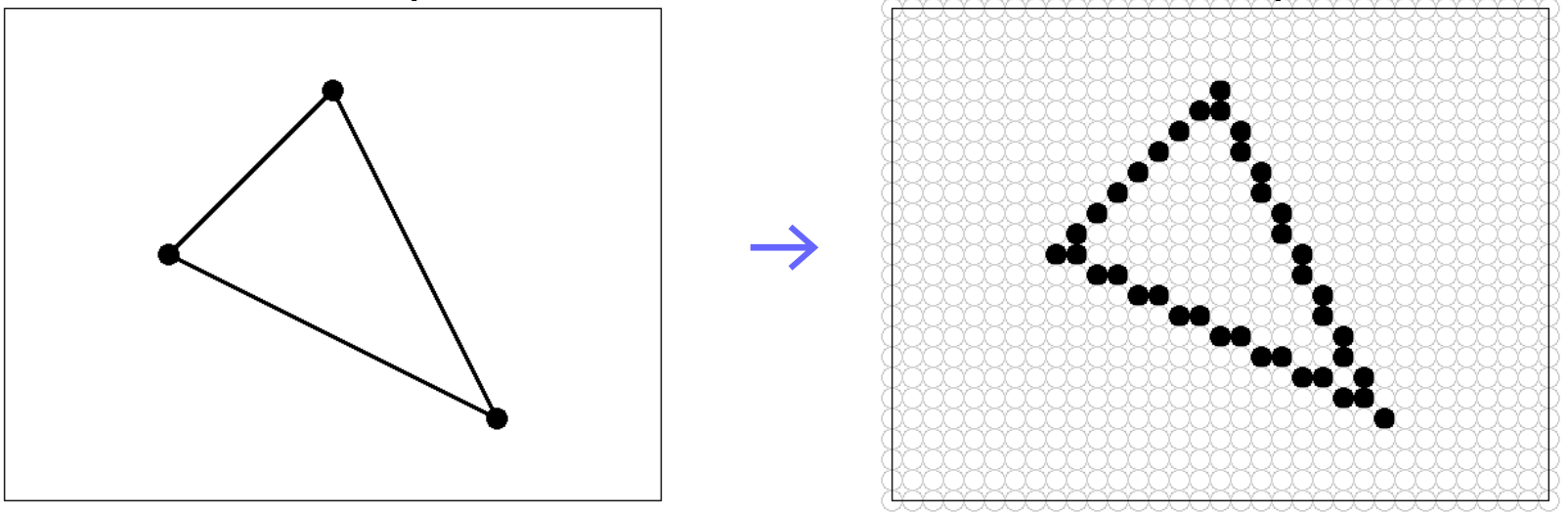
ラスタグラフィックス
(raster graphics)

- 2D デジタル画像の生成
  - 現状，3D-CG といっても実際は 2D 表示
    - 投影変換：3D ポリゴン → 2D 三角形
  - 離散化：画像 → 画素の集合
    - 実数座標（連続線） → 整数座標（離散点）
    - 走査変換（scan conversion，rasterize）



画像生成の効率（速度）

- ポリゴン描画速度
  - 高いほど良い：微細な形状，滑らかな動作の表現
    - 数M ポリゴン／秒　（プレステ2）
    - 数G ポリゴン／秒　（プレステ3）
    - 数10G ポリゴン／秒（プレステ4）
      - （1キャラではなく，1シーン全体の処理速度）
      - （描画だけでなく，座標変換などの処理も含む）
- 高速描画アルゴリズム
  - 目的：計算量を削減 → 品質（ポリゴン数）を向上
    - 「チリも積もれば山となる」
      - チリ×ギガ＝あなどれない→少しの工夫で大きな効果

高速化のポイント(1)

- インクリメンタルな計算
  - 差分を活用して反復計算
    - 漸化式：$f(x) = f(x-1) + \Delta$
  - 整数の加減算のみ使用
    - コンピュータが最も得意とする演算
    - （ま，最近の CPU では実数演算も高速だけどね...）
  - 実数と乗除算は不使用
    - 例外：2^n 倍は OK．シフト演算で高速計算可能.

高速化のポイント(2)

- 無駄な計算を排除
  - 当たり前？意識しないと気づかない!
  - 特に，同じ計算を何度も実行しないこと!!
    - sin(x)*sin(x)
    - ➔　y=sin(x), y*y
    - for(i=0;i<strlen(s);i++) { ... }
    - ➔　n=strlen(s); for(i=0;i<n;i++) { ... }

高速描画アルゴリズム

- 線分，円，多角形，等の高速描画
  - DDA（Digital Differential Analyzer）
    - デジタル微分解析器
    - 直線の方程式 $y(x) = mx$ →差分化 $y(x+1) = y(x) + m$
  - Bresenham のアルゴリズム
    - DDA の差分方程式→整数化　…と結果は同じ
      - アプローチは異なる
    - 円への応用も可能

教科書 pp.45-47