

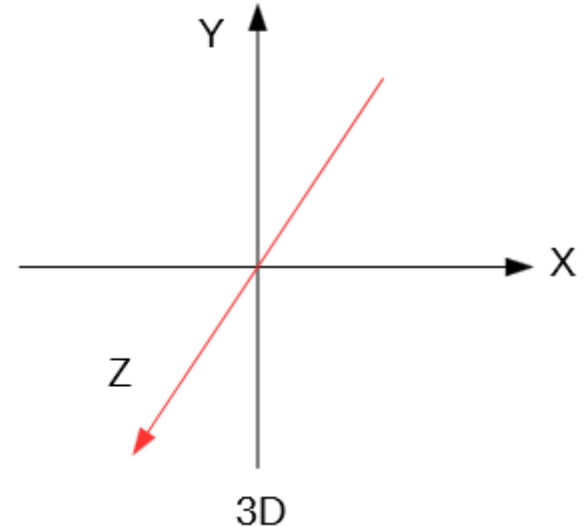
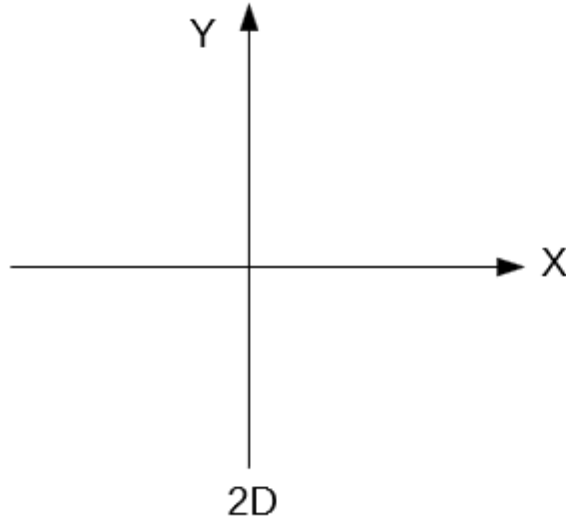


# Graficas por computadora 3D

## Unidad II

# Sistema de coordenadas 3D

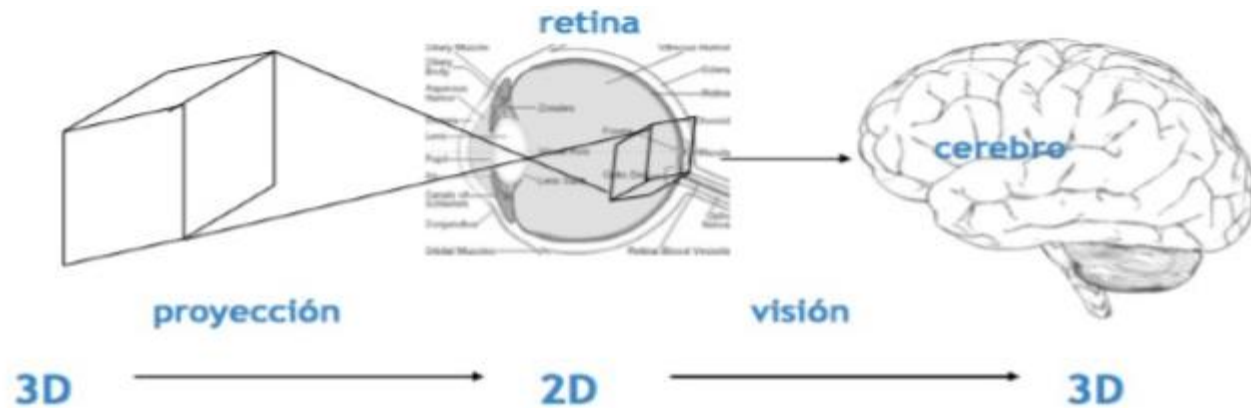
Ya hemos hablado de transformaciones bidimensionales de objetos en 2D, las cuales no representan ningún problema para su proyección en la pantalla de la computadora (o el papel de la impresora) porque son superficies planas, es decir son de solo dos dimensiones, pero necesitamos de una forma de proyectar nuestros objetos tridimensionales.



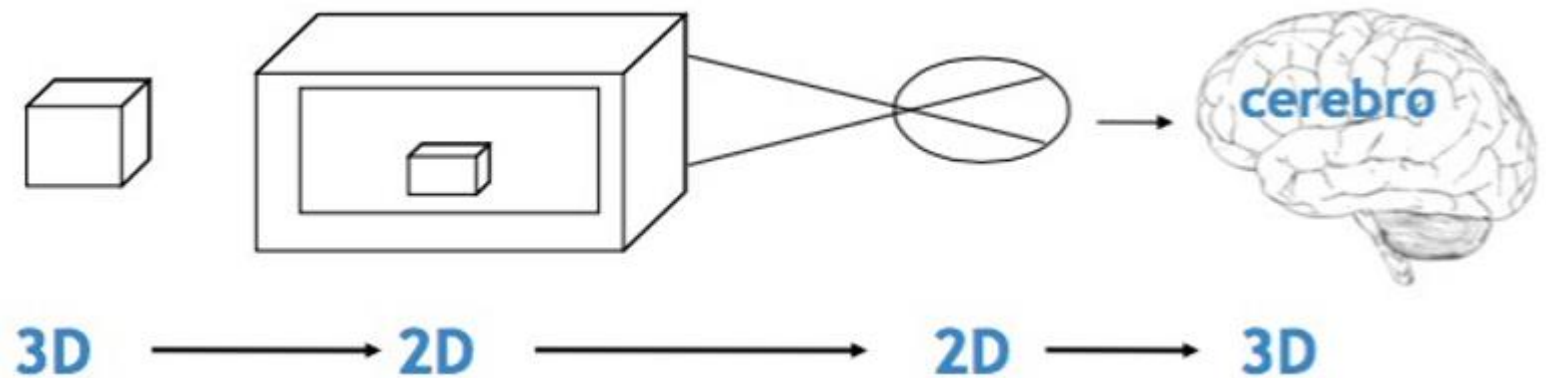
# Proceso gráfico en la vista tridimensional

El proceso de visualización tridimensional es más complejo que el bidimensional, esta complejidad proviene del hecho de que los dispositivos de display son bidimensionales.

La solución a ésta problemática viene dada por las proyecciones, las cuales transforman objetos tridimensionales en objetos en un plano de proyección bidimensional.



# Proceso gráfico en la vista tridimensional

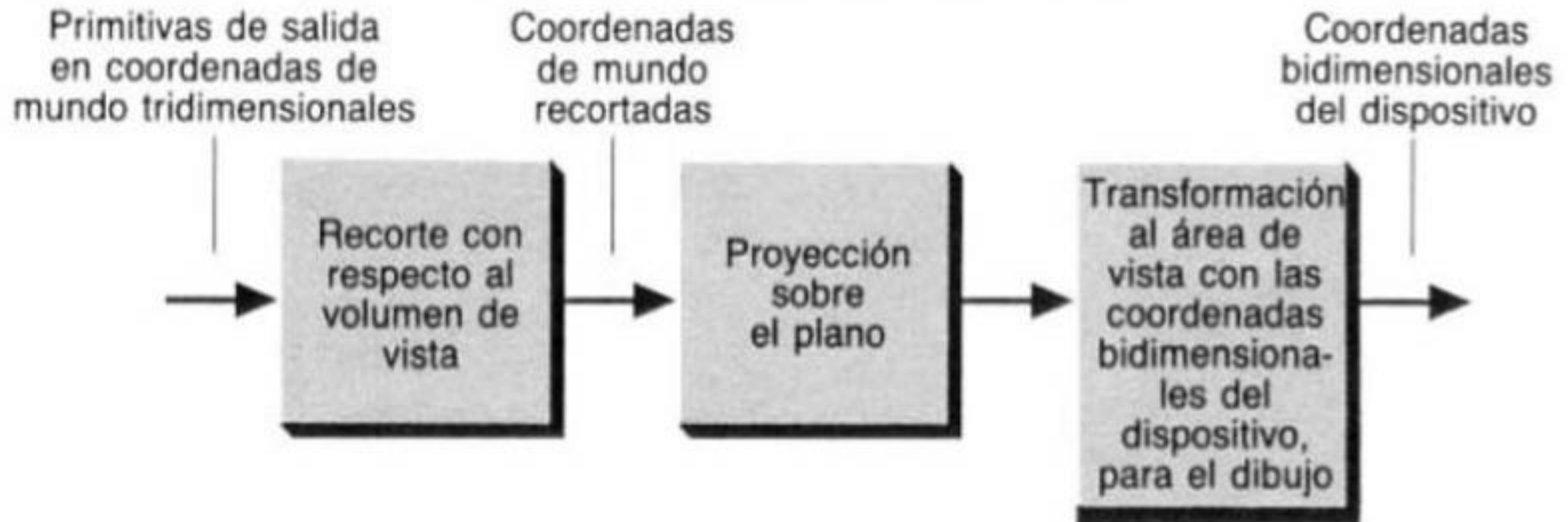


Vista tridimensional  
en computación gráfica

Aquí esta nuestra labor...

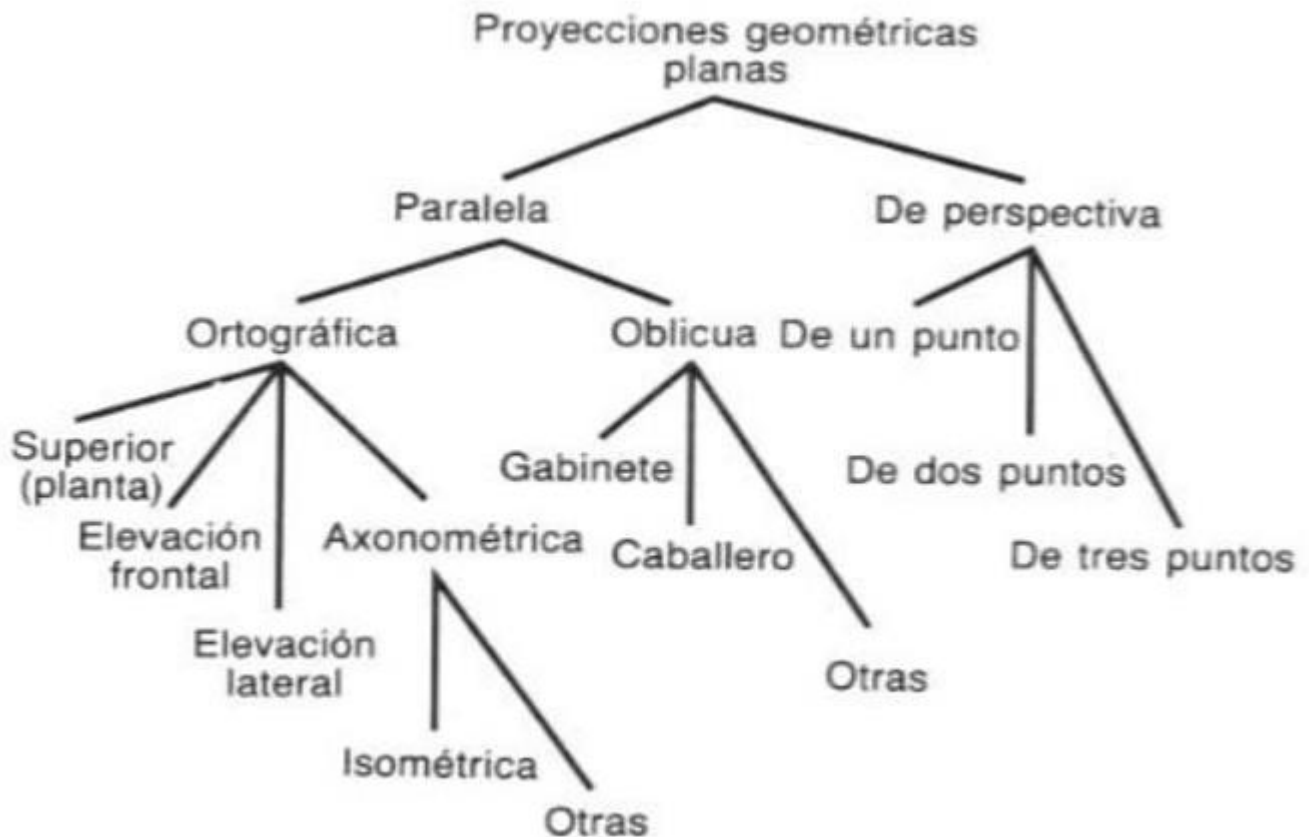
# Proceso gráfico en la vista tridimensional

## Modelo conceptual



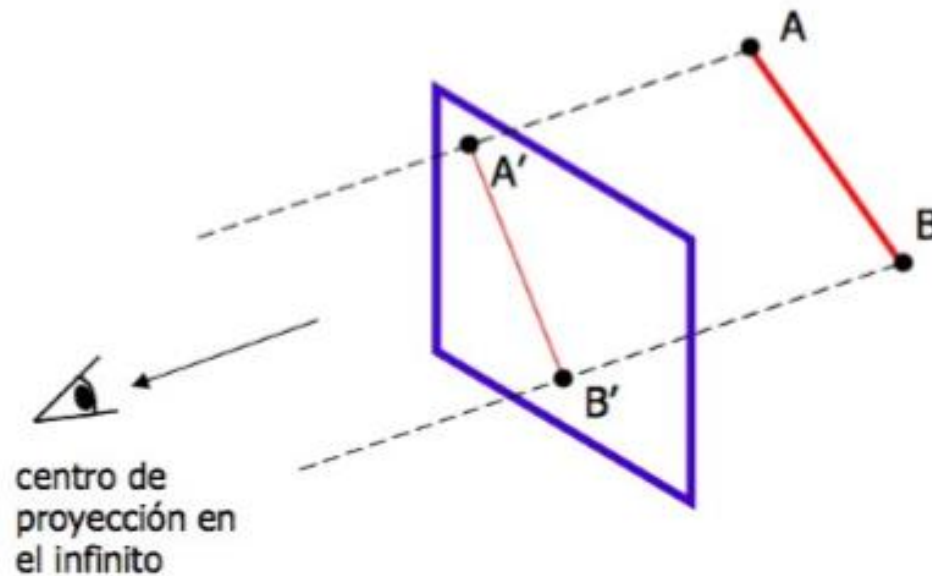
# Proyecciones

## Subclases de las proyecciones planas



# Proyección paralela

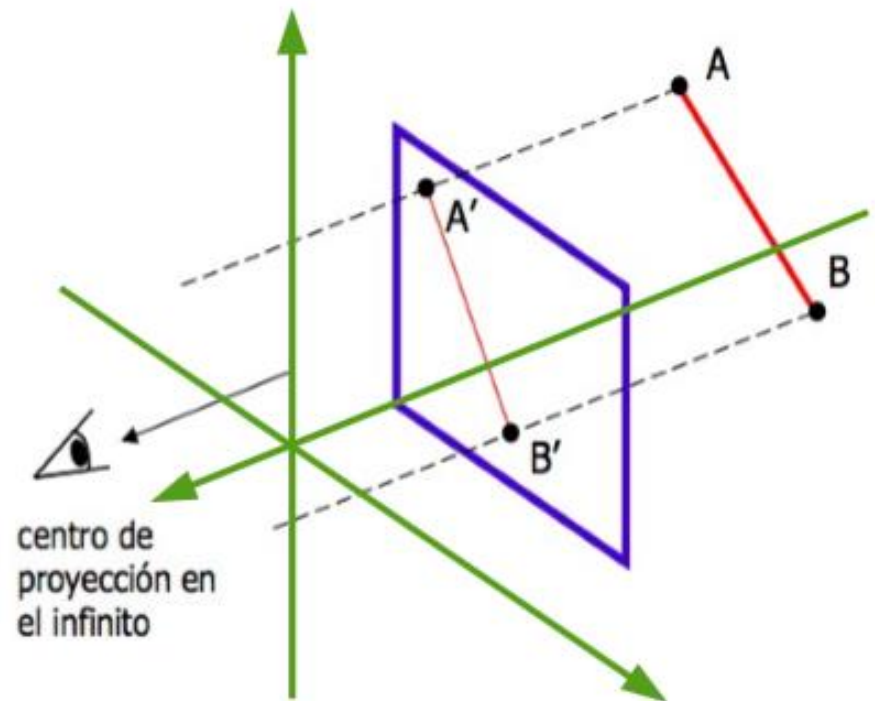
La proyección más simple posible consiste en simplemente descartar la coordenada z. Una proyección paralela se forma extendiendo líneas paralelas desde cada vértice del objeto hasta que intersecten el plano de visión, los puntos de intersección con el plano son los vértices proyectados y se conectan mediante líneas de la misma manera que los vértices originales están conectados en el objeto tridimensional.



# Proyección paralela

El caso particular donde el plano de visión es el plano  $xy$  y la dirección de proyección es paralela al eje  $z$  corresponde con el simple descarte de la coordenada  $z$ . En una proyección paralela general se puede elegir cualquier dirección de proyección.

Digamos que la dirección de proyección está definida por el vector  $(x_p, y_p, z_p)$  y que la imagen se proyecta sobre el plano  $xy$ . Si tenemos un punto en el objeto cuyas coordenadas son  $(x_1, y_1, z_1)$  y queremos determinar el punto proyectado  $(x_2, y_2)$ .





# Proyección paralela

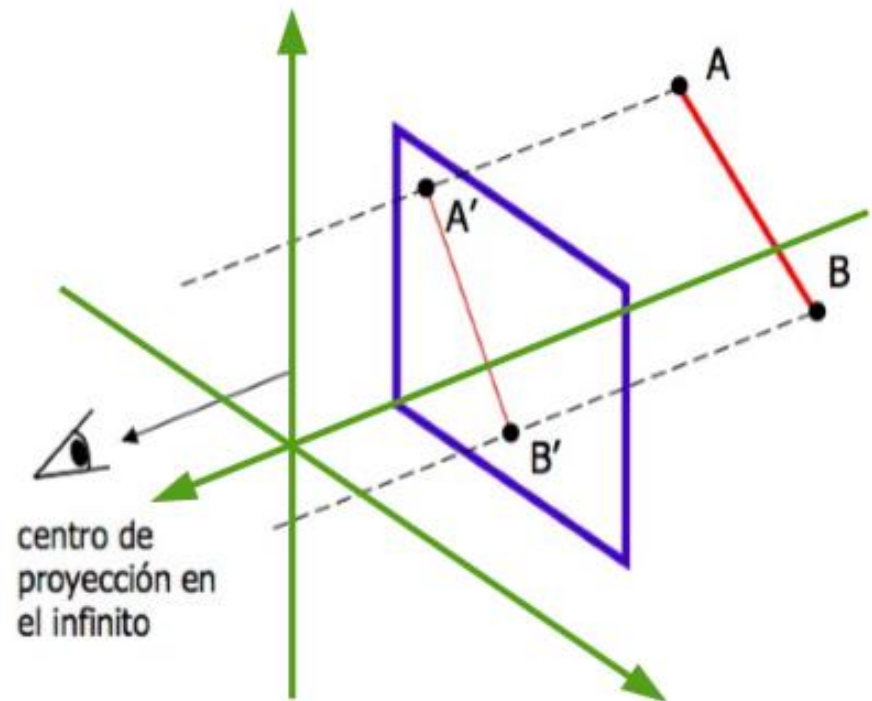
Ecuaciones paramétricas

$$x = x_1 + x_p u$$

$$y = y_1 + y_p u$$

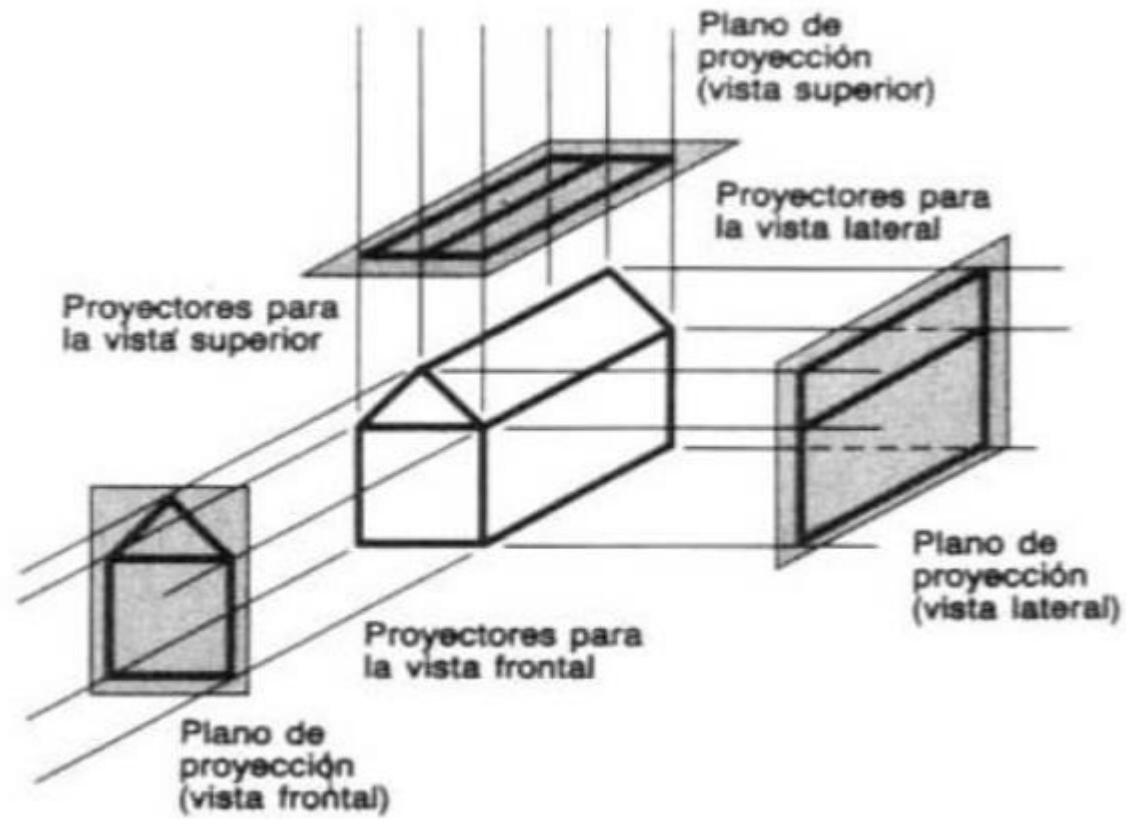
$$z = z_1 + z_p u$$

Queremos determinar  $x, y$



# Proyección paralela

Ejemplos...



# Proyección paralela

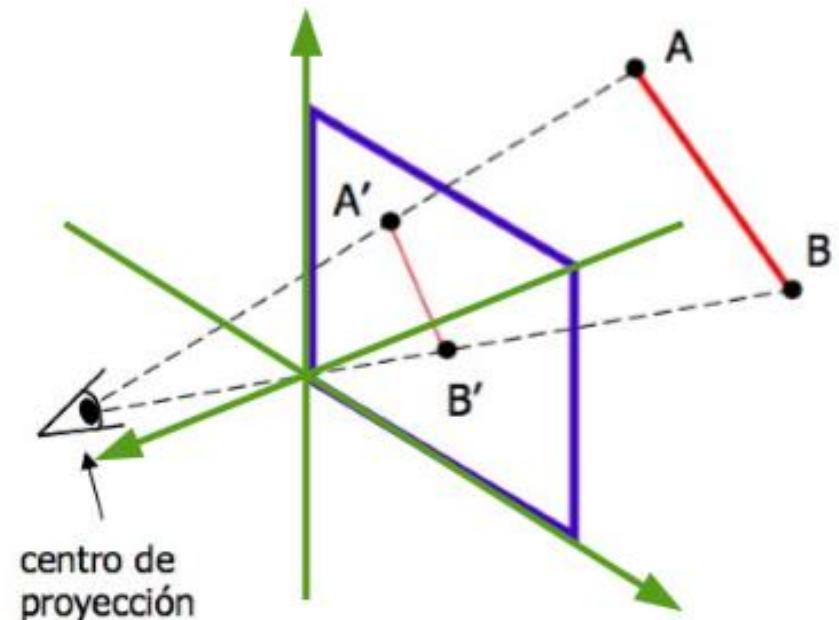
Isométrica...



# Proyección en perspectiva

En una proyección en perspectiva, mientras más lejos se encuentra un objeto del observador, mas pequeño parece ser, esto provee al observador con una indicación de profundidad, es decir, que partes de un objeto se encuentran mas cercanas o mas lejanas. En la proyección en perspectiva las líneas de proyección no son paralelas sino que convergen en un punto llamado centro de proyección.

La línea que va del centro de proyección ( $x_c, y_c, z_c$ ) hacia el punto a proyectar ( $x_l, y_l, z_l$ )



# Proyección en perspectiva

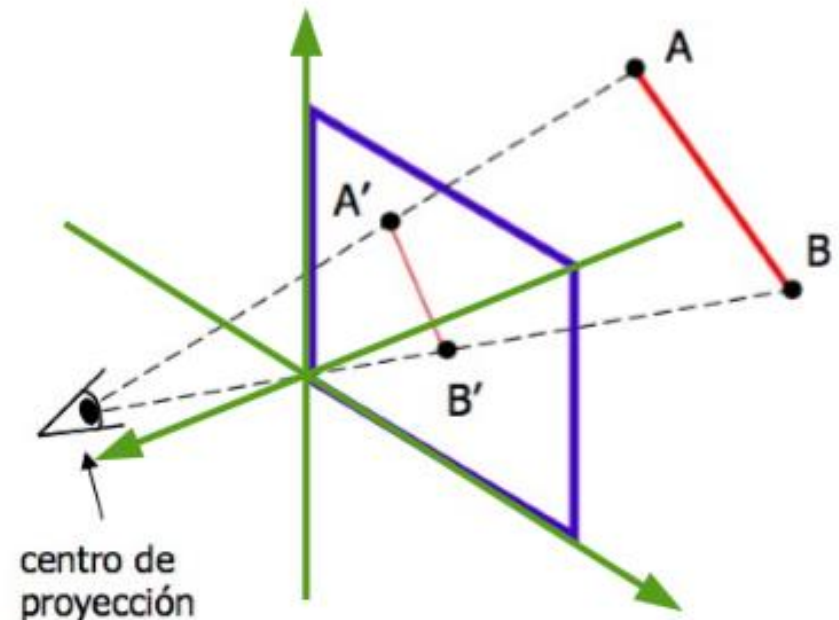
Ecuaciones paramétricas

$$x = x_c + (x_1 - x_c)u$$

$$y = y_c + (y_1 - y_c)u$$

$$z = z_c + (z_1 - z_c)u$$

Queremos determinar  $x, y$





# Primitivas gráficas

Migración de dibujos 2D a 3D



# Presentación de estructuras, vértices y procesamiento

a

# Redes de estructuras jerárquicas

a





# Transformaciones tridimensionales

Traslación

Escalamiento

**Rotación**

# Polilíneas

Consta de varias líneas rectas, estas líneas están conectadas y ordenadas de manera que el final coincide con el inicio de la segunda, el final de la segunda coincide con el inicio de la tercera y así sucesivamente.

- Polilíneas cerradas
- Polilíneas abiertas

# Curvas Splines cúbicas naturales

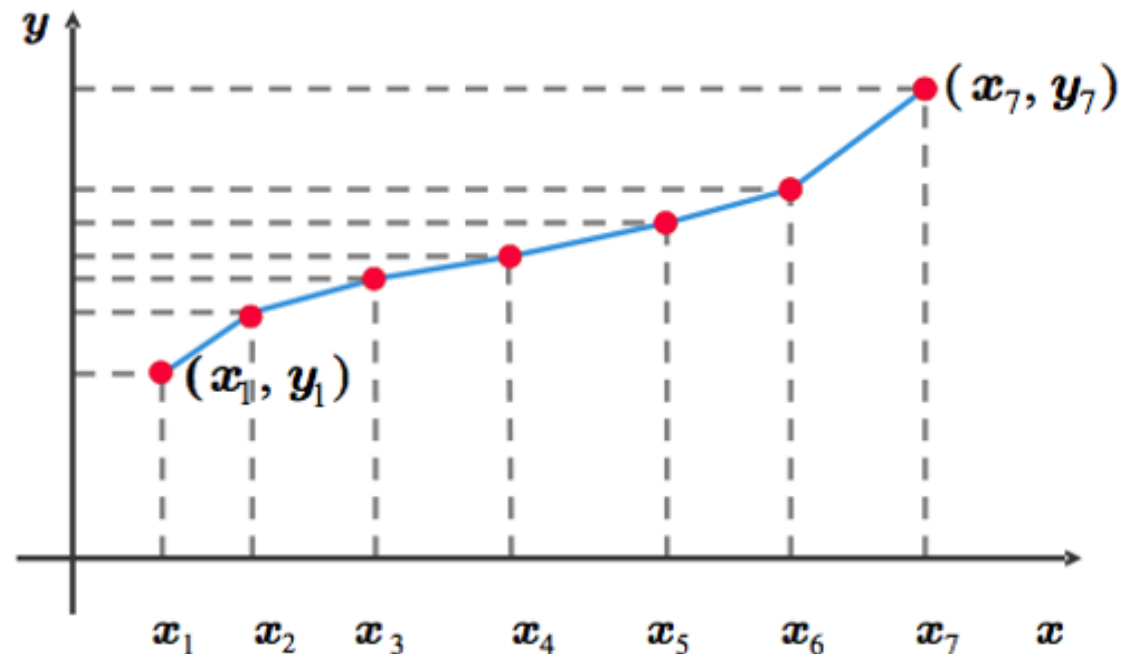
Un spline solía ser un instrumento utilizado por dibujantes para hacer curvas suaves, consistía en una banda flexible que se hacía pasar por puntos específicos, actualmente una spline es una curva formada por segmentos polinomiales que satisfacen condiciones de continuidad entre ellos.

- Splines de interpolación
- Splines de aproximación

# Curvas y superficies

## Representación gráfica de curvas bidimensionales

La representación gráfica de una curva en un ordenador es una línea poligonal construida uniendo mediante segmentos rectos un conjunto discreto y ordenado de puntos:  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$



# Curvas y superficies

La línea así obtenida tendrá mayor apariencia de “suave” cuanto más puntos se utilicen para construirla, ya que los segmentos serán imperceptibles.

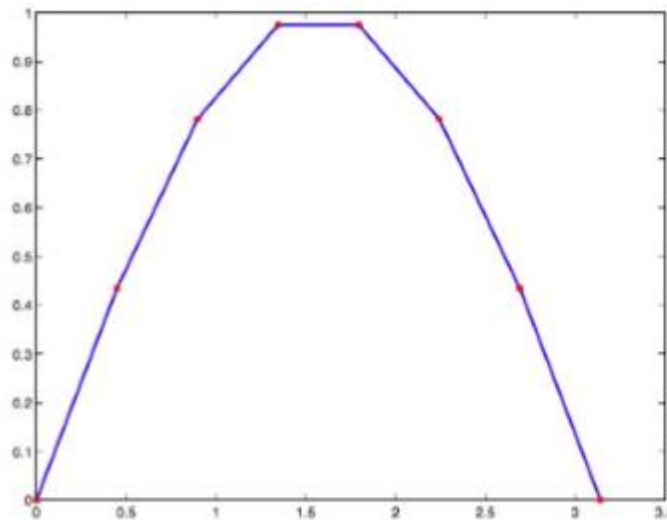


Figura 3.2: Representación de  $y = \sin(x)$  en  $[0, \pi]$  con 8 puntos.

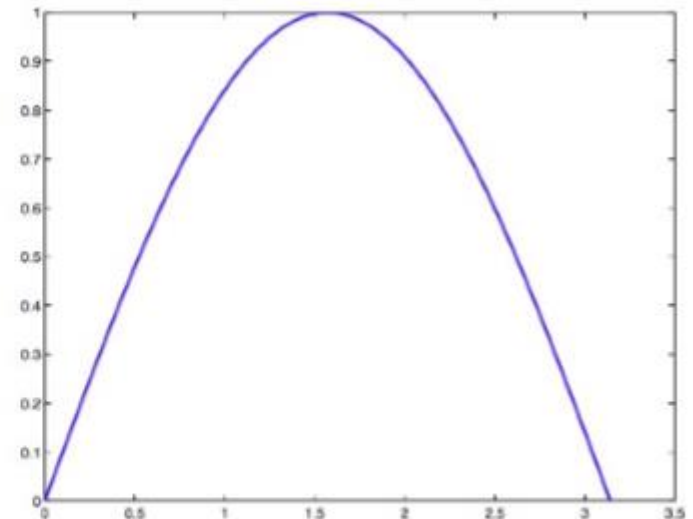


Figura 3.3: Representación de  $y = \sin(x)$  en  $[0, \pi]$  con 100 puntos.

# Curvas y superficies

## Representación gráfica de funciones de una variable real

La relación  $y = f(x)$ , es una función de una variable real, se puede representar gráficamente mediante una curva plana. La construcción de dicha grafica en la computadora básicamente sigue los siguientes pasos:

- Construir un conjunto de puntos (tantos como se quiera) en el intervalo determinado  $[a,b]$ , que serán las abscisas de los puntos que determinan la poligonal a construir:

$$\{a = x1, x2, \dots, xn = b\}$$

- Calcular los valores de la función  $f$  en los puntos anteriores:

$$\{y1 = f(x1), y2 = f(x2), \dots, yn = f(xn)\}$$

- Unir los puntos  $(x,y)$  consecutivos mediante segmentos rectos.

# Curvas y superficies

Cuando una curva viene definida por una relación del tipo  $y = f(x)$  se dice que **está definida de forma explícita**, pero también puede ser definida por  $x = g(y)$

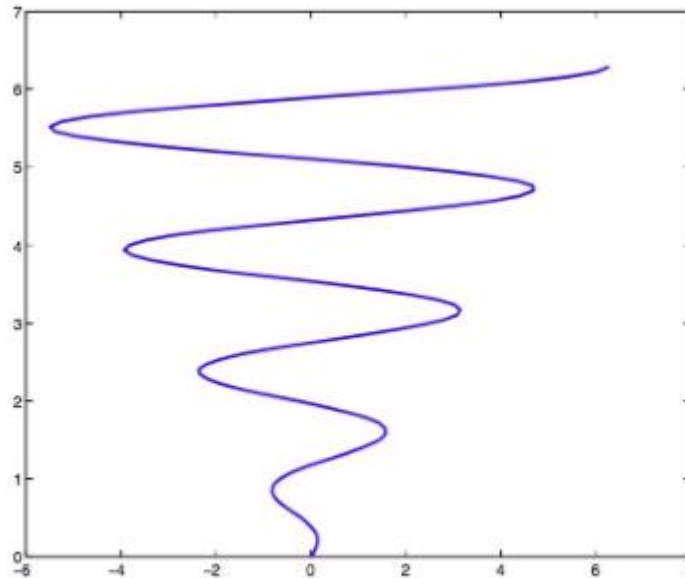


Figura 3.4: Curva definida por la relación  $x = y \cos(4y)$ ,  $y \in [0, 2\pi]$ .

# Curvas y superficies

## Curvas planas definidas mediante ecuaciones paramétricas

Otra forma de definir una curva plana es mediante sus ecuaciones paramétricas, en la cual los puntos  $(x, y)$  que forman la curva vienen dados por dos funciones que dependen de una variable auxiliar:

$$x = f(t), y = g(t), t \in [a, b].$$

La variable  $t$  se suele llamar el parámetro de la curva.

- Construir un conjunto de valores del parámetro  $t \in [a, b]$ :  
 $\{a = t_1, t_2, \dots, t_n = b\}$
- Calcular los valores  $x$  y de  $y$  para dichos valores del parámetro:  
 $\{x_1 = f(t_1), x_2 = f(t_2), \dots, x_n = f(t_n)\}$   
 $\{y_1 = g(t_1), y_2 = g(t_2), \dots, y_n = g(t_n)\}$
- Unir los puntos  $(x, y)$  consecutivos mediante segmentos rectos.



# Curvas y superficies

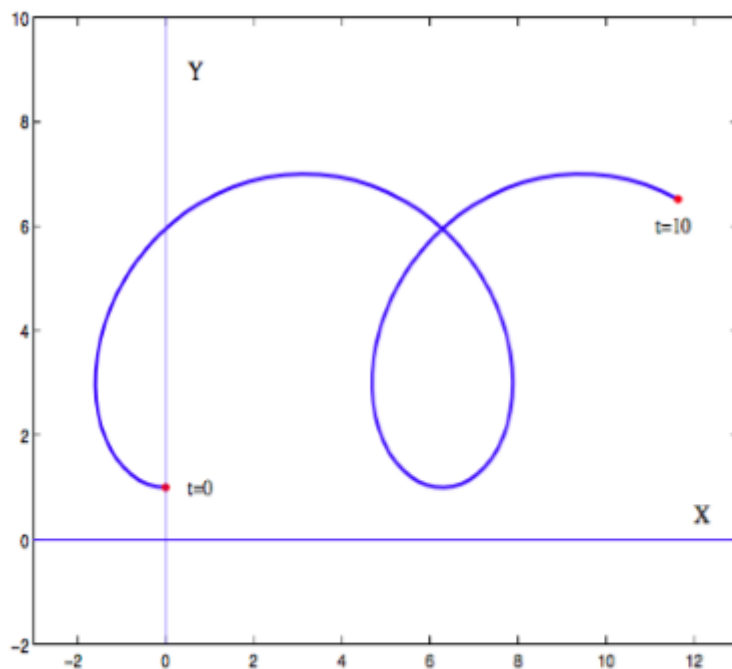


Figura 3.5: Representación de la curva de ecuaciones paramétricas  $x = t - 3 \sin(t)$ ,  $y = 4 - 3 \cos(t)$  para  $t \in [0, 10]$ . Obsérvese que no hay eje  $t$ .

t	x	y
0	0	1
1	-1.5	2.4
2	-0.7	5.2
3	2.6	7.0
4	6.3	6
5	7.9	3.1
6	6.8	1.1
7	5.0	1.7
8	5.0	4.4
9	7.8	6.7
10	11.6	6.5

# Curvas y superficies

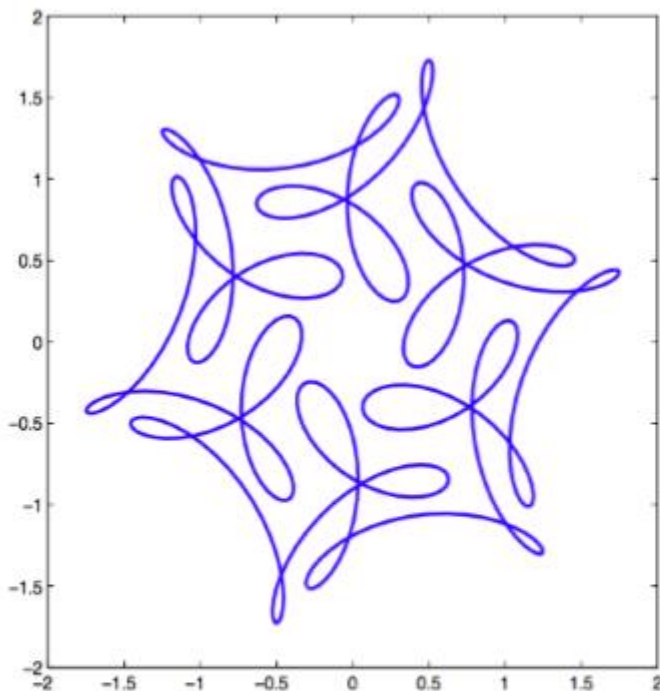


Figura 3.6: Representación de la curva de ecuaciones paramétricas  $x = \cos(t) + 1/2 \cos(7t) + 1/3 \sin(17t)$ ,  $y = \sin(t) + 1/2 \sin(7t) + 1/3 \cos(17t)$ , para  $t \in [0, 2\pi]$ .

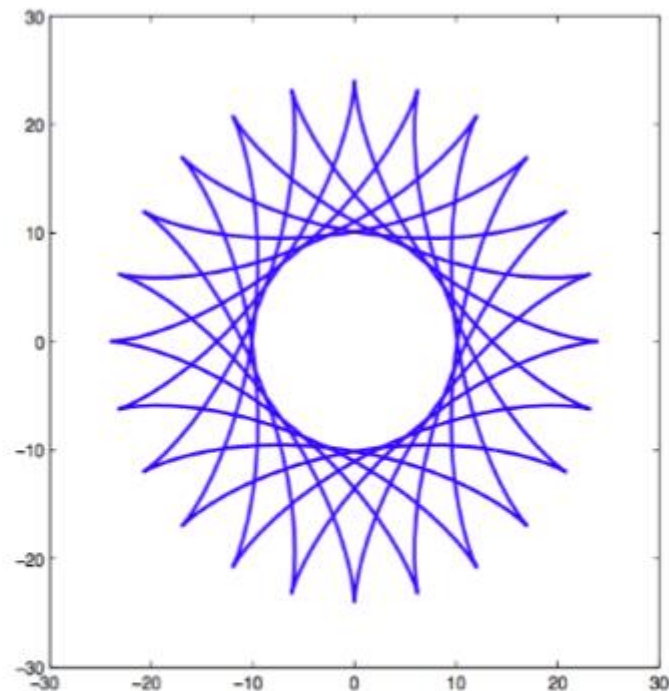


Figura 3.7: Representación de la curva de ecuaciones paramétricas  $x = 17 \cos(t) + 7 \cos(\frac{17}{7}t)$ ,  $y = 17 \sin(t) - 7 \sin(\frac{17}{7}t)$ , para  $t \in [0, 14\pi]$ .

# Curvas y superficies

El paso de las coordenadas polares a cartesianas y viceversa se efectúa mediante las siguientes fórmulas, tomando el polo como origen de coordenadas y el eje polar como semi-eje positivo de abscisas

$$x = r \cos(\theta), \quad y = r \sin(\theta);$$

$$r = \sqrt{x^2 + y^2}, \quad \theta = \arctan \frac{y}{x}$$

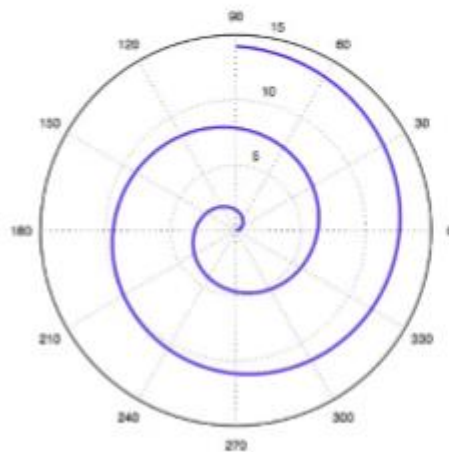


Figura 3.10: Curva de ecuación, en coordenadas polares,  $r = \theta$ ,  $\theta \in [0, 9\pi/2]$

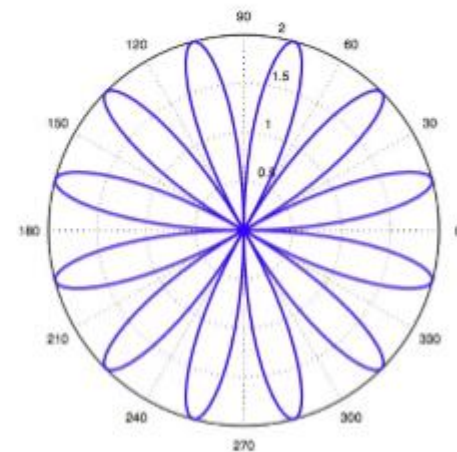


Figura 3.11: Curva de ecuación, en coordenadas polares,  $r = 2\sin(6\theta)$ ,  $\theta \in [0, 2\pi]$ .

# Curvas y superficies

Curvas en tres dimensiones La gráfica de una curva tridimensional se dibuja, igual que la bidimensional, uniendo mediante segmentos rectos (en 3D) los puntos consecutivos de un conjunto discreto y ordenado. Estos segmentos se “proyectan” sobre el plano del dibujo para obtener impresión tridimensional.

La forma más sencilla de describir matemáticamente una curva tridimensional es mediante sus ecuaciones paramétricas. Estas ecuaciones describen los valores de las coordenadas (x, y, z) de cada punto de la curva en función de una variable auxiliar, llamada parámetro:

$$x = f(t)$$

$$y = g(t)$$

$$z = h(t)$$

$$\text{para } t \in [a, b]$$

# Curvas y superficies

## Curvas en tres dimensiones

Para dibujar su gráfica habrá que construir las coordenadas de un conjunto discreto y ordenado de puntos de la curva. De forma similar a como se hizo en el caso bidimensional:

- Construir un conjunto de valores del parámetro  $t \in [a, b]$ :

$$\{a = t_1, t_2, \dots, t_n = b\}$$

- Calcular los valores de  $x$ , de  $y$  y de  $z$  para dichos valores del parámetro:

$$\begin{aligned} \{x_1 = f(t_1), x_2 = f(t_2), \dots, x_n = f(t_n)\} \\ \{y_1 = g(t_1), y_2 = g(t_2), \dots, y_n = g(t_n)\} \\ \{z_1 = h(t_1), z_2 = h(t_2), \dots, z_n = h(t_n)\} \end{aligned}$$

- Unir los puntos  $(x, y, z)$  consecutivos mediante segmentos rectos.

# Curvas y superficies

## Curvas en tres dimensiones

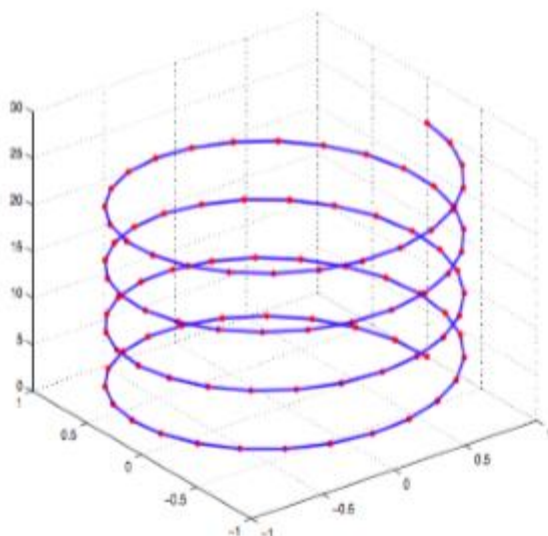


Figura 3.12: Gráfica de la curva 3D de ecuaciones paramétricas  $x(t) = \cos(t)$ ,  $y(t) = \sin(t)$ ,  $z(t) = t$ ,  $t \in [0, 8\pi]$ .

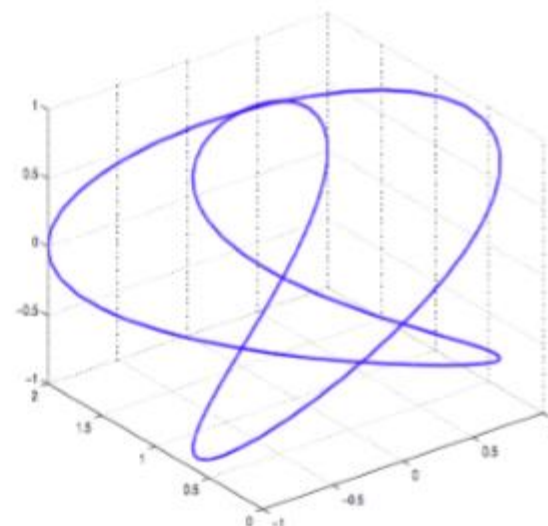


Figura 3.13: Gráfica de la curva  $x(t) = \cos(3t)$ ,  $y(t) = 2\cos^2(t)$ ,  $z(t) = \sin(2t)$ ,  $t \in [-\pi, \pi]$ .

# Curvas y superficies

## Gráficas de funciones de dos variables: superficies

La ecuación explícita:  $z = f(x, y)$

la función  $f$  le hace corresponder un valor  $z$  que representa la “altura” (o la profundidad) de la superficie en ese punto.

Para dibujar la superficie es preciso disponer de una “discretización” del conjunto total de puntos en el que está definida la función, es decir un conjunto de polígonos (normalmente triángulos o rectángulos) cuya unión sea el conjunto total.

Un mallado en rectángulos de un dominio rectangular es fácil de construir a partir de sendas particiones de sus lados. Un mallado en triángulos es más complicado y precisa de algoritmos especializados.



# Curvas y superficies

## Gráficas de funciones de dos variables: superficies

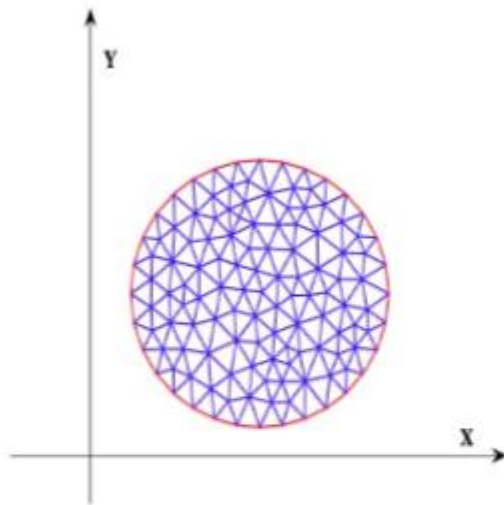


Figura 3.14: Mallado en triángulos de un dominio de frontera curva

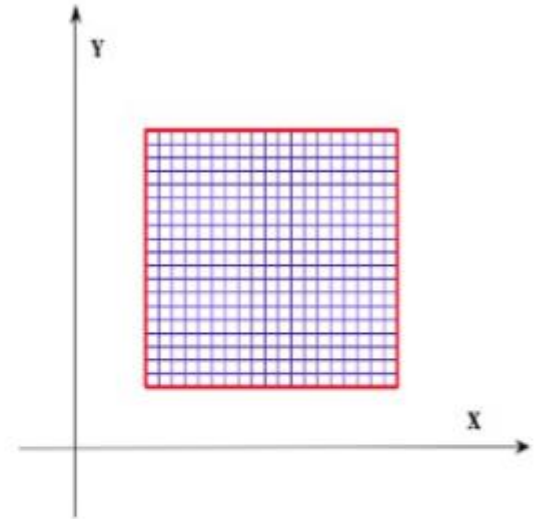


Figura 3.15: Mallado en rectángulos de un dominio rectangular



# Curvas y superficies

## Gráficas de funciones de dos variables: superficies

La forma de proporcionar los datos en uno y otro caso es diferente. Un mallado rectangular de un dominio  $\Omega = [a, b] \times [c, d]$  queda definido mediante las particiones de los intervalos  $[a, b]$  y  $[c, d]$  cuyo producto cartesiano produce los nodos de la malla:  $\{x_1, x_2, \dots, x_n\}$  e  $\{y_1, y_2, \dots, y_m\}$ .

Para definir un mallado mediante triángulos es preciso, por un lado numerar sus vértices y disponer de sus coordenadas,  $(x_i, y_i)$ ,  $i = 1, \dots, n$  y, por otro, numerar sus triángulos y describirlos enumerando, para cada uno, sus tres vértices.

Elevando cada vértice del mallado según el valor de  $f$  en ese punto se consigue una representación de la superficie como una red deformada, y podemos dar un color a cada arista dependiendo del valor de la función en sus extremos.

# Curvas y superficies

## Gráficas de funciones de dos variables: superficies

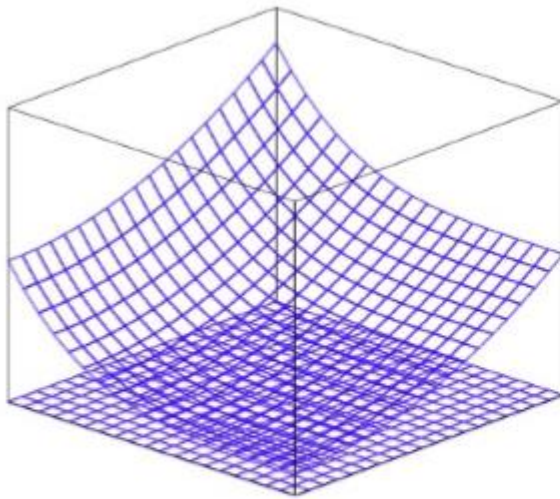


Figura 3.18: Red rectangular deformada.

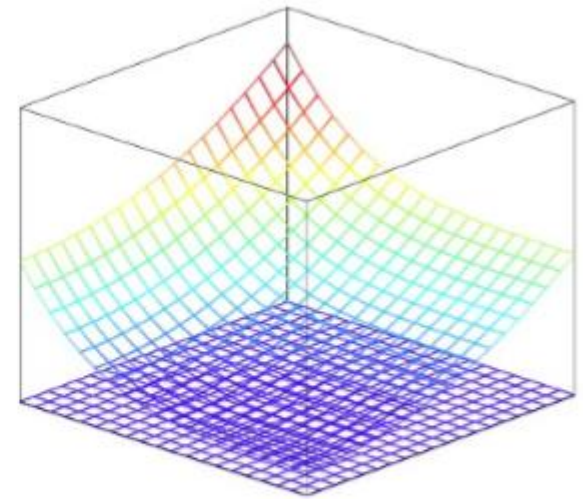


Figura 3.19: Red rectangular deformada. El color de las aristas depende del valor de la función.

# Curvas y superficies

## Gráficas de funciones de dos variables: superficies

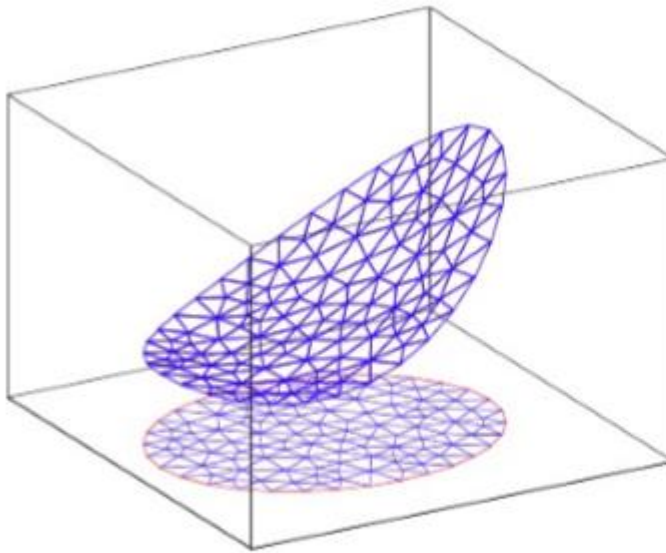


Figura 3.17: Red triangular deformada.

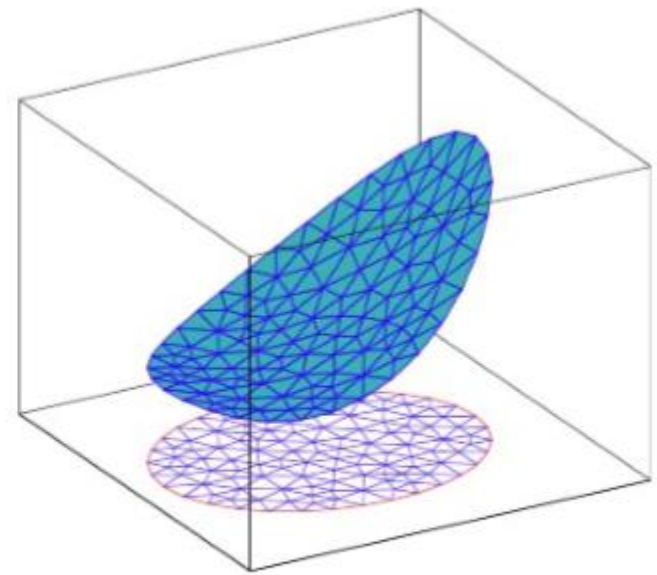


Figura 3.21: Todas las caras del mismo color.

# Curvas y superficies

## Gráficas de funciones de dos variables: superficies

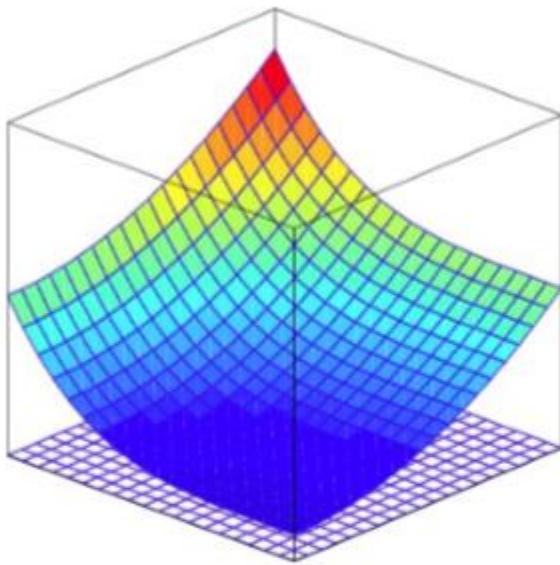


Figura 3.22: Color constante en cada cara, dependiente de la altura.

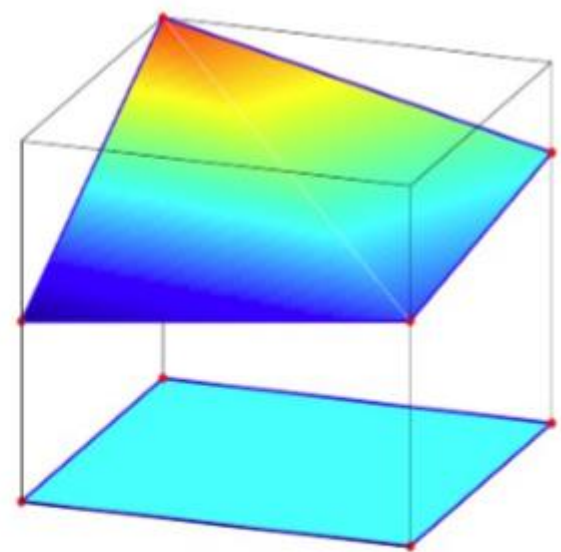


Figura 3.23: Color interpolado a partir de los valores en los vértices.

# Curvas y superficies

## Superficies definidas mediante ecuaciones paramétricas

Una superficie en el espacio de tres dimensiones pueden también venir definida mediante ecuaciones paramétricas.

$$x = f1(s,t), y = f2(s,t), z = f3(s,t), (s,t) \in [a,b] \times [c,d]$$

En este caso, para construir la grafica de la superficie es preciso crear una desratización del dominio donde varían los parámetros,  $[a, b] \times [c, d]$ , y utilizar las ecuaciones paramétricas para calcular los puntos correspondientes sobre la superficie.

Por ejemplo, para dibujar la superficie cilíndrica definida por las ecuaciones

$$x(t, \varphi) = (2 + \cos(t)) \cos(\varphi)$$

$$y(t, \varphi) = (2 + \cos(t)) \sen(\varphi)$$

$$z(t, \varphi) = t$$

$$t \in [0, 2\pi], \varphi \in [0, 2\pi]$$

# Curvas y superficies

## Superficies definidas mediante ecuaciones paramétricas

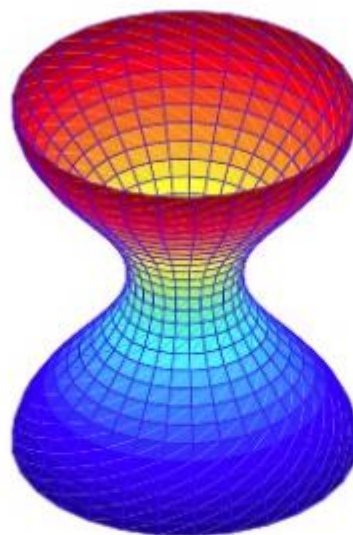


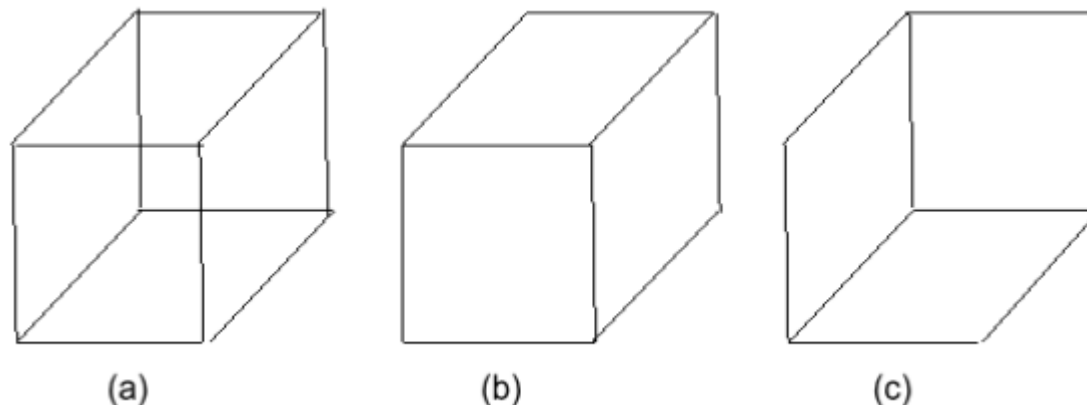
Figura 3.24: Superficie cilíndrica de ecuaciones paramétricas  $x = (2 + \cos(t)) \cos(\varphi)$ ,  $y = (2 + \cos(t)) \sin(\varphi)$ ,  $z = t$ .

# Supresión de líneas y superficies ocultas

Un problema importante en el proceso de desplegar imágenes que luzcan reales es el de diferenciar las partes visibles de aquellas que no se podrían ver desde la posición de algún observador.

## Supresión de segmentos de líneas ocultas

No solamente para mejorar el aspecto de una grafica o de un objeto sino para eliminar la ambigüedad de una figura y por ende poderla entender, es necesario eliminar las líneas o los segmentos de línea que no deberían verse.



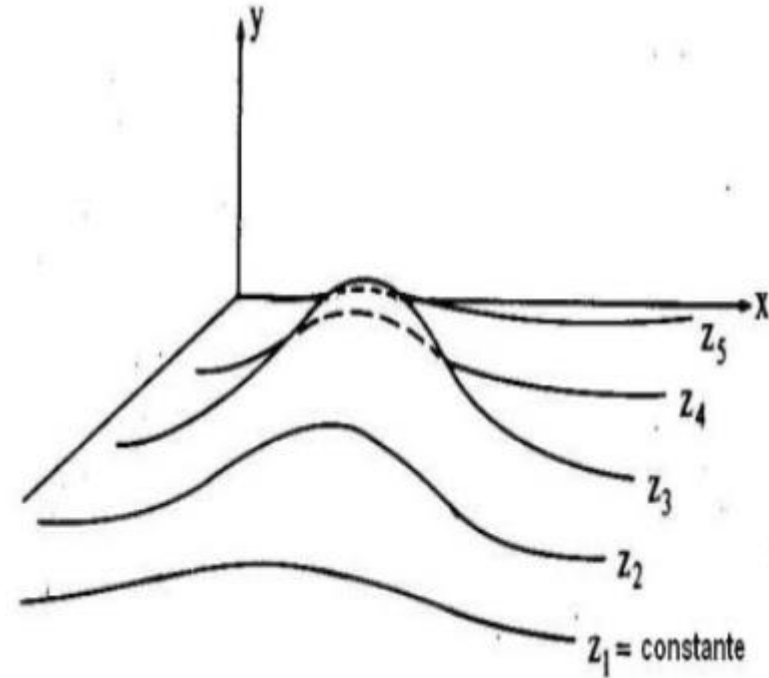


# Supresión de líneas y superficies ocultas

## Algoritmo del horizonte flotante

Si para cada valor dado de  $x$ , el valor  $y$  de la curva en el plano actual es mayor que el valor  $y$  para todas las curvas anteriores, entonces la curva es visible para ese valor específico de  $x$ , en caso contrario será invisible.

Para implementar este algoritmo simplemente hay que mantener un arreglo del mismo tamaño que la resolución de la imagen en el eje  $x$ . Los valores de este arreglo representan el horizonte y este horizonte “flota” a medida que cada curva es dibujada.



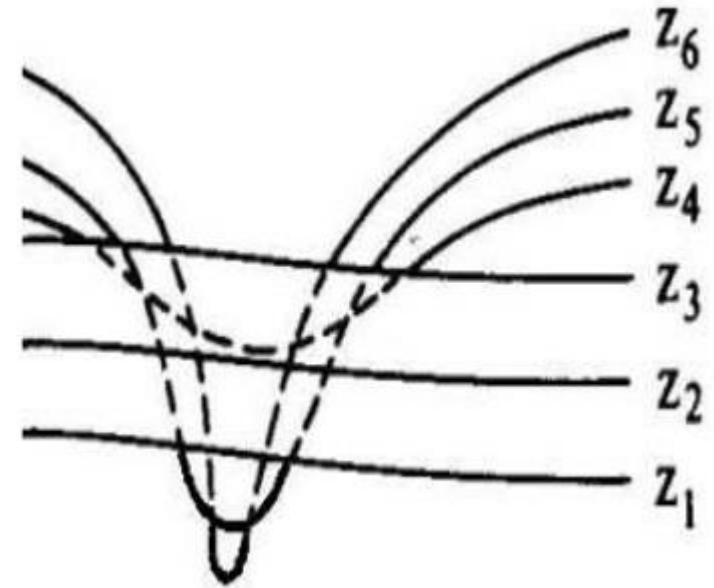


# Supresión de líneas y superficies ocultas

## Algoritmo del horizonte flotante

Si para algún valor dado de  $x$ , el valor  $y$  correspondiente de la curva en el plano actual es superior al máximo valor o inferior al mínimo valor entre todas las curvas anteriores, entonces la curva actual en dicho valor  $x$  es visible, si no se cumple ninguna de las dos cosas, entonces es invisible.

Para implementar el algoritmo del horizonte flotante se requieren entonces dos arreglos uno para almacenar los valores máximos y el otro para almacenar los mínimos, a estos arreglos se les denominan horizontes flotantes superior e inferior respectivamente.

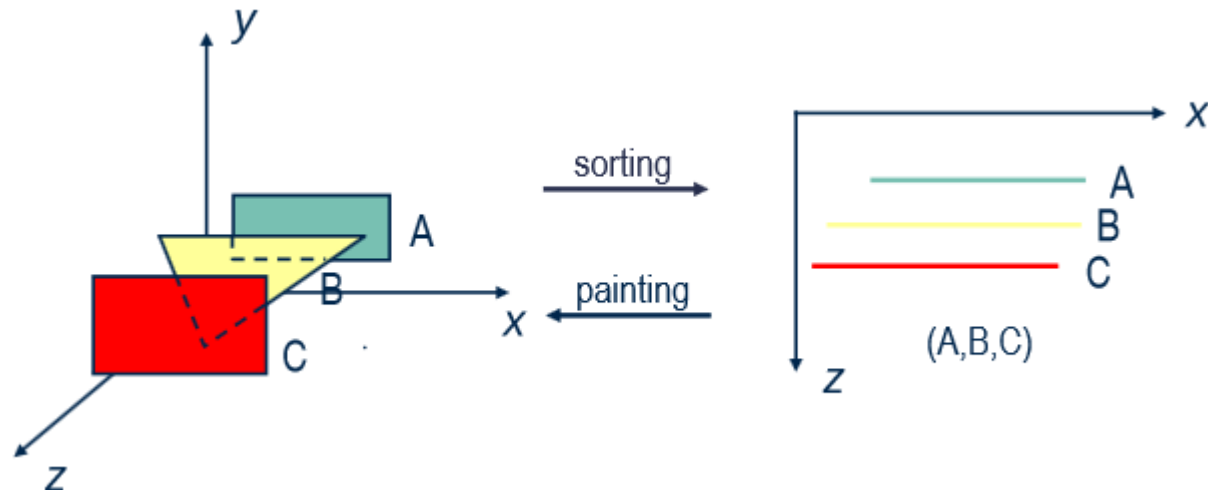


# Supresión de líneas y superficies ocultas

## Algoritmo del pintor

Se procede a dibujar todos los polígonos que aparecen en la escena de atrás hacia adelante, de manera que los polígonos situados en primer plano se dibujan sobre los polígonos situados en el fondo.

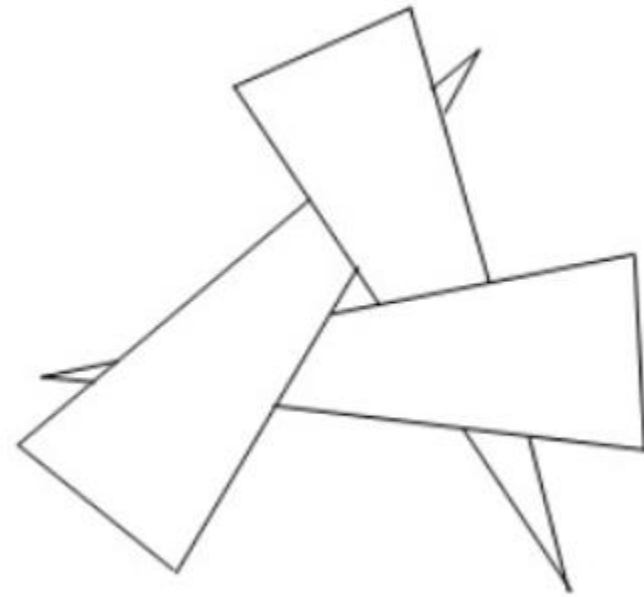
De esta manera, los polígonos más cercanos obstaculizan la visión de los polígonos más lejanos.



# Supresión de líneas y superficies ocultas

## Algoritmo del pintor

- La clasificación se realiza sobre la coordenada  $z$  máxima de cada polígono.
- Existe una situación en la cual el algoritmo del pintor no es capaz de realizar una clasificación correcta de los polígonos que se pretenden dibujar.
- Se trata del denominado **solapamiento mutuo** que tiene lugar cuando tres o más polígonos se solapan entre sí de manera circular.
- Si dichos polígonos fuesen clasificados mediante el algoritmo del pintor, no habría un orden correcto en el cual clasificarlos.



La única solución totalmente satisfactoria consistiría en dividir cada uno de los polígonos en dos.

# Supresión de líneas y superficies ocultas

## Algoritmo Z-Buffer

El único sistema realmente exhaustivo para llevar a cabo una clasificación de profundidades implicaría determinar la profundidad de cada punto sobre la superficie de cada polígono en la pantalla y proceder a dibujar sólo los puntos que estén más cercanos con respecto al espectador.

Afortunadamente, sólo es preciso clasificar aquellos puntos que van a ser dibujados, es decir, aquellos que corresponden a los píxeles situados en el plano de proyección.

El algoritmo Z-Buffer determina qué puntos situados sobre determinados polígonos están más cerca del espectador para cada píxel en el plano de proyección.

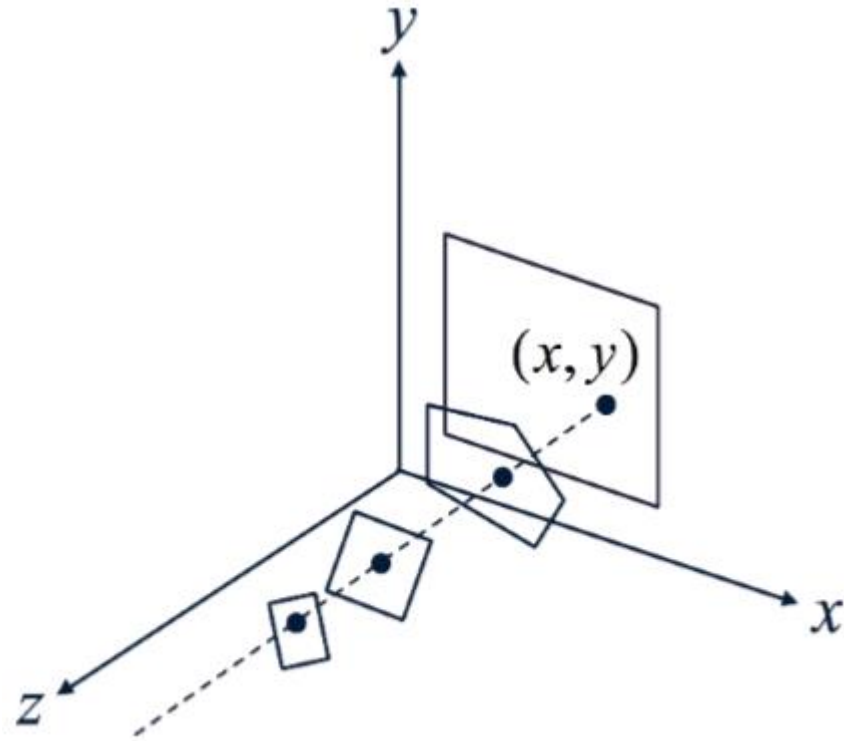
En cada posición del píxel  $x,y$  sobre el plano de proyección, la superficie con la menor coordenada  $z$  en esa posición es visible.

# Supresión de líneas y superficies ocultas

## Algoritmo Z-Buffer

Cada vez que un punto de la superficie de un polígono se dibuja, la coordenada  $z$  del punto es comparada con el valor actual de dicha posición en el buffer.

Si la coordenada  $z$  en el buffer es menor que la del nuevo punto, el nuevo pixel no se dibuja, ya que dicho punto estaría más lejos que el punto anterior y formaría parte de una superficie oculta, en caso contrario el nuevo pixel se dibuja sobre la anterior y la coordenada  $z$  del nuevo pixel se coloca, reemplazando al anterior.



# Modelos básicos de iluminación

Entendemos por modelo de iluminación el cálculo de la intensidad de cada punto de la escena. En el cálculo de la intensidad de un punto intervienen:

- El tipo e intensidad de la fuente de luz
- El material del objeto
- La orientación del objeto con respecto a la luz

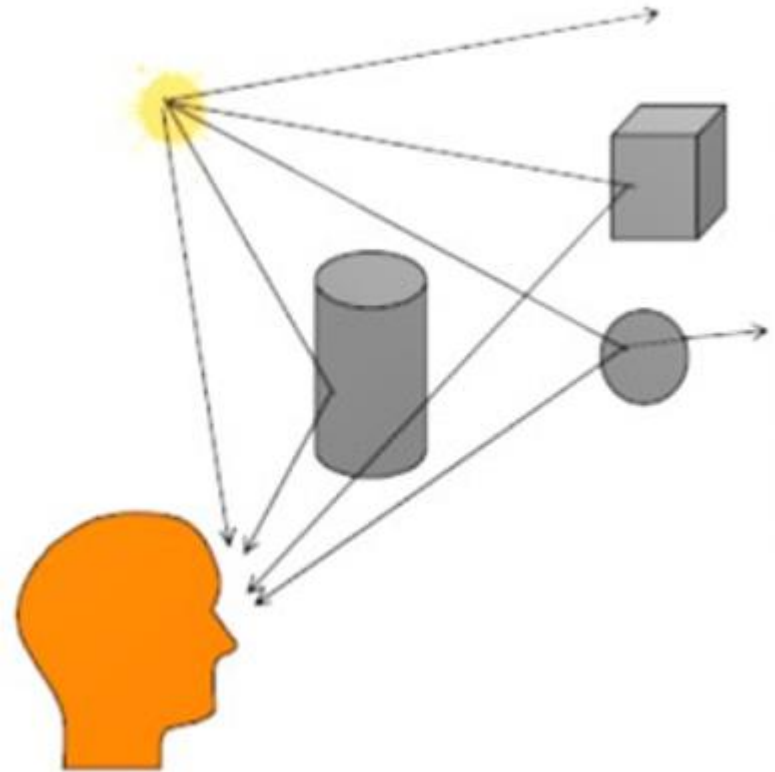
El modelo más utilizado es el modelo de Phong.

# Modelos básicos de iluminación

## Proceso de iluminación

Si un rayo de luz entra al ojo directamente de la fuente, se verá el color de la fuente.

Si un rayo de luz pega en una superficie que es visible al observador, el color visto se basará en la interacción entre la fuente y el material de la superficie: se verá el color de la luz reflejado de la superficie a los ojos.



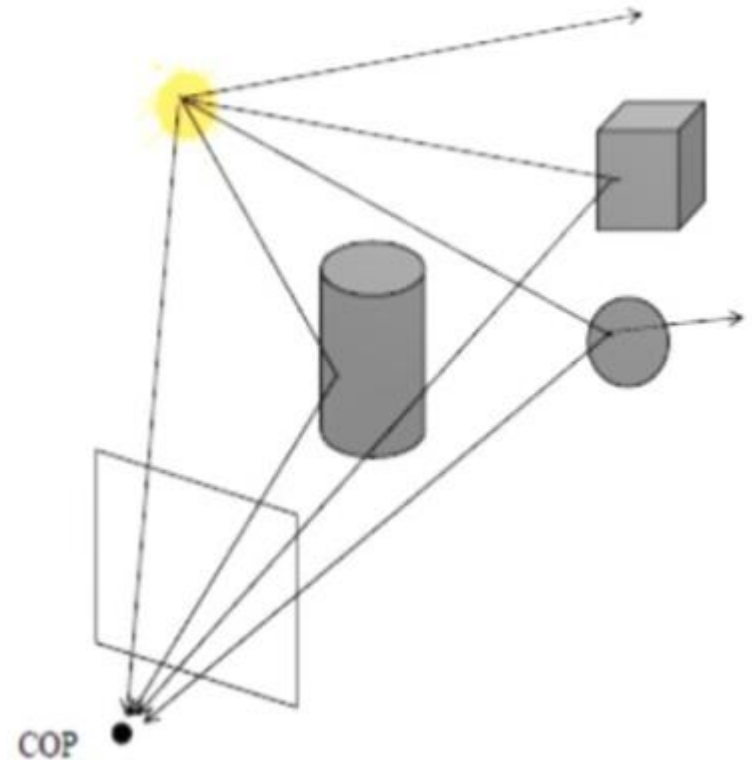
# Modelos básicos de iluminación

## Proceso de iluminación

El color de la fuente de luz y las superficies determina el color de uno o más píxeles en el frame buffer.

Se debe considerar solo aquellos rayos que dejan las fuentes y llegan al ojo del observador, el COP, después de pasar por el rectángulo de recorte.

Cuando la luz da en una superficie, parte se absorbe, y parte se refleja.





# Modelos básicos de iluminación

## Proceso de iluminación

- Si la superficie es opaca, reflexión y absorción significarán toda la luz que dé en la superficie.
- Si la superficie es translúcida, parte de la luz será transmitida a través del material y podrá luego interactuar con otros objetos.

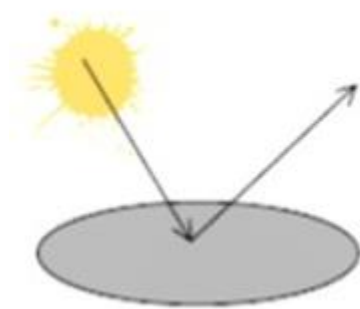
Un objeto iluminado por luz blanca se ve rojo porque absorbe la mayoría de la luz incidente pero refleja luz en el rango rojo de frecuencias.

Un objeto relumbrante se ve así porque su superficie es regular, al contrario de las superficies irregulares.

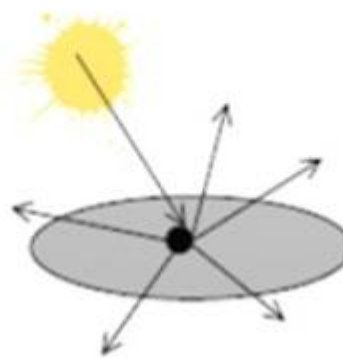
El sombreado de los objetos también depende de la orientación de las superficies, caracterizado por el vector normal a cada punto.

# Modelos básicos de iluminación

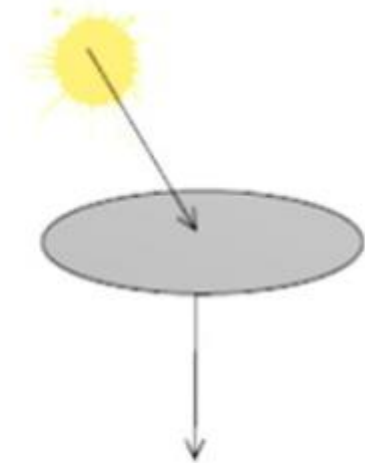
## Interacciones entre luz y materiales



especular



difusa



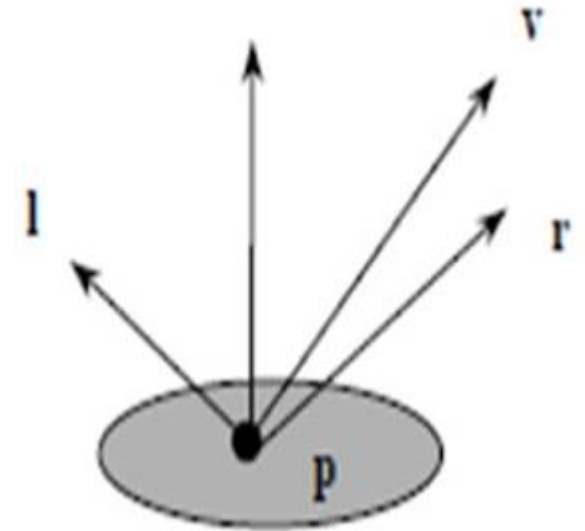
translúcida

opaco

# Modelos básicos de iluminación

## Modelo de iluminación Phong

- El vector  $n$  es la normal en  $p$ .
- El vector  $v$  tiene dirección de  $p$  al observador.
- El vector  $l$  tiene dirección de una línea de  $p$  a un punto arbitrario sobre la superficie para una fuente de luz distribuida, o una fuente de luz de punto.
- El vector  $r$  tiene la dirección de un rayo perfectamente reflejado de  $l$ . La dirección de  $r$  está determinada por  $n$  y  $l$ .



# Modelos básicos de iluminación

## Reflexión de ambiente

La intensidad de la luz ambiente  $L_a$  es la misma sobre cada punto de la superficie. Parte de la luz es absorbida y parte es reflejada. La cantidad reflejada está dada por el coeficiente de reflexión de ambiente  $k_a$ . Como sólo se refleja una fracción positiva de luz, se debe tener

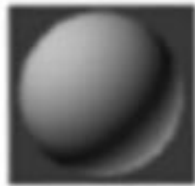
$$0 \leq k_a \leq 1 \quad \text{y por lo tanto} \quad I_a = k_a L_a$$

Aquí,  $L_a$  puede ser cualquiera de las fuentes de luz individuales, o puede el término ambiente global.

Una superficie tiene tres coeficientes ambiente,  $k_{ar}$ ,  $k_{ag}$  y  $k_{ab}$ , que pueden ser distintas. Por ejemplo, una esfera se vería amarilla bajo luz ambiente blanca si su coeficiente ambiente azul es pequeño y sus coeficientes rojo y verde son grandes.

# Modelos básicos de iluminación

## Reflexión de ambiente



Objeto a representar



Sólo con iluminación ambiente  
Constante para los puntos de la esfera

# Modelos básicos de iluminación

## Reflexión difusa

Si se agrega un coeficiente de reflexión  $kd$  que representa la fracción de luz difusa entrante que es reflejada, se tiene el siguiente término de reflexión

$$I_d = kd (\mathbf{l} \cdot \mathbf{n}) L_d \quad 0 \leq kd \leq 1$$

Si se desea incorporar el término de distancia, para considerar la atenuación de la luz según esta viaja una distancia  $1/d^2$  desde la fuente a la superficie, se puede agregar el término cuadrático de atenuación:

$$I_d = \frac{k_d}{a + bd + cd^2} (\mathbf{l} \cdot \mathbf{n}) L_d$$

Una mejor aproximación empírica de atenuación utiliza coeficientes  $a$ ,  $b$  y  $c$ , que pueden manipularse para conseguir diferentes efectos de iluminación.

# Modelos básicos de iluminación

## Reflexión difusa



Objeto a representar



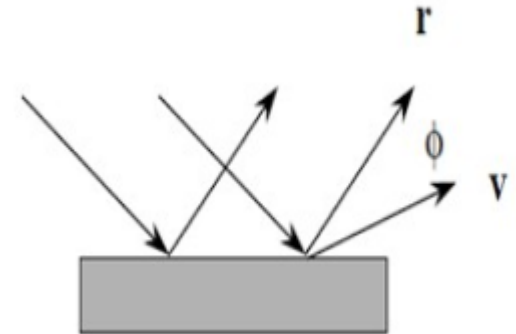
Objeto con luz difusa

Sólo una zona está iluminada

# Modelos básicos de iluminación

## Reflexión especular

Mientras que una superficie difusa es rugosa, una superficie especular es suave. Mientras más lisa se la superficie, más se parece a un espejo.



Según la superficie se hace mas lisa, la luz reflejada se concentra en un rango más pequeño de ángulos.

La cantidad de luz que el observador ve depende del ángulo  $\phi$  entre  $r$ , la dirección de un reflector perfecto, y  $v$ , la dirección del observador.

$$I_s = k_s L_s \cos(\alpha \phi) \quad 0 \leq k_s \leq 1$$

El coeficiente  $k_s$  es la fracción reflejada de la luz especular entrante. El exponente  $\alpha$  es el coeficiente de brillantez.



# Modelos básicos de iluminación

## Reflexión especular

En el límite, según  $\alpha$  tiende a infinito, se obtiene un espejo; valores entre 100 y 500 corresponden a la mayoría de las superficies metálicas, y valores menores ( $<100$ ) corresponden a materiales que muestran brillantez gruesa.

La ventaja computacional del modelo de Phong es que, si se normaliza  $\mathbf{r}$  y  $\mathbf{v}$  a valores unitarios, se puede usar el producto punto, y el termino especular se vuelve.

$$I_s = k_s L_s (\mathbf{r} \cdot \mathbf{v})^\alpha$$

Se puede agregar el termino de distancia, como se hizo con las reflexiones difusas.

# Modelos básicos de iluminación

Finalmente, se refiere al modelo Phong, incluyendo el término de distancia, a la siguiente ecuación:

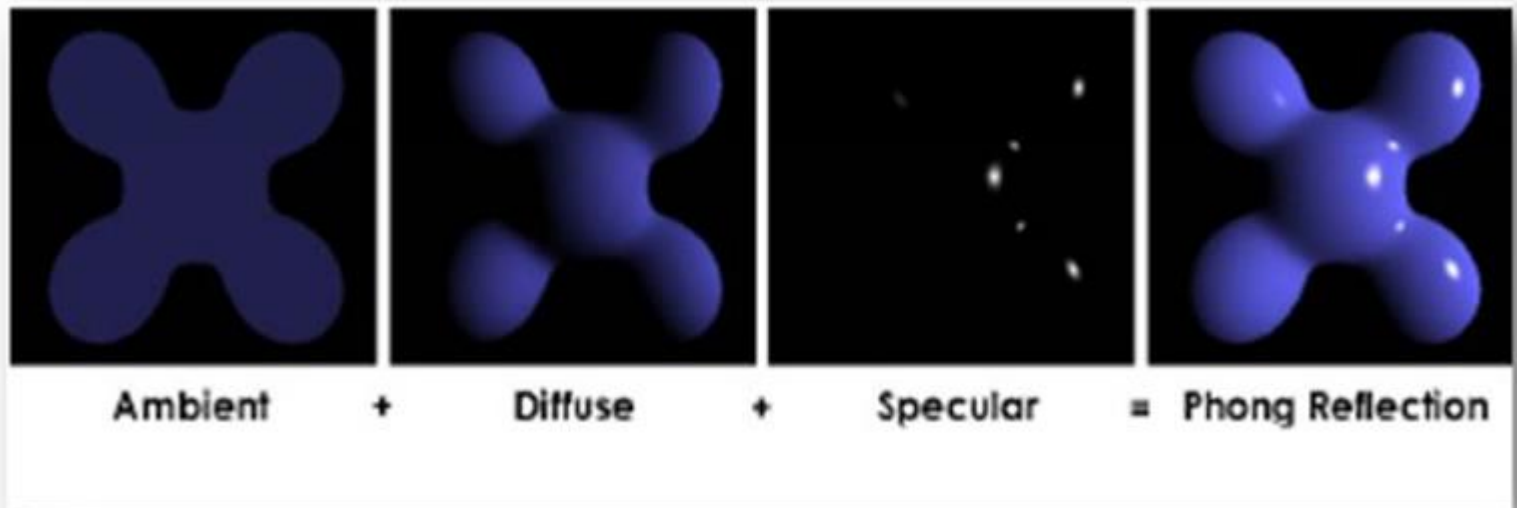
$$I = \frac{1}{a + bd + cd^2} (k_d L_d (\mathbf{l} \cdot \mathbf{n}) + k_s L_s (\mathbf{r} \cdot \mathbf{v})^\alpha) + k_a L_a$$

Esta fórmula se computa para cada fuente de luz y para cada factor de color.



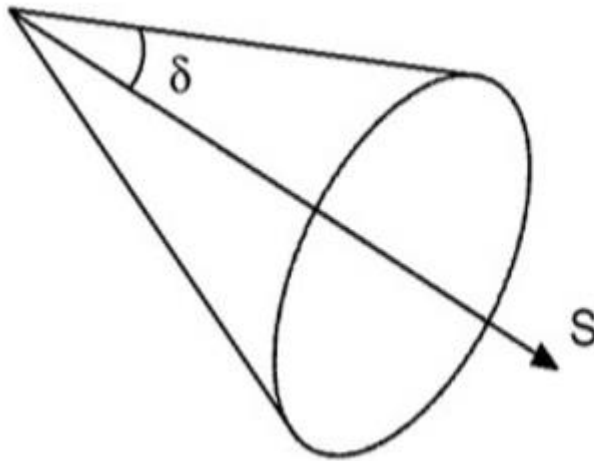
# Modelos básicos de iluminación

## Ejemplo de iluminación Phong



# Foco de luz

- Requiere una dirección  $S$  y un ángulo  $\delta$  para la forma del cono.
- Atenuación: se calcula mediante el coseno del ángulo entre los vectores  $L$  y  $S$  elevado a un exponente. Cuanto mayor sea este exponente, mayor la concentración de luz alrededor del eje del cono.



# Sombreado

El sombreado o visualización de polígonos es la asignación de intensidades a cada punto de los polígonos que forman un objeto.

Visualización de objetos:

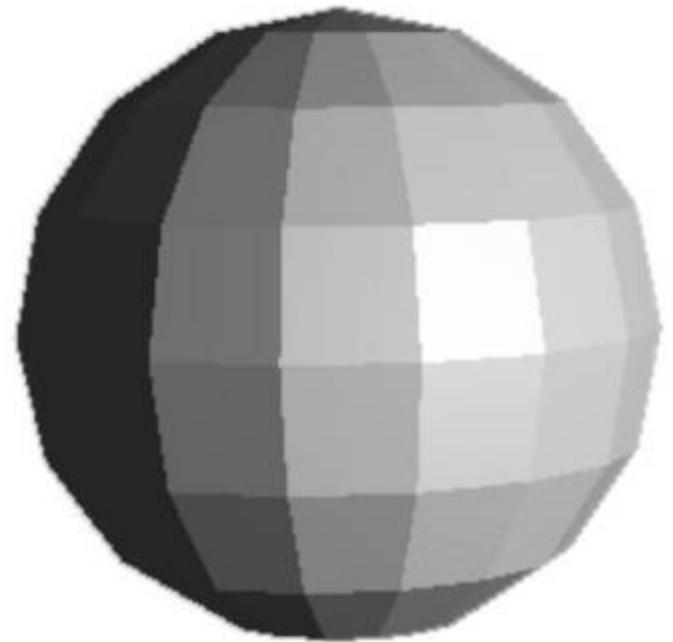
- Sombreado plano o de intensidad constante
- Sombreado Gouraud
- Sombreado de Phong

# Sombreado plano

Se calcula un único valor de intensidad para cada polígono mediante el modelo de iluminación de Phong, y se asigna a todos sus puntos.

Es exacto cuando:

- El objeto es un poliedro
- fuentes de luz se encuentran alejadas del objeto (en esos casos  $N \cdot L$  y la función de atenuación se pueden considerar constantes)
- El observador está lejos del objeto ( $V \cdot R$  se puede considerar constante)

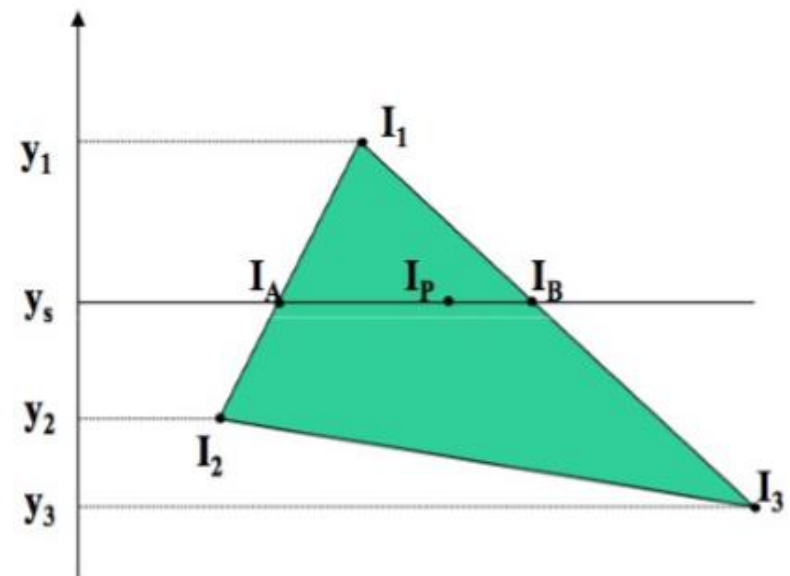


# Sombreado Gouraud

Es un método incremental que realiza una interpolación de intensidades. En cada vértice del polígono se calcula la intensidad. La intensidad de los puntos intermedios se calcula por interpolación bilineal.

Calcular la intensidad de cada vértice, y finalmente ir interpolando por el borde y por la scan-line:

- A partir de  $I_1$  e  $I_2$  se calcula  $I_A$
- A partir de  $I_1$  e  $I_3$  se calcula  $I_B$
- A partir de  $I_A$  e  $I_B$  se calcula  $I_P$



# Sombreado Phong

Es un método incremental que realiza una interpolación de normales (en vez de interpolación de intensidades).

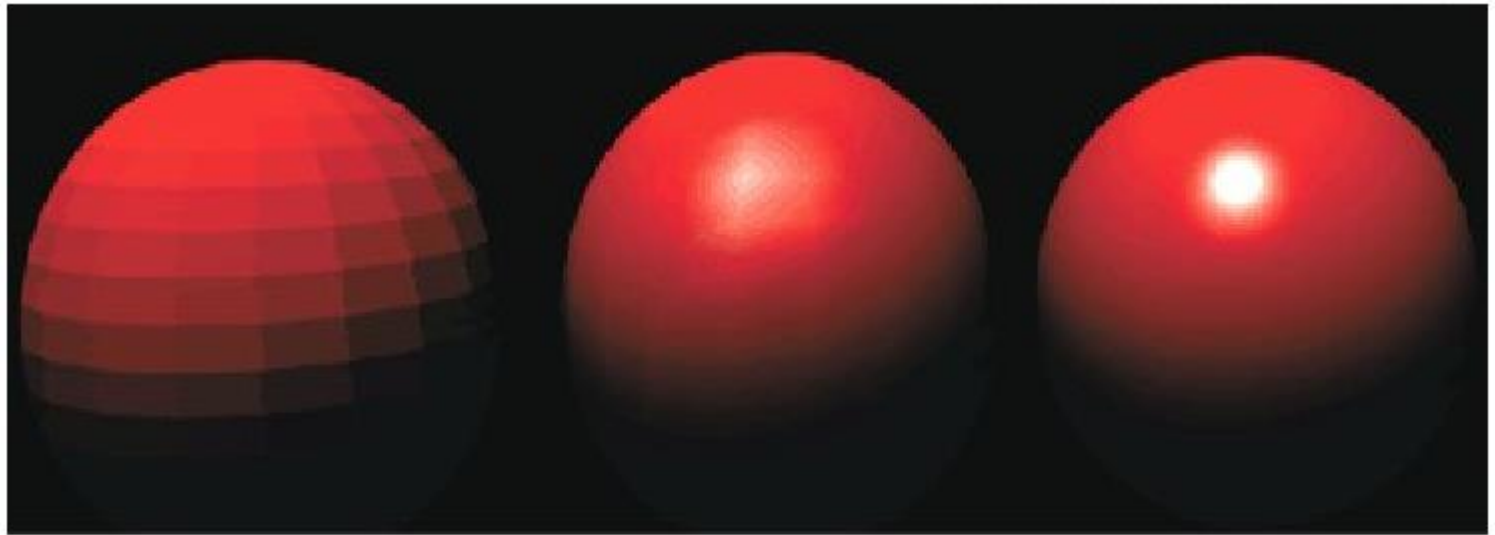
En cada vértice del polígono se calcula la normal como media de las normales de los polígonos adyacentes. La normal de los puntos intermedios se calcula por interpolación lineal.

En cada punto se aplica el modelo de iluminación de Phong.

- Calcular normal en los vértices, como en el sombreado de Gouraud, para evitar visualizar las aristas, la normal se calcula como la media de las normales de los polígonos adyacentes.
- Calcular la normal de los puntos interiores por interpolación lineal de los vértices.
- Calcular la intensidad en cada punto mediante el modelo de iluminación Phong.



# Sombreado



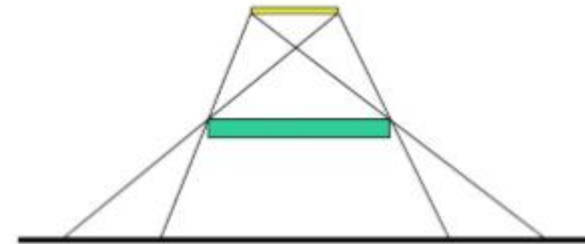
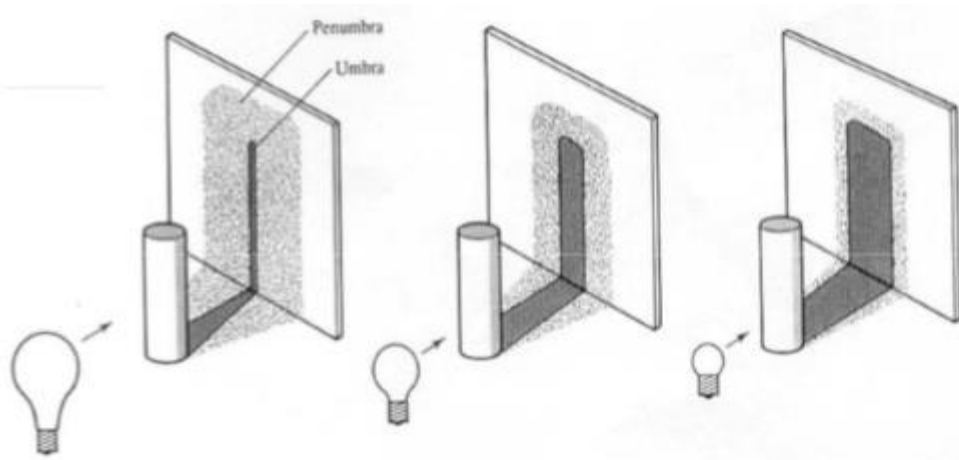
Flat

Gouraud

Phong

# Cálculo de sombras

Hay que determinar qué superficies son vistas desde la(s) fuente(s) de luz, si no se ven, hay sombras, además también se tienen que considerar las penumbras.



# Cálculo de sombras

## Algoritmo de doble pasada

- Determina las sombras antes que las superficies visibles
- Ejecuta el mismo algoritmo:
  1. Desde la fuente de luz (sólo una vez)
  2. Luego desde el observador
- El resultado es una lista de polígonos
- Se transforman al sistema de coordenadas global y se mezclan con la escena original para obtener los polígonos de detalle
- Finalmente se aplica un algoritmo de visibilidad y otro de iluminación
- Si hay múltiples fuentes de luz hay que hacer una pasada más por cada fuente

# Cálculo de sombras

## Algoritmo de sombras del Z-Buffer

- Se ejecuta dos veces el z-buffer:
  - ✓ Una vez desde la fuente de luz
  - ✓ Otra vez desde el observador
- Para crear la imagen final se usa un z-buffer modificado
- Para cada pixel visible, transformo sus coordenadas  $(x,y,z)$  con referencia al ojo en  $(x',y',z')$  con referencia a la luz
- Comparo  $z'$  con la  $z$  correspondiente del buffer  $z$  de la luz
- Si  $z' > Z_L$  el punto está en sombra
- En realidad, el z-buffer

