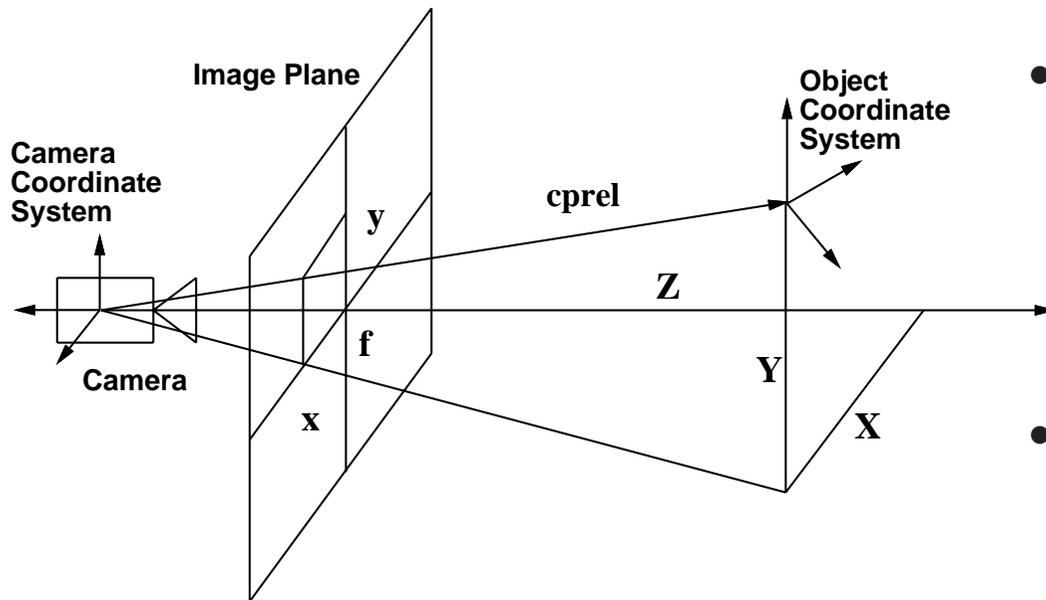


特異値分解の応用

- キャリブレーション
- ビジョンシステムの特性評価
- ロボットの操作性評価

透視射影変換



- 幾何光学 (レンズのモデル)

- 深さ Z , 焦点距離 f

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z}$$

- 透視投影変換

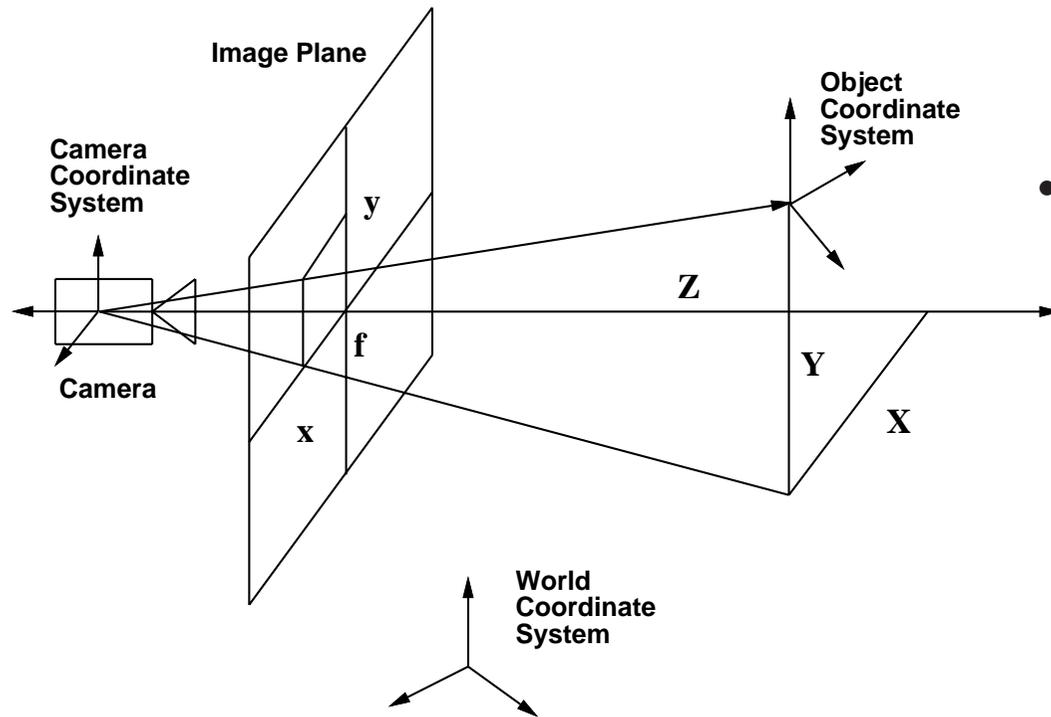
$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- 透視投影変換行列

$$A = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- CCD の縦横の解像度の違い, CCD 面のゆがみなどを考慮すると A はさらに複雑になる.

ビジョンシステム



- 透視投影変換

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = A \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- 対象物とカメラの関係 (${}^C T_O$)

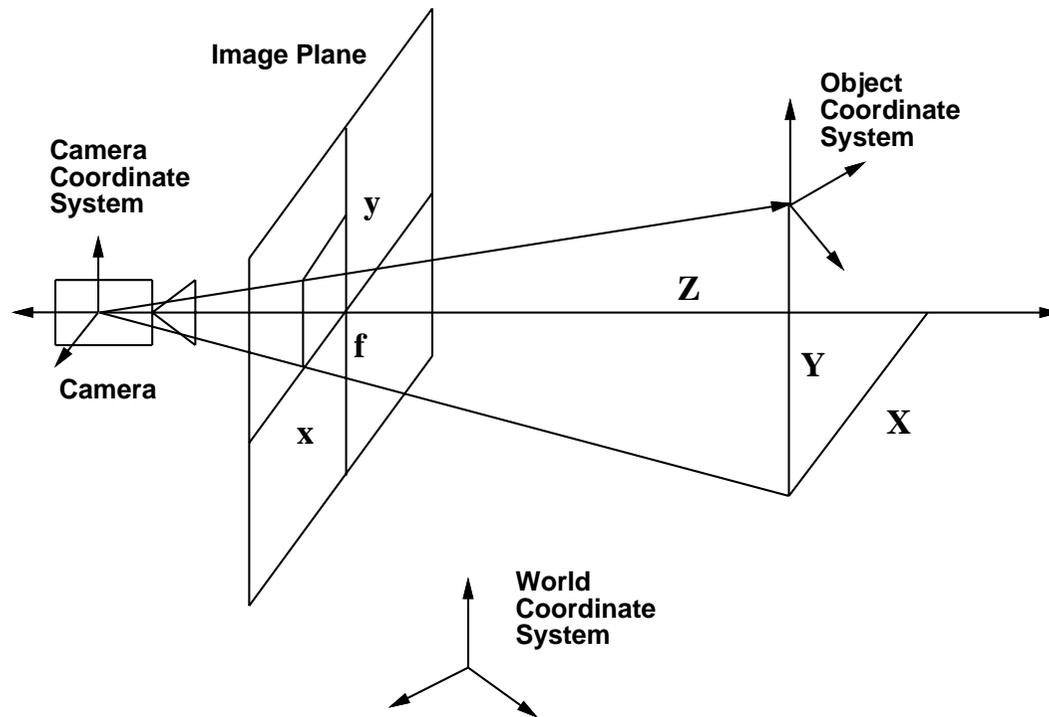
- 対象物の座標系において表現 ${}^O r_i$ をもつ点のカメラ座標系での位置

$$\begin{bmatrix} {}^C r_i \\ 1 \end{bmatrix} = {}^C T_O \begin{bmatrix} {}^O r_i \\ 1 \end{bmatrix}, \quad {}^C r_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$$

- 撮像面上の位置

$$\begin{aligned} s_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} &= A {}^C T_O \begin{bmatrix} {}^O r_i \\ 1 \end{bmatrix} \\ &= A ({}^W T_C)^{-1} {}^W T_O \begin{bmatrix} {}^O r_i \\ 1 \end{bmatrix} \end{aligned}$$

キャリブレーション



$$s_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = A({}^W T_C)^{-1} {}^W T_O \begin{bmatrix} {}^O r_i \\ 1 \end{bmatrix}$$

- 正確に測定された

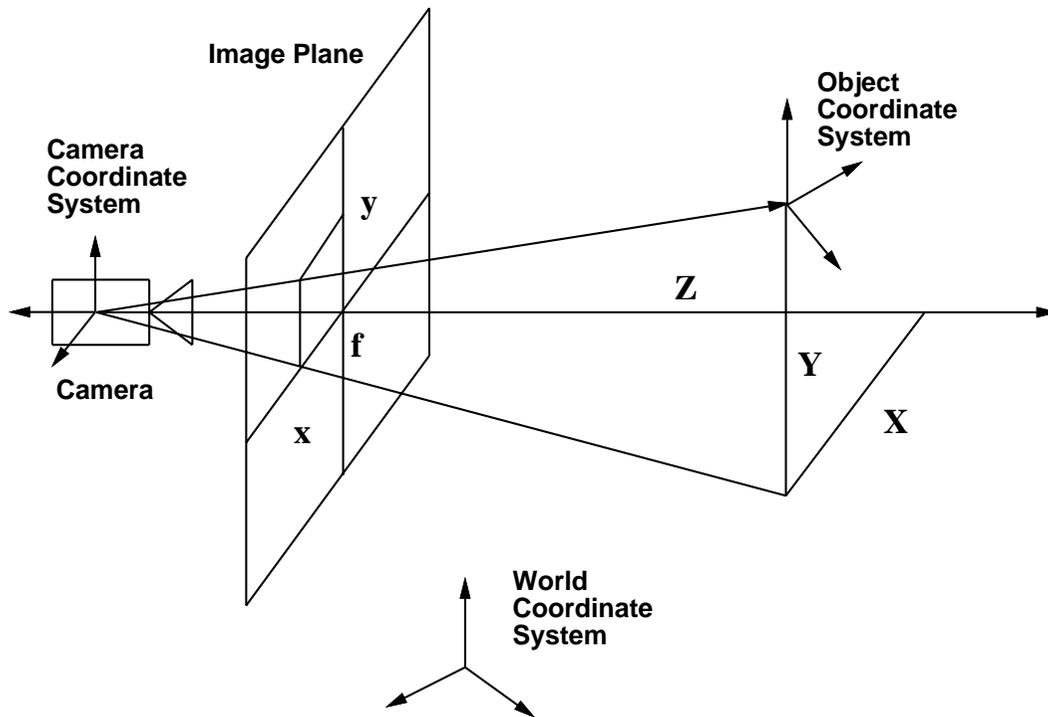
- 対象位置: ${}^W T_O$
- 物体形状: ${}^O r_i$
- 物体の画像: x_i, y_i
ただし $(i = 1, \dots, N)$

に基づいて, カメラの

- 内部パラメータ: A
- 外部パラメータ: ${}^W T_C$

を同定する問題.

ビジョンシステムのキャリブレーション



$$s_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = A({}^W\mathbf{T}_C)^{-1} {}^W\mathbf{T}_O \begin{bmatrix} {}^O\mathbf{r}_i \\ 1 \end{bmatrix}$$
$$= P^W \mathbf{T}_O \begin{bmatrix} {}^O\mathbf{r}_i \\ 1 \end{bmatrix}$$

- P のサイズは 3×4 であり、自由度は12。ただし、 $P' = \alpha P$ に対しても上式は成立するため、本質的には11自由度。
- 1点の観測につき2自由度の情報が得られるため、6点を観測すれば P は決定可能。
- しかしノイズの影響などにより、精度よく推定するためには数十程度の点を計測する必要があり、現在もまだアクティブな研究対象である。

キャリブレーションアルゴリズム

$$s_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = P^W \mathbf{T}_O \begin{bmatrix} 0 \\ \mathbf{r}_i \\ 1 \end{bmatrix}$$

問題: Find $P \in R^{3 \times 4}$ s.t.

$$\sum_i \left\| s_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} - P \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \right\|^2 \rightarrow \min$$

点 i に関して

- $P = [P_{jk}]$ とおくと

$$s_i x_i - P_{11} X_i + P_{12} Y_i + P_{13} Z_i + P_{14}$$

$$s_i y_i - P_{21} X_i + P_{22} Y_i + P_{23} Z_i + P_{24}$$

$$s_i - P_{31} X_i + P_{32} Y_i + P_{33} Z_i + P_{34}$$

- ここで P の要素を縦に並べたベクトルを $\mathbf{P} = [P_{11}, P_{12}, \dots, P_{34}]$ とすると

$$0 - \begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & x_i X_i & x_i Y_i & x_i Z_i & x_i \end{bmatrix} \mathbf{P}$$

$$0 - \begin{bmatrix} 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & y_i X_i & y_i Y_i & y_i Z_i & y_i \end{bmatrix} \mathbf{P}$$

キャリブレーションアルゴリズム

- $\hat{P} = [P_{11}, P_{12}, \dots, P_{33}]$ とおく . $P_{34} = 1$ に注意して , 点 $i = 1, \dots, N$ に関して前の式を連立すると最小化すべき式として

$$\begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & x_1 X_1 & x_1 Y_1 & x_1 Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & y_1 X_1 & y_1 Y_1 & y_1 Z_1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & x_2 X_2 & x_2 Y_2 & x_2 Z_2 \\ 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & y_2 X_2 & y_2 Y_2 & y_2 Z_2 \\ & & & & & & & \vdots & & & \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & y_N X_N & y_N Y_N & y_N Z_N \end{bmatrix} \hat{P}$$

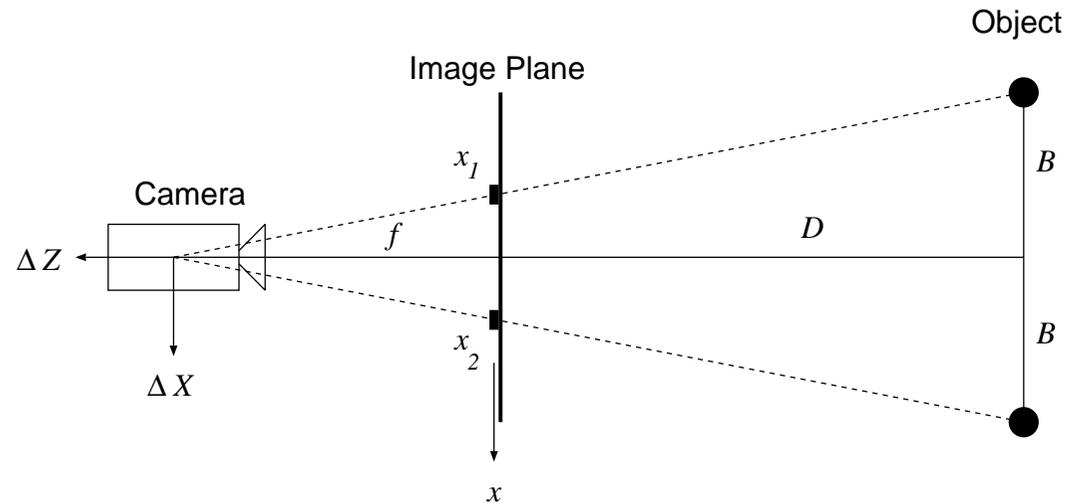
を得る .

- したがって , キャリブレーション問題は下記の最小2乗問題となる

$$\min_{\mathbf{x}} \|\mathbf{y} - A\mathbf{x}\|^2$$

- $P = A(WT_C)^{-1}$ と分解するにはさらに作業が必要
- オブジェクト位置を精度良く設定することは困難
- 実際はノイズのために上記の線形解法では精度が悪い

ビジュアルサーボシステムの感度



対象物上の2点を観測

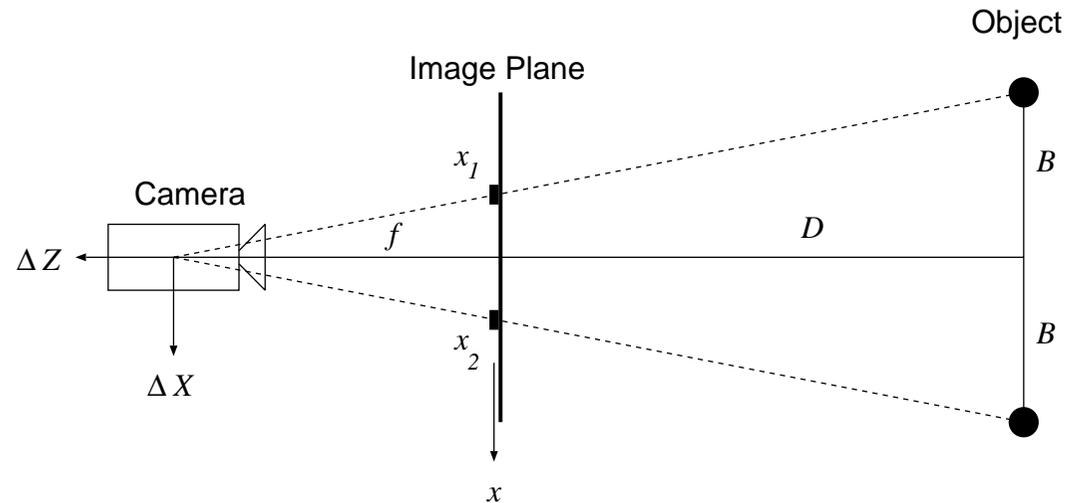
- 2点の像

$$\xi = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad x_1 = -f \frac{B}{D}, \quad x_2 = f \frac{B}{D}$$

- カメラの自由度

$$p = \begin{bmatrix} \Delta X \\ \Delta Z \end{bmatrix}$$

ビジュアルサーボシステムの感度



- カメラが動くことによる像の変位

$$\frac{\partial \xi}{\partial p} = \begin{bmatrix} -\frac{f}{D} & \frac{fB}{D^2} \\ -\frac{f}{D} & -\frac{fB}{D^2} \end{bmatrix}$$

- ヤコビ行列

$$J = \frac{\partial \xi}{\partial p}$$

感度解析の例

- たとえば $D = 1000, B = 100, f = 100$ のとき

$$J = \begin{bmatrix} -0.1 & 0.01 \\ -0.1 & -0.01 \end{bmatrix}$$

- 特異値分解 ($J = U\Sigma V^T$)

$$U = \begin{bmatrix} -0.707 & -0.707 \\ -0.707 & 0.707 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 0.141 & 0.0 \\ 0.0 & 0.014 \end{bmatrix}, \quad V^T = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

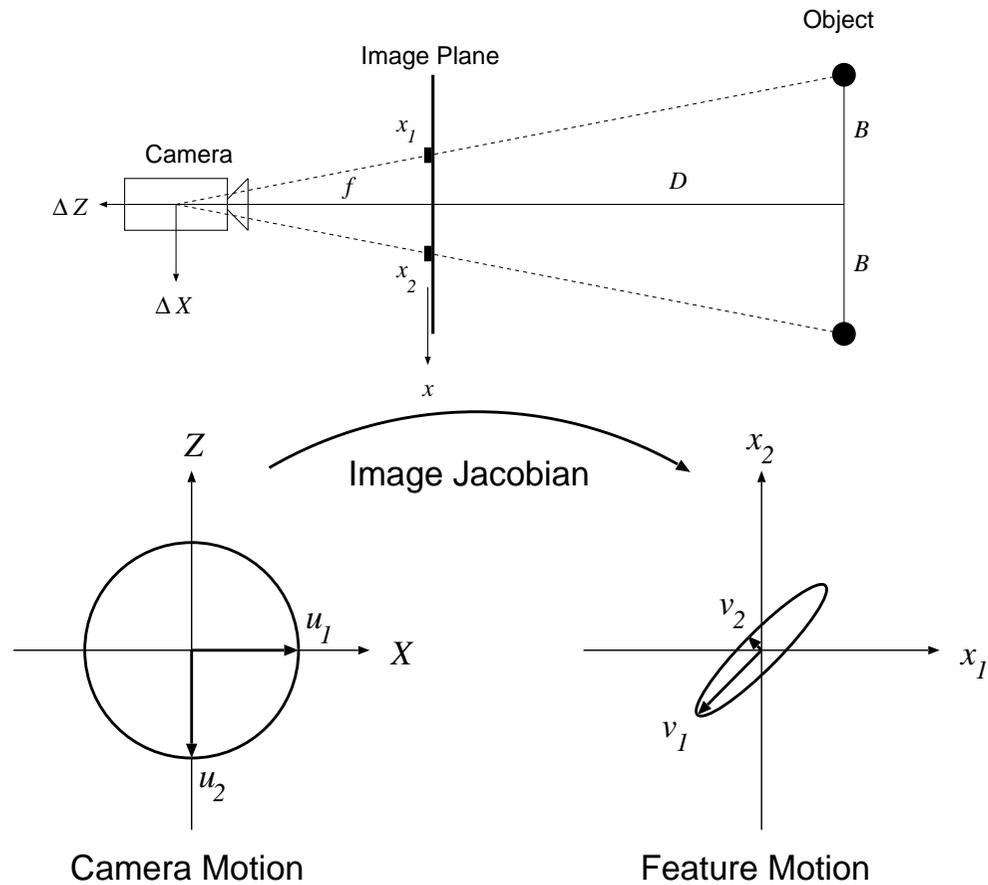
- かたい方向・柔らかい方向

$$JV = UD, \quad Jv_1 = \sigma_1 u_1, \quad Jv_2 = \sigma_2 u_2$$

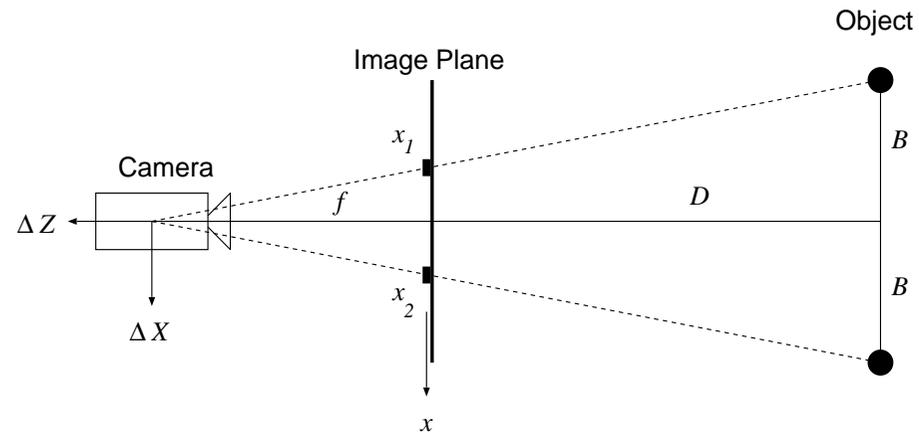
カメラの動きやすさ

カメラの運動 → 特徴量の運動

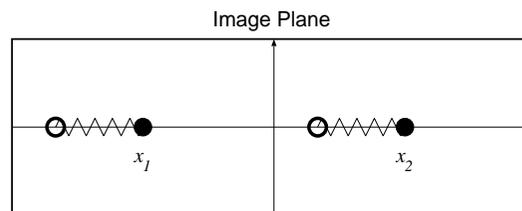
$$J \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 0.141 \begin{bmatrix} -0.707 \\ -0.707 \end{bmatrix}, \quad J \begin{bmatrix} 0 \\ -1 \end{bmatrix} = 0.014 \begin{bmatrix} -0.707 \\ 0.707 \end{bmatrix}$$



感度の解釈

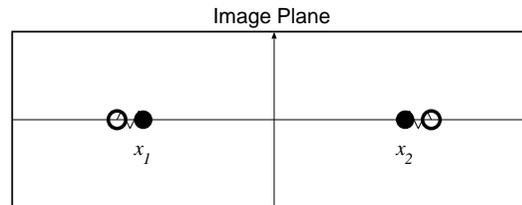


- カメラの u_1 ($= X$) 方向の運動は大きな画像の変化 (v_1 方向)



感度の解釈

- u_2 ($= -Z$) 方向の運動は小さな画像の変化 (v_2 方向)



- ビジュアルサーボは画像偏差に基づいてアクチュエータを動かすので， X 方向にはかたく， Z 方向には柔らかくなる．
- ヤコビ行列の特異値分解を用いることで制御システムの持つ特性を評価できる．

可操作度

- n 自由度をもつマニピュレータ

- 関節角度ベクトル: $\mathbf{q} \in R^n$, 手先位置ベクトル: $\mathbf{r} \in R^m$

- キネマティクス: $\mathbf{r} = f(\mathbf{q})$

- ヤコビ行列: $J = \frac{\partial \mathbf{r}}{\partial \mathbf{q}}$

- 速度関係式: $\dot{\mathbf{r}} = J\dot{\mathbf{q}}$

- 可操作楕円体: $\|\dot{\mathbf{q}}\| \leq 1$ をみたす関節速度 $\dot{\mathbf{q}}$ を用いて実現しうる手先速度 $\dot{\mathbf{r}}$ の集合

- * 主軸半径の長い方向: 大きな速度を出しやすい方向

- * 主軸半径の短い方向: 小さな速度しか出せない方向

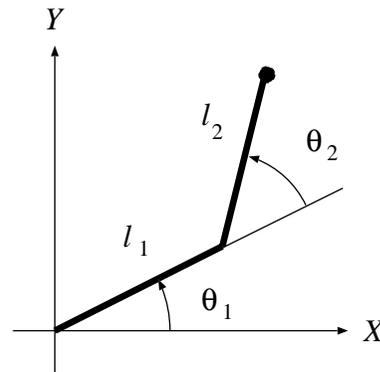
- 可操作楕円体

$$\begin{aligned} 1 &\geq \|\dot{\mathbf{q}}\|^2 = \|J^+\mathbf{v} + (I - J^+J)\mathbf{k}\|^2 \\ &\geq \mathbf{v}^T (J^+)^T J^+ \mathbf{v}, \quad \mathbf{v} \in \text{Im}J \end{aligned}$$

- 可操作度: 操作のしやすさの指標, 可操作楕円体の体積

$$w = \sqrt{\det JJ^T} \quad (m = n \text{ のとき } w = |\det J|)$$

可操作度の例



- 手先位置: r , ヤコビ行列: J

$$\mathbf{r} = \begin{bmatrix} l_1 C_1 + l_2 C_{12} \\ l_1 S_1 + l_2 S_{12} \end{bmatrix}, \quad \mathbf{J} = \frac{\partial \mathbf{r}}{\partial \boldsymbol{\theta}} = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} \end{bmatrix}$$

- 可操作度: w

$$w = \sqrt{\det \mathbf{J}\mathbf{J}^T} = l_1 l_2 |S_2|$$

- l_1, l_2, θ_1 の値とは無関係に $\theta_2 = \pm\pi/2$ のときに最適姿勢

可操作楕円体

$l_1 = l_2 = 1, \theta_2 = -2\theta_1$ のとき

