

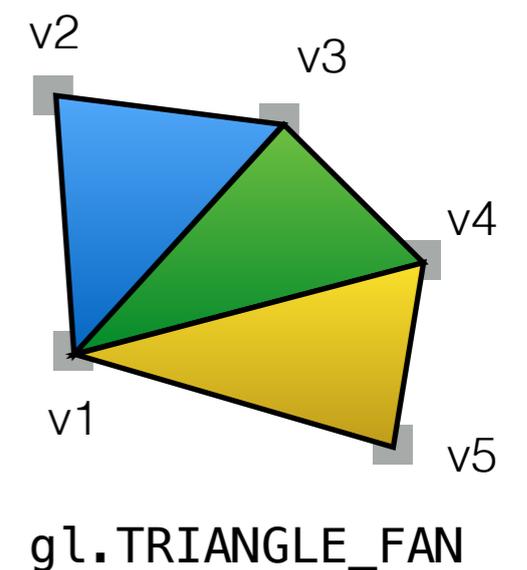
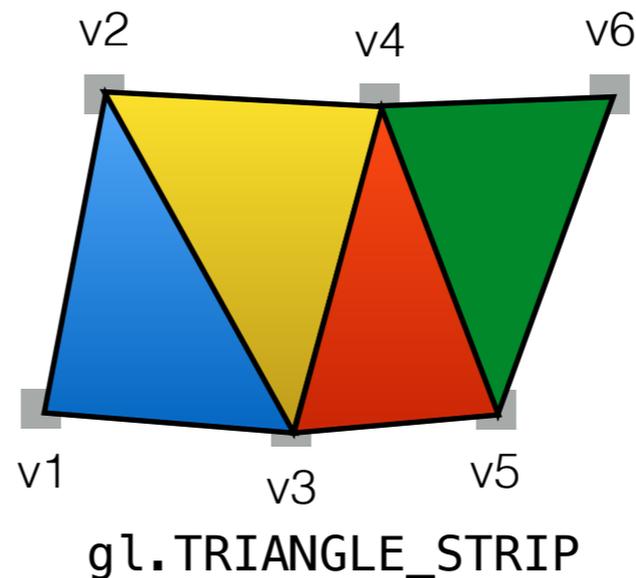
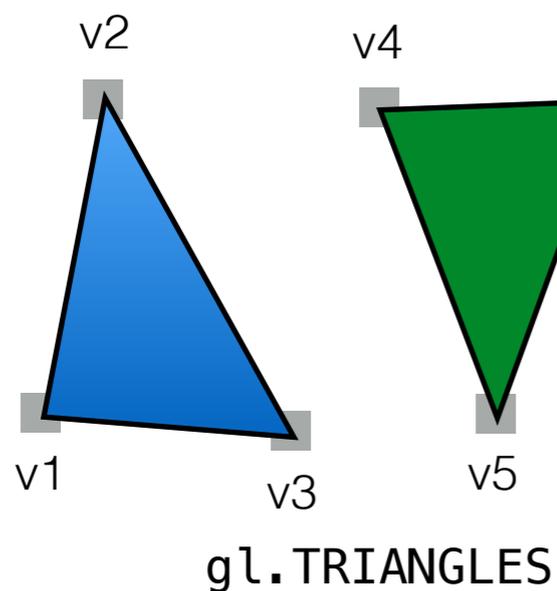
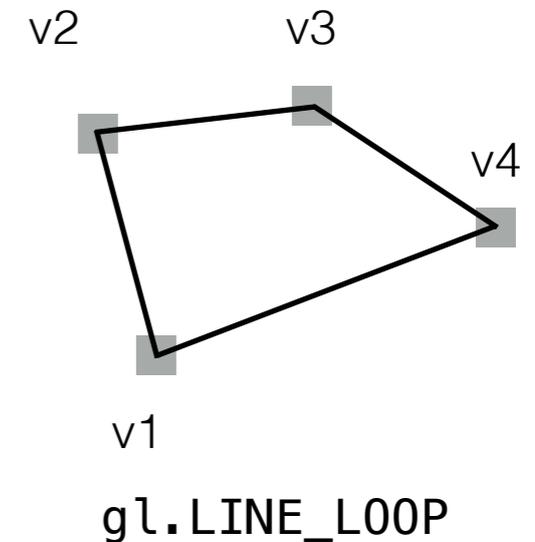
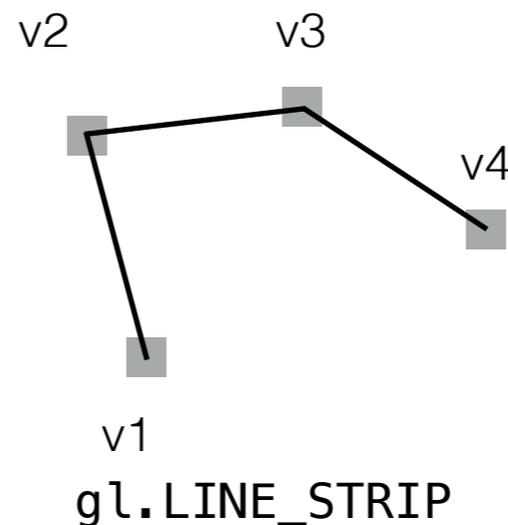
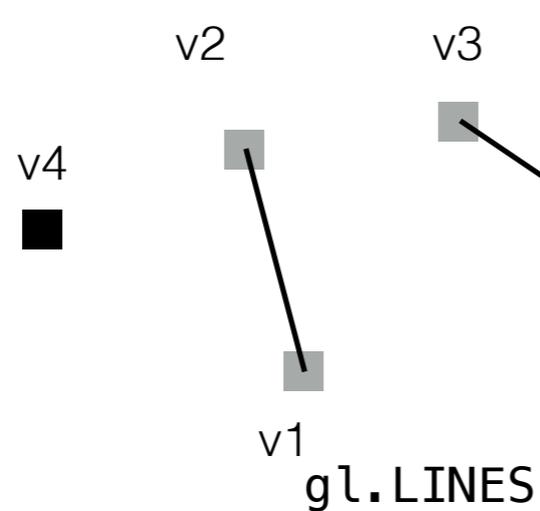
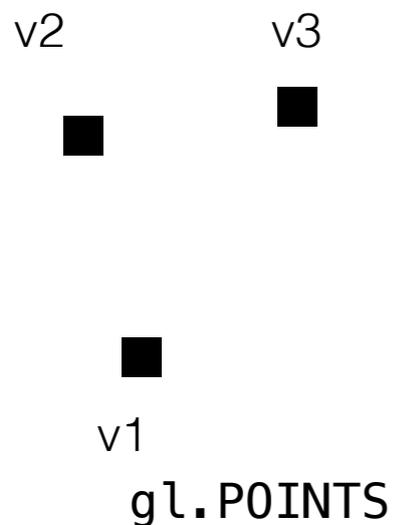
WebGL Primitivas

2019-2020
Fernando Birra

Objetivos

- Conhecer as primitivas WebGL
- Converter polígonos em triângulos
- Interpolação de atributos durante o varrimento

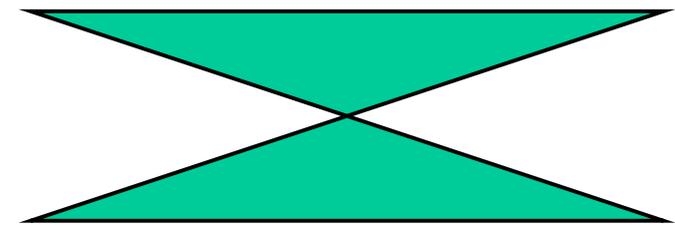
Primitivas WebGL



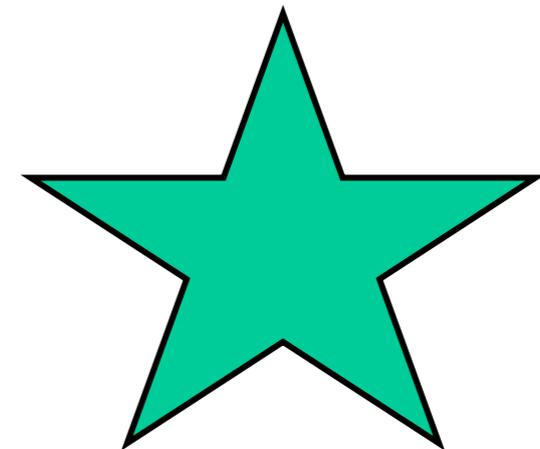
Polígonos (>3 lados)

- WebGL apenas suporta triângulos porque:
 - são simples: as arestas não se cruzam
 - são convexos: todos os pontos num segmento de reta entre dois pontos no polígono também pertencem ao polígono
 - são planos: todos os vértices (e pontos no seu interior) pertencem ao mesmo plano
- As aplicações são obrigadas a decompor (*tessellation*) polígonos em triângulos (triangulação)
- OpenGL 4.1 inclui um *tessellator*, mas não o WebGL

Exemplos de polígonos:



não simples



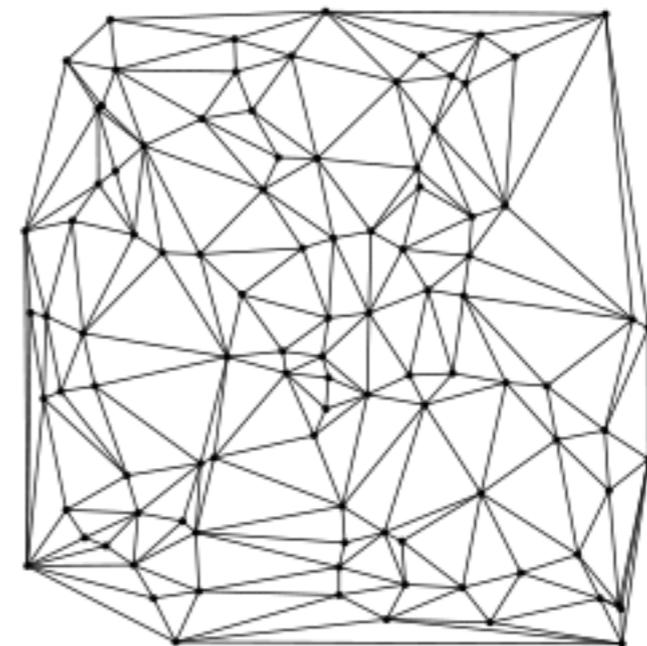
não convexo (côncavo)

Teste de Polígonos

- Conceptualmente é fácil testar um polígono quanto à simplicidade e convexidade...
 - ... mas é dispendioso
 - versões anteriores assumiam simplicidade e convexidade e deixavam o teste para a aplicação
- a versão atual apenas desenha triângulos
- é necessário um algoritmo para triangular um polígono arbitrário

Bons e maus triângulos

- Triângulos longos e finos podem gerar artefactos
- Triângulos equiláteros são os mais adequados
- objetivo: maximizar o ângulo mínimo
- Delaunay triangulation: para triangular uma malha de pontos não estruturados



Delaunay triangulation

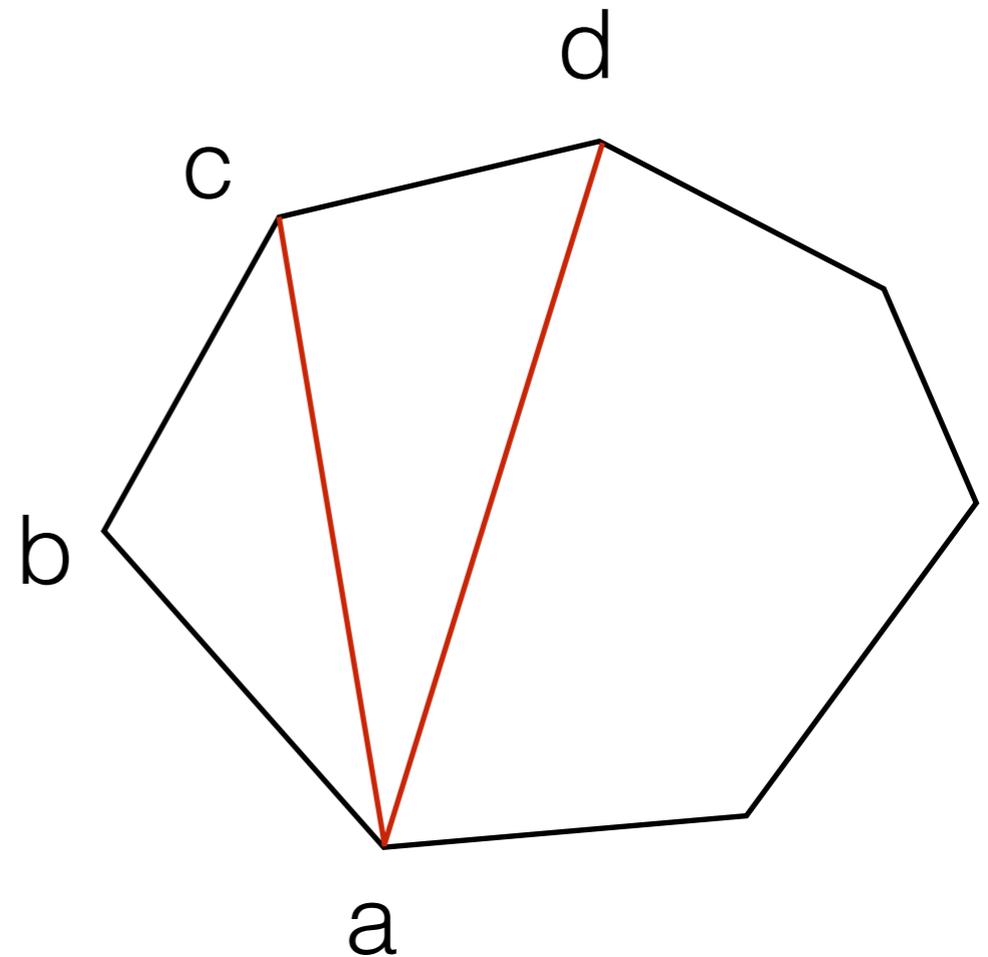
Triangulação de polígonos convexos

Dado um polígono convexo:

Gerar os triângulos: $[a,b,c]$, $[a,c,d]$, ...

Algoritmo

1. Escolhe-se um vértice pivot, neste caso **a**.
2. Formam-se os triângulos varrendo os vértices no sentido horário ou no seu inverso, usando sempre a e o último vértice do triângulo anterior.



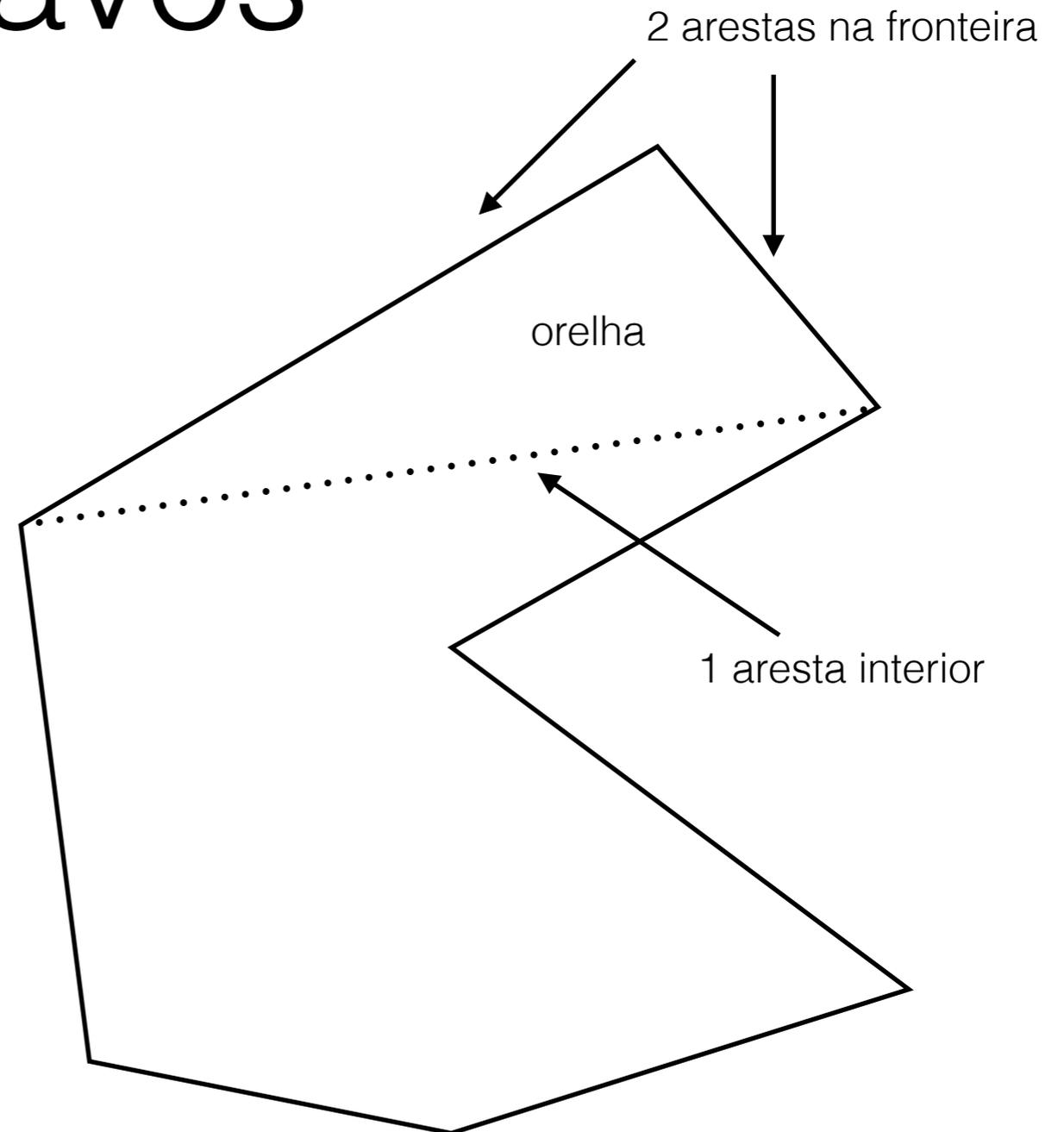
Nota: Pode implementar-se em WebGL usando a primitiva `gl.TRIANGLE_FAN`

Triangulação de polígonos côncavos

Pode aplicar-se o algoritmo *earcut*.
Os polígonos simples (sem buracos) e
côncavos possuem pelo menos 2 orelhas.

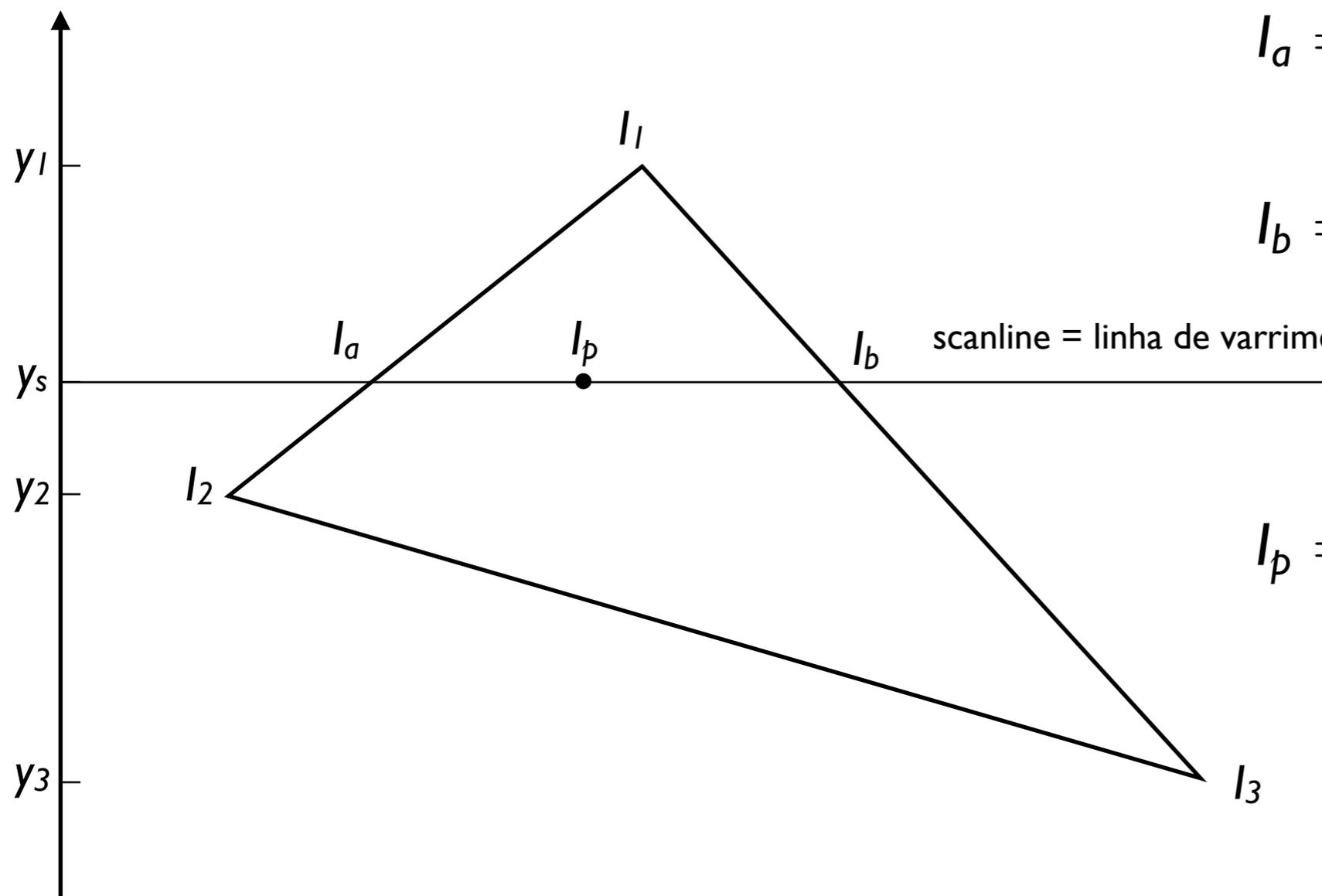
Uma orelha é um triângulo formado por
duas arestas consecutivas (do polígono
original), estando a aresta que fecha o
triângulo completamente no interior do
polígono.

- Encontrar uma das orelhas e usá-la para
decompor o polígono.
- Repetir o passo anterior ao polígono restante
da remoção da orelha.



Interpolação de atributos

Componentes de Cores/Intensidades



$$I_a = I_2 + (I_1 - I_2) \frac{y_s - y_2}{y_1 - y_2}$$

$$I_b = I_3 + (I_1 - I_3) \frac{y_s - y_3}{y_1 - y_3}$$

$$I_p = I_a + (I_b - I_a) \frac{x_p - x_a}{x_b - x_a}$$

scanline = linha de varrimento

Interpolação de atributos

Componentes de Cores/Intensidades

