



Render

De los Números a la Realidad Virtual

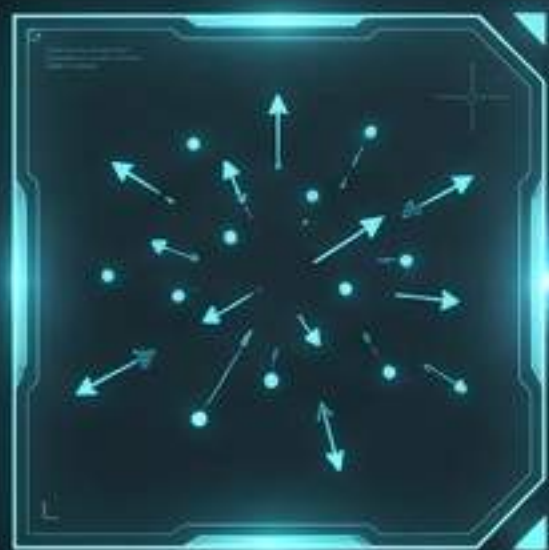
El ADN matemático detrás de los gráficos por computadora.



Una exploración de la geometría computacional y los algoritmos de renderizado.



El Viaje del Vértice: El "Pipeline" de Renderizado



(S: 1.3, Y: -4.5, Z: 3.1)

V18, 1, 0)
V18, 1, 0)
V19, 1, 0)

1. El Espacio Vectorial:
Puntos y direcciones puras en el vacío.

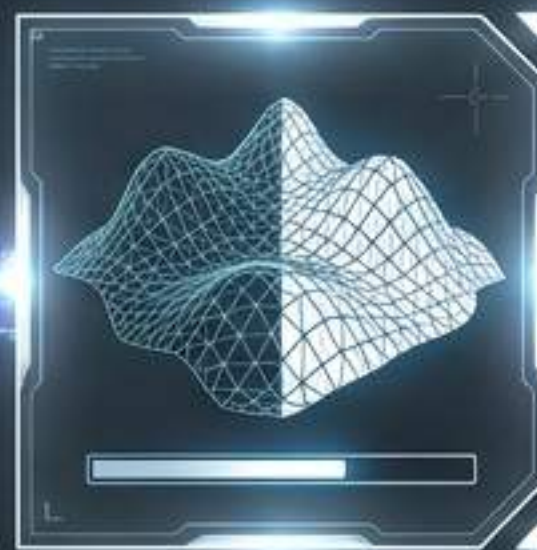


```
1.0  0.0  0.0  0.0  
0.0  0.10 0.0  0.0  
0.0  cos(θ) -sin(θ) 1.0  
1.0  0.0  0.0  1.0
```

World Space → View Space

```
1.0  0.0  0.0  0.0  
0.0  cos(θ) -sin(θ) 1.0  
0.0  0.0  0.0  1.0
```

2. Transformación:
Moviendo y escalando el universo.



Fragment Shader:
Triangles Placed: 5,340
Triangles Set:
Screen Coords: (900, 540)

Diffuse + Specular + Ambient
Light Intensity: 1.5
Bounce 1, Bounce 2

3. Rasterización:
Rellenando la malla y definiendo superficies.



4. Iluminación:
Simulando el comportamiento de los fotones.

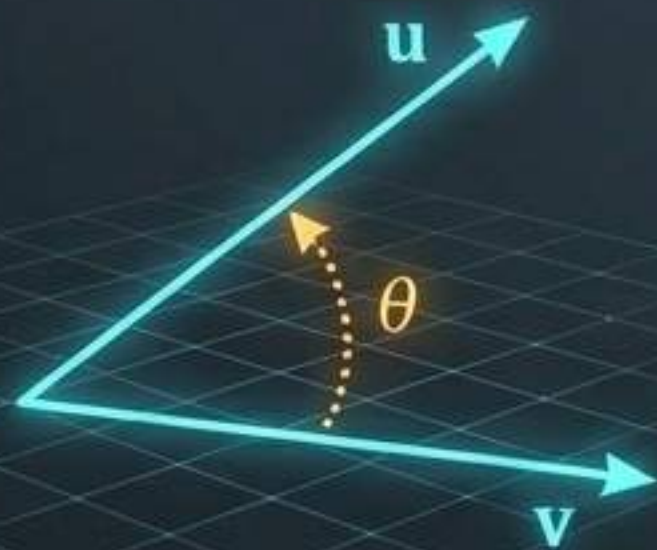
Dato Clave:

Cada fotograma a 60 FPS requiere que millones de vértices completen este viaje exacto en cuestión de milisegundos.

FPS: 60
Vertices: 4,258,090
Frame Time: ~10.0ms

Fundamentos: Midiendo el Vacío

El Producto Punto

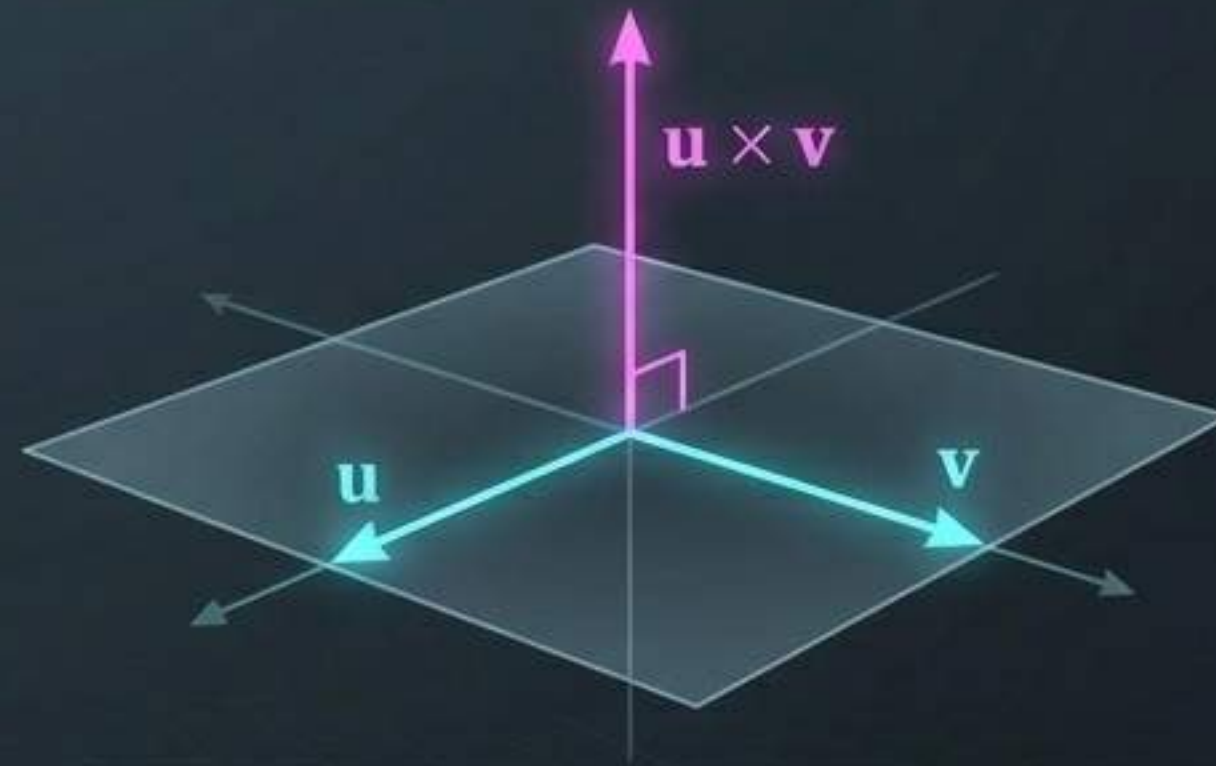


Concepto: Alineación y energía.

Fórmula: $\mathbf{u} \cdot \mathbf{v} = |\mathbf{u}||\mathbf{v}|\cos \theta$

Aplicación: ¿Cuánta luz incide sobre esta superficie? La energía es máxima cuando los vectores son paralelos ($\cos(0^\circ) = 1$).

El Producto Cruz

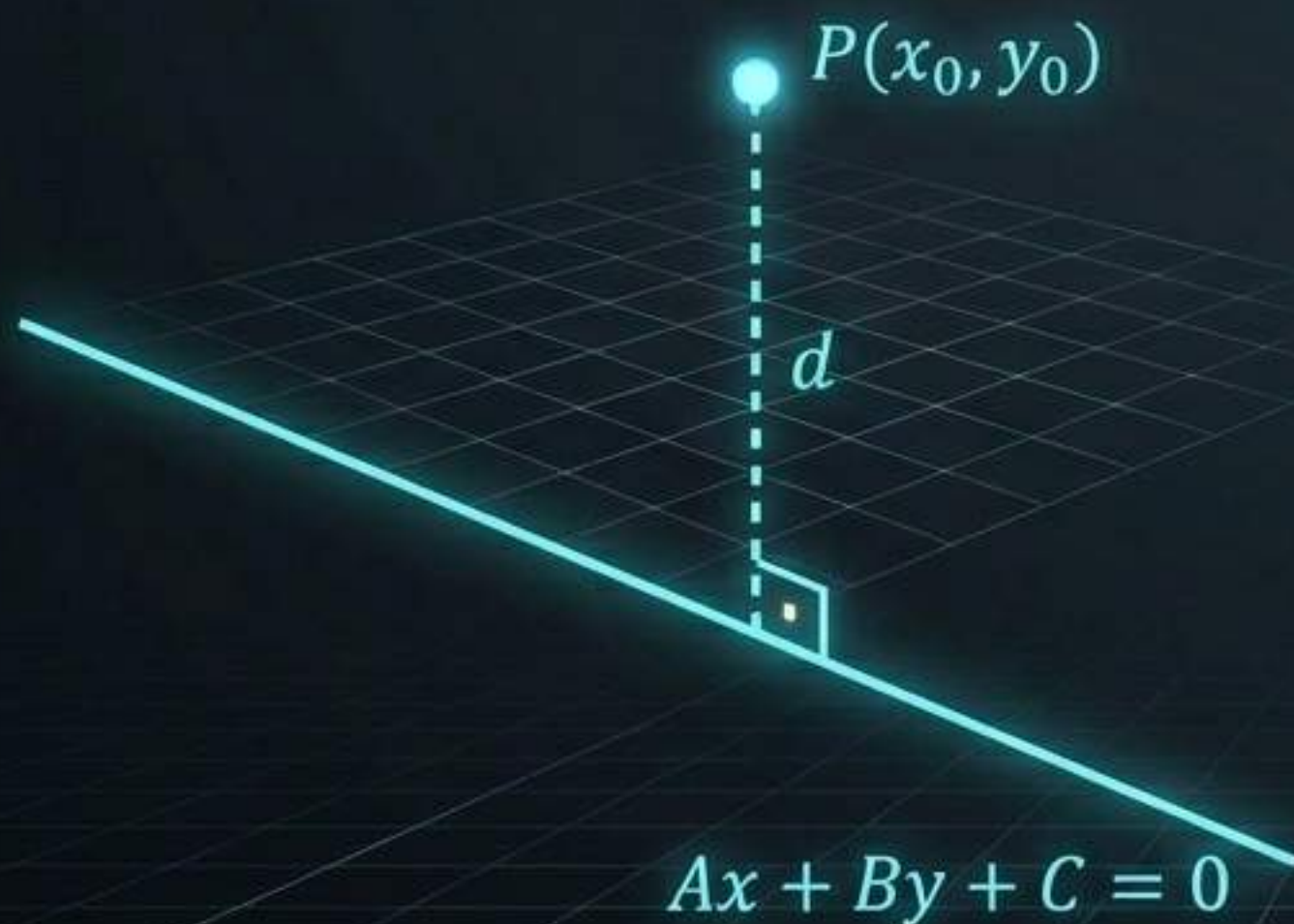


Concepto: Orientación y perpendicularidad.

Fórmula: $\mathbf{u} \times \mathbf{v}$

Aplicación: Construcción de normales. ¿Hacia dónde mira esta cara del polígono? (Esencial para el descarte de caras ocultas).

Colisión y Proyección: La Física del Impacto



En un entorno virtual, calcular interacciones físicas como colisiones o la proyección de sombras requiere encontrar la distancia más corta entre un objeto y un límite.

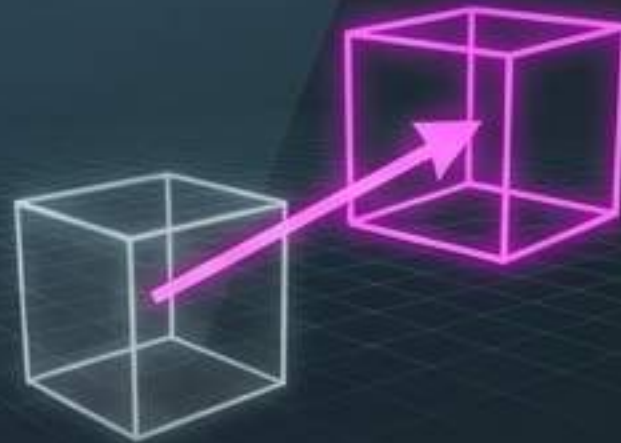
Distancia Perpendicular:
$$d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$$

Insight: La proyección del punto sobre la recta es la sombra matemática que caería siguiendo el mínimo recorrido. Esta es la base para la detección de colisiones a alta velocidad en motores físicos.

La Magia del Movimiento: Transformaciones Afines

Para animar un objeto, no reescribimos todos sus vértices. Aplicamos una transformación afín: una operación matricial que altera la posición mientras preserva la colinealidad y el paralelismo.

1. Traslación



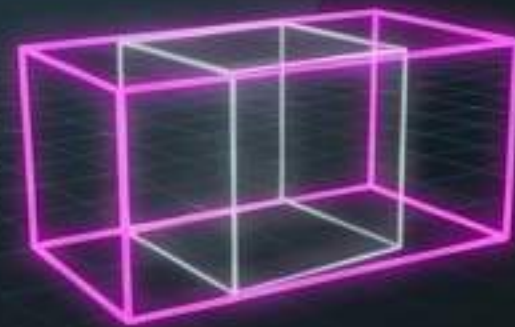
Desplazamiento en el espacio
($T(x) = x + b$)

2. Rotación



Giro matemático sobre
ejes definidos

3. Escala



Compresión o expansión a lo
largo de los ejes.

4. Sesgo (Shear)



Inclinación geométrica para
falsos efectos de perspectiva.

El Poder de Mover el Origen

Transformación Lineal ($L(x) = Ax$)

- **Regla Central:**
El origen (0,0) es inamovible y nunca cambia.
- **Capacidades:**
Puede ejecutar Rotación, Escala y Sesgo.
- **Limitación Crítica:**
Es matemáticamente incapaz de trasladar objetos en el espacio.

Transformación Afín ($T(x) = Ax + b$)

- **Regla Central:**
El origen puede cambiar de posición. Es una transformación lineal más un desplazamiento.
- **El Truco:**
Coordenadas Homogéneas.
- **Mecanismo:**
Al añadir una cuarta dimensión temporal auxiliar ($W=1$), convertimos la suma del desplazamiento en una multiplicación de matrices de 4x4 limpia, elegante y altamente paralelizable.

Nota de Hardware: Las GPUs modernas están diseñadas arquitectónicamente para calcular millones de estas matrices 4x4 simultáneamente, formando el motor principal del movimiento 3D.

Detectando Interacciones: El Láser y el Muro

Algoritmo Ray Casting (Point in Polygon).
¿Cómo sabe la computadora matemáticamente si un píxel está dentro de una forma irregular para colorearlo?



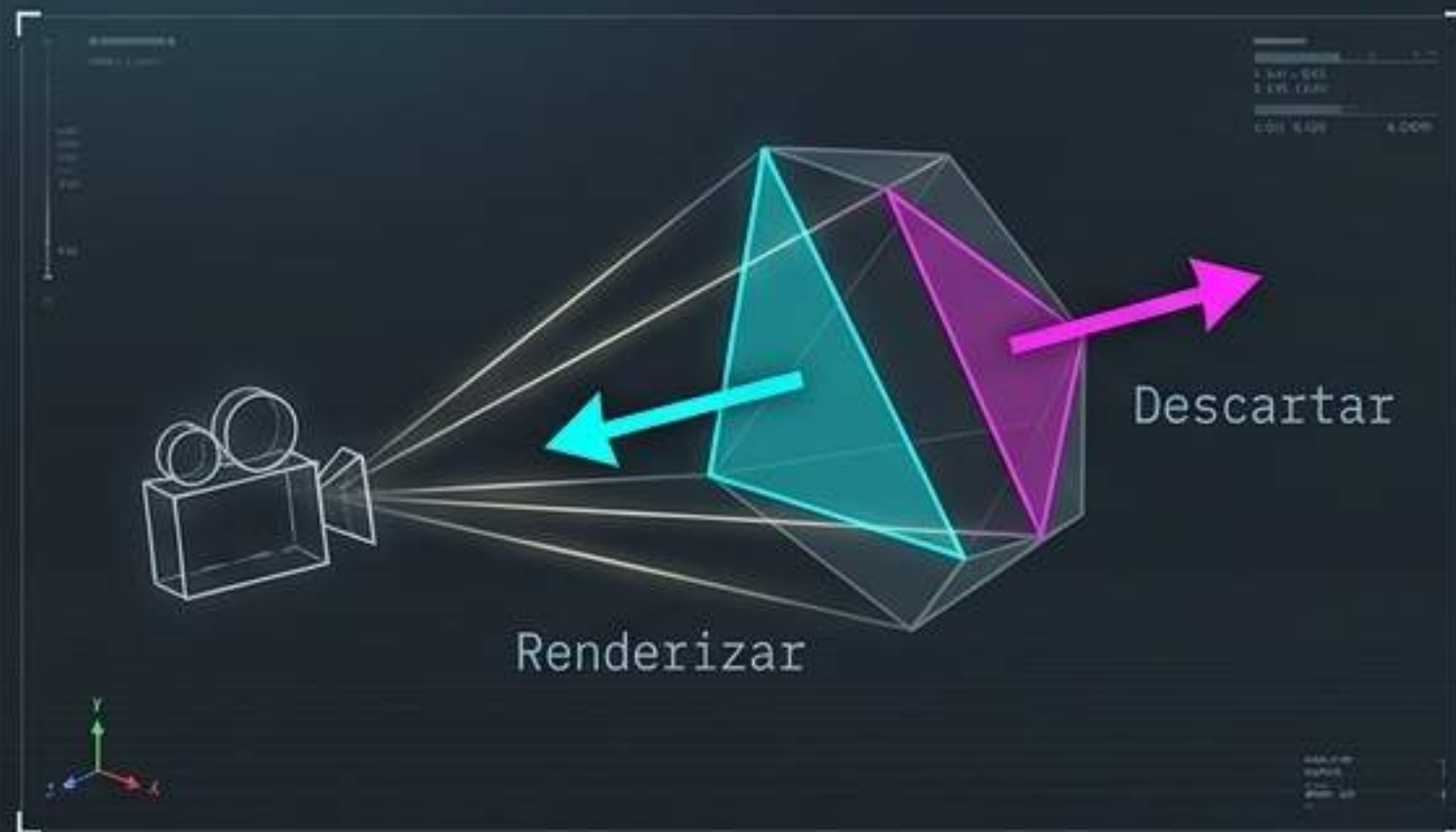
La Metáfora del Láser Infinito:

1. Disparamos un rayo horizontal desde el punto a evaluar.
2. Si el láser atraviesa los muros un número impar de veces, el punto quedó atrapado (DENTRO).
3. Si cruza un número par de veces, el punto ha escapado (FUERA).

Esta lógica $O(n)$ se aplica recorriendo los bordes del polígono y calculando las intersecciones lineales, siendo la piedra angular de la rasterización 2D.

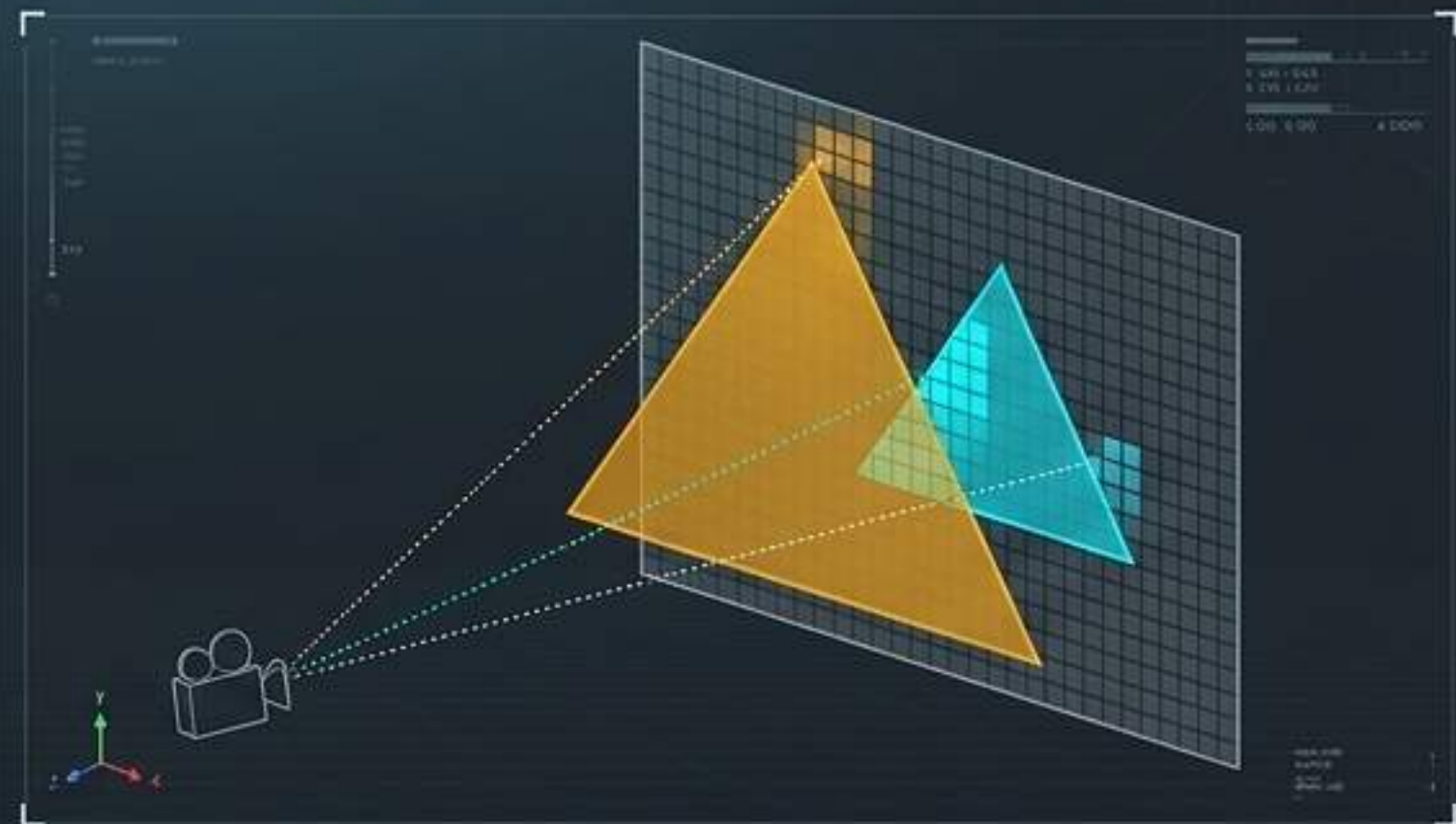
Visibilidad: Descartando el Exceso

Backface Culling (Descarte de Caras Ocultas)



- Usamos el Producto Cruz de los bordes del triángulo para obtener la Normal.
- Si la componente Z del vector normal apunta en dirección opuesta a la cámara, la cara está oculta. La ignoramos antes de gastar recursos en renderizarla.

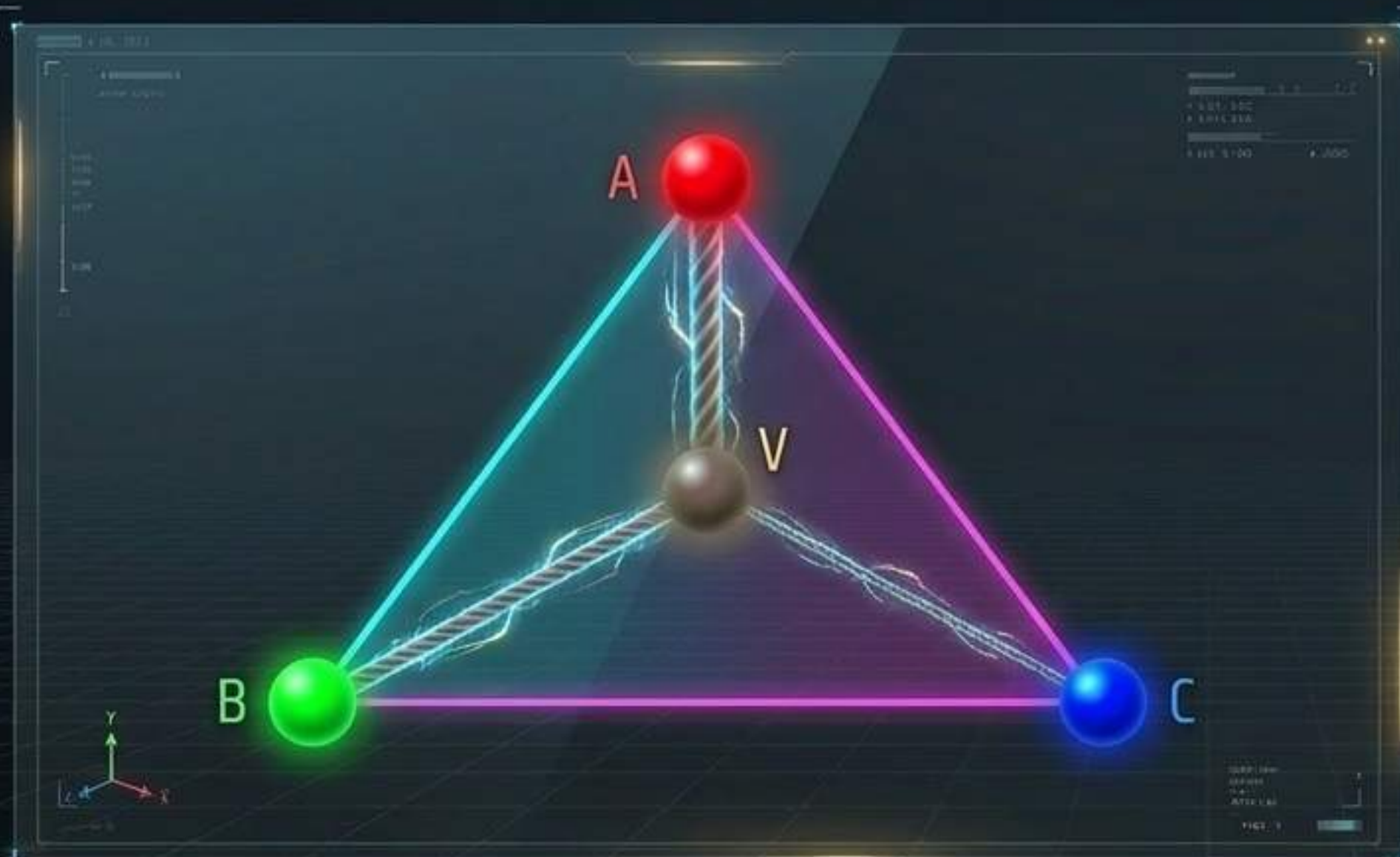
Z-Buffering (Buffer de Profundidad)



- Una cuadrícula de memoria invisible almacena la profundidad (Z) del objeto más cercano para cada píxel.
- Si el algoritmo intenta dibujar un nuevo punto, pero ya existe una pared más cercana en el 'Z-Buffer', el punto nuevo se descarta.
- Regla de Oro: ¡No calcules luz ni color para lo que el usuario jamás podrá ver!

Rellenando el Vacío: Coordenadas Baricéntricas

Una vez que sabemos qué está dentro y es visible, debemos interpolar los datos de las esquinas (color, textura, luz) hacia el centro de la superficie.



La Metáfora Visual: El Tirón de Cuerdas

Imagina el punto interior sostenido por tres cuerdas ancladas a los vértices. El grosor (peso) de cada cuerda define cuánta influencia tiene esa esquina sobre el punto central.

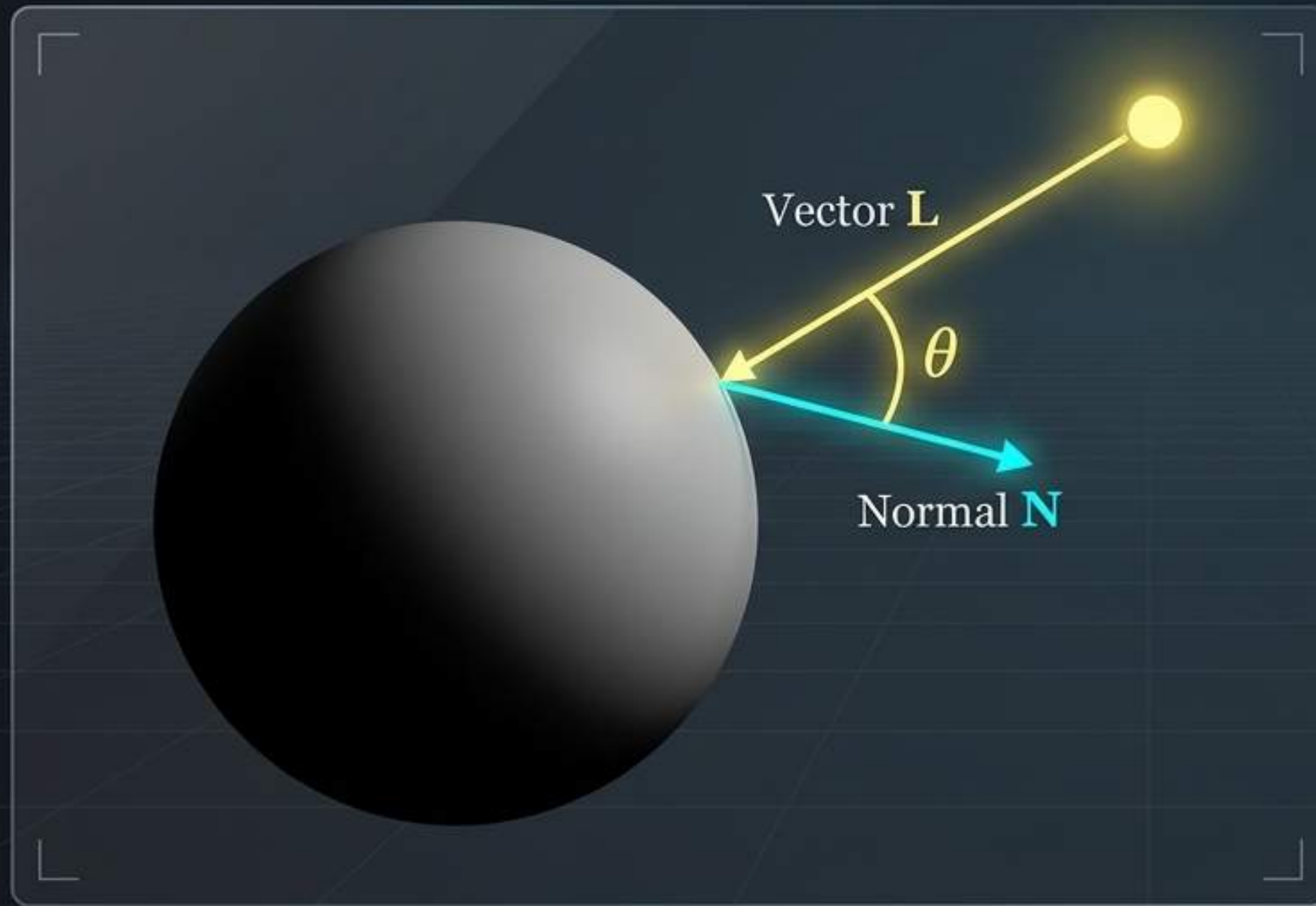
$$V = \alpha A + \beta B + \gamma C$$

Donde los pesos están determinados por el área del sub-triángulo opuesto.

Condición Estricta: $\alpha + \beta + \gamma = 1$. Si los tres valores están entre 0 y 1, tenemos la certeza matemática de que el punto reside estrictamente dentro de los límites del polígono.

Matemáticas de la Luz: Reflectancia Lambertiana

¿Cómo sabe una superficie digital cuán brillante debe ser? La física fundamental de la luz difusa (superficies mate como madera, yeso o papel) se reduce a una sola operación de álgebra lineal.



El Núcleo Matemático:

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

La Traducción Visual:

- La intensidad de la luz es directamente proporcional al Producto Punto del vector de la luz y el vector normal de la superficie.
- Si la luz golpea de forma perfectamente perpendicular, la alineación matemática es máxima (1.0).
- Si la luz proviene desde atrás de la superficie, el producto punto arroja un número negativo. La función 'max(0, ...)' descarta esto, convirtiéndolo en sombra pura (0.0).

Añadiendo Volumen: El Modelo de Phong

Ambiente (Constante base)

La luz residual ambiental. Evita que las sombras virtuales sean 100% negras.

+

Difusa (Lambertiana)

La luz directa y suave basada en el ángulo de impacto del fotón contra la superficie.

+

Especular (El Brillo Fotorrealista)

El reflejo directo y agudo de la fuente de luz rebotando exactamente hacia la cámara virtual.

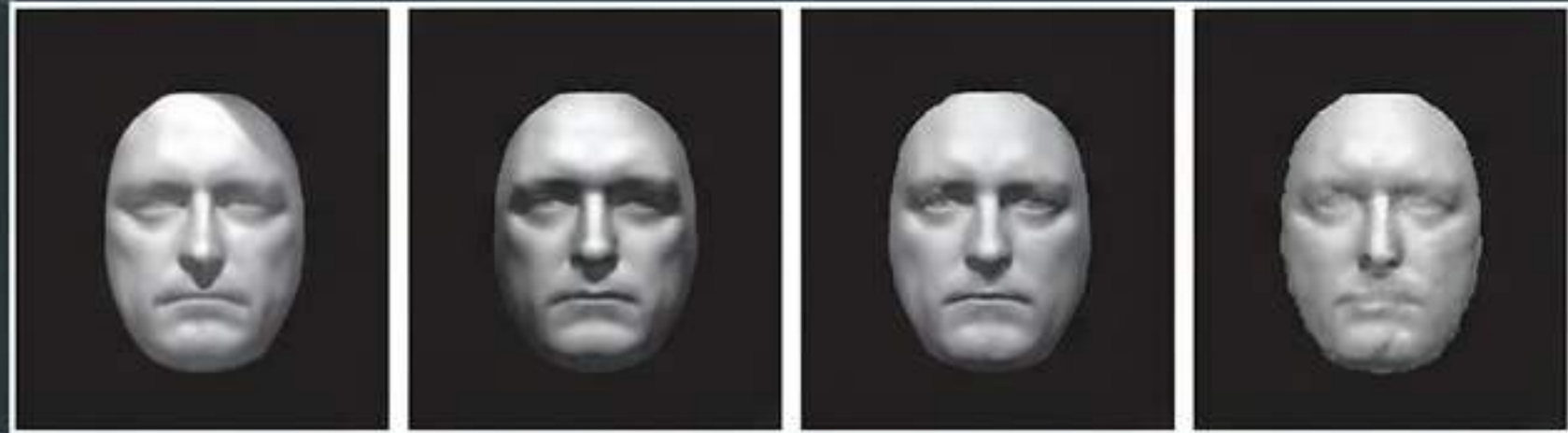
El Secreto Especular

Para lograr el brillo, usamos un exponente matemático de brillo (p). Elevar el producto punto a una potencia alta comprime y afila la luz resultante, creando un punto brillante cerrado y nítido, imitando perfectamente la física del plástico pulido o el metal.



El Límite Matemático: El Problema del Bajorrelieve

La simulación de iluminación genera ambigüedades perceptuales severas que las matemáticas puras no pueden resolver sin un contexto geométrico adicional.



La Ambigüedad Visual:

- Una cara cóncava iluminada desde abajo produce exactamente el mismo mapa matemático de sombras que una cara convexa iluminada desde arriba.

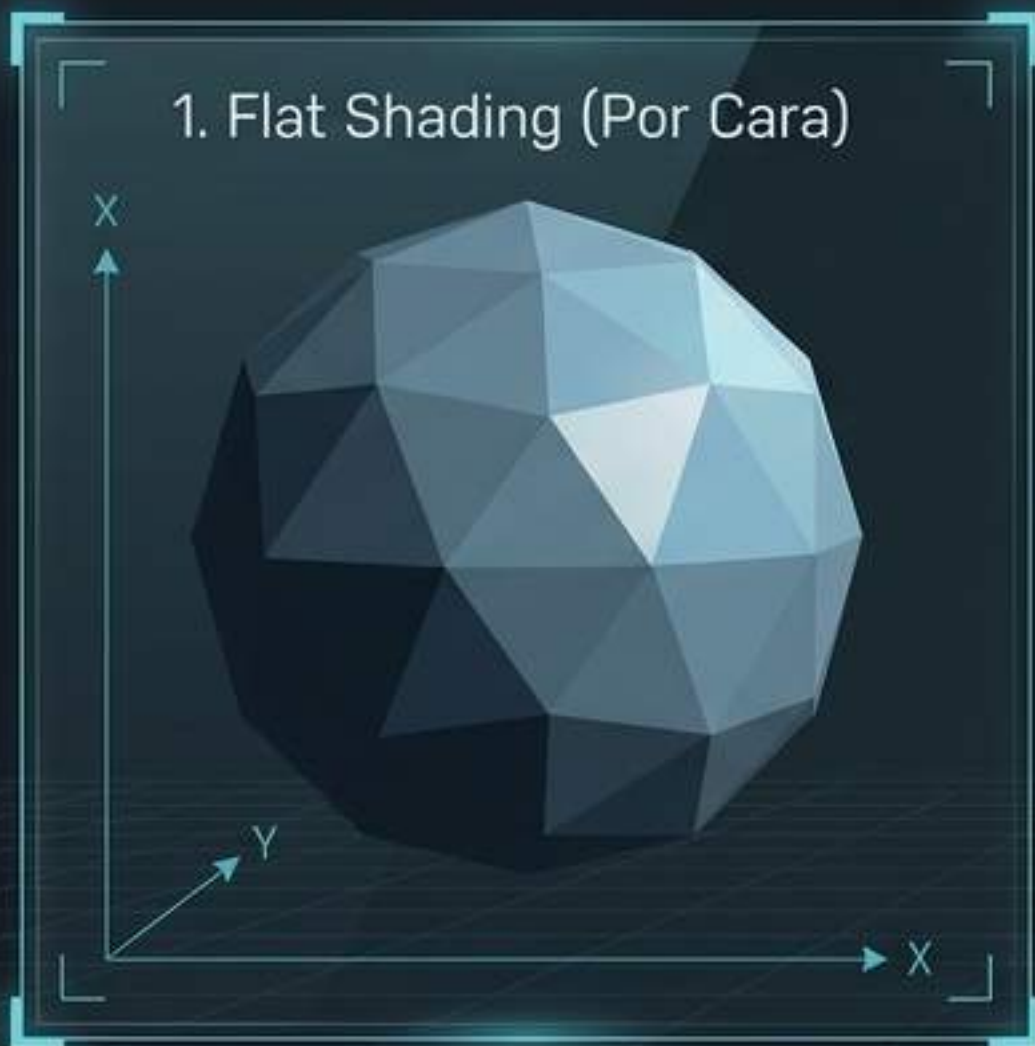
Transformación KGBR (K-Generalized Bas-Relief):

- Bajo esta transformación afín, las sombras proyectadas y adheridas se preservan matemáticamente idénticas, engañando a nuestra visión.

Conclusión: El motor de renderizado calcula la luz simulando la realidad, pero es el cerebro humano del espectador quien finalmente ensambla la imagen 3D.

Pintando la Geometría: Algoritmos de Sombreado (Shading)

Tenemos la geometría y tenemos las ecuaciones de luz. La pregunta computacional es: ¿Con qué frecuencia y en qué momento calculamos esa luz?



La fórmula de luz se calcula una sola vez por cada triángulo. Es ultrarrápido, pero produce un aspecto rígido, poligonal y computarizado antiguo.



La luz se calcula en las 3 esquinas y el color final se interpola hacia el centro usando coordenadas baricéntricas. Suaviza la forma, pero destruye y deforma los brillos especulares.



Interpolamos los vectores normales en la superficie y ejecutamos la matemática de luz en cada píxel individual de la pantalla. Requiere cómputo masivo, pero logra realismo absoluto.

Matriz de Diagnóstico: Calidad vs Rendimiento

Técnica	Flat Shading	Gouraud Shading	Phong Shading
Punto de Cómputo	 Centro del Polígono	 Los 3 Vértices	 Cada Píxel
Interpolación	 Ninguna	 Solo Color (Baricéntrico)	 Normales (Baricéntricas)
Costo GPU	Ultrabajo  (Barras de progreso cortas verdes)	Medio  (Barras amarillas)	Extremo  (Barras rojas al máximo)
Precisión Especular	 Pobre / Inexistente	 Distorsionada / Quebrada	 Impecable / Fotorrealista 

El Estándar Moderno: En motores contemporáneos como Unreal Engine o Unity, el Sombreado de Phong (o variaciones PBR) es el estándar predeterminado. Esto es posible únicamente gracias al procesamiento paralelo masivo (miles de núcleos) de las GPUs modernas.

Síntesis: La Anatomía de un Fotograma



Malla Vectorial: Posiciones puras en el abismo del espacio 3D (Álgebra Vectorial).

Capa Afin: Transformaciones proyectando los vértices a la pantalla 2D (Matrices Homogéneas).

Capa Baricéntrica: Coordenadas rellenando vacíos y descartando lo oculto (Ray Casting & Z-Buffer).

Capa de Shading: Vectores normales colisionando matemáticamente con luz virtual (Producto Punto & Phong).

Todas estas complejas ecuaciones matemáticas ocurren de forma simultánea, miles de veces por segundo, transformando el abismo numérico en la pincelada invisible del arte digital.