



1.- Dispositivos de despliegue en video

El principal dispositivo de salida en un sistema de gráficas es un monitor de video. La mayor parte de los monitores de video se basa en el diseño estándar de tubo de rayos catódicos (CRT).

Tubos de rayos catódicos repasados: un haz de electrones emitido por un cañón de electrones, pasa a través de sistemas de enfoque y deflexión que dirigen el haz hacia posiciones específicas en la pantalla recubierta con una película de fósforo. Entonces, el fósforo emite una pequeña mancha de luz en cada posición en que hace contacto el haz de electrones. Ya que la luz que el fósforo emite se desvanece con rapidez, se requiere de algún método para mantener la imagen de la pantalla. Una forma de mantener el fósforo resplandeciente es volver a trazar la imagen en repetidas ocasiones al dirigir con rapidez el haz de electrones hacia los mismos puntos. Este tipo de despliegue se conoce como **CRT con repasado**.

La principal diferencia entre los fósforos es su **persistencia**, que se define como el tiempo que la luz emitida desde la pantalla tarda en disminuir una décima parte su intensidad original. Los fósforos con persistencia más baja requieren índices de repasado mayores para mantener una imagen en la pantalla sin parpadear.

La **resolución** es el número de puntos por centímetro que se pueden trazar en sentidos horizontal y vertical.

2.- Despliegues de CRT en color

Podemos distinguir los despliegues de barrido con rastreador y los despliegues de barrido aleatorio:

1. **Despliegues de barrido con rastreador:** el haz de electrones recorre a través de la pantalla, un renglón a la vez, de arriba abajo. Conforme el haz de electrones se mueve a través de cada línea, se activa y desactiva la intensidad del haz de electrones para crear un patrón de manchas iluminadas. La definición de la imagen se almacena en un área de memoria llamada búfer de repasado o búfer de marco o estructura. Esta área de memoria contiene el conjunto de valores de intensidad para todos los puntos de la pantalla. Los valores de intensidad almacenados se recuperan después del búfer de repasado y "pintan" en la pantalla un renglón (línea de barrido). Cada punto de la pantalla se conoce como píxel.
En un sistema simple en blanco y negro, cada punto de la pantalla está encendido o apagado, de manera que sólo se necesita un bit por píxel (**mapa de bits BITMAP**). En sistemas de despliegue de variaciones de color e intensidad se pueden incluir hasta 24 bits por píxel, que pueden requerir varios megabytes de almacenamiento (**mapa de pixels PIXMAP**).
Los índices de repasado se describen en unidades de ciclos por segundo (Hz).
Algunos sistemas de barrido con rastreador usan un procedimiento de repasado entrelazado: cada cuadro se despliega en dos pasos, en una pasada se recorren las líneas de barrido impares y en la siguiente las pares. Permite ver la imagen en la mitad de tiempo cuando la frecuencia es baja y evita el parpadeo si las líneas de barrido adyacentes contienen información de despliegue similar.
2. **Despliegues de barrido aleatorio:** el CRT dirige el haz de electrones sólo a las partes de la pantalla donde se debe crear la imagen, también llamados de despliegue vectorial. La imagen se almacena como un conjunto de comandos para el trazo de líneas en un área de la memoria que los contiene denominada archivo de repaso de despliegue. Están diseñados para aplicaciones de trazo de líneas y no pueden desplegar escenas sombreadas realistas. Pero en cambio tienen una resolución mayor que el de barrido con rastreador.

Diferencia entre los despliegues de barrido con rastreador y los de barrido aleatorio

Los despliegues de barrido aleatorio no pueden desplegar escenas realistas, los de barrido con rastreador si.

Los de barrido aleatorio generalmente tiene una resolución mayor que el de barrido con rastreador.

Los de barrido aleatorio producen trazos más suaves, los de barrido con rastreador produce líneas dentadas.



3.- Monitores CRT de Color

Un monitor CRT despliega imágenes a color utilizando una combinación de fósforos que emiten luz con colores distintos. Las dos técnicas básicas para producir despliegues a color con un CRT son el método de penetración de haz y el método de máscara de sombra:

1. **Método de penetración de haz** (monitores de barrido aleatorio). Se recubren dos capas de fósforo, por lo general, rojo y azul, en el interior de la pantalla del CRT y el color que se despliega depende de cuánto penetra el haz de electrones en las capas de fósforo. Un haz de electrones lentos sólo excita la capa roja exterior. Un haz de electrones muy rápidos penetra a través de la capa roja y excita la capa verde interior. A velocidad de haz intermedia, se emiten combinaciones de luz roja y azul para mostrar dos colores adicionales, anaranjado y amarillo. La penetración de haz es una forma económica de producir monitores de rastreo aleatorio de color, pero sólo es posible generar cuatro colores y la calidad de la imagen no es tan buena como con otros métodos.
2. **Método de máscara de sombra** (barrido por rastreo), para producir una variedad de colores mucho más amplia. Un CRT de máscara de sombra tiene tres puntos de color de fósforo en cada posición de píxel. Un punto de fósforo emite una luz roja, otro emite una luz verde y el tercero emite una luz azul. Este tipo de CRT tiene tres cañones de electrones, una para cada punto de color, y una rejilla de máscara de sombra justo atrás de la pantalla con recubrimiento de fósforo. Se obtienen variaciones de color al variar los niveles de intensidad de los tres haces de electrones.

Los **monitores RGB** utilizan métodos de máscara de sombra y toman el nivel de intensidad para cada cañón directamente sin ningún procesamiento intermedio. Un sistema RGB de 24 bits por píxel, se conoce como **color real**.

4.- Despliegues de panel plano

Una característica importante de los despliegues de panel plano es que son más delgados que los CRT y que se pueden colgar en una pared o utilizarlos en la muñeca, incluso podemos escribir en algunos. Podemos distinguir dos categorías:

1. **Despliegues emisivos o emisores:** son dispositivos que convierten la energía eléctrica en luz (Panales de plasma y despliegues electro luminiscentes en película delgada). Los **panales de plasma** se construyen mediante el llenado de la zona entre dos placas de cristal con una mezcla de gases de neón. Una serie de cintas de conducción verticales se coloca en un panel de cristal y un conjunto de cintas horizontales se monta en el otro panel de cristal. La desventaja de los paneles de plasma es que eran dispositivos monocromáticos, aunque ahora son capaces de desplegar colores y escala de gris. Los **despliegues electro luminiscentes de película delgada** son similares a un panel de plasma, solo que entre las placas de cristal se llena con fósforo. **Diodo de emisión de luz (LED)** es una matriz de diodos que forma las posiciones de los píxeles en pantalla. Como en el repaso de una línea de barrido de un CRT, la información se lee del búfer de retrasado y se convierte a niveles de voltaje de diodos.
2. **Despliegues no emisivos o no emisores:** utilizan efectos ópticos para convertir luz solar o luz de alguna otra fuente en patrones gráficos (Cristal líquido). Los **despliegues de cristal líquido (LCD)**, producen una imagen al pasar luz polarizada a través de un material de cristal líquido. Se usan en sistemas pequeños (calculadoras, ...), producen la imagen al pasar luz polarizada de su alrededor o de una fuente de luz interna a través de una material de cristal líquido que puede alinearse con cualquier bloque o transmitir la luz.



5.- SISTEMAS DE BARRIDO CON RASTREADOR

Los sistemas gráficos de barrido interactivos utilizan varias unidades de procesamiento. Con el fin de controlar la operación del dispositivo de despliegue, además de la CPU, se emplea un procesador para propósitos especiales llamado **controlador de video** o **procesador de despliegue**.

Controlador de video

Se utilizan en sistemas de barrido con rastreador, para operaciones de retrazado básicas del controlador de video, se usan dos registros (x,y) para almacenar las coordenadas de rastreo de los pixels de la pantalla, después de ejecutar el ciclo a través de todos los pixels a lo largo de la línea de rastreo, el controlador de video vuelve a determinar los registros para la primera posición de píxel en la línea de rastreo superior y el proceso de retrazado empieza de nuevo.

Debido a que el tiempo de repaso es muy lento el controlador de video puede recuperar valores múltiples de búfer de despliegue en cada paso.

Aparte de las operaciones básicas el controlador de video puede realizar varias operaciones más:

- Recuperar intensidades de píxel de distintas áreas de la memoria en diferentes ciclos de repaso.
- Realizar ampliaciones, reducciones o mover áreas de la pantalla de una posición a otra.
- Frecuentemente contiene una tabla de búsqueda con los valores de los pixels

Procesador de despliegue de barrido con rastreador

El propósito del procesador de despliegues es liberar la CPU de los trabajos gráficos. Además de la memoria del sistema, se puede contar con un área de memoria separada del procesador de despliegue.

Una función importante del procesador de despliegue es digitalizar la definición de una imagen que se da en un programa de aplicación en un conjunto de valores de intensidad de píxel para su almacenamiento en el búfer de despliegue (**conversión de rastreo**).

Los procesadores de despliegue por lo general están diseñados para operar en interfaz con dispositivos de entrada interactivos, como un ratón.

Existen distintos métodos para organizar el búfer de despliegue como una lista asociada y codificar la información de intensidad:

- Codificación por longitud del tramo: se almacena cada línea de rastreo como un conjunto de pares enteros, uno indica la intensidad y el otro el número de píxel adyacentes que deben tener esa intensidad.
- Codificación de celdas: tiene desventajas en los cambios de intensidad y en las necesidades de almacenamiento.

6.- De las coordenadas de modelado a las coordenadas de dispositivo en 3D. Explicar todos los pasos

Los paquetes generales de gráficos están diseñados para utilizarse con especificaciones de **coordenadas cartesianas**.

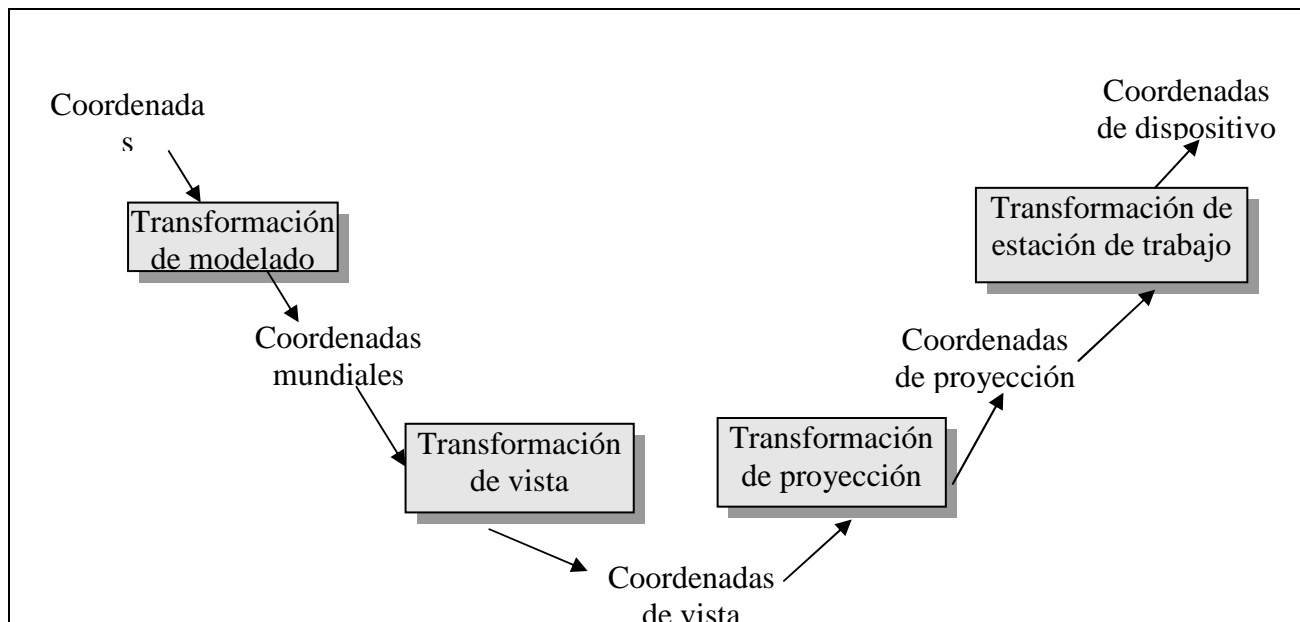
En general, se pueden utilizar varias estructuras cartesianas de referencia distintas para crear y desplegar una escena. Podemos construir la forma de objetos individuales en una escena en estructuras de coordenadas de referencia separadas, que se conocen como **coordenadas de modelado**, **coordenadas locales** o **coordenadas maestras**. Una vez que se especifican las formas de objetos individuales, podemos colocar los objetos en las posiciones adecuadas en la escena al utilizar una estructura de referencia llamada **coordenadas mundiales**. Por último, la descripción de las coordenadas mundiales se transfiere a una o más estructuras de referencia de dispositivos. Estos sistemas de coordenadas de despliegue reciben el nombre de **coordenadas de dispositivo** o **coordenadas de pantalla**.

Por lo general, un sistema de gráficos primero convierte las posiciones de coordenadas mundiales a **coordenadas de dispositivo normalizado**, en el rango de 0 a 1, antes de realizar la conversión final para especificar las coordenadas de dispositivo.

Un posición inicial de coordenadas de modelado (x_{mc}, y_{mc}) se transfiere a una posición de coordenadas de dispositivo (x_{dc}, y_{dc}) con la secuencia:

$(x_{mc}, y_{mc}) \rightarrow (x_{wc}, y_{wc}) \rightarrow (x_{nc}, y_{nc}) \rightarrow (x_{dc}, y_{dc})$
modelado \rightarrow mundiales \rightarrow normalizadas \rightarrow dispositivo

La secuencia de transformación de coordenadas de modelado a coordenadas de dispositivo en 3D, es la siguiente: primero las formas de los objetos se definen en sistemas de coordenadas de modelado local, luego se colocan en la escena general de coordenadas mundiales. Entonces, las especificaciones de las coordenadas mundiales se transforman a coordenadas normalizadas. En el paso final, manejadores de dispositivo individuales transfieren la representación de coordenadas normalizadas de la escena a los dispositivos de salidas para desplegarla.



7.- Algoritmo DDA de generación de rectas (Analizador Diferencial Digital)

Es un algoritmo de línea de conversión de rastreo que se basa en el cálculo ya sea de Δy , o de Δx , por medio de las ecuaciones $\Delta y = m\Delta x$ y $\Delta x = \Delta y / m$. Efectuamos un muestreo de la línea en intervalos unitarios en una coordenada y determinamos los valores enteros correspondientes más próximos a la trayectoria de la línea para la otra coordenada.

Considérese primero una línea con pendiente positiva, la pendiente es menor o igual a 1, llevamos a cabo un muestreo de x en intervalos unitarios ($\Delta x = 1$) y calculamos cada valor sucesivo de y .

Para líneas con una pendiente positiva mayor a 1 se realiza un muestreo de y en intervalos unitarios ($\Delta y = 1$) y calculamos cada valor sucesivo de x .

Si la pendiente es negativa, tenemos que si el valor absoluto de la pendiente es menor que 1 y el extremo en que se comienza es el izquierdo, determinamos que $\Delta x = 1$ y calculamos los valores de y . Cuando se empieza en el extremo derecho determinamos que $\Delta x = -1$ y obtenemos los valores de y . De modo similar cuando el valor absoluto de una pendiente negativa es mayor que 1, utilizamos $\Delta y = -1$, o bien usamos $\Delta y = 1$.

El algoritmo DDA es un método para calcular posiciones de píxel, que es más rápido que la aplicación directa, al utilizar la característica de rastreo. Sin embargo, para segmentos de línea largos, la acumulación de errores de redondeo en adiciones sucesivas del incremento del punto flotante puede provocar que las posiciones de píxel calculadas se desvíen de la trayectoria real de la línea.

El algoritmo sería el siguiente:

$m \geq 0$ (pendiente positiva)

$m \leq 1$

de izquierda a derecha:

* muestreo de x ($\Delta x = 1$)

* $y_{k+1} = \text{redondeo}(y_k + m)$ $k=1,2,\dots$

de derecha a izquierda:

* muestreo de x ($\Delta x = -1$)

* $y_{k+1} = \text{redondeo}(y_k - m)$ $k=1,2,\dots$

$m > 1$ (para evitar la aparición de agujeros)

de izquierda a derecha:

* muestreo de y ($\Delta y = 1$)

* $x_{k+1} = \text{redondeo}(x_k + 1/m)$ $k=1,2,\dots$



de derecha a izquierda:

- * muestreo de y ($\Delta y = -1$)
- * $x_{k+1} = \text{redondeo}(x_k - 1/m)$ $k=1,2,\dots$

$m < 0$ (pendiente negativa)

$|m| < 1$

de izquierda a derecha:

- * muestreo de x ($\Delta x = 1$)
- * $y_{k+1} = \text{redondeo}(y_k + m)$ $k=1,2,\dots$

de derecha a izquierda:

- * muestreo de x ($\Delta x = -1$)
- * $y_{k+1} = \text{redondeo}(y_k - m)$ $k=1,2,\dots$

$|m| > 1$

de izquierda a derecha:

- * muestreo de y ($\Delta y = 1$)
- * $x_{k+1} = \text{redondeo}(x_k + 1/m)$ $k=1,2,\dots$

de derecha a izquierda:

- * muestreo de y ($\Delta y = -1$)
- * $x_{k+1} = \text{redondeo}(x_k - 1/m)$ $k=1,2,\dots$

8.- Algoritmo de Bresenham

Es un algoritmo preciso y efectivo para la generación de líneas de rastreo que convierte mediante rastreo las líneas al utilizar sólo cálculos incrementales con enteros que se pueden adaptar para desplegar circunferencias y curvas. Los ejes verticales muestran las posiciones de línea de rastreo y los ejes horizontales identifican columnas de píxel.

Consideremos el proceso de conversión de rastreo para líneas con pendiente positiva menor que 1. Las posiciones de píxel a lo largo de la trayectoria de una línea se determinan entonces al efectuar un muestreo de x en intervalos unitarios.

Para una línea con una pendiente positiva mayor que 1, intercambiamos las funciones de las direcciones de x y y . Es decir, pasamos a lo largo de la dirección de y en pasos unitarios y calculamos los valores sucesivos de x que se aproximen más a la trayectoria de la línea.

En el caso de pendientes negativas, los procedimientos son similares excepto que ahora, una coordenada decrece conforme la otra aumenta.

El algoritmo sería el siguiente:

$0 < |m| < 1$

1. Se capturan los dos extremos de la línea y se almacena el extremo izquierdo en (x_0, y_0)
2. Se carga (x_0, y_0) en el búfer de estructura (se traza el primer pto)
3. Se calculan las constantes Δx , Δy , $2\Delta y$ y $2\Delta y - 2\Delta x$ y se obtiene el valor inicial para el parámetro de decisión como $p_0 = 2\Delta y - \Delta x$
4. En cada x_k a lo largo de la línea, que inicia en $k=0$, se efectúa la prueba siguiente: si $p_k < 0$, el siguiente punto que se debe trazar es (x_k+1, y_k) y $p_{k+1} = p_k + 2\Delta y$. De otro modo, el siguiente pto a trazar es (x_k+1, y_k+1) y $p_{k+1} = p_k + 2\Delta y - 2\Delta x$
5. Se repite el paso 4 Δx veces.

$|m| > 1$ --> recorreremos la dirección y en pasos unitarios y calculamos los valores sucesivos de x que se aproximen más a la trayectoria de la línea.

9.- Algoritmo de la circunferencia de punto medio

Una circunferencia se define como un conjunto de puntos que se encuentran, en su totalidad, a una distancia determinada r de una posición central.

Es posible reducir el cálculo al considerar la simetría de las circunferencias. La forma de la circunferencia es similar en cada cuadrante y también se puede señalar que es simétrica entre octantes.

Efectuamos un muestreo en intervalos unitarios y determinamos la posición del píxel más cercano a la trayectoria específica de la circunferencia en cada paso.

Podemos tomar pasos unitarios en la dirección positiva de x en este octante y utilizar un parámetro de decisión para determinar cuál de las dos posiciones posibles de y está más próxima a la



trayectoria de la circunferencia en cada paso. Las posiciones de los otros siete octantes se obtienen entonces por simetría. Para aplicar el método del punto medio, definimos una función de circunferencia como:

$$p_k = f_{\text{circunferencia}}(x,y) = x^2 + y^2 - r^2$$

$f_{\text{circunferencia}}(x,y) < 0$, si (x,y) está dentro de la frontera de la circunferencia --> pixel y_k más cercano
 $f_{\text{circunferencia}}(x,y) = 0$, si (x,y) está en la frontera de la circunferencia --> seleccionamos y_{k-1}
 $f_{\text{circunferencia}}(x,y) > 0$, si (x,y) está fuera de la frontera de la circunferencia --> pixel y_{k-1} más cercano

Los parámetros de decisión sucesivos se obtienen al utilizar cálculos incrementales.

El método del punto medio calcula las posiciones de píxel a lo largo de una circunferencia utilizando adiciones y sustracciones de enteros, si se supone que los parámetros de la circunferencia se especifican en coordenadas enteras de pantalla.

El algoritmo sería el siguiente:

1. Se capturan el radio r y el centro de la circunferencia (x_c, y_c) y se obtiene el primer punto de una circunferencia centrada en el origen como $(x_0, y_0) = (0, r)$
2. Se calcula el valor inicial del parámetro de decisión como $p_0 = (5/4) - r$
3. En cada x_k posición, al iniciar en $k=0$, se realiza la prueba siguiente.
 - Si $p_k < 0$, el siguiente punto a lo largo de la circunferencia centrada en $(0,0)$ es (x_{k+1}, y_k) y $p_{k+1} = p_k + 2x_{k+1} + 1$
 - si no, el siguiente punto a lo largo de la circunferencia es (x_{k+1}, y_{k-1}) y $p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$

Donde $2x_{k+1} = 2x_k + 2$ y $2y_{k+1} = 2y_k - 2$
4. Se determinan los puntos de simetría en los otros siete octantes.
5. Se mueve cada posición del píxel calculada (x,y) a la trayectoria circular centrada en (x_c, y_c) y trazamos los valores de las coordenadas: $x = x + x_c$ y $y = y + y_c$
6. Se repiten los pasos 3 a 5 hasta que $x \geq y$

10.- Algoritmo de la elipse de punto medio. Resalte que tiene de interesante este algoritmo frente a los otros algoritmos de rastreo (Falta este último punto)

Una elipse se define como el conjunto de puntos en que la suma de las distancias desde dos posiciones fijas sea la misma para todos los puntos. Una elipse en posición estándar es simétrica entre cuadrantes.

Dados los parámetros r_x, r_y y (x_c, y_c) , determinamos los puntos (x,y) para una elipse en posición estándar centrada en el origen y luego alteramos los puntos, de modo que la elipse esté centrada en (x_c, y_c) .

El método de punto medio para elipse se aplica a lo largo del primer cuadrante en dos partes. Procesamos este cuadrante tomando pasos unitarios en la dirección de x donde la pendiente de la curva tiene una magnitud menor que 1 y tomando pasos unitarios en la dirección de y donde la pendiente tiene una magnitud mayor que 1.

Las dos partes pueden procesarse de varias maneras. Podemos iniciar en la posición $(0, r_y)$ y pasar en el sentido del reloj a lo largo de la trayectoria elíptica en el primer cuadrante, al alternar de pasos unitarios en x a pasos unitarios en y cuando la pendiente adquiere un valor menor que -1. De modo alternativo, podemos iniciar en $(r_x, 0)$ y seleccionar puntos en el sentido contrario a las agujas del reloj, alternando de pasos unitarios en y a pasos unitarios en x cuando la pendiente adquiere un valor mayor que -1.

Definimos la función de una elipse con $(x_c, y_c) = (0,0)$ como:

$$p_k = f_{\text{elipse}}(x,y) = r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2$$

$f_{\text{elipse}}(x,y) < 0$, si (x,y) está dentro de la frontera de la elipse
 $f_{\text{elipse}}(x,y) = 0$, si (x,y) está en la frontera de la elipse
 $f_{\text{elipse}}(x,y) > 0$, si (x,y) está fuera de la frontera de la elipse



Algoritmo de la elipse de punto medio:

1. Se capturan r_x, r_y y el centro de la elipse (x_c, y_c) y se obtiene el primer punto de una elipse centrada en el origen como

$$(x_0, y_0) = (0, r_y)$$

2. Se calcula el valor inicial del parámetro de decisión en la región 1 como

$$p_{10} = r_y^2 - r_x^2 r_y + 1/4 r_x^2$$

3. En cada posición x_k en la región 1, al iniciar en $k=0$, se realiza la prueba siguiente. Si $p_{1k} < 0$, el punto siguiente a lo largo de la elipse centrada en $(0,0)$ es (x_{k+1}, y_k) y

$$p_{1k+1} = p_{1k} + 2r_y^2 x_{k+1} + r_y^2$$

4. Se calcula el valor inicial del parámetro de decisión en la región 2 utilizando el último (x_0, y_0) calculado en la región 1 como

$$p_{20} = r_y^2 (x_0 + 1/2)^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$$

5. En cada y_k posición en la región 2, al iniciar en $k=0$, se realiza la prueba siguiente. Si $p_{2k} > 0$, el punto siguiente a lo largo de la elipse centrada en $(0,0)$ es (x_k, y_{k-1}) y

$$p_{2k+1} = p_{2k} - 2r_x^2 y_{k+1} + r_x^2$$

6. Se determinan puntos de simetría en los otros tres cuadrantes.
7. Se mueve cada posición de píxel calculada (x, y) a la trayectoria elíptica centrada en (x_c, y_c) y se trazan los valores de las coordenadas:

$$X = x + x_c \quad y = y + y_c$$

8. Se repiten los pasos para la región 1 hasta que $2r_y^2 x \geq 2r_x^2 y$.

11.- Diferencias fundamentales entre el algoritmo de Bresenhan y el algoritmo DDA para el trazado de rectas

El algoritmo DDA utiliza aritmética en coma flotante y para segmentos de línea largos, la acumulación de errores de redondeo en adiciones sucesivas del incremento del punto flotante puede provocar que las posiciones de píxel calculadas se desvíen de la trayectoria real de la línea, y el algoritmo de Bresenhan aritmética entera y evita el uso del redondeo.

12.- Antialias

Los primitivos desplegados por los algoritmos de rastreo tienen una apariencia de peldaños o escalonada debido al proceso de digitalización de coordenadas a posiciones discretas de píxels enteros. Esta distorsión de la información se llama **creación de alias**. Para mejorar la apariencia de las líneas de rastreo desplegadas al aplicar métodos de creación de **antialias** que compensan el proceso de submuestreo. Para ello necesitamos reducir el intervalo de muestreo o desplegar objetos con una resolución más alta, pero no suponen una respuesta completa al problema.

Métodos de antialias:

1. **Sobremuestreo o postfiltración:** los objetos se muestrean con una resolución más alta y se despliegan con resolución menor. Se incrementa el índice de muestreo al considerar la pantalla como si estuviera cubierta con una cuadrícula más delgada de la real. Se calculan las intensidades en las posiciones de la cuadrícula de subpíxel y entonces se combinan los resultados para obtener las intensidades del píxel.
2. **Máscaras de peso de píxels:** consiste en dar más peso a los subpíxels cercanos al centro del área de un píxel, puesto que se considera que estos subpíxels son más importantes al determinar la intensidad general de un píxel.



Una matriz de valores que especifican la importancia relativa de los subpixels se conoce como una "máscara" de pesos subpixels.

3. **Muestreo de área o prefiltración:** determinar la intensidad del píxel al calcular el traslape de cada píxel con los objetos que se deben desplegar. Las áreas en que se superponen píxels se obtienen al determinar dónde se intersecan las fronteras del objeto con las fronteras de píxels individuales.
4. **Fases de píxel:** cambiar la posición de despliegue de las áreas de píxels. Se aplica mediante micro colocación del haz de electrones en relación con la geometría del objeto.

13.- Recorte de Líneas. Algoritmos típicos con breve descripción de su funcionamiento

Primero, podemos probar un segmento de línea dado para determinar si cae por completo adentro de la ventana de recorte. Si no es así, intentamos determinar si cae por completo afuera de la ventana. Por último, si no podemos identificar una línea que se localice por completo adentro o afuera, debemos realizar cálculos de intersección con una o más fronteras de recorte.

Recorte de líneas de Cohen-Sutherland

El método acelera el procesamiento de segmentos de línea al realizar pruebas iniciales que reducen el número de intersecciones que se deben calcular. A todos los extremos de línea de una imagen se asigna un código binario de cuatro dígitos (código de región).

1001	1000	1010
0001	0000	0010
0101	0100	0110

Bit 4 arriba – bit 3 abajo – bit2 derecha – bit 1 izquierda

Un valor 1 en cualquier posición del bit indica que el punto está en esa posición.

- Si ambos extremos tienen código 0000 la aceptamos pues está dentro de la ventana.
- Si los extremos tienen un 1 en la misma posición del bit se halla fuera de la ventana y por lo tanto la recortamos.
- Si no está en los casos anteriores hay que realizar pruebas para saber si interseca las fronteras de la ventana:
 - Iniciamos el proceso al comparar un extremo que se halla fuera con una frontera (1º la frontera izquierda, luego la derecha, abajo y arriba) para determinar el porcentaje de la línea que se puede eliminar.
 - Luego verificamos la parte restante de la línea contra las demás fronteras y continuamos hasta que se elimine por completo o se localice la sección que está dentro de la ventana.

Recorte de líneas de Liang-Barsky

Se basan en el análisis de la ecuación paramétrica de un segmento de línea:

$$\begin{aligned}x &= x_1 + u\Delta x \\ y &= y_1 + u\Delta y \quad 0 \leq u \leq 1\end{aligned}$$

donde $\Delta x = x_2 - x_1$ y $\Delta y = y_2 - y_1$

El algoritmo de Liang-Barsky es más eficiente que el algoritmo de Cohen-Sutherland, ya que los cálculos de las intersecciones se reducen.

Expresa las condiciones de recorte de puntos como desigualdades de 2 parámetros (p_k , q_k) y a partir de su relación analiza la posición de la recta y los puntos de intersección con las fronteras de la ventana de recorte.

**Recorte de líneas de Nicholl-Lee-Nicholl**

Evita los recortes múltiples de un segmento de línea individual. Al realizar más pruebas de región antes de calcular las posiciones de las intersecciones, lleva a cabo menos comparaciones y divisiones que los algoritmos anteriores.

El algoritmo NLN sólo se puede aplicar al recorte bidimensional.

Para una línea con extremos p1 y p2, primero determinamos la posición del punto p1. A continuación, determinamos la posición de p2 en relación a p1. Para hacer esto, creamos algunas regiones nuevas en el plano, dependiendo de la ubicación de p1. Las fronteras de las nuevas regiones son segmentos de línea medio infinitos que empiezan en la posición de p1 y pasan a través de las esquinas de la ventana.

14.- Recorte de polígonos. Algoritmos típicos con breve descripción de su funcionamiento

Para el recorte de polígonos, requerimos de un algoritmo que genere una o más áreas cerradas que después se convierten por rastreo en el área de llenado apropiada. La salida de un algoritmo de recorte de polígonos debe ser una secuencia de vértices que define las fronteras de los polígonos recortados.

Recorte de polígonos de Sutherland-Hodgeman

Empezando por el conjunto inicial de vértices del polígono, primero recorta el polígono contra una frontera para producir una nueva secuencia de vértices, con esta nueva secuencia se recorta contra otra frontera y así sucesivamente con las restantes.

Los polígonos cóncavos se pueden desplegar con líneas ajenas cuando el polígono recortado debe tener dos o más secciones separadas. Lo cual requiere medidas adicionales en estos casos como por ejemplo dividir el polígono cóncavo en varios convexos y procesarlos por separado.

Recorte de polígonos de Weiler-Atherton

Se puede aplicar a regiones arbitrarias de polígonos, no tiene problema con los cóncavos.

Consiste en que en lugar de procesar siempre alrededor de las aristas del polígono como se procesan los vértices, en ocasiones deseamos seguir la frontera de la ventana.

Por ejemplo en sentido de las manecillas del reloj, para un par de vértices del polígono del exterior al interior de la ventana, seguimos la frontera del polígono y para un par del interior al exterior seguimos la frontera de la venta en el sentido de las manecillas del reloj.

Recorte de polígonos de Weiler

Calcula el polígono recortado como la intersección del polígono de recorte y el polígono a recortar.

Adaptación métodos paramétricos de recorte de líneas (Liang-Barsky)

Se amplía con un planteamiento general similar al del método de Sutherland-Hodgeman. Las representaciones paramétricas de líneas se utilizan para procesar las aristas de los polígonos en orden alrededor del perímetro del polígono utilizando procedimientos de prueba similares a aquellos que se emplean en el recorte de líneas.

15.- Métodos de entrada interactivos

1. **Modo de solicitud** --> el programa solicita valores de entrada y se suspende el procesamiento hasta que se reciben, un sólo dispositivo a la vez puede dar entrada.
2. **Modo de muestra** --> el programa y los dispositivos operan de manera independiente. Los dispositivos pueden operar mientras el programa procesa otros datos. Los nuevos valores se almacenan reemplazando los valores anteriores, cuando el programa requiere datos realiza un muestreo de los valores actuales de los dispositivos de entrada. Cualquier cantidad de dispositivos de entrada pueden operar al mismo tiempo.
3. **Modo de evento** --> los dispositivos de entrada inician la entrada. El programa y los dispositivos operan de forma concurrente. Los dispositivo presentan los datos en una lista de espera y todos los datos de entrada se guardan, cuando el programa requiere nuevos datos sigue la lista de espera de datos. Varios dispositivos pueden operar al mismo tiempo.

**16.- Entrada de datos gráficos. Clasificación lógica de los dispositivos de entrada**

Los programas de gráficos utilizan varias clases de datos de entrada, que se pueden estructurar de acuerdo con la descripción de datos que cada función deba manejar.

Clasificación lógica de los dispositivos de entrada

- Localizadores: especifica la posición de una coordenada.
 - Mediante el posicionamiento del cursores en la pantalla (ratón, palanca, bola, tableta,..) cuando se encuentra en la posición deseada se activa un botón.
 - Mediante teclado, bien utilizando las teclas de cursores para situar el cursor en la pantalla o introduciendo directamente las coordenadas.
- Indicación: especifica una serie de posiciones de coordenadas, equivale a llamadas múltiples para un dispositivo localizador, así que puede utilizar los dispositivos indicados anteriormente.
- Cadena: especifica cadenas de entrada, principalmente por teclado.
- Valuadores: especifica valores escalares, mediante las teclas numéricas del teclado y barras de control que usados con un dispositivo localizador (ratón) para seleccionar el valor deseado.
- Selección: selecciona opciones de menú, mediante el teclado convencional o de funciones, dispositivo localizador o pantalla táctil y reconocimiento de voz.
- Elección: selecciona componentes de la imagen, mediante dispositivo localizador y presionando botón de selección.

17.- Técnicas interactivas de creación de imágenes

Se pueden proporcionar distintas opciones de entrada, de modo que la información de coordenadas que se captura con los dispositivos localizadores e indicadores se puede ajustar e interpretar de acuerdo con la opción seleccionada:

1. **Método básico de posicionamiento** --> seleccionar las coordenadas con un dispositivo indicador, generalmente el cursor. Como ayuda se pueden reflejar en pantalla los valores numéricos.
2. **Restricciones** --> es una norma para alterar los valores de coordenadas de entrada con el fin de producir una orientación o alineación específica (ej: trazar líneas horizontales)
3. **Cuadrículas** --> se despliega una cuadrícula de líneas rectangulares y cualquier posición de coordenadas se redondea a la intersección más cercana.
4. **Campo de gravedad** --> convierte cualquier posición de entrada cerca de una línea en una posición sobre la línea (ayuda a unir líneas por ejemplo). Se logra al crear un área de *campo de gravedad* alrededor de la línea. El tamaño ha de ser suficiente grande para apoyar el posicionamiento, pero lo suficientemente pequeño para reducir las probabilidades de que se traslapen con otras líneas.
5. **Métodos de banda de hule** --> se selecciona una posición y conforme el cursor se mueve se despliega el objeto cambiando progresivamente hasta que seleccionamos otra posición y se fija (muy usado para trazar figuras geométricas)
6. **Arrastre** --> es la técnica de mover objetos de posición al arrastrarlos con el cursor.
7. **Dibujo a mano alzada** --> las curvas se generan al seguir la trayectoria del estilo sobre una tableta o el cursor sobre la pantalla

**18.- Métodos de despliegue tridimensional**

A fin de obtener un despliegue de una escena tridimensional que se modeló en coordenadas mundiales, primero debemos establecer una referencia de coordenadas para la "cámara". Esta referencia de coordenadas define la posición y orientación para el plano de la película de la cámara, que es el plano que deseamos utilizar para desplegar una vista de los objetos de una escena. De esta manera se transfieren las descripciones de los objetos a las coordenadas de referencia de la cámara y se proyectan sobre el plano de despliegue que se selecciona. Entonces ya podemos desplegar los objetos.

Métodos para generar una vista de una escena tridimensional:

Proyección paralela

Un método para generar una vista de un objeto sólido consiste en proyectar puntos en la superficie del objeto a lo largo de líneas paralelas sobre el plano de despliegue.

En una proyección paralela, las líneas paralelas en la escena en coordenadas mundiales se proyectan en las líneas paralelas del plano de despliegue bidimensional.

Se utiliza en ingeniería y arquitectura para representar un objeto con un conjunto de vistas que mantienen las proporciones relativas del objeto. De manera que se puede reconstruir la apariencia del objeto sólido a partir de las vistas principales.

Proyección de perspectiva

Otro método para generar una vista de una escena tridimensional es proyectar punto hacia el plano de despliegue a lo largo de trayectorias convergente. Esto hace que los objetos que están más lejos de la posición de vista se desplieguen más pequeños que aquellos del mismo tamaño que se encuentran más cerca de la posición de vista.

Indicación de la intensidad

Un objeto en armazón que se despliega con la indicación de la intensidad. Las líneas que están más cerca de la posición de vista se despliegan con las intensidades más altas y las líneas que se encuentran más lejos se despliegan con intensidades reducidas. La indicación de intensidad se aplica al seleccionar los valores de intensidad máximos y mínimos y un rango de distancias en que deben variar las distancias.

Una aplicación de la variación de intensidad es emular el efecto de la atmósfera en la intensidad que se percibe de los objetos.

Identificación de superficies y línea visible

El método más sencillo consiste en realzar las líneas visibles o desplegarlas en un color diferente. Otra técnica, que se utiliza es el despliegue de las áreas no visibles como líneas de rayas. Otro planteamiento consiste en eliminar sólo las líneas no visibles.

Presentación de superficie

Se logra un mayor realismo en los despliegues al establecer la intensidad de la superficie de los objetos de acuerdo con las condiciones de iluminación en la escena y según las características de superficie asignadas.

Vistas separadas y de recortado

Se basa en definir los objetos como estructuras jerárquicas, de manera que sea posible almacenar los detalles internos.

Las vistas separadas permiten mostrar la estructura interna y la relación de las partes del objeto.

La técnica de recortado elimina parte de las superficies visibles para mostrar la estructura interna.

Vistas tridimensionales y estereoscópicas

Agregan realismo a la vista.

19.- Representaciones de matriz de coordenadas homogéneas en 2D

Muchas aplicaciones gráficas implican secuencias de transformaciones geométricas (traslación, rotación, escalación). Con las coordenadas homogéneas se pretende un planteamiento más eficiente obteniendo las posiciones de coordenadas finales directamente a partir de las coordenadas iniciales para eliminar el cálculo de valores de coordenadas intermedias.

Para obtener esta transformación geométrica bidimensional en una sola representación de matriz se amplía las representaciones de matriz de 2 por 2 a matrices de 3 por 3.



Expresar posiciones en coordenadas homogéneas nos permite representar todas las ecuaciones de transformación geométrica como multiplicaciones de matriz.

Traslación

Rotación

Escalación

Proyecciones paralelas en 3D

Consiste en proyectar los objetos tridimensionales en el plano de visión bidimensional. Se transforman las posiciones de coordenadas en el plano de visión a lo largo de líneas paralelas.

Se conserva las proporciones relativas de los objetos, pero no ofrece una representación realista del objeto tridimensional.

Podemos especificar una proyección paralela con un **vector de proyección** que define la dirección de las líneas de proyección. Cuando la proyección es perpendicular al plano de visión, tenemos una **proyección paralela ortogonal** (vistas frontal, lateral y superior de un objeto). Al proyectar puntos a lo largo de líneas paralelas que no son perpendiculares al plano de proyección, tenemos una **proyección paralela oblicua**.

Las ecuaciones de transformación para una **proyección paralela ortogonal** son directas.

Las ecuaciones de transformación para la proyección paralela oblicua tenemos que:

$$x_p = x + L \cos \theta$$

$$y_p = y + L \sin \theta$$

La longitud L depende del ángulo θ y la coordenada de z del punto que se va a proyectar: $\tan \theta = z / L$

Por tanto, $L = z / \tan \theta = z \cot \theta$



La matriz de transformación para producir cualquier proyección paralela en el plano de visión es:

Se obtiene una proyección ortogonal cuando $L_1 = 0$ (que ocurre en un ángulo de proyección θ de 90°). Las proyecciones oblicuas se generan con valores no cero para L_1 .

Los dos valores que se emplean de forma más común para θ son aquellos para los cuales $\tan \theta = 1$ y $\tan \theta = 2$.

Para el primer caso, $\theta = 45^\circ$, las vistas que se obtienen se conocen como **proyecciones caballeras**, todas las líneas perpendiculares al plano de proyección se proyectan sin cambiar de longitud.

Para el segundo caso, $\tan \theta = 2$, la vista resultante se llama **proyección de gabinete**, las líneas perpendiculares a la superficie de vista se proyectan a la mitad de longitud.

Las proyecciones de gabinete tienen una apariencia más realista que las proyecciones caballeras como consecuencia de esta reducción de la longitud de las líneas perpendiculares.

Proyecciones en perspectiva en 3D

Consiste en proyectar los objetos tridimensionales en el plano de visión bidimensional. Se transforman las posiciones de los objetos en el plano de visión a lo largo de líneas que convergen en un punto que se denomina **centro de proyección**. La vista que se proyecta de un objeto se determina al calcular la intersección de las líneas de proyección con el plano de visión.

Genera vistas realistas pero no conserva las proporciones relativas, las proyecciones de objetos distantes son más pequeñas que las proyecciones de objetos del mismo tamaño, los cuales se encuentran más cerca del plano de proyección.

Cuando se proyecta un objeto tridimensional en un plano de visión al usar las ecuaciones de transformación de perspectiva, cualquier conjunto de líneas paralelas en el objeto que no sean paralelas al plano se proyectan en líneas convergentes. Las líneas paralelas que son paralelas al plano de visión se proyectarán de líneas paralelas. El punto en que es aparente la convergencia de un conjunto de líneas paralelas que se proyecta se denomina **punto de fuga**. Una escena puede tener cualquier número de puntos de fuga.

El punto de fuga para cualquier conjunto de líneas que son paralelas a uno de los ejes principales de un objeto se conoce como **punto de fuga principal**.

**TEMA 5 – Transformaciones geométricas bidimensionales****TRASLACIÓN**

Traslación --> cambio posición a lo largo de la trayectoria de una línea recta de una dirección de coordenadas. Se trasladan los puntos y se vuelve a trazar el objeto.

Los polígonos se trasladan al sumar el vector de traslación a la posición de coordenadas de cada vértice y se vuelve a generar el polígono utilizando el nuevo conjunto de coordenadas de vértices y las especificaciones de atributos.

Distancias de traslación --> t_x y t_y . Vector de traslación --> (t_x, t_y)

Posición original (x, y) , posición final (x', y') --> $x' = x + t_x$; $y' = y + t_y$

$$P = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad P' = \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix}, \quad T = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \quad P' = P + T$$

En coordenadas homogéneas:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad P' = T(t_x, t_y) \cdot P$$

Se obtiene el inverso de la matriz de traslación $T(t_x, t_y)$ al reemplazar los parámetros de traslación por sus valores negativos $-t_x$ y $-t_y$.

La **composición de traslaciones** es aditiva --> $T(t_{x2}, t_{y2}) \cdot T(t_{x1}, t_{y1}) = T(t_{x2} + t_{x1}, t_{y2} + t_{y1})$

ROTACIÓN

Rotación bidimensional --> cambiar su posición a lo largo de la trayectoria de una circunferencia en el plano xy. Se especifica un ángulo de rotación σ y un punto de rotación o pivote (x_r, y_r) .

Ángulo de rotación • --> distancia angular de P a P'

Valores positivos --> giro en sentido opuesto a las manecillas del reloj.

Valores negativos --> giro en el sentido de las manecillas del reloj.

Punto de rotación o pivote (x_r, y_r) --> punto del plano xy por el que pasa el eje de rotación (perpendicular al plano xy).

Las ecuaciones de transformación para girar un punto en la posición (x, y) a través de un ángulo σ alrededor del origen:

$$x' = x \cos \sigma - y \sin \sigma$$

$$y' = x \sin \sigma + y \cos \sigma$$

Las ecuaciones de transformación para la rotación de un punto con respecto de cualquier posición de rotación específica (x_r, y_r) .

$$x' = x_r + (x - x_r) \cos \sigma - (y - y_r) \sin \sigma$$

$$y' = y_r + (x - x_r) \sin \sigma + (y - y_r) \cos \sigma$$



Matriz de rotación

Rotación en forma homogénea con respecto al origen de coordenadas:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad P' = R(\theta) \cdot P$$

Obtenemos la matriz de **rotación inversa** al sustituir θ por $-\theta$:

$$R_{-\theta} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

La **composición de rotaciones** es aditiva --> $R(\theta_2) \cdot R(\theta_1) = R(\theta_2 + \theta_1)$

$$P' = R(\theta_2 + \theta_1) \cdot P$$

Rotación respecto a un punto **pivote diferente al origen de coordenadas** ((x_r, y_r) es el punto pivote):

$$\underset{3}{T(x_r, y_r)} \cdot \underset{2}{R(\theta)} \cdot \underset{1}{T(-x_r, -y_r)}$$

1. Traslación del pivote al origen.
2. Rotación respecto al origen.
3. Traslación del pivote a su posición original (la inversa del punto 1)

ESCALADO

Escalado --> altera el tamaño de un objeto, se multiplican los valores de coordenadas (x, y) de cada vértice por los factores de escalación s_x y s_y para producir las coordenadas transformadas (x', y')

Matriz de escalación

Factores de escalado --> s_x escala en la dirección x y s_y escala en la dirección y.

Pueden tomar valores positivos cualesquiera:

$s < 1$ --> reducen el tamaño

$s_x = s_y = 1$ --> mantiene el mismo tamaño (escalado uniforme)

$s > 1$ --> amplían el tamaño



$s_x < > s_y \rightarrow$ escalación diferencial

Al escalar se cambia la posición $\rightarrow s < 1$ acerca al origen y $s > 1$ lo aleja.

Para controlar la localización del objeto, se selecciona un **punto fijo** que debe permanecer sin cambios después de la escalación.

Escalado en forma matricial homogénea respecto al origen de coordenadas:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad P' = S(s_x, s_y) \cdot P$$

La matriz de **escalación inversa** se obtiene al sustituir sus inversos multiplicativos ($1/s_x$ y $1/s_y$).

La **composición de escalaciones** es multiplicativa $\rightarrow S(s_{x2}, s_{y2}) \cdot S(s_{x1}, s_{y1}) = S(s_{x2} \cdot s_{x1}, s_{y2} \cdot s_{y1})$

Escalación respecto a un **punto diferente al origen de coordenadas:**

$$T(x_f, y_f) \cdot S(s_x, s_y) \cdot T(-x_f, -y_f)$$

3 2 1

1. Traslación de modo que el pto fijo esté en el origen de coordenadas.
2. Escalación respecto al origen
3. Traslación inversa al paso 1, para regresar a la posición original.

Escalación en **dirección arbitraria**: \rightarrow rotar o girar el objeto para alinear las direcciones de escalación con los ejes antes de aplicar la escalación y después aplicar la rotación opuesta: $R(-\theta) \cdot S(s_x, s_y) \cdot R(\theta)$. Para mantener un punto fijo combinaríamos con traslaciones adecuadas:

$$T(x_f, y_f) \cdot R(-\theta) \cdot S(s_x, s_y) \cdot R(\theta) \cdot T(-x_f, -y_f)$$

TRANSFORMACIONES COMPUESTAS

La matriz de transformación compuesta se obtiene al calcular el **producto de las matrices** de transformación individuales.

Se lleva a cabo **de derecha a izquierda**, donde la matriz del extremo derecho es la primera transformación que se aplica al objeto y la matriz del extremo izquierdo es la última transformación.

La **composición de traslaciones** es aditiva $\rightarrow T(t_{x2}, t_{y2}) \cdot T(t_{x1}, t_{y1}) = T(t_{x2} + t_{x1}, t_{y2} + t_{y1})$

La **composición de rotaciones** es aditiva $\rightarrow R(\theta_2) \cdot R(\theta_1) = R(\theta_2 + \theta_1)$

La **composición de escalaciones** es multiplicativa $\rightarrow S(s_{x2}, s_{y2}) \cdot S(s_{x1}, s_{y1}) = S(s_{x2} \cdot s_{x1}, s_{y2} \cdot s_{y1})$

La multiplicación de matrices es asociativa pero no es conmutativa en general, casos especiales se presentan cuando todas las transformaciones son de la misma clase y la rotación y escalación uniforme.



Se puede expresar una matriz de transformación de cuerpo rígido general que sólo implica traslaciones y rotaciones de la forma:

donde los cuatro elementos r_{ij} son los términos de rotación multiplicativos y los elementos t_{rx} y t_{ry} son los términos de traslación.

REFLEXIÓN

Produce una imagen especular o de espejo al girar el objeto 180° alrededor del eje de reflexión.

Reflexión alrededor de la línea $y=0$ (eje x): $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Conserva los x y gira los y

Reflexión alrededor de la línea $x=0$ (eje y): $\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Conserva los y y gira los x

Eje perpendicular al plano xy que pasa por O: $\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Es la matriz de rotación $R(180^\circ)$

Para cualquier eje de reflexión perpendicular al plano xy, es una rotación de 180° en el plano xy al utilizar como punto pivote el punto de reflexión (punto del eje que está en xy).

Eje de reflexión diagonal: $y=x$ $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$y=-x$ $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Eje de reflexión cualquier línea $y = mx + b$ en el plano xy :

1. Traslación de modo que el eje pase por el origen.
2. Rotación del eje sobre uno de los ejes de coordenadas.
3. Reflexión alrededor de ese eje de coordenadas.
4. Rotación inversa a la del punto 2.
5. Traslación inversa a la del punto 1.

**REPRESENTACIONES DE MATRIZ Y COORDENADAS HOMOGÉNEAS**

$$P' = M1 \cdot P + M2$$

P y P' son las posiciones de las coordenadas representadas como columnas del vector.

M1 matriz de 2x2 que contiene los factores de multiplicación.

M2 matriz de columnas de dos elementos que contiene términos de traslación.

Para la traslación M1 es la matriz identidad.

Para la rotación o escalación M2 contiene los términos de traslación del punto fijo.

Expresar posiciones de coordenadas homogéneas nos permite representar todas las ecuaciones de transformación geométrica como multiplicaciones de matriz.

NOTAS IMPORTANTES:

1. Traslación según la recta $ay=mx+b$ una cantidad t de unidades. Calcular un vector en la dirección de la recta (a,m) , o sea el vector director, calcular un vector unitario en la misma dirección

$$\left(\frac{a}{\sqrt{a^2+m^2}}, \frac{m}{\sqrt{a^2+m^2}} \right) \text{ y multiplicarlo por } t \text{ para obtener las distancias de traslación } (t_x, t_y) = \left(t \cdot \frac{a}{\sqrt{a^2+m^2}}, t \cdot \frac{m}{\sqrt{a^2+m^2}} \right)$$

2. Para hacer la traslación y rotación necesaria del eje de reflexión cuando es una recta $y=mx+b$:

- Hay que tener en cuenta que el ángulo de inclinación de una recta está relacionado con la pendiente m por $m = \tan(\theta)$. Se multiplican las matrices utilizando las matrices ya conocidas respecto a \sin y \cos de θ y una vez obtenemos la matriz de transformación completa sustituimos sabiendo:

$$\sin(\theta) = \frac{m}{\sqrt{m^2+1}} \text{ y } \cos(\theta) = \frac{1}{\sqrt{m^2+1}}$$

- El punto de cruce de la recta con el eje y es $(0,b)$ por lo que la traslación ha de ser $(0,-b)$
3. Para representar polígonos, utilizamos una matriz donde cada columna es un vértice (el último valor a 1 para que estén en coordenadas homogéneas) y esta matriz la multiplicaremos por la matriz de transformación.

EJERCICIOS:

1. **Transformaciones compuestas. Propiedades de concatenación.** (1) 204
2. **Transformada bidimensional: Reflexión** (1) 210-212
3. **Defínase la matriz de la siguiente composición de transformaciones: Traslación según la recta $y=-5x+3$ una cantidad 20 unidades y rotación alrededor del origen de coordenadas de $2 \cdot t$ grados**

$$y=-5x+3 \rightarrow \text{vector en la dirección } (1,-5); \text{ vector director unitario } \left(\frac{1}{\sqrt{26}}, -\frac{5}{\sqrt{26}} \right)$$



como son 20 unidades: $(\frac{20}{\sqrt{26}}, -\frac{100}{\sqrt{26}})$ la matriz de traslación será

$$\begin{bmatrix} 1 & 0 & \frac{20}{\sqrt{26}} \\ 0 & 1 & -\frac{100}{\sqrt{26}} \\ 0 & 0 & 1 \end{bmatrix}$$

La matriz de rotación será

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R \cdot T = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & \frac{20}{\sqrt{26}} \\ 0 & 1 & -\frac{100}{\sqrt{26}} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & \cos\theta \cdot \frac{20}{\sqrt{26}} + \sin\theta \cdot \frac{100}{\sqrt{26}} \\ \sin\theta & \cos\theta & \sin\theta \cdot \frac{20}{\sqrt{26}} - \cos\theta \cdot \frac{100}{\sqrt{26}} \\ 0 & 0 & 1 \end{bmatrix}$$

4. Defínase la matriz de la siguiente composición de transformaciones: Traslación según la recta $y=-2x+3$ una cantidad t y rotación alrededor del origen de coordenadas de $2 \cdot t$ grados

5. Dados los 5 puntos: $P_1 = (15,10)$; $P_2 = (20,15)$; $P_3 = (25,10)$; $P_4 = (25,0)$ y $P_5 = (15,0)$ y los 6 segmentos: P_1P_2 ; P_2P_3 ; P_3P_1 ; P_3P_4 ; P_4P_5 Y P_5P_1 . Obtener la figura resultante de aplicar la siguiente secuencia de transformaciones:

1. Reflexión respecto a la recta $y = x$
2. Traslación según el vector $T_x = 10$; $T_y = -5$
3. Escalación según $S_x = 2$; $S_y = 0.5$

$$R = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 15 & 20 & 25 & 25 & 15 \\ 10 & 15 & 10 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad P' = (S \cdot T \cdot R) \cdot P$$

6. Defínase la matriz de la siguiente transformación compuesta: rotación, traslación y escalación de una cantidad A tomando como origen el punto $(2,2)$

1. T_1 = Traslación de $(2,2)$ al origen de coordenadas
2. R = Rotación σ°
3. T_2 = Traslación t_x, t_y .
4. S = Escalado con factor de escalación A



5. $T1^{-1}$ = Traslación a (2,2) inversa a pto1

$$T1^{-1} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T2 = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T1 = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Matriz de Transformación} = T1^{-1} \cdot S \cdot T2 \cdot R \cdot T1$$

7. Determinar la forma de la matriz de transformación para una reflexión con respecto de una línea arbitraria con la ecuación $y = mx + b$

1. Traslación de modo que el eje pase por el origen $T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix}$

2. Rotación del eje sobre el eje de coordenadas $R = \begin{bmatrix} \cos(\arctan m) & \sin(\arctan m) & 0 \\ -\sin(\arctan m) & \cos(\arctan m) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

3. Reflexión sobre el eje x $SH = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

4. Rotación inversa al pto 2 $R^{-1} = \begin{bmatrix} \cos(\arctan m) & -\sin(\arctan m) & 0 \\ \sin(\arctan m) & \cos(\arctan m) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

5. Traslación inversa al pto 1 $T^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}$

$$T^{-1} \cdot R^{-1} \cdot SH \cdot R \cdot T$$

8. Defínase la matriz de la siguiente transformación: simetría respecto del eje $y=2*x+3$



9. La transformación definida por la matriz A, consigue una simetría respecto de una recta $y=ax+b$. Defínase la transformación que hace la simetría respecto de la recta perpendicular a ella que pase por el origen

10. Defínase la matriz de la siguiente composición de transformaciones: Translación según la recta $y= 2*x+3$ una cantidad t y rotación alrededor del origen de coordenadas de t grados.

Un vector en la dirección de la recta es $(1,2)$, un vector unitario será ese vector dividido por el módulo, como $|(1,2)| = \sqrt{1+2^2}=\sqrt{5}$ el vector será $(\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}})$.

Las distancias de traslación serán: $t_x = \frac{t}{\sqrt{5}}$ y $t_y = \frac{2t}{\sqrt{5}}$

$$T = \begin{bmatrix} 1 & 0 & \frac{t}{\sqrt{5}} \\ 0 & 1 & \frac{2t}{\sqrt{5}} \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_L = R \cdot T = \begin{bmatrix} \cos t & -\sin t & 0 \\ \sin t & \cos t & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & \frac{t}{\sqrt{5}} \\ 0 & 1 & \frac{2t}{\sqrt{5}} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos t & -\sin t & \frac{t \cdot \cos t}{\sqrt{5}} - \frac{2t \cdot \sin t}{\sqrt{5}} \\ \sin t & \cos t & \frac{t \cdot \sin t}{\sqrt{5}} + \frac{2t \cdot \cos t}{\sqrt{5}} \\ 0 & 0 & 1 \end{bmatrix}$$

**TEMA 11 – Transformaciones geométricas y de modelado tridimensionales****TRASLACIÓN**

$$P' = T \cdot P \quad \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

ROTACIÓN

$$\text{Rotación del eje } z: \quad P' = R_z(\theta) \cdot P \quad \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\text{Rotación del eje } x: \quad P' = R_x(\theta) \cdot P \quad \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\text{Rotación del eje } y: \quad P' = R_y(\theta) \cdot P \quad \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Para una rotación sobre un eje cualquiera:

1. Traslación de modo que el eje de rotación pase por el origen de coordenadas.
2. Rotación del eje para que coincida con uno de los ejes de coordenadas.
3. Rotación específica sobre ese eje de coordenadas.
4. Rotación inversa al punto 2.
5. Traslación inversa al punto 1.

EJEMPLO: Para tener un eje de rotación en las z y regresar el eje a su posición original. Nos dan dos puntos (P1 y P2) del eje y la rotación inversa a las manecillas (ángulo positivo):

- Se obtiene un vector del eje: $v = P2 - P1$
- Se calcula el vector unitario $u = v/|v| = (a, b, c)$ y estas componentes son los cosenos de dirección

$$\text{para el eje de rotación: } a = \frac{x_2 - x_1}{|v|} \quad b = \frac{y_2 - y_1}{|v|} \quad c = \frac{z_2 - z_1}{|v|}$$

- Si la rotación es en el sentido de las manecillas (ángulo negativo), calcularíamos el vector del eje $v = P1 - P2$
- El 1er paso es la traslación de forma que el eje pase por el origen, para ello movemos el punto P1

$$\text{al origen (P2 para rotaciones horarias): } T = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- Ahora hay que realizar las transformaciones que coloquen el eje de rotación en el eje z.

Bases, operaciones vectoriales para obtener los elementos de las matrices de rotación:

- Elementos con cosenos --> producto escalar de vectores:
 $v \cdot w = |v| \cdot |w| \cdot \cos \theta$; θ es el ángulo entre los 2 vectores.
 Si $v \cdot w = 0$ --> v y w son perpendiculares.
- Elementos con senos --> producto vectorial de vectores:
 $v \times w = |v| \cdot |w| \cdot \sin \theta$; θ es el ángulo entre los 2 vectores.
 El vector $v \times w$ es perpendicular a v y a w .

Hay 2 pasos (¡OJO! con el signo de los ángulos, hay que estudiar en cada caso concreto el sentido del giro):

- Rotación alrededor del eje x para colocar el vector u en el plano xz. --> u'

El vector $u = (a, b, c)$; $d = \sqrt{b^2 + c^2} = |u'|$ donde u' es la proyección de u sobre yz que nos sirve para calcular el ángulo de giro sobre el eje x necesario para obtener el vector u' que estará sobre el plano xz; $u' = (0, b, c)$

$$\cos \theta = c/d \quad ; \quad \sin \theta = b/d \quad ==> \quad R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & \frac{-b}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Rotación sobre el eje y para colocar el vector u'' sobre el eje z

El vector $u'' = (a, 0, d)$, donde $d = \sqrt{b^2 + c^2}$; u'' es el obtenido en la transformación anterior. Calculando sus productos con el vector unitario en dirección z (u_z) obtenemos:

$$\cos \theta = d/|u''| \quad ; \quad \sin \theta = -a/|u''| \quad ==> \quad R_y(\beta) = \begin{bmatrix} \frac{d}{|u''|} & 0 & \frac{-a}{|u''|} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{a}{|u''|} & 0 & \frac{d}{|u''|} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \quad |u''| = \sqrt{a^2 + b^2 + c^2}$$

- Ahora ya se puede aplicar el ángulo de rotación específico θ como una rotación con respecto al

$$\text{eje de las z: } R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Para concluir hay que transformar el eje de rotación de nuevo a su posición original.
- La Composición de las siete transformaciones queda:

$$R(\theta) = T^{-1} \cdot R_x^{-1}(\theta) \cdot R_y^{-1}(\theta) \cdot R_z(\theta) \cdot R_y(\theta) \cdot R_x(\theta) \cdot T$$

**ESCALACIÓN**

Con respecto al origen: $P' = S \cdot P$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Para conservar la forma original $s_x = s_y = s_z$.

Escalación con respecto a una posición fija --> trasladar el punto al origen, escalar respecto del origen y volver a trasladar el punto fijo a su posición original.

REFLEXIÓN

1. Respecto a un eje --> equivalentes a rotaciones de 180° sobre ese eje.
2. Respecto a un plano. Cuando el plano de reflexión es un plano de coordenadas (xy, xz, yz) podemos considerar la transformación como una conversión entre los sistemas del lado izquierdo y el lado derecho.

3. Ejemplo como eje el plano xy: $RF_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

4. Reflexiones con respecto a otros planos como combinación de rotaciones y reflexiones del plano de coordenadas.

RECORTE**NOTAS DE GEOMETRIA:**

- Dada una ecuación implícita del plano del tipo $ax+by+cz+d=0$ primero pasarla a paramétricas:

$x = \dots \cdot t$; $y = \dots \cdot h$; $z = \dots \cdot t + \dots \cdot h + \dots$; lo que de en cada caso.

La clave está en equiparar los valores con la expresión extendida:

P u v

$$x = p_x + u_x \cdot t + v_x \cdot h$$

$$y = p_y + u_y \cdot t + v_y \cdot h$$

$$z = p_z + u_z \cdot t + v_z \cdot h$$

Obtenemos un punto del plano $P=(p_x, p_y, p_z)$ y dos vectores del plano linealmente independientes (u y v).

El producto vectorial $u \times v$ es un vector característico del plano que podemos utilizar para realizar los giros necesarios.

¡OJO! Recordar que al calcular el producto vectorial, al realizar la suma los elementos pares de la



matriz (i,k) son positivos y los impares (j) negativos.

¡OJO! si este vector lo colocamos sobre el eje z, si se trata de una reflexión hay que hacerla respecto al plano xy que es perpendicular al eje z.

1. Definir la matriz de transformación que define una simetría respecto del plano $x+y+z+3=0$

Punto del plano $P=(0,0,-3)$; vectores del plano $u=(1,0,-1)$ $v=(0,1,-1)$

Vector característico del plano (perpendicular al plano) $w = u \times v = (1,1,1)$

Pasos de la transformación:

1. Traslación del punto p al origen

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Rotaciones que sitúan el vector w sobre el eje z:

1. Rotación sobre el eje x que coloca w sobre el plano xz. El ángulo de rotación θ es el mismo que el ángulo que forma la proyección (w_{yz}) de w en el plano yz con el eje z. El sentido de rotación es el opuesto del reloj.

$$w_{yz} = (0,1,1) ; \cos \theta = 1/\sqrt{2} ; \sin \theta = 1/\sqrt{2} \implies R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Rotación sobre el eje y que coloca w sobre el eje z. Al realizar la transformación anterior obtenemos un vector $w'=(1,0,\sqrt{2})$

Tenemos que hacer una rotación un ángulo β alrededor del eje y en el sentido del reloj (ángulo negativo). Para calcularlo utilizamos el vector unitario u_z .

$$\cos \beta = \frac{\sqrt{2}}{\sqrt{3}} ; \sin \beta = \frac{-1}{\sqrt{3}} ; R_y(\beta) = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}} & 0 & \frac{-1}{\sqrt{3}} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sqrt{3}} & 0 & \frac{\sqrt{2}}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$3. \text{ Reflexión respecto al plano xy: } RF_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



4. Rotaciones inversas al p2: $R_y(-\beta) = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{\sqrt{3}} & 0 & \frac{\sqrt{2}}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $R_x(-\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

5. Traslación inversa al punto 1: $T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

La matriz de transformación M_T es: $M_T = T^{-1} \cdot R_x(-\theta) \cdot R_y(-\theta) \cdot R_{f_z} \cdot R_y(\theta) \cdot R_x(\theta) \cdot T$

2. **Defínase la siguiente transformación tridimensional. Simetría según el plano $y=x$. Aplíquese a un cubo de lado 2 unidades, centrado en el punto (4,0,0) y con caras paralelas al citado plano**

3. **Determinar la matriz de transformación tridimensional que gira 30° alrededor del eje $x=y$, $z=0$**

- Rotación del eje -45° para que coincida con el eje x.
- Rotación de 30° alrededor del eje de las x.
- Rotación inversa al punto 1



4. **Defínase la siguiente transformación tridimensional. Giro alrededor del eje $y=x=z$**
5. **Defínase la siguiente transformación tridimensional. Giro alrededor del eje $y=x=z$ con translación constante K según ese eje**
6. **Determinar la matriz de transformación tridimensional que escala al 25% respecto del punto (1,1,1)**

7. **Defínase la siguiente transformación tridimensional. Rotación alrededor del eje $x=y, z=0$, con escalado K. Aplíquese a un cubo de lado 2 unidades, centrado en el punto (4,0,0) y con caras paralelas al plano $z=0$**

1. Rotación del eje -45° para hacerlo coincidir con el eje x.
2. Rotación específica alrededor del eje x.
3. Escalado K
4. Rotación inversa al punto 1

Para aplicarlo al cubo, multiplicamos la matriz de transformación con una matriz cuyas columnas son las coordenadas de los 8 puntos que definen el cubo en forma homogénea. Obtenemos una matriz cuyas columnas son las coordenadas de los 8 puntos del cubo una vez aplicada la transformación.

8. **Defínase la matriz de transformación que aplicada de forma consecutiva a un punto y a sus transformados define una hélice de radio R y paso P con centro en el punto p_x, p_y .**

De la expresión "con centro en el punto p_x, p_y " entiendo que el eje es la recta paralela al eje z $x=p_x, y=p_y$.

La transformación estará parametrizada por el ángulo θ , que será mayor o menor según la calidad de la aproximación que queramos a la hélice teórica. Determinará los puntos de muestreo.

1. Trasladar el eje de modo que coincida con el eje z:
$$T = \begin{bmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Rotación alrededor del eje z un ángulo θ .
3. Traslación en la dirección de z una cantidad $P \cdot \theta / 2 \theta$. (θ en radianes)
4. Traslación inversa a la del punto 1

9. **Definir las matrices de transformación que utilizada de manera secuencial permitiría representar un movimiento planetario de un objeto en una órbita de radio R sobre el plano $x+y+z=0$. El objeto rota sobre si mismo.**

A.-Rotación sobre si mismo:

Si el centro del

B.-Traslación en la órbita sobre el plano:

Como vector característico del plano tenemos (1,1,1) y pasa por el pto (0,0,0).

1. Traslación del centro de la órbita (a,b,c) al origen de coordenadas
$$T = \begin{bmatrix} 1 & 0 & 0 & -a \\ 0 & 1 & 0 & -b \\ 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



2. Rotación del eje alrededor del eje x para colocarlo en el plano xy:

La proyección sobre el plano yz del vector característico es (0,1,1). El giro un ángulo α en el sentido contrario a las agujas del reloj es el mismo que el ángulo que forma la proyección con el eje z:

$$\cos \alpha = \frac{1}{\sqrt{2}} ; \quad \sin \alpha = \frac{1}{\sqrt{2}} ; \quad R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Rotación alrededor del eje y para colocar el vector sobre el eje z (ángulo negativo)

$$\cos \beta = \frac{\sqrt{2}}{\sqrt{3}} ; \quad \sin \beta = \frac{-1}{\sqrt{3}} ; \quad R_y(\beta) = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}} & 0 & \frac{-1}{\sqrt{3}} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sqrt{3}} & 0 & \frac{\sqrt{2}}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. Rotación específica que se produce en cada paso alrededor del eje z (traslación en la órbita).

El ángulo θ que ha de rotar para cada muestreo será un parámetro y su valor concreto se asignará dependiendo de la frecuencia de muestreo deseada que puede depender a su vez de las características del dispositivo de despliegue y de las técnicas que se apliquen para mejorar el resultado (por ejemplo si se utilizan técnicas de antialias basadas en sobremuestreo), su signo indicará el sentido del giro.

$$\text{Así la rotación con como parámetro será } R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5. Rotación específica que se produce en cada paso sobre sí mismo (nota: si los ejes de traslación y rotación del objeto son paralelos los pasos 5.2 y 5.4 no son necesarios)

1. Traslación del centro del objeto al origen de coordenadas será una cantidad -R

??????

2. Rotaciones adecuadas una sobre el eje x y otra sobre el y para colocar el eje de rotación sobre sí mismo sobre el eje z.

3. Rotación específica sobre sí mismo en cada paso

4. Rotaciones inversas al punto 5.2

5. Traslación inversa al punto 5.1

6. Rotación inversa al punto 3

7. Rotación inversa al punto 2

8. Traslación inversa al punto 1