実験2日目

「逆透視変換」,
inverse perspective mapping

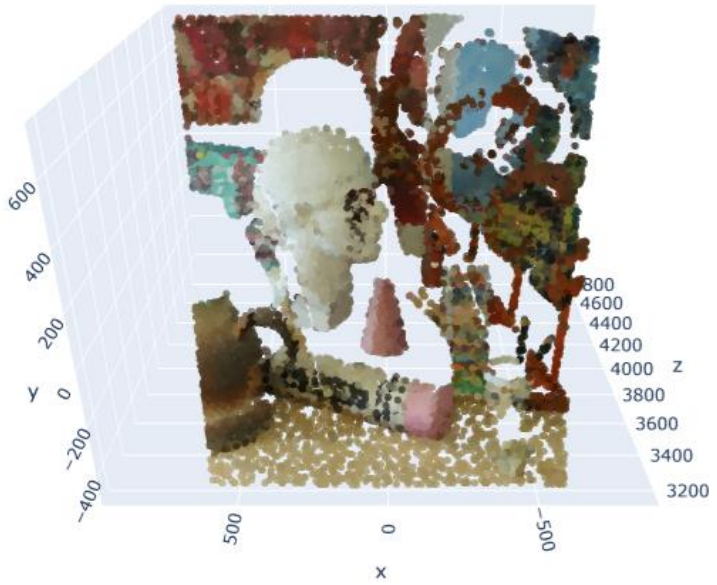
「ポイントクラウド」,
point cloud

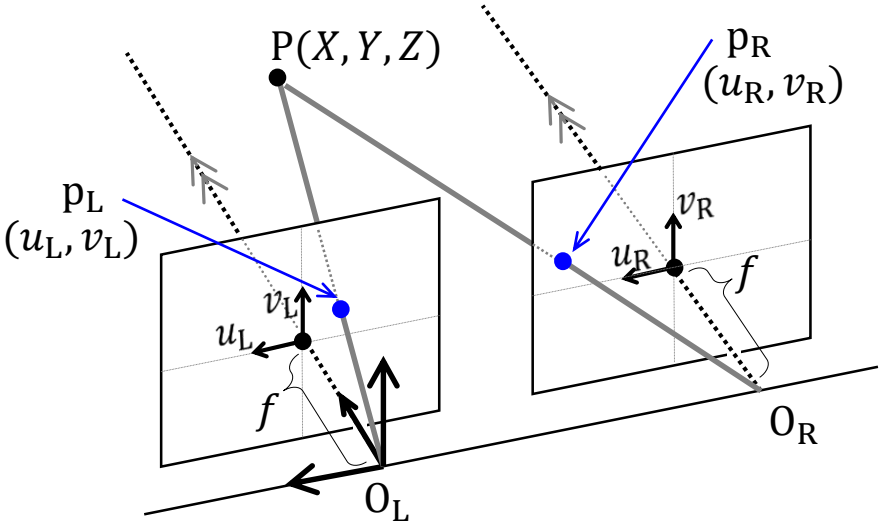
「RANSAC」
random sample consensus

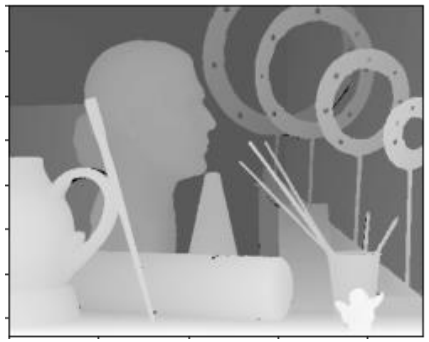
これらについて, What? / Why? / How? の
あらゆる疑問を解消する文書(実験レポート)を期待しています.

3次元計測

✓ 3次元化する

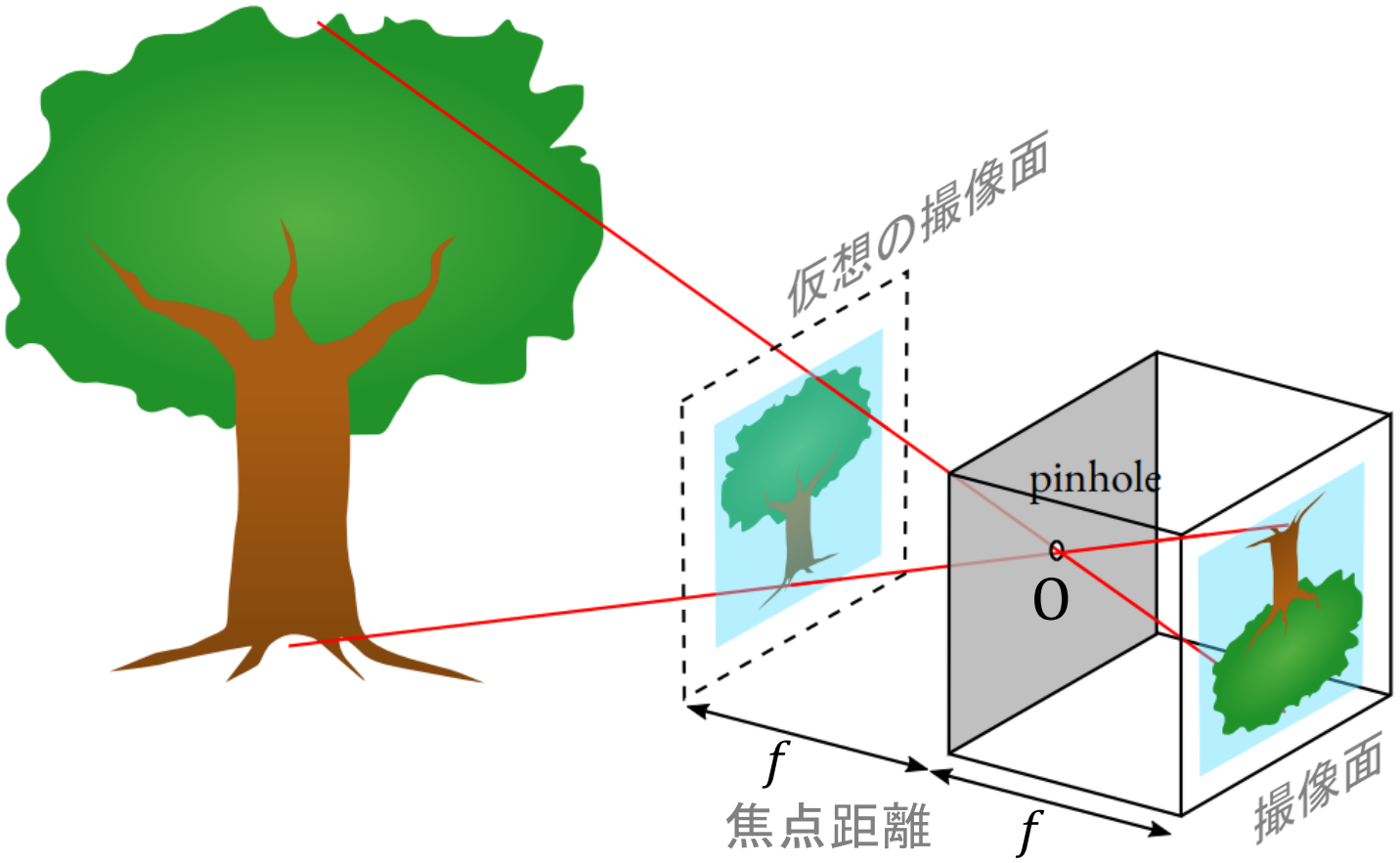




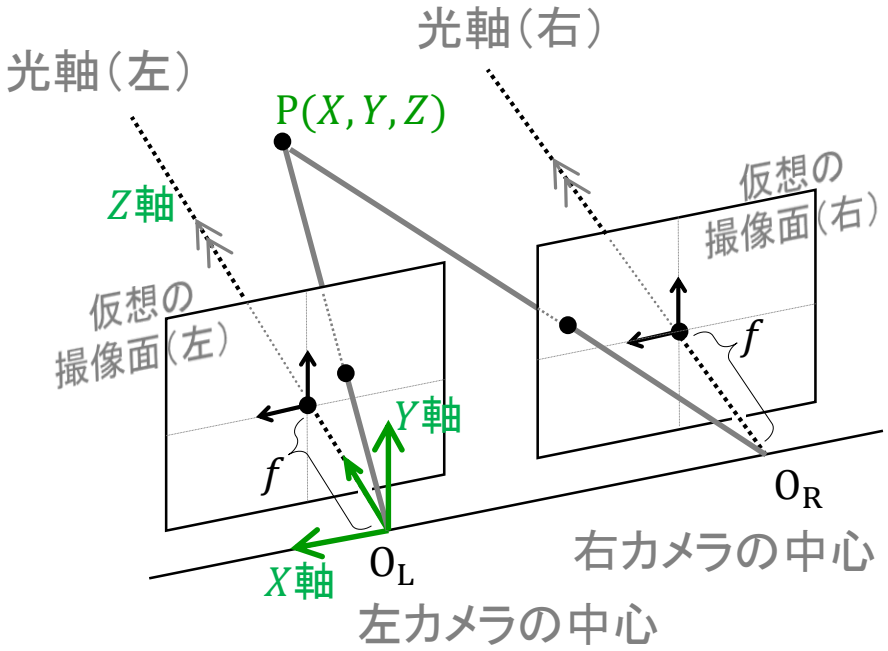


「視差」を測る
(ブロックマッチング)

画像とカメラの座標系



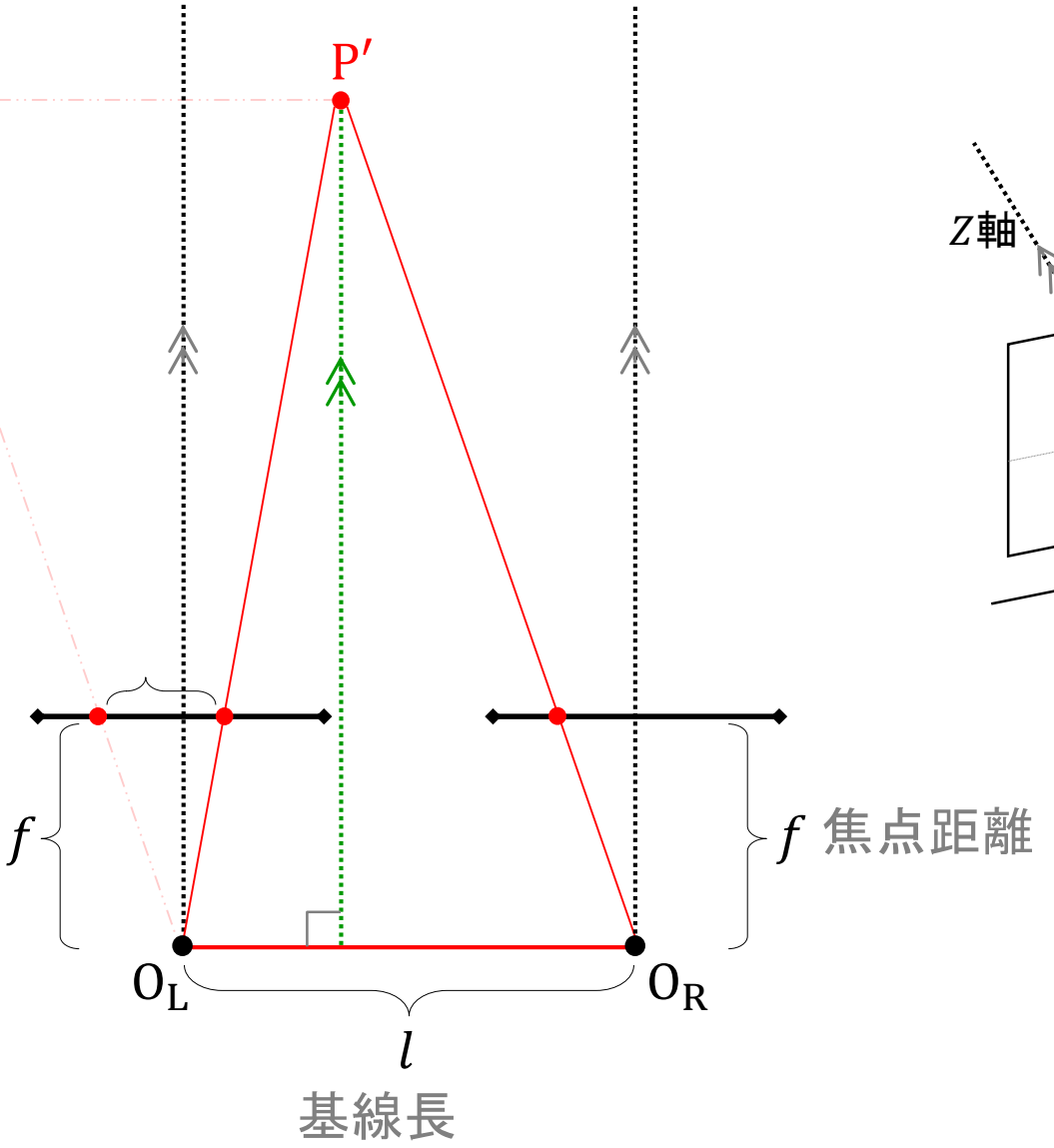
透視投影モデル

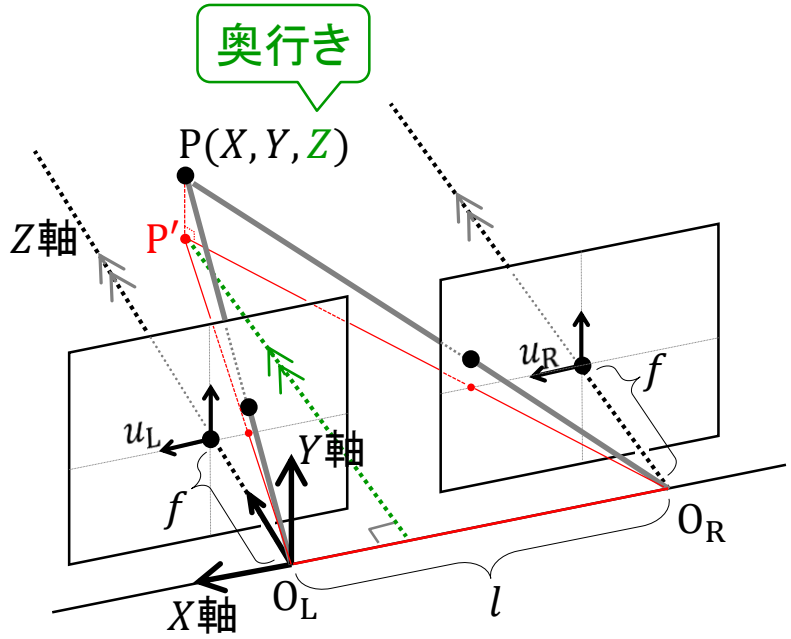


2台のカメラを
平行に設置した場合

視差 d と奥行き Z の関係

Z =【視差d，焦点距離f，基線長l で表せ】

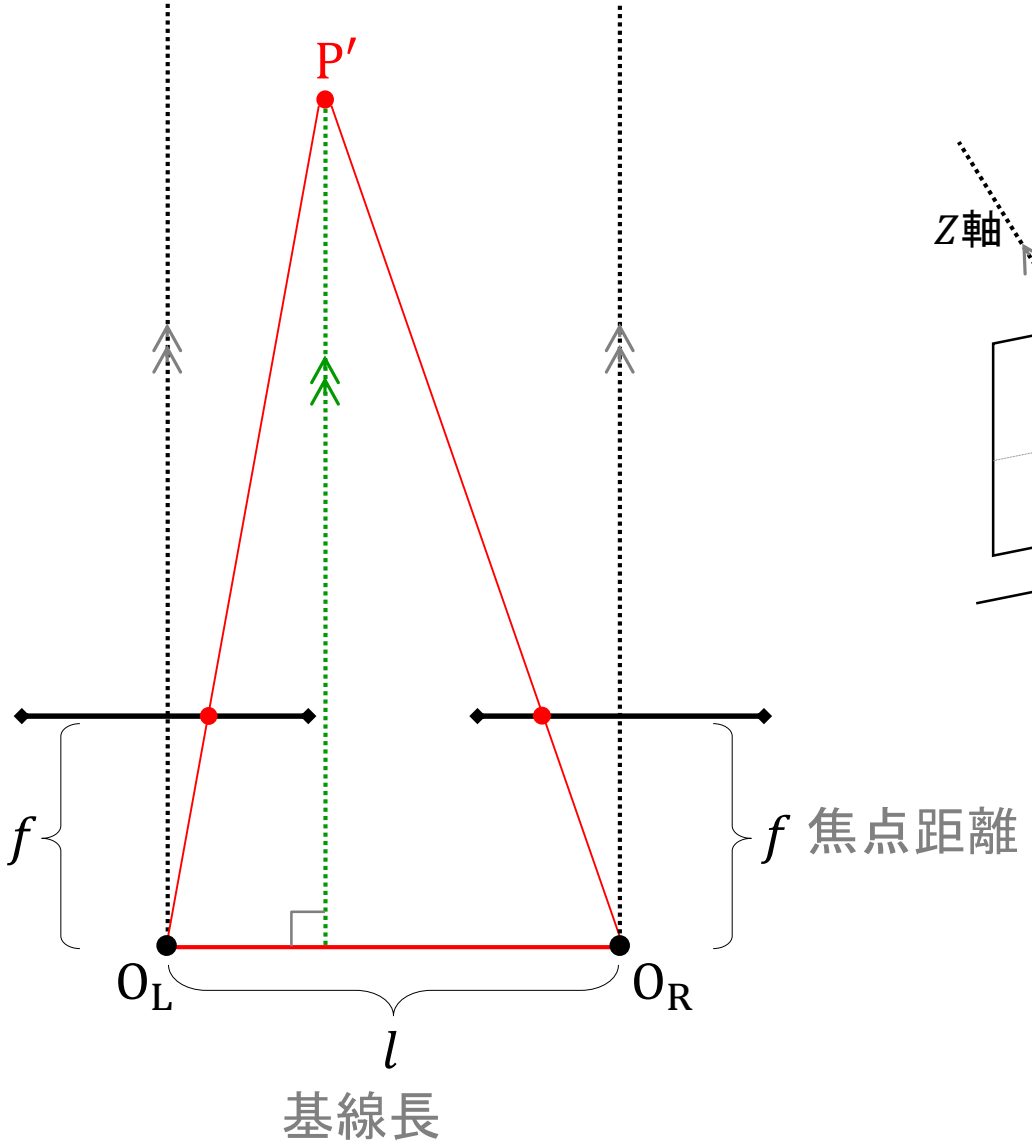


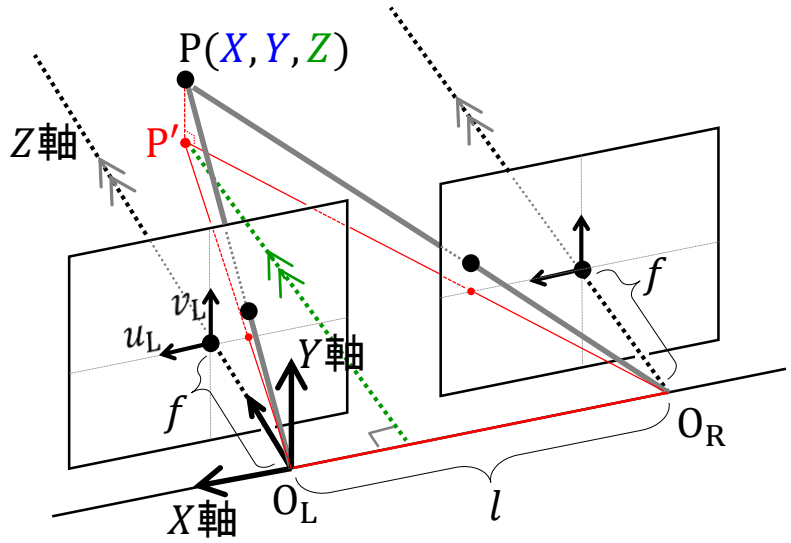


視差 $d = |u_R - u_L|$

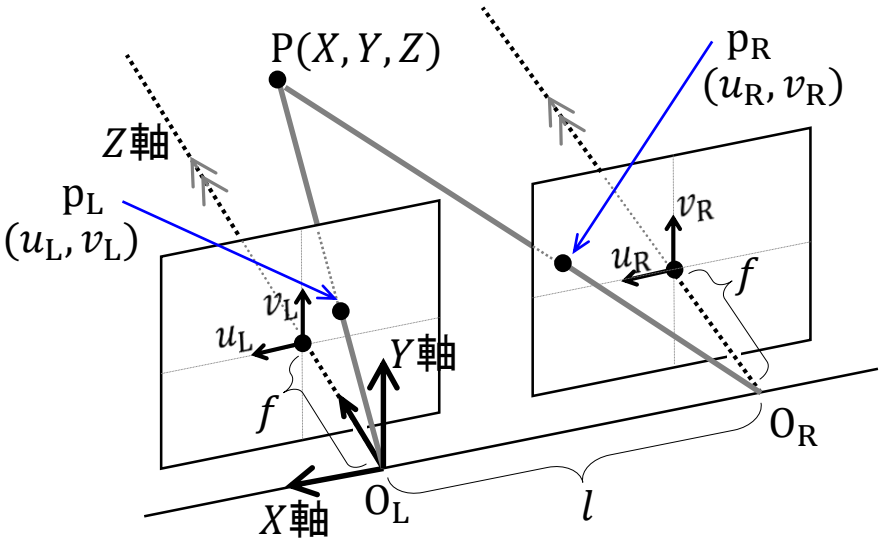
X と Y も計算できる

X =【 Z, u_L, f で表せ】, Y =【 Z, v_L, f で表せ】





3次元座標の計算手順



手順1　$\mathrm{p_L}(u_\mathrm{L}, v_\mathrm{L})$ に対応する $\mathrm{p_R}(u_\mathrm{R}, v_\mathrm{R})$ を見つける。

手順2　視差を測る。　　$d = |u_\mathrm{R} - u_\mathrm{L}|$

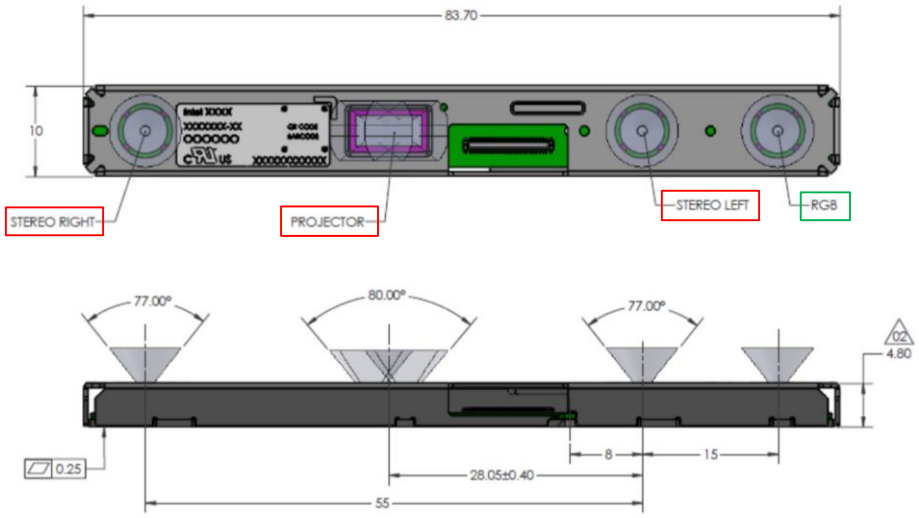
手順3　深度（奥行き）に換算する。　　$Z = \frac{f}{d} l$

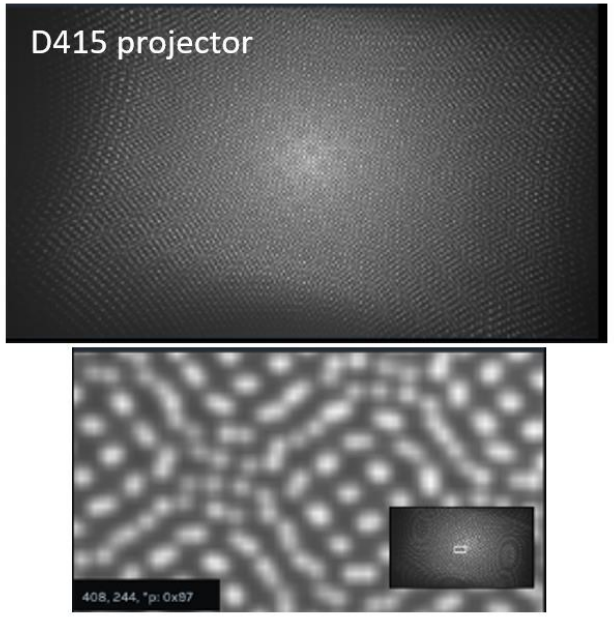
手順4　3次元座標を得る。　　$X = \frac{u_\mathrm{L}}{f} Z$,　$Y = \frac{v_\mathrm{L}}{f} Z$　　$(X, Y, Z) = \left(\frac{u_\mathrm{L}}{d} l, \frac{v_\mathrm{L}}{d} l, \frac{f}{d} l\right)$

RGB-Dカメラを使ってみよう

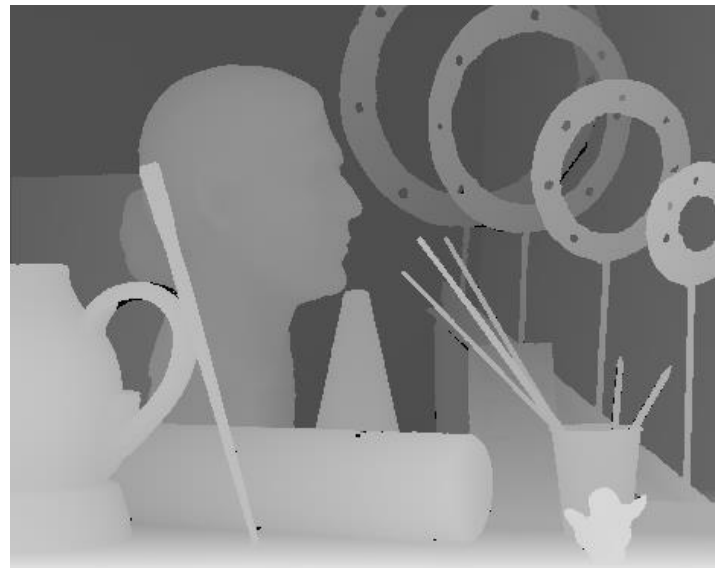
カラー画像（RGB）

深度画像（Depth）





深度から3次元座標を計算する



視差画像

手順1　$\mathrm{p_L}(u_\mathrm{L}, v_\mathrm{L})$ に対応する $\mathrm{p_R}(u_\mathrm{R}, v_\mathrm{R})$ を見つける。

手順2　視差を測る。　$d = |u_\mathrm{R} - u_\mathrm{L}|$

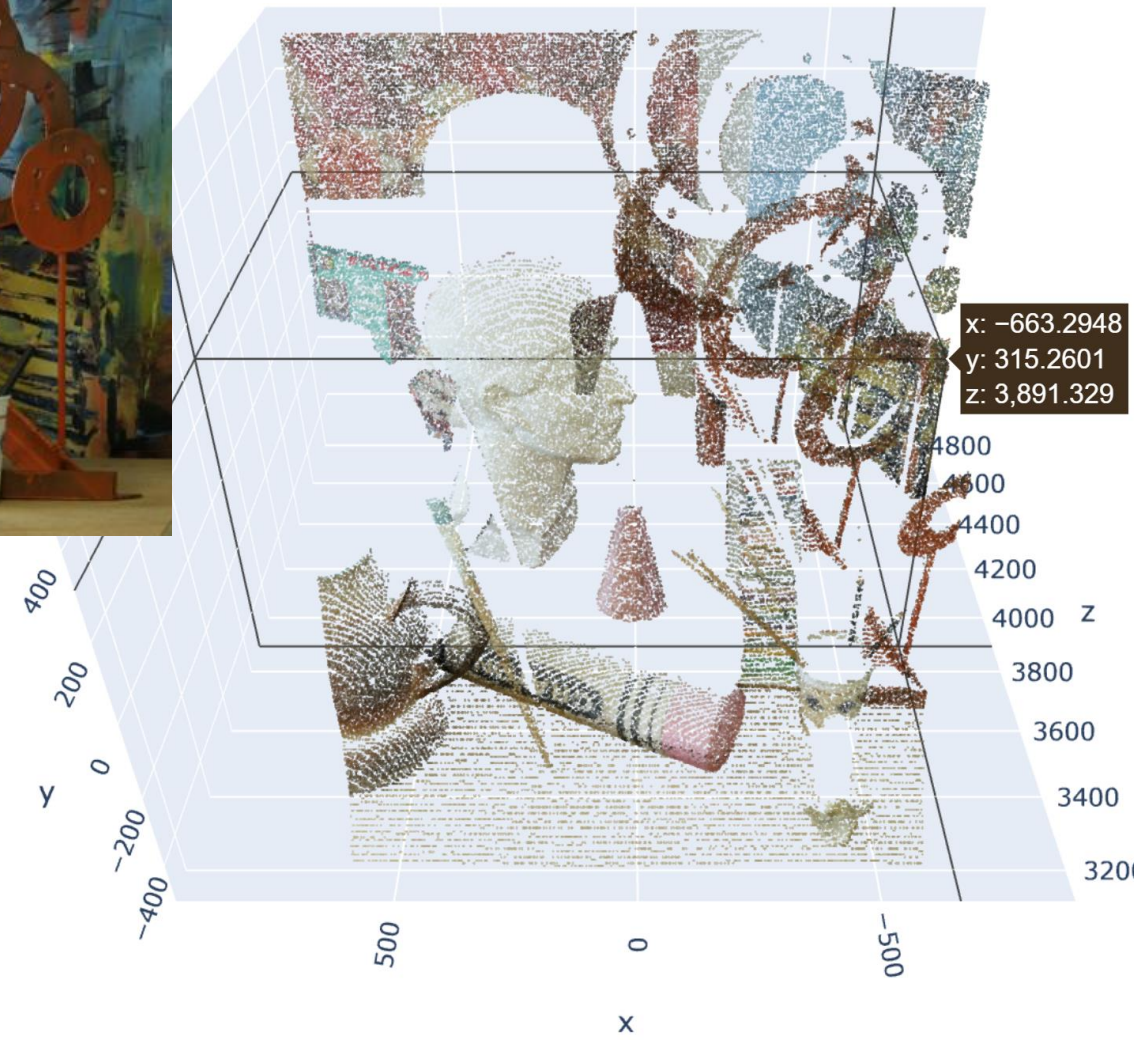
手順3　深度（奥行き）に換算する。　$Z = \frac{f}{d} l$

手順4　3次元座標を得る。　$X = \frac{u_\mathrm{L}}{f} Z$,　$Y = \frac{v_\mathrm{L}}{f} Z$

手順1～3の処理を経た
深度画像 $Z(u_\mathrm{L}, v_\mathrm{L})$ を取得する.

3次元座標を たくさん調べたら・・・！

点群（point cloud）



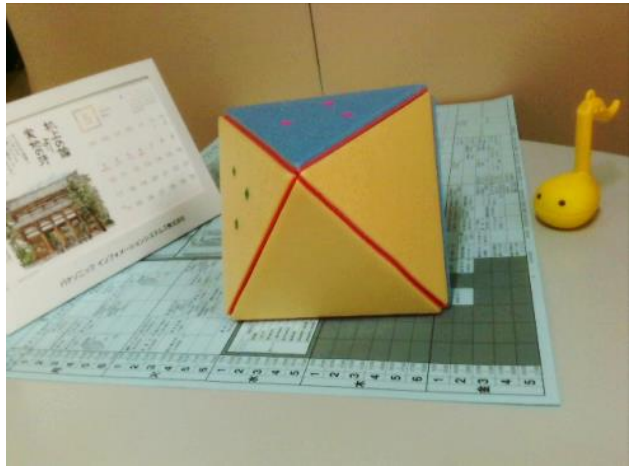
手順1　$\mathrm{p_L}(u_\mathrm{L}, v_\mathrm{L})$ に対応する $\mathrm{p_R}(u_\mathrm{R}, v_\mathrm{R})$ を見つける。

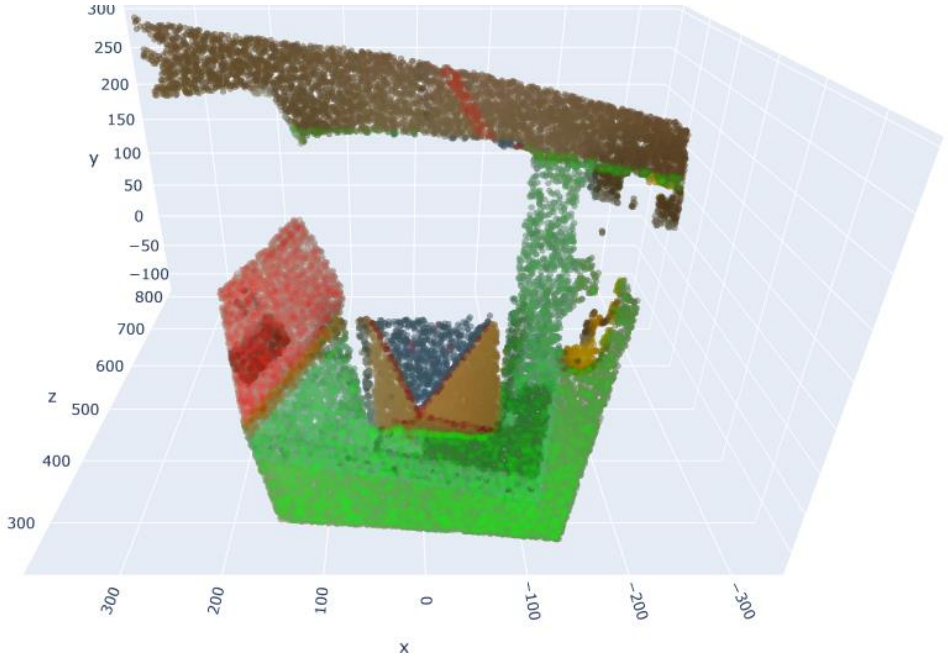
手順2　視差を測る。　　$d = |u_\mathrm{R} - u_\mathrm{L}|$

手順3　深度（奥行き）に換算する。　　$Z = \frac{f}{d} l$

手順4　3次元座標を得る。　　$X = \frac{u_\mathrm{L}}{f} Z$,　$Y = \frac{v_\mathrm{L}}{f} Z$

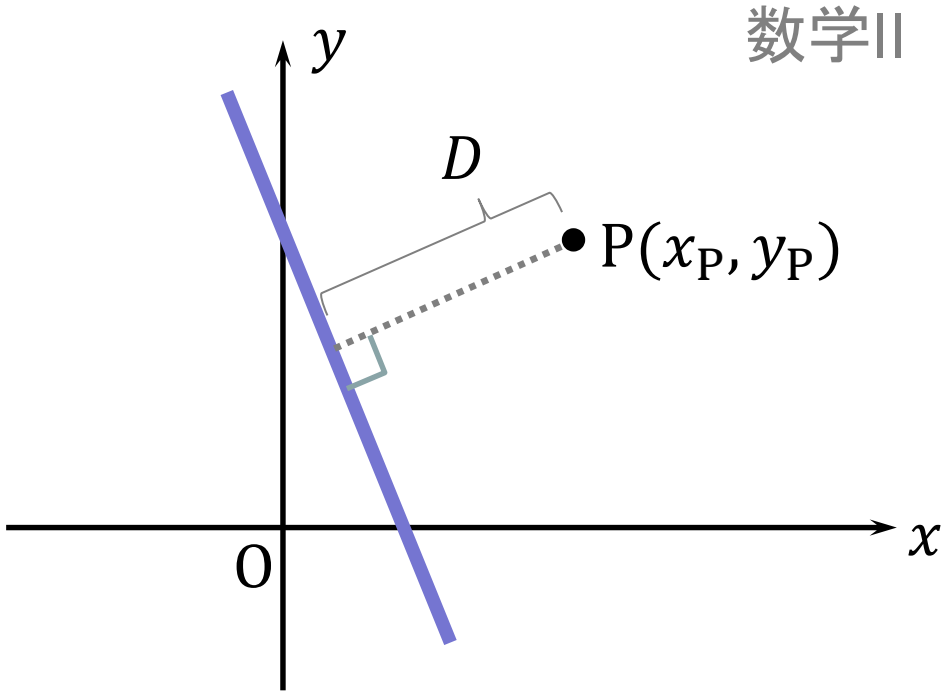
平面の自動検出




300
250
200
150
y 100
50
0
−50
−100
800
700
600
z
500
400
300
300
200
100
0
−100
−200
−300
x

点と平面の距離

数学II

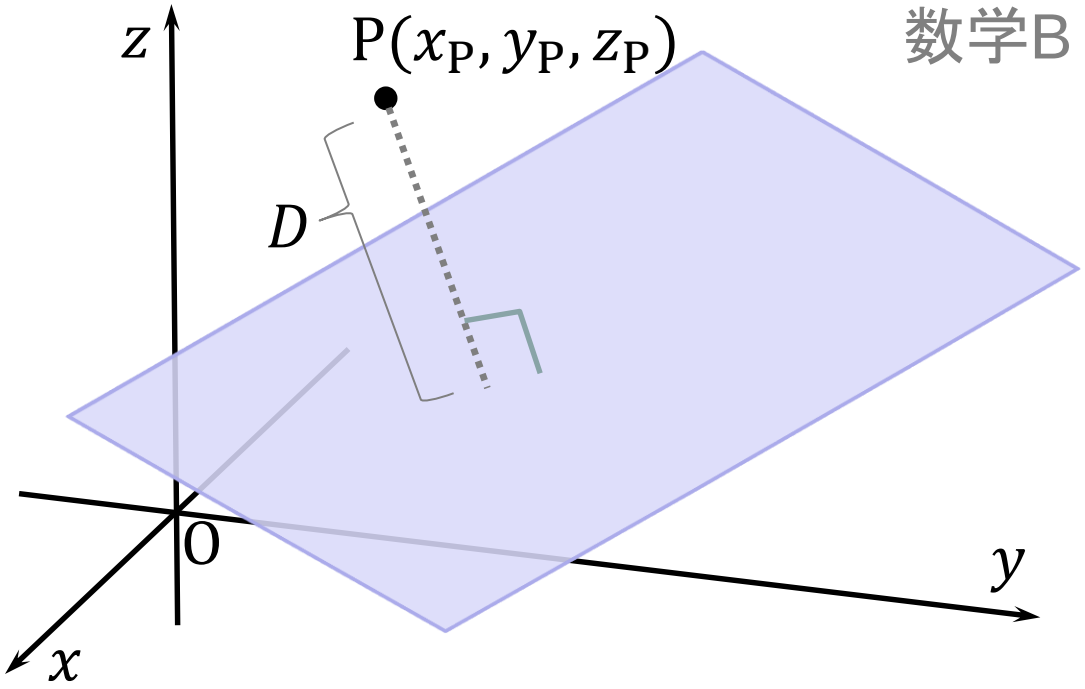


直線の方程式　$ax + by + c = 0$

点$\mathrm{P}(x_\mathrm{P}, y_\mathrm{P})$と直線の距離

$$D = \frac{|ax_\mathrm{P} + by_\mathrm{P} + c|}{\sqrt{a^2 + b^2}}$$

数学B



平面の方程式　$ax + by + cz + d = 0$

点$\mathrm{P}(x_\mathrm{P}, y_\mathrm{P}, z_\mathrm{P})$と平面の距離

$$D = \frac{|ax_\mathrm{P} + by_\mathrm{P} + cz_\mathrm{P} + d|}{\sqrt{a^2 + b^2 + c^2}}$$

平面を自動検出する

【平面を検出するRANSAC】 [F. T-Kurdi et al., 2008]

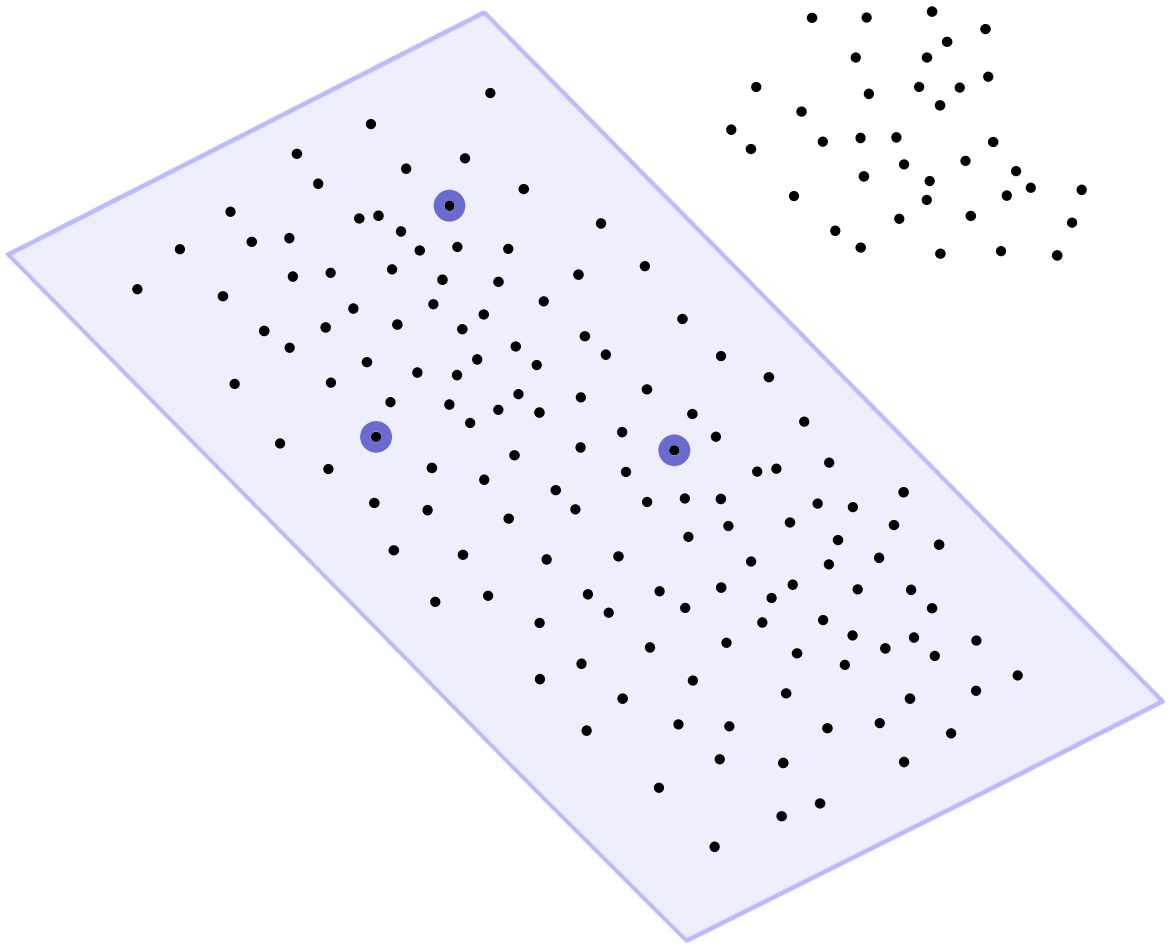
以下の手順１～４を繰り返す。

手順１　点群からランダムに３点を選ぶ。

手順２　３点を通る平面の方程式を作る。

手順３　平面にとても近い点を数える。

手順４　点の数が記録更新なら，
その平面を覚えておく。



平面を自動検出する

【平面を検出するRANSAC】 [F. T-Kurdi et al., 2008]

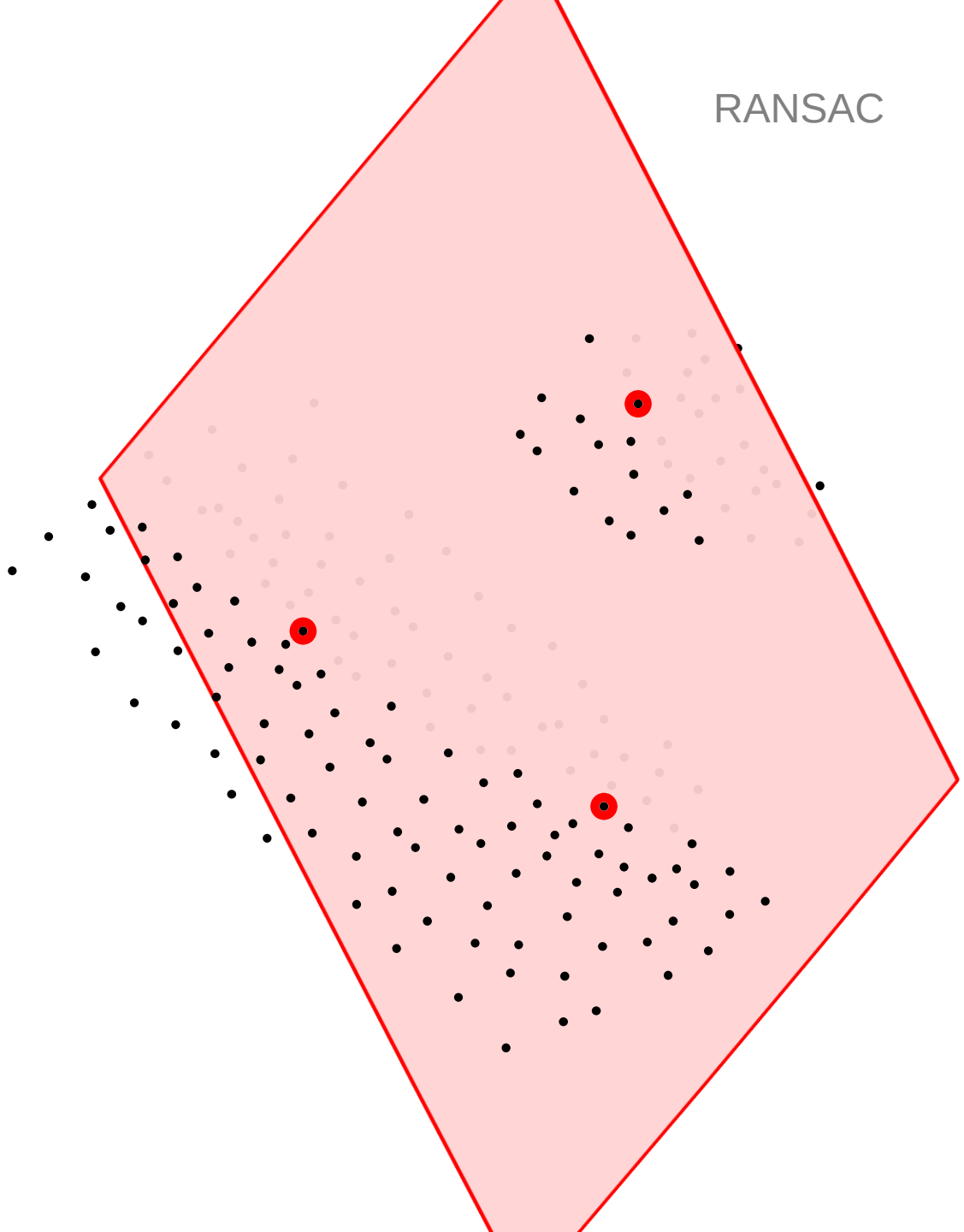
以下の手順１～４を繰り返す。

手順１　点群からランダムに３点を選ぶ。

手順２　３点を通る平面の方程式を作る。

手順３　平面にとても近い点を数える。

手順４　点の数が記録更新なら，
その平面を覚えておく。

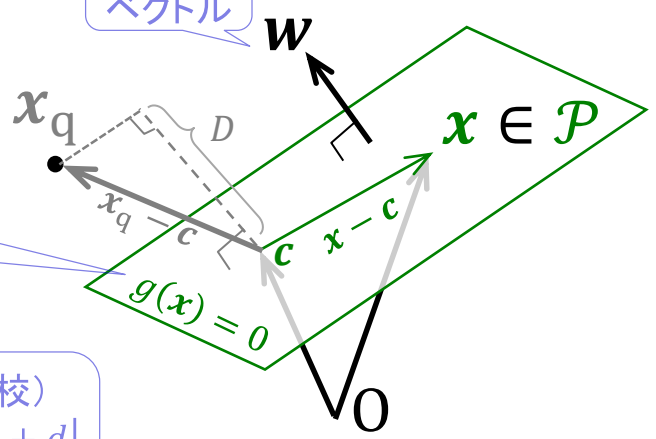


補遺：平面と符号付き距離

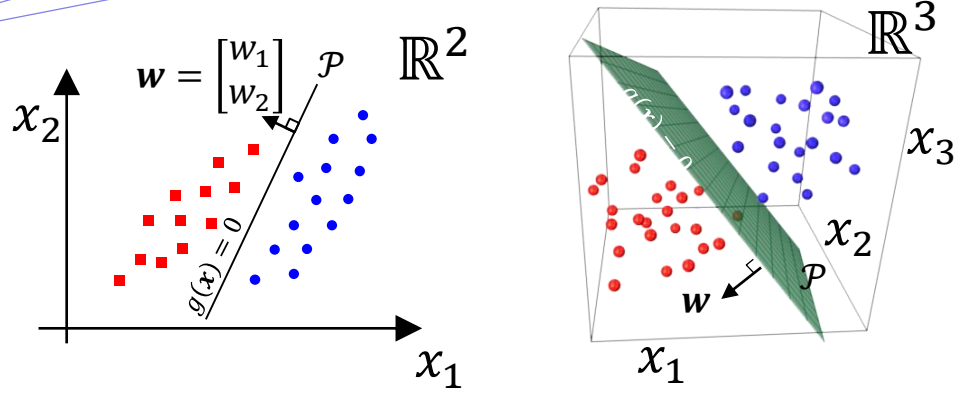
（後期「パターン認識と機械学習」でも活躍します）

- 法線 $\boldsymbol{w}$，通る点 $\boldsymbol{c}$ の超平面：（$\boldsymbol{c}$：位置ベクトル）
  $$\mathcal{P}(\boldsymbol{w},\boldsymbol{c}) = \{\, \boldsymbol{x} \mid \boldsymbol{w}\cdot(\boldsymbol{x}-\boldsymbol{c}) = 0 \,\}$$
  超平面（hyperplane）：2次元空間の直線，3次元空間の平面，・・・
  - $g(\boldsymbol{x}) = \boldsymbol{w}\cdot(\boldsymbol{x}-\boldsymbol{c}) = -\boldsymbol{w}\cdot\boldsymbol{c} + \boldsymbol{w}\cdot\boldsymbol{x} = w_0 + w_1x_1 + \cdots + w_nx_n = 0$
    （$\boldsymbol{x}$ の一次関数）
    （$-\boldsymbol{w}\cdot\boldsymbol{c}$ は定数なので w_0 と置く）
    例（$n=3$次元）：平面の方程式 $ax+by+cz+d=0$

$\boldsymbol{w} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$, $\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$



- 点 $\boldsymbol{x}_\mathrm{q}$ と超平面 $\mathcal{P}(\boldsymbol{w},\boldsymbol{c})$ の符号付き距離（signed distance）（$\boldsymbol{x}_\mathrm{q}$：位置ベクトル）
  $$D(\boldsymbol{x}_\mathrm{q},\mathcal{P}) = \left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right)\cdot(\boldsymbol{x}_\mathrm{q}-\boldsymbol{c}) = \frac{\boldsymbol{w}\cdot(\boldsymbol{x}_\mathrm{q}-\boldsymbol{c})}{\|\boldsymbol{w}\|}$$
  点と平面の距離（高校）
  $$D = \frac{|ax_\mathrm{q} + by_\mathrm{q} + cz_\mathrm{q} + d|}{\sqrt{a^2+b^2+c^2}}$$
  - $D(\boldsymbol{x}_\mathrm{q},\mathcal{P}) > 0 \iff g(\boldsymbol{x}_\mathrm{q}) > 0 \iff \boldsymbol{x}_\mathrm{q}$ は $\boldsymbol{w}$ の正の側
    $D(\boldsymbol{x}_\mathrm{q},\mathcal{P}) = 0 \iff g(\boldsymbol{x}_\mathrm{q}) = 0 \iff \boldsymbol{x}_\mathrm{q} \in \mathcal{P}$（面上）
    $D(\boldsymbol{x}_\mathrm{q},\mathcal{P}) < 0 \iff g(\boldsymbol{x}_\mathrm{q}) < 0 \iff \boldsymbol{x}_\mathrm{q}$ は $\boldsymbol{w}$ の負の側



高次元データを符号で「識別」できます．

最後のメッセージ

情報科学・データ科学

『見る・聞く・考える』を『数学』に翻訳！

コンピュータ

(ﾟДﾟ)ｳﾏｰ ⇒ ~~明日~~今日から本気出す！