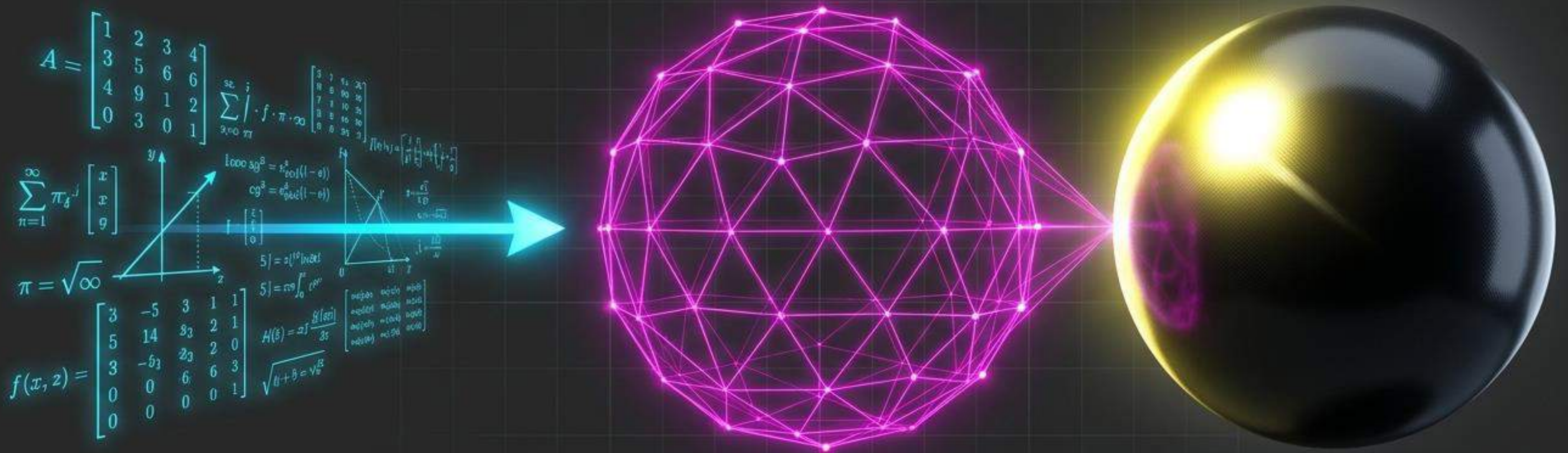


# De Vectores a Mundos Virtuales

La matemática subyacente de la luz, la forma y el renderizado 3D.

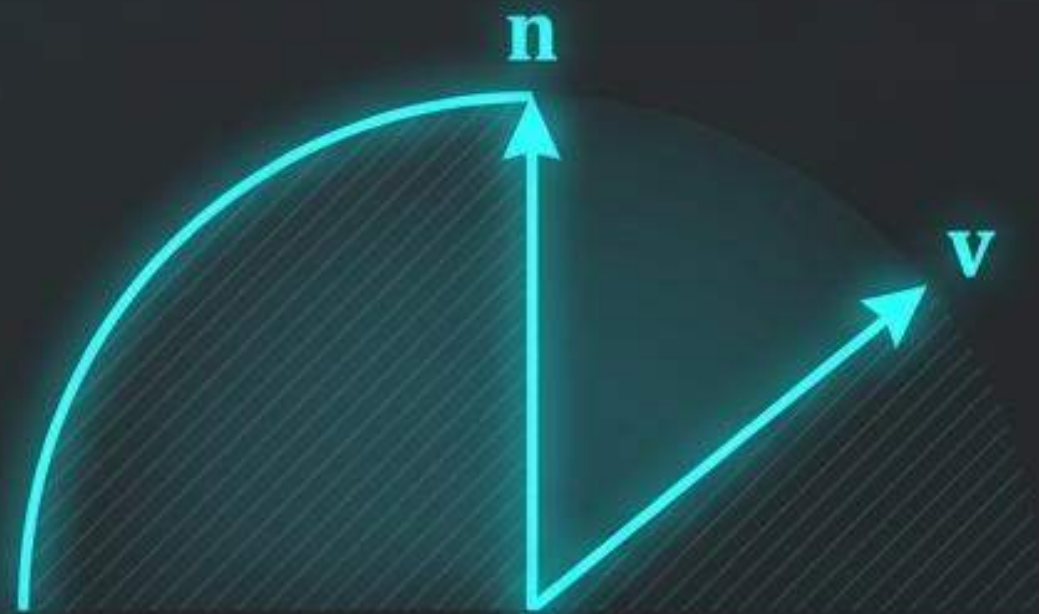




Cada bloque de construcción matemática es un prerequisite estricto para el siguiente. Este es el viaje de un vértice hasta convertirse en luz.

# Los Bloques de Construcción: Vectores y Orientación

## El Producto Punto como Detector



$$\mathbf{u} \cdot \mathbf{v} = |\mathbf{u}| |\mathbf{v}| \cos \gamma$$

En el espacio 3D, el producto punto no es solo una multiplicación escalar; es un 'Detector de Orientación'. Determina instantáneamente si dos entidades se miran de frente (positivo), son perpendiculares (cero) o se dan la espalda (negativo).

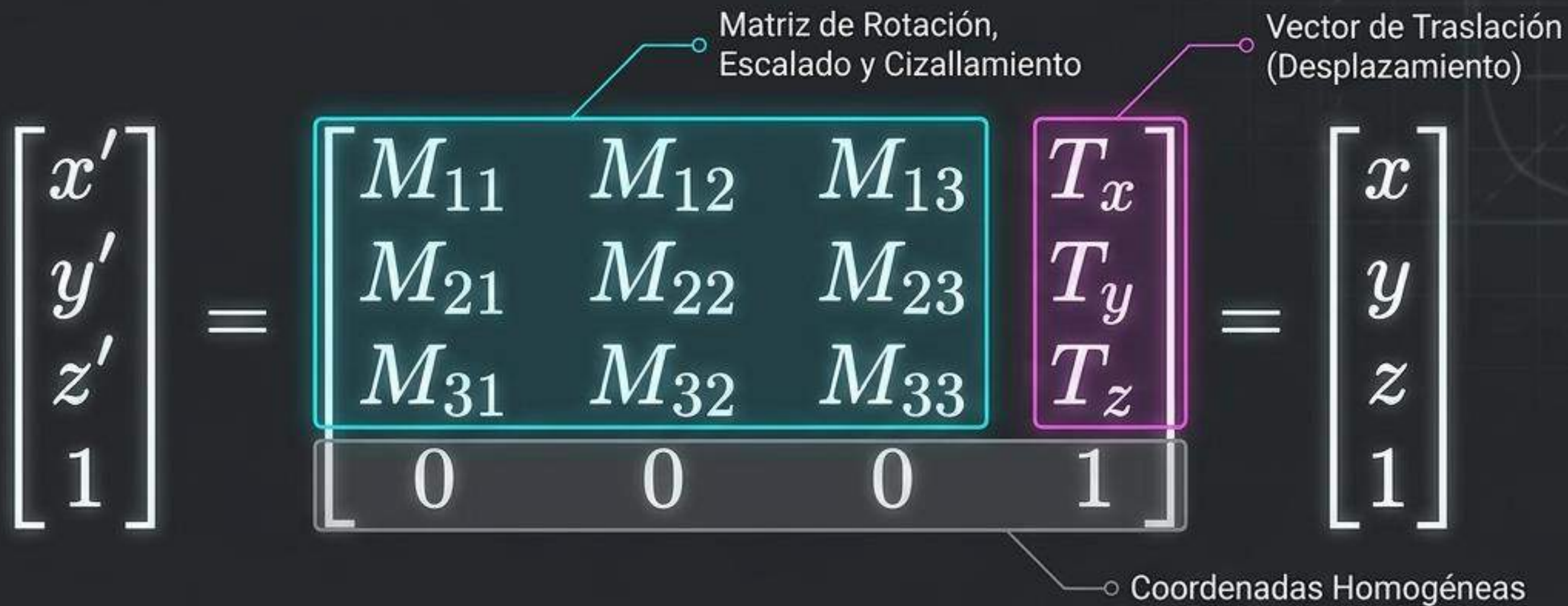
## Proyección y Distancia



$$d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$$

La proyección vectorial permite calcular la distancia perpendicular exacta entre objetos, fundamental para colisiones y oclusión.

# La Arquitectura del Movimiento: Transformaciones Afines

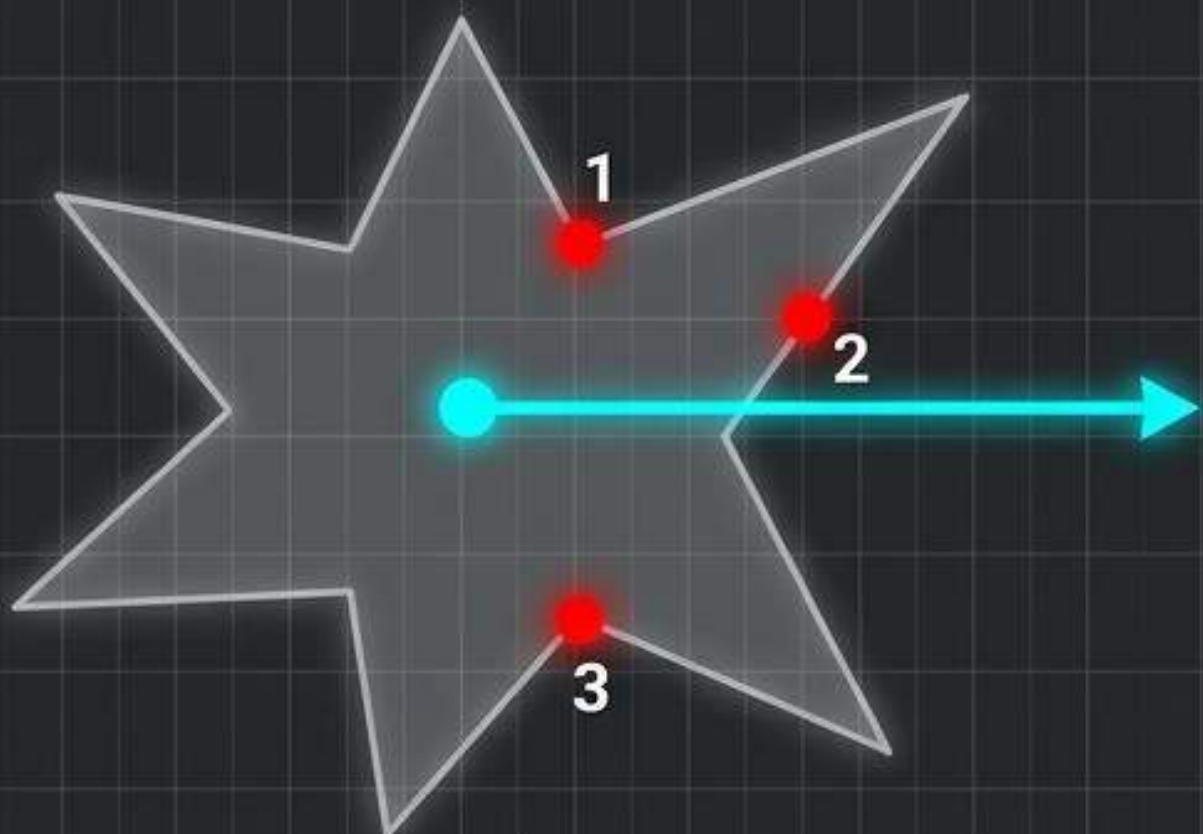


The diagram illustrates the structure of an affine transformation matrix in homogeneous coordinates. It shows the equation: 
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$
 Annotations include: 'Matriz de Rotación, Escalado y Cizallamiento' pointing to the top 3x3 sub-matrix; 'Vector de Traslación (Desplazamiento)' pointing to the top 3x1 sub-vector; and 'Coordenadas Homogéneas' pointing to the bottom row of the matrix.

Las transformaciones afines preservan la colinealidad (las líneas rectas siguen siendo rectas) y el paralelismo, pero permiten traslaciones espaciales.

**El Truco Homogéneo:** Multiplicar una matriz 3x3 no permite trasladar el origen. Al añadir una dimensión extra, el desplazamiento se integra en una única operación matricial, optimizando el procesamiento en la GPU.

## Algoritmos Ray-Casting (Cruce de Bordes)

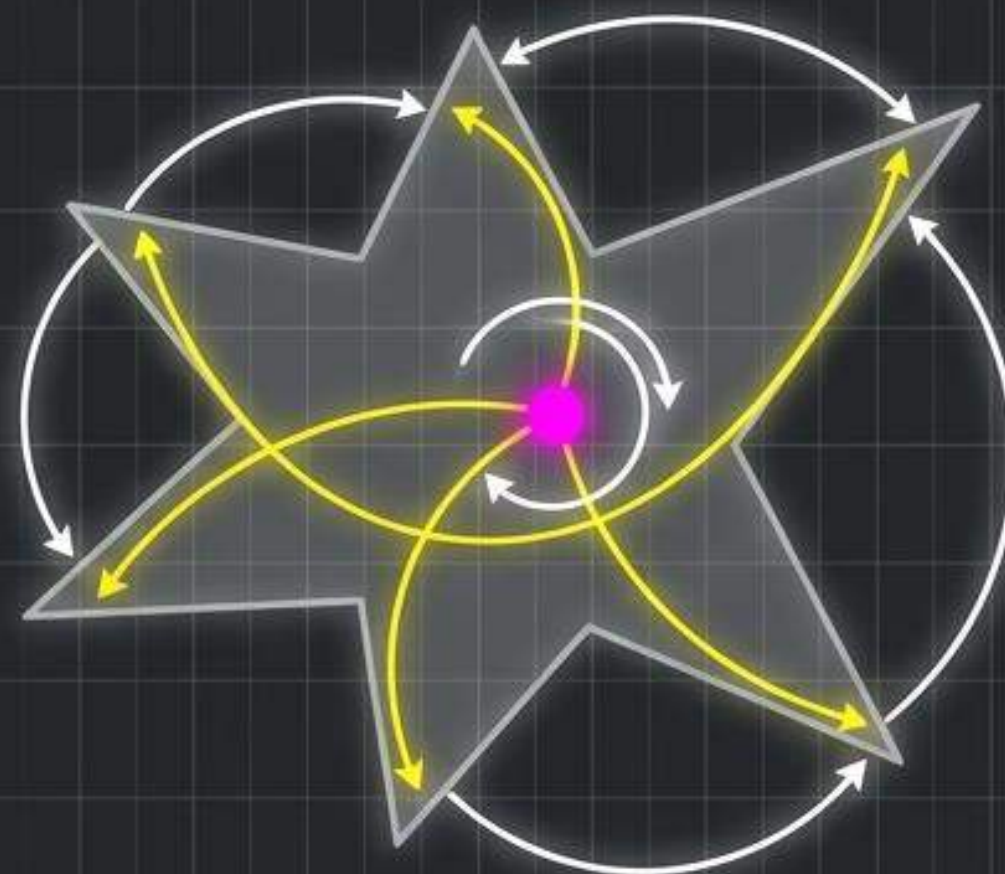


**Mecanismo:** Lanza un rayo horizontal desde el punto al infinito. Cuenta las intersecciones con los bordes.

**Lógica:** Impar = Dentro. Par = Fuera.

**Vulnerabilidad:** Vértices y líneas tangentes requieren lógica adicional compleja.

## Winding Number (Número de Vueltas)

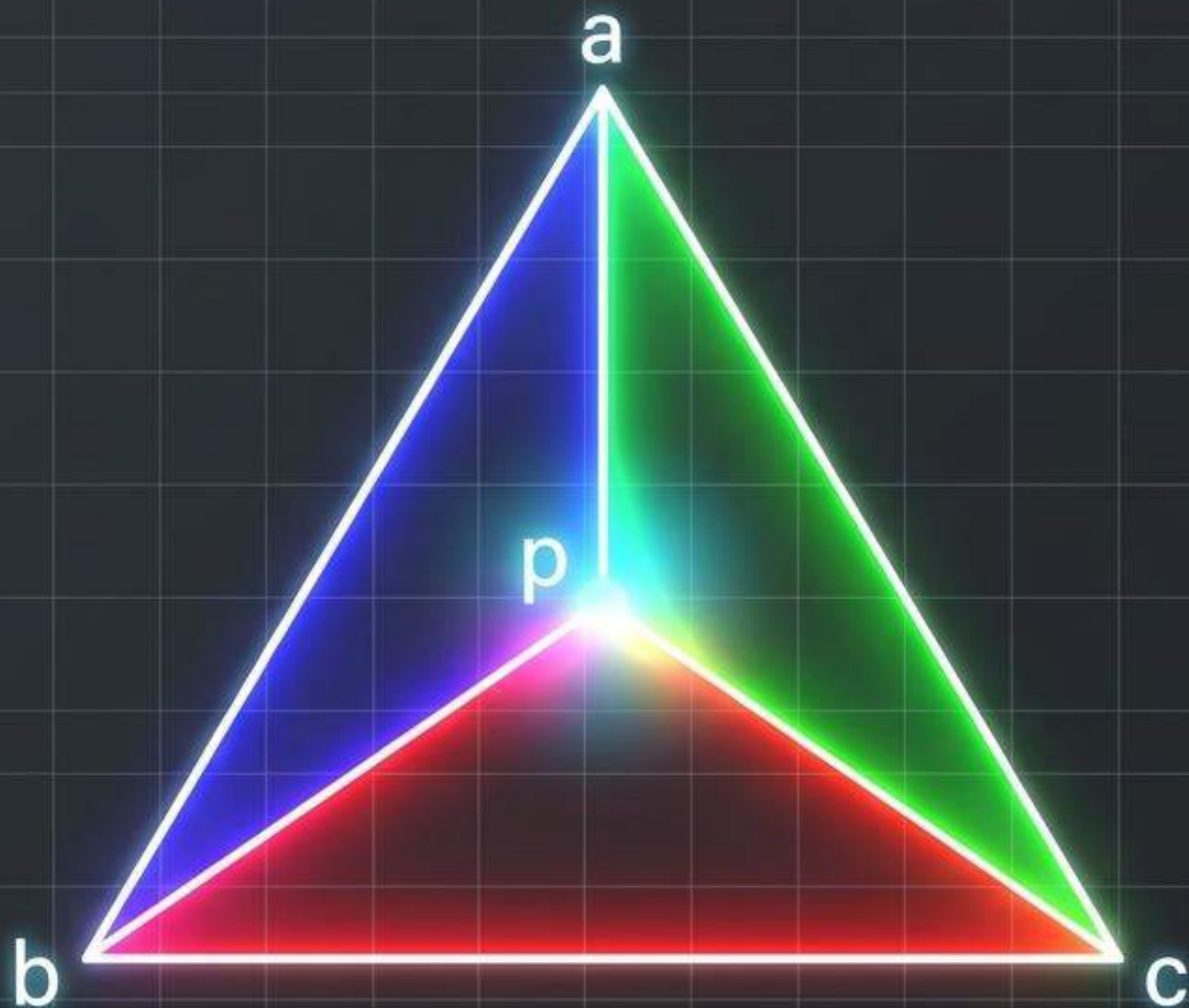


**Mecanismo:** Suma los ángulos subtendidos por cada lado del polígono desde el punto.

**Lógica:** Total  $\neq 0$  = Dentro. Total = 0 = Fuera.

**Robustez:** Matemáticamente robusto, incluso con polígonos complejos o con agujeros.

# El Triángulo Perfecto y las Coordenadas Baricéntricas



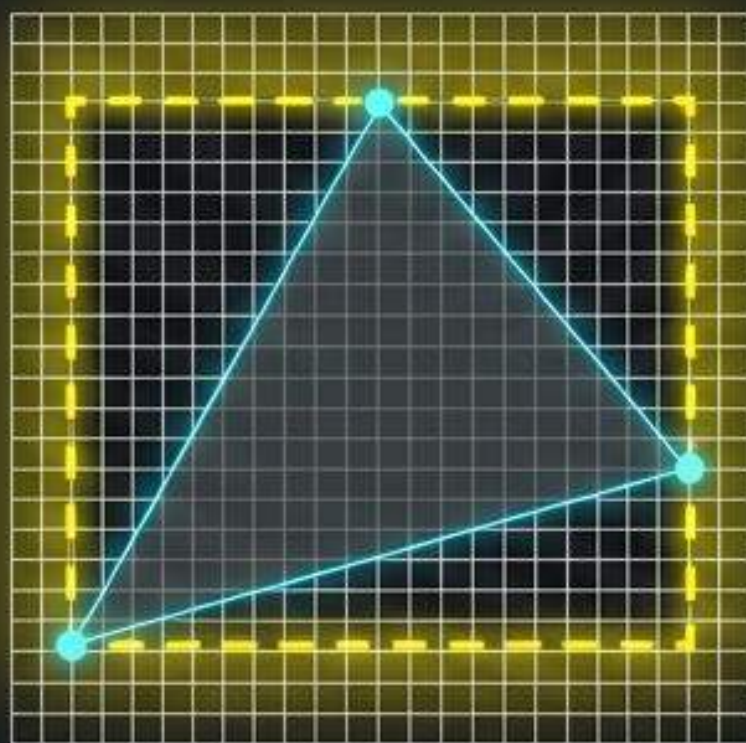
¿Cómo pintamos el interior de un polígono? Usamos un sistema de "Pesas en una Balanza" matemático:  $\alpha + \beta + \gamma = 1$ .

Cada coordenada  $(\alpha, \beta, \gamma)$  es proporcional al área del subtriángulo opuesto. Si  $\alpha = 1$ , el punto está exactamente sobre un vértice. Si todas están entre 0 y 1, el punto está estrictamente dentro del triángulo.

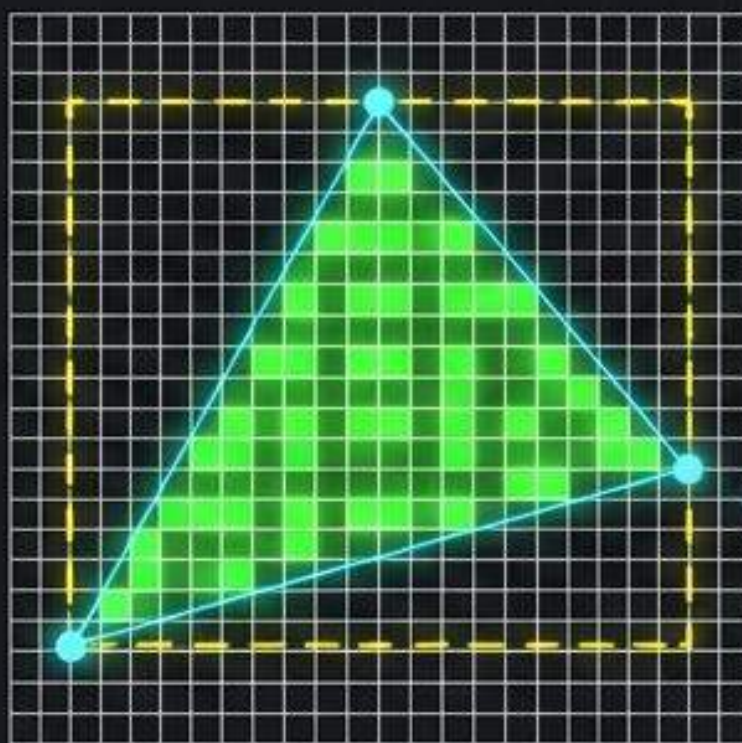
**Poder de Interpolación:** Este sistema no solo ubica posiciones; interpola cualquier atributo en la superficie: colores, coordenadas de textura y normales de superficie.

# De la Geometría a la Cuadrícula: Rasterización y Visibilidad

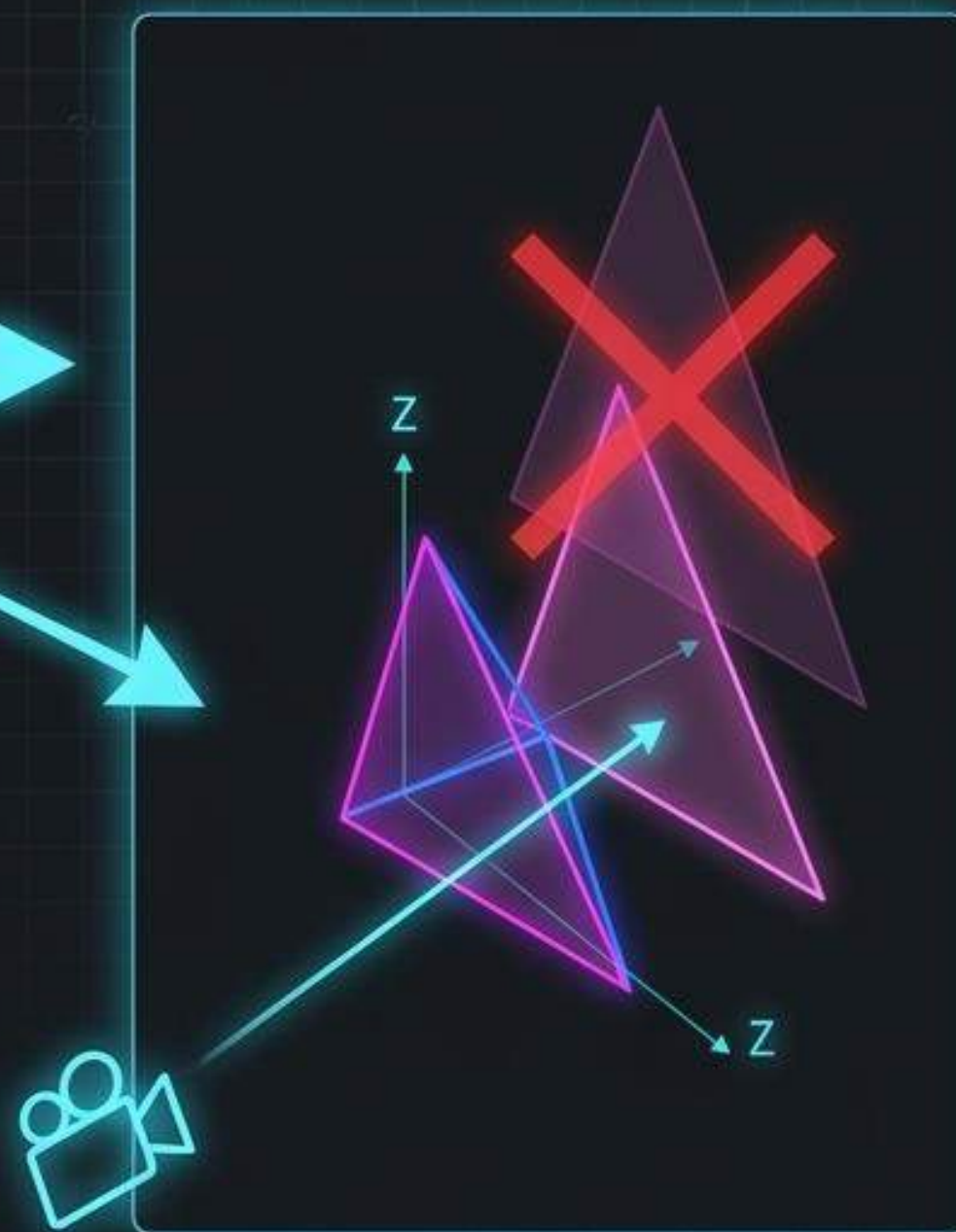
## Bounding Box



## Prueba Baricéntrica



## Z-Buffer y Culling



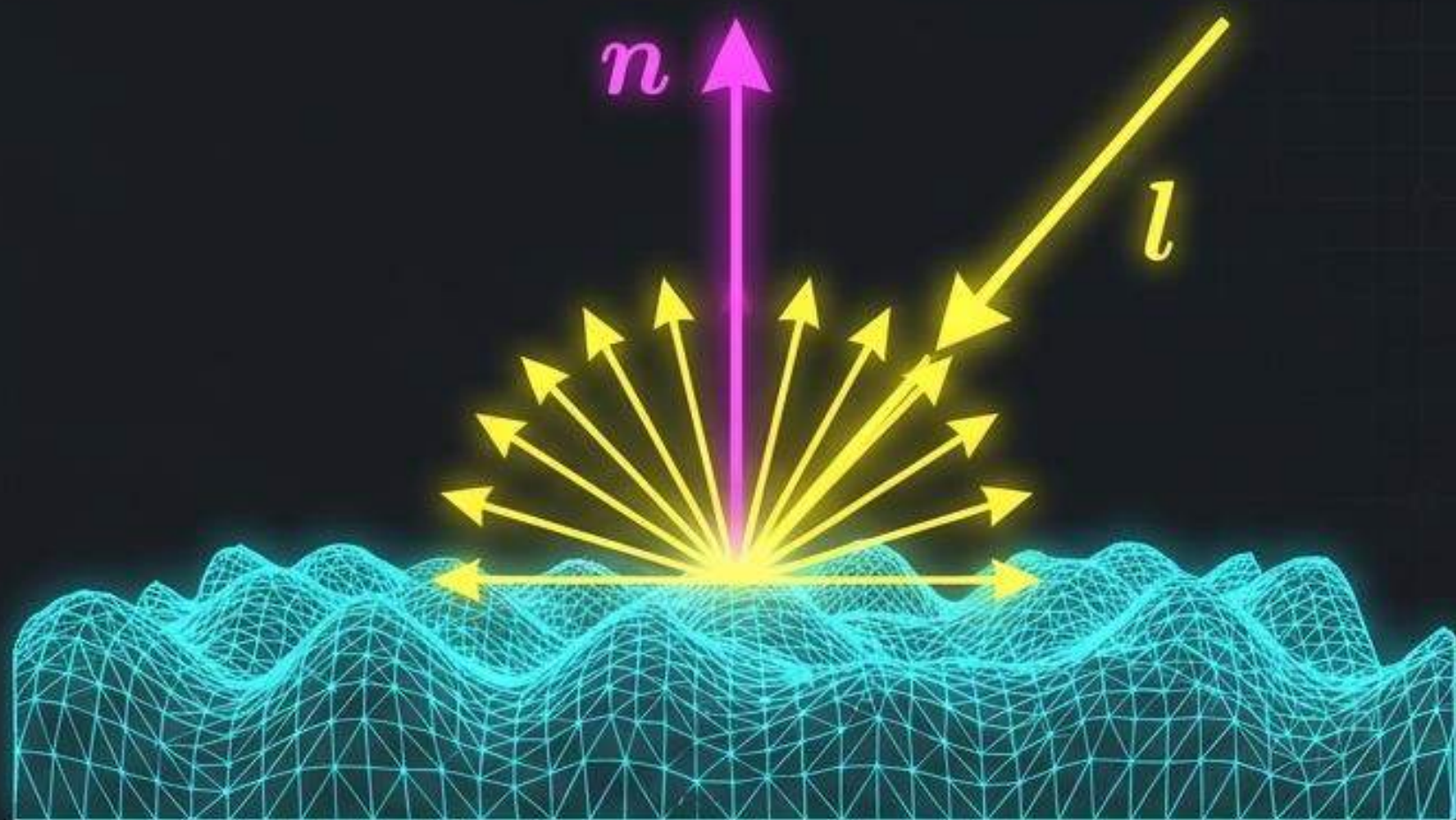
### Z-Buffer (Profundidad):

Mantiene un registro de la distancia relativa al píxel más cercano. Si el nuevo punto z es mayor, se descarta.

### Backface Culling:

Aplicando la Regla de la Mano Derecha y el producto cruz. Si el componente z es negativo respecto a la cámara, el polígono da la espalda y se ignora por completo.

# Modelando la Luz: La Ley del Coseno de Lambert



$$c = c_d c_l \max(0, \mathbf{n} \cdot \mathbf{l})$$

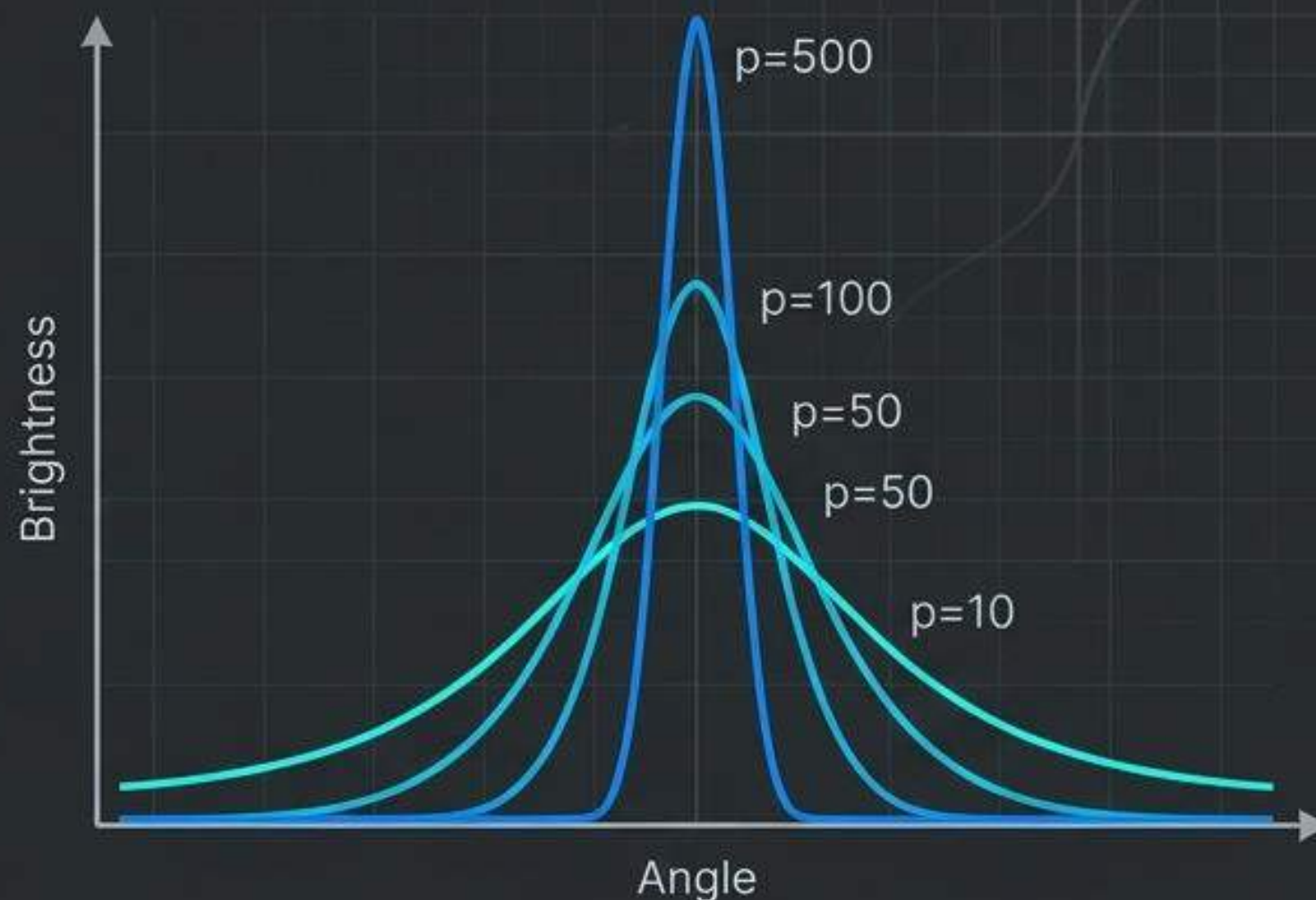
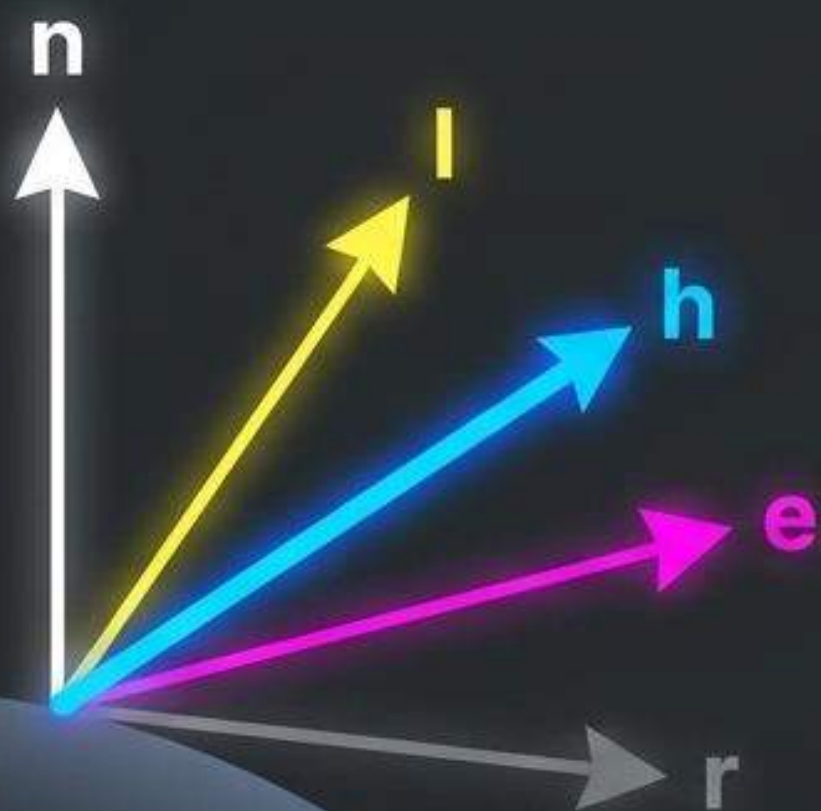
Diagrama de anotaciones para la ecuación:

- Vector normal de la superficie (apunta a  $\mathbf{n}$ )
- Vector direccional de la luz (apunta a  $\mathbf{l}$ )
- Color de superficie y luz (apunta a  $c_d c_l$ )
- Vector direccional de luz (apunta a  $\mathbf{n} \cdot \mathbf{l}$ )

Una superficie 'Lambertiana' ideal (como el papel o la piedra) refleja la luz uniformemente en todas las direcciones. La intensidad luminosa es dictada matemáticamente por el producto punto ( $\mathbf{n} \cdot \mathbf{l}$ ).

El escudo contra valores negativos: Usamos la función  $\max(0, \mathbf{n} \cdot \mathbf{l})$  para asegurar que las superficies que 'dan la espalda' a la luz no generen color negativo, resultando en sombra negra.

# El Brillo Matemático: El Modelo de Phong



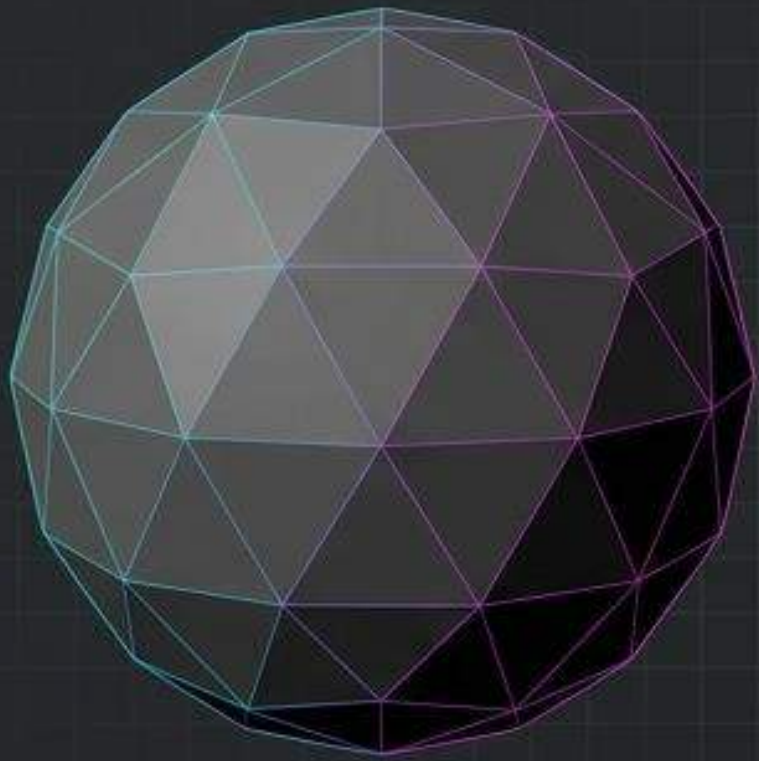
Para hiper-realismo (metal pulido, plásticos), necesitamos reflejos especulares que imitan la luz reflejándose directamente en la cámara.

En lugar de calcular costosas reflexiones absolutas, el modelo computa el Vector Medio ( $\mathbf{h}$ ). Cuando la cámara se alinea con la reflexión,  $\mathbf{h}$  se alinea exactamente con la normal  $\mathbf{n}$ .

El tamaño del brillo se controla elevando el producto punto a un exponente  $p$ :  $c_{\text{specular}} = \max(0, \mathbf{n} \cdot \mathbf{h})^p$ . Mayor  $p$  = brillo más agudo y pequeño.

# Evolución del Renderizado: La Matriz de Sombreado

## Flat Shading

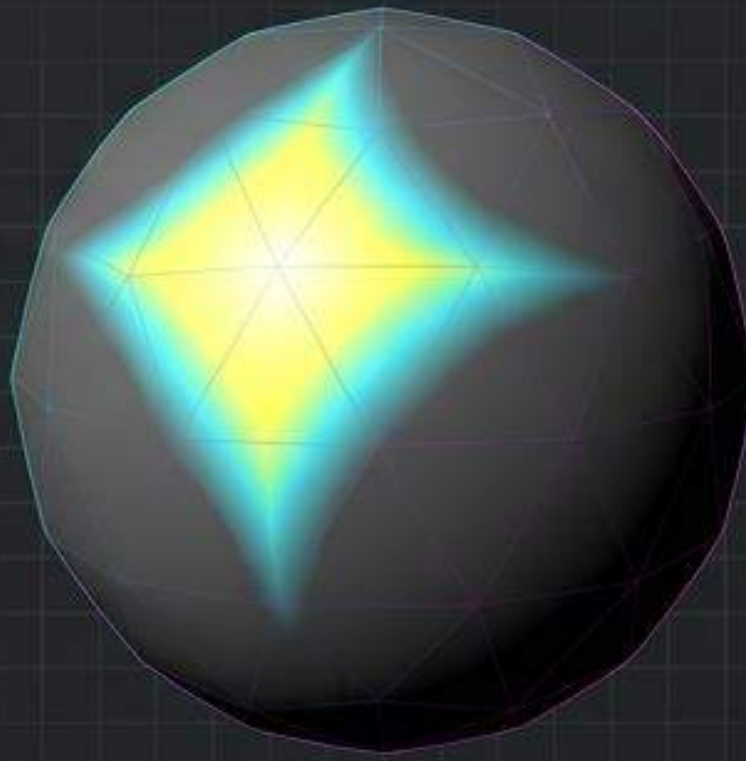


**Cálculo:** Una vez por Cara (Polígono).

**Visual:** Baja resolución poligonal. Bordes duros.

**Atributo:** Rendimiento máximo. Define el estilo estético "low-poly".

## Gouraud Shading



**Cálculo:** Por Vértice.

**Visual:** Superficie suave, pero con brillos especulares que se rompen o deforman cerca del centro de grandes polígonos.

**Atributo:** Equilibrio de rendimiento. La luz se interpola usando coordenadas baricéntricas.

## Phong Shading (Per-Pixel)

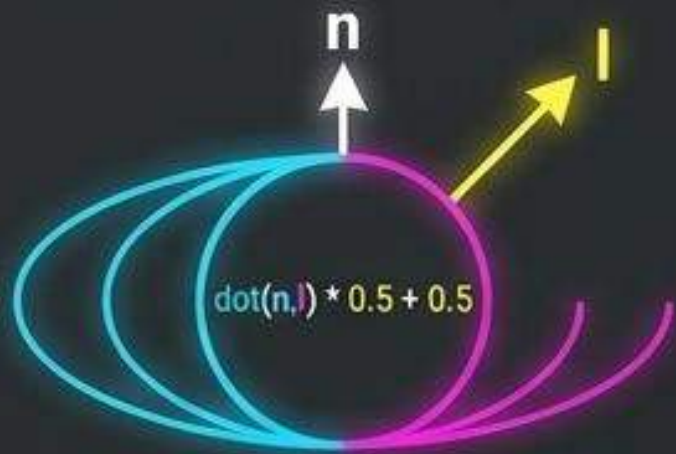


**Cálculo:** Por Píxel.

**Visual:** Gradientes immaculados, punto de luz especular perfecto.

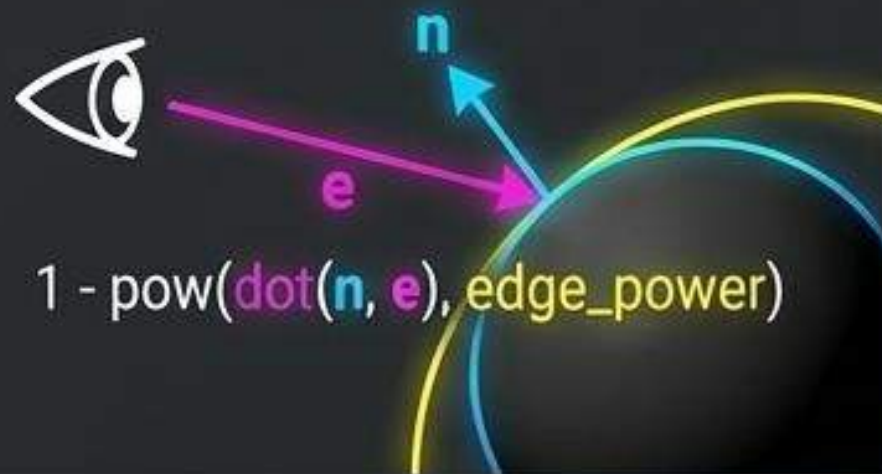
**Atributo:** Computacionalmente pesado. La normal del vértice se interpola por todo el triángulo. El estándar actual.

# El Kit del Artista Técnico: Alterando la Realidad



Half-Lambert (El Truco de Valve)

$$\text{dot}(\mathbf{n}, \mathbf{l}) * 0.5 + 0.5$$



Fresnel / Edge Glow

$$1 - \text{pow}(\text{dot}(\mathbf{n}, \mathbf{e}), \text{edge\_power})$$

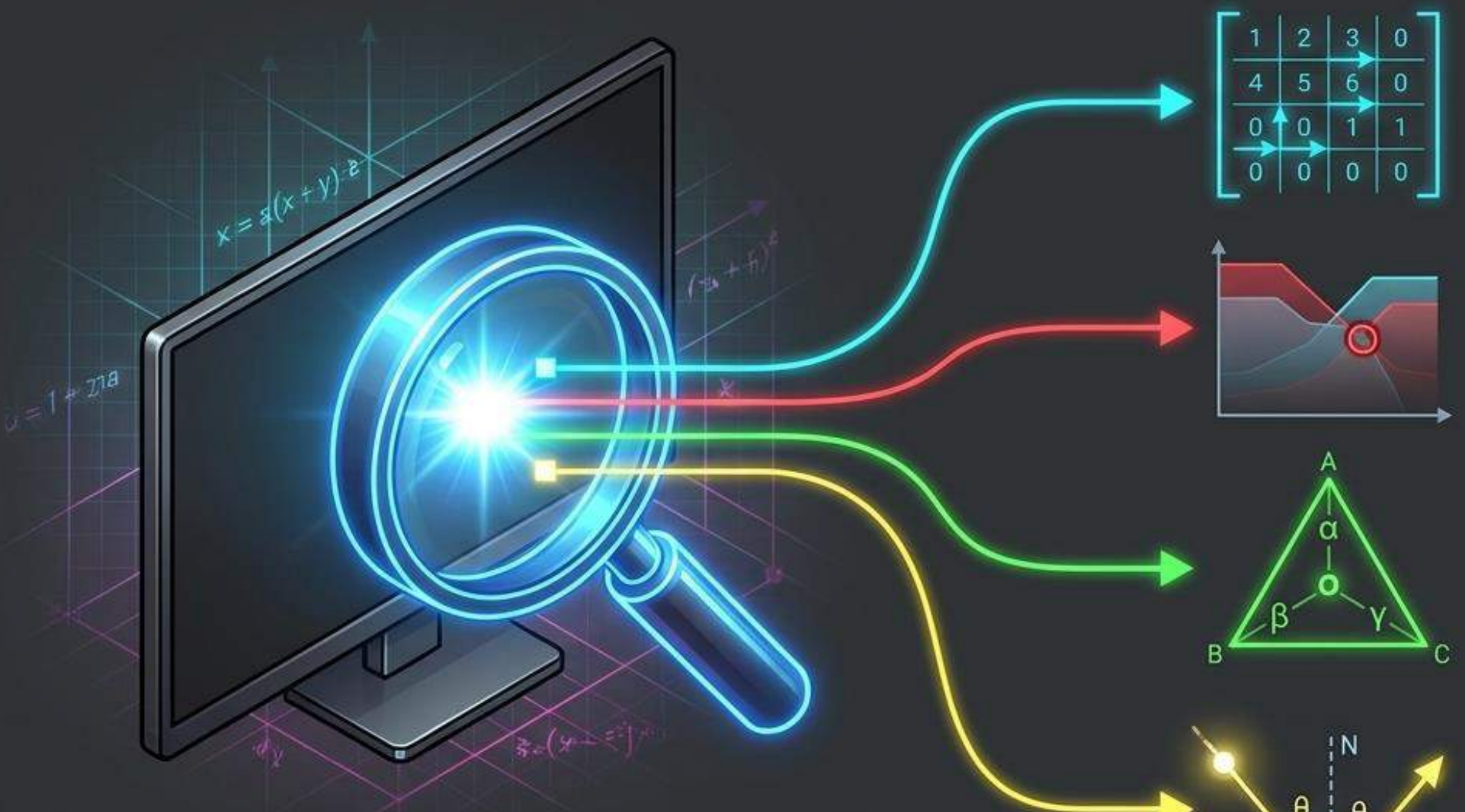


Distance Clipping

$$\text{abs}(\text{distance})$$



# Síntesis: La Anatomía de un Píxel Iluminado



**Espacio (Transformación Afín):**  
Una matriz lo colocó en el mundo.

**Visibilidad (Z-Buffer):**  
Ganó su derecho a existir frente a píxeles ocluidos.

**Geometría (Baricéntrico):**  
Heredó porcentajes genéticos de los tres vértices que lo encierran.

**Óptica (Phong/Lambert):**  
El producto punto con la luz determinó su intensidad de sombra y destello.

Detrás de los 8,294,400 píxeles de una pantalla 4K a 60 fps, no hay magia; hay una orquesta determinista. Cada destello, cada sombra suave y cada rebote metálico es la acumulación secuencial del álgebra lineal convirtiéndose en electromagnetismo simulado.

# El Lienzo de la Lógica Pura

Los mundos virtuales no están hechos de polígonos; están tallados a partir de distancias vectoriales y matrices de identidad.

Comprender la matemática subyacente otorga control total. Dejamos de ser usuarios de herramientas gráficas para convertirnos en los arquitectos de sus leyes físicas fundamentales.

$$A = \begin{bmatrix} 1 & a_{11} & a_{12} & 0 \\ 2 & a_{21} & a_{22} & b \\ 0 & 1 & a_{31} & a_2 \end{bmatrix}$$

$$\int_0^a \cdot (\sqrt{y} \cdot x) dy = \frac{2}{3} a^3$$

$$R = \frac{1}{2\pi a b}$$

$$B = \begin{bmatrix} 3 & 2 & 1 \\ 1 & 8 & 0 \end{bmatrix}$$

$$I = \int_0^a f(x) dx$$

$$g(a, b, x = kg)$$

$$f'(x) = (x^2 - 1/2x)$$

$$f(x) = x^3$$

$$f = a \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & a \end{bmatrix}$$

$$f(x) = (x^2 - 1/2x)$$

Fin.