The Liang–Barsky Algorithm

The following parametric equations represent a line from (x_1, y_1) to (x_2, y_2) along with its infinite extension:

$$\begin{cases} x = x_1 + \Delta x \cdot u \\ y = y_1 + \Delta y \cdot u \end{cases}$$

where $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$. The line itself corresponds to $0 \le u \le 1$ (see Fig. 5-6). Notice that when we traverse along the extended line with u increasing from $-\infty$ to $+\infty$, we first move from the outside to the inside of the clipping window's two boundary lines (bottom and left), and then move from the inside to the outside of the other two boundary lines (top and right). If we use u_1 and u_2, where $u_1 \le u_2$, to represent the beginning and end of the visible portion of the line, we have $u_1 = \text{maximum}(0, u_l, u_b)$ and $u_2 = \text{minimum}(1, u_t, u_r)$, where u_l, u_b, u_t, and u_r correspond to the intersection point of the extended line with the window's left, bottom, top, and right boundary, respectively.

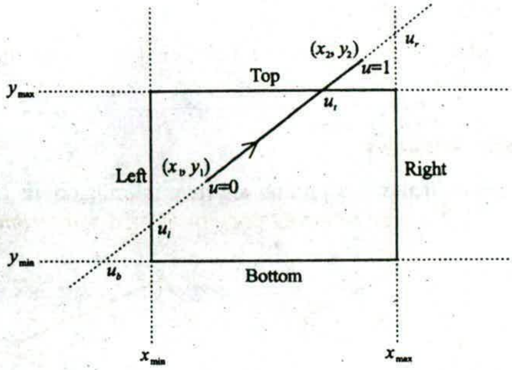


Fig. 5-6

Now consider the tools we need to turn this basic idea into an efficient algorithm. For point (x, y) inside the clipping window, we have

$$x_{\min} \leq x_1 + \Delta x \cdot u \leq x_{\max}$$
$$y_{\min} \leq y_1 + \Delta y \cdot u \leq y_{\max}$$

Rewrite the four inequalities as

$$p_k\, u \leq q_k, \qquad k = 1, 2, 3, 4$$

where

$$\begin{array}{lll} p_1 = -\Delta x & q_1 = x_1 - x_{\min} & \text{(left)} \\ p_2 = \Delta x & q_2 = x_{\max} - x_1 & \text{(right)} \\ p_3 = -\Delta y & q_3 = y_1 - y_{\min} & \text{(bottom)} \\ p_4 = \Delta y & q_4 = y_{\max} - y_1 & \text{(top)} \end{array}$$

Observe the following facts:

- if $p_k = 0$, the line is parallel to the corresponding boundary and

$$\begin{cases} \text{if } q_k < 0, & \text{the line is completely outside the boundary and can be eliminated} \\ \text{if } q_k \geq 0, & \text{the line is inside the boundary and needs further consideration,} \end{cases}$$

- if $p_k < 0$, the extended line proceeds from the outside to the inside of the corresponding boundary line,
- if $p_k > 0$, the extended line proceeds from the inside to the outside of the corresponding boundary line,
- when $p_k \neq 0$, the value of u that corresponds to the intersection point is q_k/p_k.

The Liang–Barsky algorithm for finding the visible portion of the line, if any, can be stated as a four-step process:

1. If $p_k = 0$ and $q_k < 0$ for any k, eliminate the line and stop. Otherwise proceed to the next step.
2. For all k such that $p_k < 0$, calculate $r_k = q_k/p_k$. Let u_1 be the maximum of the set containing 0 and the calculated r values.
3. For all k such that $p_k > 0$, calculate $r_k = q_k/p_k$. Let u_2 be the minimum of the set containing 1 and the calculated r values.
4. If $u_1 > u_2$, eliminate the line since it is completely outside the clipping window. Otherwise, use u_1 and u_2 to calculate the endpoints of the clipped line.

5.4 POLYGON CLIPPING

In this section we consider the case of using a polygonal clipping window to clip a polygon.

Convex Polygonal Clipping Windows

A polygon is called *convex* if the line joining any two interior points of the polygon lies completely inside the polygon (see Fig. 5-7). A non-convex polygon is said to be *concave*.

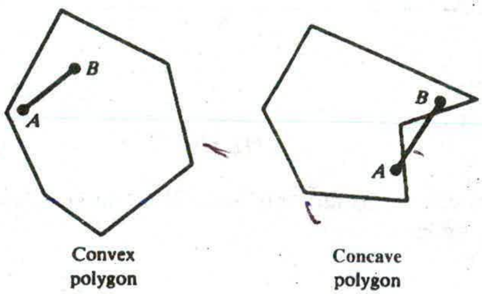
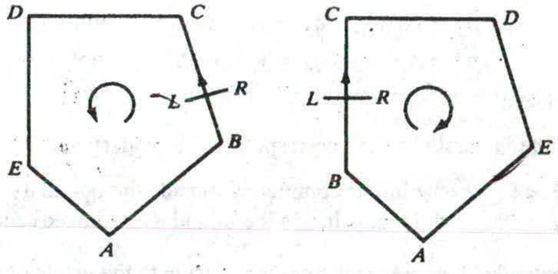


Fig. 5-7

By convention, a polygon with vertices $P_1, \ldots, P_N$ (and edges $P_{i-1}P_i$ and P_NP_1) is said to be *positively oriented* if a tour of the vertices in the given order produces a counterclockwise circuit.

Equivalently, the left hand of a person standing along any directed edge $\overline{P_{i-1}P_i}$ or $\overline{P_NP_1}$ would be pointing inside the polygon [see orientations in Figs. 5-8(*a*) and 5-8(*b*)].



(*a*) **Positive orientation.** (*b*) **Negative orientation.**

Fig. 5-8

Let $A(x_1, y_1)$ and $B(x_2, y_2)$ be the endpoints of a directed line segment. A point $P(x, y)$ will be to the *left* of the line segment if the expression $C = (x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1)$ is positive (see Prob. 5.13). We say that the point is to the *right* of the line segment if this quantity is negative. If a point P is to the right of any one edge of a positively oriented, convex polygon, it is outside the polygon. If it is to the left of *every* edge of the polygon, it is inside the polygon.

This observation forms the basis for clipping any polygon, convex or concave, against a convex polygonal clipping window.

The Sutherland–Hodgman Algorithm

Let $P_1, \ldots, P_N$ be the vertex list of the polygon to be clipped. Let edge E, determined by endpoints A and B, be any edge of the positively oriented, convex clipping polygon. We clip each edge of the polygon in turn against the edge E of the clipping polygon, forming a new polygon whose vertices are determined as follows.

Consider the edge $\overline{P_{i-1}P_i}$:

1. If both P_{i-1} and P_i are to the left of the edge, vertex P_i is placed on the *vertex output list* of the clipped polygon [Fig. 5-9(*a*)].
2. If both P_{i-1} and P_i are to the right of the edge, nothing is placed on the vertex output list [Fig. 5-9(*b*)].
3. If P_{i-1} is to the left and P_i is to the right of the edge E, the intersection point I of line segment $\overline{P_{i-1}P_i}$ with the extended edge E is calculated and placed on the vertex output list [Fig. 5-9(*c*)].
4. If P_{i-1} is to the right and P_i is to the left of edge E, the intersection point I of the line segment $\overline{P_{i-1}P_i}$ with the extended edge E is calculated. Both I and P_i are placed on the vertex output list [Fig. 5-9(*d*)].

The algorithm proceeds in stages by passing each clipped polygon to the next edge of the window and clipping. See Probs. 5.14 and 5.15.

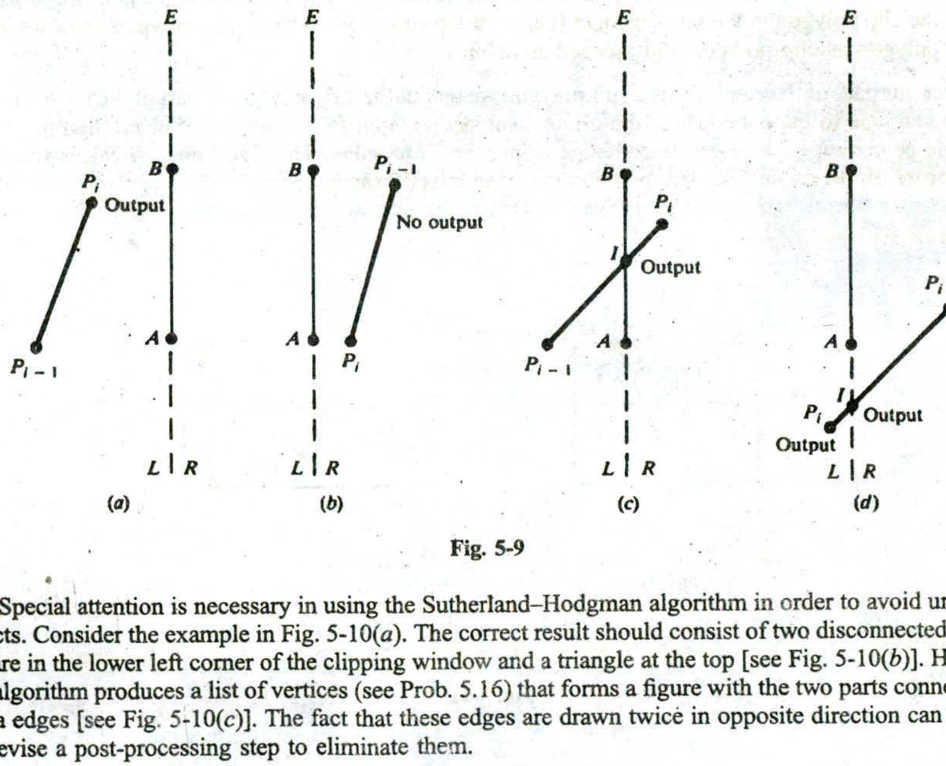


Fig. 5-9

Special attention is necessary in using the Sutherland–Hodgman algorithm in order to avoid unwanted effects. Consider the example in Fig. 5-10(*a*). The correct result should consist of two disconnected parts, a square in the lower left corner of the clipping window and a triangle at the top [see Fig. 5-10(*b*)]. However, the algorithm produces a list of vertices (see Prob. 5.16) that forms a figure with the two parts connected by extra edges [see Fig. 5-10(*c*)]. The fact that these edges are drawn twice in opposite direction can be used to devise a post-processing step to eliminate them.

The Weiler–Atherton Algorithm

Let the clipping window be initially called the clip polygon, and the polygon to be clipped the subject polygon [see Fig. 5-11(*a*)]. We start with an arbitrary vertex of the subject polygon and trace around its border in the clockwise direction until an intersection with the clip polygon is encountered:

- If the edge enters the clip polygon, record the intersection point and continue to trace the subject polygon.

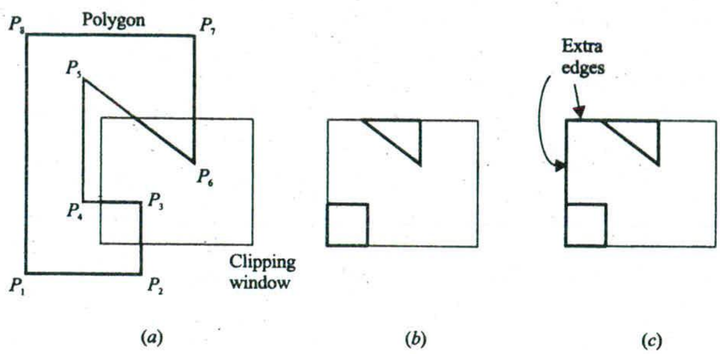


Fig. 5-10

- If the edge leaves the clip polygon, record the intersection point and make a right turn to follow the clip polygon in the same manner (i.e., treat the clip polygon as subject polygon and the subject polygon as clip polygon and proceed as before).

Whenever our path of traversal forms a sub-polygon we output the sub-polygon as part of the overall result. We then continue to trace the rest of the original subject polygon from a recorded intersection point that marks the beginning of a not-yet-traced edge or portion of an edge. The algorithm terminates when the entire border of the original subject polygon has been traced exactly once.

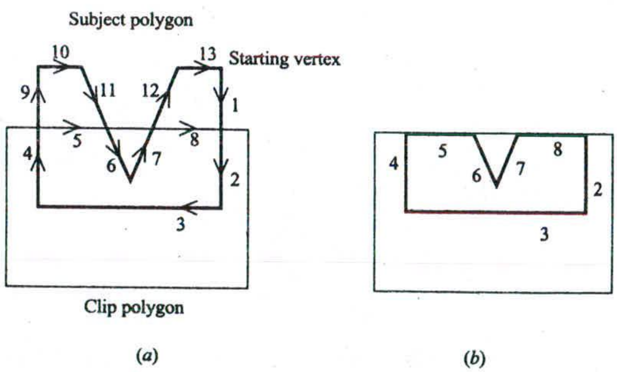
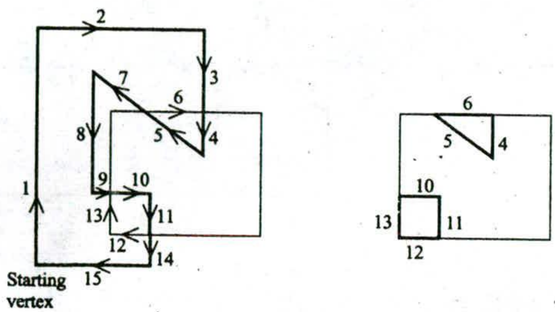


Fig. 5-11

For example, the numbers in Fig. 5-11(a) indicate the order in which the edges and portions of edges are traced. We begin at the starting vertex and continue along the same edge (from 1 to 2) of the subject polygon as it enters the clip polygon. As we move along the edge that is leaving the clip polygon we make a right turn (from 4 to 5) onto the clip polygon, which is now considered the subject polygon. Following the same logic leads to the next right turn (from 5 to 6) onto the current clip polygon, which is really the original subject polygon. With the next step done (from 7 to 8) in the same way we have a sub-polygon for output [see Fig. 5-11(b)]. We then resume our traversal of the original subject polygon from the recorded intersection point where we first changed our course. Going from 9 to 10 to 11 produces no output. After skipping the already-traversed 6 and 7, we continue with 12 and 13 and come to an end. The figure in Fig. 5-11(*b*) is the final result.

Applying the Weiler–Atherton algorithm to clip the polygon in Fig. 5-10(*a*) produces correct result [see Fig. 5-12(*a*) and (*b*)].



(*a*) (*b*)

Fig. 5-12

5.5 EXAMPLE: A 2D GRAPHICS PIPELINE

Shared by many graphics systems is the overall system architecture called the graphics pipeline. The operational organization of a 2D graphics pipeline is shown in Fig. 5-13. Although 2D graphics is typically treated as a special case ($z = 0$) of three-dimensional graphics, it demonstrates the common working principle and basic application of these pipelined systems.

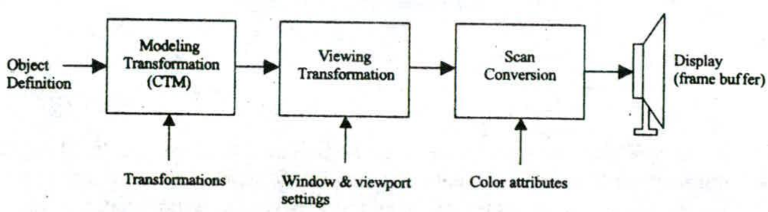
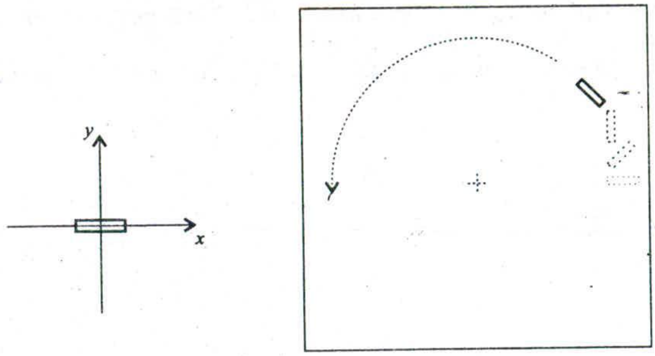


Fig. 5-13 A 2D graphics pipeline.

At the beginning of the pipeline we have object data (e.g., vertex coordinates for lines and polygons that make up individual objects) stored in application-specific data structures. A graphics application uses system subroutines to initialize and to change, among other things, the transformations that are to be applied to the original data, window and viewport settings, and the color attributes of the objects. Whenever a drawing subroutine is called to render a pre-defined object, the graphics system first applies the specified modeling transformation to the original data, then carries out viewing transformation using the current window and viewport settings, and finally performs scan conversion to set the proper pixels in the frame buffer with the specified color attributes.

Suppose that we have an object centered in its own coordinate system [see Fig. 5-14(*a*)], and we are to construct a sequence of images that animates the object rotating around its center and moving along a circular path in a square display area [see Fig. 5-14(*b*)]. We generate each image as follows: first rotate the object around its center by angle α, then translate the rotated object by offset $\cdot$ **I** to position its center on the positive x axis of the WCS, and rotate it with respect to the origin of the WCS by angle β. We control the amount of the first rotation from one image to the next by $\Delta\alpha$, and that of the second rotation by $\Delta\beta$.



(a) (b)

Fig. 5-14

```
window(-winsize/2, winsize/2, -winsize/2, winsize/2);
α = 0;
while (1) {
  setColor(background);
  clear();
  setColor(color);
  pushCTM();
  translate(offset, 0);
  rotate(α);
  drawObject();
  popCTM();
  α = α + Δα;
  rotate(Δβ);
}
```

We first set the window of winsize by winsize to be sufficiently large and centered at the origin of the WCS to cover the entire scene. The system's default viewport coincides with the unit display area in the NDCS. The default workstation window is the same as the viewport and the default workstation viewport corresponds to the whole square display area.

The graphics system maintains a stack of composite transformation matrices. The CTM on top of the stack, called the current CTM, is initially an identity matrix and is automatically used in modeling transformation. Each call to translate, scale, and rotate causes the system to generate a corresponding transformation matrix and to reset the current CTM to take into account the generated matrix. The order of transformation is maintained in such a way that the most recently specified transformation is applied first. When pushCTM() is called, the system makes a copy of the current CTM and pushes it onto the stack (now we have two copies of the current CTM on the stack). When popCTM() is called, the system simply removes the CTM on top of the stack (now we have restored the CTM that was second to the removed CTM to be the current CTM).

Panning and Zooming

Two simple camera effects can be achieved by changing the position or size of the window. When the position of the window is, for example, moved to the left, an object in the scene that is visible through the window would appear moved to the right, much like what we see in the viewfinder when we move or pan a camera. On the other hand, if we fix the window on an object but reduce or increase its size, the object would appear bigger (zoom in) or smaller (zoom out), respectively.

Double Buffering

Producing an animation sequence by clearing the display screen and constructing the next frame of image often leads to flicker, since an image is erased almost as soon as it is completed. An effective solution to this problem is to have two frame buffers: one holds the image on display while the system draws a new image into the other. Once the new image is drawn, a call that looks like swapBuffer() would cause the two buffers to switch their roles.

Lookup Table Animation

We can sometimes animate a displayed image in the lookup table representation by changing or cycling the color values in the lookup table. For example, we may draw the monochromatic object in Fig. 5-14(*a*) into the frame buffer in several pre-determined locations, using consecutive lookup table entries for the color attribute in each location (see Fig. 5-15). We initialize lookup table entry 0 with the color of the object, and all other entries with the background color. This means that in the beginning the object is visible only in its first position (labeled 0). Now if we simply reset entry 0 with the background color and entry 1 with the object color, we would have the object "moved" to its second position (labeled 1) without redrawing the image. The object's circular motion could hence be produced by cycling the object color through all relevant lookup table entries.

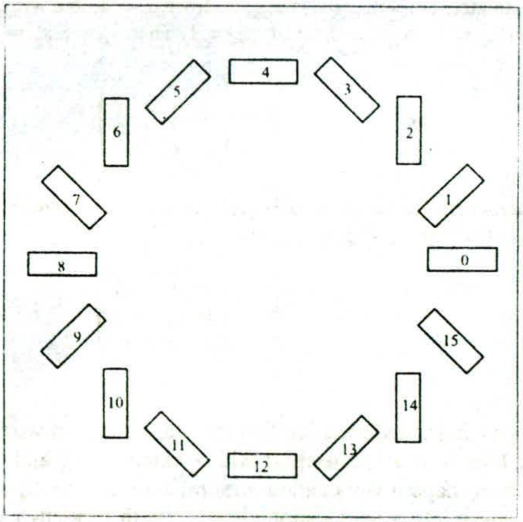


Fig. 5-15

Solved Problems

5.1 Let

$$s_x = \frac{vx_{\max} - vx_{\min}}{wx_{\max} - wx_{\min}} \qquad \text{and} \qquad s_y = \frac{vy_{\max} - vy_{\min}}{wy_{\max} - wy_{\min}}$$

Express window-to-viewport mapping in the form of a composite transformation matrix.

SOLUTION

$$N = \begin{pmatrix} 1 & 0 & vx_{\min} \\ 0 & 1 & vy_{\min} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -wx_{\min} \\ 0 & 1 & -wy_{\min} \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} s_x & 0 & -s_x wx_{\min} + vx_{\min} \\ 0 & s_y & -s_y wy_{\min} + vy_{\min} \\ 0 & 0 & 1 \end{pmatrix}$$

5.2 Find the normalization transformation that maps a window whose lower left corner is at (1, 1) and upper right corner is at (3, 5) onto (*a*) a viewport that is the entire normalized device screen and (*b*) a viewport that has lower left corner at (0, 0) and upper right corner $(\frac{1}{2}, \frac{1}{2})$.

SOLUTION

From Prob. 5.1, we need only identify the appropriate parameters.

(*a*) The window parameters are $wx_{\min} = 1$, $wx_{\max} = 3$, $wy_{\min} = 1$, and $wy_{\max} = 5$. The viewport parameters are $vx_{\min} = 0$, $vx_{\max} = 1$, $vy_{\min} = 0$, and $vy_{\max} = 1$. Then $s_x = \frac{1}{2}$, $s_y = \frac{1}{4}$, and

$$N = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & \frac{1}{4} & -\frac{1}{4} \\ 0 & 0 & 1 \end{pmatrix}$$

(*b*) The window parameters are the same as in (*a*). The viewport parameters are now $vx_{\min} = 0$, $vx_{\max} = \frac{1}{2}$, $vy_{\min} = 0$, $vy_{\max} = \frac{1}{2}$. Then $s_x = \frac{1}{4}$, $s_y = \frac{1}{8}$, and

$$N = \begin{pmatrix} \frac{1}{4} & 0 & -\frac{1}{4} \\ 0 & \frac{1}{8} & -\frac{1}{8} \\ 0 & 0 & 1 \end{pmatrix}$$

5.3 Find the complete viewing transformation that maps a window in world coordinates with x extent 1 to 10 and y extent 1 to 10 onto a viewport with x extent $\frac{1}{4}$ to $\frac{3}{4}$ and y extent 0 to $\frac{1}{2}$ in normalized device space, and then maps a workstation window with x extent $\frac{1}{4}$ to $\frac{1}{2}$ and y extent $\frac{1}{4}$ to $\frac{1}{2}$ in the normalized device space into a workstation viewport with x extent 1 to 10 and y extent 1 to 10 on the physical display device.

SOLUTION

From Prob. 5.1, the parameters for the normalization transformation are $wx_{\min} = 1$, $wx_{\max} = 10$, $wy_{\min} = 1$, $wy_{\max} = 10$, and $vx_{\min} = \frac{1}{4}$, $vx_{\max} = \frac{3}{4}$, $vy_{\min} = 0$, and $vy_{\max} = \frac{1}{2}$. Then

$$s_x = \frac{1/2}{9} = \frac{1}{18} \qquad s_y = \frac{1/2}{9} = \frac{1}{18}$$

and

$$N = \begin{pmatrix} \frac{1}{18} & 0 & \frac{7}{36} \\ 0 & \frac{1}{18} & -\frac{1}{18} \\ 0 & 0 & 1 \end{pmatrix}$$

The parameters for the workstation transformation are $wx_{\min} = \frac{1}{4}$, $wx_{\max} = \frac{1}{2}$, $wy_{\min} = \frac{1}{4}$, $wy_{\max} = \frac{1}{2}$, and $vx_{\min} = 1$, $vx_{\max} = 10$, $vy_{\min} = 1$, and $vy_{\max} = 10$. Then

$$s_x = \frac{9}{1/4} = 36 \qquad s_y = \frac{9}{1/4} = 36$$

and

$$W = \begin{pmatrix} 36 & 0 & -8 \\ 0 & 36 & -8 \\ 0 & 0 & 1 \end{pmatrix}$$

The complete viewing transformation V is

$$V = W \cdot N = \begin{pmatrix} 36 & 0 & -8 \\ 0 & 36 & -8 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{18} & 0 & \frac{7}{36} \\ 0 & \frac{1}{18} & -\frac{1}{18} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & -1 \\ 0 & 2 & -10 \\ 0 & 0 & 1 \end{pmatrix}$$

5.4 Find a normalization transformation from the window whose lower left corner is at (0, 0) and upper right corner is at (4, 3) onto the normalized device screen so that aspect ratios are preserved.

SOLUTION

The window aspect ratio is $a_w = \frac{4}{3}$. Unless otherwise indicated, we shall choose a viewport that is as large as possible with respect to the normalized device screen. To this end, we choose the x extent from 0 to 1 and the y extent from 0 to $\frac{3}{4}$. So

$$a_v = \frac{1}{3/4} = \frac{4}{3}$$

As in Prob. 5.2, with parameters $wx_{\min} = 0$, $wx_{\max} = 4$, $wy_{\min} = 0$, $wy_{\max} = 3$ and $vx_{\min} = 0$, $vx_{\max} = 1$, $vy_{\min} = 0$, $vy_{\max} = \frac{3}{4}$,

$$N = \begin{pmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

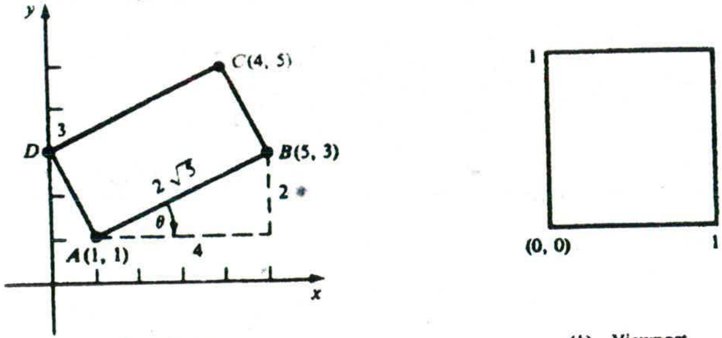
5.5 Find the normalization transformation N which uses the rectangle $A(1, 1)$, $B(5, 3)$, $C(4, 5)$, $D(0, 3)$ as a window [Fig. 5-16(*a*)] and the normalized device screen as a viewport [Fig. 5-16(*b*)].

SOLUTION

We will first rotate the rectangle about A so that it is aligned with the coordinate axes. Next, as in Prob. 5.1, we calculate s_x and s_y and finally we compose the rotation and the transformation N (from Prob. 5.1) to find the required normalization transformation N_R.

The slope of the line segment $\overline{AB}$ is

$$m = \frac{3-1}{5-1} = \frac{1}{2}$$



(a) Window. (b) Viewport.

Fig. 5-16

Looking at Fig. 5-11, we see that $-\theta$ will be the direction of the rotation. The angle θ is determined from the slope of a line (App. 1) by the equation $\tan\theta = \frac{1}{2}$. Then

$$\sin\theta = \frac{1}{\sqrt{5}}, \qquad \text{and so} \qquad \sin(-\theta) = -\frac{1}{\sqrt{5}}, \qquad \cos\theta = \frac{2}{\sqrt{5}}, \qquad \cos(-\theta) = \frac{2}{\sqrt{5}}$$

The rotation matrix about $A(1, 1)$ is then (Chap. 4, Prob. 4.4):

$$R_{-\theta,A} = \begin{pmatrix} \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \left(1 - \frac{3}{\sqrt{5}}\right) \\ -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & \left(1 - \frac{1}{\sqrt{5}}\right) \\ 0 & 0 & 1 \end{pmatrix}$$

The x extent of the rotated window is the length of $\overline{AB}$. Similarly, the y extent is the length of $\overline{AD}$. Using the distance formula (App. 1) to calculate these lengths yields

$$d(A, B) = \sqrt{2^2 + 4^2} = \sqrt{20} = 2\sqrt{5} \qquad d(A, D) = \sqrt{1^2 + 2^2} = \sqrt{5}$$

Also, the x extent of the normalized device screen is 1, as is the y extent. Calculating s_x and s_y,

$$s_x = \frac{\text{viewport } x \text{ extent}}{\text{window } x \text{ extent}} = \frac{1}{2\sqrt{5}} \qquad s_y = \frac{\text{viewport } y \text{ extent}}{\text{window } y \text{ extent}} = \frac{1}{\sqrt{5}}$$

So

$$N = \begin{pmatrix} \frac{1}{2\sqrt{5}} & 0 & -\frac{1}{2\sqrt{5}} \\ 0 & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \\ 0 & 0 & 1 \end{pmatrix}$$

The normalization transformation is then

$$N_R = N \cdot R_{-\theta,A} = \begin{pmatrix} \frac{1}{5} & \frac{1}{10} & -\frac{3}{10} \\ -\frac{1}{5} & \frac{2}{5} & -\frac{1}{5} \\ 0 & 0 & 1 \end{pmatrix}$$

5.6 Let R be the rectangular window whose lower left-hand corner is at $L(-3, 1)$ and upper right-hand corner is at $R(2, 6)$. Find the region codes for the endpoints in Fig. 5-17.

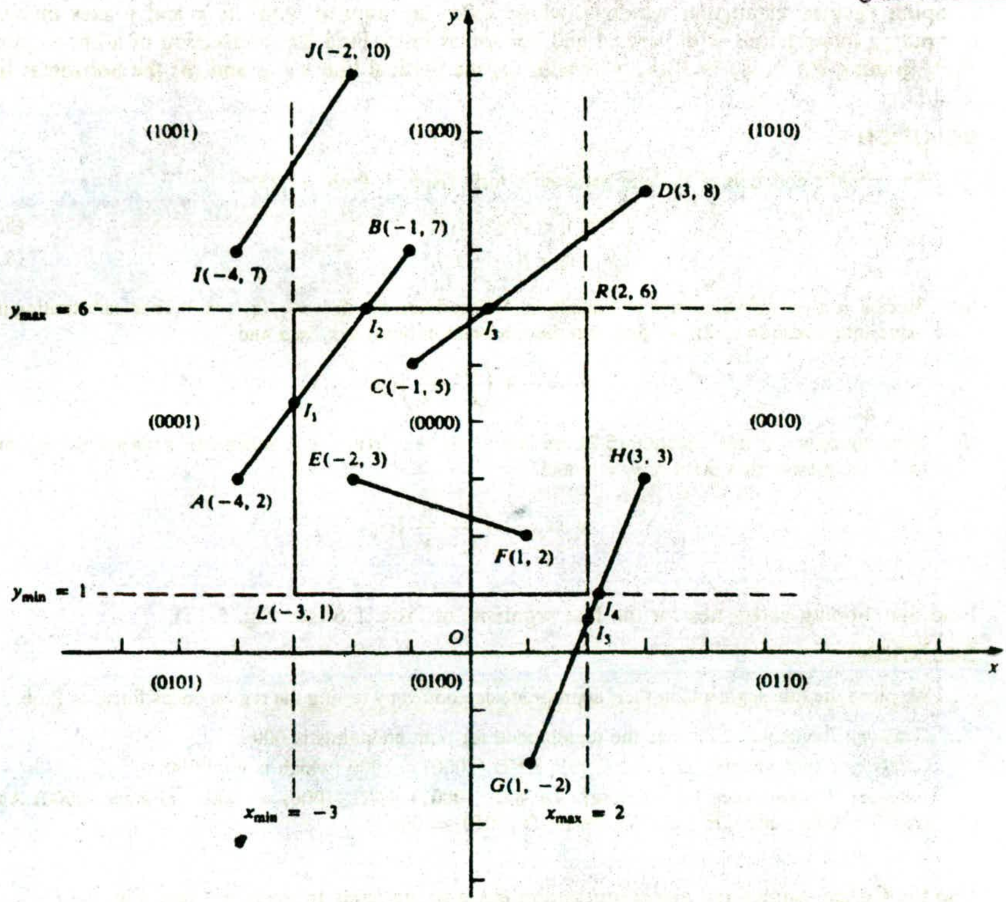


Fig. 5-17

SOLUTION

The region code for point (x, y) is set according to the scheme

Bit 1 = $\text{sign}(y - y_{max}) = \text{sign}(y - 6)$ Bit 3 = $\text{sign}(x - x_{max}) = \text{sign}(x - 2)$

Bit 2 = $\text{sign}(y_{min} - y) = \text{sign}(1 - y)$ Bit 4 = $\text{sign}(x_{min} - x) = \text{sign}(-3 - x)$

Here

$$\text{Sign}(a) = \begin{cases} 1 & \text{if } a \text{ is positive} \\ 0 & \text{otherwise} \end{cases}$$

So

$A(-4, 2) \rightarrow 0001$	$F(1, 2) \rightarrow 0000$
$B(-1, 7) \rightarrow 1000$	$G(1, -2) \rightarrow 0100$
$C(-1, 5) \rightarrow 0000$	$H(3, 3) \rightarrow 0010$
$D(3, 8) \rightarrow 1010$	$I(-4, 7) \rightarrow 1001$
$E(-2, 3) \rightarrow 0000$	$J(-2, 10) \rightarrow 1000$

5.7 Clipping against rectangular windows whose sides are aligned with the x and y axes involves computing intersections with vertical and horizontal lines. Find the intersection of a line segment $\overline{P_1P_2}$ [joining $P_1(x_1, y_1)$ to $P_2(x_2, y_2)$] with (*a*) the vertical line $x = a$ and (*b*) the horizontal line $y = b$.

SOLUTION

We write the equation of $\overline{P_1P_2}$ in parametric form (App. 1, Prob. A1.23):

$$\begin{cases} x = x_1 + t(x_2 - x_1) \\ y = y_1 + t(y_2 - y_1), \end{cases} \quad 0 \le t \le 1 \qquad \begin{matrix}(5.1)\\(5.2)\end{matrix}$$

(*a*) Since $x = a$, we substitute this into equation (5.1) and find $t = (a - x_1)/(x_2 - x_1)$. Then, substituting this value into equation (5.2), we find that the intersection point is $x_I = a$ and

$$y_I = y_1 + \left(\frac{a - x_1}{x_2 - x_1}\right)(y_2 - y_1)$$

(*b*) Substituting $y = b$ into equation (5.2), we find $t = (b - y_1)/(y_2 - y_1)$. When this is placed into equation (5.1), the intersection point is $y_I = b$ and

$$x_I = x_1 + \left(\frac{b - y_1}{y_2 - y_1}\right)(x_2 - x_1)$$

5.8 Find the clipping categories for the line segments in Prob. 5.6 (see Fig. 5-17).

SOLUTION

We place the line segments in their appropriate categories by testing the region codes found in Prob. 5.6.

Category 1 (visible): $\overline{EF}$ since the region code for both endpoints is 0000.

Category 2 (not visible): $\overline{IJ}$ since (1001) AND (1000) = 1000 (which is not 0000).

Category 3 (candidates for clipping): $\overline{AB}$ since (0001) AND (1000) = 0000, $\overline{CD}$ since (0000) AND (1010) = 0000, and $\overline{GH}$ since (0100) AND (0010) = 0000.

5.9 Use the Cohen–Sutherland algorithm to clip the line segments in Prob. 5.6 (see Fig. 5-17).

SOLUTION

From Prob. 5.8, the candidates for clipping are $\overline{AB}$, $\overline{CD}$, and $\overline{GH}$.

In *clipping* $\overline{AB}$, the code for A is 0001. To push the 1 to 0, we clip against the boundary line $x_{min} = -3$. The resulting intersection point is $I_1(-3, 3\frac{2}{3})$. We clip (do not display) $\overline{AI_1}$ and work on $\overline{I_1B}$. The code for I_1 is 0000. The clipping category for $\overline{I_1B}$ is 3 since (0000) AND (1000) is (0000). Now B is outside the window (i.e., its code is 1000), so we push the 1 to a 0 by clipping against the line $y_{max} = 6$. The resulting intersection is $I_2(-1\frac{3}{5}, 6)$. Thus $\overline{I_2B}$ is clipped. The code for I_2 is 0000. The remaining segment $\overline{I_1I_2}$ is displayed since both endpoints lie in the window (i.e., their codes are 0000).

For *clipping* $\overline{CD}$, we start with D since it is outside the window. Its code is 1010. We push the first 1 to a 0 by clipping against the line $y_{max} = 6$. The resulting intersection I_3 is $(\frac{1}{3}, 6)$ and its code is 0000. Thus $\overline{I_3D}$ is clipped and the remaining segment $\overline{CI_3}$ has both endpoints coded 0000 and so it is displayed.

For *clipping* $\overline{GH}$, we can start with either G or H since both are outside the window. The code for G is 0100, and we push the 1 to a 0 by clipping against the line $y_{min} = 1$. The resulting intersection point is $I_4(2\frac{1}{5}, 1)$, and its code is 0010. We clip $\overline{GI_4}$ and work on $\overline{I_4H}$. Segment $\overline{I_4H}$ is not displayed since (0010) AND (0010) = 0010.

5.10 Clip line segment $\overline{CD}$ of Prob. 5.6 by using the midpoint subdivision process.

SOLUTION

The midpoint subdivision process is based on repeated bisections. To avoid continuing indefinitely, we agree to say that a point (x_1, y_1) lies on any of the boundary lines of the rectangle, say, boundary line $x = x_{max}$, for example, if $-\text{TOL} \le x_1 - x_{max} \le \text{TOL}$. Here TOL is a prescribed tolerance, some small number, that is set before the process begins.

To clip $\overline{CD}$, we determine that it is in category 3. For this problem we arbitrarily choose TOL = 0.1. We find the midpoint of $\overline{CD}$ to be $M_1(1, 6.5)$. Its code is 1000.

So $\overline{M_1D}$ is not displayed since (1000) AND (1010) = 1000. We further subdivide $\overline{CM_1}$ since (0000) AND (1000) = 0000. The midpoint of $\overline{CM_1}$ is $M_2(0, 5.75)$; the code for M_2 is 0000. Thus $\overline{CM_2}$ is displayed since both endpoints are 0000 and $\overline{M_2M_1}$ is a candidate for clipping. The midpoint of $\overline{M_2M_1}$ is $M_3(0.5, 6.125)$, and its code is 1000. Thus $\overline{M_3M_1}$ is chipped and $\overline{M_2M_3}$ is subdivided. The midpoint of $\overline{M_2M_3}$ is $M_4(0.25, 5.9375)$, whose code is 0000. However, since $y_1 = 5.9375$ lies within the tolerance 0.1 of the boundary line $y_{max} = 6$—that is, $6 - 5.9375 = 0.0625 < 0.1$, we agree that M_4 lies on the boundary line $y_{max} = 6$. Thus $\overline{M_2M_4}$ is displayed and $\overline{M_4M_3}$ is not displayed. So the original line segment $\overline{CD}$ is clipped at M_4 and the process stops.

5.11 Suppose that in an implementation of the Cohen–Sutherland algorithm we choose boundary lines in the top–bottom–right–left order to clip a line in category 3, draw a picture to show a worst-case scenario, i.e., one that involves the highest number of iterations.

SOLUTION

See Fig. 5-18.

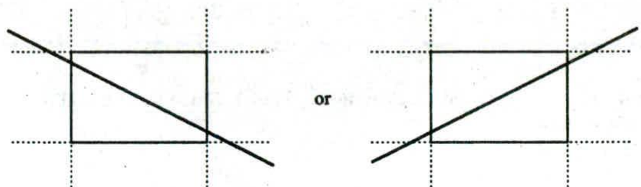


Fig. 5-18

5.12 Use the Liang–Barsky algorithm to clip the lines in Fig. 5-19.

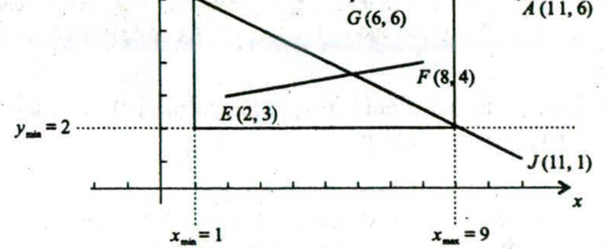


Fig. 5-19

SOLUTION

For line AB, we have

$$\begin{array}{ll} p_1 = 0 & q_1 = 10 \\ p_2 = 0 & q_2 = -2 \\ p_3 = -4 & q_3 = 4 \\ p_4 = 4 & q_4 = 2 \end{array}$$

Since $p_2 = 0$ and $q_2 < -2$, AB is completely outside the right boundary.

For line CD, we have

$$\begin{array}{lll} p_1 = 0 & q_1 = 2 & \\ p_2 = 0 & q_2 = 6 & \\ p_3 = -3 & q_3 = 5 & r_3 = -\frac{5}{3} \\ p_4 = 3 & q_4 = 1 & r_4 = \frac{1}{3} \end{array}$$

Thus $u_1 = \max(0, -\frac{5}{3}) = 0$ and $u_2 = \min(1, \frac{1}{3}) = \frac{1}{3}$. Since $u_1 < u_2$, the two endpoints of the clipped line are $(3, 7)$ and $(3, 7 + 3(\frac{1}{3})) = (3, 8)$.

For line EF, we have

$$\begin{array}{lll} p_1 = -6 & q_1 = 1 & r_1 = -\frac{1}{6} \\ p_2 = 6 & q_2 = 7 & r_2 = \frac{7}{6} \\ p_3 = -1 & q_3 = 1 & r_3 = -\frac{1}{1} \\ p_4 = 1 & q_4 = 5 & r_4 = \frac{5}{1} \end{array}$$

Thus $u_1 = \max(0, -\frac{1}{6}, -1) = 0$ and $u_2 = \min(1, \frac{7}{6}, 5) = 1$. Since $u_1 = 0$ and $u_2 = 1$, line EF is completely inside the clipping window.

For line GH, we have

$$\begin{array}{lll} p_1 = -2 & q_1 = 5 & r_1 = -\frac{5}{2} \\ p_2 = 2 & q_2 = 3 & r_2 = \frac{3}{2} \\ p_3 = -3 & q_3 = 4 & r_3 = -\frac{4}{3} \\ p_4 = 3 & q_4 = 2 & r_4 = \frac{2}{3} \end{array}$$

Thus $u_1 = \max(0, -\frac{5}{2}, -\frac{4}{3}) = 0$ and $u_2 = \min(1, \frac{3}{2}, \frac{2}{3}) = \frac{2}{3}$. Since $u_1 < u_2$, the two endpoints of the clipped line are $(6, 6)$ and $(6 + 2(\frac{2}{3}), 6 + 3(\frac{2}{3})) = (7\frac{1}{3}, 8)$.

For line IJ, we have

$$\begin{array}{lll} p_1 = -12 & q_1 = -2 & r_1 = \frac{1}{6} \\ p_2 = 12 & q_2 = 10 & r_2 = \frac{5}{6} \\ p_3 = 6 & q_3 = 5 & r_3 = \frac{5}{6} \\ p_4 = -6 & q_4 = 1 & r_4 = -\frac{1}{6} \end{array}$$

Thus $u_1 = \max(0, \frac{1}{6}, -\frac{1}{6}) = \frac{1}{6}$ and $u_2 = \min(1, \frac{5}{6}, \frac{5}{6}) = \frac{5}{6}$. Since $u_1 < u_2$, the two endpoints of the clipped line are $(-1 + 12(\frac{1}{6}), 7 + (-6)(\frac{1}{6})) = (1, 6)$ and $(-1 + 12(\frac{5}{6}), 7 + (-6)(\frac{5}{6})) = (9, 2)$.

5.13 How can we determine whether a point $P(x, y)$ lies to the left or to the right of a line segment joining the points $A(x_1, y_1)$ and $B(x_2, y_2)$?

SOLUTION

Refer to Fig. 5-20. Form the vectors **AB** and **AP**. If the point P is to the left of **AB**, then by the definition of the cross product of two vectors (App. 2) the vector **AB** × **AP** points in the direction of the vector **K** perpendicular to the xy plane (see Fig. 5-20). If it lies to the right, the cross product points in the direction

$-\mathbf{K}$. Now

$$\mathbf{AB} = (x_2 - x_1)\mathbf{I} + (y_2 - y_1)\mathbf{J} \qquad \mathbf{AP} = (x - x_1)\mathbf{I} + (y - y_1)\mathbf{J}$$

So

$$\mathbf{AB} \times \mathbf{AP} = [(x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1)]\mathbf{K}$$

Then the direction of this cross product is determined by the number

$$\bar{C} = (x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1)$$

If $\bar{C}$ is positive, P lies to the left of **AB**. If $\bar{C}$ is negative, then P lies to the right of **AB**.

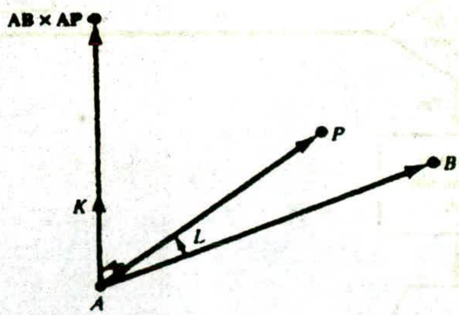


Fig. 5-20

5.14 Draw a flowchart illustrating the logic of the Sutherland–Hodgman algorithm.

SOLUTION

The algorithm inputs the vertices of a polygon one at a time. For each input vertex, either zero, one, or two output vertices will be generated depending on the relationship of the input vertices to the clipping edge E.

We denote by P the input vertex, S the previous input vertex, and F the first arriving input vertex. The vertex or vertices to be output are determined according to the logic illustrated in the flowchart in Fig. 5-21. Recall that a polygon with n vertices $P_1, P_2, \ldots, P_n$ has n edges $\overline{P_1P_2}, \ldots, \overline{P_{n-1}P_n}$ and the edge $\overline{P_nP_1}$ closing the polygon. In order to avoid the need to duplicate the input of P_1 as the final input vertex (and a corresponding mechanism to duplicate the final output vertex to close the polygon), the closing logic shown in the flowchart in Fig. 5-22 is called after processing the final input vertex P_n.

5.15 Clip the polygon $P_1, \ldots, P_9$ in Fig. 5-23 against the window $ABCD$ using the Sutherland–Hodgman algorithm.

SOLUTION

At each stage the new output polygon, whose vertices are determined by applying the Sutherland–Hodgman algorithm (Prob. 5.14), is passed on to the next clipping edge of the window $ABCD$. The results are illustrated in Figs. 5-24 through 5-27.

5.16 Clip the polygon $P_1, \ldots, P_8$ in Fig. 5-10 against the rectangular clipping window using the Sutherland–Hodgman algorithm.

SOLUTION

We first clip against the top boundary line, then the left, and finally the bottom. The right boundary is omitted since it does not affect any vertex list. The intermediate and final results are in Fig. 5-28.

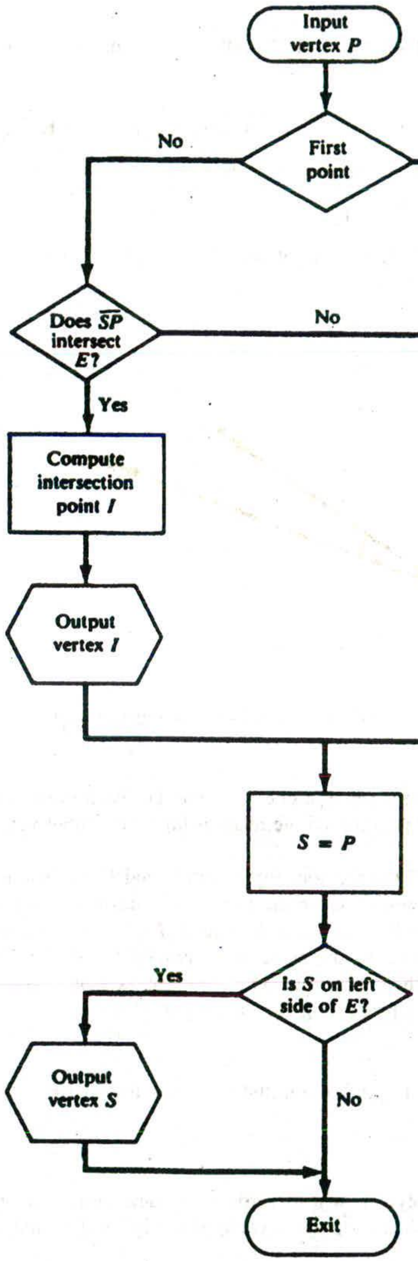

Input vertex P
First point
No
Yes
$F = P$
Does $\overline{SP}$ intersect E?
No
Yes
Compute intersection point I
Output vertex I
$S = P$
Is S on left side of E?
Yes
No
Output vertex S
Exit

Fig. 5-21

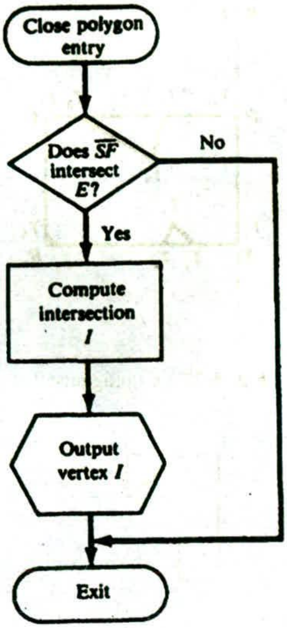


Fig. 5-22

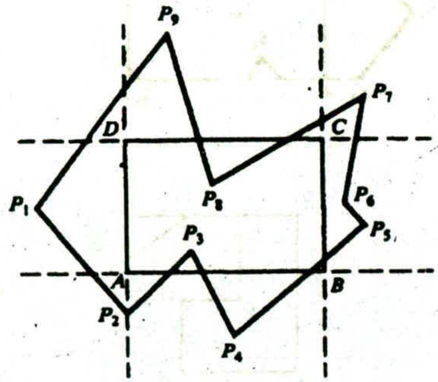


Fig. 5-23

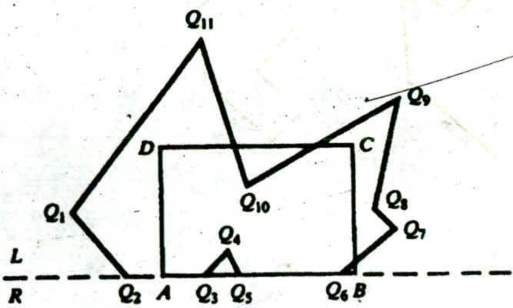


Fig. 5-24 Clip against $\overline{AB}$.

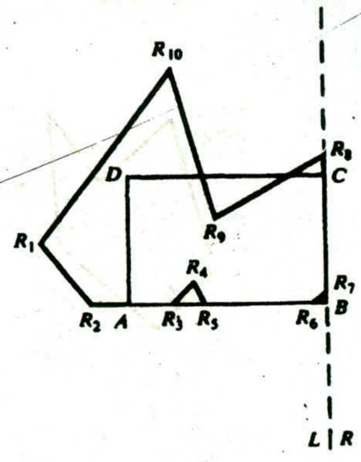


Fig. 5-25 Clip against $\overline{BC}$.

5.17 Use the Weiler–Atherton algorithm to clip the polygon in Fig. 5-29(a).

SOLUTION

See Fig. 5-29(b) and (c).

5.18 Consider the example in Sect. 5.5, where the object would appear turning slowly around its center even if we set $\Delta\alpha = 0$. How to keep the orientation of the object constant while making it rotate around the center of the display area?

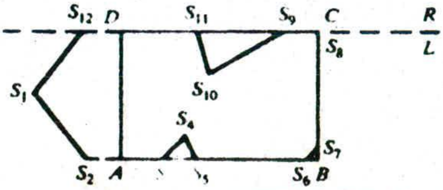


Fig. 5-26 Clip against $\overline{CD}$.

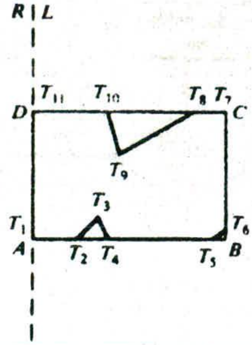
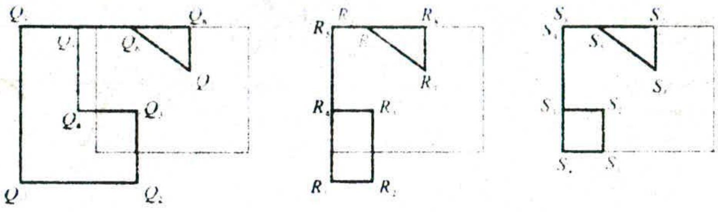


Fig. 5-27 Clip against $\overline{DA}$.

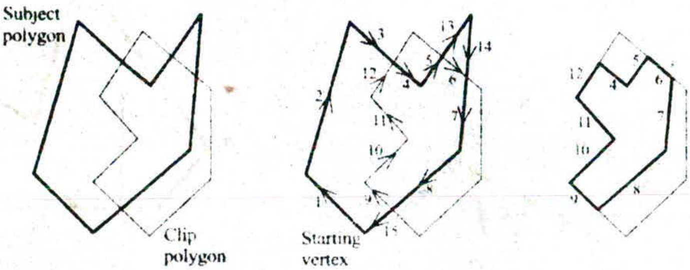


Clip against top boundary

Clip against left boundary

Clip against bottom boundary

Fig. 5-28



(a) (b) (c)

Fig. 5-29

SOLUTION

$\Delta\alpha = -\Delta\beta$, i.e., $\alpha = -\beta$.

5.19 How to animate the flag in Fig. 5-30(*a*) that may be in two different positions using lookup table animation?

SOLUTION

See Fig. 5-30(*b*). The area where position 1 overlaps position 2 is assigned entry 0 that has the color of the flag. The rest of position 1 is assigned entry 1 and that of position 2 entry 2. Now we only need to alternate entries 1 and 2 between the flag color and the background color.

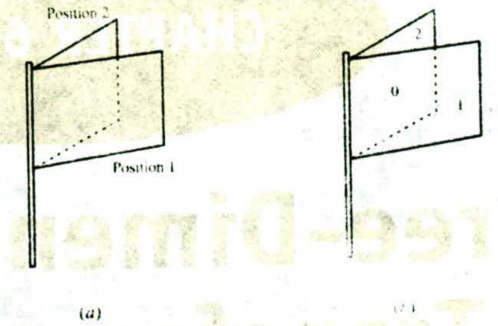


Fig. 5-30

Supplementary Problems

5.20 Find the workstation transformation that maps the normalized device screen onto a physical device whose x extent is 0 to 199 and y extent is 0 to 639 where the origin is located at the (*a*) lower left corner and (*b*) upper left corner of the device.

5.21 Show that for a viewing transformation, $s_x = s_y$ if and only if $a_w = a_v$, where a_w is the aspect ratio of the window and a_v the aspect ratio of the viewport.

5.22 Find the normalization transformation which uses a circle of radius five units and center (1, 1) as a window and a circle of radius $\frac{1}{2}$ and center $(\frac{1}{2}, \frac{1}{2})$ as a viewport.

5.23 Describe how clipping a line against a circular window (or viewport) might proceed. Refer to Fig. 5-31.

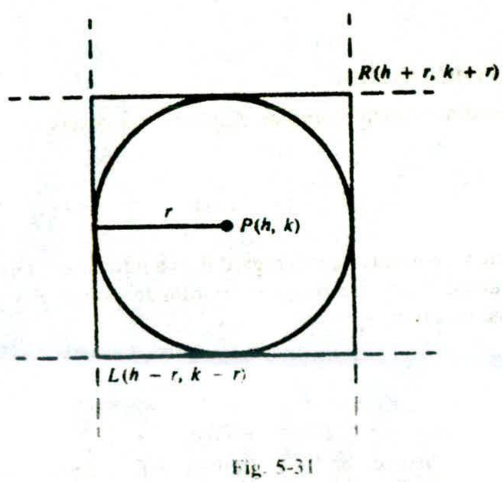


Fig. 5-31

5.24 Use the Sutherland-Hodgman algorithm to clip the line segment joining $P_1(-1, 2)$ to $P_2(6, 4)$ against the rotated window in Prob. 5.5.

CHAPTER 6

Three-Dimensional Transformations

Manipulation, viewing, and construction of three-dimensional graphic images requires the use of three-dimensional geometric and coordinate transformations. These transformations are formed by composing the basic transformations of translation, scaling, and rotation. Each of these transformations can be represented as a matrix transformation. This permits more complex transformations to be built up by use of matrix multiplication or concatenation.

As with two-dimensional transformations, two complementary points of view are adopted: either the object is manipulated directly through the use of geometric transformations, or the object remains stationary and the viewer's coordinate system is changed by using coordinate transformations. In addition, the construction of complex objects and scenes is facilitated by the use of instance transformations. The concepts and transformations introduced here are direct generalizations of those introduced in Chap. 4 for two-dimensional transformations.

6.1 GEOMETRIC TRANSFORMATIONS

With respect to some three-dimensional coordinate system, an object *Obj* is considered as a set of points:

$$Obj = \{P(x, y, z)\}$$

If the object is moved to a new position, we can regard it as a new object Obj', all of whose coordinate points $P'(x', y', z')$ can be obtained from the original coordinate points $P(x, y, z)$ of *Obj* through the application of a geometric transformation.

Translation

An object is displaced a given distance and direction from its original position. The direction and displacement of the translation is prescribed by a vector

$$\mathbf{V} = a\mathbf{I} + b\mathbf{J} + c\mathbf{K}$$

The new coordinates of a translated point can be calculated by using the transformation

$$T_{\mathbf{v}}: \begin{cases} x' = x + a \\ y' = y + b \\ z' = z + c \end{cases}$$

(see Fig. 6-1). In order to represent this transformation as a matrix transformation, we need to use homogeneous coordinates (App. 2). The required homogeneous matrix transformation can then be expressed as

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

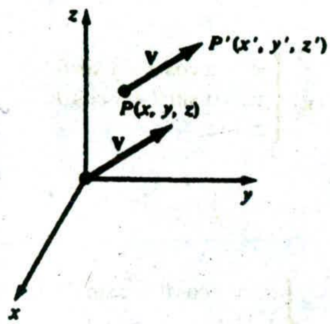


Fig. 6-1

Scaling

The process of scaling changes the dimensions of an object. The scale factor s determines whether the scaling is a magnification, $s > 1$, or a reduction, $s < 1$.

Scaling with respect to the origin, where the origin remains fixed, is effected by the transformation

$$S_{s_x, s_y, s_z}: \begin{cases} x' = s_x \cdot x \\ y' = s_y \cdot y \\ z' = s_z \cdot z \end{cases}$$

In matrix form this is

$$S_{s_x, s_y, s_z} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix}$$

Rotation

Rotation in three dimensions is considerably more complex than rotation in two dimensions. In two dimensions, a rotation is prescribed by an angle of rotation θ and a center of rotation P. Three-dimensional rotations require the prescriptioin of an angle of rotation and an axis of rotation. The *canonical* rotations are defined when one of the positive $x, y,$ or z coordinate axes is chosen as the axis of rotation. Then the construction of the rotation transformation proceeds just like that of a rotation in two dimensions about the origin (see Fig. 6-2).

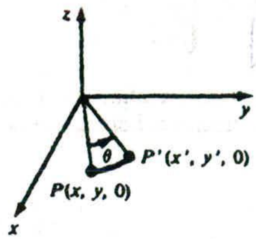


Fig. 6-2

Rotation about the z Axis

From Chap. 4 we know that

$$R_{\theta,\mathbf{K}}: \begin{cases} x' = x\cos\theta - y\sin\theta \\ y' = x\sin\theta + y\cos\theta \\ z' = z \end{cases}$$

Rotation about the y Axis

An analogous derivation leads to

$$R_{\theta,\mathbf{J}}: \begin{cases} x' = x\cos\theta + z\sin\theta \\ y' = y \\ z' = -x\sin\theta + z\cos\theta \end{cases}$$

Rotation about the x Axis

Similarly:

$$R_{\theta,\mathbf{I}}: \begin{cases} x' = x \\ y' = y\cos\theta - z\sin\theta \\ z' = y\sin\theta + z\cos\theta \end{cases}$$

Note that the direction of a positive angle of rotation is chosen in accordance to the right-hand rule with respect to the axis of rotation (App. 2).

The corresponding matrix transformations are

$$R_{\theta,\mathbf{K}} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_{\theta,\mathbf{J}} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

$$R_{\theta,\mathbf{I}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

The general use of rotation about an axis L can be built up from these canonical rotations using matrix multiplication (Prob. 6.3).

6.2 COORDINATE TRANSFORMATIONS

We can also achieve the effects of translation, scaling, and rotation by moving the observer who views the object and by keeping the object stationary. This type of transformation is called a *coordinate*

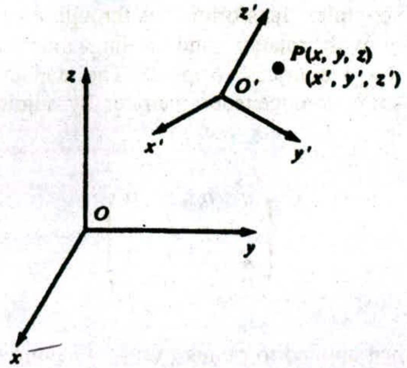


Fig. 6-3

transformation. We first attach a coordinate system to the observer and then move the observer and the attached coordinate system. Next, we recalculate the coordinates of the observed object with respect to this new observer coordinate system. The new coordinate values will be exactly the same as if the observer had remained stationary and the object had moved, corresponding to a geometric transformation (see Fig. 6-3).

If the displacement of the observer coordinate system to a new position is prescribed by a vector $\mathbf{V} = a\mathbf{I} + b\mathbf{J} + c\mathbf{K}$, a point $P(x, y, z)$ in the original coordinate system has coordinates $P(x', y', z')$ in the new coordinate system, and

$$\bar{T}_{\mathbf{V}}: \begin{cases} x' = x - a \\ y' = y - b \\ z' = z - c \end{cases}$$

The derivation of this transformation is completely analogous to that of the two-dimensional transformation (see Chap. 4).

Similar derivations hold for coordinate scaling and coordinate rotation transformations.

As in the two-dimensional case, we summarize the relationships between the matrix forms of the coordinate transformations and the geometeric transformations:

	Coordinate Transformations	**Geometric Transformations**
Translation	$\bar{T}_{\mathbf{V}}$	$T_{-\mathbf{V}}$
Rotation	$\bar{R}_\theta$	$R_{-\theta}$
Scaling	$\bar{S}_{s_x, s_y, s_z}$	$S_{1/s_x, 1/s_y, 1/s_z}$

Inverse geometric and coordinate transformations are constructed by performing the reverse operation. Thus, for coordinate transformations (and similarly for geometric transformations):

$$\bar{T}_{\mathbf{V}}^{-1} = \bar{T}_{-\mathbf{V}} \qquad \bar{R}_\theta^{-1} = \bar{R}_{-\theta} \qquad \bar{S}_{s_x, s_y, s_z}^{-1} = \bar{S}_{1/s_x, 1/s_y, 1/s_z}$$

6.3 COMPOSITE TRANSFORMATIONS

More complex geometric and coordinate transformations are formed through the process of *composition of functions*. For matrix functions, however, the process of composition is equivalent to matrix multiplication or concatenation. In Probs. 6.2, 6.3, 6.5, and 6.13, the following transformations are constructed:

1. $A_{\mathbf{V},\mathbf{N}}$ = aligning a vector **V** with a vector **N**.
2. $R_{\theta,L}$ = rotation about an axis L. This axis is prescribed by giving a direction vector **V** and a point P through which the axis passes.
3. $S_{s_x,s_y,s_z,P}$ = scaling with respect to an arbitrary point P.

In order to build these more complex transformations through matrix concatenation, we must be able to multiply translation matrices with rotation and scaling matrices. This necessitates the use of homogeneous coordinates and 4 × 4 matrices (App. 2). The standard 3 × 3 matrices of rotation and scaling can be represented as 4 × 4 homogeneous matrices by adjoining an extra row and column as follows:

$$\begin{pmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

These transformations are then applied to points $P(x, y, z)$ having the homogeneous form:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

EXAMPLE 1. The matrix of rotation about the y axis has the homogeneous 4 × 4 form:

$$R_{\theta,\mathbf{J}} = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

6.4 INSTANCE TRANSFORMATIONS

If an object is created and described in coordinates with respect to its own object coordinate space, we can place an instance or copy of it within a larger scene that is described in an independent coordinate space by the use of three-dimensional coordinate transformations. In this case, the transformations are referred to as *instance transformations*. The concepts and construction of three-dimensional instance transformations and the composite transformation matrix are completely analogous to the two-dimensional cases described in Chap. 4.

Solved Problems

6.1 Define *tilting* as a rotation about the x axis followed by a rotation about the y axis: (*a*) find the tilting matrix; (*b*) does the order of performing the rotation matter?

SOLUTION

(*a*) We can find the required transformation T by composing (concatenating) two rotation matrices:

$$T = R_{\theta_y,\mathbf{J}} \cdot R_{\theta_x,\mathbf{I}}$$

$$= \begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos\theta_y & \sin\theta_y \sin\theta_x & \sin\theta_y \cos\theta_x & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ -\sin\theta_y & \cos\theta_y \sin\theta_x & \cos\theta_y \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(*b*) We multiply $R_{\theta_x,\mathbf{I}} \cdot R_{\theta_y,\mathbf{J}}$ to obtain the matrix

$$\begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ \sin\theta_x \sin\theta_y & \cos\theta_x & -\sin\theta_x \cos\theta_y & 0 \\ -\cos\theta_x \sin\theta_y & \sin\theta_x & \cos\theta_x \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

This is not the same matrix as in part *a*; thus the order of rotation matters.

6.2 Find a transformation $A_{\mathbf{V}}$ which aligns a given vector $\mathbf{V}$ with the vector $\mathbf{K}$ along the positive z axis.

SOLUTION

See Fig. 6-4(*a*). Let $\mathbf{V} = a\mathbf{I} + b\mathbf{J} + c\mathbf{K}$. We perform the alignment through the following sequence of transformations [Figs. 6-4(*b*) and 6-4(*c*)]:

1. Rotate about the x axis by an angle θ_1 so that V rotates into the upper half of the xz plane (as the vector $\mathbf{V}_1$).
2. Rotate the vector $\mathbf{V}_1$ about the y axis by an angle $-\theta_2$ so that $\mathbf{V}_1$ rotates to the positive z axis (as the vector $\mathbf{V}_2$).

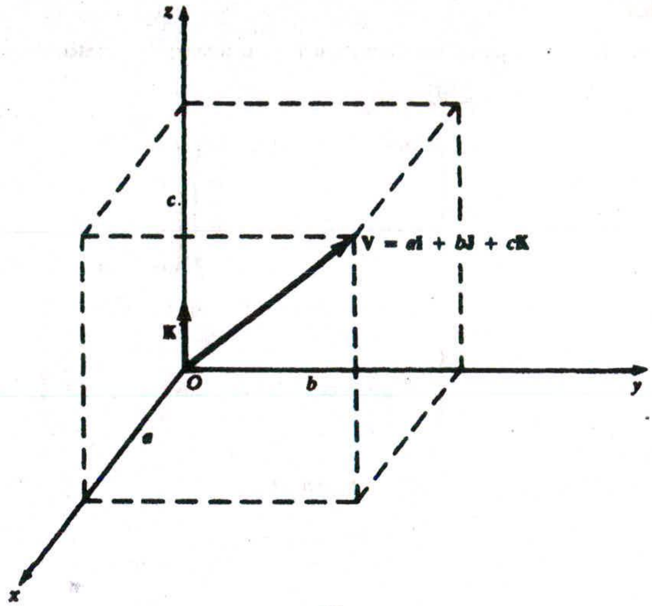
Implementing step 1 from Fig. 6-4(b), we observe that the required angle of rotation θ_1 can be found by looking at the projection of $\mathbf{V}$ onto the yz plane. (We assume that b and c are not both zero.) From triangle $OP'B$:

$$\sin\theta_1 = \frac{b}{\sqrt{b^2 + c^2}} \qquad \cos\theta_1 = \frac{c}{\sqrt{b^2 + c^2}}$$

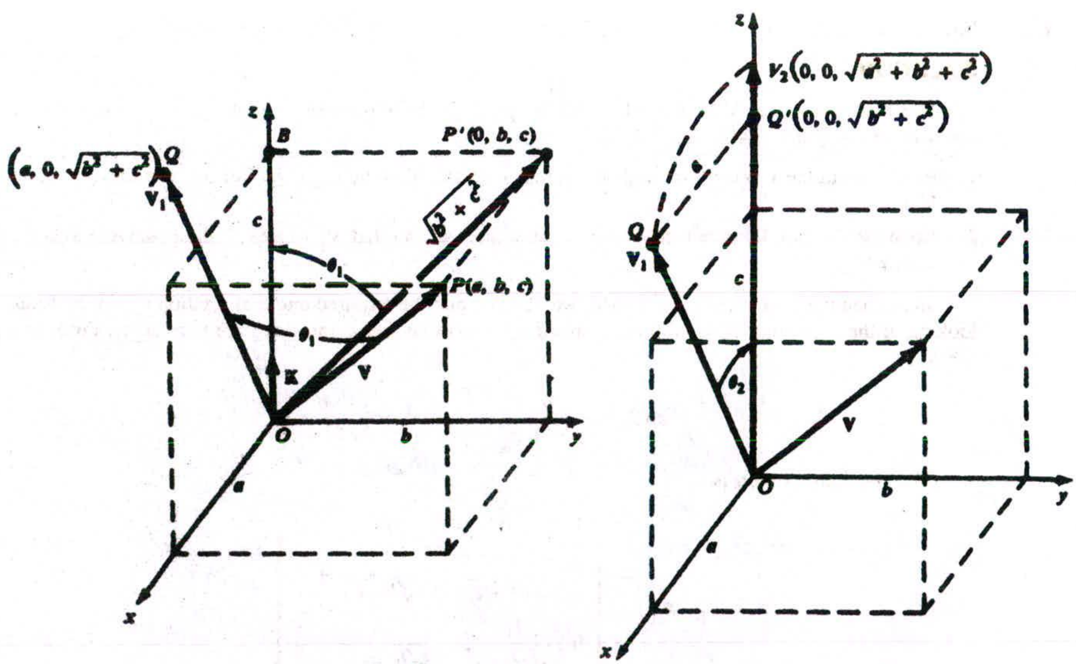
The required rotation is

$$R_{\theta_1,\mathbf{I}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \dfrac{c}{\sqrt{b^2 + c^2}} & -\dfrac{b}{\sqrt{b^2 + c^2}} & 0 \\ 0 & \dfrac{b}{\sqrt{b^2 + c^2}} & \dfrac{c}{\sqrt{b^2 + c^2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Applying this rotation to the vector $\mathbf{V}$ produces the vector $\mathbf{V}_1$ with the components $(a, 0, \sqrt{b^2 + c^2})$.


z
c
V = aI + bJ + cK
K
O
b
y
a
x

(a)


z
B
P'(0, b, c)
(a, 0, √(b² + c²)) Q
V₁
c
θ₁
P(a, b, c)
K
V
O
b
y
a
x
z
V₂(0, 0, √(a² + b² + c²))
Q'(0, 0, √(b² + c²))
Q
V₁
c
θ₂
V
O
b
y
a
x

(b) (c)

Fig. 6-4

Implementing step 2 from Fig. 6-4(*c*), we see that a rotation of $-\theta_2$ degrees is required, and so from triangle OQQ':

$$\sin(-\theta_2) = -\sin\theta_2 = -\frac{a}{\sqrt{a^2+b^2+c^2}} \quad \text{and} \quad \cos(-\theta_2) = \cos\theta_2 = \frac{\sqrt{b^2+c^2}}{\sqrt{a^2+b^2+c^2}}$$

Then

$$R_{-\theta_2,\mathbf{J}} = \begin{pmatrix} \dfrac{\sqrt{b^2+c^2}}{\sqrt{a^2+b^2+c^2}} & 0 & \dfrac{-a}{\sqrt{a^2+b^2+c^2}} & 0 \\ 0 & 1 & 0 & 0 \\ \dfrac{a}{\sqrt{a^2+b^2+c^2}} & 0 & \dfrac{\sqrt{b^2+c^2}}{\sqrt{a^2+b^2+c^2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Since $|\mathbf{V}| = \sqrt{a^2+b^2+c^2}$, and introducing the notation $\lambda = \sqrt{b^2+c^2}$, we find

$$A_{\mathbf{V}} = R_{-\theta_2,\mathbf{J}} \cdot R_{\theta_1,\mathbf{I}}$$

$$= \begin{pmatrix} \dfrac{\lambda}{|\mathbf{V}|} & \dfrac{-ab}{\lambda|\mathbf{V}|} & \dfrac{-ac}{\lambda|\mathbf{V}|} & 0 \\ 0 & \dfrac{c}{\lambda} & \dfrac{-b}{\lambda} & 0 \\ \dfrac{a}{|\mathbf{V}|} & \dfrac{b}{|\mathbf{V}|} & \dfrac{c}{|\mathbf{V}|} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

If both b and c are zero, then $\mathbf{V} = a\mathbf{I}$, and so $\lambda = 0$. In this case, only a $\pm 90°$ rotation about the y axis is required. So if $\lambda = 0$, it follows that

$$A_{\mathbf{v}} = R_{-\theta_2,\mathbf{J}} = \begin{pmatrix} 0 & 0 & \dfrac{-a}{|a|} & 0 \\ 0 & 1 & 0 & 0 \\ \dfrac{a}{|a|} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In the same manner we calculate the inverse transformation that aligns the vector $\mathbf{K}$ with the vector $\mathbf{V}$:

$$A_{\mathbf{v}}^{-1} = (R_{-\theta_2,\mathbf{J}} \cdot R_{\theta_1,\mathbf{I}})^{-1} = R_{\theta_1,\mathbf{I}}^{-1} \cdot R_{-\theta_2,\mathbf{J}}^{-1} = R_{-\theta_1,\mathbf{I}} \cdot R_{\theta_2,\mathbf{J}}$$

$$= \begin{pmatrix} \dfrac{\lambda}{|\mathbf{V}|} & 0 & \dfrac{a}{|\mathbf{V}|} & 0 \\ \dfrac{-ab}{\lambda|\mathbf{V}|} & \dfrac{c}{\lambda} & \dfrac{b}{|\mathbf{V}|} & 0 \\ \dfrac{-ac}{\lambda|\mathbf{V}|} & -\dfrac{b}{\lambda} & \dfrac{c}{|\mathbf{V}|} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

6.3 Let an axis of rotation L be specified by a direction vector $\mathbf{V}$ and a location point P. Find the transformation for a rotation of $\theta°$ about L. Refer to Fig. 6-5.

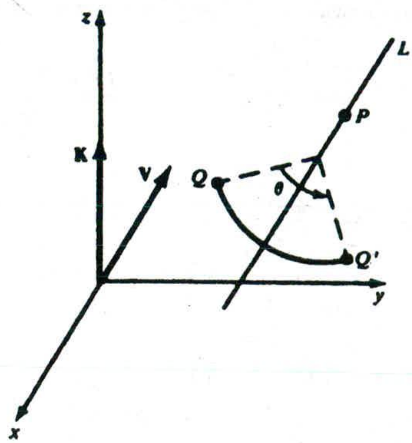


Fig. 6-5

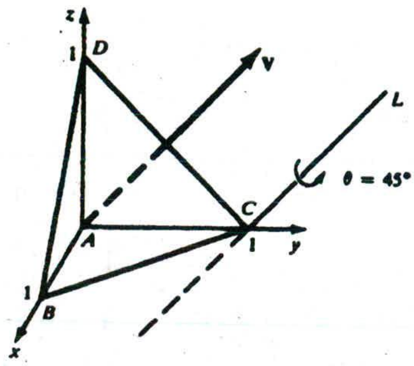


Fig. 6-6

SOLUTION

We can find the required transformation by the following steps:

1. Translate P to the origin.
2. Align $\mathbf{V}$ with the vector $\mathbf{K}$.
3. Rotate by $\theta°$ about $\mathbf{K}$.
4. Reverse steps 2 and 1.

So

$$R_{\theta,L} = T_{-P}^{-1} \cdot A_{\mathbf{V}}^{-1} \cdot R_{\theta,\mathbf{K}} \cdot A_{\mathbf{V}} \cdot T_{-P}$$

Here, $A_{\mathbf{V}}$ is the transformation described in Prob. 6.2.

6.4 The pyramid defined by the coordinates $A(0, 0, 0)$, $B(1, 0, 0)$, $C(0, 1, 0)$, and $D(0, 0, 1)$ is rotated 45° about the line L that has the direction $\mathbf{V} = \mathbf{J} + \mathbf{K}$ and passing through point $C(0, 1, 0)$ (Fig. 6-6). Find the coordinates of the rotated figure.

SOLUTION

From Prob. 6.3, the rotation matrix $R_{\theta,L}$ can be found by concatenating the matrices

$$R_{\theta,L} = T_{-P}^{-1} \cdot A_{\mathbf{V}}^{-1} \cdot R_{\theta,\mathbf{K}} \cdot A_{\mathbf{V}} \cdot T_{-P}$$

With $P = (0, 1, 0)$, then

$$T_{-P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Now $\mathbf{V} = \mathbf{J} + \mathbf{K}$. So from Prob. 6.2, with $a = 0$, $b = 1$, $c = 1$, we find $\lambda = \sqrt{2}$, $|\mathbf{V}| = \sqrt{2}$, and

$$A_{\mathbf{V}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad A_{\mathbf{V}}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Also

$$R_{45^\circ,\mathbf{K}} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad T_{-P}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Then

$$R_{\theta,L} = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{2+\sqrt{2}}{4} & \frac{2-\sqrt{2}}{4} & \frac{2-\sqrt{2}}{4} \\ -\frac{1}{2} & \frac{2-\sqrt{2}}{4} & \frac{2+\sqrt{2}}{4} & \frac{\sqrt{2}-2}{4} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

To find the coordinates of the rotated figure, we apply the rotation matrix $R_{\theta,L}$ to the matrix of homogeneous coordinates of the vertices A, B, C, and D:

$$C = (ABCD) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

So

$$R_{\theta,L} \cdot C = \begin{pmatrix} \frac{1}{2} & \frac{1+\sqrt{2}}{2} & 0 & 1 \\ \frac{2-\sqrt{2}}{4} & \frac{4-\sqrt{2}}{4} & 1 & \frac{2-\sqrt{2}}{2} \\ \frac{\sqrt{2}-2}{4} & \frac{\sqrt{2}-4}{4} & 0 & \frac{\sqrt{2}}{2} \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

The rotated coordinates are (Fig. 6-7)

$$A' = \left(\frac{1}{2}, \frac{2-\sqrt{2}}{4}, \frac{\sqrt{2}-2}{4}\right) \qquad C' = (0, 1, 0)$$

$$B' = \left(\frac{1+\sqrt{2}}{2}, \frac{4-\sqrt{2}}{4}, \frac{\sqrt{2}-4}{4}\right) \qquad D' = \left(1, \frac{2-\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$$

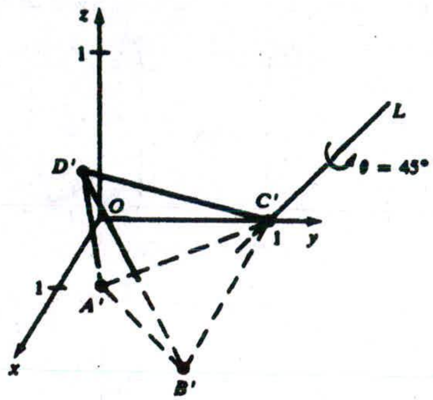


Fig. 6-7

6.5 Find a transformation $A_{\mathbf{V},\mathbf{N}}$ which aligns a vector **V** with a vector **N**.

SOLUTION

We form the transformation in two steps. First, align **V** with vector **K**, and second, align vector **K** with vector **N**. So from Prob. 6.2,

$$\mathbf{A}_{\mathbf{V},\mathbf{N}} = \mathbf{A}_{\mathbf{N}}^{-1} \cdot \mathbf{A}_{\mathbf{V}}$$

Referring to Prob. 6.12, we could also get $\mathbf{A}_{\mathbf{V},\mathbf{N}}$ by rotating **V** towards **N** about the axis $\mathbf{V} \times \mathbf{N}$.

6.6 Find the transformation for mirror reflection with respect to the xy plane.

SOLUTION

From Fig. 6-8, it is easy to see that the reflection of $P(x, y, z)$ is $P'(x, y, -z)$. The transformation that performs this reflection is

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

6.7 Find the transformation for mirror reflection with respect to a given plane. Refer to Fig. 6-9.

SOLUTION

Let the plane of reflection be specified by a normal vector **N** and a reference point $P_0(x_0, y_0, z_0)$. To reduce the reflection to a mirror reflection with respect to the xy plane:

1. Translate P_0 to the origin:
2. Align the normal vector **N** with the vector **K** normal to the xy plane.
3. Perform the mirror reflection in the xy plane (Prob. 6.6).
4. Reverse steps 1 and 2.

So, with translation vector $\mathbf{V} = -x_0\mathbf{I} - y_0\mathbf{J} - z_0\mathbf{K}$

$$M_{\mathbf{N},P_0} = T_{\mathbf{V}}^{-1} \cdot A_{\mathbf{N}}^{-1} \cdot M \cdot A_{\mathbf{N}} \cdot T_{\mathbf{V}}$$

Here, $A_{\mathbf{N}}$ is the alignment matrix defined in Prob. 6.2. So if the vector $\mathbf{N} = n_1\mathbf{I} + n_2\mathbf{J} + n_3\mathbf{K}$, then from Prob.

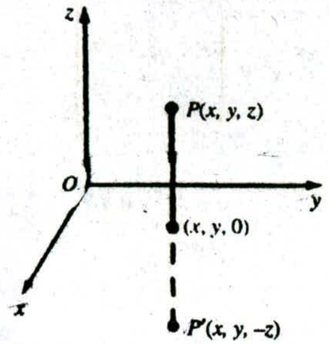


Fig. 6-8

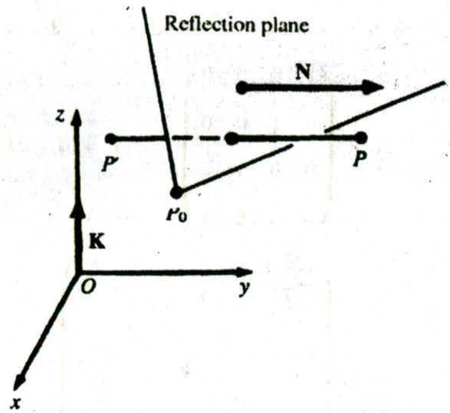


Fig. 6-9

6.2, with $|\mathbf{N}| = \sqrt{n_1^2 + n_2^2 + n_3^2}$ and $\lambda = \sqrt{n_2^2 + n_3^2}$, we find

$$A_{\mathbf{N}} = \begin{pmatrix} \frac{\lambda}{|\mathbf{N}|} & \frac{-n_1n_2}{\lambda|\mathbf{N}|} & \frac{-n_1n_3}{\lambda|\mathbf{N}|} & 0 \\ 0 & \frac{n_3}{\lambda} & \frac{-n_2}{\lambda} & 0 \\ \frac{n_1}{|\mathbf{N}|} & \frac{n_2}{|\mathbf{N}|} & \frac{n_3}{|\mathbf{N}|} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad A_{\mathbf{N}}^{-1} = \begin{pmatrix} \frac{\lambda}{|\mathbf{N}|} & 0 & \frac{n_1}{|\mathbf{N}|} & 0 \\ \frac{-n_1n_2}{\lambda|\mathbf{N}|} & \frac{n_3}{\lambda} & \frac{n_2}{|\mathbf{N}|} & 0 \\ \frac{-n_1n_3}{\lambda|\mathbf{N}|} & \frac{-n_2}{\lambda} & \frac{n_3}{|\mathbf{N}|} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In addition

$$T_{\mathbf{V}} = \begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad T_{\mathbf{V}}^{-1} = \begin{pmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Finally, from Prob. 6.6, the homogeneous form of M is

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

6.8 Find the matrix for mirror reflection with respect to the plane passing through the origin and having a normal vector whose direction is $\mathbf{N} = \mathbf{I} + \mathbf{J} + \mathbf{K}$.

SOLUTION

From Prob. 6.7, with $P_0(0, 0, 0)$ and $\mathbf{N} = \mathbf{I} + \mathbf{J} + \mathbf{K}$, we find $|\mathbf{N}| = \sqrt{3}$ and $\lambda = \sqrt{2}$. Then

$$T_{\mathbf{V}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (\mathbf{V} = 0\mathbf{I} + 0\mathbf{J} + 0\mathbf{K}) \qquad T_{\mathbf{V}}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_{\mathbf{N}} = \begin{pmatrix} \frac{\sqrt{2}}{\sqrt{3}} & \frac{-1}{\sqrt{2}\sqrt{3}} & \frac{-1}{\sqrt{2}\sqrt{3}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad A_{\mathbf{N}}^{-1} = \begin{pmatrix} \frac{\sqrt{2}}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}} & 0 \\ \frac{-1}{\sqrt{2}\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & 0 \\ \frac{-1}{\sqrt{2}\sqrt{3}} & \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The reflection matrix is

$$M_{\mathbf{N},O} = T_{\mathbf{V}}^{-1} \cdot A_{\mathbf{N}}^{-1} \cdot M \cdot A_{\mathbf{N}} \cdot T_{\mathbf{V}}$$

$$= \begin{pmatrix} \frac{1}{3} & -\frac{2}{3} & -\frac{2}{3} & 0 \\ -\frac{2}{3} & \frac{1}{3} & -\frac{2}{3} & 0 \\ -\frac{2}{3} & -\frac{2}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Supplementary Problems

6.9 Align the vector $\mathbf{V} = \mathbf{I} + \mathbf{J} + \mathbf{K}$ with the vector $\mathbf{K}$.

6.10 Find a transformation which aligns the vector $\mathbf{V} = \mathbf{I} + \mathbf{J} + \mathbf{K}$ with the vector $\mathbf{N} = 2\mathbf{I} - \mathbf{J} - \mathbf{K}$.

6.11 Show that the alignment transformation satisfies the relation $A_{\mathbf{V}}^{-1} = A_{\mathbf{V}}^{T}$.

6.12 Show that the alignment transformation $A_{\mathbf{V},\mathbf{N}}$ is equivalent to a rotation of $\theta°$ about an axis having the direction of the vector $\mathbf{V} \times \mathbf{N}$ and passing through the origin (see Fig. 6-10). Here θ is the angle between vectors $\mathbf{V}$ and $\mathbf{N}$.

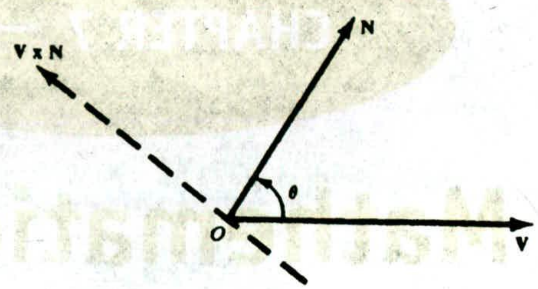


Fig. 6-10

6.13 How can scaling with respect to a point $P_0(x_0, y_0, z_0)$ be defined in terms of scaling with respect to the origin?

CHAPTER 7

Mathematics of Projection

Needless to say, there are fundamental differences between the true three-dimensional world and its pictorial description. For centuries, artists, engineers, designers, drafters, and architects have tried to come to terms with the difficulties and constraints imposed by the problem of representing a three-dimensional object or scene in a two-dimensional medium—the problem of *projection*. The implementers of a computer graphics system face the same challenge.

Projection can be defined as a mapping of point $P(x, y, z)$ onto its image $P'(x', y', z')$ in the *projection plane* or *view plane*, which constitutes the display surface (see Fig. 7-1). The mapping is determined by a projection line called the *projector* that passes through P and intersects the view plane. The intersection point is P'.

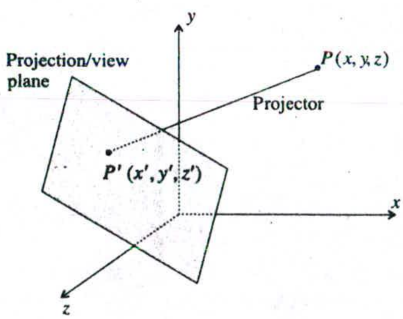


Fig. 7-1 The problem of projection.

The result of projecting an object is dependent on the spatial relationship among the projectors that project the points on the object, and the spatial relationship between the projectors and the view plane (see Sec. 7.1). An important observation is that projection preserves lines. That is, the line joining the projected images of the endpoints of the original line is the same as the projection of that line.

The two basic methods of projection—*perspective* and *parallel*—are designed to solve the basic but mutually exclusive problems of pictorial representation: showing an object as it appears and preserving its true size and shape. We characterize each method and introduce the mathematical description of the projection process in Sec. 7.2 and 7.3, respectively.

7.1 TAXONOMY OF PROJECTION

We can construct different projections according to the view that is desired.

Figure 7-2 provides a taxonomy of the families of perspective and parallel projections. Some projections have names—cavalier, cabinet, isometric, and so on. Other projections qualify the main type of projection—one principal vanishing-point perspective, and so forth.

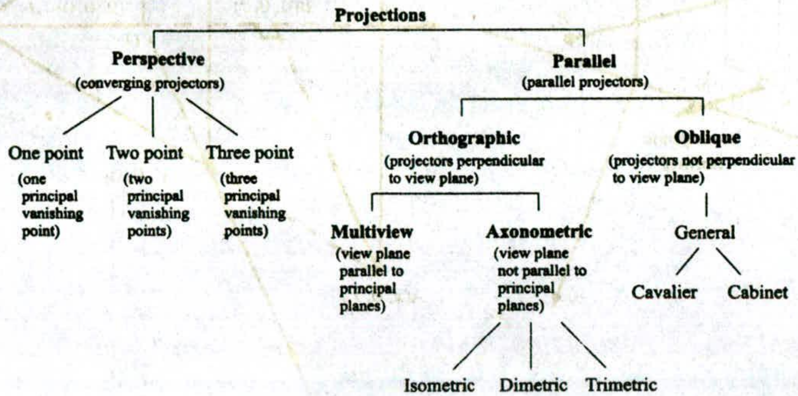


Fig. 7-2 Taxonomy of projection.

7.2 PERSPECTIVE PROJECTION

Basic Principles

The techniques of perspective projection are generalizations of the principles used by artists in preparing perspective drawings of three-dimensional objects and scenes. The eye of the artist is placed at the *center of projection*, and the canvas, or more precisely the plane containing the canvas, becomes the view plane. An image point is determined by a projector that goes from an object point to the center of projection (see Fig. 7-3).

Perspective drawings are characterized by perspective foreshortening and vanishing points. *Perspective foreshortening* is the illusion that objects and lengths appear smaller as their distance from the center of projection increases. The illusion that certain sets of parallel lines appear to meet at a point is another feature of perspective drawings. These points are called *vanishing points*. *Principal vanishing points* are formed by the apparent intersection of lines parallel to one of the three principal x, y, or z axes. The number of principal vanishing points is determined by the number of principal axes intersected by the view plane (Prop. 7.7).

Mathematical Description of a Perspective Projection

A perspective transformation is determined by prescribing a center of projection and a view plane. The view plane is determined by its *view reference point* R_0 and *view plane normal* N. The *object point* P is located in world coordinates at (x, y, z). The problem is to determine the *image point* coordinates $P'(x', y', z')$ (see Fig. 7-3).

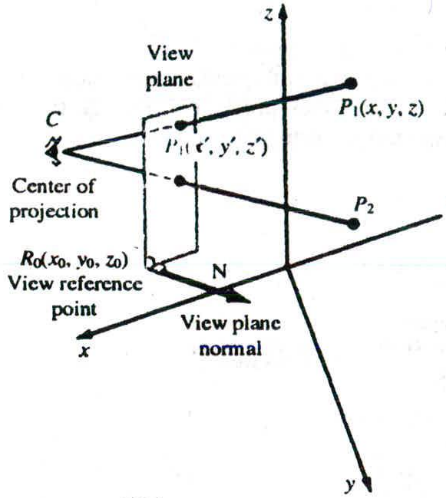


Fig. 7-3

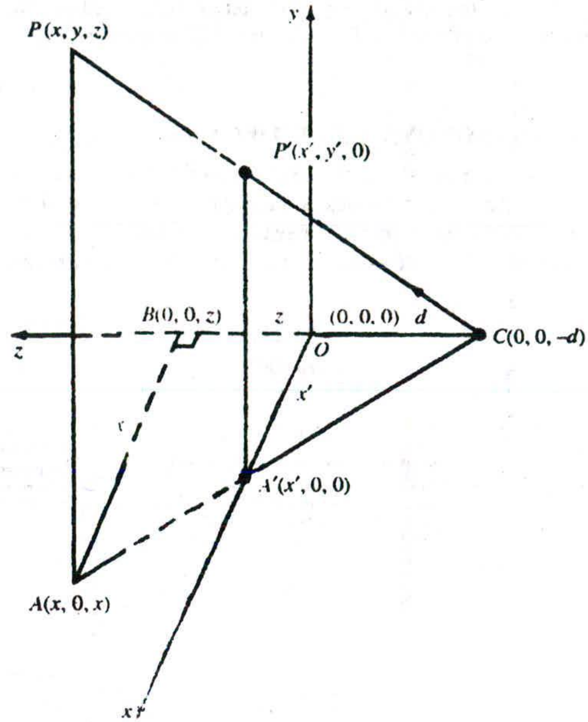


Fig. 7-4

EXAMPLE 1. The standard perspective projection is shown in Fig. 7-4. Here, the view plane is the xy plane, and the center of projection is taken as the point $C(0, 0, -d)$ on the negative z axis.

Using similar triangles ABC and $A'OC$, we find

$$x' = \frac{d \cdot x}{z + d} \qquad y' = \frac{d \cdot y}{z + d} \qquad z' = 0$$

The perspective transformation between object and image point is nonlinear and so cannot be represented as a 3×3 matrix transformation. However, if we use homogeneous coordinates, the perspective transformation can be represented as a 4×4 matrix:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} d \cdot x \\ d \cdot y \\ 0 \\ z + d \end{pmatrix} = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

The general form of a perspective transformation is developed in Prob. 7.5.

Perspective Anomalies

The process of constructing a perspective view introduces certain anomalies which enhance realism in terms of depth cues but also distort actual sizes and shapes.

1. *Perspective foreshortening*. The farther an object is from the center of projection, the smaller it appears (i.e. its projected size becomes smaller). Refer to Fig. 7-5.

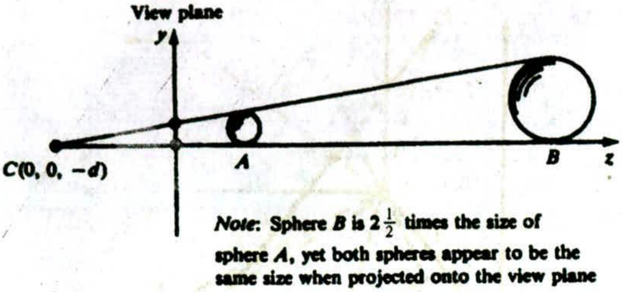


Fig. 7-5

2. *Vanishing points*. Projections of lines that are not parallel to the view plane (i.e. lines that are not perpendicular to the view plane normal) appear to meet at some point on the view plane. A common manifestation of this anomaly is the illusion that railroad tracks meet at a point on the horizon.

EXAMPLE 2. For the standard perspective projection, the projections L_1' and L_2' of parallel lines L_1 and L_2 having the direction of the vector **K** appear to meet at the origin (Prob. 7.8). Refer to Fig. 7-6.

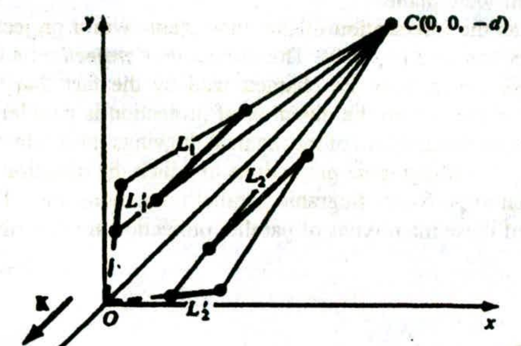


Fig. 7-6

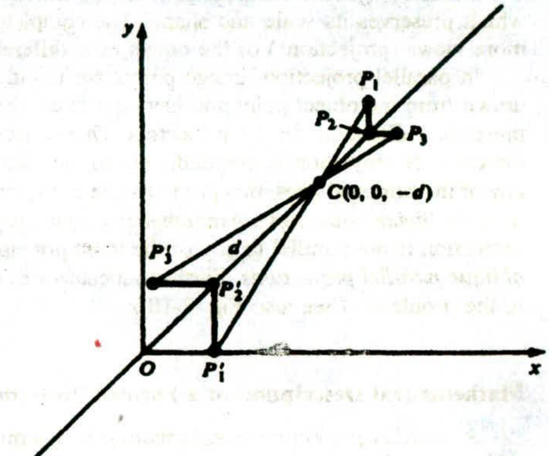


Fig. 7-7

3. *View confusion*. Objects behind the center of projection are projected upside down and backward onto the view plane. Refer to Fig. 7-7.
4. *Topological distortion*. Consider the plane that passes through the center of projection and is parallel to the view plane. The points of this plane are projected to infinity by the perspective transformation. In particular, a finite line segment joining a point which lies in front of the viewer to a point in back of the viewer is actually projected to a broken line of infinite extent (Prob. 7.2) (see Fig. 7-8).

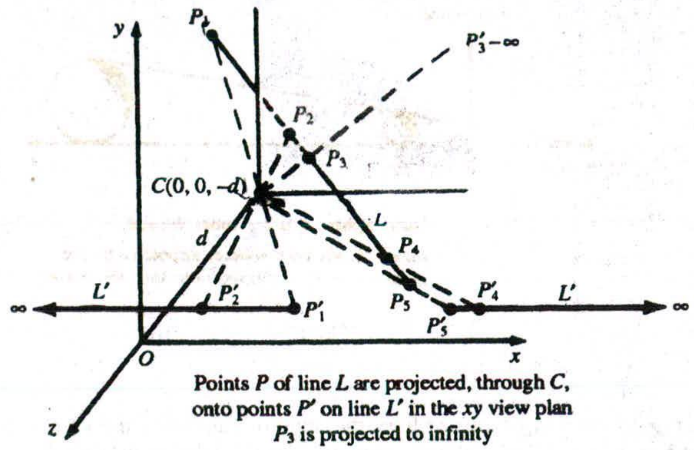


Fig. 7-8

7.3 PARALLEL PROJECTION

Basic Principles

Parallel projection methods are used by drafters and engineers to create working drawings of an object which preserves its scale and shape. The complete representation of these details often requires two or more views (projections) of the object onto different view planes.

In parallel projection, image points are found as the intersection of the view plane with a projector drawn from the object point and having a fixed direction (see Fig. 7-9). The *direction of projection* is the prescribed direction for all projectors. *Orthographic projections* are characterized by the fact that the direction of projection is perpendicular to the view plane. When the direction of projection is parallel to any of the principal axes, this produces the front, top, and side views of mechanical drawings (also referred to as *multiview drawings*). *Axonometric projections* are orthographic projections in which the direction of projection is not parallel to any of the three principal axes. Nonorthograhic parallel projections are called *oblique parallel projections*. Further subcategories of these main types of parallel projection are described in the problems. (See also Fig. 7-10.)

Mathematical Description of a Parallel Projection

A *parallel projective transformation* is determined by prescribing a *direction of projection vector* **V** and a view plane. The view plane is specified by its view plane reference point R_0, and view plane normal **N**. The object point P is located at (x, y, z) in world coordinates. The problem is to determine the image point coordinates $P'(x', y', z')$. See Fig. 7-9.

If the projection vector **V** has the direction of the view plane normal **N**, the projection is said to be *orthographic*. Otherwise it is called *oblique* (see Fig. 7-10).

Some common subcategories of orthographic projections are:

1. *Isometric*—the direction of projection makes equal angles with all of the three principal axes (Prob. 7.14).
2. *Dimetric*—the direction of projection makes equal angles with exactly two of the principal axes (Prob. 7.15).
3. *Trimetric*—the direction of projection makes unequal angles with the three principal axes.

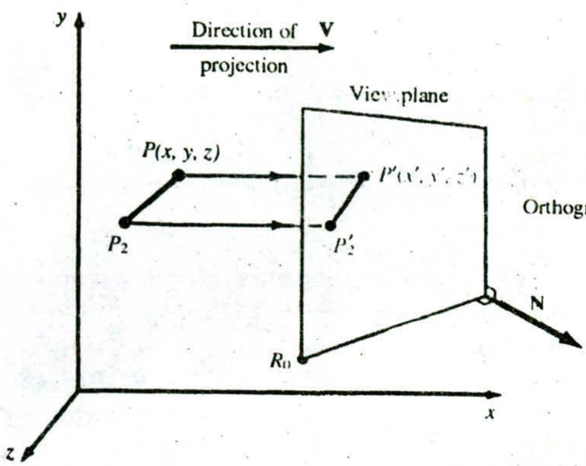


Fig. 7-9

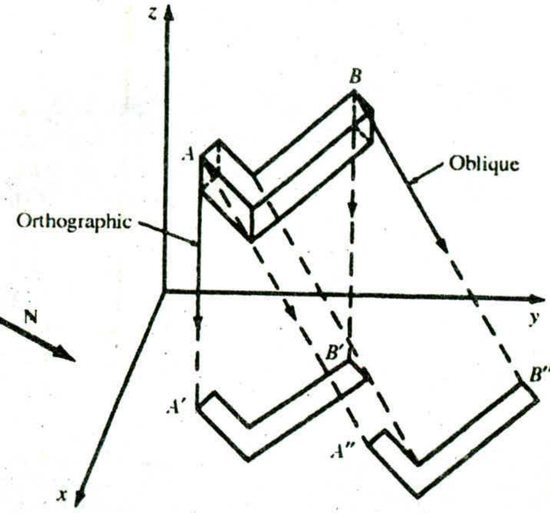


Fig. 7-10

Some common subcategories of oblique projections are:

1. *Cavalier*—the direction of projection is chosen so that there is no foreshortening of lines perpendicular to the *xy* plane (Prob. 7.13).
2. *Cabinet*—the direction of projection is chosen so that lines perpendicular to the *xy* planes are foreshortened by half their lengths (Prob. 7.13).

EXAMPLE 3. For orthographic projection onto the *xy* plane, from Fig. 7-11 it is easy to see that

$$Par_{\mathbf{K}}: \begin{cases} x' = x \\ y' = y \\ z' = 0 \end{cases}$$

The matrix form of $Par_{\mathbf{K}}$ is

$$Par_{\mathbf{K}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The general parallel projective transformation is derived in Prob. 7.11.

Solved Problems

7.1 The unit cube (Fig. 7-12) is projected onto the *xy* plane. Note the position of the *x*, *y*, and *z* axes. Draw the projected image using the standard perspective transformation with (*a*) $d = 1$ and (*b*) $d = 10$, where *d* is distance from the view plane.

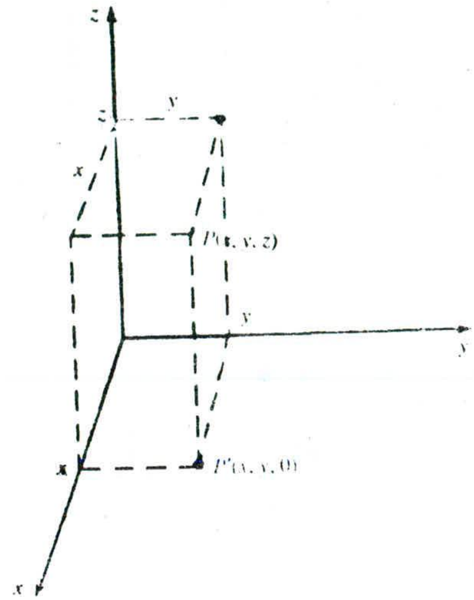


Fig. 7-11

SOLUTION

We represent the unit cube in terms of the, homogeneous coordinates of its vertices:

$$\mathbf{V} = (ABCDEFGH) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

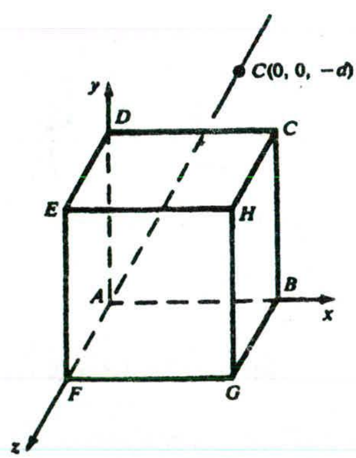


Fig. 7-12

From Example 1 the standard perspective matrix is

$$Per_{\mathbf{K}} = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d \end{pmatrix}$$

(a) With $d = 1$, the projected coordinates are found by applying the matrix $Per_{\mathbf{K}}$ to the matrix of coordinates **V**. Then

$$Per_{\mathbf{K}} \cdot \mathbf{V} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \end{pmatrix}$$

If these homogeneous coordinates are changed to three-dimensional coordinates, the projected image has coordinates:

$$\begin{array}{ll} A' = (0, 0, 0) & E' = (0, \tfrac{1}{2}, 0) \\ B' = (1, 0, 0) & F' = (0, 0, 0) \\ C' = (1, 1, 0) & G' = (\tfrac{1}{2}, 0, 0) \\ D' = (0, 1, 0) & H' = (\tfrac{1}{2}, \tfrac{1}{2}, 0) \end{array}$$

We draw the projected image by preserving the edge connections of the original object (see Fig. 7-13). [Note the vanishing point at (0, 0, 0).]

(b) With $d = 10$, the perspective matrix is

$$Per_{\mathbf{K}} = \begin{pmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 10 \end{pmatrix}$$

Then

$$Per_{\mathbf{K}} \cdot \mathbf{V} = \begin{pmatrix} 0 & 10 & 10 & 0 & 0 & 0 & 10 & 10 \\ 0 & 0 & 10 & 10 & 10 & 0 & 0 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 10 & 10 & 10 & 11 & 11 & 11 & 11 \end{pmatrix}$$

is the matrix image coordinates in homogeneous form. The projected image coordinates are then

$$\begin{array}{ll} A' = (0, 0, 0) & E' = (0, \tfrac{10}{11}, 0) \\ B' = (1, 0, 0) & F' = (0, 0, 0) \\ C' = (1, 1, 0) & G' = (\tfrac{10}{11}, 0, 0) \\ D' = (0, 1, 0) & H' = (\tfrac{10}{11}, \tfrac{10}{11}, 0) \end{array}$$

Note the different perspectives of the face $E'F'G'H'$ in Figs. 7-13 and 7-14. [To a viewer standing at the center of projection $(0, 0, -d)$, this face is the back face of the unit cube.]

7.2 Under the standard perspective transformation $Per_{\mathbf{K}}$, what is the projected image of (*a*) a point in the plane $z = -d$ and (*b*) the line segment joining $P_1(-1, 1, -2d)$ to $P_2(2, -2, 0)$? (See Fig. 7-15.)

SOLUTION

(*a*) The plane $z = -d$ is the plane parallel to the *xy* view plane and located at the center of projection $C(0, 0, -d)$. If $P(x, y, -d)$ is any point in this plane, the line of projection $\overline{CP}$ does not intersect the *xy* view plane. We then say that P is projected out to infinity (∞).

(*b*) The line $\overline{P_1P_2}$ passes through the plane $z = -d$. Writing the equation of the line (App. 2), we have

$$x = -1 + 3t \qquad y = 1 - 3t \qquad z = -2d + 2dt$$

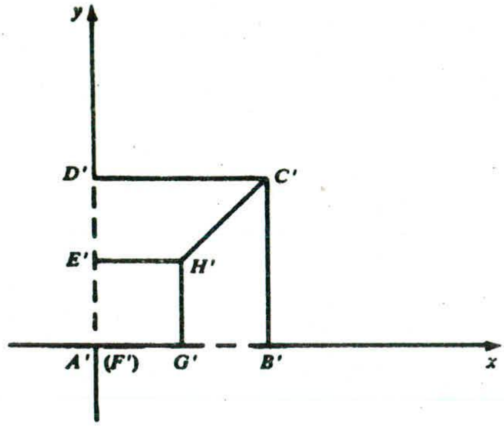


Fig. 7-13

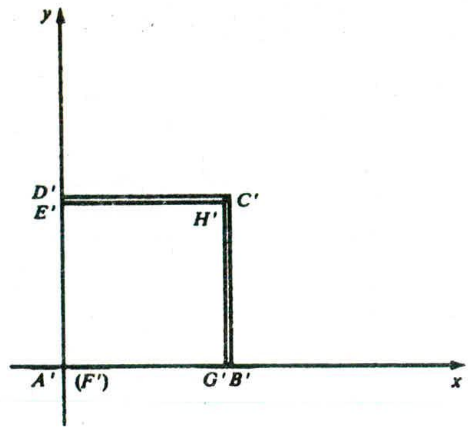


Fig. 7-14

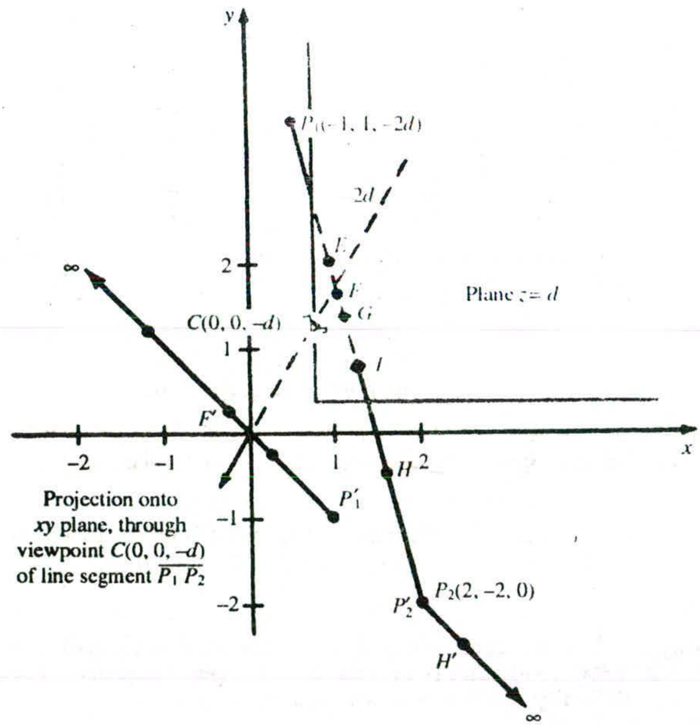


Fig. 7-15

We see that at $t = \frac{1}{2}: x = \frac{1}{2}, y = -\frac{1}{2}$, and $z = -d$. These are the coordinates of the intersection point I. We now describe the perspective projection of this line segment.

Applying the standard projection to the equation of the line, we find

$$\begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d \end{pmatrix} \begin{pmatrix} -1+3t \\ 1-3t \\ -2d+2dt \\ 1 \end{pmatrix} = \begin{pmatrix} -d+3dt \\ d-3dt \\ 0 \\ -d+2dt \end{pmatrix}$$

Changing from homogeneous to three-dimensional coordinates, the equations of the projected line segment are

$$x = \frac{-d+3dt}{-d+2dt} = \frac{-1+3t}{-1+2t} \qquad y = \frac{d-3dt}{-d+2dt} = \frac{1-3t}{-1+2t} \qquad z = 0$$

(In App. 1, Prob. A1.12, it is shown that this is the equation of a line.) When $t = 0$, then $x = 1$ and $y = -1$. These are the coordinates of the projection P_1' of point P_1. When $t = 1$, it follows that $x = 2$ and $y = -2$ (the coordinates of the projection P_2' of point P_2). However, when $t = \frac{1}{2}$, the denominator is 0. Thus this line segment "passes" through the point at infinity in joining $P_1'(1, -1)$ to $P_2'(2, -2)$. In other words, when a line segment joining endpoints P_1 and P_2 passes through the plane containing the center of projection and which is parallel to the view plane, the projection of this line segment is *not* the simple line segment joining the projected endpoints P_1' and P_2'. (See also Prob. A1.13 in App. 1.)

7.3 Using the origin as the center of projection, derive the perspective transformation onto the plane passing through the point $R_0(x_0, y_0, z_0)$ and having the normal vector $\mathbf{N} = n_1 I + n_2 J + n_3 K$.

SOLUTION

Let $P(x, y, z)$ be projected onto $P'(x', y', z')$. From Fig. 7-16, the vectors $\overline{\mathbf{PO}}$ and $\overline{\mathbf{P'O}}$ have the same direction. Thus there is a number α so that $\overline{\mathbf{P'O}} = \alpha\overline{\mathbf{PO}}$. Comparing components, we have

$$x' = \alpha x \qquad y' = \alpha y \qquad z' = \alpha z$$

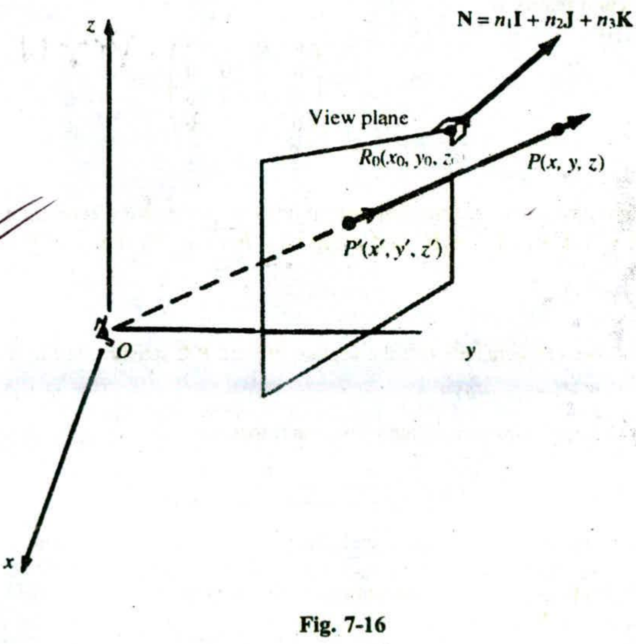


Fig. 7-16

We now find the value of α. Since any point $P'(x', y', z')$ lying on the plane satisfies the equation (App. 2)

$$n_1x' + n_2y' + n_3z' = d_0$$

(where $d_0 = n_1x_0 + n_2y_0 + n_3z_0$), substitution of $x' = \alpha x$, $y' = \alpha y$, and $z' = \alpha z$ into this equation gives

$$\alpha = \frac{d_0}{n_1x + n_2y + n_3z}$$

This projection transformation cannot be represented as a 3×3 matrix transformation. However, by using the homogeneous coordinate representation for three-dimensional points, we can write the projection transformation as a 4×4 matrix:

$$Per_{\mathbf{N},R_0} = \begin{pmatrix} d_0 & 0 & 0 & 0 \\ 0 & d_0 & 0 & 0 \\ 0 & 0 & d_0 & 0 \\ n_1 & n_2 & n_3 & 0 \end{pmatrix}$$

Application of this matrix to the homogeneous representation $P(x, y, z, 1)$ of points P gives $P'(d_0x, d_0y, d_0z, n_1x + n_2y + n_3z)$, which is the homogeneous representation of $P'(x', y', z')$ found above.

7.4 Find the perspective projection onto the view plane $z = d$ where the center of projection is the origin $(0, 0, 0)$.

SOLUTION

The plane $z = d$ is parallel to the xy plane (and d units away from it). Thus the view plane normal vector $\mathbf{N}$ is the same as the normal vector $\mathbf{K}$ to the xy plane, that is, $\mathbf{N} = \mathbf{K}$. Choosing the view reference point as $R_0(0, 0, d)$, then from Prob. 7.3, we identify the parameters

$$\mathbf{N}(n_1, n_2, n_3) = (0, 0, 1) \qquad R_0(x_0, y_0, z_0) = (0, 0, d)$$

So

$$d_0 = n_1x_0 + n_2y_0 + n_3z_0 = d$$

and then the projection matrix is

$$Per_{\mathbf{K},R_0} = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

7.5 Derive the general perspective transformation onto a plane with reference point $R_0(x_0, y_0, z_0)$, normal vector $\mathbf{N} = n_1\mathbf{I} + n_2\mathbf{J} + n_3\mathbf{K}$, and using $C(a, b, c)$ as the center of projection. Refer to Fig. 7-17.

SOLUTION

As in Prob. 7.3, we can conclude that the vectors $\overline{\mathbf{PC}}$ and $\overline{\mathbf{P'C}}$ satisfy (see Fig. 7-17) $\overline{\mathbf{P'C}} = \alpha\overline{\mathbf{PC}}$. Then

$$x' = \alpha(x - a) + a \qquad y' = \alpha(y - b) + b \qquad z' = \alpha(z - c) + c$$

Also, we find (by using the equation of the view plane) that

$$\alpha = \frac{d}{n_1(x - a) + n_2(y - b) + n_3(z - c)}$$

[i.e. $P'(x', y', z')$ is on the view plane and thus satisfies the view plane equation $n_1(x' - x_0) + n_2(y' - y_0) + n_3(z' - z_0) = 0$]. Here, $d = (n_1x_0 + n_2y_0 + n_3z_0) - (n_1a + n_2b + n_3c)$.

From App. 2, Prob. A2.13, d is proportional to the distance D from the view plane to the center of projection, that is, $d = \pm|\mathbf{N}|D$.

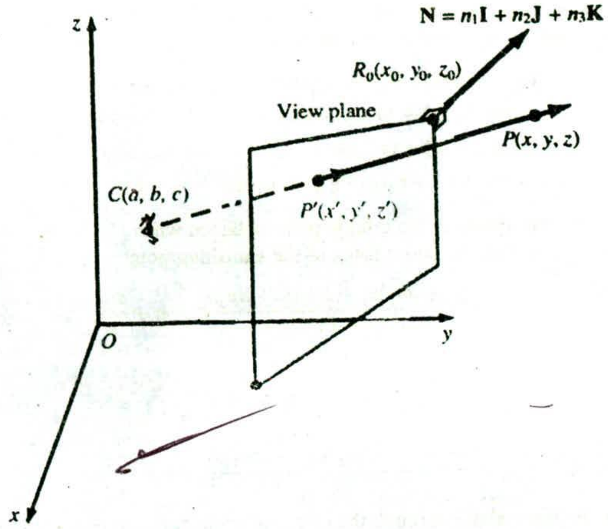


Fig. 7-17

To find the homogeneous coordinate matrix representation, it is easiest to proceed as follows:

1. Translate so that the center of projection C lies at the origin. Now $R_0' = (x_0 - a, y_0 - b, z_0 - c)$ becomes the reference point of the translated plane (the normal vector is unchanged by translation).
2. Project onto the translated plane using the origin as the center of projection by constructing the transformation $Per_{\mathbf{N},R_0'}$ (Prob. 7.3).
3. Translate back.

Introducing the intermediate quantities

$$d_0 = n_1x_0 + n_2y_0 + n_3z_0 \qquad \text{and} \qquad d_1 = n_1a + n_2b + n_3c$$

we obtain $d = d_0 - d_1$, and so $Per_{\mathbf{N},R_0,C} = T_C \cdot Per_{\mathbf{N},R_0'} \cdot T_{-C}$. Then with R_0' used as the reference point in constructing the projection $P_{\mathbf{N},R_0'}$,

$$Per_{\mathbf{N},R_0,C} = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ n_1 & n_2 & n_3 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -a \\ 0 & 1 & 0 & -b \\ 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} d + an_1 & an_2 & an_3 & -ad_0 \\ bn_1 & d + bn_2 & bn_3 & -bd_0 \\ cn_1 & cn_2 & d + cn_3 & -cd_0 \\ n_1 & n_2 & n_3 & -d_1 \end{pmatrix}$$

7.6 Find the (*a*) vanishing points for a given perspective transformation in the direction given by a vector $\mathbf{U}$ and (*b*) principal vanishing points.

SOLUTION

(*a*) The family of (parallel) lines having the direction of $\mathbf{U} = u_1\mathbf{I} + u_2\mathbf{J} + u_3\mathbf{K}$ can be written in parametric form as

$$x = u_1t + p \qquad y = u_2t + q \qquad z = u_3t + r$$

where $P(p, q, r)$ is any point (see App. 2). Application of the perspective transformation (Prob. 7.5) to the homogeneous point $(x, y, z, 1)$ produces the result (x', y', z', h), where

$$\begin{aligned} x' &= (d + an_1)(u_1 t + p) + an_2(u_2 t + q) + an_3(u_3 t + r) - ad_0 \\ y' &= bn_1(u_1 t + p) + (d + bn_2)(u_2 t + q) + bn_3(u_3 t + r) - bd_0 \\ z' &= cn_1(u_1 t + p) + cn_2(u_2 t + q) + (d + cn_3)(u_3 t + r) - cd_0 \\ h &= n_1(u_1 t + p) + n_2(u_2 t + q) + n_3(u_3 t + r) - d_1 \end{aligned}$$

The vanishing point corresponds to the infinite point obtained when $t = \infty$. So after dividing x', y', and z' by h, we let $t \to \infty$ to find the coordinates of the vanishing point:

$$x_u = \frac{(d + an_1)u_1 + an_2 u_2 + an_3 u_3}{k} = a + \frac{du_1}{k}$$

(Here, $k = \mathbf{N} \cdot \mathbf{U} = n_1 u_1 + n_2 u_2 + n_3 u_3$.)

$$y_u = \frac{bn_1 u_1 + (d + bn_2)u_2 + bn_3 u_3}{k} = b + \frac{du_2}{k}$$

$$z_u = \frac{cn_1 u_1 + cn_2 u_2 + (d + cn_3)u_3}{k} = c + \frac{du_3}{k}$$

This point lies on the line passing through the center of projection and parallel to the vector **U** (see Fig. 7-18). Note that $k = 0$ only when **U** is parallel to the projection plane, in which case there is no vanishing point.

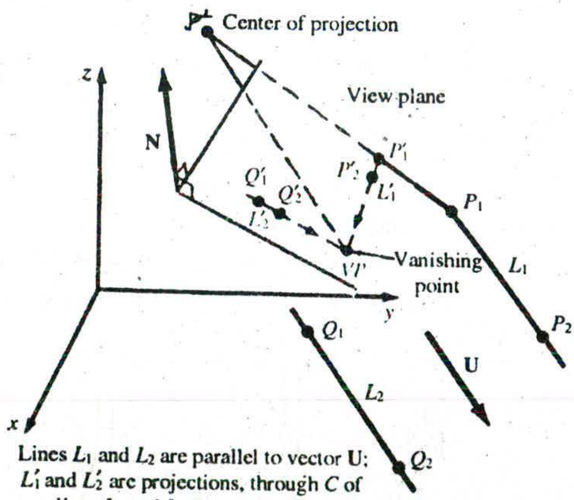


Fig. 7-18

(*b*) The principal vanishing points P_1, P_2, and P_3 correspond to the vector directions **I**, **J**, and **K**. In these cases

$$P_1: \begin{cases} x_1 = a + \dfrac{d}{n_1} \\ y_1 = b \\ z_1 = c \end{cases} \qquad P_2: \begin{cases} x_2 = a \\ y_2 = b + \dfrac{d}{n_2} \\ z_2 = c \end{cases} \qquad P_3: \begin{cases} x_3 = a \\ y_3 = b \\ z_3 = c + \dfrac{d}{n_3} \end{cases}$$

(Recall from Prob. 7.5 that a, b, c are the coordinates of the center of projection. Also, n_1, n_2, n_3 are the components of the view plane normal vector and d is proportional to the distance D from the view plane to the center of projection.) (*Note*: If any of the components of the normal vector are zero, say, $n_1 = 0$, then $k = \mathbf{N} \cdot \mathbf{I} = 0$, and there is no principal vanishing point in the $\mathbf{I}$ direction.)

7.7 Describe the (*a*) one-principal-vanishing-point perspective, (*b*) two-principal-vanishing-point perspective, and (*c*) three-principal-vanishing-point perspective.

SOLUTION

(*a*) The one-principal-vanishing-point perspective occurs when the projection plane is perpendicular to one of the principal axes (x, y, or z). Assume that it is the z axis. In this case the view plane normal vector $\mathbf{N}$ is the vector $\mathbf{K}$, and from prob. 7.6, the principal vanishing point is

$$P_3: \begin{cases} x_3 = a \\ y_3 = b \\ z_3 = c + \dfrac{d}{n_3} \end{cases}$$

(*b*) The two-principal-vanishing-point projection occurs when the projection plane intersects exactly two of the principal axes. Refer to Fig. 7-19, which is a perspective drawing with two principal vanishing points. In the case where the projection plane intersects the x and y axes, for example, the normal vector satisfies the relatioinship $\mathbf{N} \cdot \mathbf{K} = 0$ or $n_3 = 0$, and so the principal vanishing points are

$$P_1: \begin{cases} x_1 = a + \dfrac{d}{n_1} \\ y_1 = b \\ z_1 = c \end{cases} \qquad P_2: \begin{cases} x_2 = a \\ y_2 = b + \dfrac{d}{n_2} \\ z_2 = c \end{cases}$$

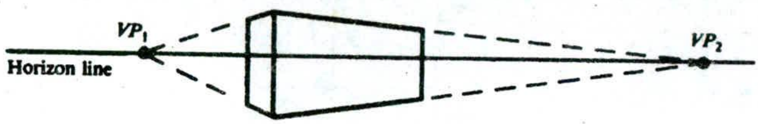


Fig. 7-19

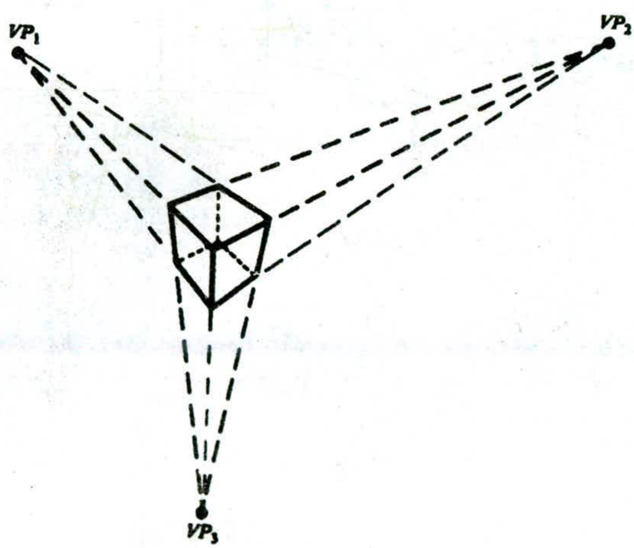


Fig. 7-20

(c) The three-vanishing-point perspective projection occurs when the projection plane intersects all three of the principal axes—x, y, and z axes. Refer to Fig. 7-20, which is a perspective drawing with three principal vanishing points. In this case, the principal vanishing points are points P_1, P_2, and P_3 from Prob. 7.6(b).

7.8 What are the principal vanishing points for the standard perspective transforamtion?

SOLUTION

In this case, the view plane normal $\mathbf{N}$ is the vector $\mathbf{K}$. From Prob. 7.6(b), since $\mathbf{N}\cdot\mathbf{I}=0$ and $\mathbf{N}\cdot\mathbf{J}=0$, there are no vanishing points in the directions $\mathbf{I}$ and $\mathbf{J}$. On the other hand, $\mathbf{N}\cdot\mathbf{K}=\mathbf{K}\cdot\mathbf{K}=1$. Thus there is only one principal vanishing point, and it is in the $\mathbf{K}$ direction. From Prob. 7.7(a), the coordinates of the principal vanishing point VP in the $\mathbf{K}$ direction are

$$x=a=0 \qquad y=b=0 \qquad z=-d+\frac{d}{1}=0$$

So $VP=(0,0,0)$ is the principal vanishing point.

7.9 An artist constructs a two-vanishing-point perspective by locating the vanishing points VP_1 and VP_2 on a given horizon line in the view plane. The horizon line is located by its height h above the ground (Fig. 7-21). Construct the corresponding perspective projection transformation for the cube shown in Fig. 7-21.

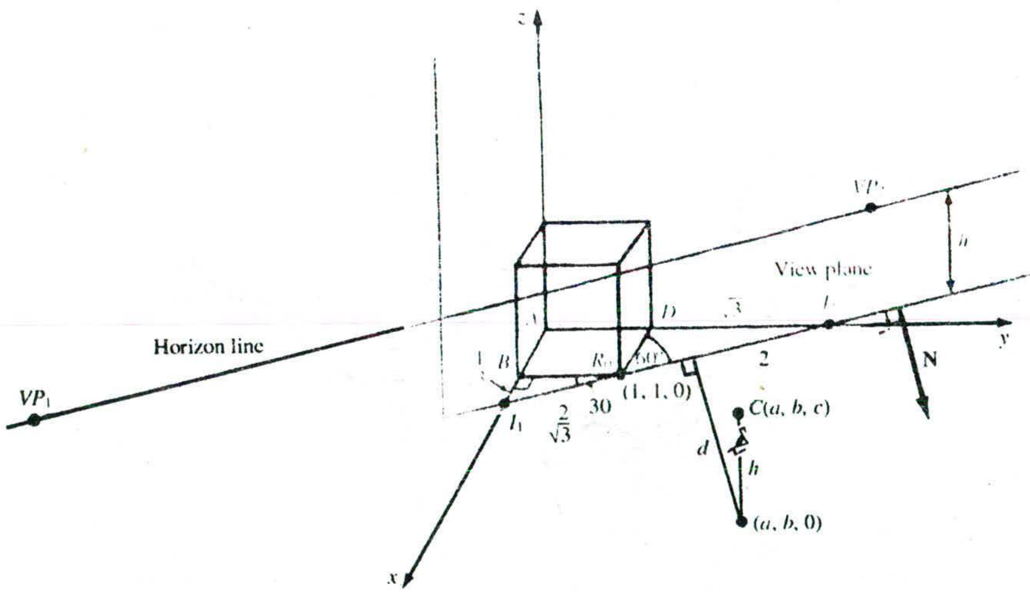


Fig. 7-21

SOLUTION

A two-principal-vanishing-point perspective must intersect two axes, say, x and y. We locate the view plane at the point $R_0(1,1,0)$ so that it makes angles of 30° and 60° with the corresponding faces of the cube (see Fig. 7-21). In this plane we locate the horizon line a given height h above the "ground" (the xy plane).

The vanishing points VP_1 and VP_2 are located on this horizon line. To construct the perspective transformation, we need to find the normal vector $\mathbf{N} = n_1\mathbf{I} + n_2\mathbf{J} + n_3\mathbf{K}$ of the view plane, the coordinates $C(a, b, c)$ of the center of projection, and the view parameters d_0, d_1, and d (Prob. 7.5). To calculate the coordinates of the vanishing points, we first find the equation of the horizon line. Let I_1 and I_2 be the points of intersection of the view plane and the x and y axes. The horizon line is parallel to the line $\overline{I_1I_2}$ and lies h units above it.

From triangles I_1BR_0 and I_2DR_0, we find

$$I_1 = \left(1 + \frac{1}{\sqrt{3}}, 0, 0\right) = \left(\frac{1+\sqrt{3}}{\sqrt{3}}, 0, 0\right) \quad \text{and} \quad I_2 = (0, 1+\sqrt{3}, 0)$$

The equation of the line through I_1 and I_2 (App. 2) is

$$x = \left(\frac{1+\sqrt{3}}{\sqrt{3}}\right) - \left(\frac{1+\sqrt{3}}{\sqrt{3}}\right)t \qquad y = (1+\sqrt{3})t \qquad z = 0$$

This line lies in the view plane. So if the equation of the horizon line is then taken to be a line parallel to this line and h units above it, the horizon line is guaranteed to be in the view plane. The equation of the horizon line is then

$$x = \left(\frac{1+\sqrt{3}}{\sqrt{3}}\right)(1-t) \qquad y = (1+\sqrt{3})t \qquad z = h$$

The vanishing points VP_1 and VP_2 are chosen to lie on the horizon line. So VP_1 has coordinates of the form

$$VP_1 = \left[\left(\frac{1+\sqrt{3}}{\sqrt{3}}\right)(1-t_1), (1+\sqrt{3})t_1, h\right] \quad \text{and} \quad VP_2 = \left[\left(\frac{1+\sqrt{3}}{\sqrt{3}}\right)(1-t_2), (1+\sqrt{3})t_2, h\right]$$

(Here, t_1 and t_2 are chosen so as to place the vanishing points at the desired locations.)

To find the normal vector $\mathbf{N}$ and the center of projection C, we use the equations in Prob. 7.6, part (b) for locating the vanishing points of a given perspective transformation. So

$$a + \frac{d}{n_1} = \left(\frac{1+\sqrt{3}}{\sqrt{3}}\right)(1-t_1) \quad \text{and} \quad a = \left(\frac{1+\sqrt{3}}{\sqrt{3}}\right)(1-t_2)$$

and

$$b = (1+\sqrt{3})t_1 \quad \text{and} \quad b + \frac{d}{n_2} = (1+\sqrt{3})t_2 \quad \text{and} \quad c = h$$

Using the values

$$a = \left(\frac{1+\sqrt{3}}{\sqrt{3}}\right)(1-t_2) \qquad b = (1+\sqrt{3})t_1 \qquad c = h$$

and then substituting, we find

$$\frac{d}{n_1} = \left(\frac{1+\sqrt{3}}{\sqrt{3}}\right)(t_2 - t_1) \tag{7.1}$$

and

$$\frac{d}{n_2} = (1+\sqrt{3})(t_2 - t_1) \tag{7.2}$$

Since the plane does not intersect the z axis, then $\mathbf{N} \cdot \mathbf{K} = 0$, or using components: $n_3 = 0$. Finally, we choose the normal vector $\mathbf{N}$ to be of unit length:

$$|\mathbf{N}| = \sqrt{n_1^2 + n_2^2 + n_3^2} = \sqrt{n_1^2 + n_2^2} = 1$$

From equations (7.1) and (7.2)

$$n_1 = \frac{d\sqrt{3}}{(1+\sqrt{3})(t_2 - t_1)} \qquad n_2 = \frac{d}{(1+\sqrt{3})(t_2 - t_1)}$$

So

$$|\mathbf{N}| = \sqrt{\frac{(d\sqrt{3})^2}{(1+\sqrt{3})^2(t_2-t_1)^2} + \frac{d^2}{(1+\sqrt{3})^2(t_2-t_1)^2}} = 1$$

or

$$\frac{2d}{(1+\sqrt{3})(t_2-t_1)} = 1 \qquad \text{and so} \qquad d = \frac{1+\sqrt{3}}{2}(t_2-t_1)$$

Also

$$n_1 = \frac{\sqrt{3}[(1+\sqrt{3})/2]}{1+\sqrt{3}} = \frac{\sqrt{3}}{2} \qquad \text{and} \qquad n_2 = \frac{(1+\sqrt{3})/2}{1+\sqrt{3}} = \frac{1}{2}$$

Finally, we have

$$d_1 = n_1 a + n_2 b + n_3 c = \left(\frac{\sqrt{3}}{2}\frac{1+\sqrt{3}}{\sqrt{3}}\right)(1-t_2) + \frac{1}{2}(1+\sqrt{3})t_1 = \frac{1+\sqrt{3}}{2}[1-(t_2-t_1)]$$

and

$$d_0 = d + d_1 = \frac{1+\sqrt{3}}{2}$$

From Prob. 7.5, the perspective transformation matrix is then

$$Per_{\mathbf{N},R_0,C} = \frac{1+\sqrt{3}}{2}\begin{pmatrix} 1-t_1 & \frac{1}{\sqrt{3}}(1-t_2) & 0 & -\left(\frac{1+\sqrt{3}}{\sqrt{3}}\right)(1-t_2) \\ \sqrt{3}t_1 & t_2 & 0 & -(1+\sqrt{3})t_1 \\ \frac{\sqrt{3}h}{1+\sqrt{3}} & \frac{h}{1+\sqrt{3}} & t_2-t_1 & -h \\ \frac{\sqrt{3}}{1+\sqrt{3}} & \frac{1}{1+\sqrt{3}} & 0 & -[1-(t_2-t_1)] \end{pmatrix}$$

In Chap. 8, Prob. 8.2, it is shown how to convert the transformed image of the cube into x, y coordinates for viewing.

7.10 Derive the equations of parallel projection onto the xy plane in the direction of projection $\mathbf{V} = a\mathbf{I} + b\mathbf{J} + c\mathbf{K}$.

SOLUTION

From Fig. 7-22 we see that the vectors $\mathbf{V}$ and $\overline{\mathbf{PP}'}$ have the same direction. This means that $\overline{\mathbf{PP}'} = k\mathbf{V}$. Comparing components, we see that

$$x' - x = ka \qquad y' - y = kb \qquad z' - z = kc$$

So

$$k = -\frac{z}{c} \qquad x' = x - \frac{a}{c}z \qquad \text{and} \qquad y' = y - \frac{b}{c}z$$

In 3 × 3 matrix form, this is

$$Par_{\mathbf{V}} = \begin{pmatrix} 1 & 0 & -\frac{a}{c} \\ 0 & 1 & -\frac{b}{c} \\ 0 & 0 & 0 \end{pmatrix}$$

and so $P' = Par_{\mathbf{V}} \cdot P$.

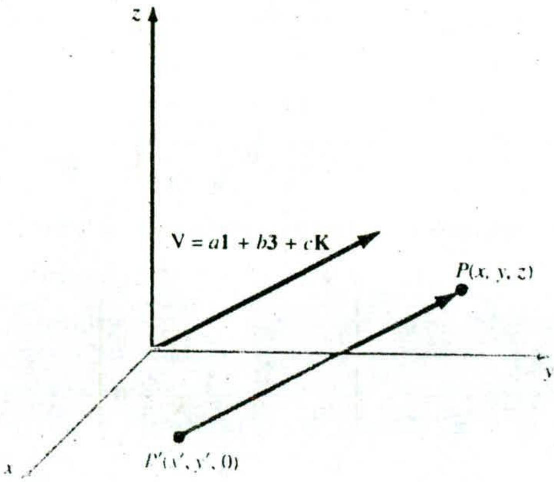


Fig. 7-22

7.11 Derive the general equation of parallel projection onto a given view plane in the direction of a given projector **V** (see Fig. 7-23).

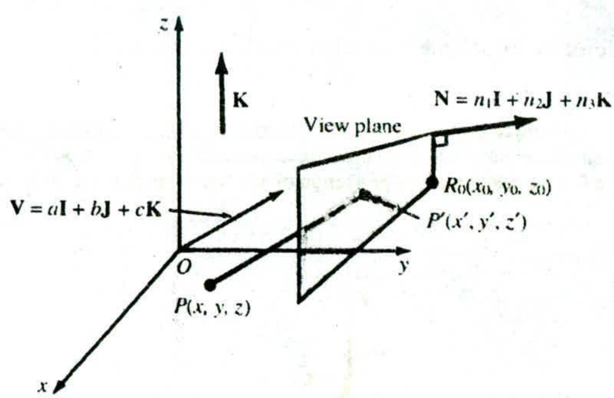


Fig. 7-23

SOLUTION

We reduce the problem to parallel projection onto the xy plane in the direction of the projector $\mathbf{V} = a\mathbf{I} + b\mathbf{J} + c\mathbf{K}$ by means of these steps:

1. Translate the view reference point R_0 of the view plane to the origin using the translation matrix T_{-R_0}.
2. Perform an alignment transformation $A_{\mathbf{N}}$ so that the view normal vector **N** of the view plane points in the direction **K** of the normal to the xy plane. The direction of projection vector **V** is transformed to a new vector $\mathbf{V}' = A_{\mathbf{N}}\mathbf{V}$.
3. Project onto the xy plane using $Par_{\mathbf{V}'}$.

4. Perform the inverse of steps 2 and 1. So finally $Par_{\mathbf{V},\mathbf{N},R_0} = T_{-R_0}^{-1} \cdot A_{\mathbf{N}}^{-1} \cdot Par_{\mathbf{V}'} \cdot A_{\mathbf{N}} \cdot T_{-R_0}$. From what we learned in Chap. 6, we know that

$$T_{-R_0} = \begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and further from Chap. 6, Prob. 6.2, where $\lambda = \sqrt{n_2^2 + n_3^2}$ and $\lambda \neq 0$, that

$$A_{\mathbf{N}} = \begin{pmatrix} \frac{\lambda}{|\mathbf{N}|} & \frac{-n_1 n_2}{\lambda|\mathbf{N}|} & \frac{-n_1 n_3}{\lambda|\mathbf{N}|} & 0 \\ 0 & \frac{n_3}{\lambda} & \frac{-n_2}{\lambda} & 0 \\ \frac{n_1}{|\mathbf{N}|} & \frac{n_2}{|\mathbf{N}|} & \frac{n_3}{|\mathbf{N}|} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Then, after multiplying, we find

$$Par_{\mathbf{V},\mathbf{N},R_0} = \begin{pmatrix} d_1 - an_1 & -an_2 & -an_3 & ad_0 \\ -bn_1 & d_1 - bn_2 & -bn_3 & bd_0 \\ -cn_1 & -cn_2 & d_1 - cn_3 & cd_0 \\ 0 & 0 & 0 & d_1 \end{pmatrix}$$

Here $d_0 = n_1 x_0 + n_2 y_0 + n_3 z_0$ and $d_1 = n_1 a + n_2 b + n_3 c$. An alternative and much easier method to derive this matrix is by finding the intersection of the projector through P with the equation of the view plane (see Prob. A2.14).

7.12 Find the general form of an oblique projection onto the xy plane.

SOLUTION

Refer to Fig. 7-24. Oblique projections (to the xy plane) can be specified by a number f and an angle θ. The number f prescribes the ratio that any line L perpendicular to the xy plane will be foreshortened after projection. The angle θ is the angle that the projection of any line perpendicular to the xy plane makes with the (positive) x axis.

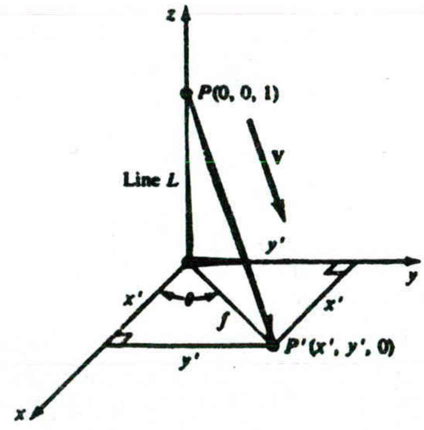


Fig. 7-24

To determine the projection transformation, we need to find the direction vector **V**. From Fig. 7-24, with line L of length 1, we see that the vector $\overline{P'P}$ has the same direction as **V**. We choose **V** to be this vector:

$$\mathbf{V} = \overline{P'P} = x'\mathbf{I} + y'\mathbf{J} - \mathbf{K} \qquad (= a\mathbf{I} + b\mathbf{J} + c\mathbf{K})$$

From Fig. 7-24 we find $a = x' = f\cos\theta$, $b = y' = f\sin\theta$, and $c = -1$.

From Prob. 7.10, the required transformation is

$$Par_{\mathbf{V}} = \begin{pmatrix} 1 & 0 & f\cos\theta & 0 \\ 0 & 1 & f\sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

7.13 Find the transformation for (*a*) cavalier with $\theta = 45°$ and (*b*) cabinet projections with $\theta = 30°$. (*c*) Draw the projection of the unit cube for each transformation.

SOLUTION

(*a*) A cavalier projection is an oblique projection where there is no foreshortening of lines perpendicular to the *xy* plane. From Prob. 7.12 we then see that $f = 1$. With $\theta = 45°$, we have

$$Par_{\mathbf{V}_1} = \begin{pmatrix} 1 & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 1 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(*b*) A cabinet projection is an oblique projection with $f = \frac{1}{2}$. With $\theta = 30°$, we have

$$Par_{\mathbf{V}_2} = \begin{pmatrix} 1 & 0 & \frac{\sqrt{3}}{4} & 0 \\ 0 & 1 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

To construct the projections, we represent the vertices of the unit cube by a matrix whose columns are homogeneous coordinates of the vertices (see Prob. 7.1):

$$V = (ABCDEFGH) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

(*c*) To draw the cavalier projection, we find the image coordinates by applying the transformation matrix $Par_{\mathbf{V}_1}$ to the coordinate matrix **V**:

$$Par_{\mathbf{V}_1} \cdot V = \begin{pmatrix} 0 & 1 & 1 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 1+\frac{\sqrt{2}}{2} & 1+\frac{\sqrt{2}}{2} \\ 0 & 0 & 1 & 1 & 1+\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 1+\frac{\sqrt{2}}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

The image coordinates are then

$$A' = (0, 0, 0) \qquad E' = \left(\frac{\sqrt{2}}{2}, 1 + \frac{\sqrt{2}}{2}, 0\right)$$
$$B' = (1, 0, 0) \qquad F' = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0\right)$$
$$C' = (1, 1, 0) \qquad G' = \left(1 + \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0\right)$$
$$D' = (0, 1, 0) \qquad H' = \left(1 + \frac{\sqrt{2}}{2}, 1 + \frac{\sqrt{2}}{2}, 0\right)$$

Refer to Fig. 7-25.

To draw the cabinet projection:

$$Par_{\mathbf{V}_2} \cdot \mathbf{V} = \begin{pmatrix} 0 & 1 & 1 & 0 & \frac{\sqrt{3}}{4} & \frac{\sqrt{3}}{4} & 1 + \frac{\sqrt{3}}{4} & 1 + \frac{\sqrt{3}}{4} \\ 0 & 0 & 1 & 1 & 1\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 1\frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

The image coordinates are then (see Fig. 7-26)

$$A' = (0, 0, 0) \qquad E' = \left(\frac{\sqrt{3}}{4}, 1\frac{1}{4}, 0\right)$$
$$B' = (1, 0, 0) \qquad F' = \left(\frac{\sqrt{3}}{4}, \frac{1}{4}, 0\right)$$
$$C' = (1, 1, 0) \qquad G' = \left(1 + \frac{\sqrt{3}}{4}, \frac{1}{4}, 0\right)$$
$$D' = (0, 1, 0) \qquad H'\left(1 + \frac{\sqrt{3}}{4}, 1\frac{1}{4}, 0\right)$$

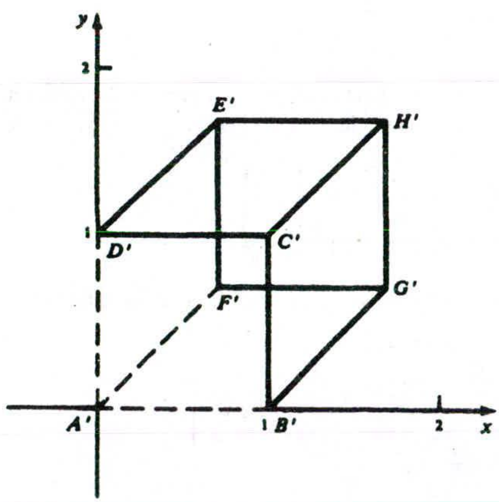


Fig. 7-25

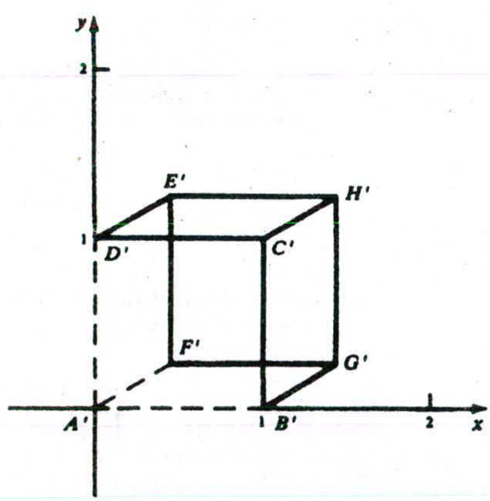


Fig. 7-26

7.14 Construct an isometric projection onto the *xy* plane. Refer to Fig. 7-27.

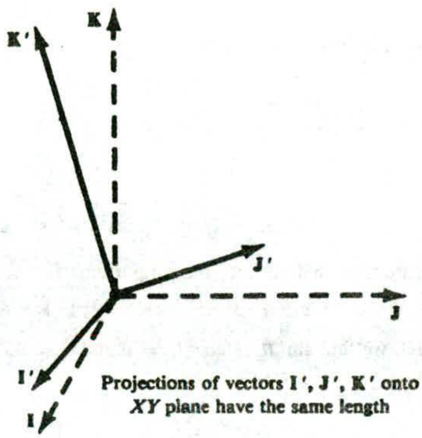


Fig. 7-27

SOLUTION

We shall find a "tilting" of the x, y, z axes that transforms the **IJK** vector tried to a new set **I′J′K′** whose orthographic projections onto the *xy* plane produce vectors of equal lengths.

Denoting the tilting transformation by T and the orthographic projection onto the *xy* plane by $Par_{\mathbf{K}}$, the final projection can be written as $Par = Par_{\mathbf{K}} \cdot T$, where $Par_{\mathbf{K}}$ is as defined in Example 3 and T is as defined in Prob. 6.1 in Chap. 6. Multiplying, we find

$$Par = \begin{pmatrix} \cos\theta_y & \sin\theta_y \sin\theta_x & \sin\theta_y \cos\theta_x & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Now

$$Par \cdot \mathbf{I} = (\cos\theta_y, 0, 0) \qquad Par \cdot \mathbf{J} = (\sin\theta_y \sin\theta_x, \cos\theta_x, 0) \qquad Par \cdot \mathbf{K} = (\sin\theta_y \cos\theta_x, -\sin\theta_x, 0)$$

(the projections of the vectors **I**, **J**, and **K**). To complete the specification of the transformation M, we need to find the angles θ_x and θ_y. To do this, we use the requirement that the images $Par \cdot \mathbf{I}$, $Par \cdot \mathbf{J}$, and $Par \cdot \mathbf{K}$ are to all have equal lengths. Now

$$|Par \cdot \mathbf{I}| = \sqrt{\cos^2\theta_y} \qquad |Par \cdot \mathbf{J}| = \sqrt{\sin^2\theta_y \sin^2\theta_x + \cos^2\theta_x}$$

and

$$|Par \cdot \mathbf{K}| = \sqrt{\sin^2\theta_y \cos^2\theta_x + \sin^2\theta_x}$$

Setting $|Par \cdot \mathbf{J}| = |Par \cdot \mathbf{K}|$ leads to the conclusion that $\sin^2\theta_x - \cos^2\theta_x = 0$ and to a solution $\theta_x = 45°$ (and so $\sin\theta_x = \cos\theta_x = \sqrt{2}/2$). Setting $|Par \cdot \mathbf{I}| = |Par \cdot \mathbf{J}|$ leads to $\cos^2\theta_y = \frac{1}{2}(\sin^2\theta_y + 1)$. Multiplying both sides by 2 and adding $\cos^2\theta_y$ to both sides gives $3\cos^2\theta_y = 2$ and a solution is $\theta_y = 35.26°$ (and so $\sin\theta_y = \sqrt{1/3}$, $\cos\theta_y = \sqrt{2/3}$). Finally

$$Par = \begin{pmatrix} \sqrt{\frac{2}{3}} & \frac{1}{2}\sqrt{\frac{2}{3}} & \frac{1}{2}\sqrt{\frac{2}{3}} & 0 \\ 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

7.15 Construct a dimetric projection onto the *xy* plane.

SOLUTION

Following the procedures in Prob. 7.14, we shall tilt the x, y, z axes and then project on the xy plane. We then have, as before,

$$|Par \cdot \mathbf{I}| = \sqrt{\cos^2 \theta_y} \qquad |Par \cdot \mathbf{J}| = \sqrt{\sin^2 \theta_y \sin^2 \theta_x + \cos^2 \theta_x}$$

and

$$|Par \cdot \mathbf{K}| = \sqrt{\sin^2 \theta_y \cos^2 \theta_x + \sin^2 \theta_x}$$

To define a dimetric projection, we will specify the proportions

$$|Par \cdot \mathbf{I}| : |Par \cdot \mathbf{J}| : |Par \cdot \mathbf{K}| = l : 1 : 1 \qquad (l \neq 1)$$

Setting $|Par \cdot \mathbf{J}| = |Par \cdot \mathbf{K}|$, we find $\sin^2 \theta_x - \cos^2 \theta_x = 0$ and $\theta_x = 45°$, so $\sin \theta_x = \cos \theta_x = \sqrt{2}/2$. Setting $|Par \cdot \mathbf{I}| = l|Par \cdot \mathbf{J}|$ gives

$$\cos^2 \theta_y = \frac{l^2}{2}[\sin^2 \theta_y + 1] \tag{7.3}$$

Multiplying both sides by 2 and adding $l^2 \cos^2 \theta_y$ to both sides gives

$$(2 + l^2)\cos^2 \theta_y = 2l^2$$

So

$$\cos \theta_y = l\sqrt{\frac{2}{2 + l^2}}$$

From equation (*7.3*) we can also find

$$\sin^2 \theta_y = \frac{2 - l^2}{2 + l^2} \qquad \text{and} \qquad \sin \theta_y = \sqrt{\frac{2 - l^2}{2 + l^2}}$$

(Note the restriction $l \leq \sqrt{2}$.) Thus

$$Par = \begin{pmatrix} l\sqrt{\frac{2}{2+l^2}} & \frac{\sqrt{2}}{2}\sqrt{\frac{2-l^2}{2+l^2}} & \frac{\sqrt{2}}{2}\sqrt{\frac{2-l^2}{2+l^2}} & 0 \\ 0 & \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and $0 \leq l \leq \sqrt{2}$.

Note that any other projection ratio, say, $1:1:l$, can be achieved by performing an appropriate rotation before applying *Par*. In this example, a rotaiton of 90° about the y axis aligns the z axis with the x axis so that *Par* can be applied.

Supplementary Problems

7.16 Construct a perspective transformation given three principal vanishing points and the distance D from the center of projection to the projection plane.

7.17 Draw the (*a*) isometric and (*b*) dimetric projections of the unit cube onto the *xy* plane.

7.18 How many view planes (at the origin) produce isometric projections of an object?

CHAPTER 8

Three-Dimensional Viewing and Clipping

An important step in photography is to position and aim the camera at the scene in order to compose a picture. This parallels the specification of 3D viewing parameters in computer graphics that prescribe the projector (the center of projection for perspective projection or the direction of projection for parallel projection) along with the position and orientation of the projection/view plane.

In addition, a *view volume* defines the spatial extent that is visible through a rectangular window in the view plane. The bounding surfaces of this view volume is used to tailor/clip the objects that have been placed in the scene via modeling transformations (Chaps. 4 and 6) prior to viewing. The clipped objects are then projected into the window area, resulting in a specific view of the 3D scene that can be further mapped to the viewport in the NDCS (Chap. 5).

In this chapter we are concerned with the specification of 3D viewing parameters, including a viewing coordinate system for defining the view plane window, and the formation of the corresponding view volume (Sec. 8.1). We also discuss 3D clipping strategies and algorithms (Sec. 8.2). We then summarize the three-dimensional viewing process (Sec. 8.3). Finally, we examine the operational organization of a typical 3D graphics pipeline (Sec. 8.4).

8.1 THREE-DIMENSIONAL VIEWING

Three-dimensional viewing of objects requires the specification of a projection plane (called the *view plane*), a *center of projection* (*viewpoint*) or the direction of projection, and a *view volume* in world coordinates.

Specifying the View Plane

We specify the view plane by prescribing (1) a reference point $R_0(x_0, y_0, z_0)$ in world coordinates and (2) a *unit* normal vector $\mathbf{N} = n_1\mathbf{I} + n_2\mathbf{J} + n_3\mathbf{K}$, $|\mathbf{N}| = 1$, to the view plane (see Fig. 8-1). From this information, we can construct the projections used in presenting the required view with respect to the given viewpoint or direction of projection (Chap. 7).

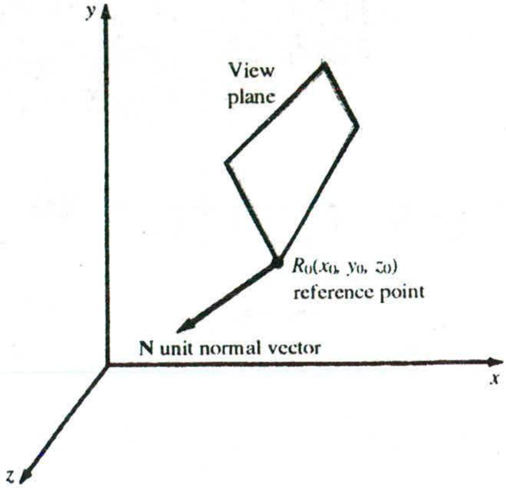


Fig. 8-1

View Plane Coordinates

The *view plane coordinate system* or *viewing coordinate system* can be specified as follows: (1) let the reference point $R_0(x_0, y_0, z_0)$ be the origin of the coordiante system and (2) determine the coordinate axes. To do this, we first choose a reference vector **U** called the *up vector.* A *unit* vector $\mathbf{J_q}$ can then be determined by the projection of the vector **U** onto the view plane. We let the vector $\mathbf{J_q}$ define the direction of the positive q axis for the view plane coordinate system. To calculate $\mathbf{J_q}$, we proceed as follows: with **N** being the view plane unit normal vector, let $\mathbf{U_q} = \mathbf{U} - (\mathbf{N} \cdot \mathbf{U})\mathbf{N}$ (App. 2, Prob. A2.14). Then

$$\mathbf{J_q} = \frac{\mathbf{U_q}}{|\mathbf{U_q}|}$$

is the unit vector that defines the direction of the positive q axis (see Fig. 8-2).

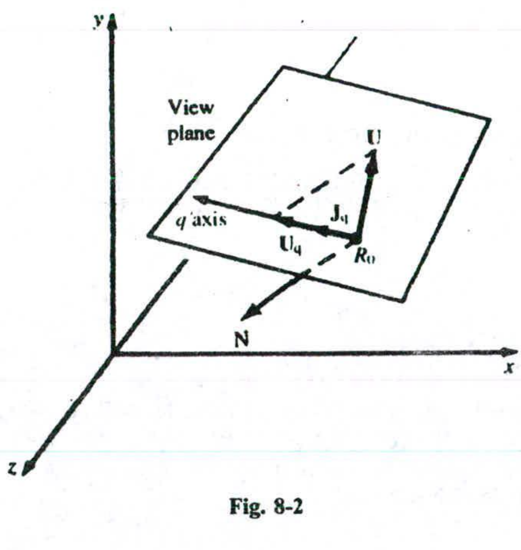


Fig. 8-2

Finally, the direction vector $\mathbf{I_p}$ of the positive p axis is chosen so that it is perpendicular to $\mathbf{J_q}$, and, by convention, so that the triad $\mathbf{I_p}$, $\mathbf{J_q}$, and $\mathbf{N}$ form a *left-handed* coordinate system. That is:

$$\mathbf{I_p} = \frac{\mathbf{N} \times \mathbf{J_q}}{|\mathbf{N} \times \mathbf{J_q}|}$$

This coordinate system is called the *view plane coordinate system* or *viewing coordinate system*. A left-handed system is traditionally chosen so that, if one thinks of the view plane as the face of a display device, then with the p and q coordinate axes superimposed on the display device, the normal vector $\mathbf{N}$ will point away from an observer facing the display. Thus the direction of increasing distance away from the observer is measured along $\mathbf{N}$ [see Fig. 8-3(*a*)].

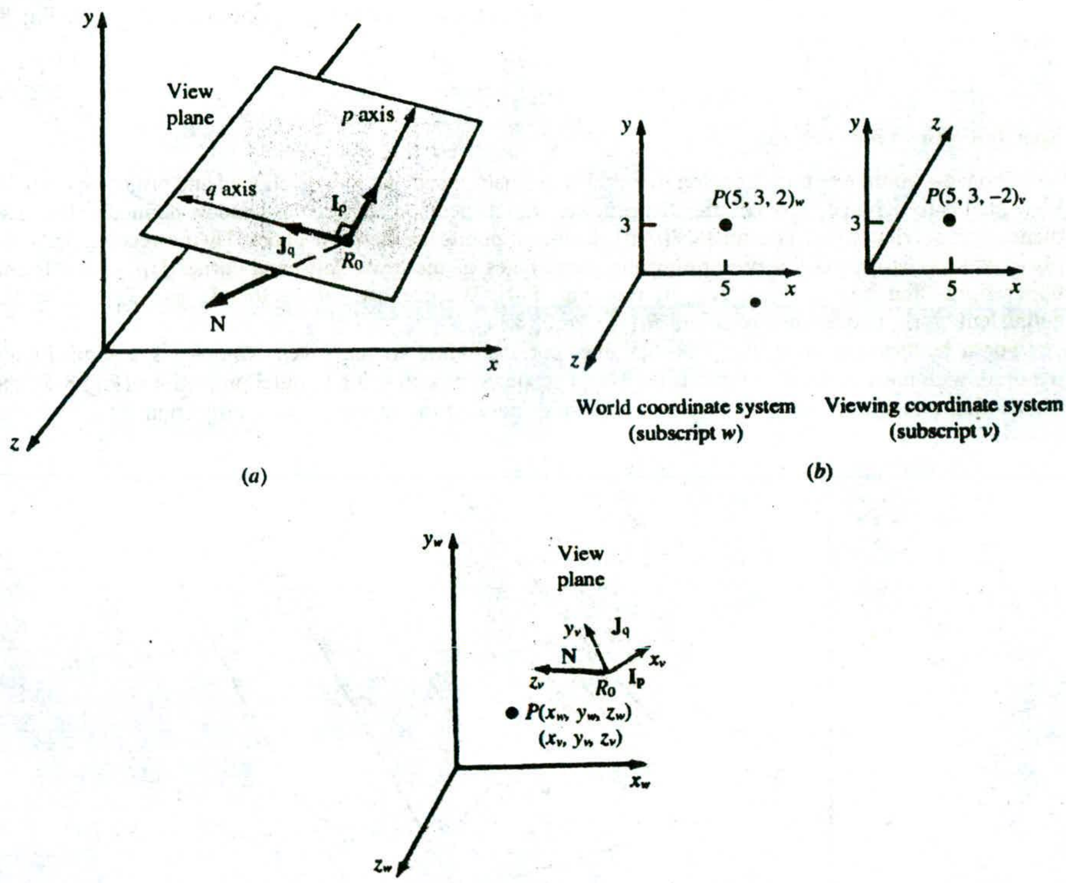


Fig. 8-3

EXAMPLE 1. If the view plane is the xy plane, then $\mathbf{I_p} = \mathbf{I}$, $\mathbf{J_q} = \mathbf{J}$, and the unit normal $\mathbf{N} = -\mathbf{K}$ form a left-handed system. The z coordinate of a point measures the depth or distance of the point from the view plane. The sign indicates whether the point is in front or in back of the view plane with respect to the center or direction of projection. In this example, we change from right-handed world coordinates (x, y, z) to left-handed view plane coordinates

(x', y', z') [see Fig. 8-3(*b*)] be performing the transformation:

$$T_{RL}: \begin{cases} x' = x \\ y' = y \\ z' = -z \end{cases}$$

In matrix form, for homogeneous coordinates:

$$T_{RL} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The general transformation for changing from world coordinates to view plane coordinates [see Fig. 8-3(*c*)] is developed in Prob. 8.3.

Specifying the View Volume

The view volume bounds a region in world coordinate space that will be clipped and projected onto the view plane. To define a view volume that projects onto a specified rectangular window defined in the view plane, we use view plane coordinates $(p, q)_v$ to locate points on the view plane. Then a rectangular view plane window is defined by prescribing the coordinates of the lower left-hand corner $L(p_{\min}, q_{\min})_v$ and upper right-hand corner $R(p_{\max}, q_{\max})_v$ (see Fig. 8-4). We can use the vectors $\mathbf{I_p}$ and $\mathbf{J_q}$ to find the equivalent world coordinates of L and R (see Prob. 8.1).

For a perspective view, the view volume, corresponding to the given window, is a semi-infinite pyramid, with apex at the viewpoint (Fig. 8-5). For views created using parallel projections (Fig. 8-6), the view volume is an infinite parallelepiped with sides parallel to the direction of projection.

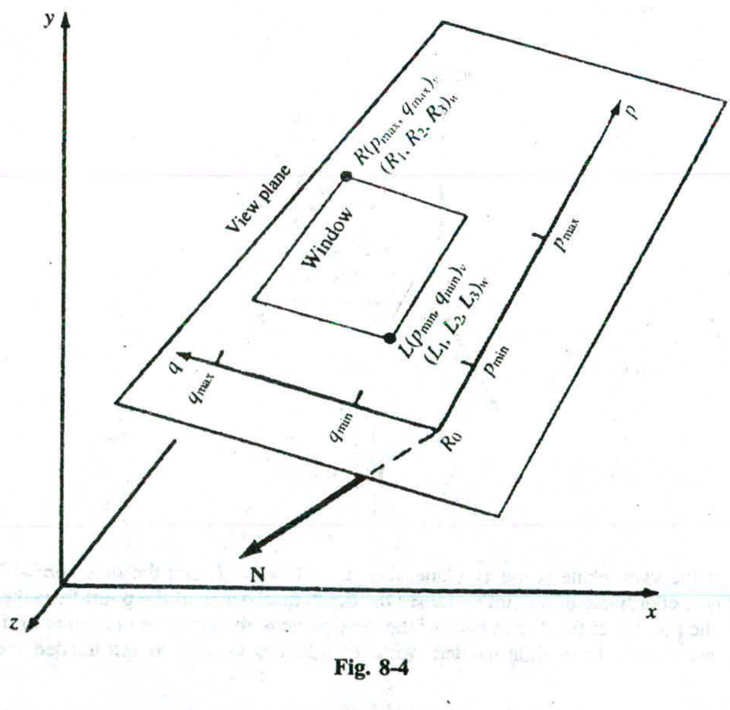


Fig. 8-4

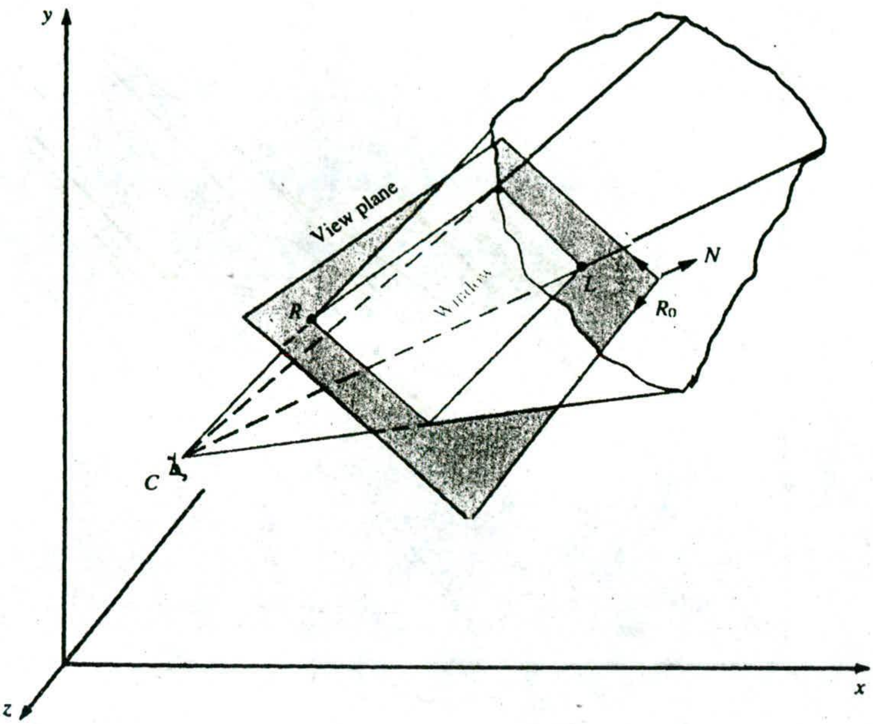


Fig. 8-5

8.2 CLIPPING

Clipping against a Finite View Volume

The view volumes created above are infinite in extent. In practice, we prefer to use a finite volume to limit the number of points to be projected. In addition, for perspective views, very distant objects from the view plane, when projected, appear as indistinguishable spots, while objects very close to the center of projection appear to have disjointed structure. This is another reason for using a finite view volume.

A finite volume is deliminated by using *front* (*near*) and *back* (*far*) *clipping planes* parallel to the view plane. These planes are specified by giving the front distance f and back distance b relative to the view plane reference point R_0 and measured along the normal vector **N**. The signed distance b and f can be positive or negative (Figs. 8-7 and 8-8).

Clipping Strategies

Two differing strategies have been devised to deal with the extraordinary computational effort required for three-dimensional clipping:

1. *Direct clipping*. In this method, as the name suggests, clipping is done directly against the view volume.
2. *Canonical clipping*. In this method, normalizing transformations are applied which transform the original view volume into a so-called canonical view volume. Clipping is then performed against the canonical view volume.

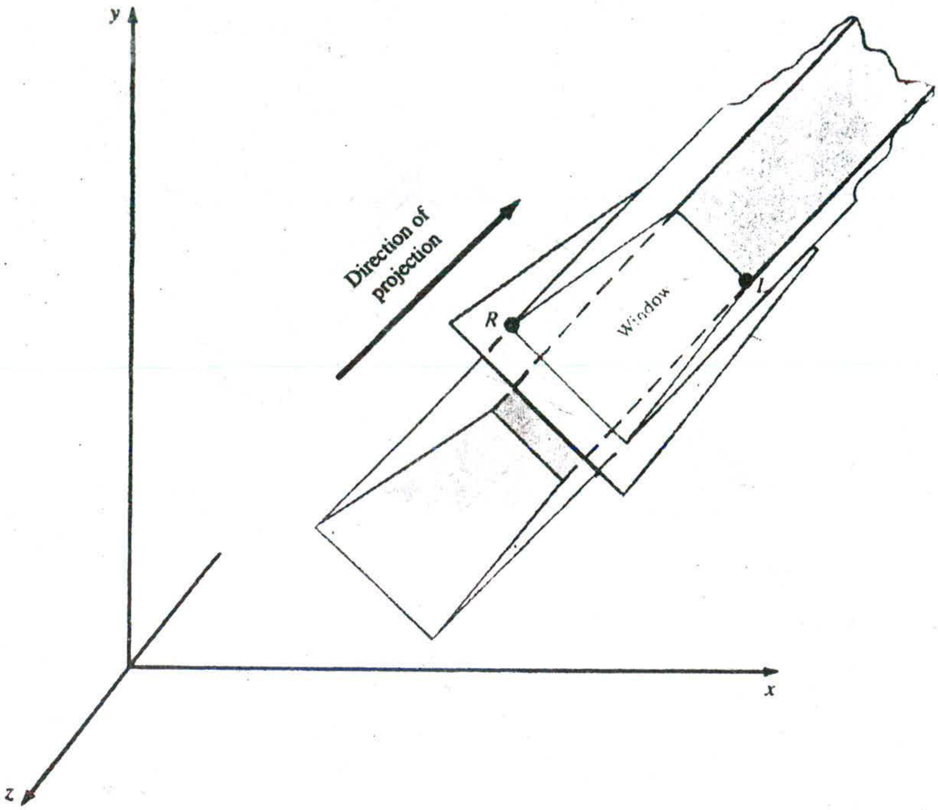


Fig. 8-6

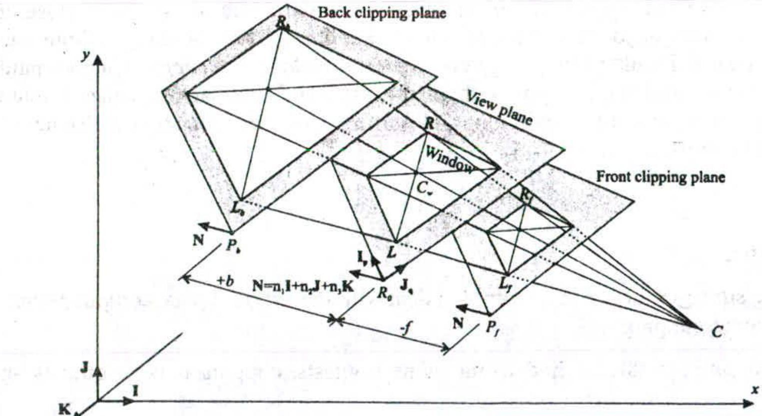


Fig. 8-7 Pespective view volume.

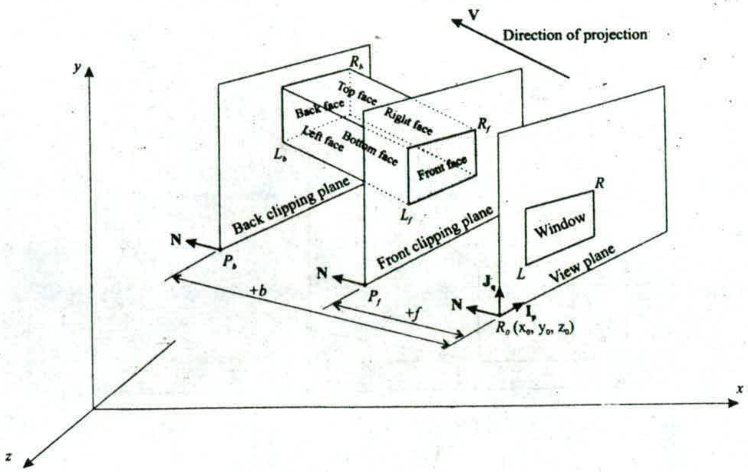


Fig. 8-8 Parallel view volume.

The canonical view volume for parallel projection is the unit cube whose faces are defined by the planes $x = 0, x = 1, y = 0, y = 1, z = 0$, and $z = 1$. The corresponding normalization transformation N_{par} is constructed in Prob. 8.5 (Fig. 8-9).

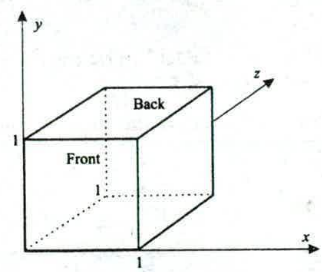


Fig. 8-9

The canonical view volume for perspective projections is the truncated pyramid whose faces are defined by the planes $x = z, x = -z, y = z, y = -z, z = z_f$, and $z = 1$ (where z_f is to be calculated) (Fig. 8-10). The corresponding normalization transformation N_{per} is constructed in Prob. 8.6.

The basis of the canonical clipping strategy is the fact that the computations involved such operations as finding the intersections of a line segment with the planes forming the faces of the canonical view volume are minimal (Prob. 8.9). This is balanced by the overhead involved in transforming points, many of which will be subsequently clipped.

For perspective views, additional clipping may be required to avoid the perspective anomalies produced by projecting objects that are behind the viewpoint (see Chap. 7).

Clipping Algorithms

Three-dimensional clipping algorithms are often direct adaptations of their two-dimensional counterparts (Chap. 5). The modifications necessary arise from the fact that we are now clipping against the six

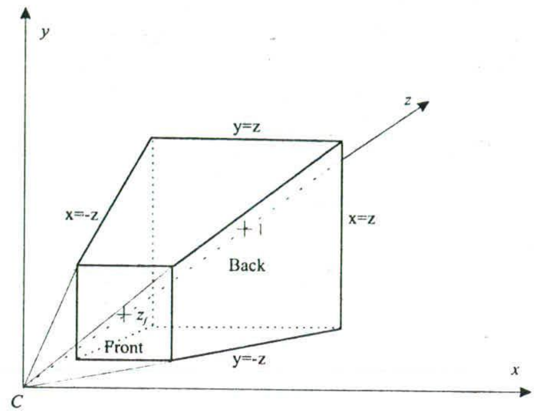


Figure 8-10

faces of the view volume, which are planes, as opposed to the four edges of the two-dimensional window, which are lines.

The technical differences involve:

1. Finding the intersection of a line and a plane (Prob. 8.12).
2. Assigning region codes to the endpoints of line segments for the Cohen–Sutherland algorithm (Prob. 8.8).
3. Deciding when a point is to the right (also said to be *outside*) or to the left (*inside*) of a plane for the Sutherland–Hodgman algorithm (Prob. 8.7).
4. Determining the inequalities for points inside the view volume (Prob. 8.10).

8.3 VIEWING TRANSFORMATION

Normalized Viewing Coordinates

We can view the normalizing transformations N_{par} and N_{per} from Sec. 8.2, under "Clipping Strategies," as geometric transformations. That is, *Obj* is an object defined in the world coordinate system, the transformation

$$Obj' = N_{par} \cdot Obj \qquad \text{or} \qquad Obj' = N_{per} \cdot Obj$$

yields an object *Obj'* defined in the *normalized viewing coordinate system.*

Canonical clipping is now equivalent to clipping in normalized viewing coordinates. That is, the transformed object *Obj'* is clipped against the canonical view volume. In Chap. 10, where hidden-surface algorithms are discussed, it is assumed that the coordinate description of geometric objects refers to normalized viewing coordinates.

Screen Projection Plane

After clipping in viewing coordinates, we project the resulting structure onto the *screen projection plane.* This is the plane that results from applying the transformations N_{par} or N_{per} to the given view plane. In the case N_{par}, from Prob. 8.5, we find that the screen projection plane is the plane $z = -f/(b-f)$ and that the direction of projection is that of the vector **K**. Thus the parallel projection is orthographic (Chap. 7), and, since the plane $z = -f/(b-f)$ is parallel to the xy plane, we can choose this latter plane as the projection plane. So parallel projection *Par* in normalized viewing coordinates reduces to orthographic projection onto the *xy* plane. The projection matrix is (Chap. 7, Sec. 7.3)

$$Par = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In the case of perspective projections, the screen projection plane is the plane $z = c'_z(c'_z + b)$ (Prob. 8.6). The transformed center of projection is the origin. So perspective projection *Per* in normalized viewing coordinates is accomplished by applying the matrix (Chap. 7, Prob. 7.4)

$$Per = \begin{pmatrix} \frac{c'_z}{c'_z + b} & 0 & 0 & 0 \\ 0 & \frac{c'_z}{c'_z + b} & 0 & 0 \\ 0 & 0 & \frac{c'_z}{c'_z + b} & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Constructing a Three-dimensional View

The complete three-dimensional viewing process (without hidden surface removal) is described by the following steps:

1. Transform from world coordinates to normalized viewing coordinates by applying the transformations N_{par} or N_{per}.
2. Clip in normalized viewing coordinates against the canonical clipping volumes.
3. Project onto the screen projection plane using the projections *Par* or *Per*.
4. Apply the appropriate (two-dimensional) viewing transformations (Chap. 5).

In terms of transformations, we can describe the above process in terms of a *viewing transformation* V_T, where

$$V_T = V_2 \cdot Par \cdot CL \cdot N_{par} \quad \text{or} \quad V_T = V_2 \cdot Per \cdot CL \cdot N_{per}$$

Here CL and V_2 refer to the appropriate clipping operations and two-dimensional viewing transformations.

8.4 EXAMPLE: A 3D GRAPHICS PIPELINE

The two-dimensional graphics pipeline introduced in Chap. 5 can non be extended to three dimensions (Fig. 8-11), where modeling transformation first places individually defined objectes into a common scene (i.e. the 3D WCS). Viewing transformation and projection are then carried out according to the viewing parameters set by the application. The result of projection in the view plane window is further mapped to

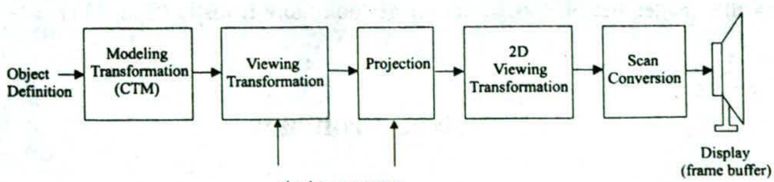


Fig. 8-11 A 3D graphics pipeline

the appropriate workstation viewpoint via 2D viewing transformation and scan-converted to a discrete image in the frame buffer for display.

An application typically specifies the method of projection and the corresponding view volume with system calls such as

$$\text{perspective } (\alpha, a_w, z_f, z_b)$$

where the viewpoint of perspective projection C is assumed to be at the origin of the WCS and the perspective view volume centers on the negative z axis (away from the viewer); α denotes the angle between the top and bottom clipping planes, a_w the aspect ratio of the view plane window, z_f the distance from C to the front clipping plane (which is essentially also the view plane), and z_b the distance from C to the back clipping plane.

On the other hand, orthographic parallel projection can be specified by

$$\text{orthographic } (x_{\min}, x_{\max}, y_{\min}, y_{\max}, z_f, z_b)$$

where the direction of projection is along the negative z axis of the WCS; the first four parameters of the call define the left, right, bottom, and top clipping planes, respectively; and the role of z_f and z_b remains the same as in the perspective case above.

Other calls to the system library often provide additional functionality. For example, the center of perspective projection can be placed anywhere in the WCS by a call that looks like

$$\text{lookat } (x_C, y_C, z_C, x_P, y_P, z_P)$$

where x_c, y_c, z_c are the coordinates of C and x_p, y_p, z_p are the coordinates of a reference point P—the perspective of view volume now centers on the line from C to P. The y axis of the WCS, or more precisely, vector $\mathbf{J}$, serves as the up vector that determines $\mathbf{I}_p$ and $\mathbf{J}_q$. An additional parameter may be included to allow rotation of the viewing coordinate system (with $R_0 = C_w$, the center of the view plane window) about its z axis, i.e. line CP.

Using perspective( ) and lookat( ), we can conveniently produce a sequence of images that animate a "walk-by" or "fly-by" experience by placing P on an object and moving C along the path of the camera from one frame to the next (Fig. 8-12).

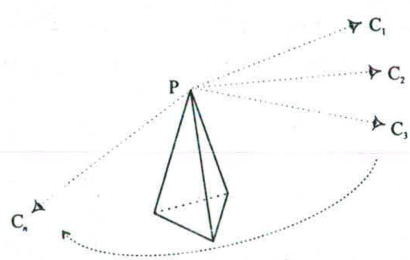


Fig. 8-12

Finally, we want to note a couple of crucial operations of the 3D graphics pipeline that have not yet been discussed. The first is to prevent objects and portions of objects that are hidden from the viewer's eyesight from being included in the projected view (Chap. 10). The second is to assign color attributes to pixels in a way that makes the objects in the image look more realistic (Chap. 11).

Solved Problems

8.1 Let $P(p, q)_v$ be the view plane coordinates of a point on the view plane. Find the world coordinates $P(x, y, z)_w$ of the point.

SOLUTION

Refer to Fig. 8-13. Let R_0 be the view plane reference point. Let **R** be the position vector of R_0 and **W** the position vector of P, both with respect to the world coordinate origin (see Fig. 8-13). Let **V** be the position vector of P with respect to the view plane origin R_0. Now

$$\mathbf{V} = p\mathbf{I_p} + q\mathbf{J_q} \qquad \text{and} \qquad \mathbf{W} = \mathbf{R} + \mathbf{V}$$

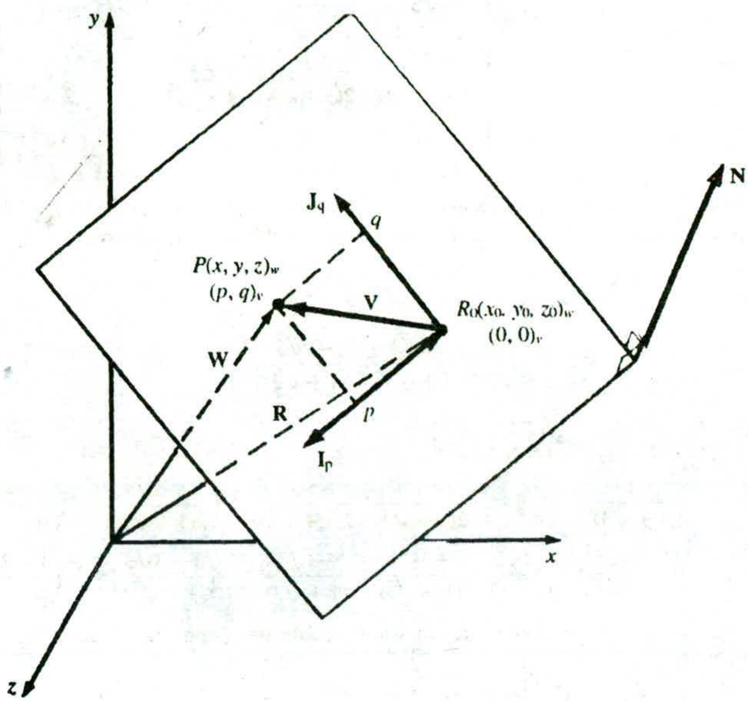


Fig. 8-13

So

$$\mathbf{W} = \mathbf{R} + p\mathbf{I_p} + q\mathbf{J_q}$$

Let the components of $\mathbf{I_p}$ and $\mathbf{J_q}$ be

$$\mathbf{I_p} = a_p\mathbf{I} + b_p\mathbf{J} + c_p\mathbf{K} \qquad \mathbf{J_q} = a_q\mathbf{I} + b_q\mathbf{J} + c_q\mathbf{K}$$

Also

$$\mathbf{R} = x_0\mathbf{I} + y_0\mathbf{J} + z_0\mathbf{K}$$

and so from $\mathbf{W} = \mathbf{R} + p\mathbf{I_p} + q\mathbf{J_q}$ we find

$$\mathbf{W} = (x_0 + pa_p + qa_q)\mathbf{I} + (y_0 + pb_p + qb_q)\mathbf{J} + (z_0 + pc_p + qc_q)\mathbf{K}$$

The required world coordinates of P can be read off from W:

$$P(x_0 + pa_p + qa_q, y_0 + pb_p + qb_q, z_0 + pc_p + qc_q)_w$$

8.2 Find the projection of the unit cube onto the view plane in Prob. 7.9 in Chap. 7. Find the corresponding view plane coordinates of the projected cube.

SOLUTION

Following Prob. 7.9, we must specify several parameters in order to calculate the corresponding perspective projection matrix $Per_{\mathbf{N},R_0,C}$. Choosing $h=\frac{1}{2}$, $t_1=1$, and $t_2=(1-\sqrt{3})/(1+\sqrt{3})$, we obtain

$$Per_{\mathbf{N},R_0,C}=\frac{1+\sqrt{3}}{2}\begin{pmatrix} 0 & \dfrac{2}{1+\sqrt{3}} & 0 & -2 \\ \sqrt{3} & \dfrac{1-\sqrt{3}}{1+\sqrt{3}} & 0 & -(1+\sqrt{3}) \\ \dfrac{\sqrt{3}}{2(1+\sqrt{3})} & \dfrac{1}{2(1+\sqrt{3})} & \dfrac{-2\sqrt{3}}{1+\sqrt{3}} & -\dfrac{1}{2} \\ \dfrac{\sqrt{3}}{1+\sqrt{3}} & \dfrac{1}{1+\sqrt{3}} & 0 & -\left(\dfrac{1+3\sqrt{3}}{1+\sqrt{3}}\right) \end{pmatrix}$$

Applying $Per_{\mathbf{N},R_0,C}$ to the matrix V of homogeneous coordinates of the unit cube, we have $Per_{\mathbf{N},R_0,C}\cdot V=V'$, where V' is the matrix $(A'B'C'D'E'F'G'H')$. After matrix multiplication, we have

$$\mathbf{V}'=\frac{1+\sqrt{3}}{2}\times$$

$$\begin{pmatrix} -2 & -2 & \dfrac{-2\sqrt{3}}{1+\sqrt{3}} & \dfrac{-2\sqrt{3}}{1+\sqrt{3}} & \dfrac{-2\sqrt{3}}{1+\sqrt{3}} & -2 & -2 & \dfrac{-2\sqrt{3}}{1+\sqrt{3}} \\ -(1+\sqrt{3}) & -1 & \dfrac{-2\sqrt{3}}{1+\sqrt{3}} & -3 & -3 & -(1+\sqrt{3}) & -1 & \dfrac{-2\sqrt{3}}{1+\sqrt{3}} \\ -\dfrac{1}{2} & \dfrac{-1}{2(1+\sqrt{3})} & 0 & \dfrac{-\sqrt{3}}{2(1+\sqrt{3})} & \dfrac{-5\sqrt{3}}{2(1+\sqrt{3})} & -\left(\dfrac{1+5\sqrt{3}}{2(1+\sqrt{3})}\right) & -\left(\dfrac{1+4\sqrt{3}}{2(1+\sqrt{3})}\right) & \dfrac{-2\sqrt{3}}{1+\sqrt{3}} \\ -\left(\dfrac{1+3\sqrt{3}}{1+\sqrt{3}}\right) & -\left(\dfrac{1+2\sqrt{3}}{1+\sqrt{3}}\right) & \dfrac{-2\sqrt{3}}{1+\sqrt{3}} & \dfrac{-3\sqrt{3}}{(1+\sqrt{3})} & \dfrac{-3\sqrt{3}}{(1+\sqrt{3})} & -\left(\dfrac{1+3\sqrt{3}}{1+\sqrt{3}}\right) & -\left(\dfrac{1+2\sqrt{3}}{1+\sqrt{3}}\right) & \dfrac{-2\sqrt{3}}{1+\sqrt{3}} \end{pmatrix}$$

Changing from homogeneous coordinates to world coordinates (App. 2), we find the coordinates of the projected cube to be

$$A'\left[2\left(\frac{1+\sqrt{3}}{1+3\sqrt{3}}\right), 2\left(\frac{2+\sqrt{3}}{1+3\sqrt{3}}\right), \frac{1+\sqrt{3}}{2(1+3\sqrt{3})}\right] \qquad E'\left(\frac{2}{3}, \frac{1+\sqrt{3}}{\sqrt{3}}, \frac{5}{6}\right)$$

$$B'\left[2\left(\frac{1+\sqrt{3}}{1+2\sqrt{3}}\right), \frac{1+\sqrt{3}}{1+2\sqrt{3}}, \frac{1}{2(1+2\sqrt{3})}\right] \qquad F'\left[2\left(\frac{1+\sqrt{3}}{1+3\sqrt{3}}\right), \frac{2(2+\sqrt{3})}{1+3\sqrt{3}}, \frac{1+5\sqrt{3}}{2(1+3\sqrt{3})}\right]$$

$$C'(1,1,0) \qquad G'\left[2\left(\frac{1+\sqrt{3}}{1+2\sqrt{3}}\right), \frac{1+\sqrt{3}}{1+2\sqrt{3}}, \frac{1+4\sqrt{3}}{2(1+2\sqrt{3})}\right]$$

$$D'\left(\frac{2}{3}, \frac{1+\sqrt{3}}{\sqrt{3}}, \frac{1}{6}\right) \qquad H'(1,1,1)$$

To change from world coordinates to view plane coordinates, we first choose an up vector. Choosing the vector $\mathbf{K}$, the direction of the positive z axis, as the up vector, we next find the view plane coordinate vectors $\mathbf{I_p}$ and $\mathbf{J_q}$.

With our choices t_1 and t_2, we find that the unit normal vector $\mathbf{N}$ (Prob. 7.9) is

$$\mathbf{N}=\frac{\sqrt{3}}{2}\mathbf{I}+\frac{1}{2}\mathbf{J}$$

Choosing $\mathbf{U}=\mathbf{K}$, and using Prob. A2.14 (App. 2), we find that

$$\mathbf{U_q}=\mathbf{U}-(\mathbf{N}\cdot\mathbf{U})\mathbf{N}=\mathbf{U}\quad(\text{since } \mathbf{N}\cdot\mathbf{U}=0)\quad=\mathbf{K}\qquad\text{and}\qquad\mathbf{J_q}=\frac{\mathbf{U_q}}{|\mathbf{U_q}|}=\mathbf{K}$$

(Note to student using equation (*A2-3*) of Prob. A2.14: we have used the fact that $|\mathbf{N}| = 1$ and replaced $\mathbf{V_p}$ with $\mathbf{U_q}$ and $\mathbf{V}$ and $\mathbf{U}$.)

Now

$$\mathbf{I_p} = \frac{\mathbf{N} \times \mathbf{J_q}}{|\mathbf{N} \times \mathbf{J_q}|}$$

Calculating (App. 2), we obtain

$$\mathbf{N} \times \mathbf{J_q} = \frac{1}{2}\mathbf{I} - \frac{\sqrt{3}}{2}\mathbf{J} \qquad \text{and} \qquad |\mathbf{N} \times \mathbf{J_q}| = 1$$

So

$$\mathbf{I_p} = \frac{1}{2}\mathbf{I} - \frac{\sqrt{3}}{2}\mathbf{J}$$

To convert a point P with world coordinates $(x, y, z)_w$ to view plane coordinates $(p, q)_v$, we use the equations from Prob. 8.1:

$$x = x_0 + pa_p + qa_q \qquad y = y_0 + pb_p + qb_q \qquad z = z_0 + pc_p + qc_q$$

where (x_0, y_0, z_0) are the coordinates of the view plane reference point R_0. Now

$$\mathbf{I_p} = a_p\mathbf{I} + b_p\mathbf{J} + c_p\mathbf{K} = \frac{1}{2}\mathbf{I} - \frac{\sqrt{3}}{2}\mathbf{J} + 0\mathbf{K} \qquad \mathbf{J_q} = a_q\mathbf{I} + b_q\mathbf{J} + c_q\mathbf{K} = 0\mathbf{I} + 0\mathbf{J} + 1\mathbf{K}$$

Choosing $R_0(1, 1, 0)$ as the view plane reference point, we find

$$x = \frac{1}{2}p + 1 \qquad y = \frac{-\sqrt{3}}{2}p + 1 \qquad z = q$$

Solving for p and q, we have

$$p = 2(x - 1) \qquad \text{and} \qquad q = z$$

Using these equations, we convert the transformed coordinates to view plane coordinates:

$$A'\left[2\left(\frac{1-\sqrt{3}}{1+3\sqrt{3}}\right), \frac{1+\sqrt{3}}{2(1+3\sqrt{3})}\right] \qquad E'\left(-\frac{2}{3}, \frac{5}{6}\right)$$

$$B'\left[\frac{2}{1+2\sqrt{3}}, \frac{1}{2(1+2\sqrt{3})}\right] \qquad F'\left[2\left(\frac{1-\sqrt{3}}{1+3\sqrt{3}}\right), \frac{1+5\sqrt{3}}{2(1+3\sqrt{3})}\right]$$

$$C'(0, 0) \qquad G'\left[\frac{2}{1+2\sqrt{3}}, \frac{1+4\sqrt{3}}{2(1+2\sqrt{3})}\right]$$

$$D'\left(-\frac{2}{3}, \frac{1}{6}\right) \qquad H'(0, 1)$$

Refer to Fig. 8-14. Note also that the coordinates of the view point or center of projection C and the vanishing

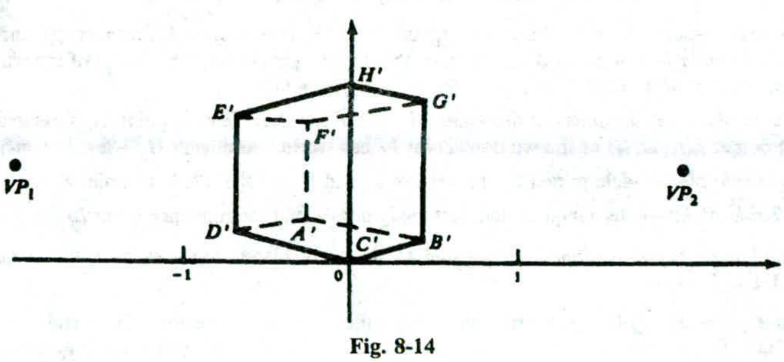


Fig. 8-14

points VP_1 and VP_2 can be found by using the equations from Prob. 7.9:

$$C(a,b,c) = C\left(2, 1+\sqrt{3}, \frac{1}{2}\right) \qquad VP_1\left(0, 1+\sqrt{3}, \frac{1}{2}\right)_w \qquad VP_2\left(2, 1-\sqrt{3}, \frac{1}{2}\right)_w$$

In view plane coordinates:

$$VP_1\left(-2, \frac{1}{2}\right)_v \quad \text{and} \quad VP_2\left(2, \frac{1}{2}\right)_v$$

8.3 Find the transformation T_{wv} that relates world coordinates to view plane coordinates.

SOLUTION

The world coordinate axes are determined by the right-handed triad of unit vectors $[\mathbf{I}, \mathbf{J}, \mathbf{K}]$.

The view plane coordinate axes are determined by the left-handed triad of vectors $[\mathbf{I_p}, \mathbf{J_q}, \mathbf{N}]$ and the view reference point $R_0(x_0, y_0, z_0)$.

Referrring to Fig. 8-3(*a*), we construct the transformation T_{wv} through the concatenation of the matrices determined by the following steps:

1. Translate the view plane reference point $R_0(x_0, y_0, z_0)$ to the world coordinate origin via the translation matrix $T_\mathbf{V}$. Here $\mathbf{V}$ is the vector with components $-x_0\mathbf{I} - y_0\mathbf{J} - z_0\mathbf{K}$.
2. Align the view plane normal $\mathbf{N}$ with the vector $-\mathbf{K}$ (the direction of the negative z axis) using the transformation $A_{\mathbf{N},-\mathbf{K}}$ (Chap. 6, Prob. 6.5). Let $\mathbf{I}'_\mathbf{p}$ be the new position of the vector $\mathbf{I_p}$ after performing the alignment transformation, i.e.

$$\mathbf{I}'_\mathbf{p} = A_{\mathbf{N},-\mathbf{K}} \cdot \mathbf{I_p}$$

3. Rotate $\mathbf{I}'_\mathbf{p}$ about the z axis so that it aligns with $\mathbf{I}$, the direction of the x axis. With θ being the angle between $\mathbf{I}'_\mathbf{p}$ and $\mathbf{I}$, the rotation is $R_{\theta,\mathbf{K}}$ (Chap. 6).
4. Change from the right-handed coordinates to left-handed coordinates by applying the transformation T_{RL} from Example 1. Then $T_{wv} = T_{RL} \cdot R_{\theta,\mathbf{K}} \cdot A_{\mathbf{N},-\mathbf{K}} \cdot T_\mathbf{V}$. If (x_w, y_w, z_w) are the world coordinates of point P, the view plane coordinates (x_v, y_v, z_v) of P can be found by applying the transformation T_{wv}.

8.4 Find the equations of the planes forming the view volume for the general parallel projection.

SOLUTION

The equation of a plane is determined by two vectors that are contained in the plane and a reference point (App. 2, Prob. A2.10). The cross product of the two vectors determines the direction of the normal vector to the plane.

In Fig. 8-8, the sides of the window in the view plane have the directions of the view plane coordinate vectors $\mathbf{I_p}$ and $\mathbf{J_q}$. With $\mathbf{V}$ as the vector determining the direction of projection, we find the following planes:

1. *Top plane*—determined by the vectors $\mathbf{I_p}$ and $\mathbf{V}$ and reference point R_f, measured f units along the unit normal vector $\mathbf{N} = n_1\mathbf{I} + n_2\mathbf{J} + n_3\mathbf{K}$ from the upper right corner $R(r_1, r_2, r_3)$ of the window. Reference point R_f has world coordinates $(r_1 + fn_1, r_2 + fn_2, r_3 + fn_3)$.
2. *Bottom plane*—determined by the vectors $\mathbf{I_p}$ and $\mathbf{V}$ and the reference point L_f, measured from the lower left corner $L(l_1, l_2, l_3)$ of the window. Point L_f has world coordinates $(l_1 + fn_1, l_2 + fn_2, l_3 + fn_3)$.
3. *Right side plane*—determined by the vectors $\mathbf{J_q}$ and $\mathbf{V}$ and the reference point R_f.
4. *Left side plane*—determined by the vectors $\mathbf{J_q}$ and $\mathbf{V}$ and the reference point L_f.

Front and back clipping planes are parallel to the view plane, and thus have the same normal vector $\mathbf{N} = n_1\mathbf{I} + n_2\mathbf{J} + n_3\mathbf{K}$.

5. *Front (near) plane*—determined by the normal vector $\mathbf{N}$ and reference point $P_f(x_0 + fn_1, y_0 + fn_2, z_0 + fn_3)$, measured from the view reference point $R_0(x_0, y_0, z_0)$.

6. *Back (far) plane*—determined by the normal vector **N** and reference point $P_b(x_0 + bn_1, y_0 + bn_2, z_0 + bn_3)$, measured b units from the view plane reference point R_0.

8.5 Find the normalizing transformation that transforms the parallel view volume to the canonical view volume determined by the planes $x = 0$, $x = 1$, $y = 0$, $y = 1$, $z = 0$, and $z = 1$ (the unit cube).

SOLUTION

Referring to Fig. 8-8, we see that the required transformation N_{par} is built by performing the following series of transformations:

1. Translate so that R_0, the view plane reference point, is at the origin. The required transformation is the translation T_{-R_0}.
2. The vectors $\mathbf{I_p}$, $\mathbf{J_q}$, and **N** form the left-handed view plane coordinate system. We next align the view plane normal vector **N** with the vector $-\mathbf{K}$ (the direction of the negative z axis). The alignment transformation $A_{\mathbf{N},-\mathbf{K}}$ was developed in Chap. 6, Prob. 6.5. Let $\mathbf{I}'_\mathbf{p}$ be the new position of the vector $\mathbf{I_p}$; that is, $\mathbf{I}'_\mathbf{p} = A_{\mathbf{N},-\mathbf{K}} \cdot \mathbf{I_p}$.
3. Align the vector $\mathbf{I}'_\mathbf{p}$ with the vector **I** (the direction of the positive x axis) by rotating $\mathbf{I}'_\mathbf{p}$ about the z axis. The required transformation is $R_{\theta,\mathbf{K}}$. Here, θ is the angle between $\mathbf{I}'_\mathbf{p}$ and **I** (Chap. 6). When $R_{\theta,\mathbf{K}}$ aligns $\mathbf{I}'_\mathbf{p}$ with **I**, the vector $\mathbf{J}'_\mathbf{q}$ (where $\mathbf{J}'_\mathbf{q} = A_{\mathbf{N},-\mathbf{K}} \cdot \mathbf{J_q}$) is aligned with the vector **J** (the direction of the positive y axis).
4. We change from the right-handed world coordinate system to a left-handed coordinate system. The required orientation changing transformation is [see Fig. 8-3(*b*)] (see also Example 1)

$$T_{RL} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

5. Let $\mathbf{V}'$ be the new position of the direction of projection vector **V**; that is, $\mathbf{V}' = T_{RL} \cdot R_{\theta,\mathbf{K}} \cdot A_{\mathbf{N},-\mathbf{K}} \cdot \mathbf{V}$. The new position of the transformed view volume is illustrated in Fig. 8-15. Note how the view volume is skewed along the line having the direction of the vector $\mathbf{V}'$. Suppose that the components of $\mathbf{V}'$ are $\mathbf{V}' = v'_x\mathbf{I} + v'_y\mathbf{J} + v'_z\mathbf{K}$. We now perform a shearing transformation that transforms the newly skewed view volume to a rectangular view volume aligned along the z axis. The required shearing transformation is determined by preserving the new view volume base vectors **I** and **J** and shearing $\mathbf{V}'$ to the vector $v'_z\mathbf{K}$ (the **K** component of $\mathbf{V}'$); that is, **I** is transformed to **I**, **J** is transformed to **J**, and $\mathbf{V}'$ is transformed to $v'_z\mathbf{K}$. The required transformation is the matrix

$$Sh = \begin{pmatrix} 1 & 0 & -\dfrac{v'_x}{v'_z} \\ 0 & 1 & -\dfrac{v'_y}{v'_z} \\ 0 & 0 & 1 \end{pmatrix}$$

In order to concatenate the transformation so as to buld N_{par}, we use the 4×4 homogeneous form of Sh

$$\left(\begin{array}{ccc|c} & Sh & & 0 \\ & & & 0 \\ & & & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)$$

6. We now translate the new view volume so that its lower left corner L'_f will be at the origin. To do this, we note that the first four transformations correspond to the view plane coordinate system transformation in Prob. 8.3. So after performing these transformations, we find that the lower left corner of the view plane window $L(p_{\min}, q_{\min})_v$ (view plane coordinates) transforms to a point L' on the xy plane whose coordinates are $(p_{\min}, q_{\min}, 0)$. Similarly, the upper right corner R is transformed to $R'(p_{\max}, q_{\max}, 0)$. After performing the shearing transformation Sh, we see that the view volume is aligned with the z axis and the back and

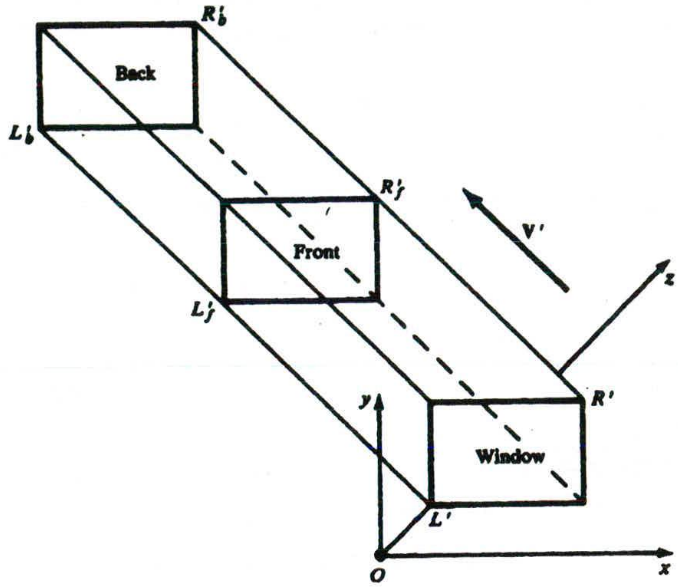


Fig. 8-15

front faces are, respectively, b and f units from the xy plane. Thus the lower left corner of the view volume is at $L_f'(p_{\min}, q_{\min}, f)$ and the bounds of the view volume are $p_{\min} \le x \le p_{\max}$, $q_{\min} \le y \le q_{\max}$, $f \le z \le b$. The required translation is $T_{-L_f'}$.

7. We now scale the rectangular view volume to the unit cube. The base of the present view volume has the dimensions of the base of the original volume, which corresponds to the view plane window; that is

$$w = p_{\max} - p_{\min} \text{ (width)} \qquad h = q_{\max} - q_{\min} \text{ (height)}$$

The depth of the new view volume is the distance from the front clipping plane to the back clipping plane: $d = b - f$. The required scaling is the matrix (in 4×4 homogeneous form)

$$S_{1/w,1/h,1/d} = \begin{pmatrix} \frac{1}{w} & 0 & 0 & 0 \\ 0 & \frac{1}{h} & 0 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The required transformation is then

$$N_{\text{par}} = S_{1/w,1/h,1/d} \cdot T_{-L_f'} \cdot Sh \cdot T_{RL} \cdot R_{\theta,\mathbf{K}} \cdot A_{\mathbf{N},-\mathbf{K}} \cdot T_{-R_0}$$

Note also that after performing the transformation N_{par}, the view plane transforms to the plane $z = -f/(b-f)$, parallel to the xy plane. Also, the direction of projection vector **V** transforms to a vector parallel to the vector **K** having the direction of the z axis.

8.6 **Find the normalizing transformation that transforms the perspective view volume to the canonical view volume determined by the bounding planes $x = z$, $x = -z$, $y = z$, $y = -z$, $z = z_f$, and $z = 1$.**

SOLUTION

Referring to Fig. 8-7, we build the normalizing transformation N_{per} through a series of transformations. As in Prob. 8.5:

1. Translate the center of projection C to the origin using the translation T_{-C}.
2. Align the view plane normal **N** with the vector $-\mathbf{K}$ using $A_{\mathbf{N},-\mathbf{K}}$.
3. Rotate $\mathbf{I}'_{\mathbf{p}}$ to the vector **I** using the rotation $R_{\theta,\mathbf{K}}$. (Recall that $\mathbf{I}'_{\mathbf{p}} = A_{\mathbf{N},-\mathbf{K}} \cdot \mathbf{I}_{\mathbf{p}}$.)
4. We now change from right-handed world coordinates to left-handed coordinates by applying the transformation

$$T_{RL} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

5. The newly transformed view volume is skewed along the centerline joining the origin (the translated center of projection) with the center of the (transformed) view plane window (Fig. 8-16). Let C_w be the coordinates of the center of the original view plane window. Then C_w has view plane coordinates

$$\left(\frac{p_{\min} + p_{\max}}{2}, \frac{q_{\min} + q_{\max}}{2}\right)_v$$

These are changed to world coordinates as in Prob. 8.1. Let $\overline{\mathbf{CC}}_w$ be the vector from the center of projection to the center of the window. Let $(\overline{\mathbf{CC}}_w)'$ be the transformation of the vector $\overline{\mathbf{CC}}_w$; that is, $(\overline{\mathbf{CC}}_w)' = T_{RL} \cdot R_{\theta,\mathbf{K}} \cdot A_{\mathbf{N}-\mathbf{K}} \cdot \overline{\mathbf{CC}}_w$. Then $(\overline{\mathbf{CC}}_w)'$ is the vector that joins the origin to the center of the transformed view plane window (Fig. 8-16). Suppose that $(\overline{\mathbf{CC}}_w)' = c'_x\mathbf{I} + c'_y\mathbf{J} + c'_z\mathbf{K}$. We shear the view volume so that it transforms to a view volume whose center line lies along the z axis. As in Prob. 8.5, the

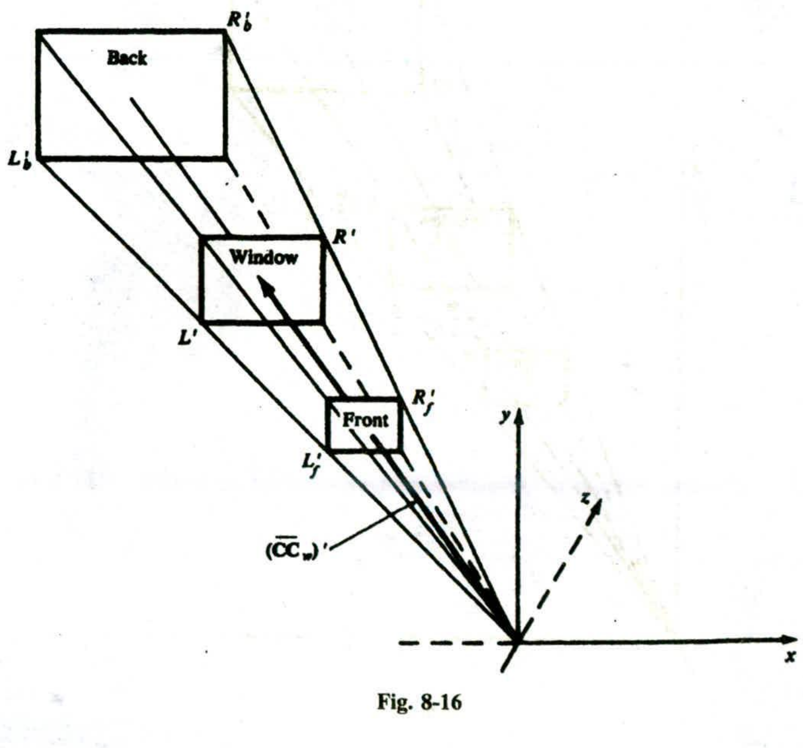


Fig. 8-16

required shearing transformation is

$$Sh = \begin{pmatrix} 1 & 0 & -\frac{c'_x}{c'_z} & 0 \\ 0 & 1 & -\frac{c'_y}{c'_z} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The newly transformed window is, after applying the shearing transformation Sh, located on the z axis at $z_c = c'_z$.

6. Referring to Fig. 8-17, the transformed window is now centered on the z axis. The dimensions of the window are

$$w = p_{\max} - p_{\min} \text{ (width)} \qquad \text{and} \qquad h = q_{\max} - q_{\min} \text{ (height)}$$

The depth of the new view volume is the distance between the front and back clipping planes: $d = b - f$. The transformed window is centered on the z axis at $z_c = c'_z$ and is bounded by

$$-\frac{w}{2} \le x \le \frac{w}{2} \qquad -\frac{h}{2} \le y \le \frac{h}{2}$$

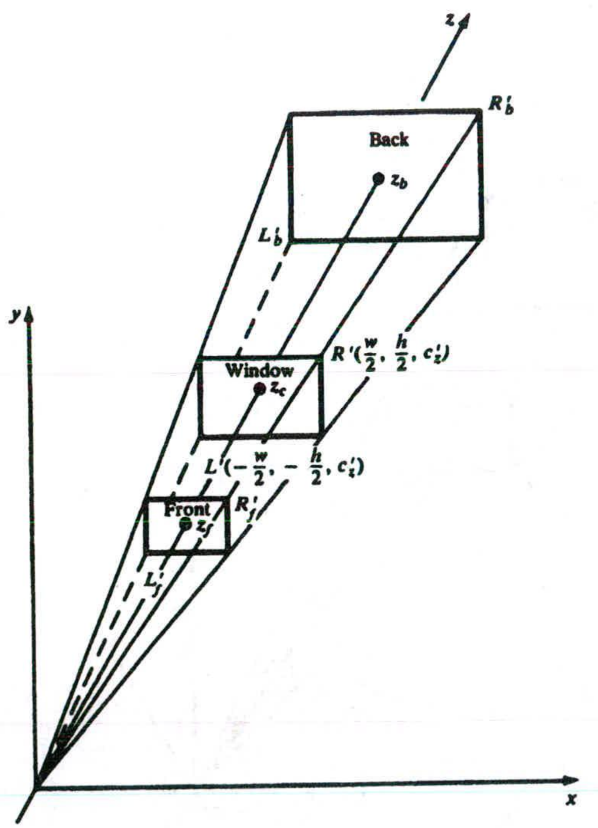


Fig. 8-17

The transformed view plane is located at $z_c = c'_z$. The transformed front clipping plane is located at $z_f = c'_z + f$. The back clipping plane is now located at $z_b = c'_z + b$.

To transform this view volume into the canonical view volume, we first scale in the z direction so that the back-clipping plane is transformed to $z = 1$. The required scale factor is

$$s_z = \frac{1}{c'_z + b}$$

The scaling matrix is

$$S_{1,1,s_z} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{1}{c'_z + b} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

To find the new window boundaries R'' and L'', we apply this scaling transformation to the present window coordinates

$$R'\left(\frac{w}{2}, \frac{h}{2}, c'_z\right) \qquad L'\left(-\frac{w}{2}, -\frac{h}{2}, c'_z\right)$$

Then

$$R'' = \left(\frac{w}{2}, \frac{h}{2}, \frac{c'_z}{c'_z + b}\right) \quad \text{and} \quad L'' = \left(-\frac{w}{2}, -\frac{h}{2}, \frac{c'_z}{c'_2 + b}\right)$$

Next we scale in the x and y directions so that the window boundaries will be

$$R'''\left(\frac{c'_z}{c'_z + b}, \frac{c'_z}{c'_z + b}, \frac{c'_z}{c'_z + b}\right) \quad \text{and} \quad L'''\left(-\frac{c'_z}{c'_z + b}, -\frac{c'_z}{c'_z + b}, \frac{c'_z}{c'_z + b}\right)$$

That is, the window boundaries will lie on the planes $x = z$, $x = -z$, $y = z$,and $y = -z$. The required scale factors are

$$s_x = \frac{2c'_z}{w(c'_z + b)} \quad \text{and} \quad s_y = \frac{2c'_z}{h(c'_z + b)}$$

The corresponding scaling transformation is

$$S_{s_x,s_y,1} = \begin{pmatrix} \dfrac{2c'_z}{w(c'_z + b)} & 0 & 0 & 0 \\ 0 & \dfrac{2c'_z}{h(c'_z + b)} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Multiplication of these scaling transformations into one transformation yields

$$S_{s_x,s_y,s_z} = \begin{pmatrix} \dfrac{2c'_z}{w(c'_z + b)} & 0 & 0 & 0 \\ 0 & \dfrac{2c'_x}{h(c'_z + b)} & 0 & 0 \\ 0 & 0 & \dfrac{1}{c'_z + b} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

To find the location of the front clipping plane, z_f, we apply the transformation S_{s_x,s_y,s_z} to the present location of the center of the front clipping plane, which is $C_f(0, 0, c'_z + f)$. So

$$S_{s_x,s_y,s_z} \cdot C_f = \left(0, 0, \frac{c'_z + f}{c'_z + b}\right)$$

That is

$$z_f = \frac{c'_z + f}{c'_z + b}$$

The complete transformation can be written as

$$N_{\text{per}} = S_{s_x,s_y,s_z} \cdot Sh \cdot T_{RL} \cdot R_{\theta,\mathbf{K}} \cdot A_{\mathbf{N},-\mathbf{K}} \cdot T_{-C}$$

Note that after performing the transformation N_{per}, the view plane is transformed to the plane

$$z = \frac{c'_z}{c'_z + b}$$

parallel to the xy plane. Also, the center of projection C is transformed to the origin.

8.7 How do we determine whether a point P is inside or outside the view volume?

SOLUTION

A plane divides space into the two sides. The general equation of a plane is (App. 2)

$$n_1(x - x_0) + n_2(y - y_0) + n_3(z - z_0) = 0$$

We define a scalar function, $f(P)$, for any point $P(x, y, z)$ by

$$f(P) \equiv f(x, y, z) = n_1(x - x_0) + n_2(y - y_0) + n_3(z - z_0)$$

We say that a point P is on the same side (with respect to the plane) as point Q if $\text{sign}\, f(P) = \text{sign}\, f(Q)$. Referring to Figs. 8-7 or 8-8, let f_T, f_B, f_R, f_L, f_N, and f_F be the functions associated with the top, bottom, right, left, near (front), and far (back) planes, respectively (Probs. 8.4 and 8.10).

Also, L and R are the lower left and upper right corners of the window and P_b and P_f are the reference points of the back and front clipping planes, respectively.

Then a point P is inside the view volume if all the following hold:

P is on the same side as L with respect to f_T
P is on the same side as R with respect to f_B
P is on the same side as L with respect to f_R
P is on the same side as R with respect to f_L
P is on the same side as P_b with respect to f_N
P is on the same side as P_f with respect to f_F

Equivalently

$$\begin{aligned} \text{sign}\, f_T(P) &= \text{sign}\, f_T(L) & \text{sign}\, f_L(P) &= \text{sign}\, f_L(R) \\ \text{sign}\, f_B(P) &= \text{sign}\, f_B(R) & \text{sign}\, f_N(P) &= \text{sign}\, f_N(P_b) \\ \text{sign}\, f_R(P) &= \text{sign}\, f_R(L) & \text{sign}\, f_F(P) &= \text{sign}\, f_F(P_f) \end{aligned}$$

8.8 Show how region codes would be assigned to the endpoints of a line segment for the three-dimensional Cohen–Sutherland clipping algorithm for (*a*) the canonical parallel view volume and (*b*) the canonical perspective view volume.

SOLUTION

The procedure follows the logic of the two-dimensional algorithm in Chap. 5. For three dimensions, the planes describing the view volume divide three-dimensional space into six overlapping exterior regions (i.e., above, below, to right of, to left of, behind, and in front of view volume), plus the interior of the view volume; thus 6-bit codes are used. Let $P(x, y, z)$ be the coordinates of an endpoint.

(*a*) For the canonical parallel view volume, each bit is set to true (1) or false (0) according to the scheme

$$\begin{aligned}
&\text{Bit 1} \equiv \text{endpoint is above view volume} = \text{sign}\,(y-1)\\
&\text{Bit 2} \equiv \text{endpoint is below view volume} = \text{sign}\,(-y)\\
&\text{Bit 3} \equiv \text{endpoint is to the right of view volume} = \text{sign}\,(x-1)\\
&\text{Bit 4} \equiv \text{endpoint is to the left of view volume} = \text{sign}\,(-x)\\
&\text{Bit 5} \equiv \text{endpoint is behind view volume} = \text{sign}\,(z-1)\\
&\text{Bit 6} \equiv \text{endpoint is in front of view volume} = \text{sign}\,(-z)
\end{aligned}$$

Recall that sign $(a) = 1$ if a is positive, 0 otherwise.

(*b*) For the canonical perspective view volume:

$$\begin{aligned}
&\text{Bit 1} \equiv \text{endpoint is above view volume} = \text{sign}\,(y-z)\\
&\text{Bit 2} \equiv \text{endpoint is below view volume} = \text{sign}\,(-z-y)\\
&\text{Bit 3} \equiv \text{endpoint is to the right of view volume} = \text{sign}\,(x-z)\\
&\text{Bit 4} \equiv \text{endpoint is to the left of view volume} = \text{sign}\,(-z-x)\\
&\text{Bit 5} \equiv \text{endpoint is behind view volume} = \text{sign}\,(z-1)\\
&\text{Bit 6} \equiv \text{endpoint is in front of view volume} = \text{sign}\,(z_f-z)
\end{aligned}$$

The category of a line segment (Chap. 5) is (1) visible if both region codes are 000000, (2) not visible if the bitwise logical AND of the region codes is *not* 000000, and (3) clipping candidate if the bitwise logical AND of the region codes is 000000.

8.9 Find the intersecting points of a line segment with the bounding planes of the canonical view volumes for (*a*) parallel and (*b*) perspective projections.

SOLUTION

Let $P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$ be the endpoints of the line segment. The parametric equations of the line segment are

$$x = x_1 + (x_2 - x_1)t \qquad y = y_1 + (y_2 - y_1)t \qquad z = z_1 + (z_2 - z_1)t$$

From Prob. 8.11, the intersection parameter is

$$t_I = \frac{-\mathbf{N} \cdot \overline{\mathbf{R}_0\mathbf{P}_1}}{\mathbf{N} \cdot \overline{\mathbf{P}_1\mathbf{P}_2}}$$

where **N** is the normal vector and R_0 is a reference point on the plane.

(*a*) The bounding planes for the parallel canonical view volume are $x = 0$, $x = 1$, $y = 0$, $y = 1$, $z = 0$, and $z = 1$. For the plane $x = 1$, we have $\mathbf{N} = \mathbf{I}$ and R_0 (1, 0, 0). Then

$$t_I = \frac{-(x_1 - 1)}{x_2 - x_1}$$

If $0 \le t_I \le 1$, the line segment intersects the plane. The point of intersection is then

$$x = x_1 + (x_2 - x_1)\left(-\frac{x_1 - 1}{x_2 - x_1}\right) = 1 \qquad y = y_1 + (y_2 - y_1)\left(-\frac{x_1 - 1}{x_2 - x_1}\right)$$

$$z = z_1 + (z_2 - z_1)\left(-\frac{x_1 - 1}{x_2 - x_1}\right)$$

The intersections with the other planes are found in the same way.

(*b*) The bounding planes for the perspective canonical view volume are $x = z$, $x = -z$, $y = z$, $y = -z$, $z = z_f$, and $z = 1$ (where z_f is calculated as in Prob. 8.6).

To find the intersection with the plane $x = z$, for example, we write the equation of the plane as $x - z = 0$. From this equation, we read off the normal vector as $\mathbf{N} = \mathbf{I} - \mathbf{K}$ (App. 2, Prob. A2.9), and the reference point is $R_0(0, 0, 0)$. Then

$$t_I = -\frac{x_1 - z_1}{(x_2 - x_1) - (z_2 - z_1)}$$

If $0 \le t_I \le 1$, we substitute t_I into the parametric equations of the line segment to calculate the intersection point.

The other intersections are found in the same way.

8.10 Determine the inequalities that are needed to extend the Liang–Barsky line-clipping algorithm to three dimensions for (*a*) the canonical parallel view volume and (*b*) the canonical perspective view volume.

SOLUTION

Let $P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$ be the endpoints of a line. The parametric representation of the line is

$$\begin{cases} x = x_1 + \Delta x \cdot u \\ y = y_1 + \Delta y \cdot u \\ z = z_1 + \Delta z \cdot u \end{cases}$$

where $0 \le u \le 1$, $\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$, and $\Delta z = z_2 - z_1$. The infinite extension of the line corresponds to $u < 0$ and $1 < u$.

(*a*) Points inside the canonical parallel view volume satisfy

$$\begin{aligned} x_{min} &\le x_1 + \Delta x \cdot u \le x_{max} \\ y_{min} &\le y_1 + \Delta y \cdot u \le y_{max} \\ z_{min} &\le z_1 + \Delta z \cdot u \le z_{max} \end{aligned}$$

where $x_{min} = y_{min} = z_{min} = 0$ and $x_{max} = y_{max} = z_{max} = 1$.

Rewrite the six inequalities as

$$p_k \cdot u \le q_k, \qquad k = 1, 2, 3, 4, 5, 6$$

where

$$\begin{array}{lll} p_1 = -\Delta x, & q_1 = x_1 - x_{min} = x_1 & \text{(left)} \\ p_2 = \Delta x, & q_2 = x_{max} - x_1 = 1 - x_1 & \text{(right)} \\ p_3 = -\Delta y, & q_3 = y_1 - y_{min} = y_1 & \text{(bottom)} \\ p_4 = \Delta y, & q_4 = y_{max} - y_1 = 1 - y_1 & \text{(top)} \\ p_5 = -\Delta z, & q_5 = z_1 - z_{min} = z_1 & \text{(front)} \\ p_6 = \Delta z, & q_6 = z_{max} - z_1 = 1 - z_1 & \text{(back)} \end{array}$$

(*b*) Points inside the canonical perspective view volume satisfy (see Fig. 8-10).

$$\begin{aligned} -z &\le x \le z \\ -z &\le y \le z \\ z_f &\le z \le 1 \end{aligned}$$

i.e.

$$\begin{aligned} -z_1 - \Delta z \cdot u &\le x_1 + \Delta x \cdot u \le z_1 + \Delta z \cdot u \\ -z_1 - \Delta z \cdot u &\le y_1 + \Delta y \cdot u \le z_1 + \Delta z \cdot u \\ z_f &\le z_1 + \Delta z \cdot u \le 1 \end{aligned}$$

Rewrite the six inequalities as

$$p_k \cdot u \le q_k, \qquad k = 1, 2, 3, 4, 5, 6$$

where

$$
\begin{array}{lll}
p_1 = -\Delta x - \Delta z & q_1 = x_1 + z_1 & \text{(left)} \\
p_2 = \Delta x - \Delta z, & q_2 = z_1 - x_1 & \text{(right)} \\
p_3 = -\Delta y - \Delta z, & q_3 = y_1 + z_1 & \text{(bottom)} \\
p_4 = \Delta y - \Delta z, & q_4 = z_1 - y_1 & \text{(top)} \\
p_5 = -\Delta z, & q_5 = z_1 - z_f & \text{(front)} \\
p_6 = \Delta z, & q_6 = 1 - z_1 & \text{(back)}
\end{array}
$$

Supplementary Problems

8.11 Find the equations of the planes forming the view volume for the general perspective projection.

8.12 Find the intersection point of a plane and a line segment.