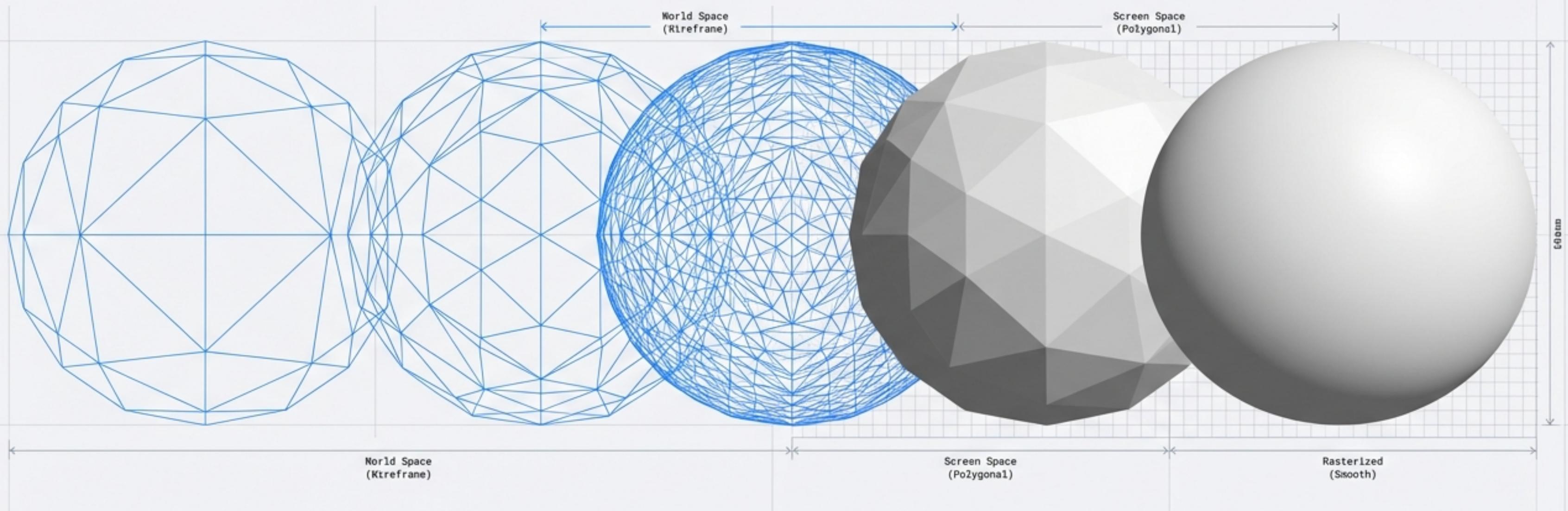


Rasterización de Triángulos

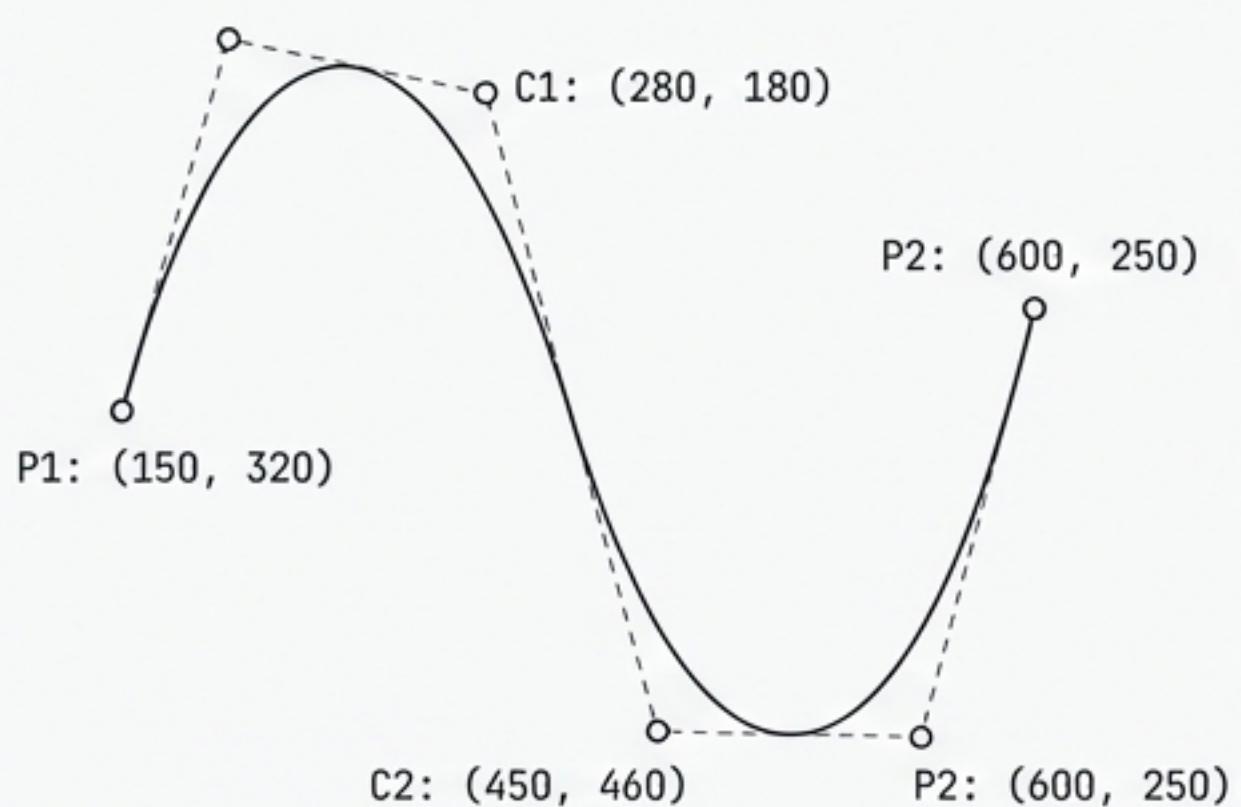
Cómo la GPU convierte geometría en píxeles



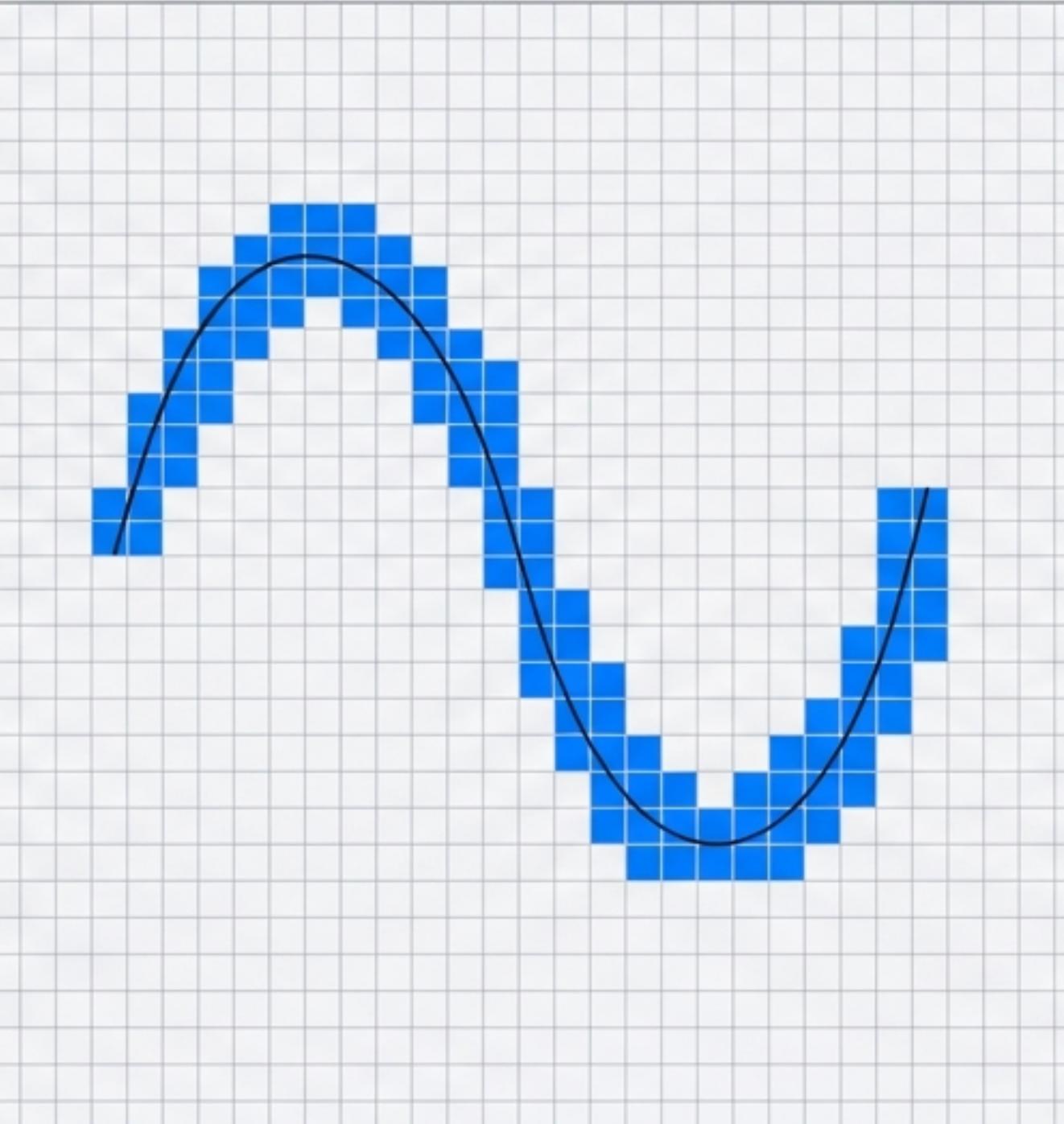
Una inmersión técnica en el puente crítico entre el mundo vectorial continuo y la grilla discreta de la pantalla. Un análisis de los algoritmos que definen la computación gráfica moderna.

El Problema de la Discretización

Mundo Vectorial



Mundo Raster



Definición

La rasterización es el proceso de muestreo que convierte geometría vectorial continua (vértices en espacio 3D) en una cuadricula discreta de píxeles.

Contexto

A diferencia del ray tracing, esta técnica domina los gráficos en tiempo real debido a su ejecución en paralelo masivo en el hardware.

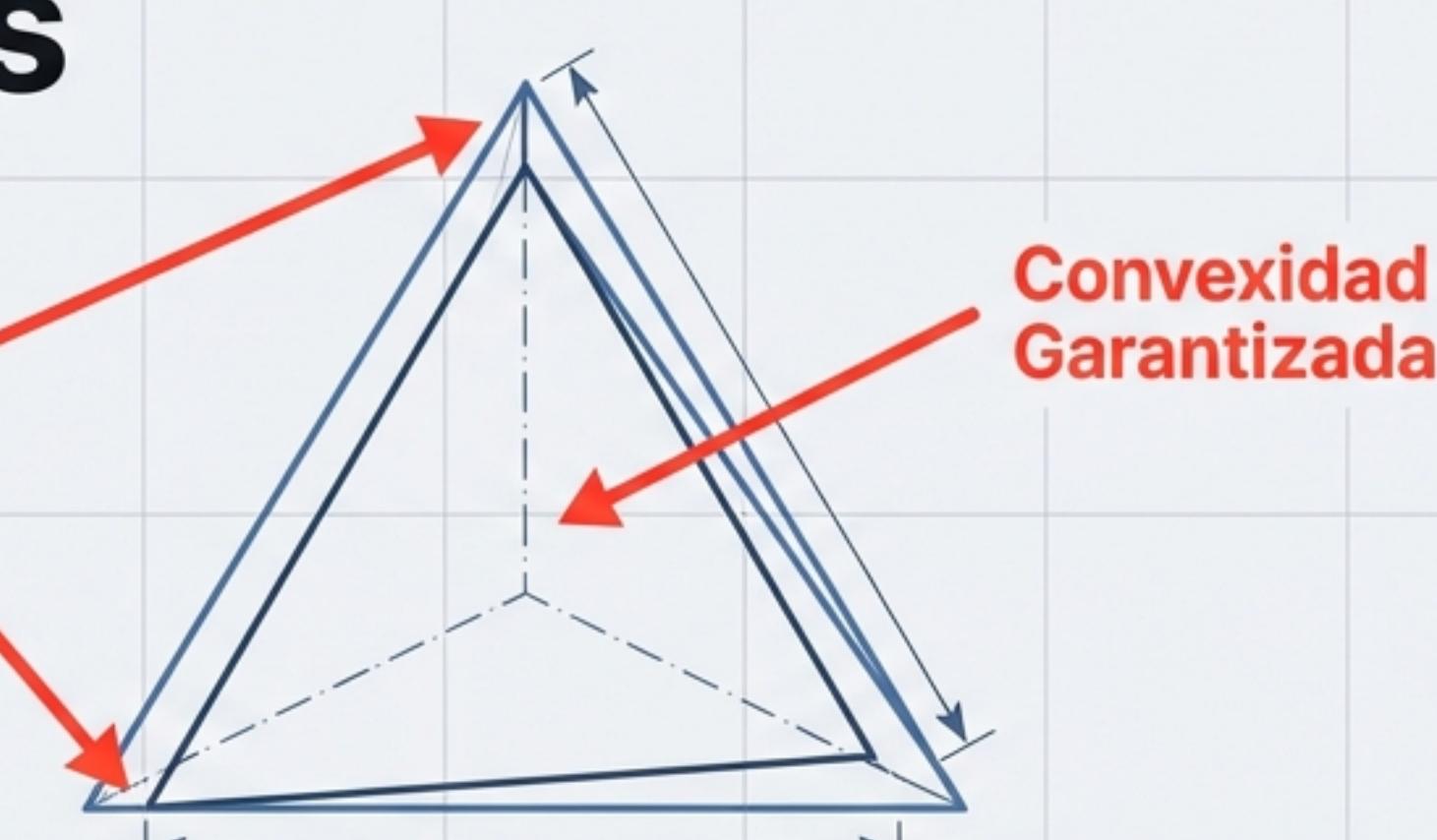
La Arquitectura del Pipeline



- **Posición Central:** La rasterización vive en el centro exacto del flujo de la GPU.
- **Input:** Recibe triángulos transformados y preparados.
- **Output:** Entrega 'fragmentos' listos para ser coloreados.

Por Qué el Hardware Ama los Triángulos

3 Puntos =
Plano Único



Universalidad:

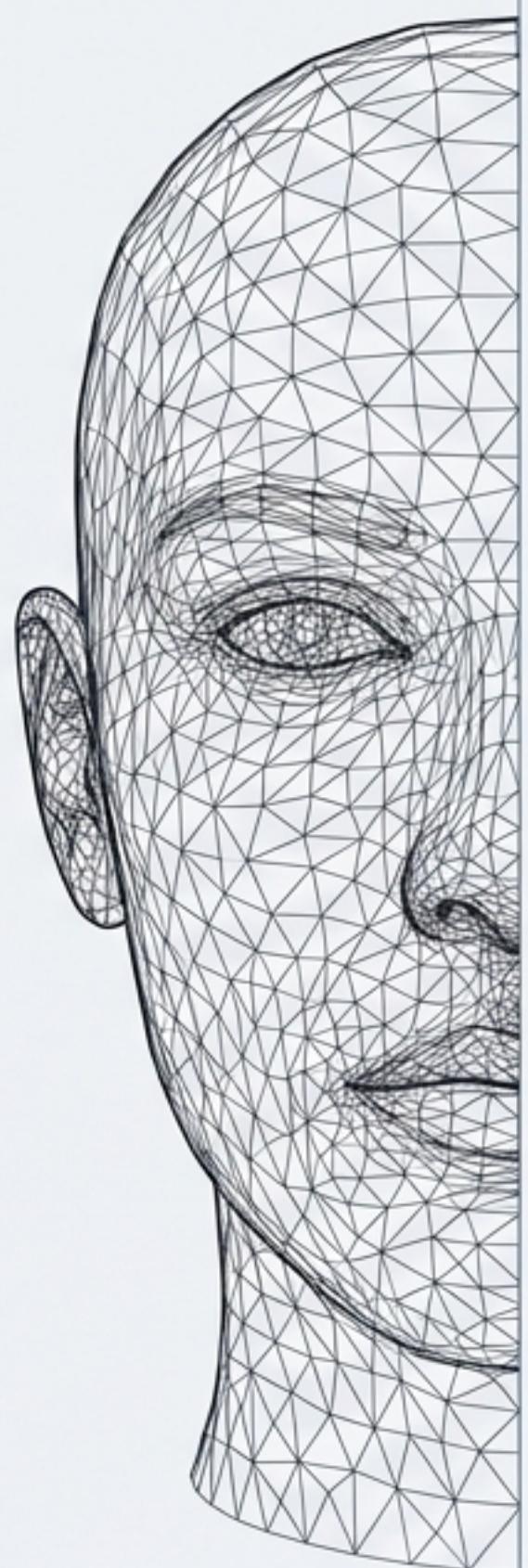
Cualquier malla compleja se descompone en triángulos.

Garantía Planar:

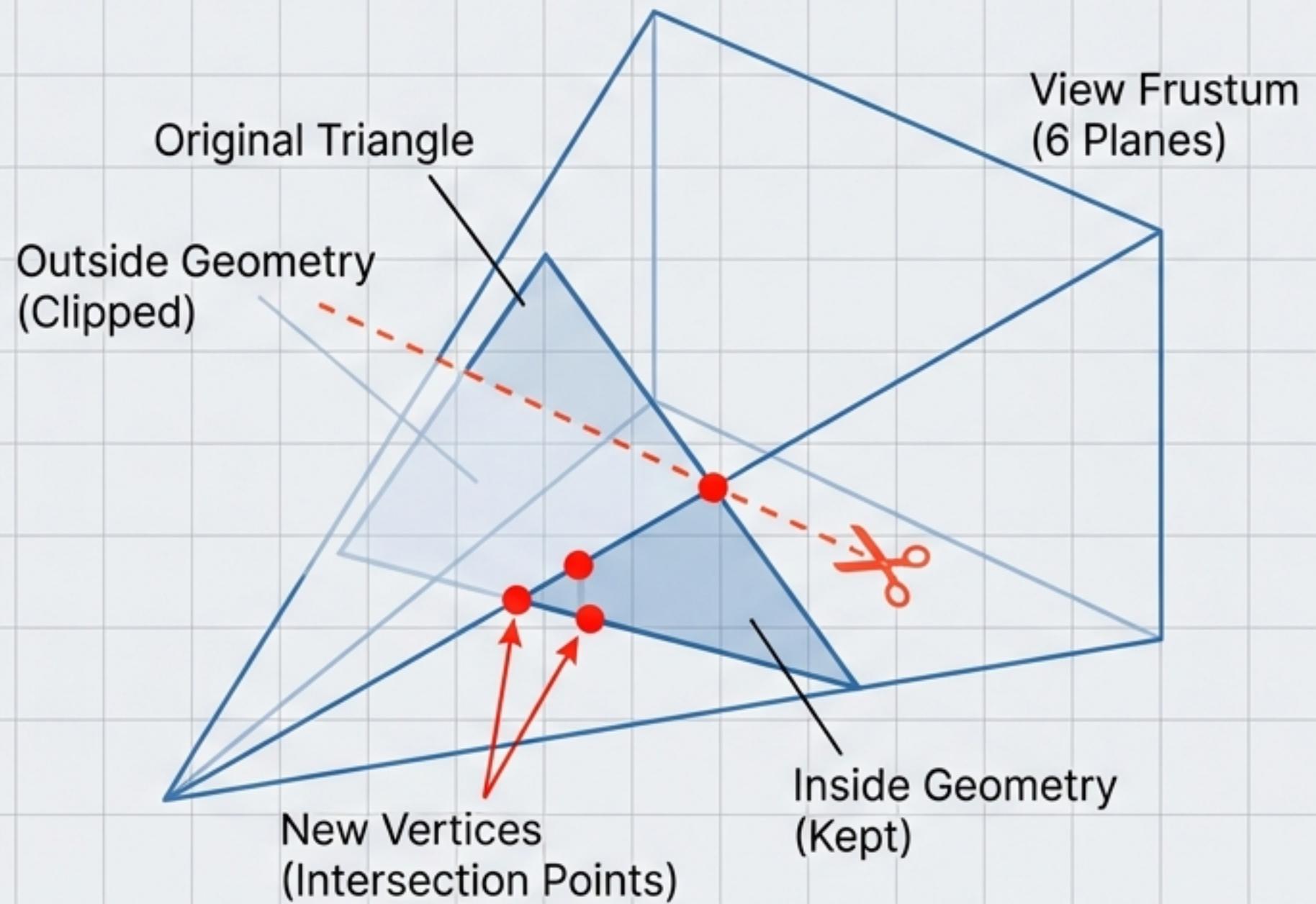
Tres puntos siempre definen un plano inequívoco en el espacio.

Optimización:

El hardware gráfico lleva décadas siendo diseñado exclusivamente para procesar esta primitiva.



Preparación 1: Clipping (Recorte)

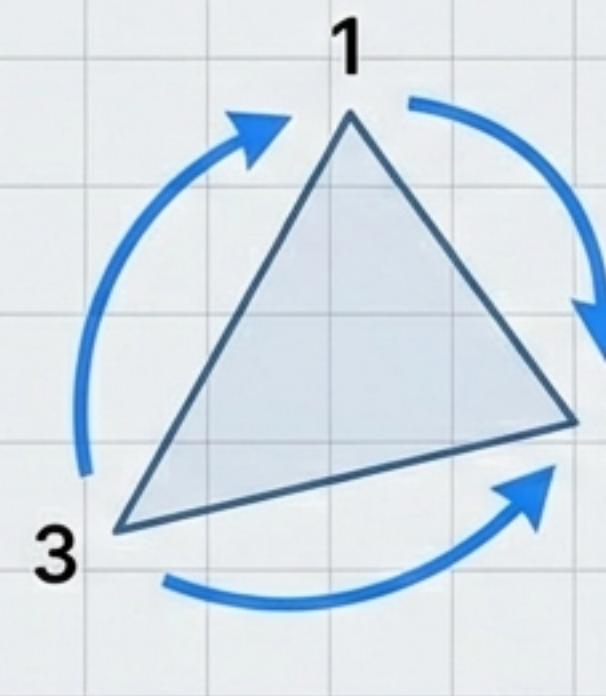


El Algoritmo: Se utiliza **Sutherland-Hodgman** para recortar la geometría contra los **6 planos del frustum**.

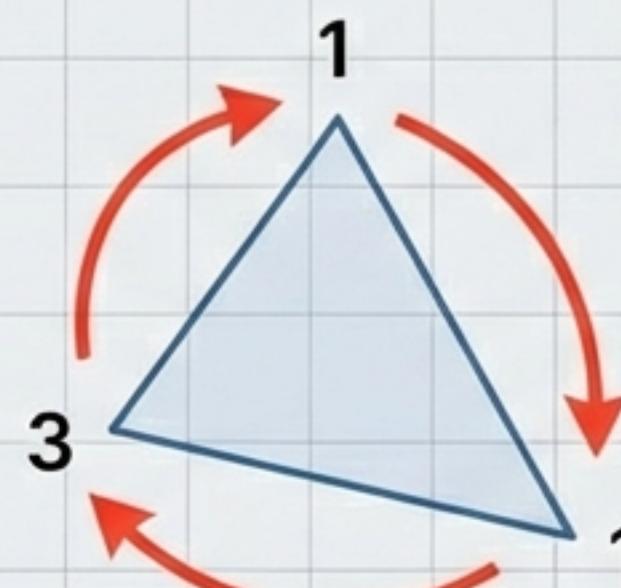
****La Transformación****: Un triángulo que cruza un plano se **subdivide**. Un solo triángulo parcialmente visible puede convertirse en un polígono de hasta 7 vértices.

Preparación 2: Back-Face Culling

A. Visible (Frontal)

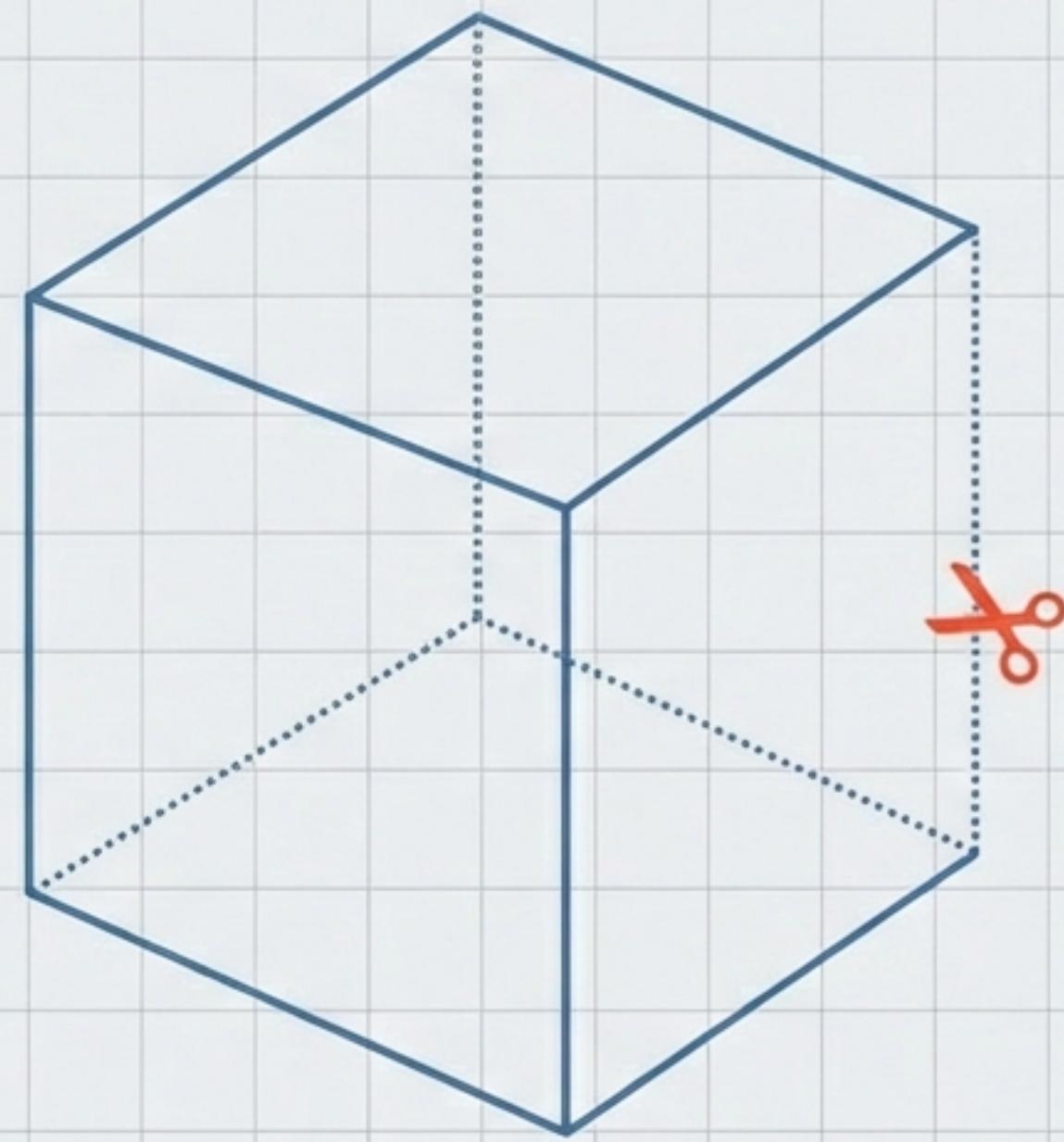


B. Descartado (Trasera)



Eficiencia: Descartar las caras traseras invisibles ahorra ~50% del procesamiento.

Winding Order: La GPU determina la visibilidad basándose en el orden de los vértices (horario vs antihorario).

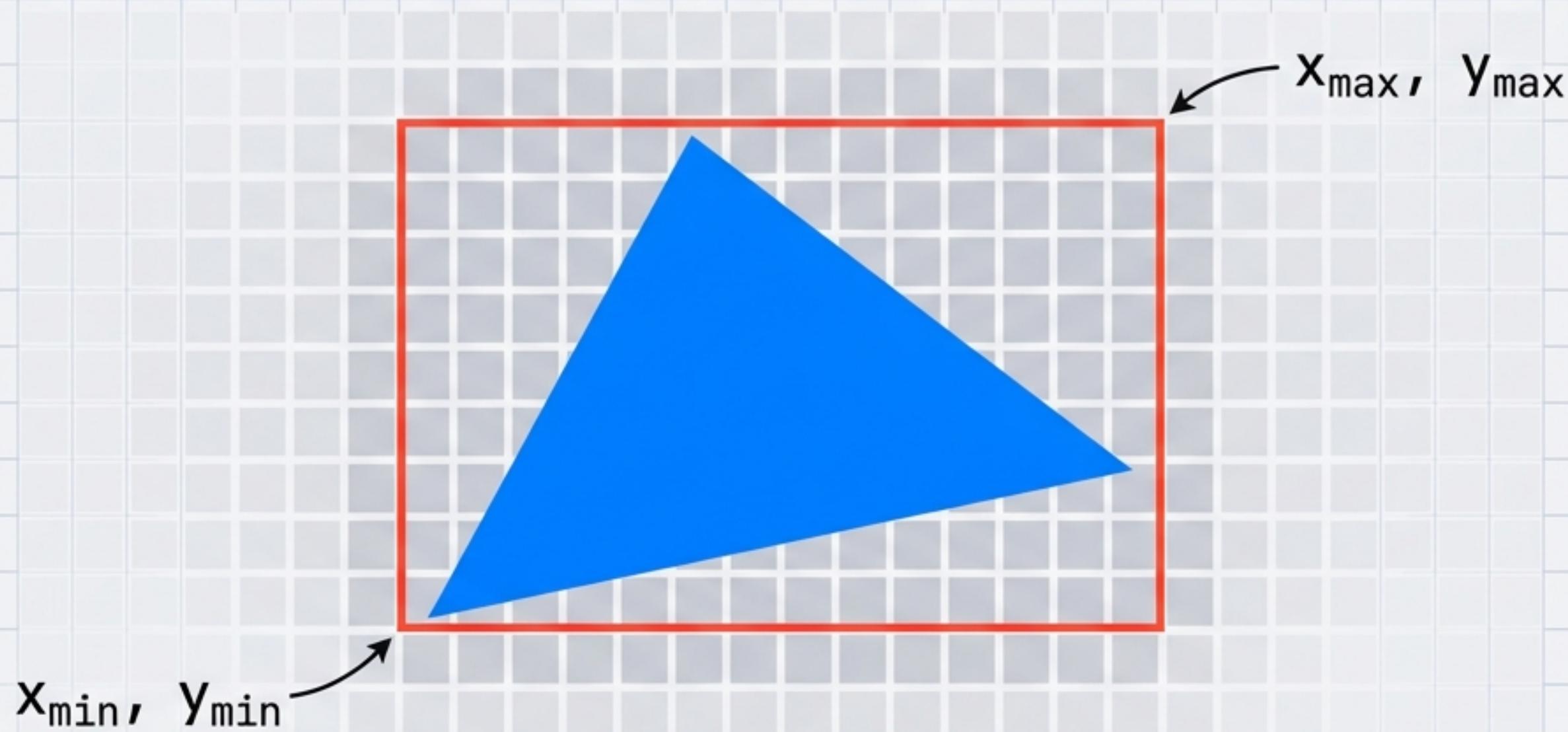


El Algoritmo de Rasterización



Una vez que el triángulo es visible y está en pantalla, **el hardware ejecuta cuatro pasos críticos para convertirlo en píxeles. Esta es la "Caja Negra" matemática del proceso.**

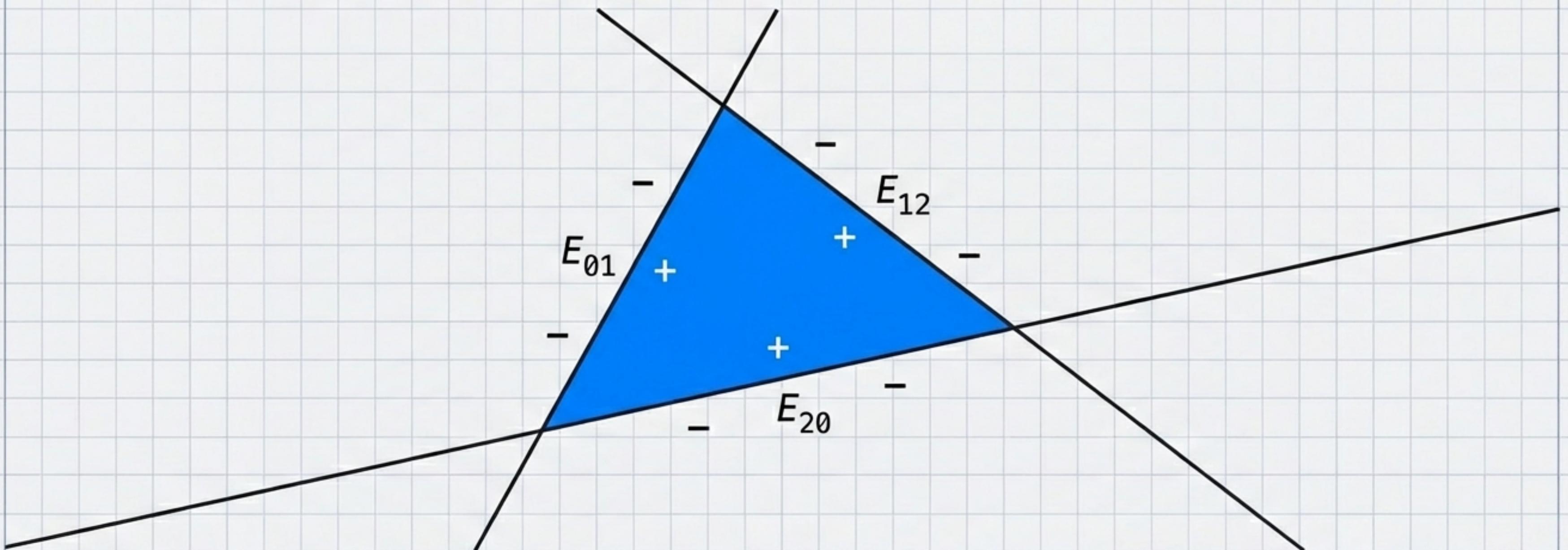
Paso 1: Bounding Box



Estrategia: Se calcula el rectángulo mínimo que encierra el triángulo.

Eficiencia: El rasterizador solo iterará y probará los píxeles que caigan dentro de esta caja delimitadora, evitando procesar millones de píxeles vacíos.

Paso 2: Edge Functions



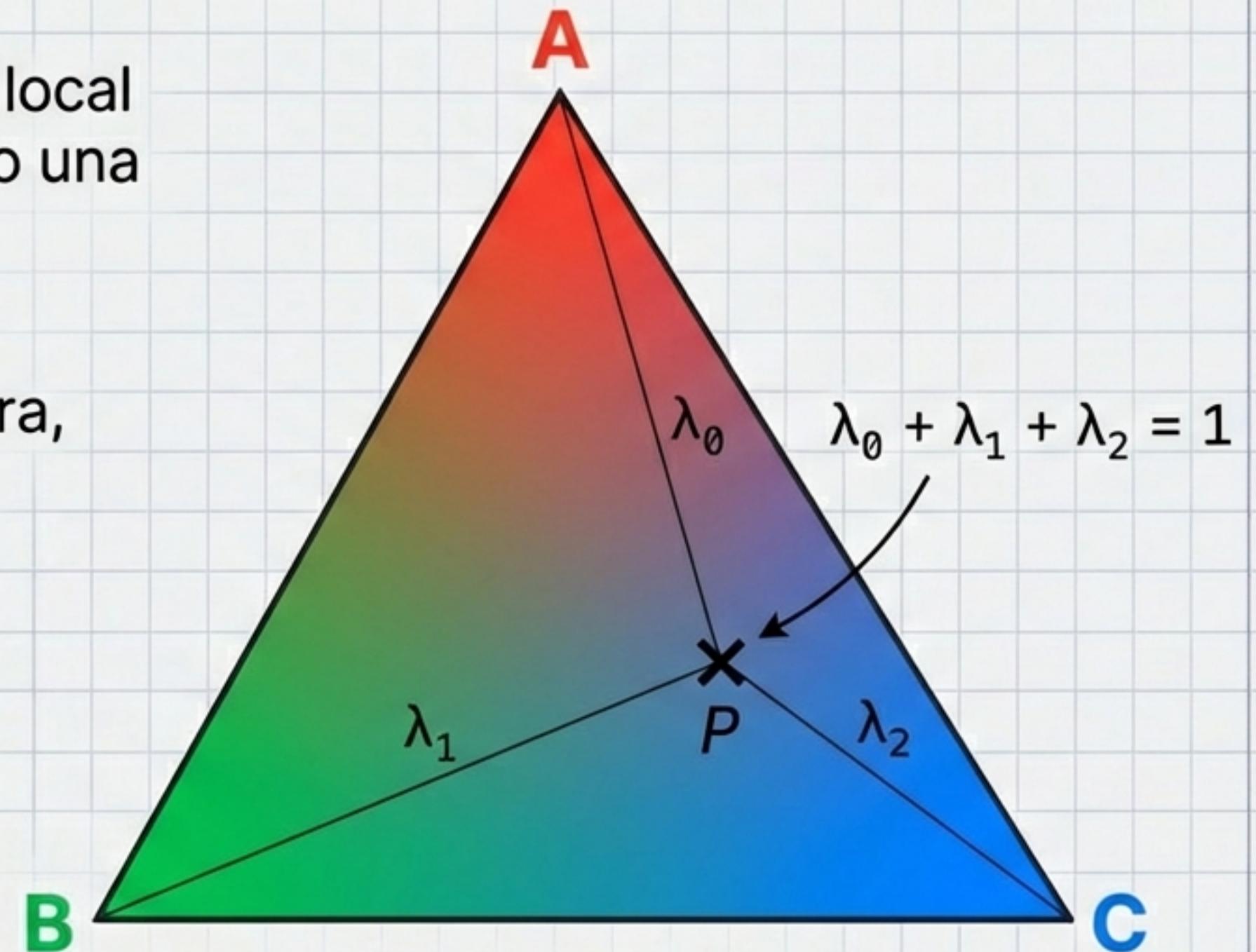
La Ecuación: $E(x, y) = (x - x_0)(y_1 - y_0) - (y - y_0)(x_1 - x_0)$

El Veredicto: Si el resultado de las tres funciones lineales es positivo, el píxel está matemáticamente 'dentro' del triángulo.

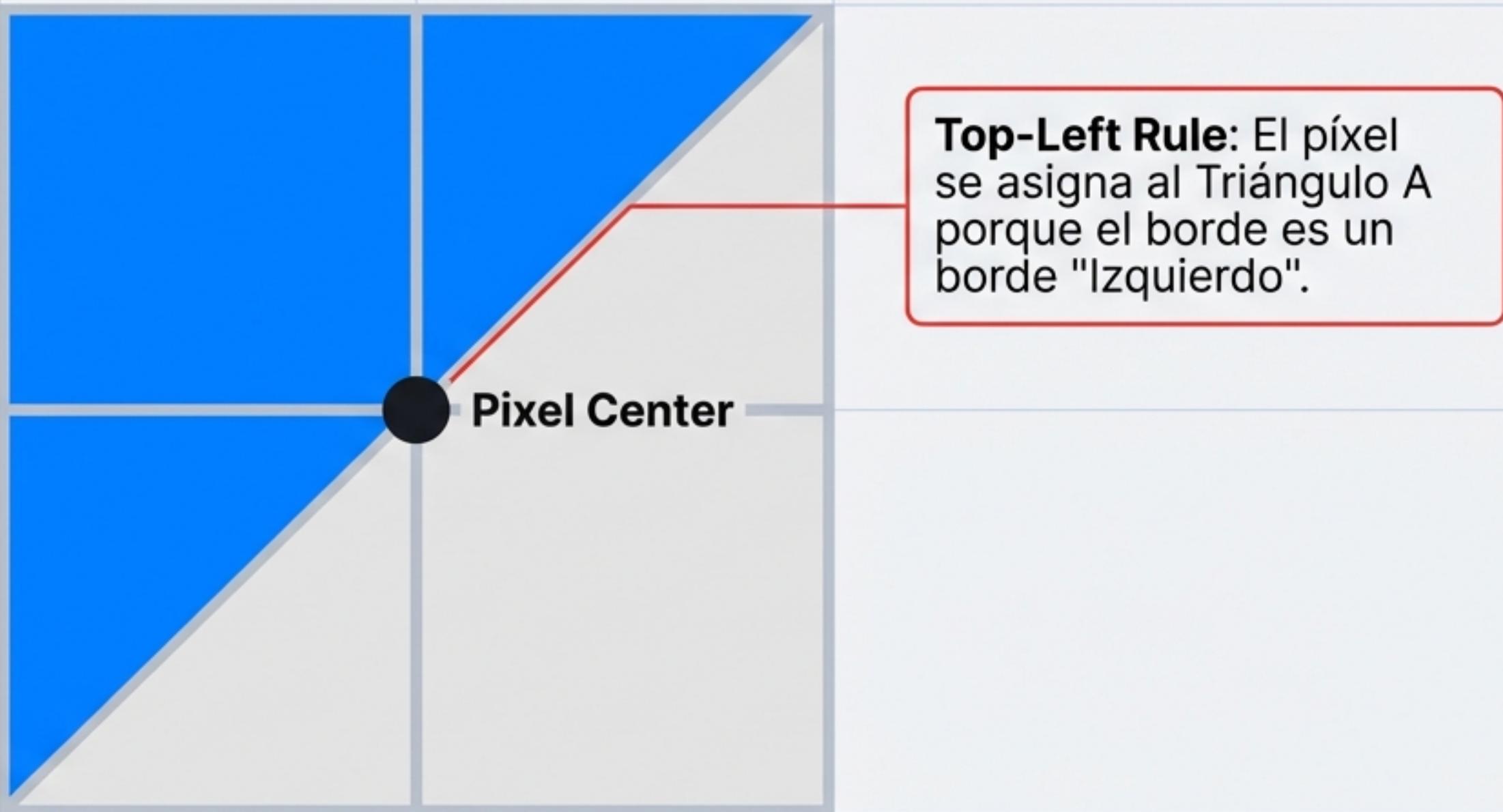
Paso 3: Coordenadas Baricéntricas

Definición: Un sistema de coordenadas local que expresa la posición de un píxel como una suma ponderada de los tres vértices.

Uso: Estos pesos (λ) son la clave para interpolar cualquier atributo (color, textura, profundidad) en el interior del triángulo.



Paso 4: Fill Rules



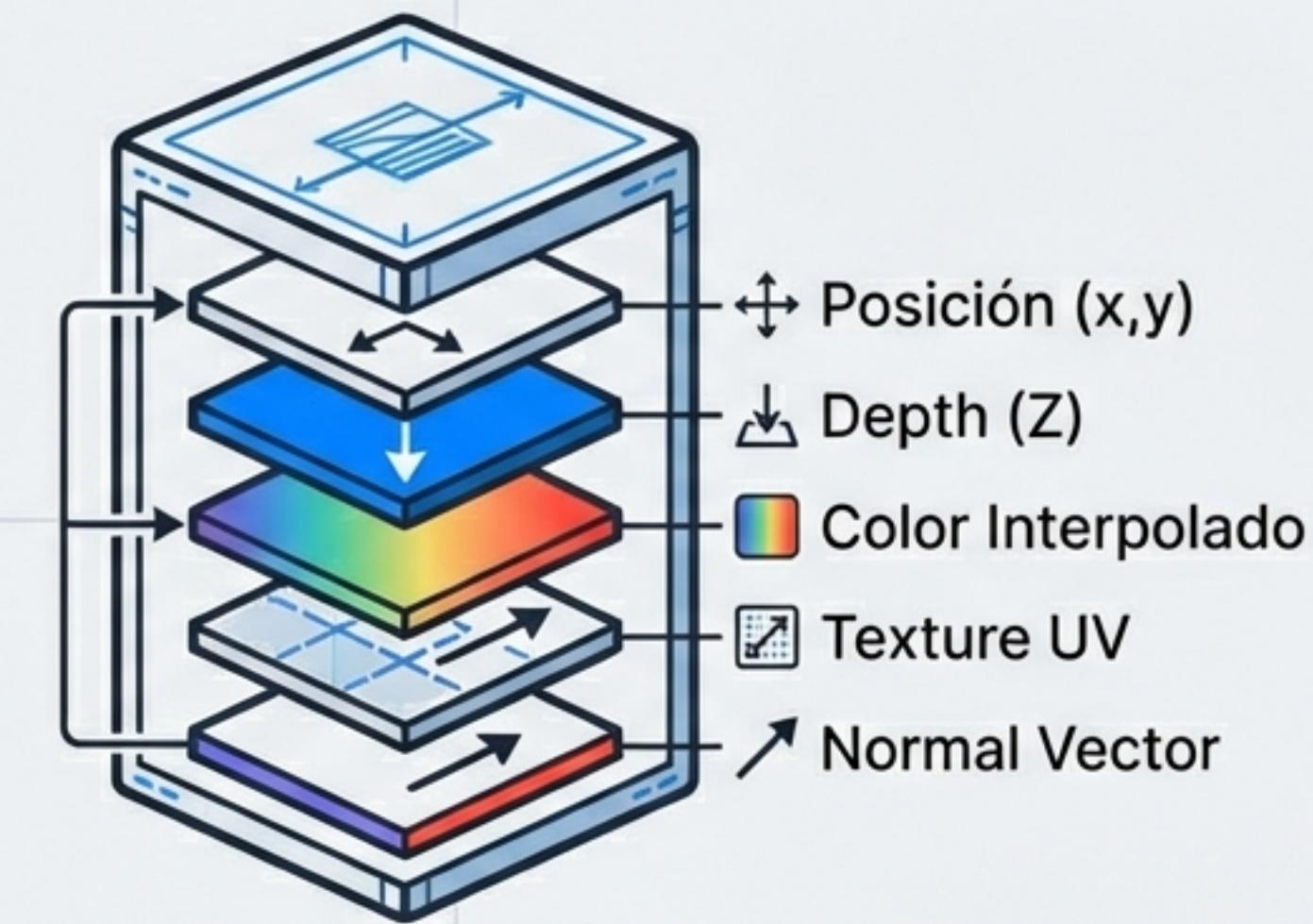
El Dilema: ¿Qué pasa si un centro de píxel cae exactamente en el borde compartido?

La Solución: La **Top-Left Rule**. Garantiza que cada píxel pertenezca a un solo triángulo en una malla conectada, evitando huecos o doble renderizado.

El Resultado: Nace el Fragmento



Píxel (Solo Luz)

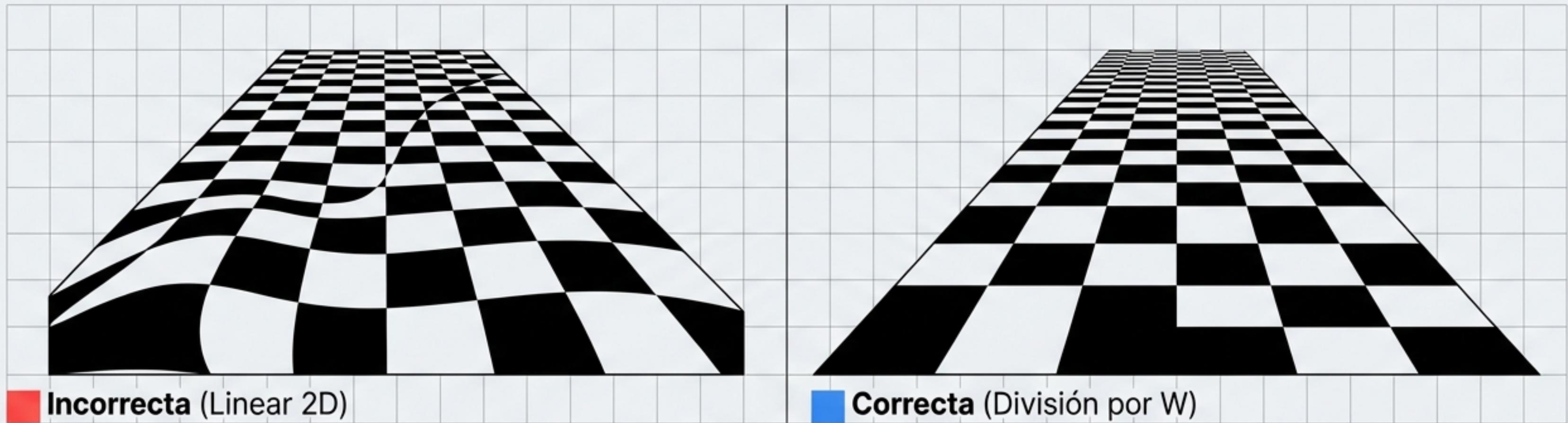


Fragmento (Candidato)

Distinción Crucial: Un fragmento no es todavía un píxel. Es un *candidato*.

El Payload: Transporta un paquete de datos interpolados necesarios para el sombreado (Shading).

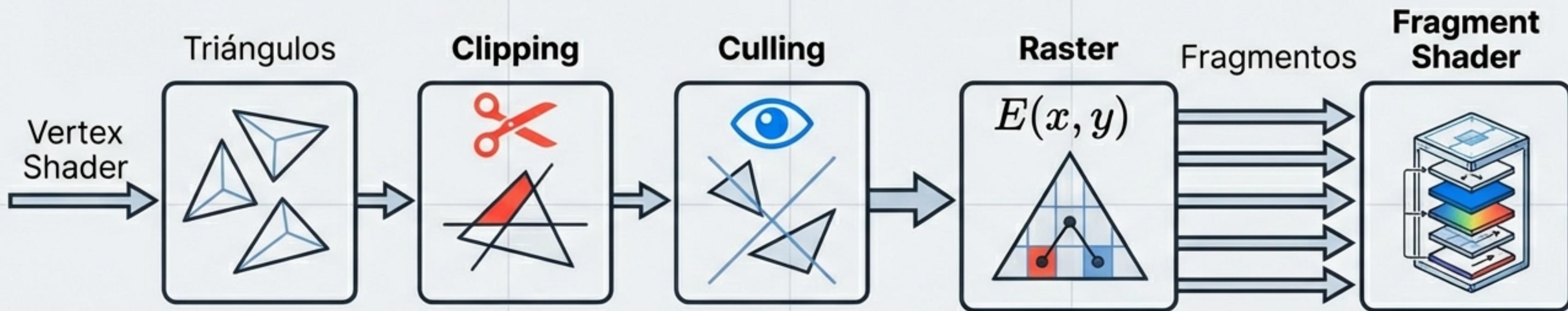
Interpolación y Corrección de Perspectiva



El Desafío: La interpolación lineal en 2D distorsiona las texturas en 3D.

La Matemática: La GPU realiza la interpolación en los valores divididos por W (componente homogénea) para recuperar las coordenadas correctas.

Recapitulación del Viaje



1. **Input**: Triángulos desde el Vertex Shader.
2. **Input**: Triángulos desde el Vertex Shader.
3. **Clipping/Culling**: Recorte y descarte de invisibles.
3. **Rasterización**: Test de cobertura y mezcla baricéntrica.
4. **Output**: Fragmentos enviados al Fragment Shader.

Escala Masiva

Todo el proceso matemático detallado en este documento —recortes, funciones de borde, interpolaciones— ocurre millones de veces por segundo, en paralelo.

Esta es la potencia de la rasterización: la fuerza bruta de la ingeniería convertida en arte visual.

