

Zバッファ 法

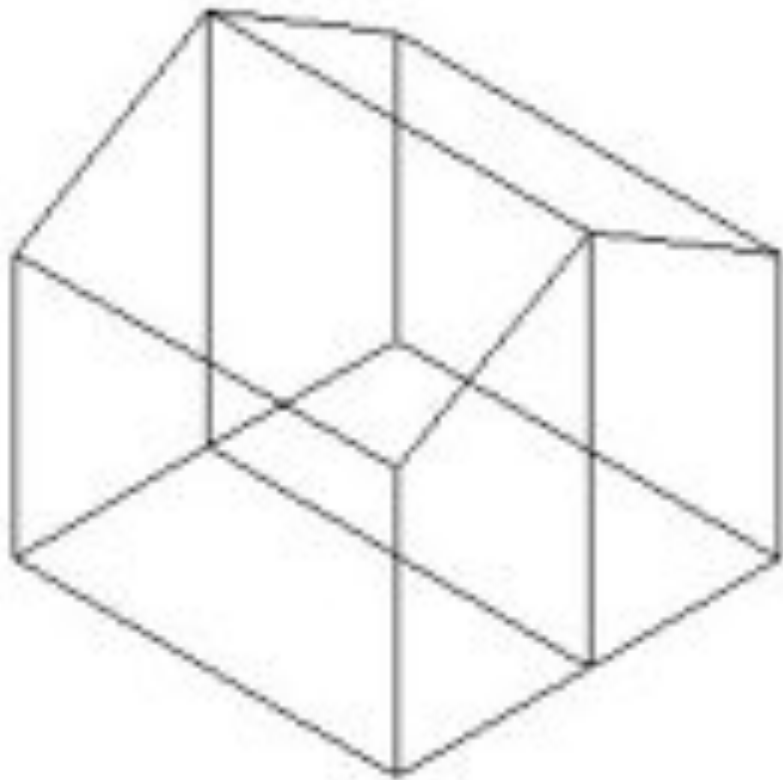
「3次元CGの基礎と応用」より

隠面処理

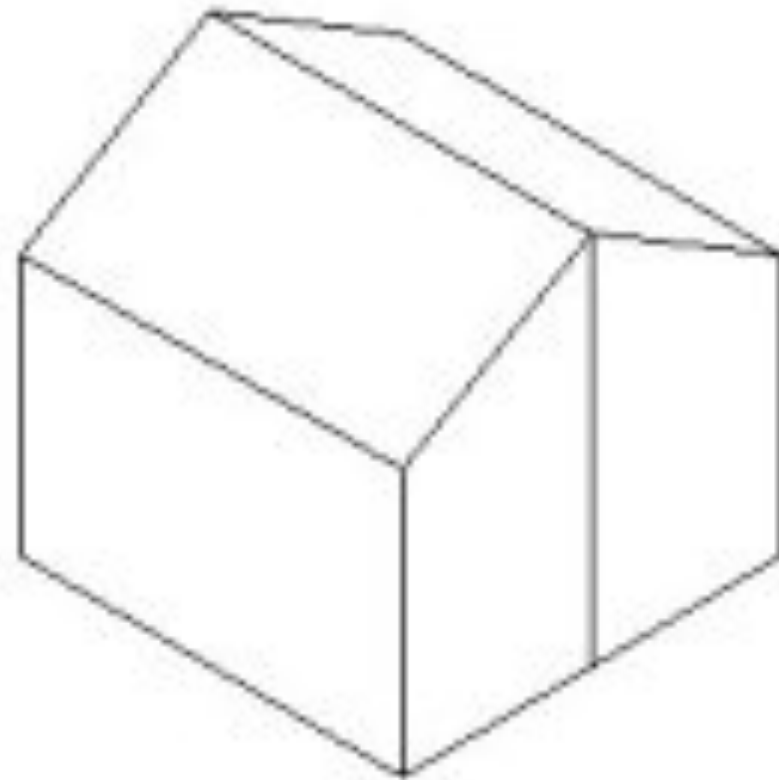
- ・ 見えてはいけない線や面を見えないように処理すること
- ・ ポリゴンではなくてラインでグラフィックを描画していた時代は「陰線処理」とも呼んでいた。

隠面処理の例

処理なし



処理あり

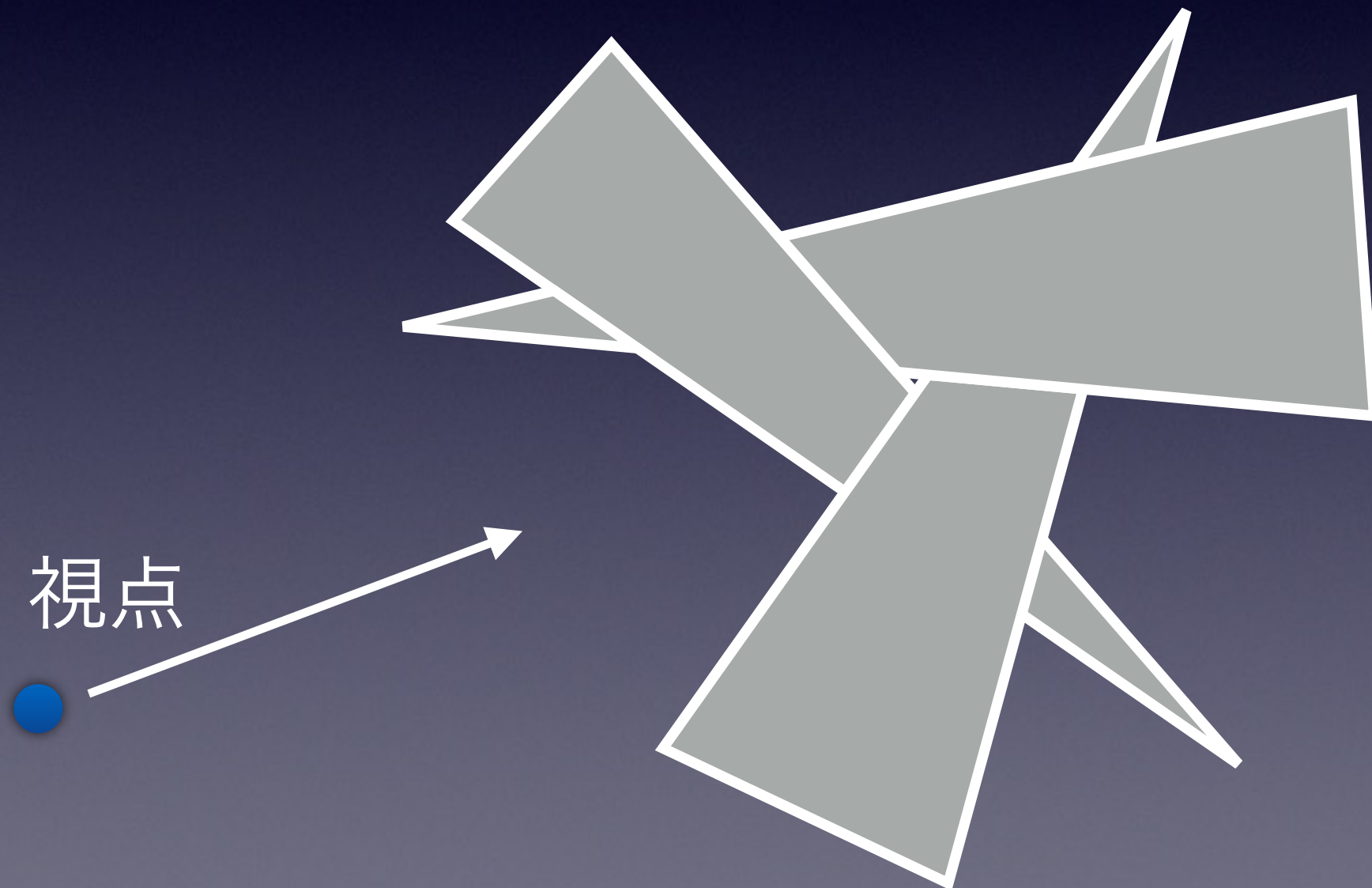


隠面処理のアルゴリズム

- ・ 後面除去
- ・ ペインタアルゴリズム
- ・ スキャンライン法
- ・ Zバッファ法 ← 今回はこの手法を紹介

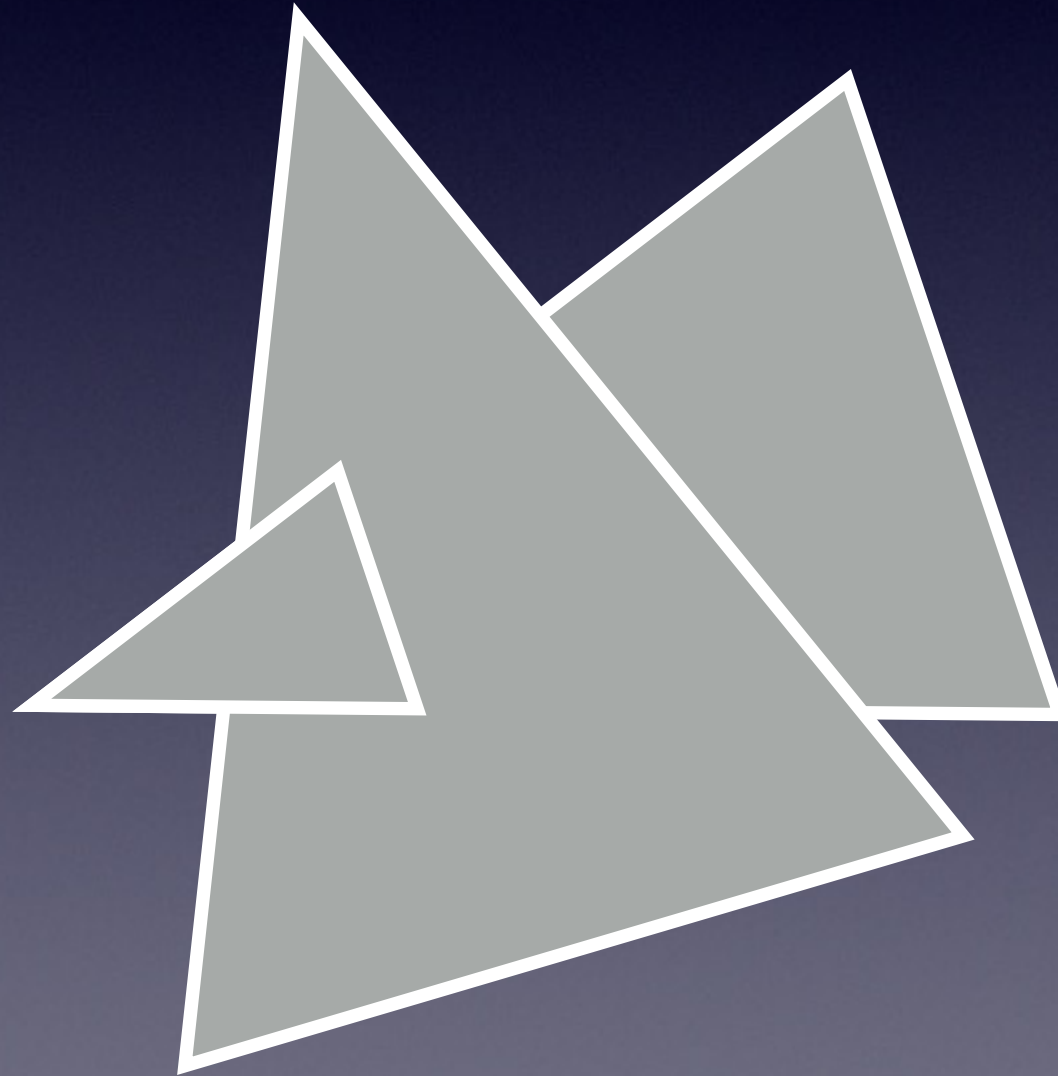
失敗する例 (a)

3すくみの場合



失敗する例 (b)

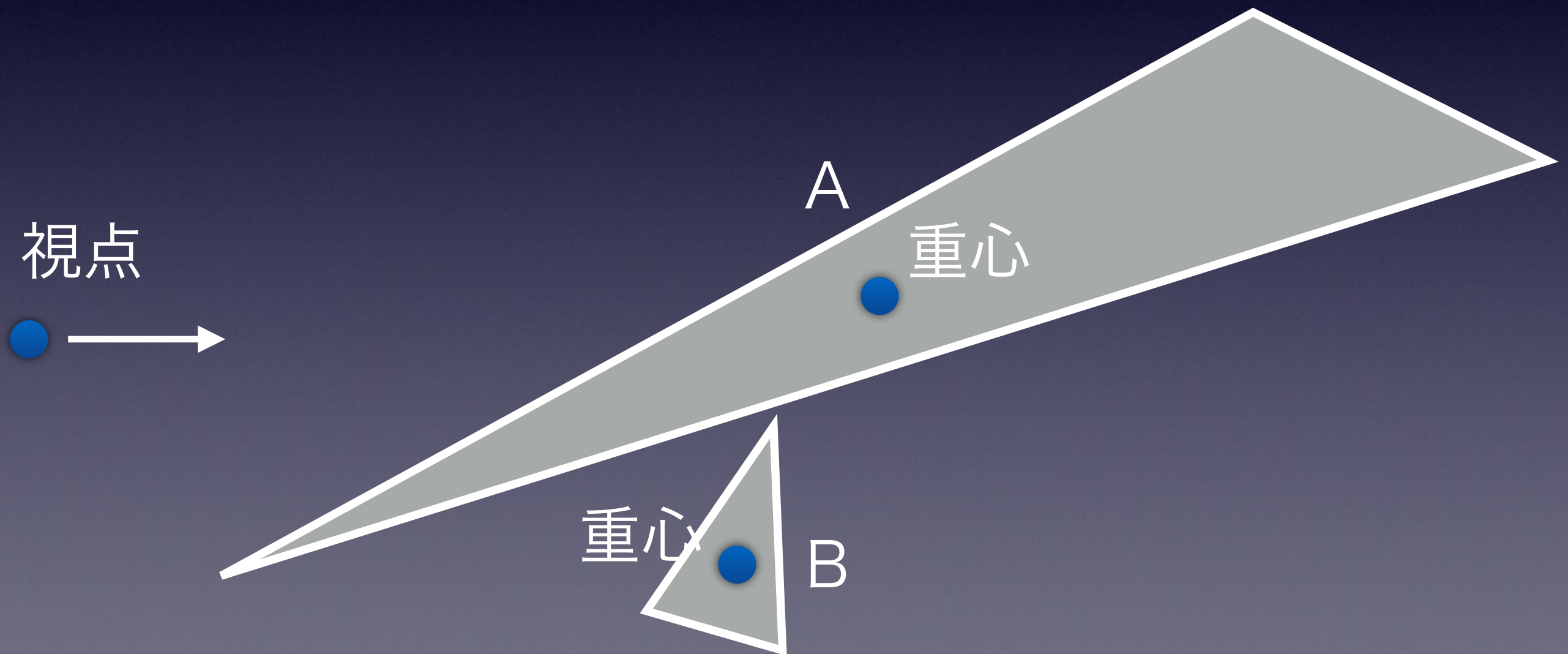
3 角形が貫いている場合



失敗する例 (c)

奥の3角形を手前に表示する例

→ 重心が奥のAが手前のBを覆っている



Zバッファ法

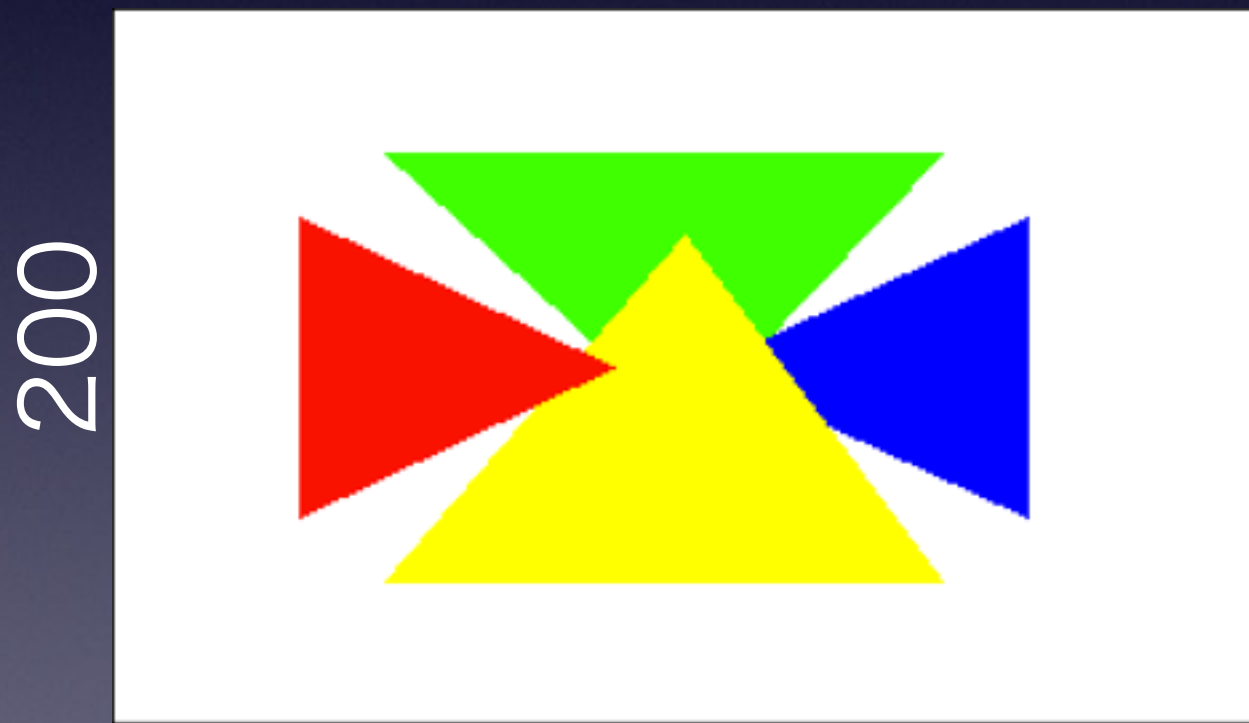
- ・ 3次元の隠面消去問題を1次元の陰点消去問題に帰着させる方法
- ・ 「Zバッファ」と呼ばれる、描画エリアと同じ大きさを持つ、画素の奥行きを保存するメモリを使用する。

Zバッファ

描画エリアと同じ大きさを持つ
画素の奥行きを保存するメモリ

320

320



描画エリア



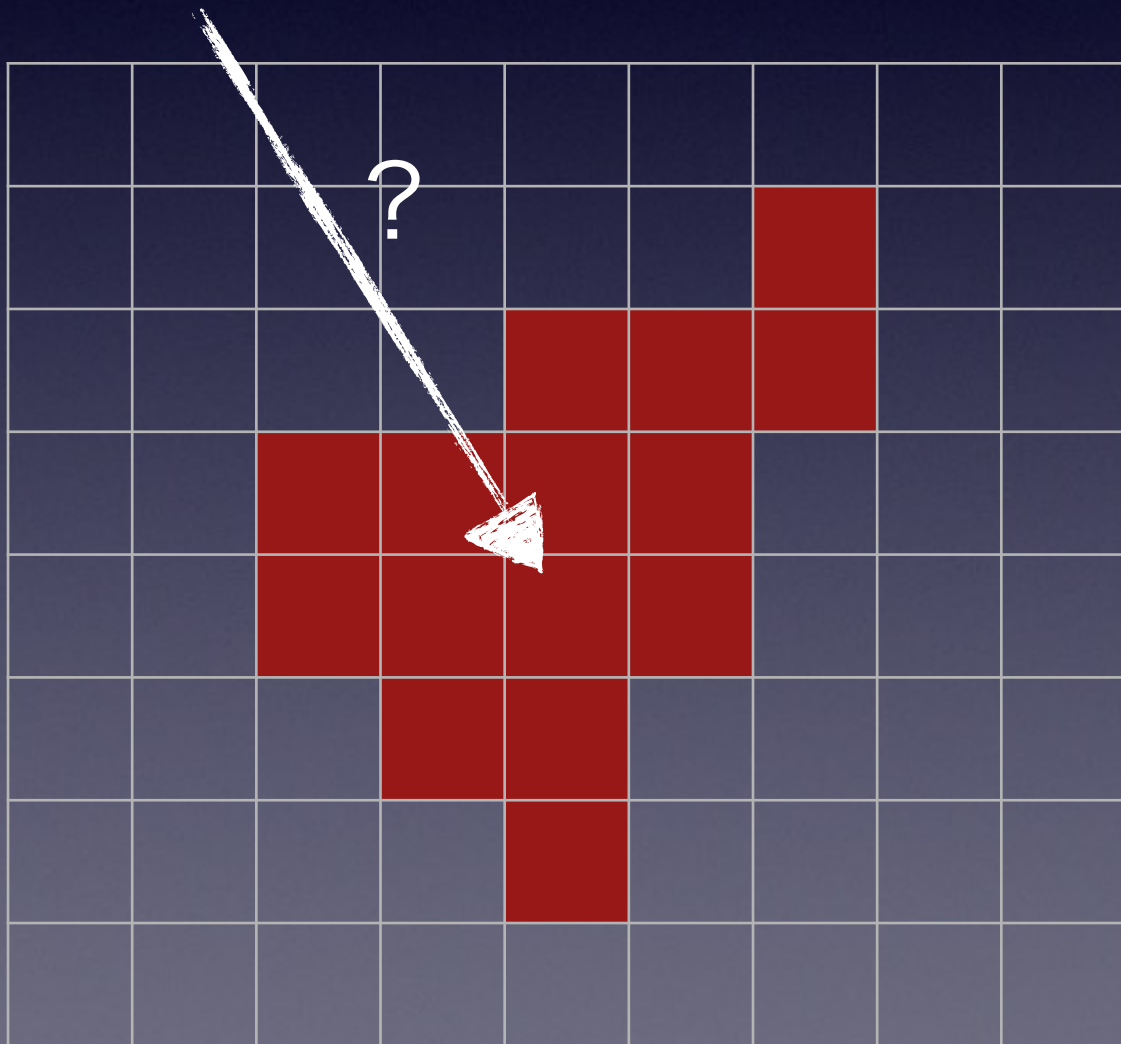
Zバッファ

(今回のプログラム内では配列で確保)

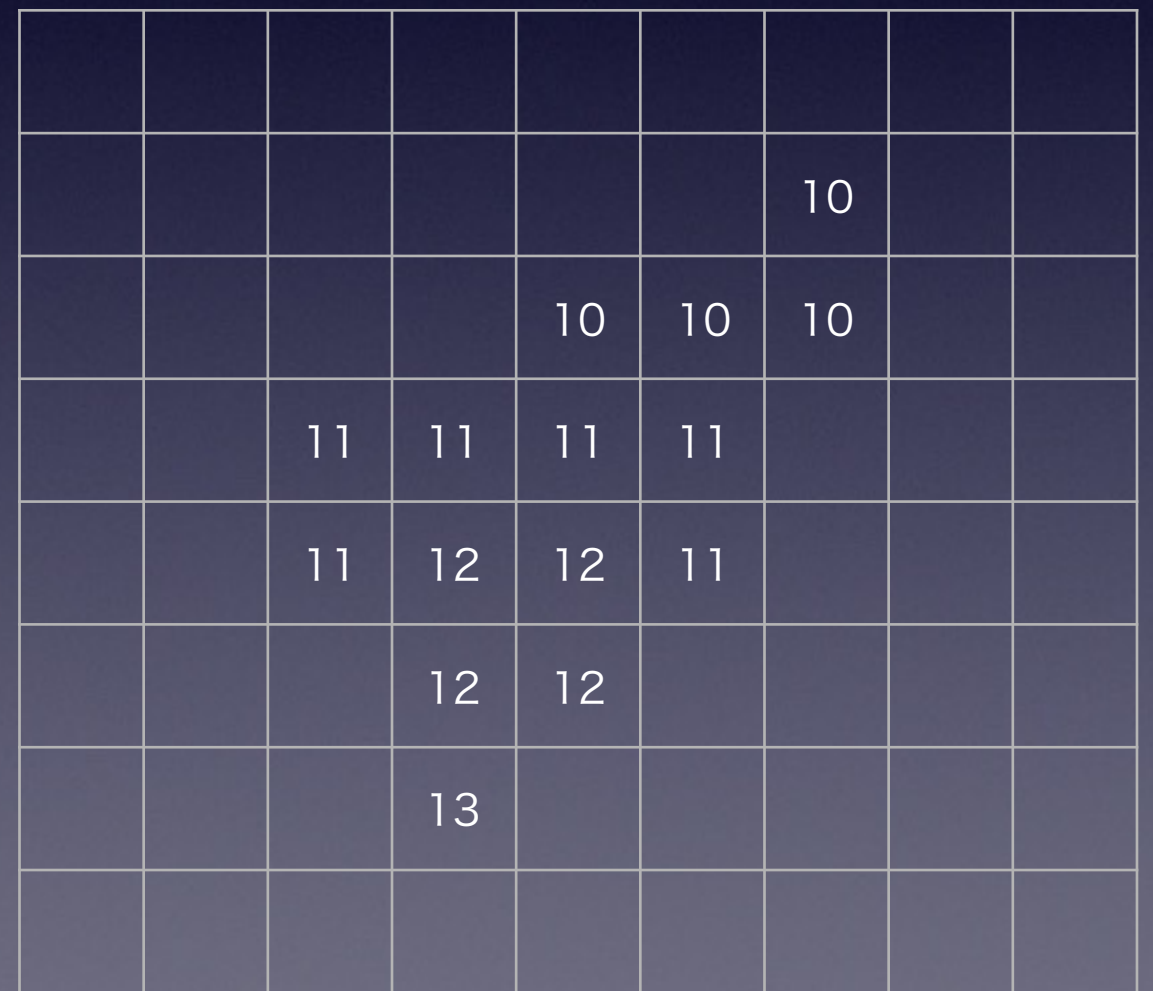
一次元の隠点消去問題

点を描画するときに、すでに書かれた
点があれば、その奥行きを比較する

10



描画エリア

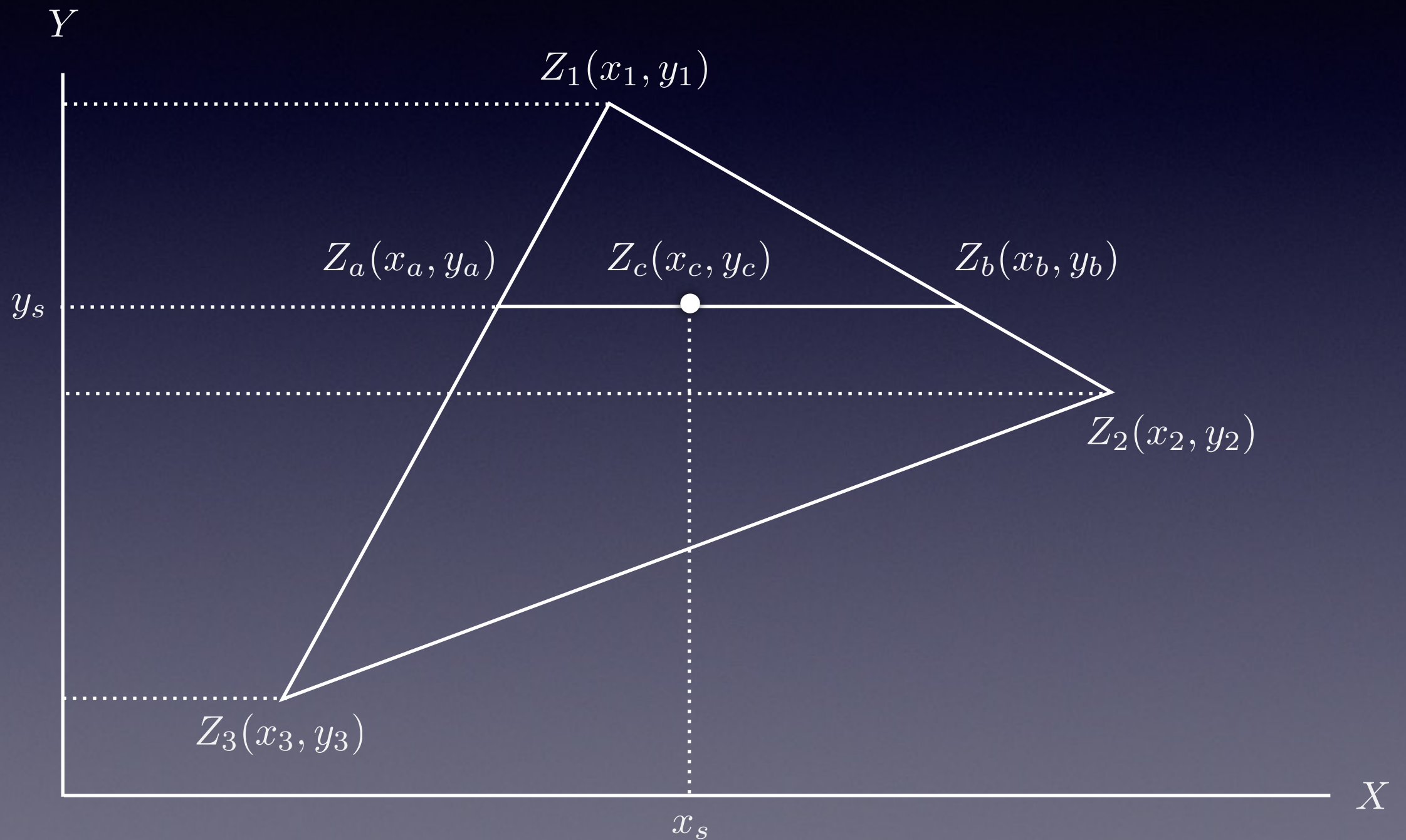


Zバッファ

3 角形面内部の奥行きの計算

- ・ Gouraudシェーディングの色や、Phongのスムーズシェーディングの法線と同じく、頂点の奥行き（Z軸の値）を保存して、線形補間で求める。

3 角形面内部の奥行き計算

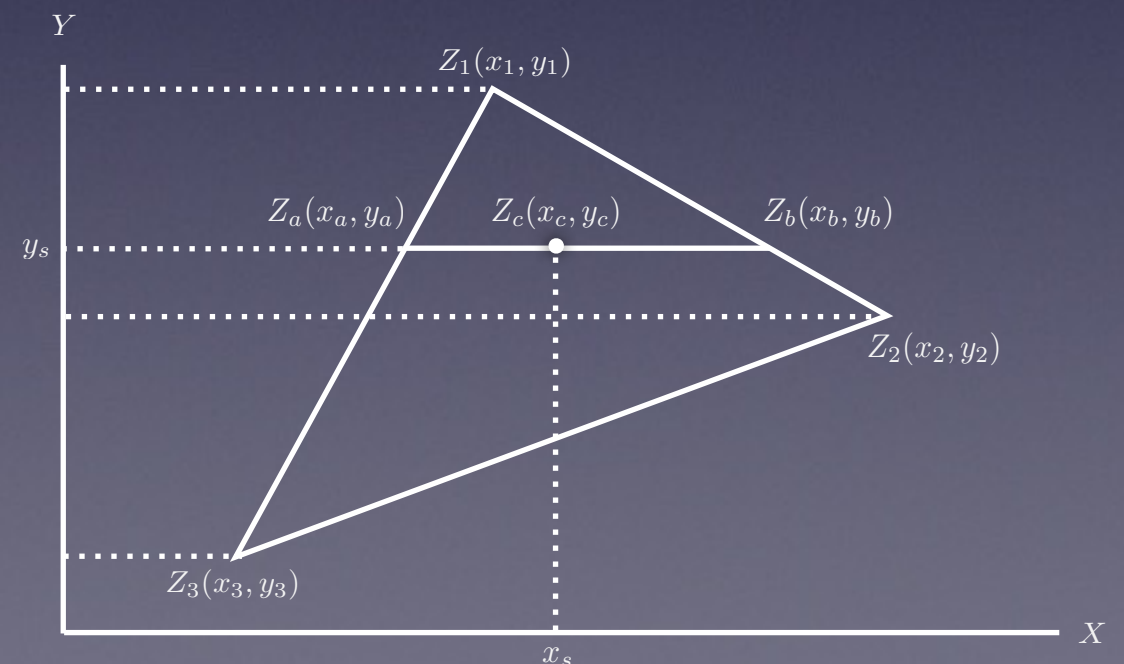


3 角形面内部の奥行きの計算

$$Z_a = \frac{1}{y_1 - y_2} (Z_1(y_s - y_2) + Z_2(y_1 - y_s))$$

$$Z_b = \frac{1}{y_1 - y_3} (Z_1(y_s - y_3) + Z_3(y_3 - y_s))$$

$$Z_c = \frac{1}{x_b - x_a} (Z_a(x_b - x_s) + Z_b(x_s - x_a))$$



特徴

- ・ Zバッファ分のデータをメモリ上に保持する必要があり、メモリ使用量は多い
- ・ 3 角形ごとに並列処理が可能なので、ハードウェア化がしやすい
- ・ 奥行き情報で計算誤差が発生する可能性がある

ソースコード

- <https://github.com/nakaken0629/3dstudy2>