

# 画像認識の基礎から応用

和歌山大学

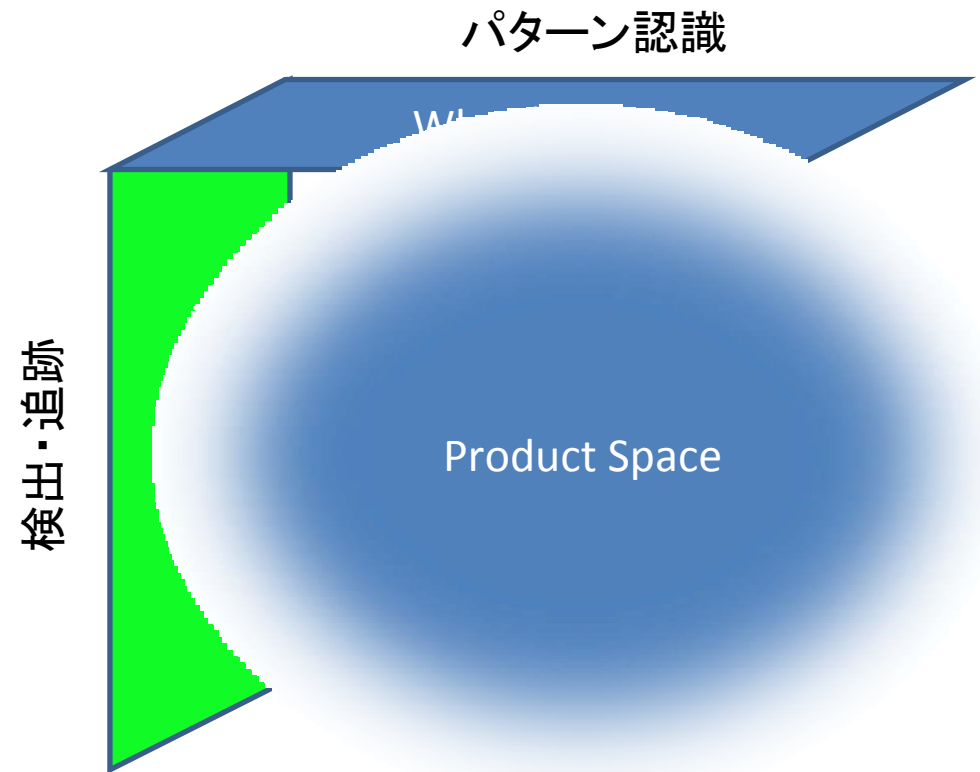
和田俊和

# 画像認識:「画像」と「認識」, どちらが大事?

- 認識: WHAT, WHICHを問う問題
  - Cognition or Re-cognition?=>知ってるから分かる  
( Pattern Recognition: 分類問題 )
- 画像: WHERE, HOWを求めなければいけない.
  - 2次元なのに位置・向き・大きさ等の探索が必要  
( 検出, 追跡の問題 )

# 画像認識の問題を要約すると

- A) WHAT・WHICHが分かれば探索できる.
- B) WHERE・HOWがわかれば認識できる.
- C) 上のA)B)を同時に行うことは困難な問題



# 近年の進歩

- (WHAT・WHICH) 認識技術の進歩
  - SVM, ADA Boost, (計算論的学習理論)
- (WHERE・HOW) 画像解析の進歩
  - リッチで不変な画像特徴の抽出 (SIFT)
    - 対応付けの安定化⇒GeometryとPhotometryの進歩
    - 画像識別の高性能化⇒一般画像認識(テキスト検索技術の導入)
  - 画像の離散最適化 (Graph Cuts, Belief Propagation)
    - ノイズ除去, セグメンテーション, 貼り合わせ等々が大幅に改善された.
    - MRFを仮定した処理が可能になった.
- 融合領域の発展
  - 顔検出 (Cascaded ADA Boosting + Integral Image)
  - 事例ベース処理 (Irregularity Detection, Hallucination, 最近傍探索・識別)

# パターン認識

過去から現在まで

# パターン認識を育んだ人々(1)

- Bayes の定理1763 : T. Bayes
- 正規分布1818: C.F.Gauss
- 最近傍識別理論1967: P.E. Hart
- 複合類似度法1973: 飯島泰蔵
- 部分空間法1987: K. Fukunaga



# パターン認識を育んだ人々(2)

- Support Vector Machine1995: V.N. Vapnik



- ADA Boosting1995:

Y. Freund and  
R.E. Schapire



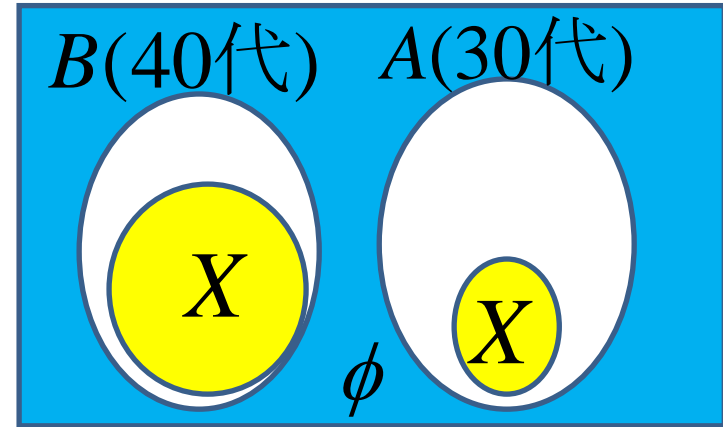
他多数.

# 基本1: Bayesの定理

$$P(\omega_i | X) = \frac{P(\omega_i)P(X | \omega_i)}{\sum_{j=1}^{\Omega} P(\omega_j)P(X | \omega_j)}$$

↑ 帰結      ↑ 条件

例(「30代と40代が全体集合」と仮定)  
 A:30代, B:40代, X:老眼である.  
 ⇒ 老眼の人が30代である確率は?



$$P(A | X) = \frac{P(A)P(X | A)}{P(A)P(X | A) + P(B)P(X | B)} = \frac{\text{small yellow circle}}{\text{small yellow circle} + \text{large yellow circle}}$$

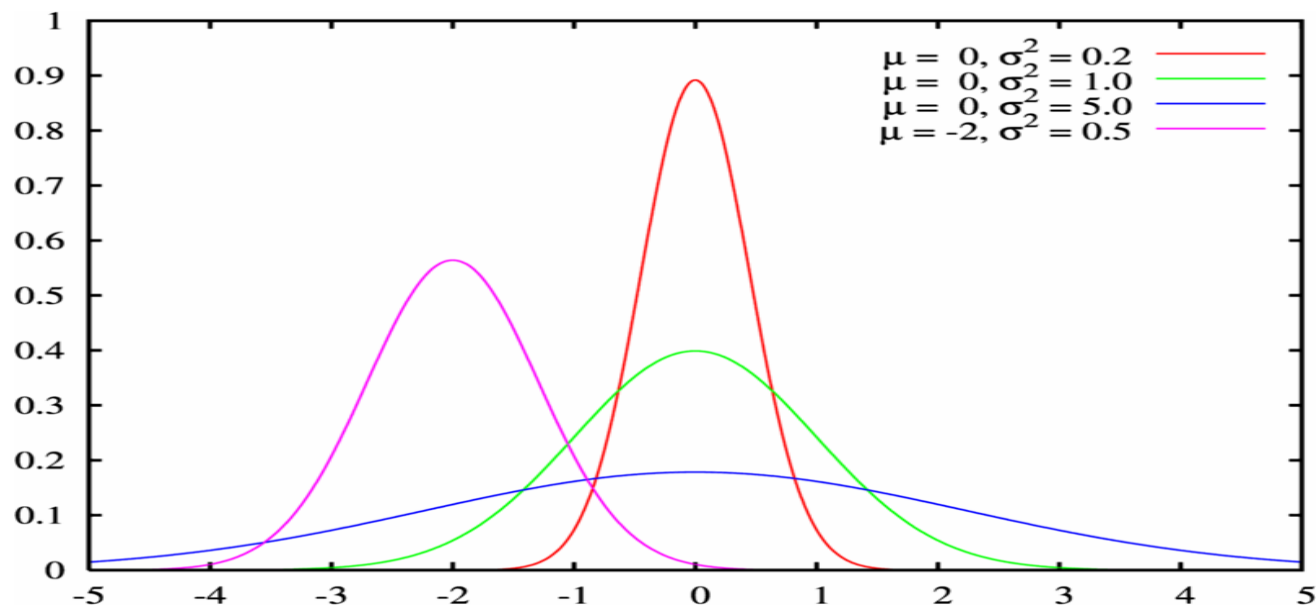


## 基本2：正規分布（一般化された2項分布）

$$P(\omega_i | \mathbf{x}) = P(\omega_i) p(\mathbf{x} | \omega_i) / \sum_{j=1}^{\Omega} P(\omega_j) p(\mathbf{x} | \omega_j)$$

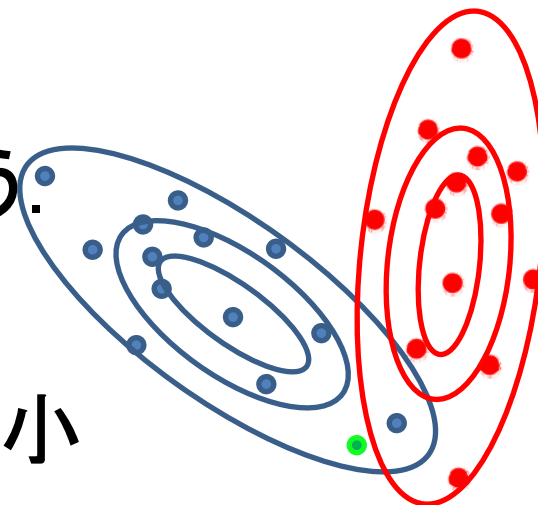
$p(\mathbf{x} | \omega_i)$ :  $\omega_i$ に属するデータ  $\mathbf{x}$ が生起する確率密度

$$p(x | \omega_i) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \right\}$$



## 基本1, 2の組み合わせ(統計的識別)

1. 正規分布の当てはめを行う.
2. 入力の特徴密度の計算を行う.  
 $P(\omega_1)p(\mathbf{x} | \omega_1) \quad P(\omega_2)p(\mathbf{x} | \omega_2)$
3. 事前確率をかけ, この値の大小によって  $\mathbf{x}$  を分類する.



(事後確率の分子を比較していることに相当)

$$P(\omega_1 | \mathbf{x}) = \frac{P(\omega_1)P(\mathbf{x} | \omega_1)}{P(\omega_1)P(\mathbf{x} | \omega_1) + P(\omega_2)P(\mathbf{x} | \omega_2)}$$

# 正規分布の母数の推定と主成分分析

平均値と共分散行列の最尤推定

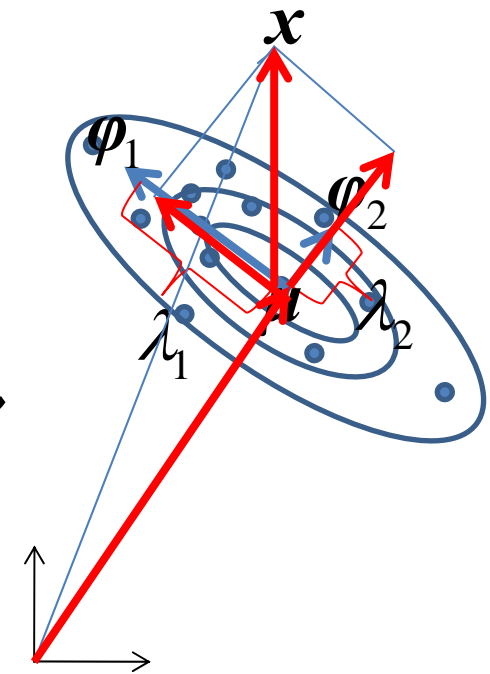
$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \quad \mathbf{C} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$$

共分散行列の固有値, 固有ベクトル

$$\mathbf{C} = [\boldsymbol{\varphi}_1 \quad \boldsymbol{\varphi}_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi}_1^T \\ \boldsymbol{\varphi}_2^T \end{bmatrix}$$

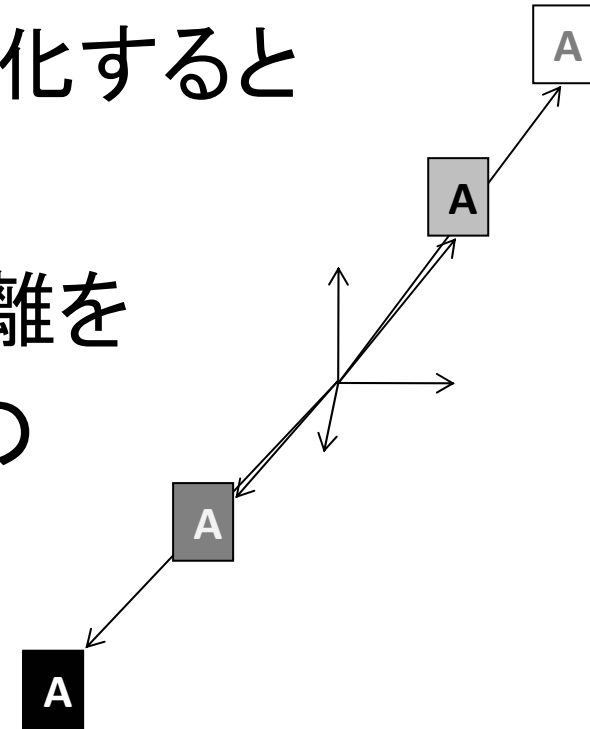
任意のベクトルの直交分解

$$\mathbf{x} = [\boldsymbol{\varphi}_1 \quad \boldsymbol{\varphi}_2] \begin{bmatrix} \boldsymbol{\varphi}_1^T \\ \boldsymbol{\varphi}_2^T \end{bmatrix} (\mathbf{x} - \boldsymbol{\mu}) + \boldsymbol{\mu}$$



# 明るさの変化に対して不変な 画像の類似性評価

- 同じパターンでも明るさが変化するとベクトルの長さが変化する.
- したがって, ベクトル間の距離を評価する方法では, 画像間の類似度は評価できない.
- この代わりに, ベクトル間の角度を評価する「単純類似度」がある.



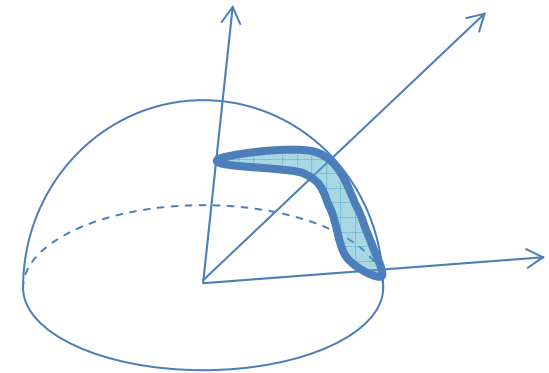
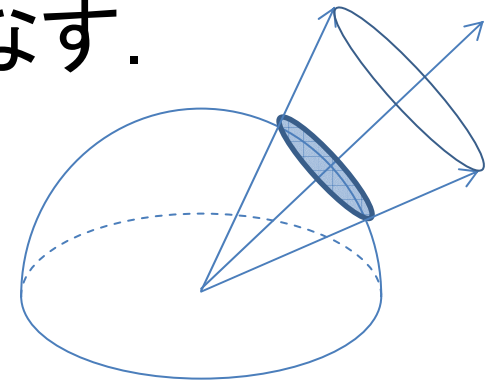
$$s(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{\|\mathbf{x}_1\| \|\mathbf{x}_2\|} = \cos \angle \mathbf{x}_1, \mathbf{x}_2$$

# 単純類似度の限界

- 代表パターンとの角度差が一定値未満のものを同じクラスに属するとみなす.

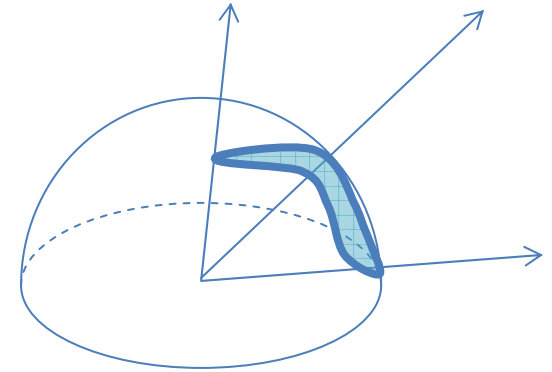
$$s(\mathbf{x}_r, \mathbf{x}) = \frac{\mathbf{x}_r^T \mathbf{x}}{\|\mathbf{x}_r\| \|\mathbf{x}\|} \geq \text{Threshold}$$

- 統計的変動が表現できない.
- 統計的変動を表現するために考え出されたのが「複合類似度」



# 複合類似度

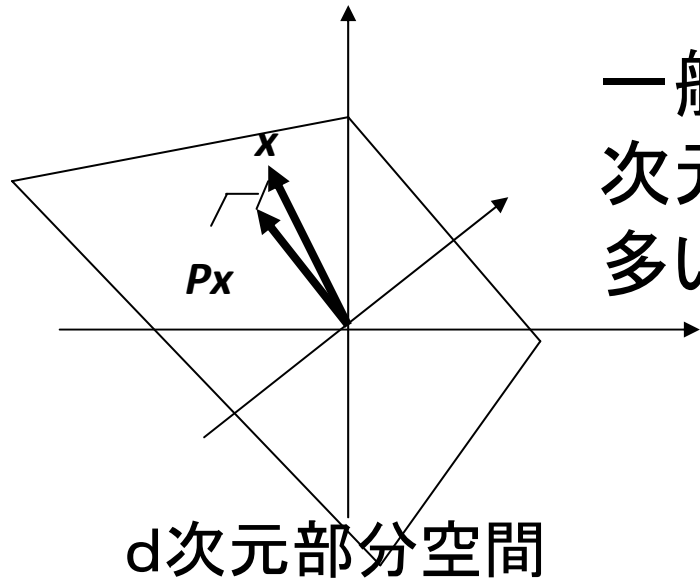
- 自己相関行列の固有値問題を解き,  $\lambda_j, \varphi_j$  を得る.
- これを用いて定義される次式が複合類似度である.



$$s(\mathbf{x}) = \sum_{j=1}^d \frac{\lambda_j (\varphi_j^T \mathbf{x})^2}{\lambda_1 \|\mathbf{x}\|^2}$$

各主軸への射影の2乗量に  $\frac{\lambda_j}{\lambda_1}$  という量をかけてばらつきの違いを表現している.

# 部分空間法



一般に同一クラスに属するパターンは低次元の部分空間内に偏在するケースが多い。

$$P = UU^T$$

$$U = [\varphi_1 \quad \cdots \quad \varphi_d]$$

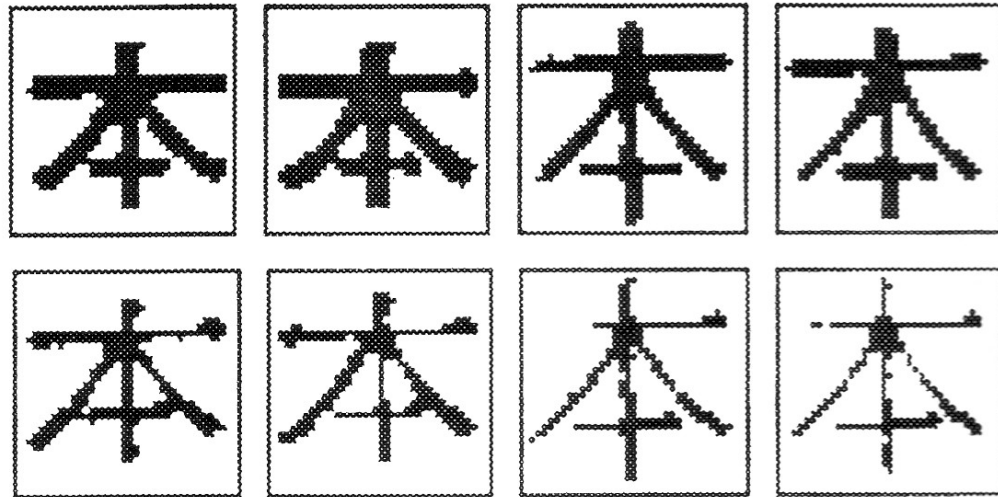
d次元正規直交基底 (K-L展開によって求める)

$Px$  部分空間への射影成分

# 部分空間の構築

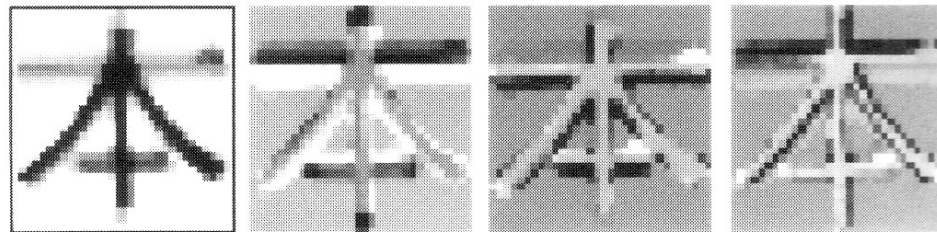
自己相関行列の固有値問題を解くことによって  $\varphi_i$  が得られる。

$$R = \frac{1}{n} \sum_{i=0}^n \mathbf{x}_i \mathbf{x}_i^T$$



トレーニングパターン

の例



$u_1$

$u_2$

$u_3$

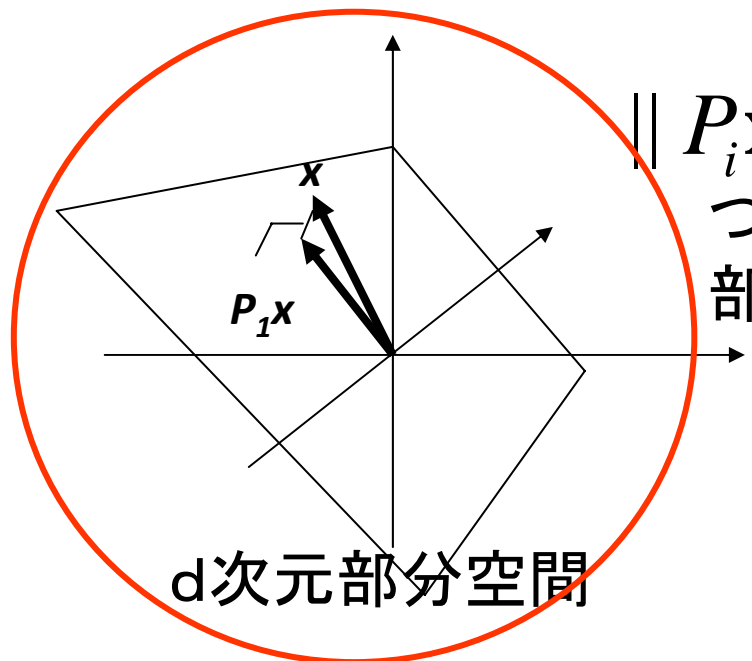
$u_4$

固有ベクトル

ベクトル



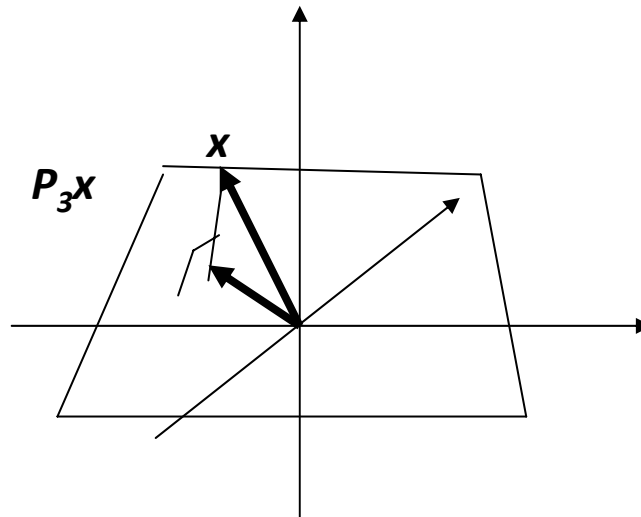
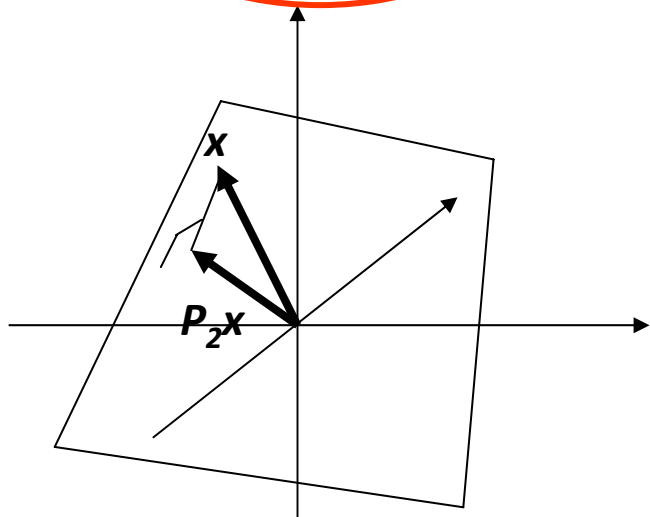
# 部分空間法 (subspace method)



$\|P_i \mathbf{x}\|^2$  を最大化するクラスに分類する。  
つまり、入力との角度差が最も小さい  
部分空間に属すると判定する方法。

$$U_i = \begin{bmatrix} \phi_{i1} & \cdots & \phi_{id_i} \end{bmatrix}$$

$$P_i = U_i U_i^T$$



# 部分空間法(評価尺度)

$P_i P_i = P_i$  および  $P_i^T = P_i$  が成り立つ。

このことから、

$$\|P_i \mathbf{x}\|^2 = \mathbf{x}^T P_i^T P_i \mathbf{x} = \mathbf{x}^T P_i P_i \mathbf{x} = \mathbf{x}^T P_i \mathbf{x}$$

が成り立つ。さらに、効率の良い計算方法を求めると

$$\|P_i \mathbf{x}\|^2 = \mathbf{x}^T P_i \mathbf{x} = \mathbf{x}^T U_i U_i^T \mathbf{x} = \sum_{j=1}^{d_i} (\boldsymbol{\varphi}_{ij}^T \mathbf{x})^2$$

となり、この値の大小によって識別が行われる。つまり、パターン  $\mathbf{x}$  がクラス  $\omega_i$  に属すると考えた場合の類似度  $S_i(\mathbf{x})$  は、次式で表される。

$$S_i(\mathbf{x}) = \sum_{j=1}^{d_i} (\boldsymbol{\varphi}_{ij}^T \mathbf{x})^2$$

# 複合類似度と部分空間法

- 分母  $\lambda_1 \|\mathbf{x}\|^2$  の部分は正規化項なので、無視できる.

$$s(\mathbf{x}) = \sum_{j=1}^d \frac{\lambda_j (\boldsymbol{\varphi}_j^T \mathbf{x})^2}{\lambda_1 \|\mathbf{x}\|^2}$$

- すると、複合類似度と部分空間法は、各軸の射影成分の2乗量に固有値をかけているかかけていないかの違いになる.

$$S_i(\mathbf{x}) = \sum_{j=1}^{d_i} (\boldsymbol{\varphi}_{ij}^T \mathbf{x})^2$$

# Linear Support Vector Machine:

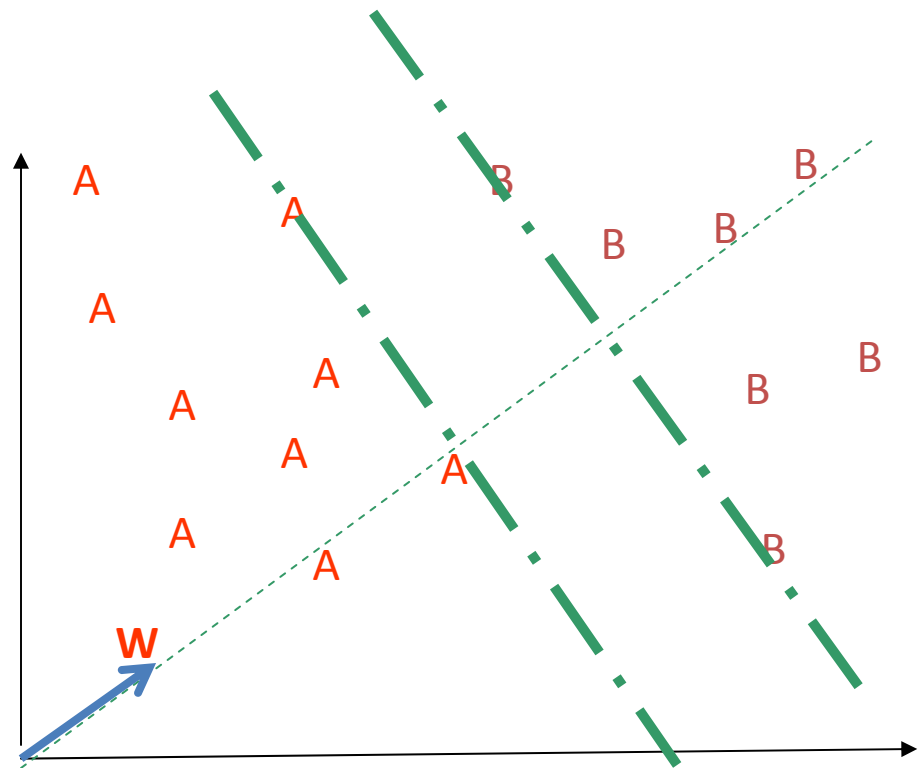
## 線形分離可能な場合

識別面の方程式を  $\mathbf{x} \cdot \mathbf{w} + b = 0$  とし, 教師信号  $t$  を  $\pm 1$  とする. このとき,  
Bの点が  $\mathbf{x} \cdot \mathbf{w} + b \geq 1$ , Aの点が  $\mathbf{x} \cdot \mathbf{w} + b \leq -1$   
を満足するとする。

このとき、全サンプル  $(\mathbf{x}_i, t_i)$  は、  
 $t_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0$   
を満足する。

上の条件が成立する範囲内で  
識別面間の距離  $2/||\mathbf{w}||$  を最大  
化, すなわち,  $||\mathbf{w}||$  を最小  
化する.  $\mathbf{w}$  および  $b$  を求める。

$$\frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^d \alpha_i (t_i (\mathbf{w}^T \mathbf{x}_i + b) - 1)$$



# Linear Support Vector Machine:

線形分離可能な場合

$$\frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^d \alpha_i (t_i (\mathbf{w}^T \mathbf{x}_i + b) - 1)$$

線形分離可能でない場合

$$\frac{1}{2} \|\mathbf{w}\|^2 - C \sum_{i=1}^d \xi_i + \sum_{i=1}^d \mu_i \xi_i$$

$$+ \sum_{i=1}^d \alpha_i (t_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i)$$



$$\sum_{i=1}^d \alpha_i - \sum_{j=1}^d \sum_{i=1}^d \alpha_i \alpha_j t_i t_j \mathbf{x}_i^T \mathbf{x}_j$$

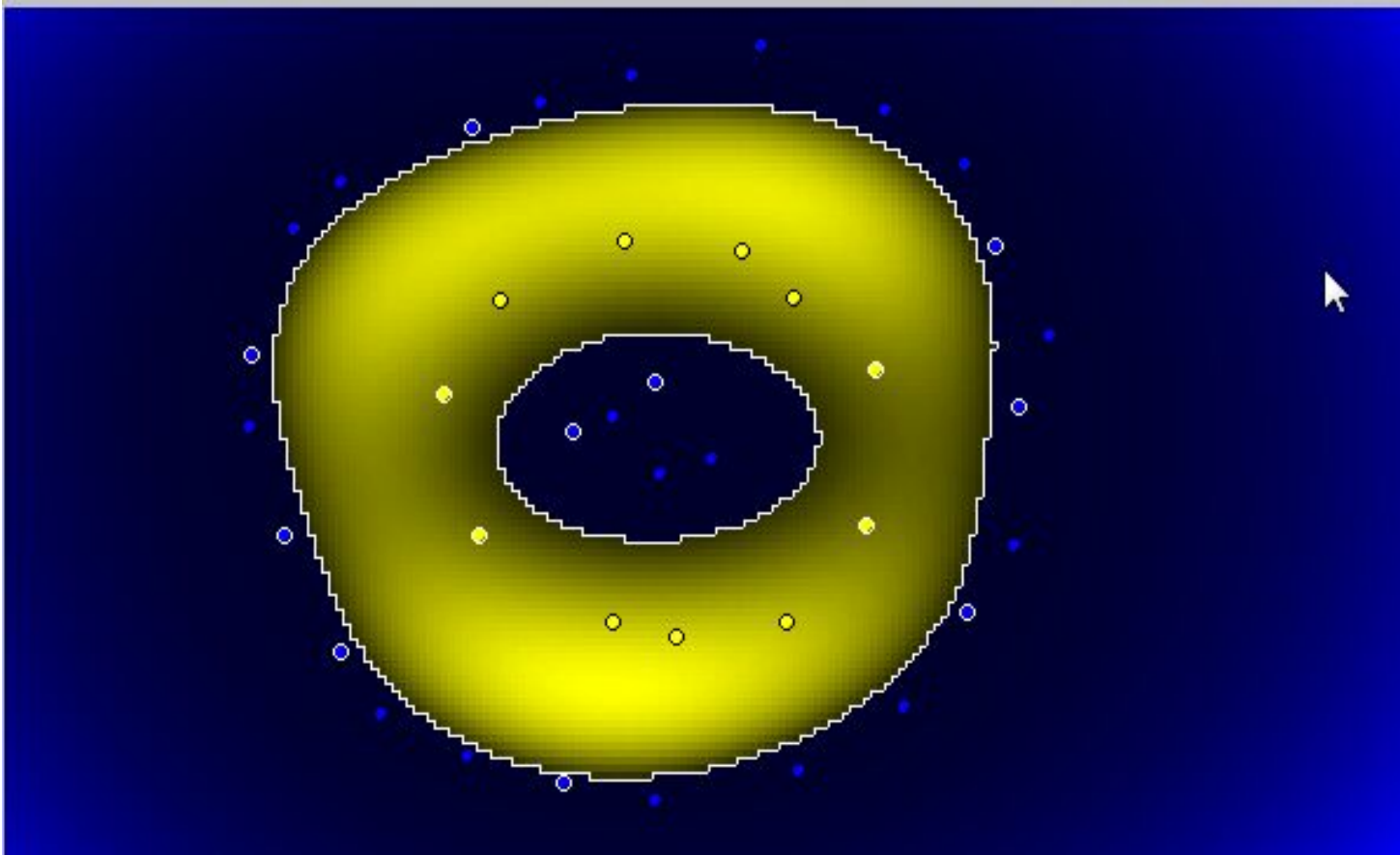


カーネル関数による内積の一般化

$$\sum_{i=1}^d \alpha_i - \sum_{j=1}^d \sum_{i=1}^d \alpha_i \alpha_j t_i t_j K(\mathbf{x}_i, \mathbf{x}_j)$$

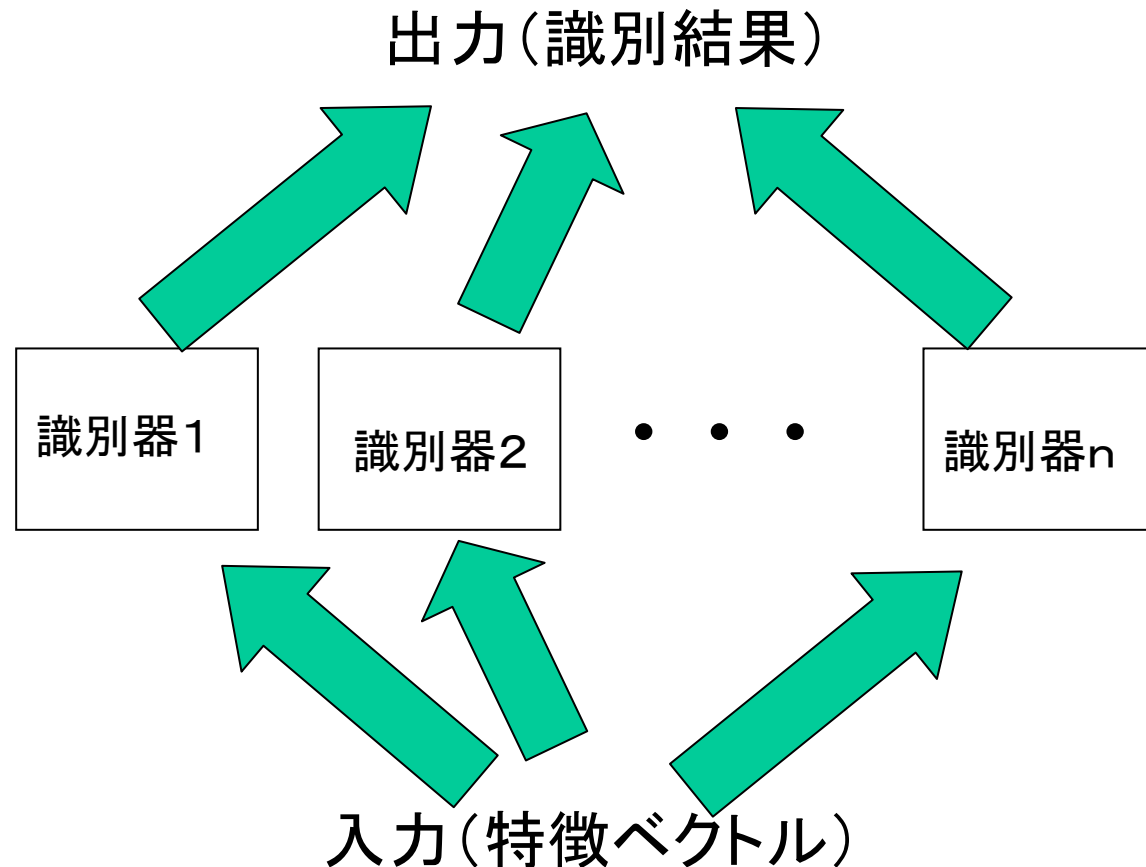
一般的なSVMの目的関数

# SVM(多項式カーネル)の例



# ADA Boosting

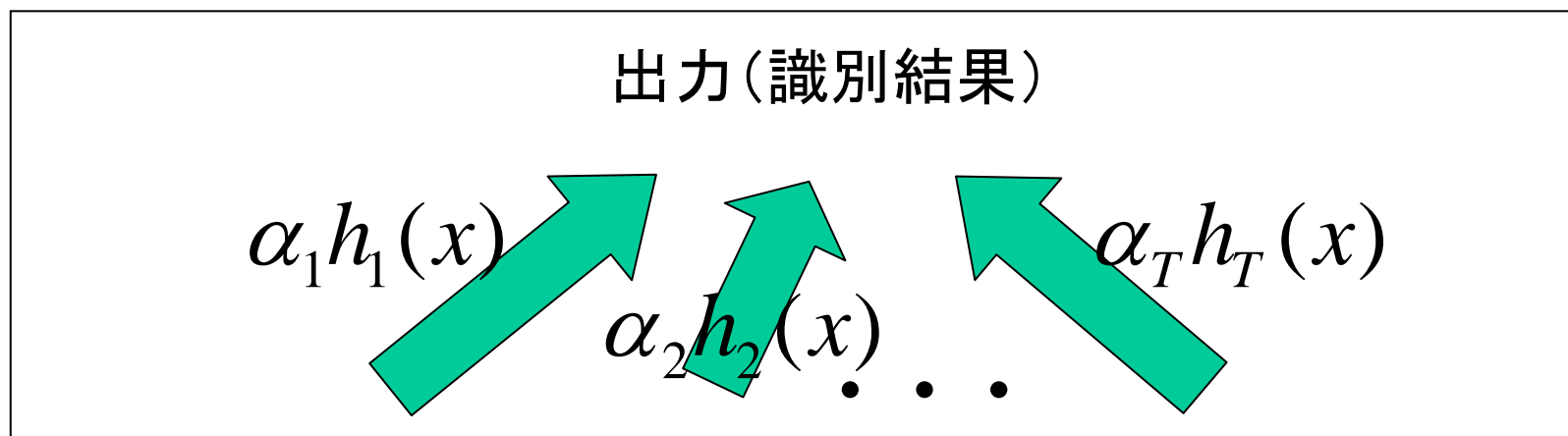
- サイコロを振るよりも少しましな性能を持つ2クラスの識別器が無数にあれば、無限に性能の良い識別器が構成できる。



# 識別器出力の統合

- 組み合わされた出力  $f(x)$  は、個々の識別器の出力  $h_t(x) \rightarrow \{-1, +1\}$  の重み付き和として表される。(この式の符号で判定する.)

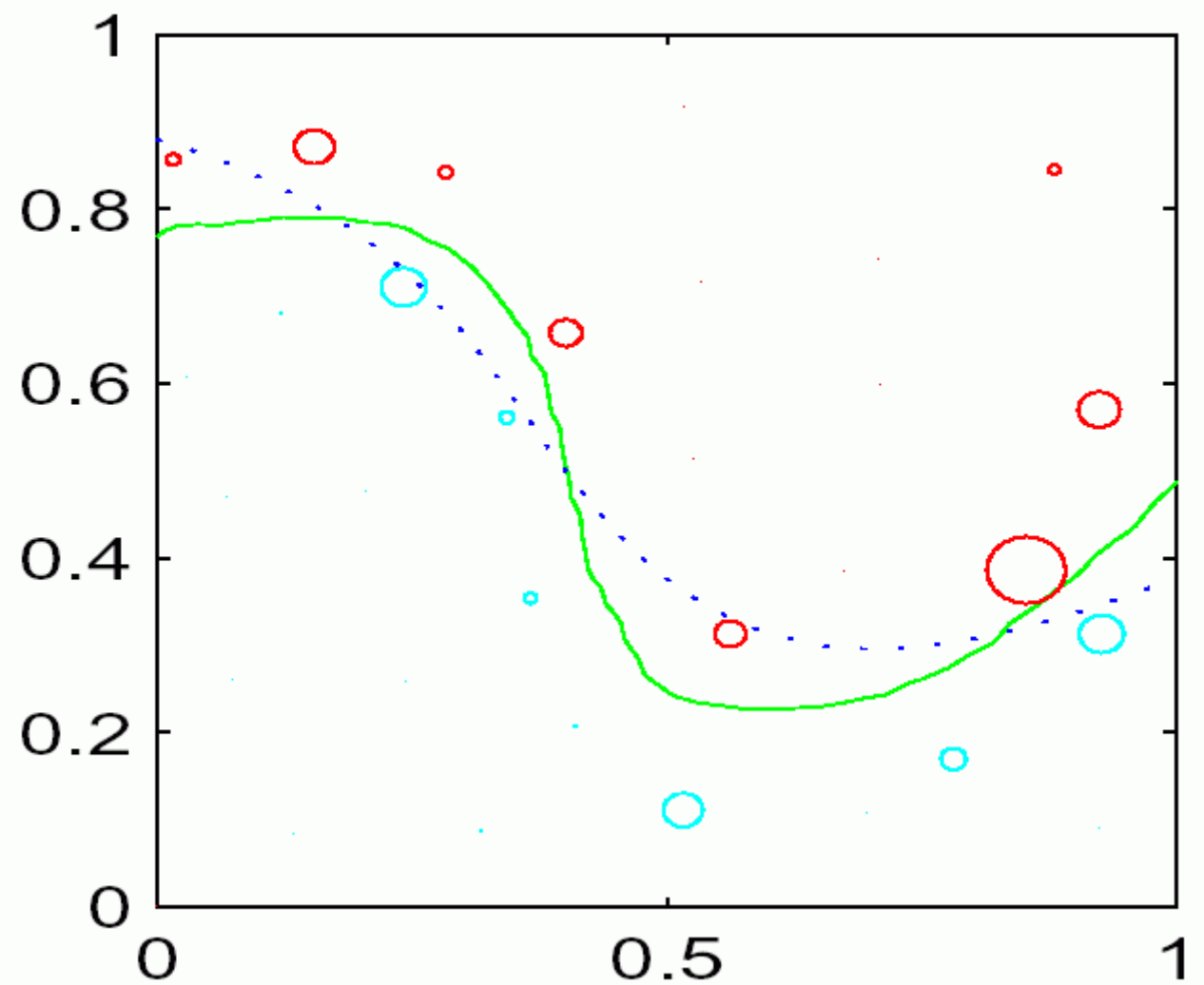
$$f(x) = \sum_{t=1}^T \alpha_t h_t(x)$$





# 例

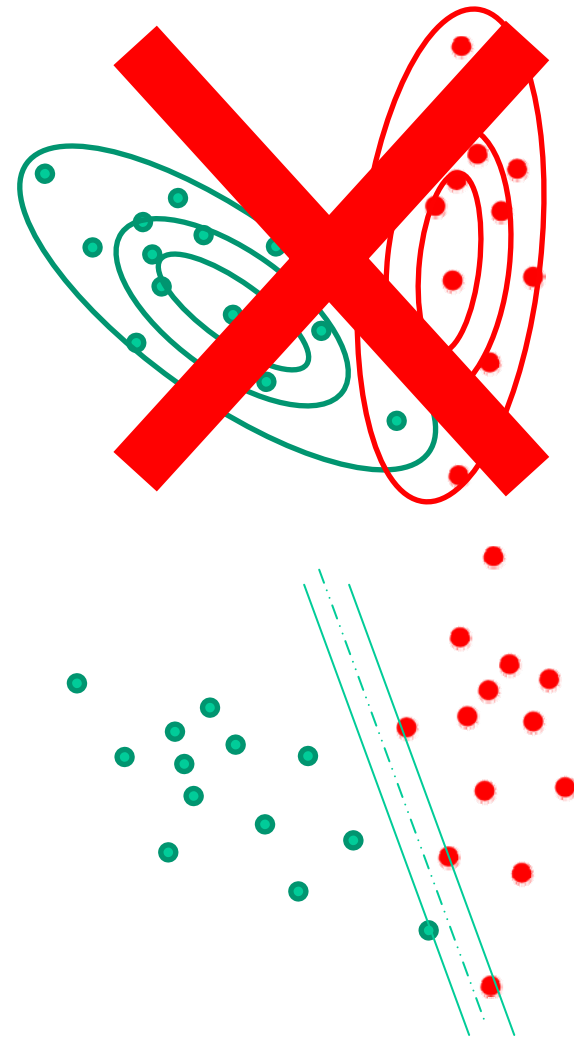
## 100th Iteration



# SVMとADABoostingの共通性

トレーニング用学習データ  
への分布モデルの当ては  
めは行わない。

誤識別を起こしそうな特殊例  
だけが識別面の決定に寄与  
している。



# 画像からの対象検出・追跡

手法中心で人名は略

# 対象検出・追跡

- 対象モデルのマッチングに基づく検出
  - (見えモデル + 全走査) テンプレートマッチング
  - (識別モデル + 全走査) Cascaded ADA Boost
- 対象モデルのマッチングに基づく追跡
  - (見えモデル + 局所探索) テンプレートマッチング
  - (ヒストグラム + 山登り) Mean Shift
  - (エッジモデル + 山登り<sup>\*</sup>) Chamfer マッチング
  - (任意 + ランダムサンプリング) パーティクルフィルタ
- 背景モデルに基づく検出

# 要するに

- モデルの表現
- 類似性・相違性尺度
- スキャン方法

の観点から, ほとんどの追跡アルゴリズムは特徴づけられる.

しかし, これらの組み合わせは極めて多数なので, 全てを説明することはできない.

# 単一モデルとのマッチング 相違性・類似性の尺度



- Sum of Absolute Difference

$$SAD(x, y) = \sum_{j=1}^{T_y} \sum_{i=1}^{T_x} |I(x+i, y+j) - T(i, j)|$$

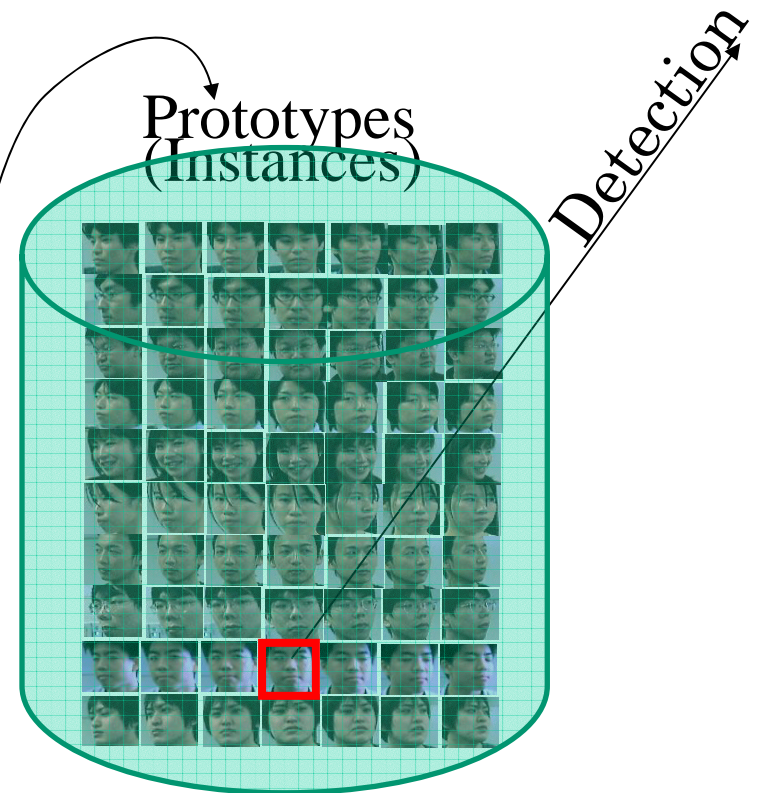
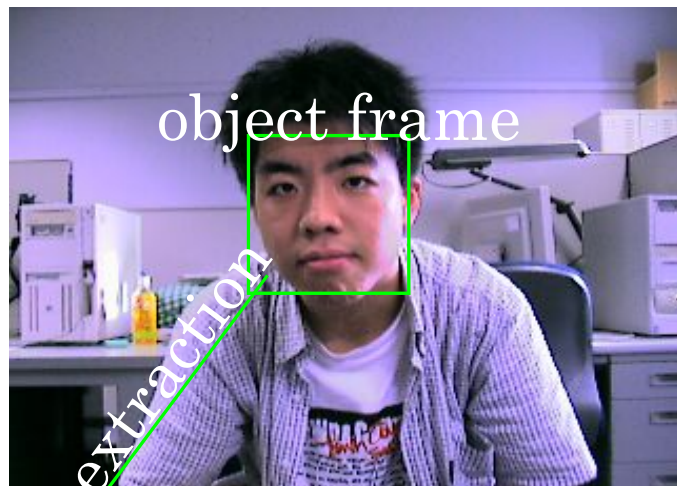
- Sum of Squared Difference

$$SSD(x, y) = \sum_{j=1}^{T_y} \sum_{i=1}^{T_x} (I(x+i, y+j) - T(i, j))^2$$

- Normalized Correlation ( $\cos \theta$ )

$$Corr(x, y) = \frac{\sum_{j=1}^{T_y} \sum_{i=1}^{T_x} I(x+i, y+j)T(i, j)}{\sqrt{\sum_{j=1}^{T_y} \sum_{i=1}^{T_x} I^2(x+i, y+j) \sum_{j=1}^{T_y} \sum_{i=1}^{T_x} T^2(i, j)}}$$

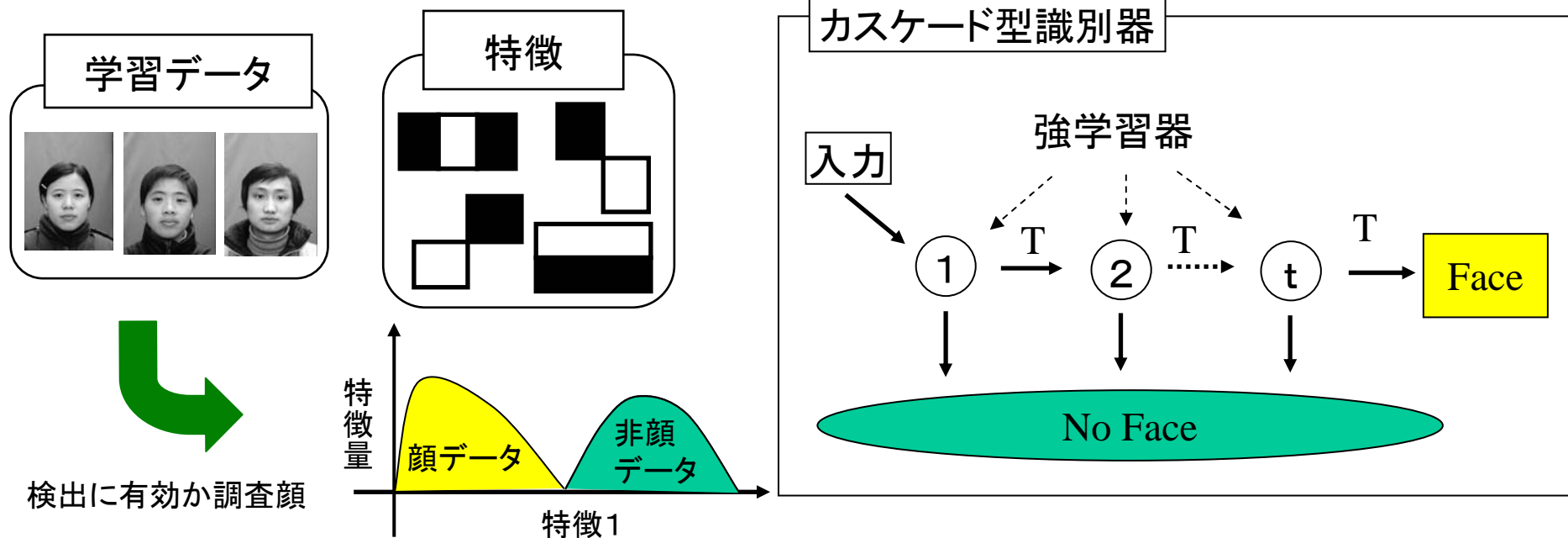
# マルチテンプレートマッチング (複数モデルとの) マッチング



最近傍探索or最大類似度探索

# 識別による検出 (クラスモデルとのマッチング) Cascaded ADABoosting

- ADABoosting で作られた強識別器を多段に接続する(速度と精度を両立させる工夫)



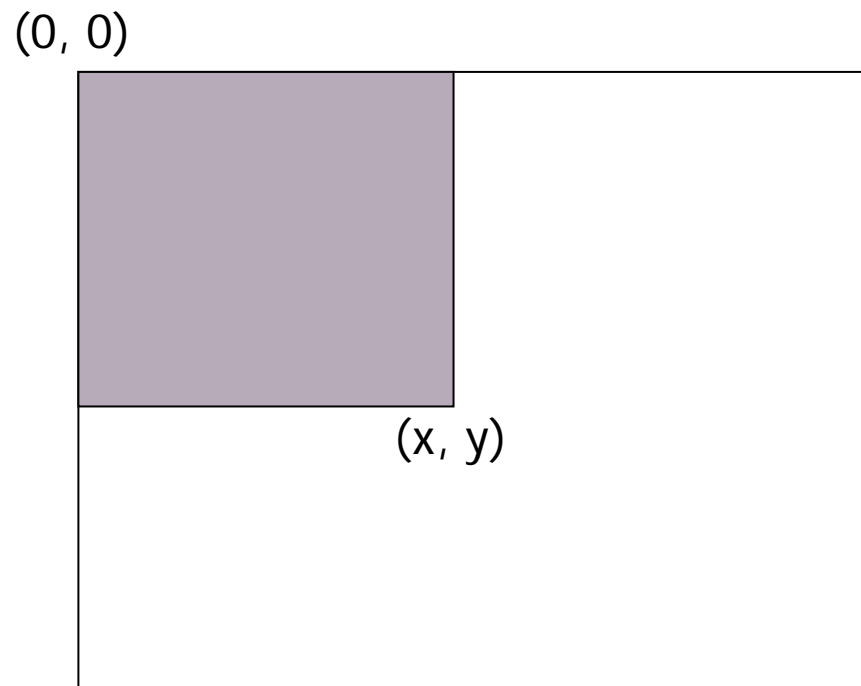
検出に有効か調査顔



# 内積計算の高速化

## Integral Image

- Integral Imageとは矩形領域内の画素値の総和を画素値とする画像



$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

$ii(x, y)$ : integral image

$i(x, y)$ : 画素

$s(x, y)$ : 縦の画素の総和

$$s(x, y) = s(x, y-1) + i(x, y)$$

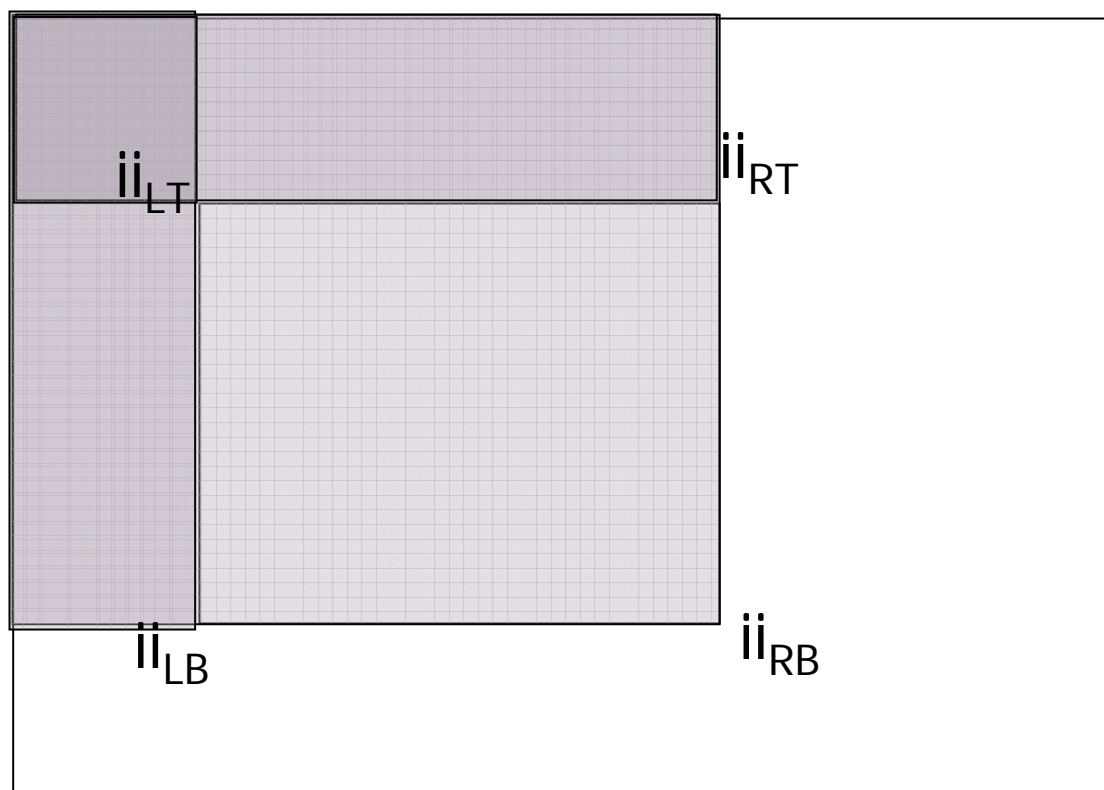
$$ii(x, y) = ii(x-1, y) + s(x, y)$$

$$s(x, -1) = 0, ii(-1, y) = 0$$

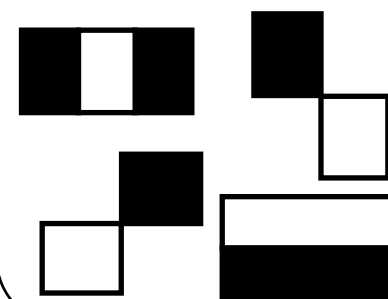
# Integral Imageを用いた矩形内積分

$$ii_{RB} - ii_{RT} - ii_{LB} + ii_{LT}$$

(0, 0)



Harr Like特徴



わずか3回の加減算で、任意の矩形領域の積分が行え、顔検出で用いられる  
Harr Like特徴も高速に計算可能

# 走査方法

## 検出用

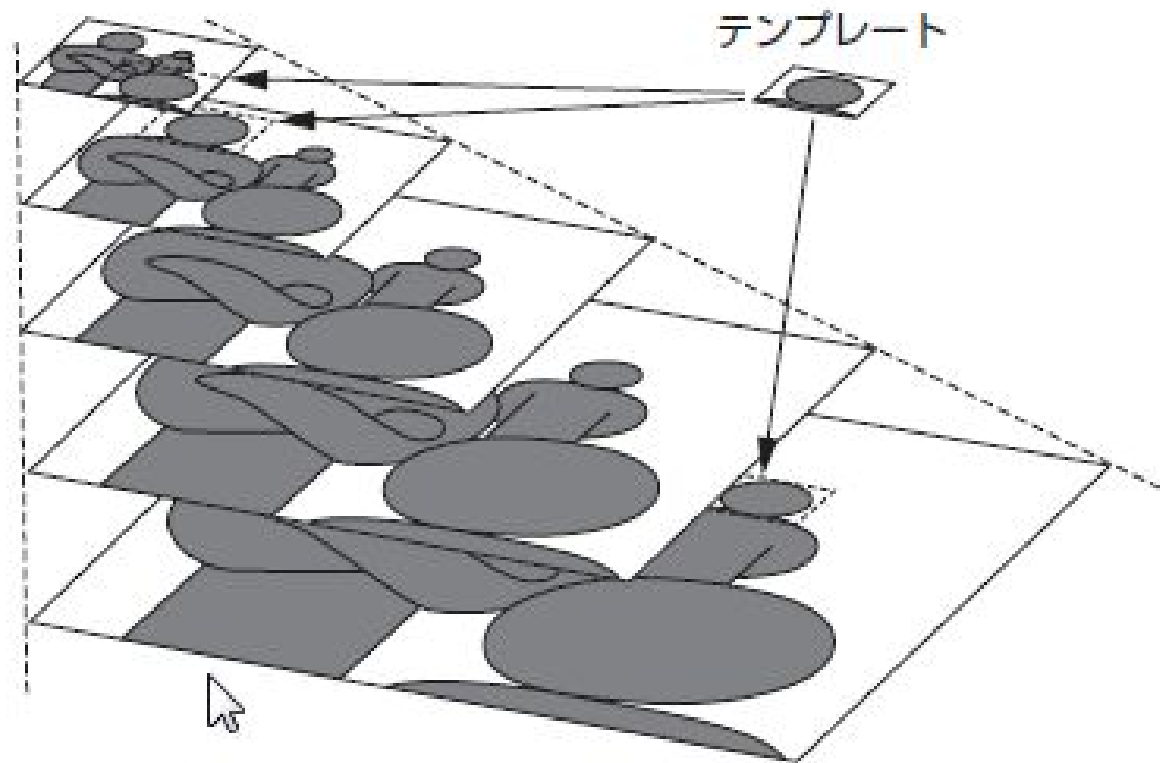
- ラスタースキャン
- ピラミッドスキャン

## 追跡用

- 局所スキャン
- 山登り法
- ランダム(重点)サンプリング

# 検出時の位置・大きさ の変化に対する対処法

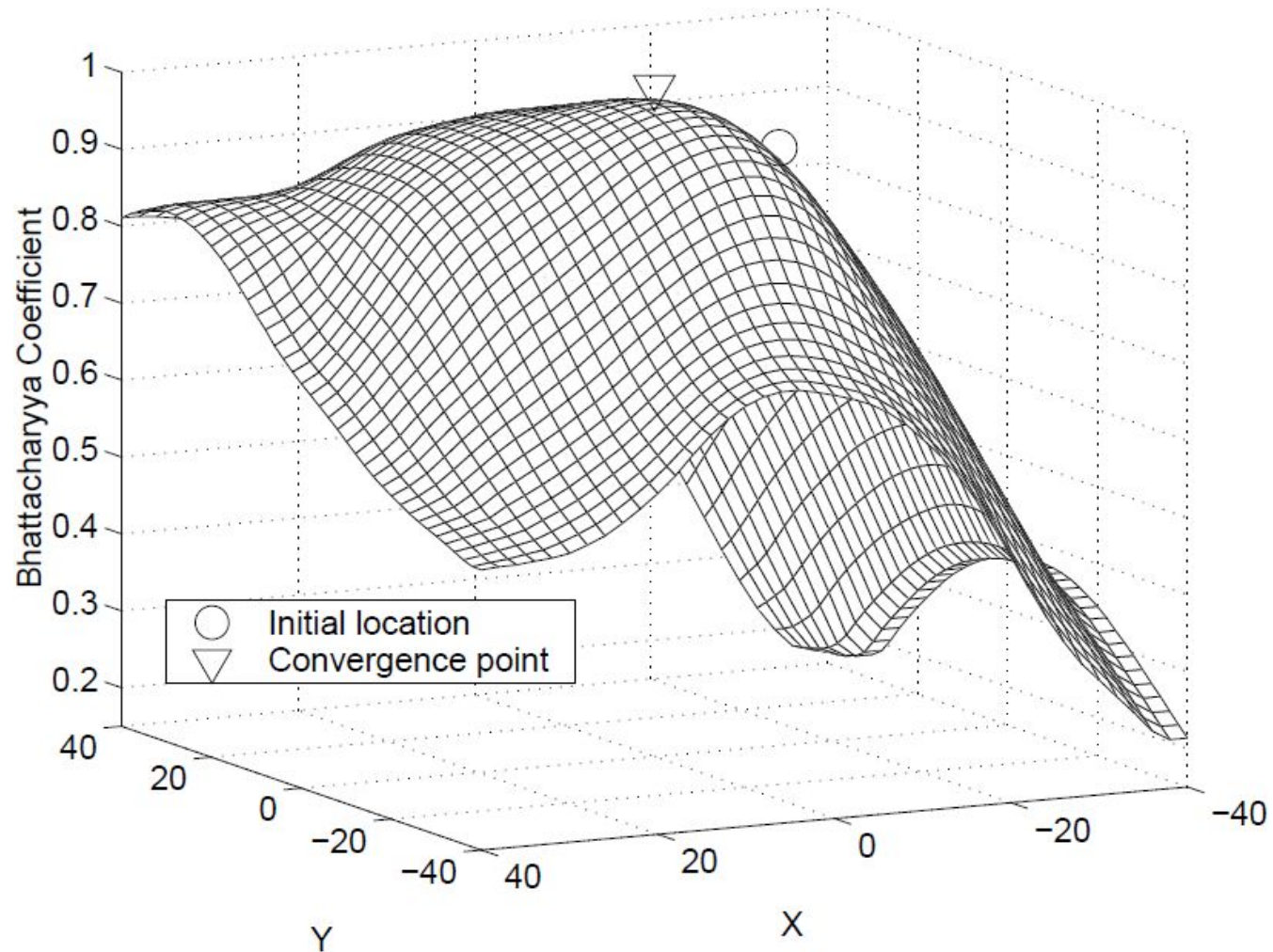
- イメージピラミッドの走査



# Meanshift: 山登り



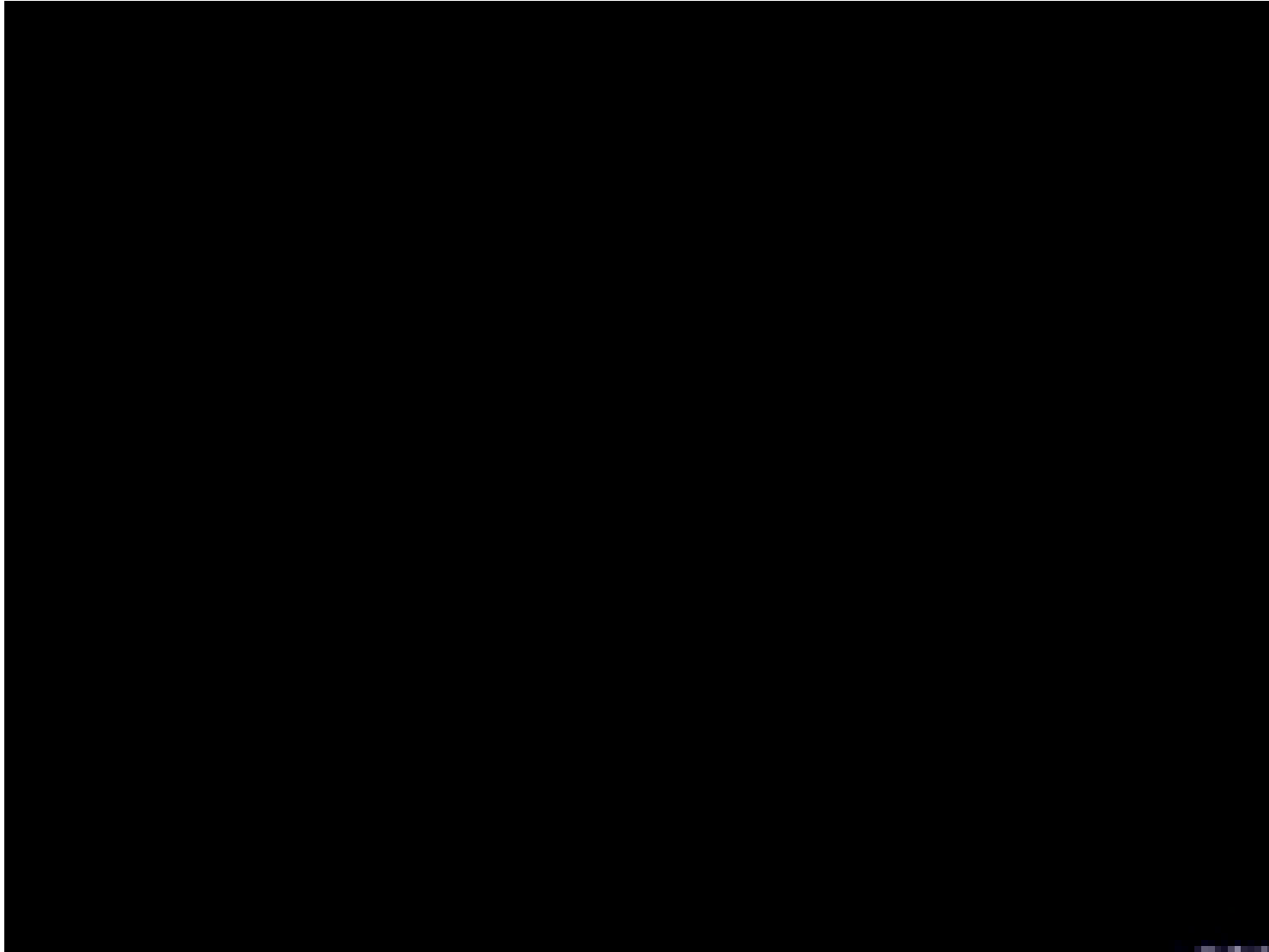
# Meanshift: 原理



# Particle Filter: ランダムサンプリング



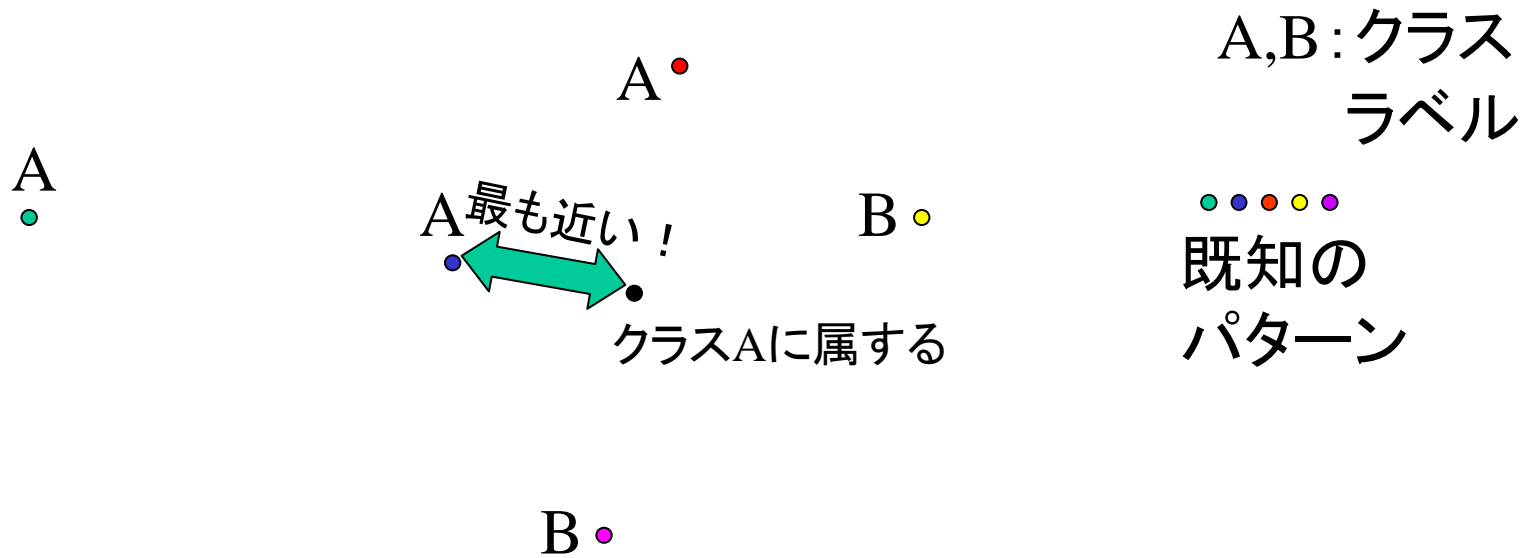
# Particle Filter:原理



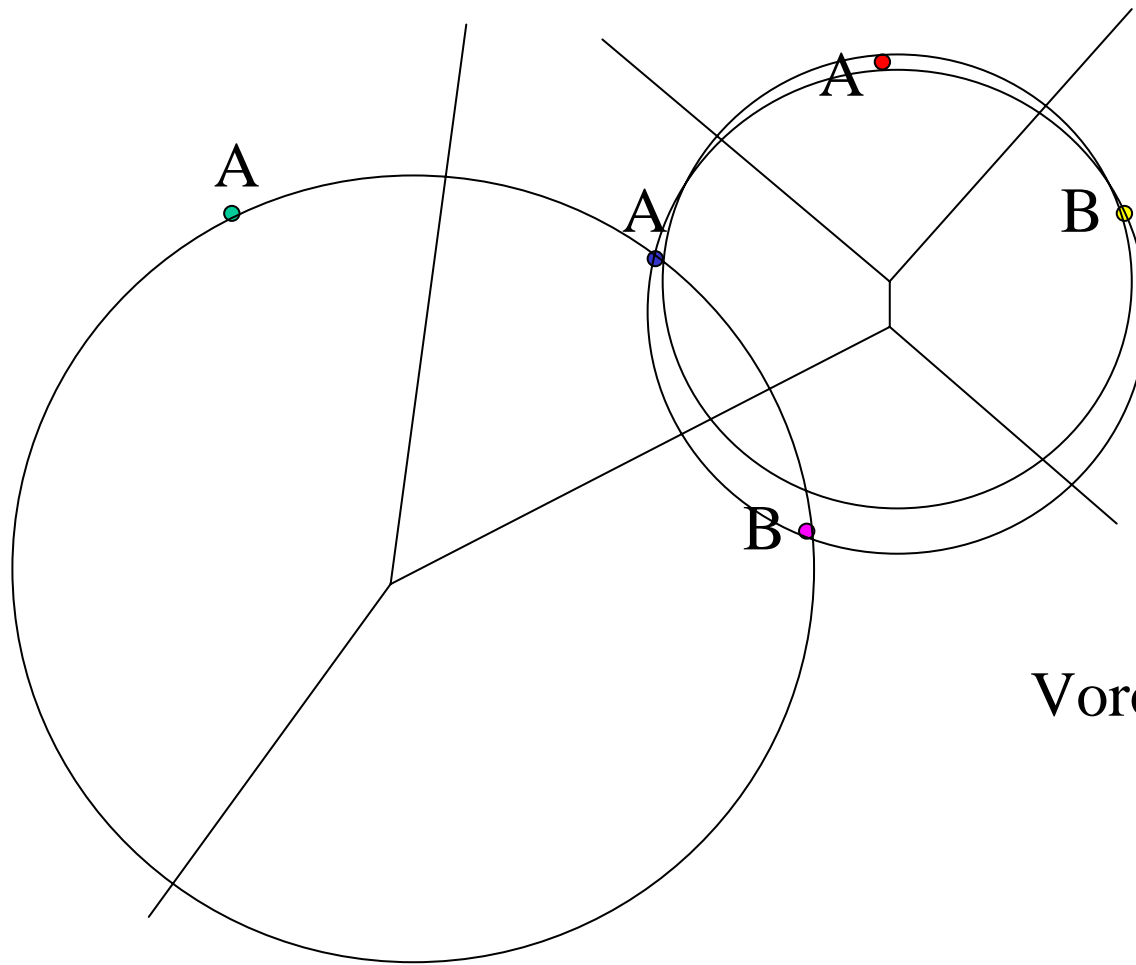


# 和歌山大VRLでの研究紹介

# 検出：最近傍識別の基本

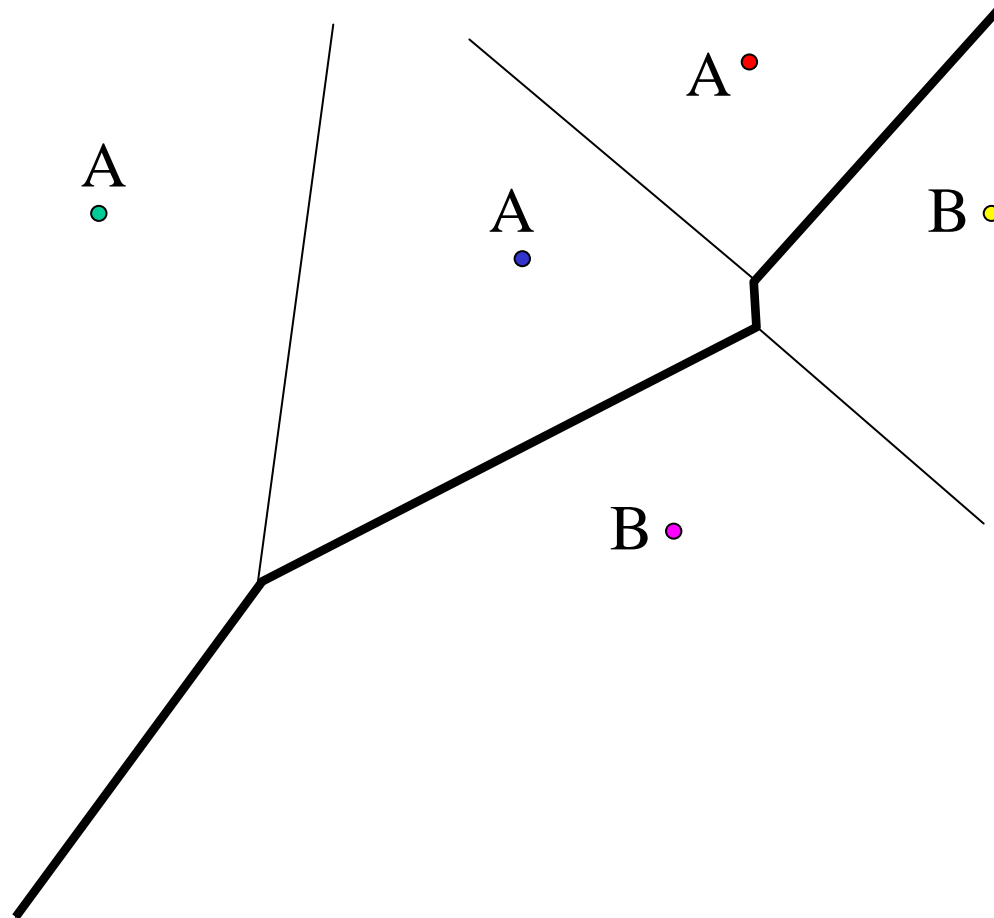


# 最近傍識別における識別境界



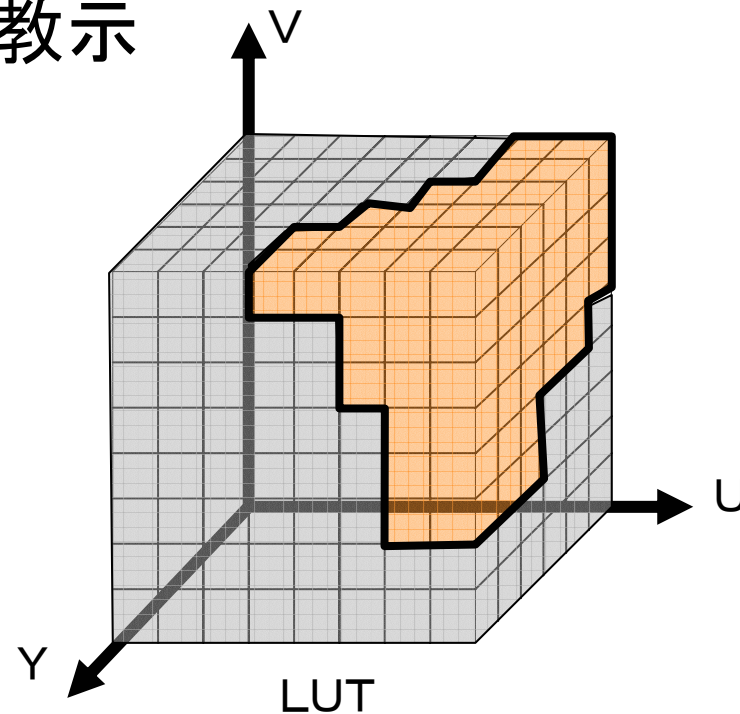
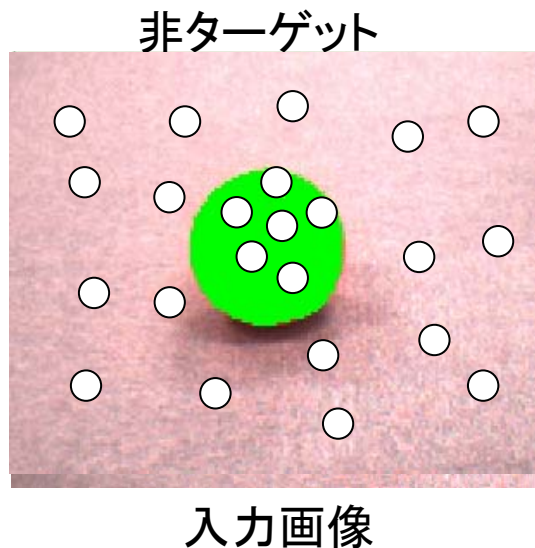
Voronoi分割

# 最近傍識別における識別境界



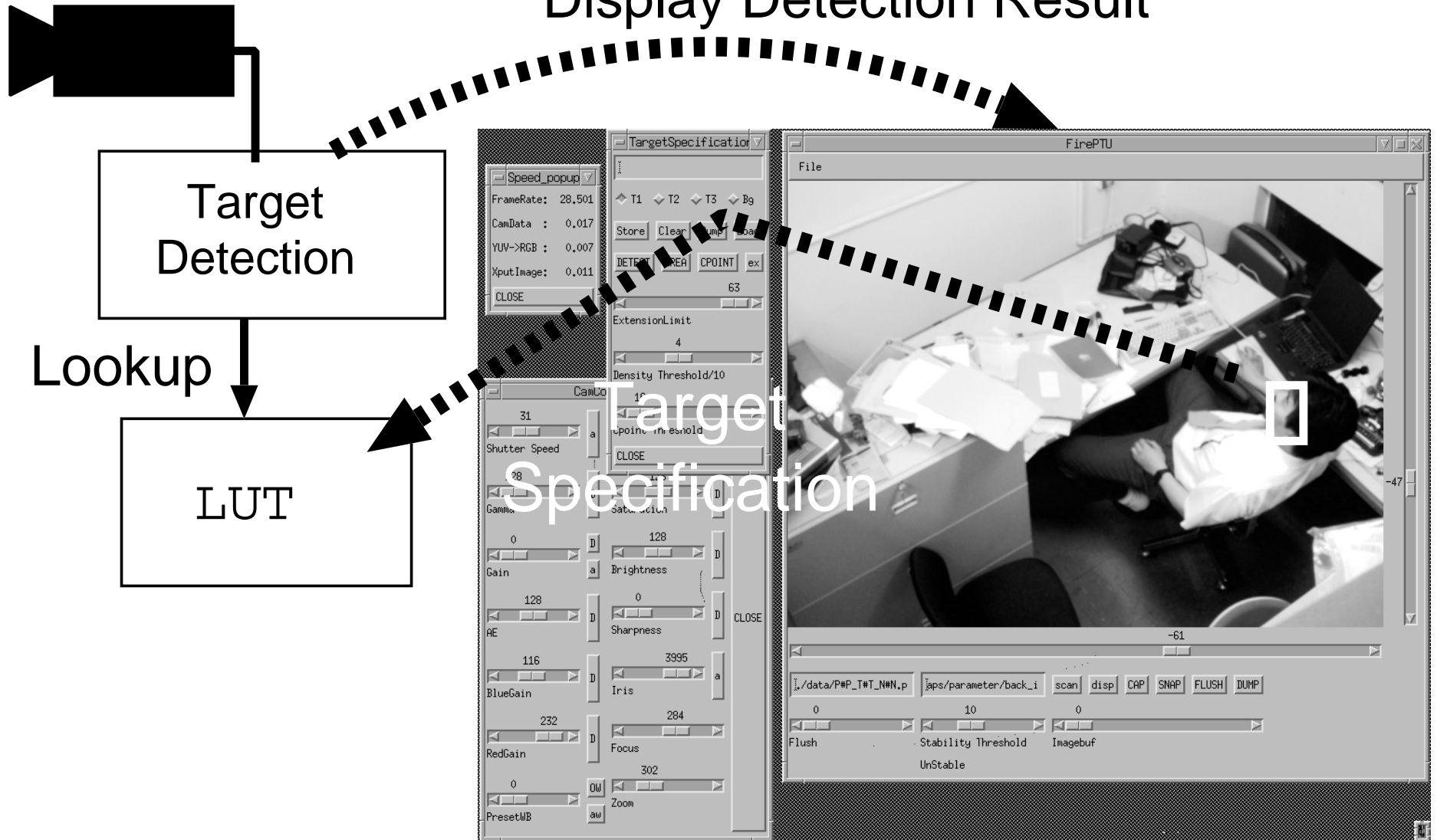
# 最近傍識別器による色ターゲット検出

- 画像の各画素をターゲット・非ターゲットに識別
- LUTを用いることで実時間での検出を実現
- インタラクティブな色教示



# 色ターゲット検出システム

Display Detection Result



# サンプリングした色を膨張させた検出結果



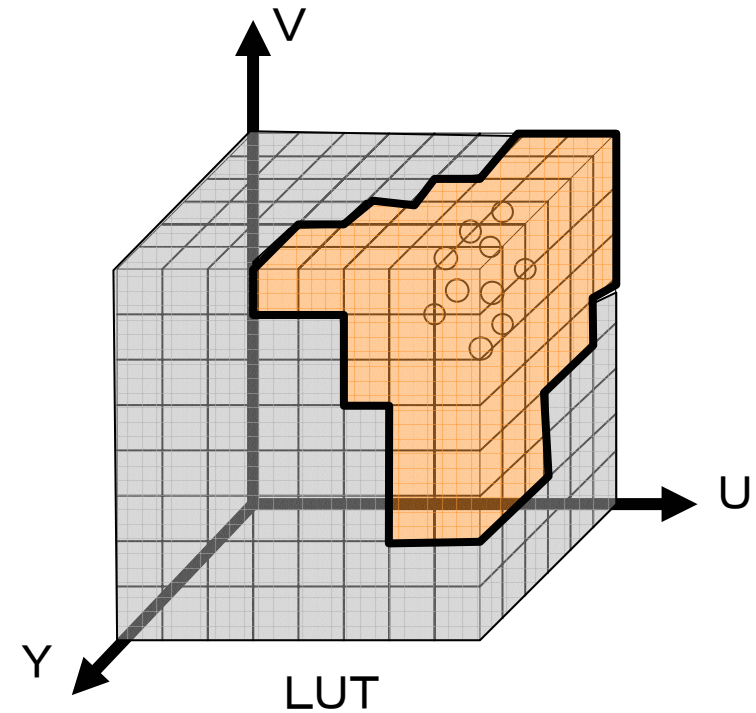
# 本手法の結果



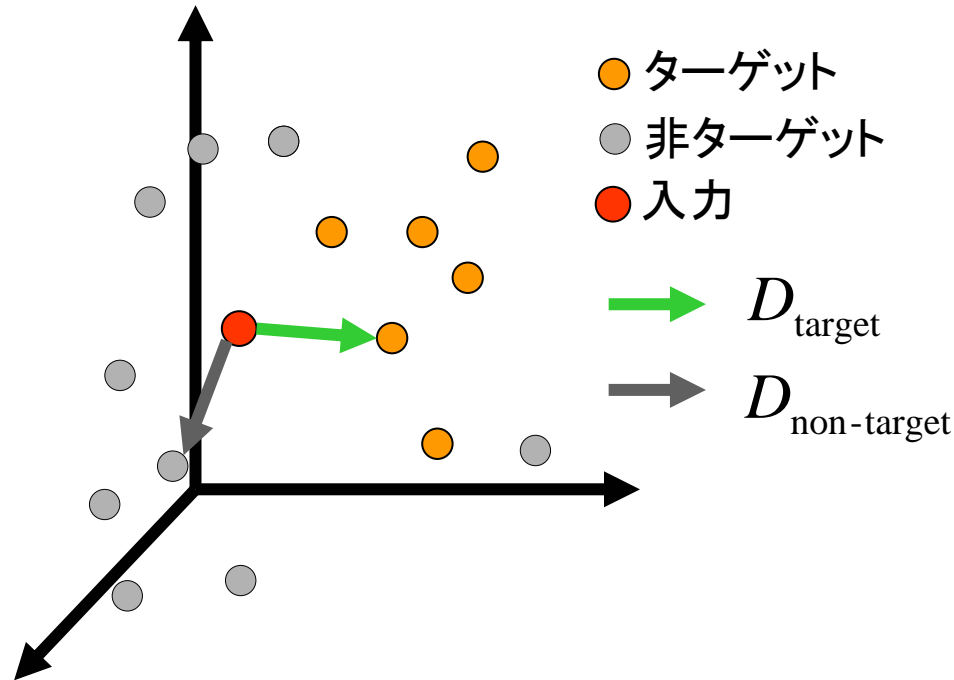


# 追跡における色ターゲット検出の問題点

- 追跡における問題点
  - 照明の変動などによる色の変化に対して脆弱



# 提案手法：事例を用いた弁別度の算出



弁別度の算出式

$$d = \frac{D_{\text{non-target}}}{D_{\text{target}} + D_{\text{non-target}}}$$

- 特徴

- 対象プロトタイプ上で最大値1, 非対象上で0
- [0,1]の範囲の値
- 0.5がちょうど最近傍識別での識別境界
- 識別境界に面している非対象プロトタイプから最近傍の対象プロトタイプに至るパス上で値が単調に増加

# 対象追跡

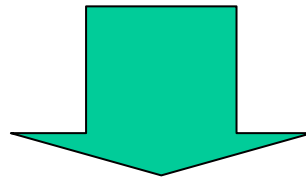
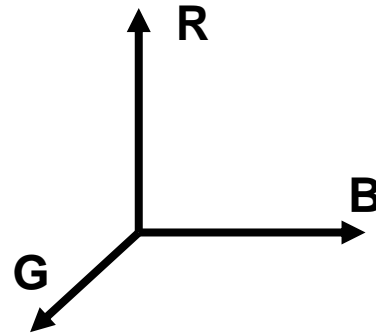
- 特徴空間

- 画素値 (RGB)

- 求められる性能

- リアルタイム性

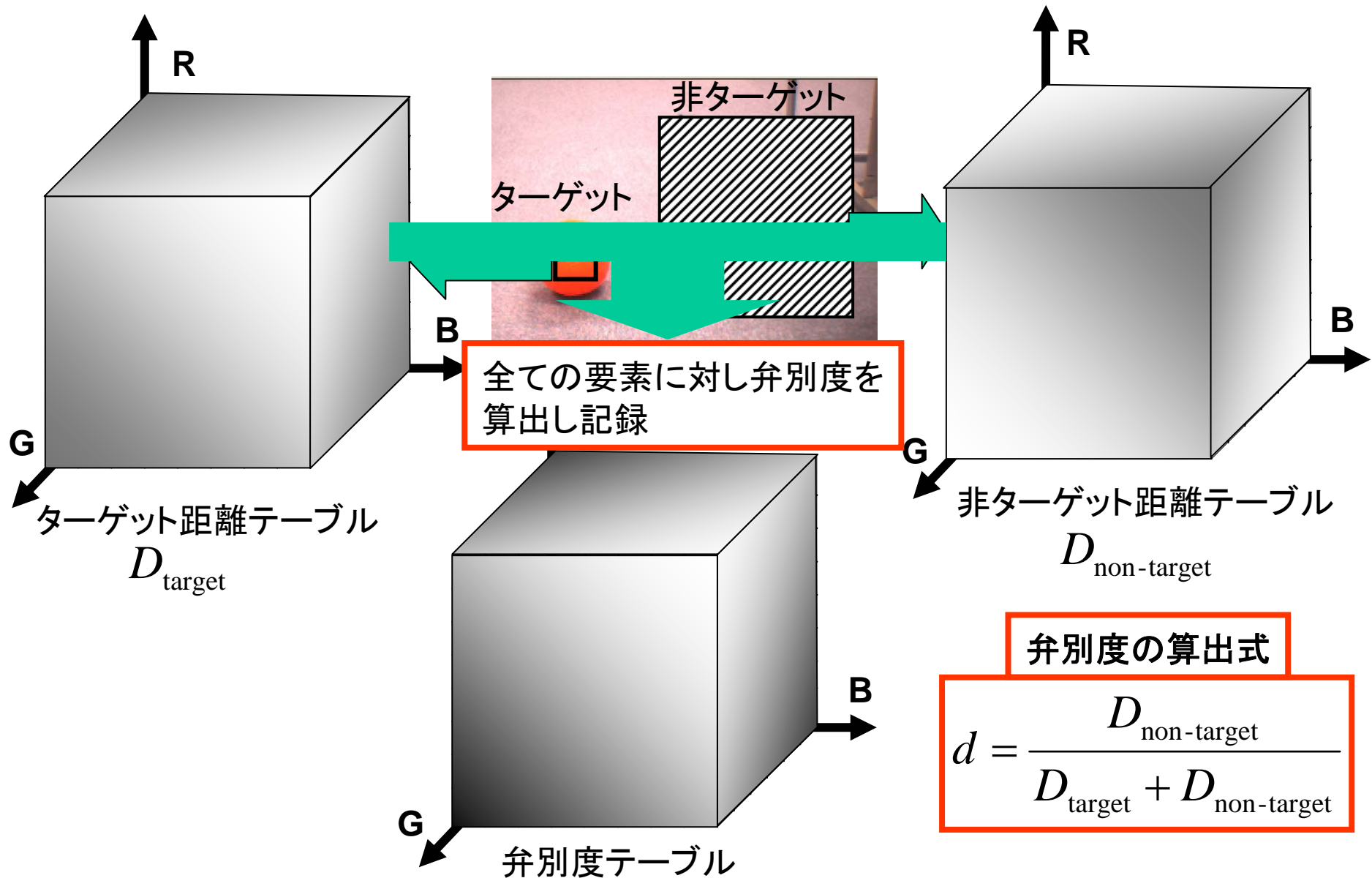
- 全ての入力画素に対し算出式を適用していたのでは多くの処理時間が必要となる



- 弁別度算出の高速化

- LUTを用いることでメモリ参照のみで弁別度を測ることが可能

# LUT構築方法



# 実験: 弁別性マップの構築

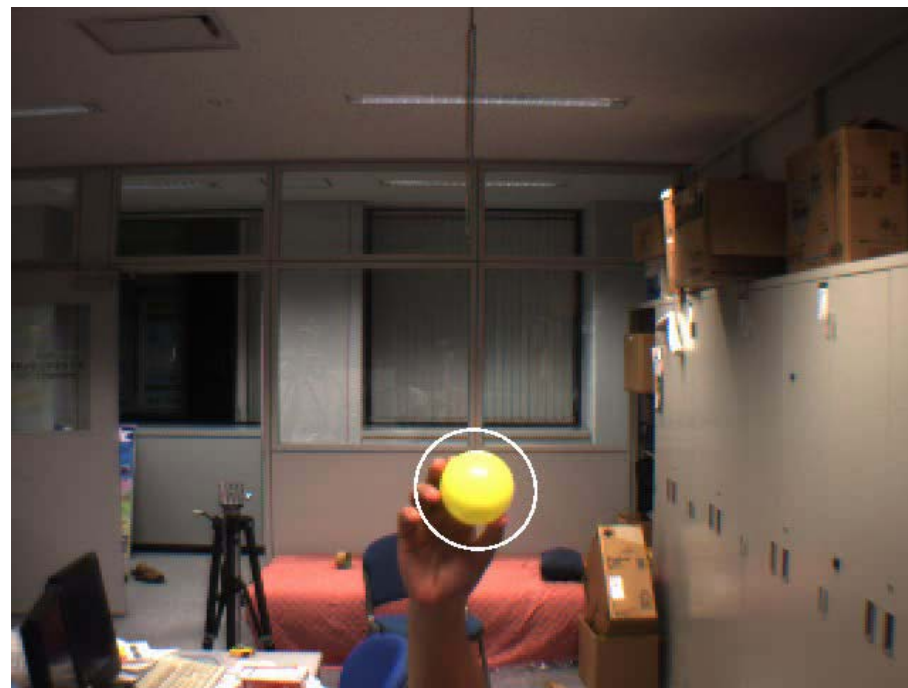
- 実験環境

- 計算機: Intel Pentium4 3.0GHz
- OS: Windows XP
- コンパイラ: Intel C++ Compiler ver9.0
- 入力画像: 640x480 RGB
- 入力パターン: 各要素6bit(64段階)
- LUTサイズ: 64x64x64

# 対象追跡実験

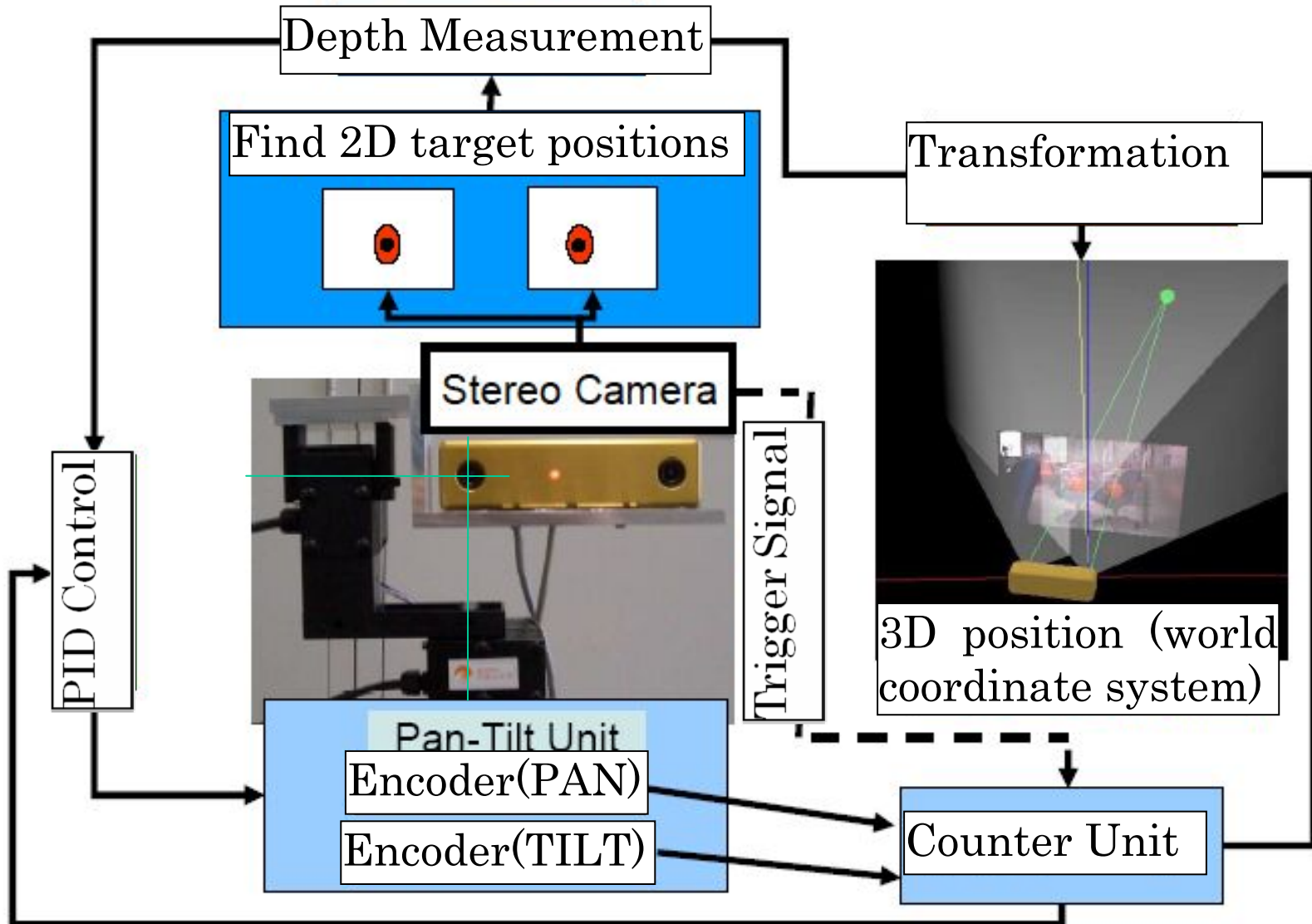


色ターゲット検出



提案手法

# 実時間能動ステレオ追跡システム

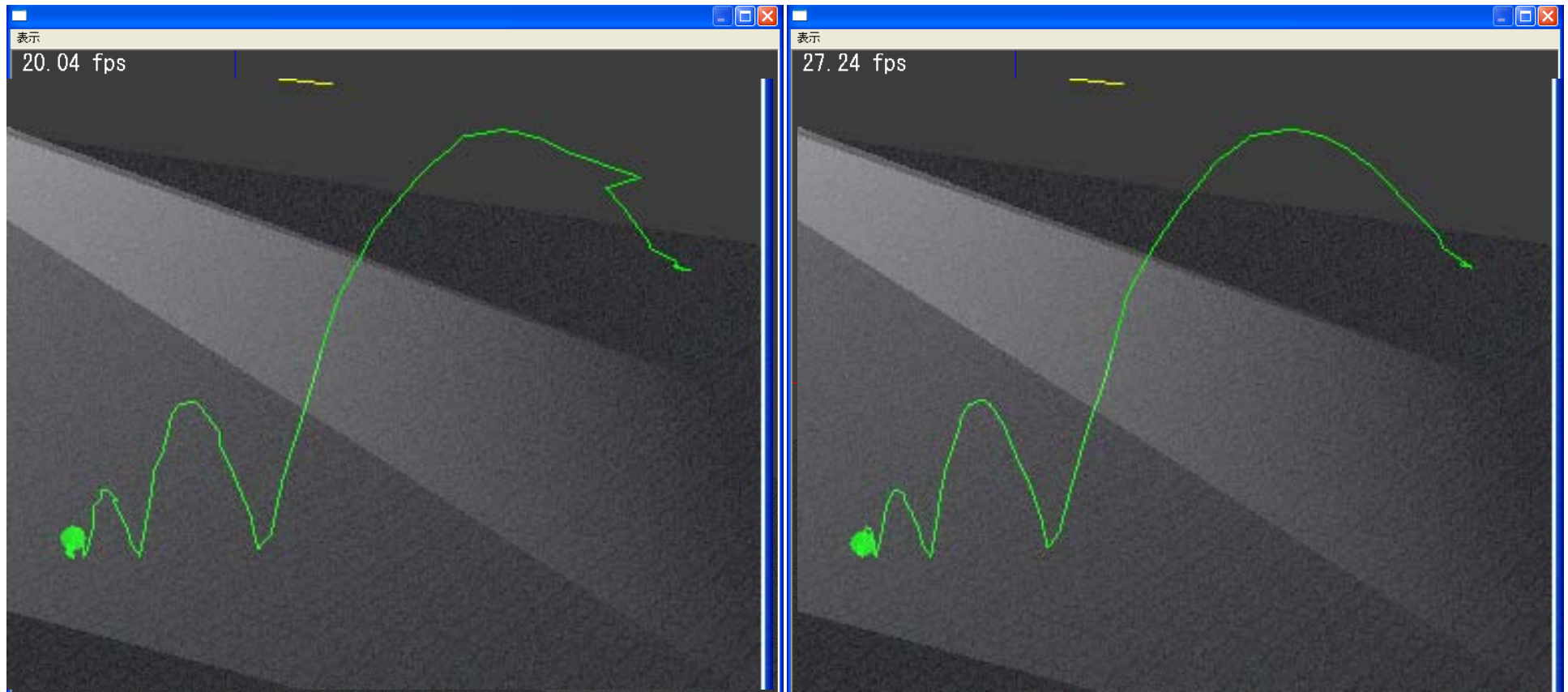


# ボールの3次元位置計測(提案手法)





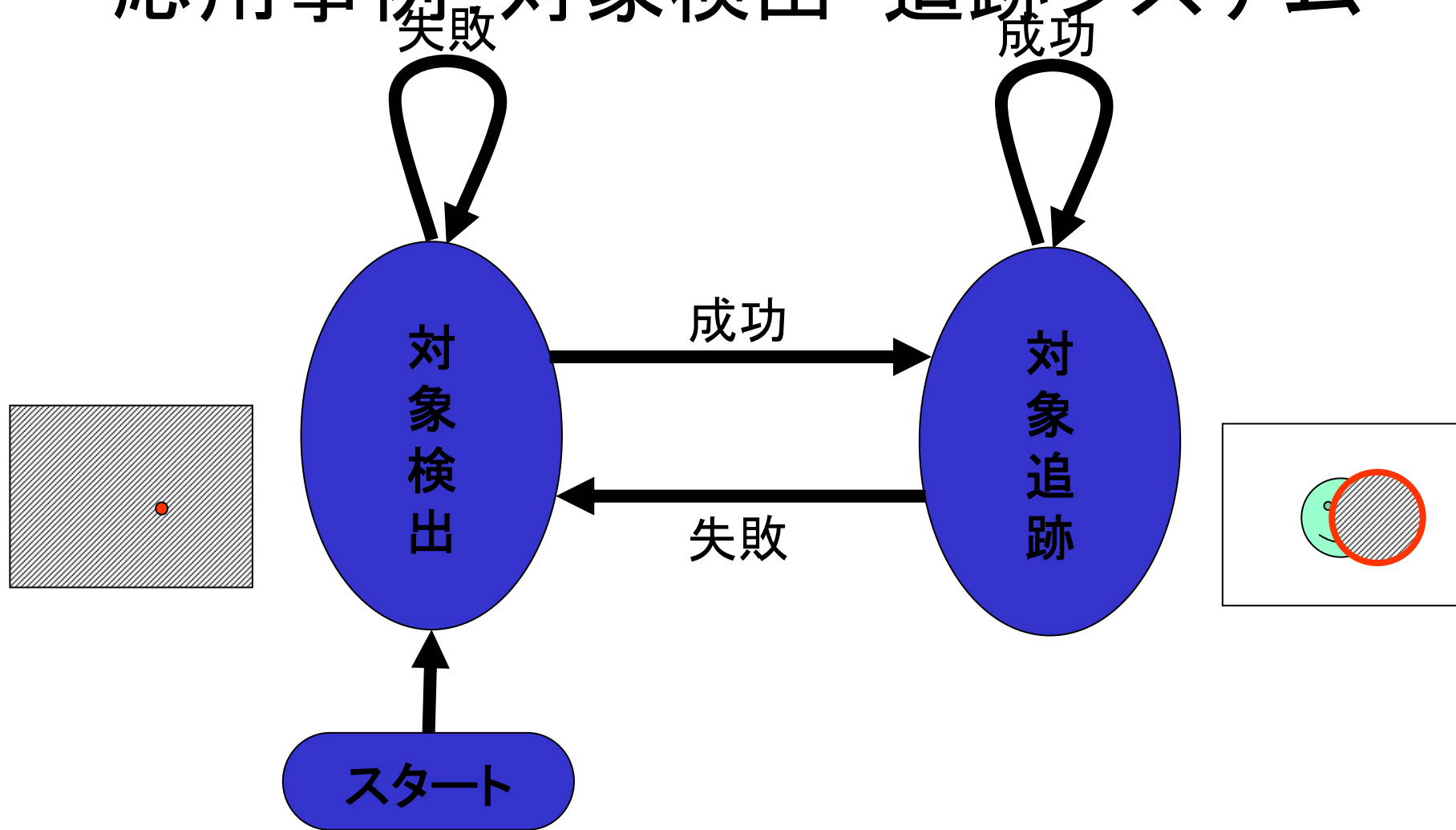
# 3次元計測結果の比較



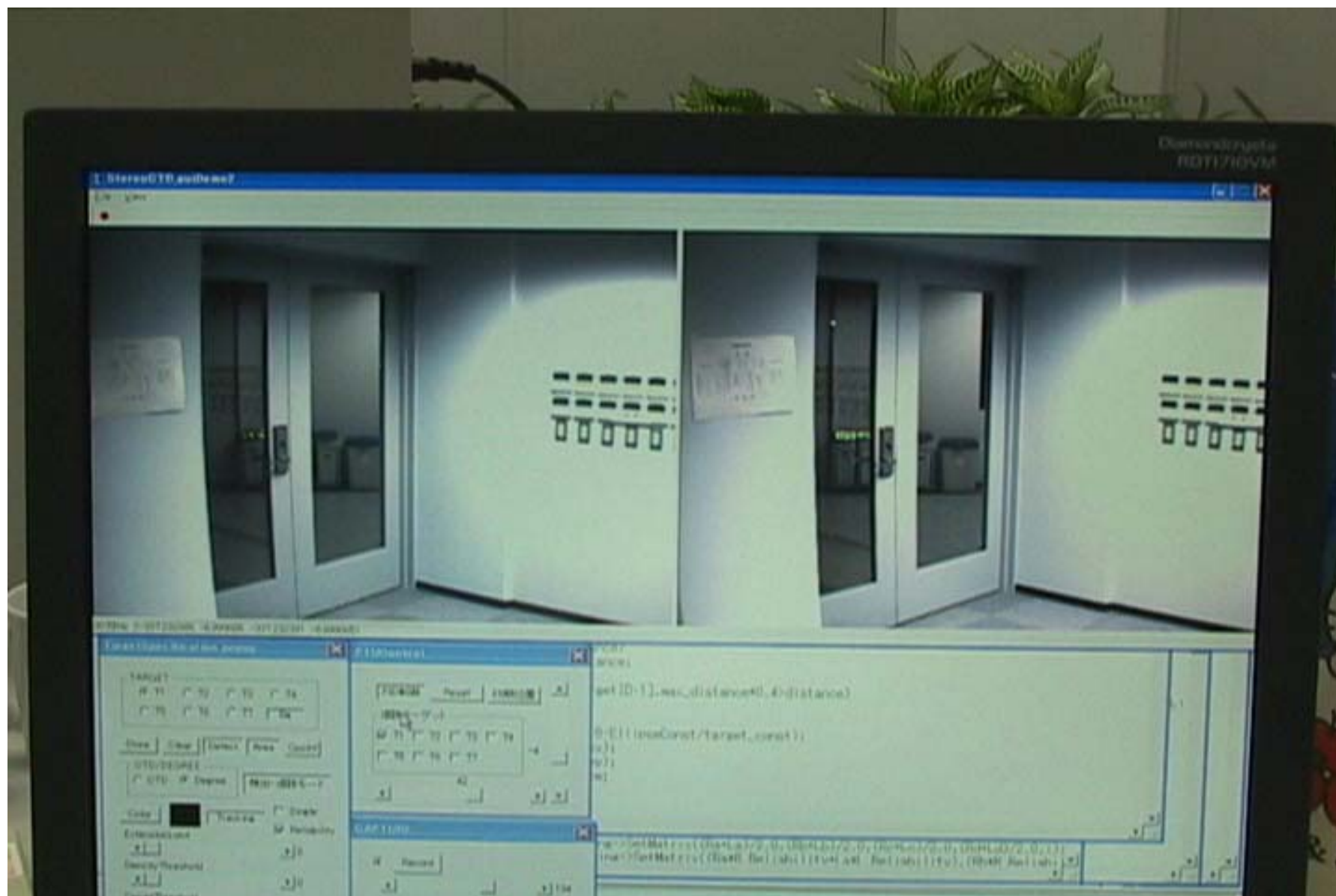
色ターゲット検出

提案手法

# 応用事例・対象検出・追跡システム

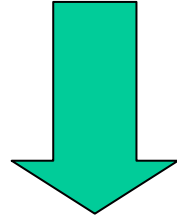


# 対象検出・追跡システム



# 正と負の事例

- 負事例をいちいち指定するのは面倒.
- 正事例も時々刻々と変化する.



- 一旦事例を与えれば、自動的に事例を更新してくれる仕組みが必要.

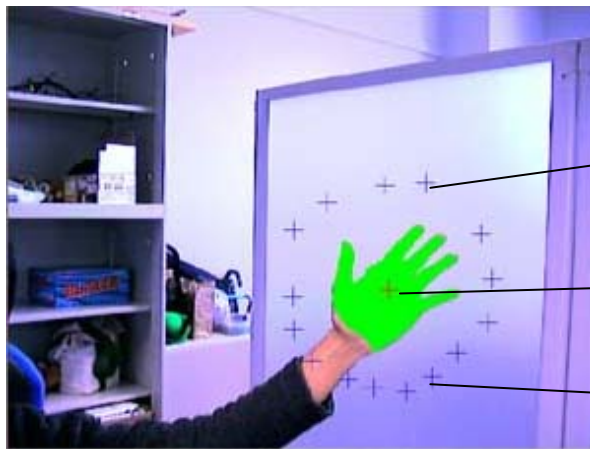
# K-means Trackerの基本原理

■ ターゲットと非ターゲットの情報を  $(R,G,B,x,y)$  の5次元ベクトルで表す

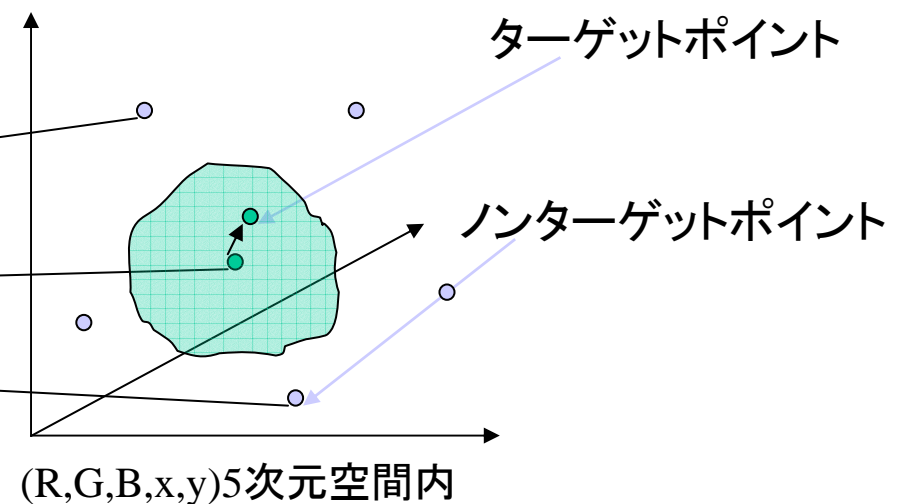
➡ 色情報と空間的な位置情報を同時に取り扱う

■ 5次元空間内でK-meansクラスタリングを行う

➡ 背景とターゲットを区別して追跡



画像



# 提案手法の特長

■ターゲットを(R,G,B,x,y)5次元空間内の点集合で表現している

➡ ・ターゲット画素位置の重心の追跡(x,y)

➡ ・ターゲット画素の色の追跡(R,G,B)

・単一の処理で追跡とターゲットモデルの更新を同時に行うことができる

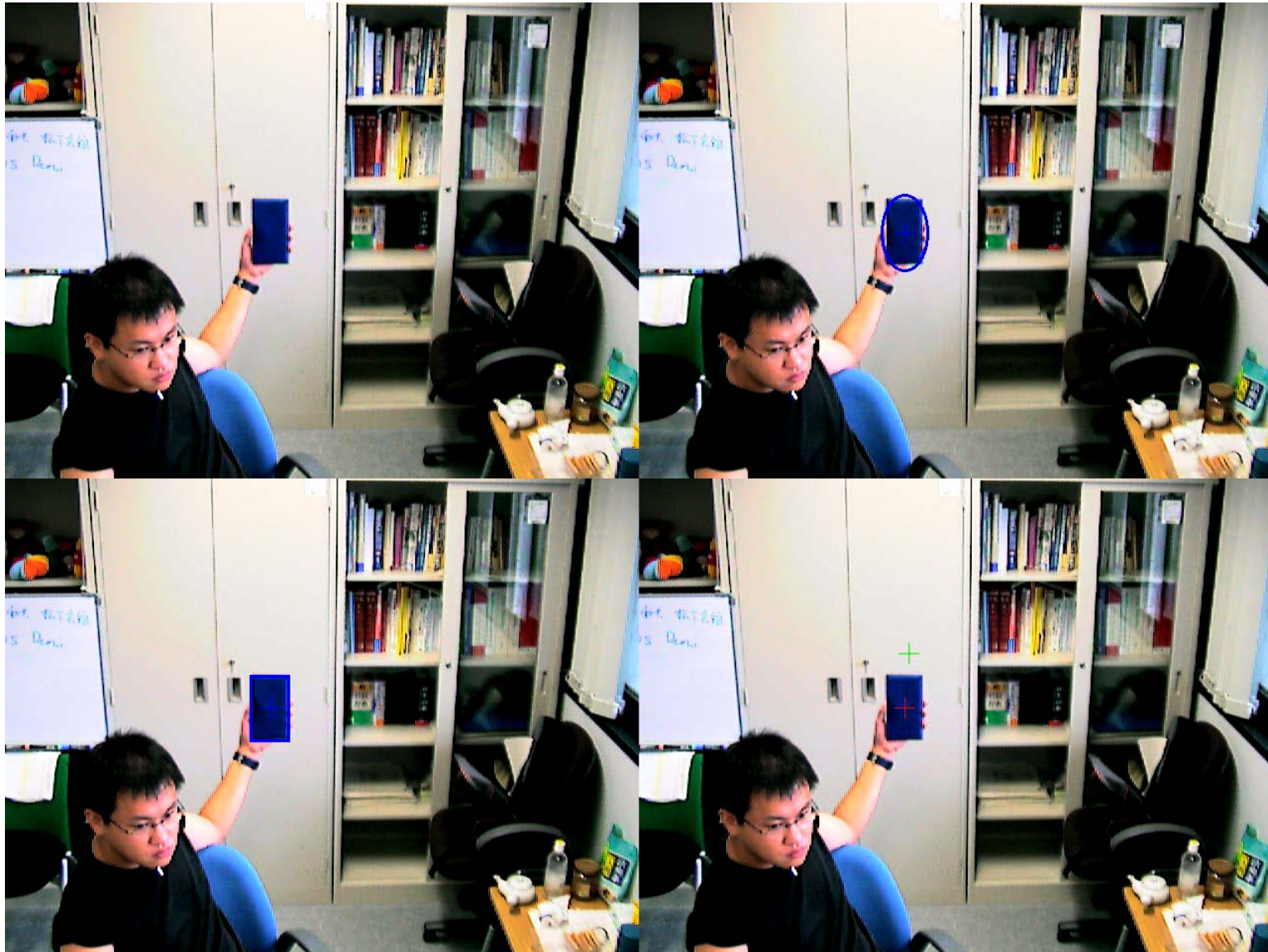
■K-meansクラスタリングによる画像のクラスタリング

➡ ・非ターゲットとターゲットを区別して追跡ができ、背景混入に対して頑健

# 実時間化+高機能化

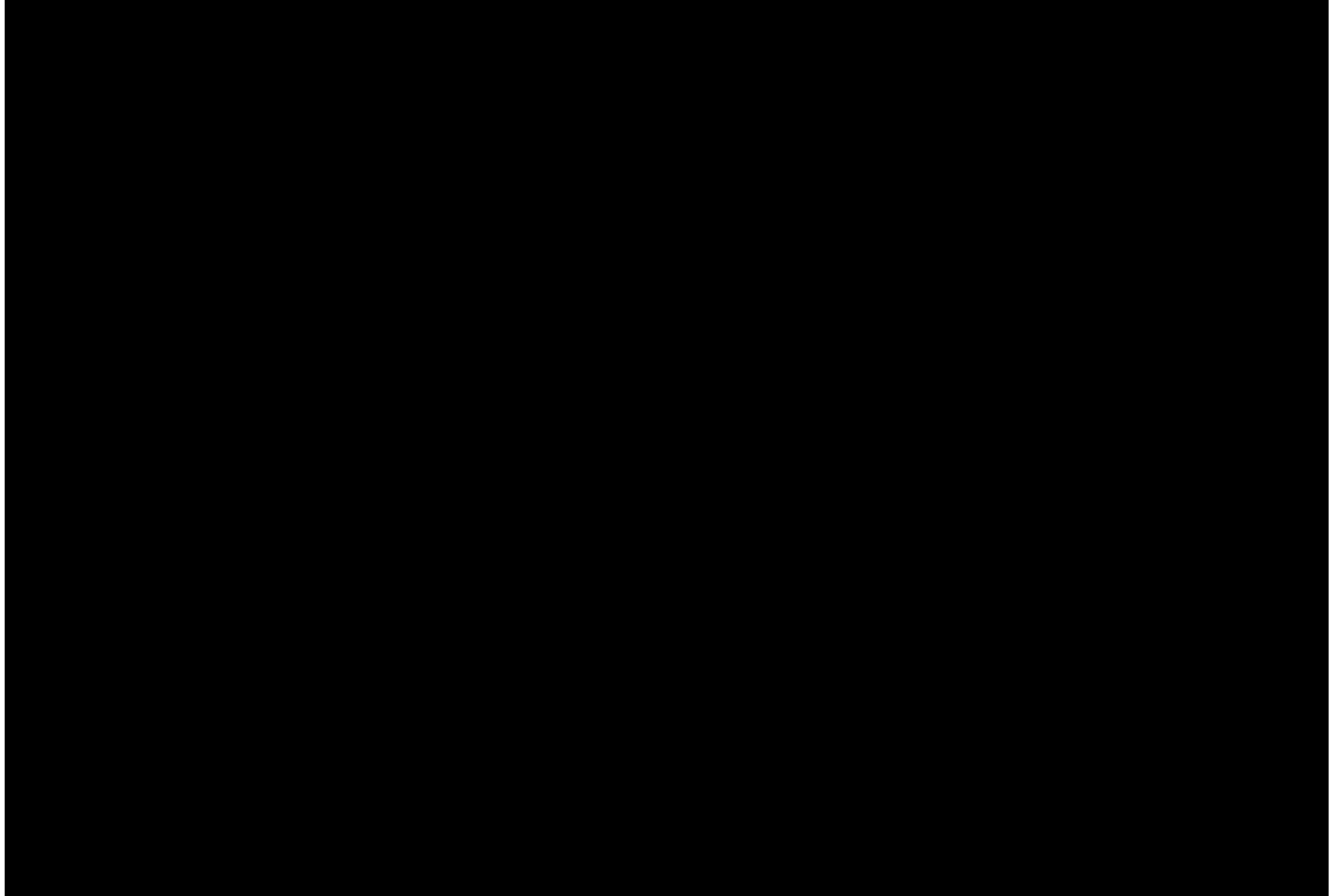
- 非ターゲットを楕円上の点から取る.
- ターゲットクラスタ内の画素にGaussian Fittingを行い, 楕円を求める.
- 色の近接性を用いて失敗の検出を行う.
- 間引きサンプリングを行う
- 複数のターゲット色に対応する

# Robustness against Monotone Color Shift





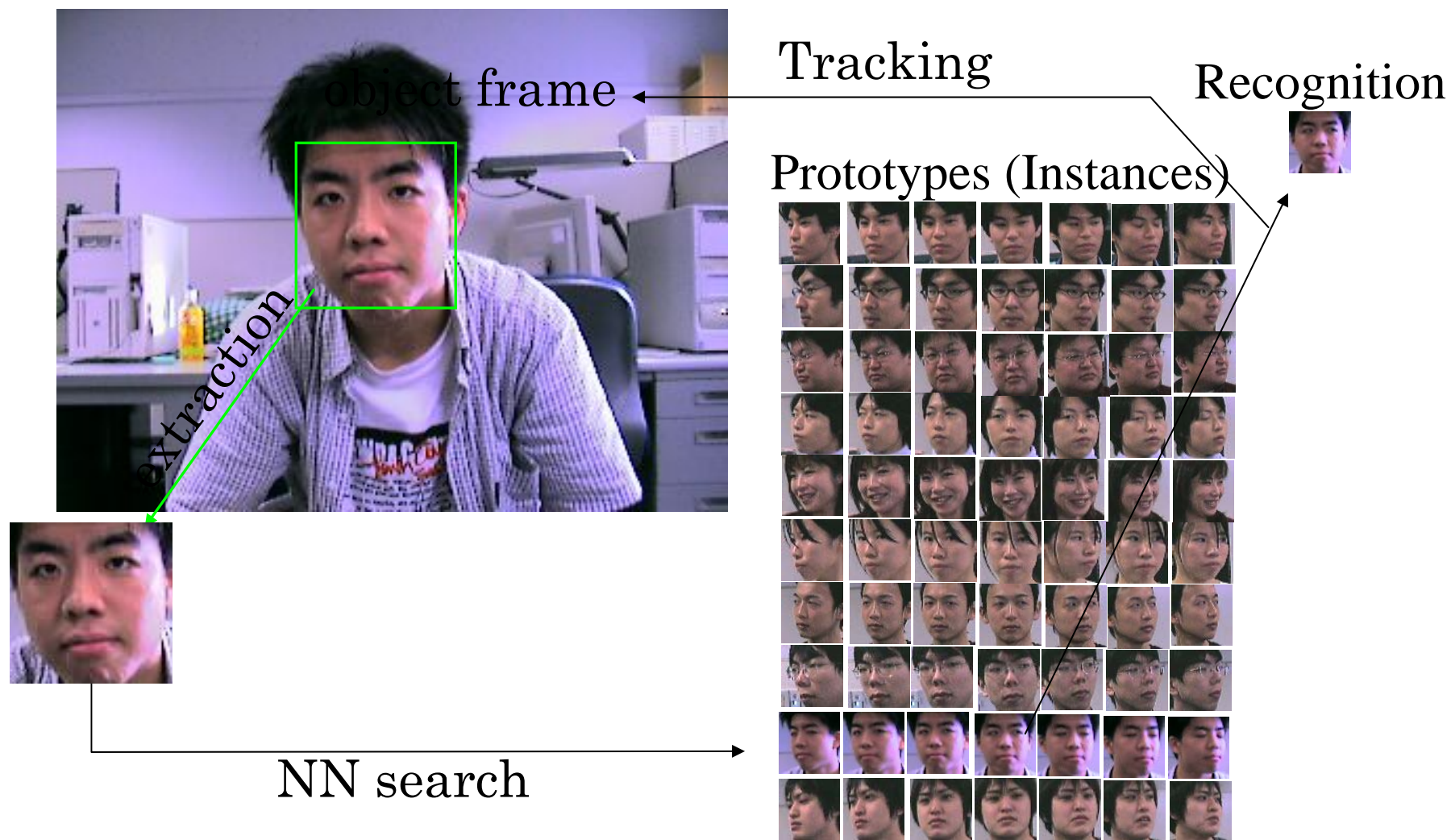
# Robustness against Textured Lighting



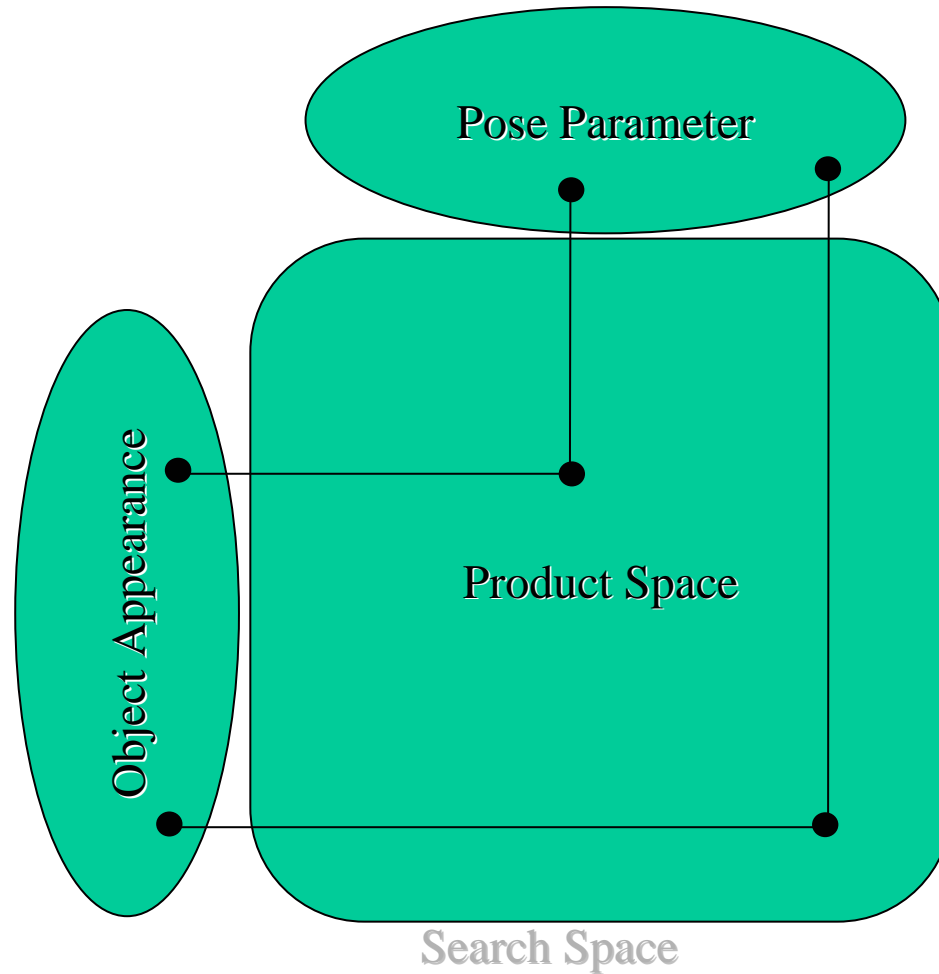




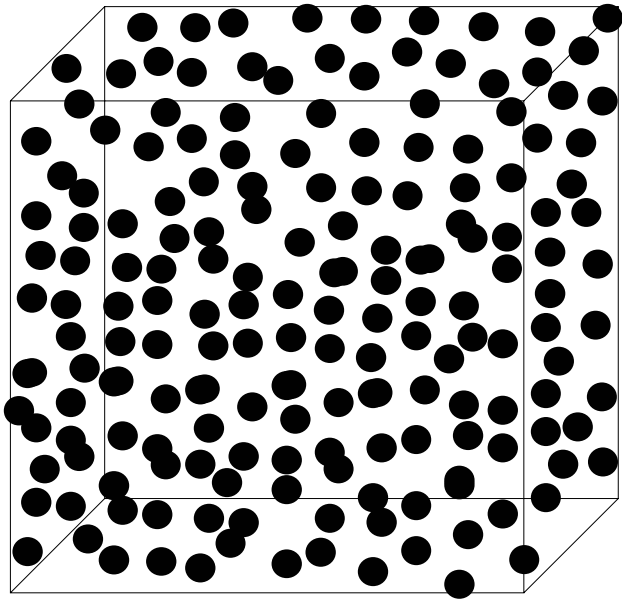
# 追跡と認識 (NFTG+PPE)



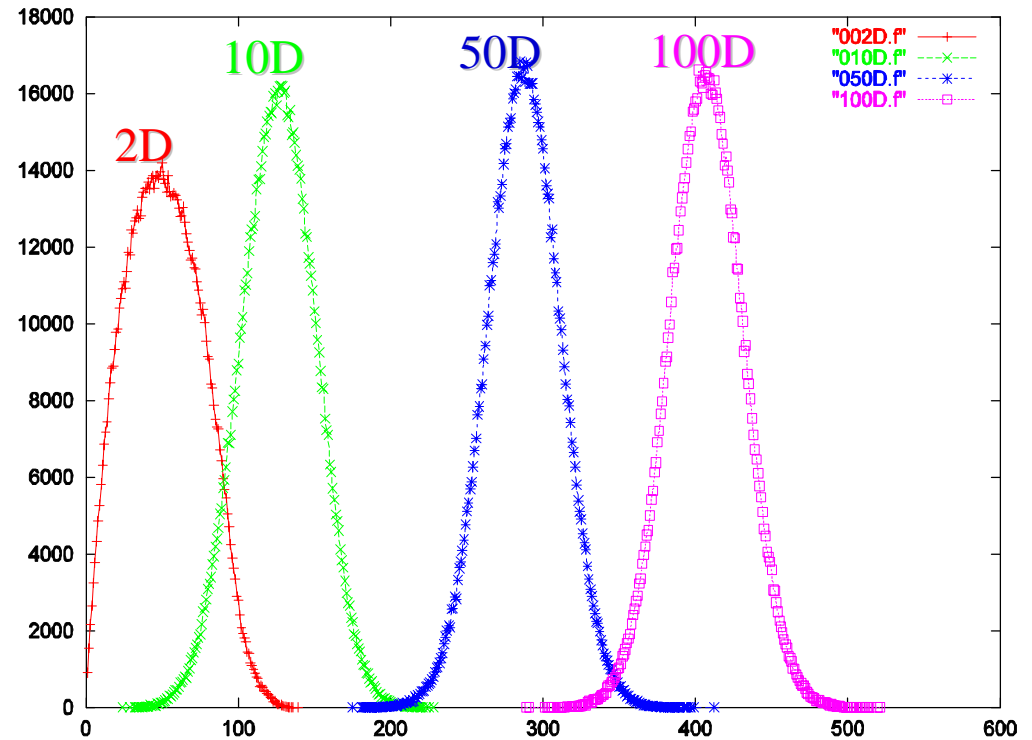
# Difficulty 1: Vast Search Space



# Difficulty2: Nearest Neighbor Search in High-dimensional Space



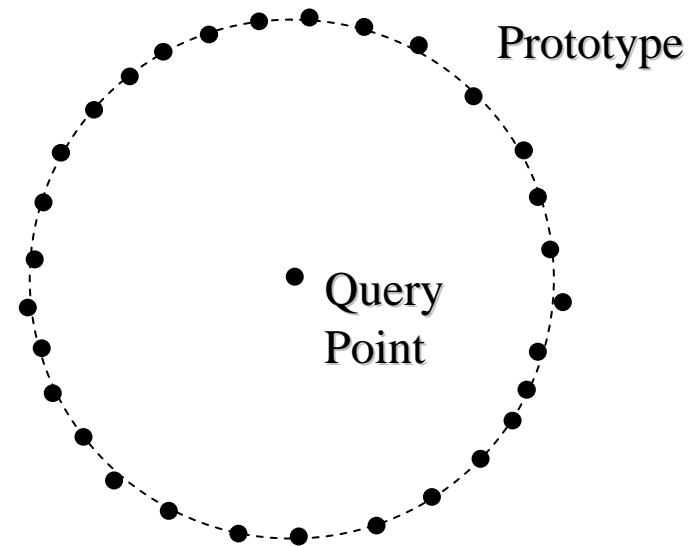
Uniform distribution  
in hypercube



Point distribution for mutual  
distance

# Difficulty2: Nearest Neighbor Search in High-dimensional Space (cont.)

Every Nearest Neighbor Search Algorithm becomes exhaustive search when the prototypes are located on a hyper-sphere and the query is given at its center.



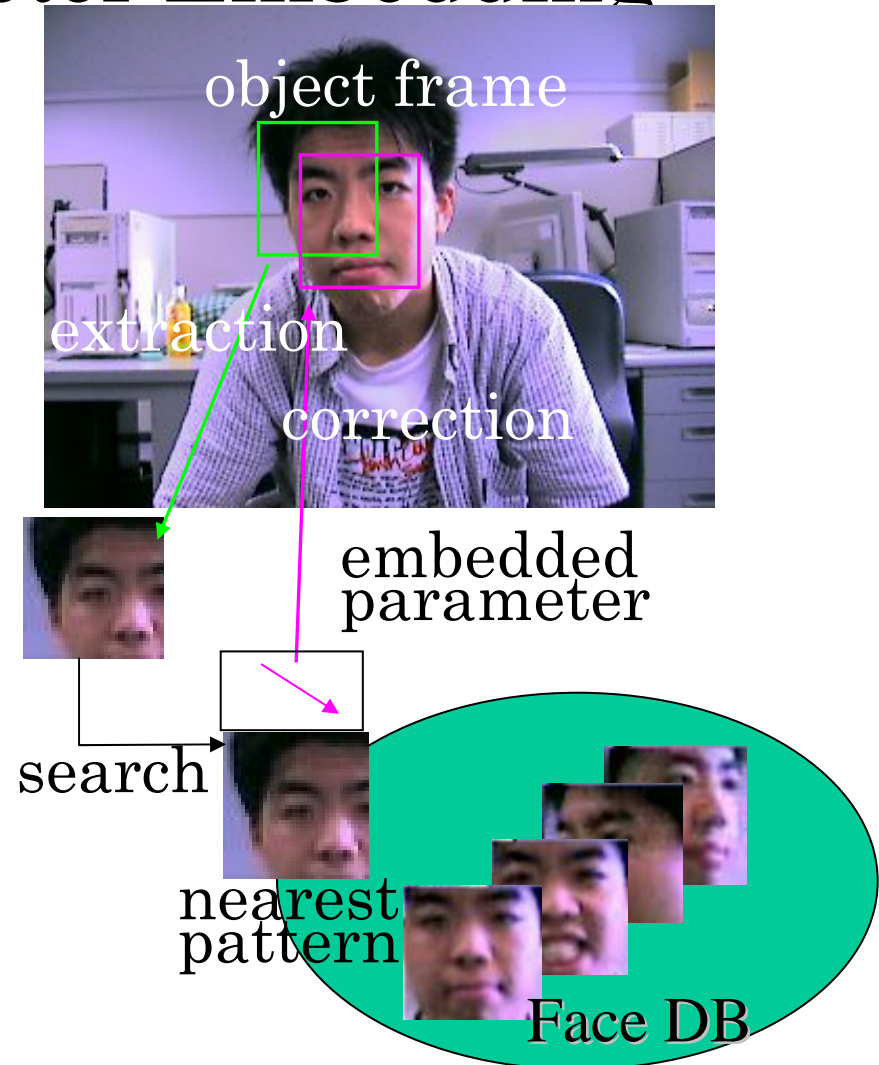
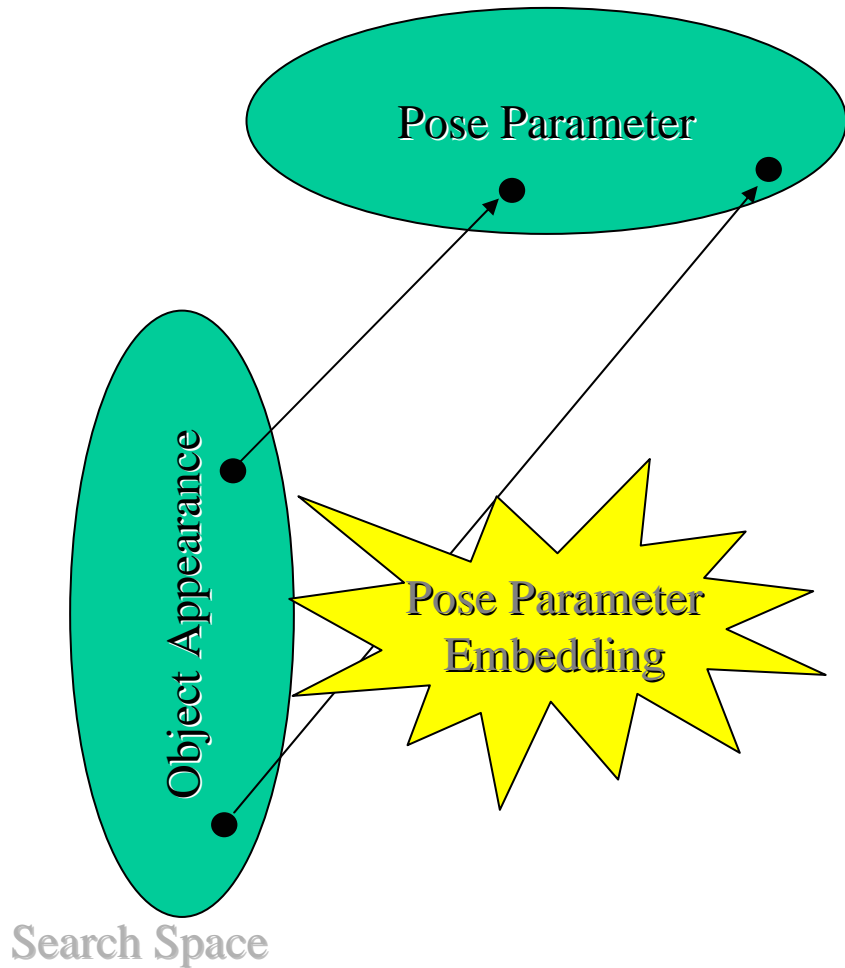
# Key Ideas

- Pose Parameter Embedding (PPE) for reducing the search space
- Nearest First Traversing Graph (NFTG) for continuous NN search in high-dimensional space



# Idea 1

## PPE: Pose Parameter Embedding



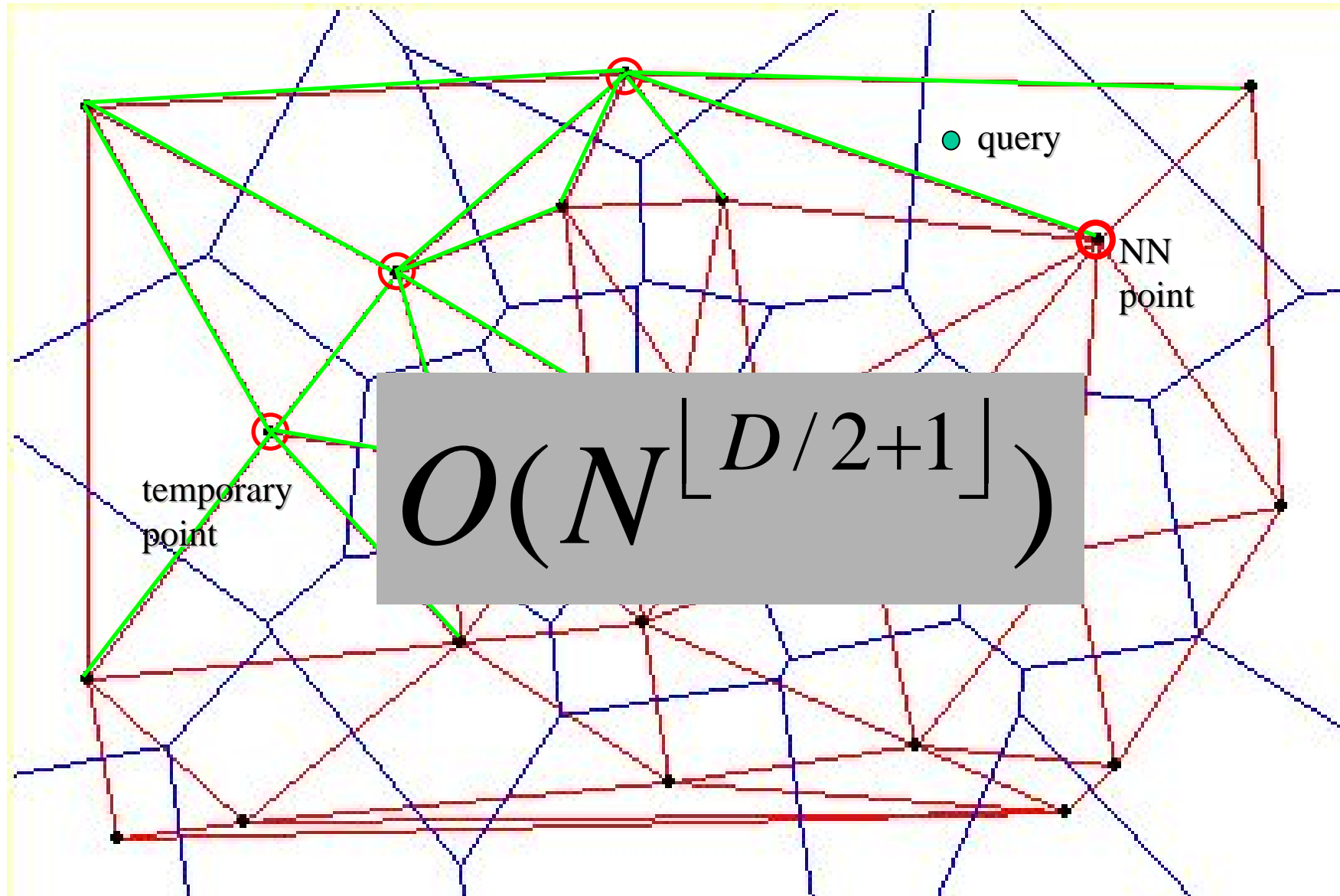
# Nearest Neighbor Search in high dimensional space.

- Tree based methods: VP-tree, MVP-tree, GNAT, S-tree, kd-tree, ANN, ....
- Table lookup methods: AESA, LAESA
- Graph based method: Delaunay Graph (Nearest First Traversing)



Graph based method has an advantage in the tracking task, because the previous NN search result can be utilized for next search.

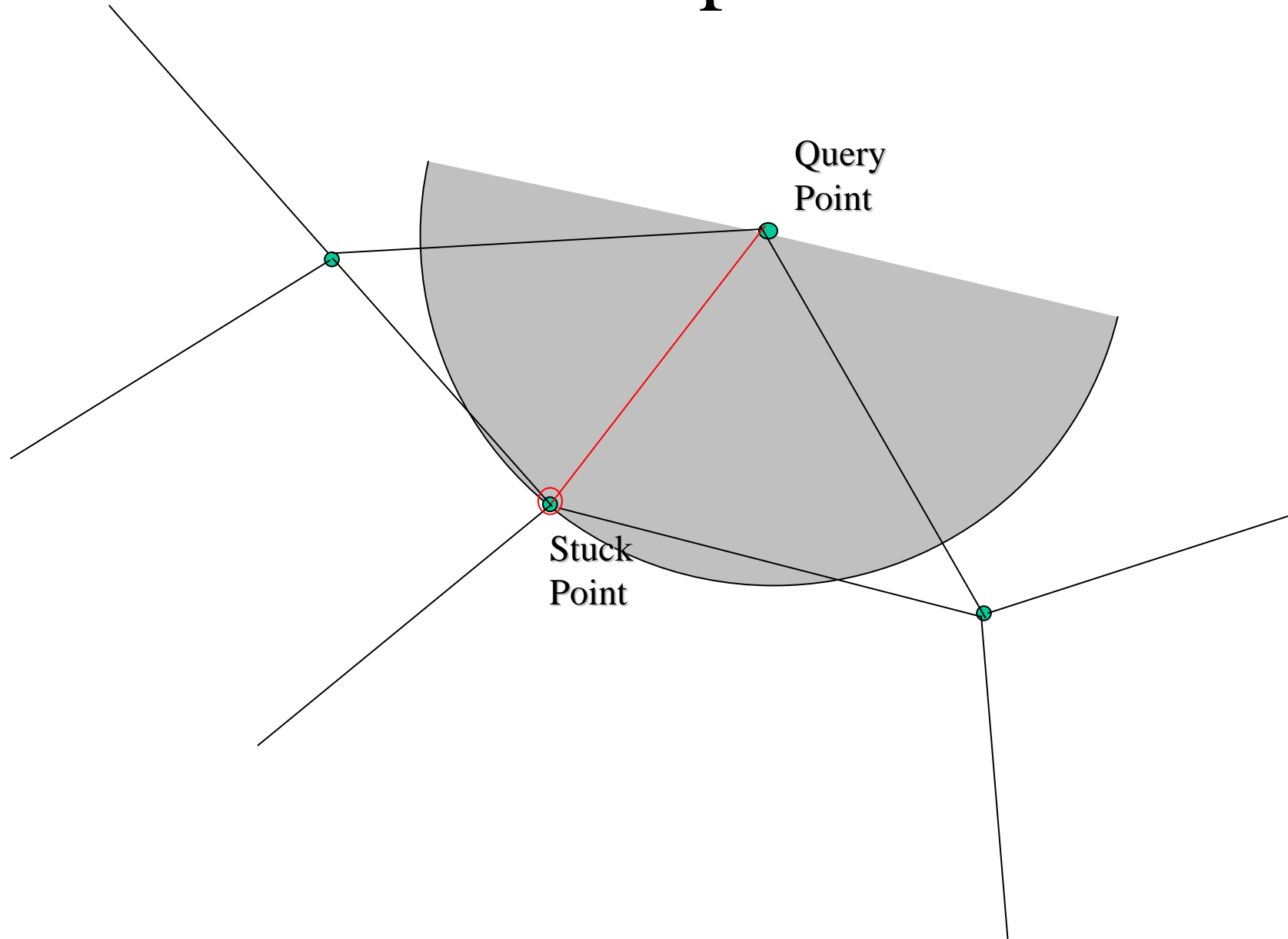
# Nearest First Traversing on Delaunay Graph



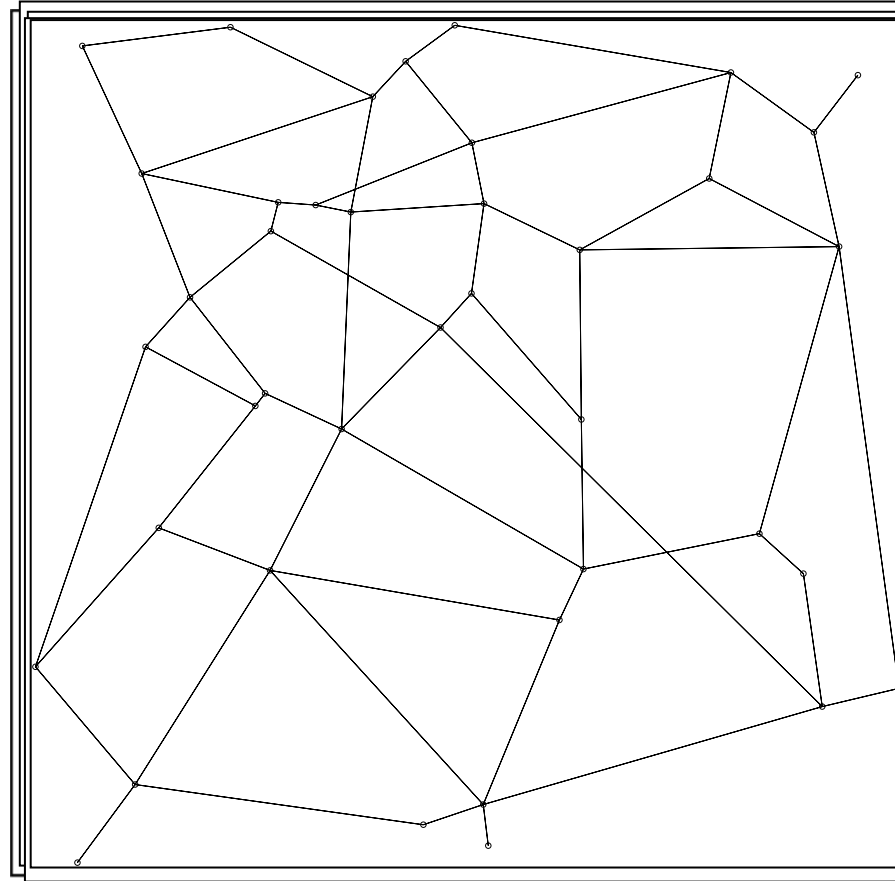
# Drawbacks

- High degree: Too many edges are connected to a single vertex ( $\geq D+1$ )
- Delaunay graph cannot be constructed in high dimensional space ( $N > 1000, D > 5$ )

# Stuck point



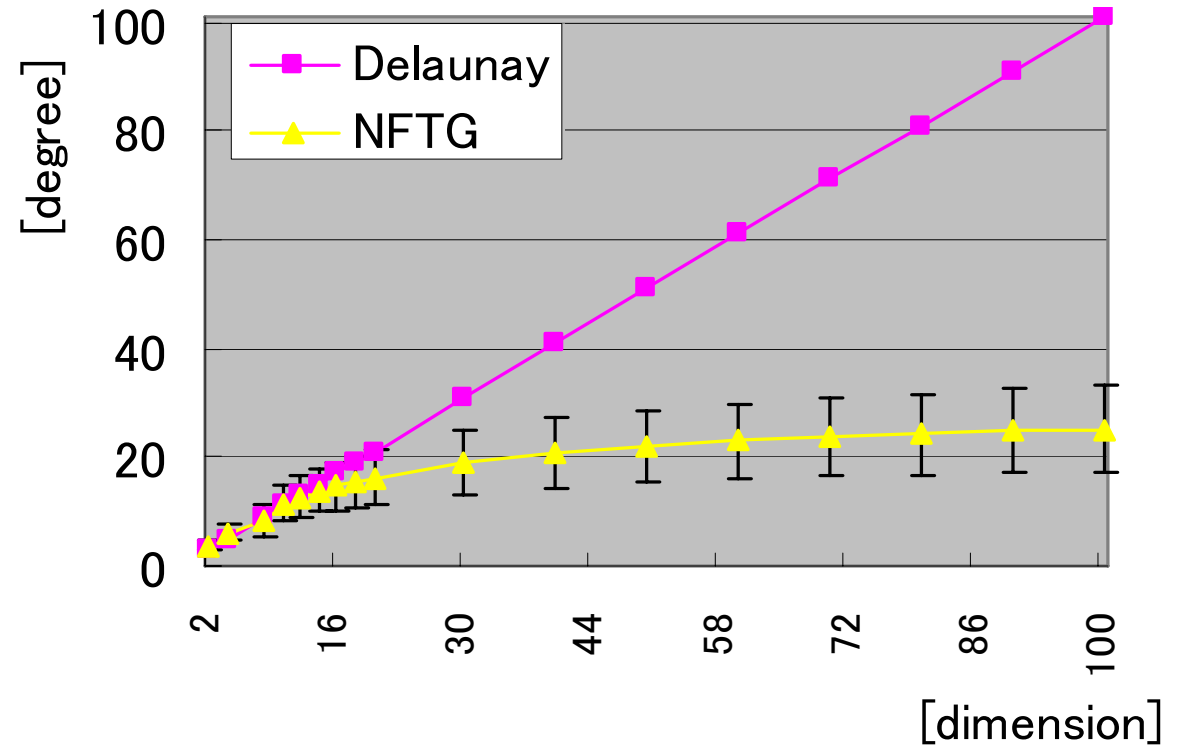
# Idea2: Connecting Stuck points



**NFTG: Nearest First Traversing Graph**

# Properties of NFTG

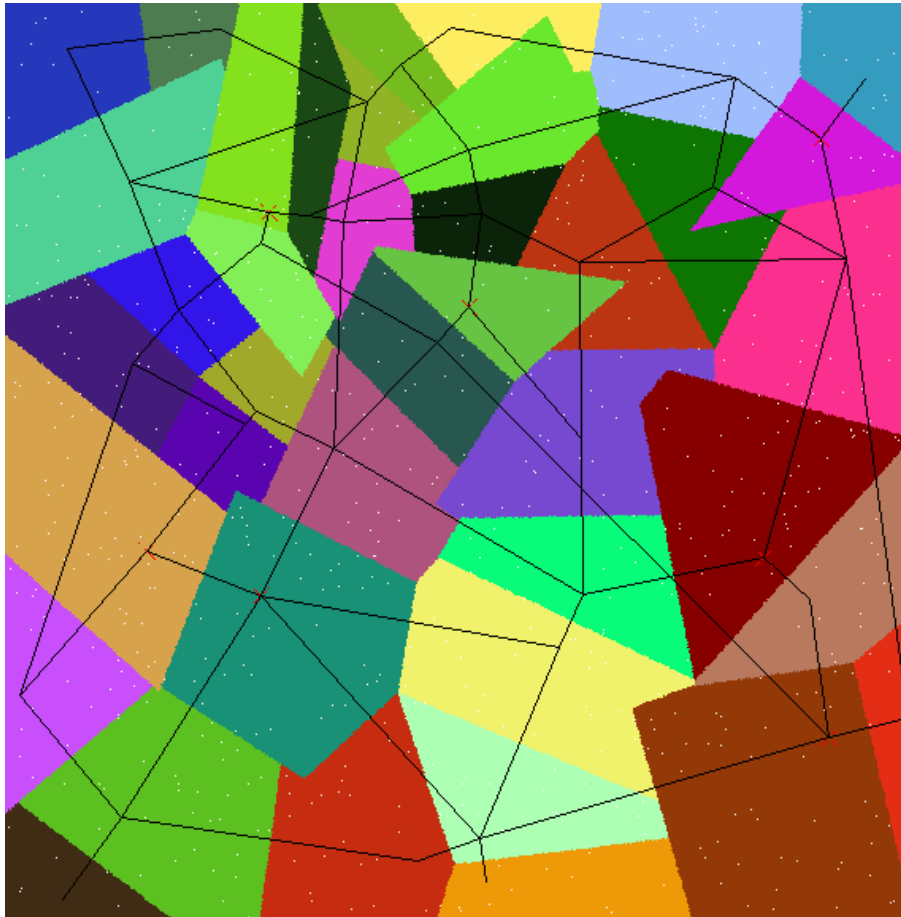
Small degree



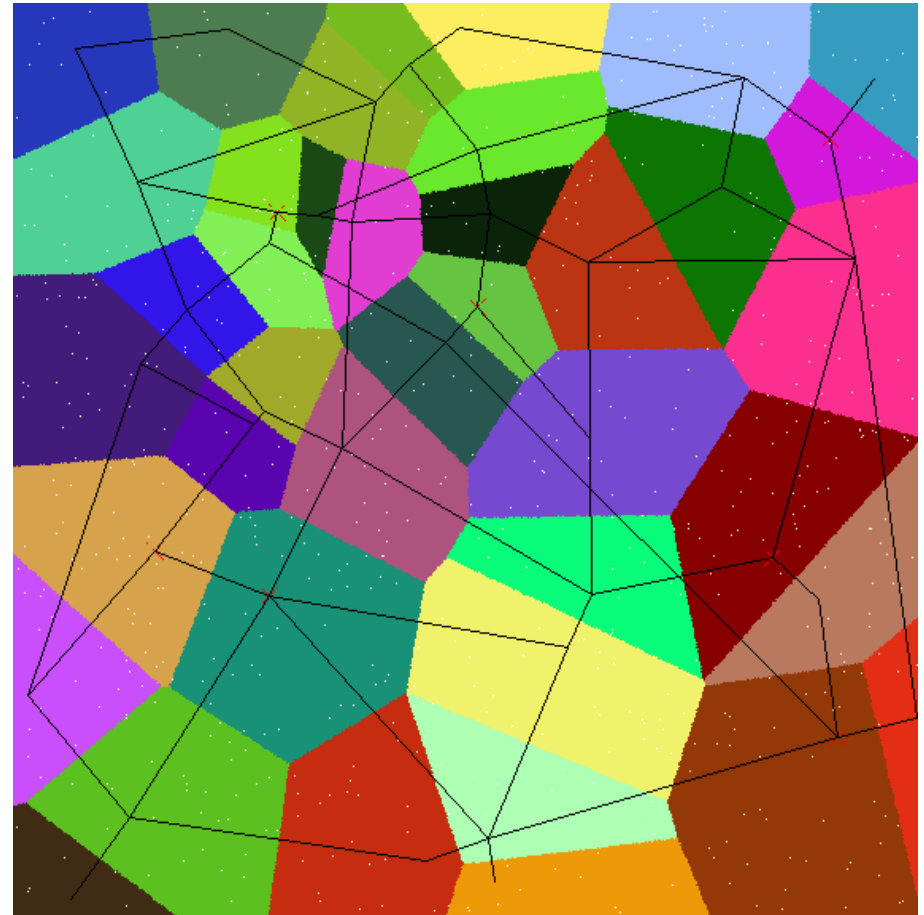
Easy to construct

# Properties of NFTG

NstFT using NFTG



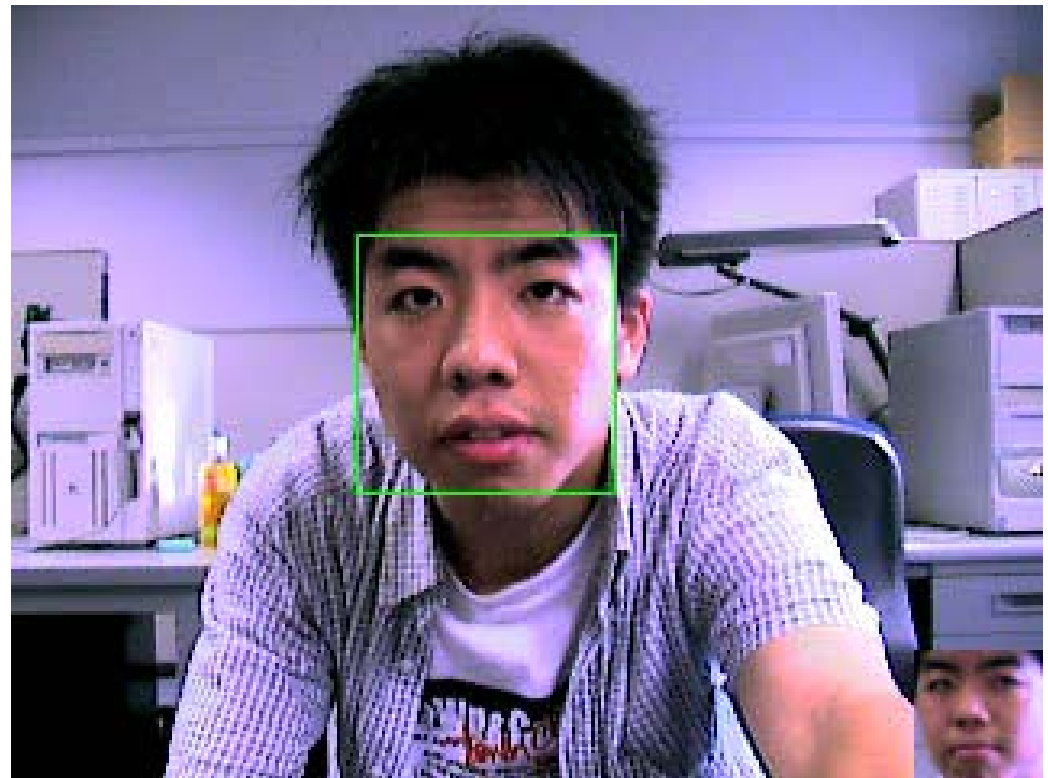
Nearest Neighbor





# Simultaneous Tracking and Recognition: Result

- Image Size 640x480
- Prototypes 10290
- Dimensions 4800
  
- Number of Distance Computation 20.8
- Elapsed time 4.16[ms]



# Other Results

(5 person, 5 direction, Translation  $7 \times 7$ ,  
Scale 3,  $5 \times 5 \times 7 \times 7 \times 3 = 3675$  prototypes)



# Results(5 person)

Elapsed Time  
/frame [ms]

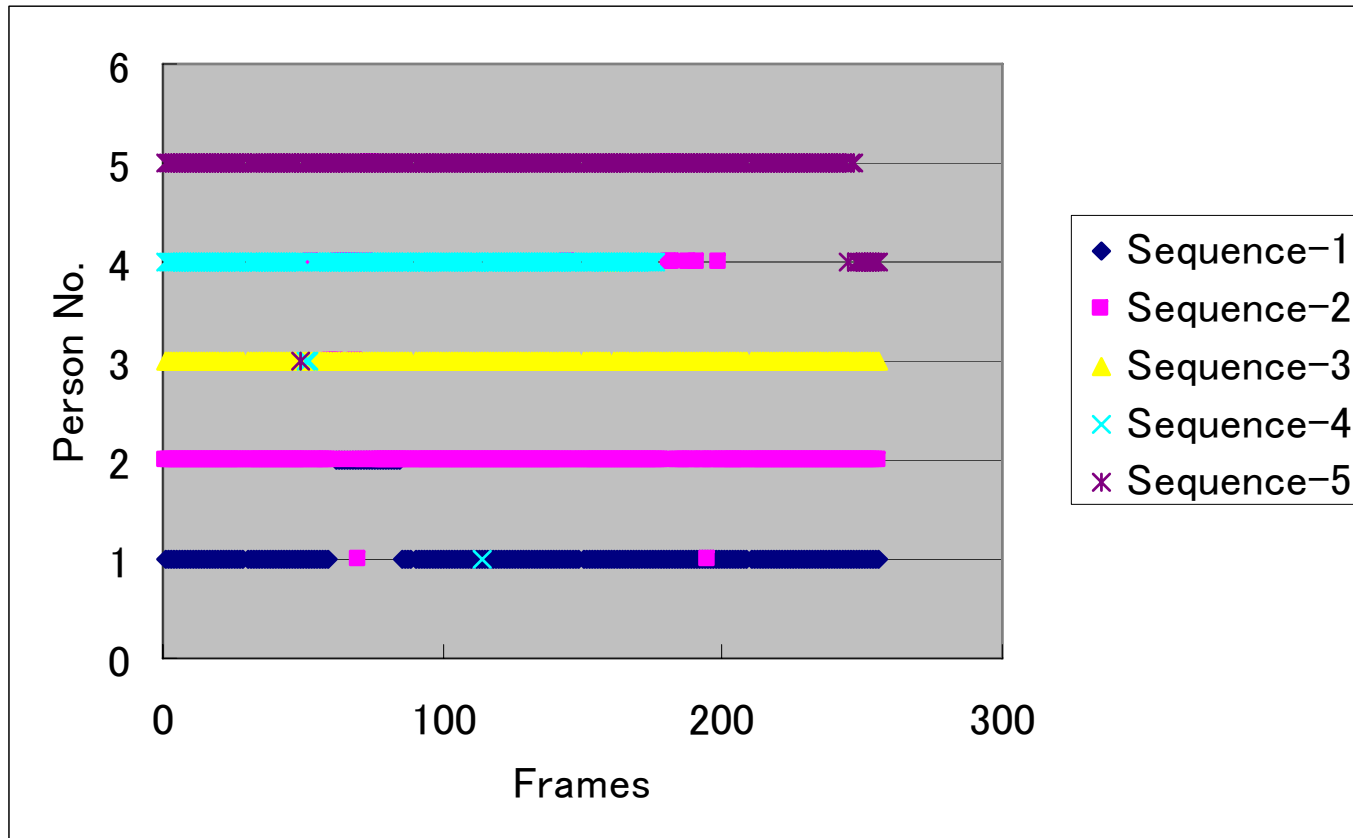
11.7

11.8

10.9

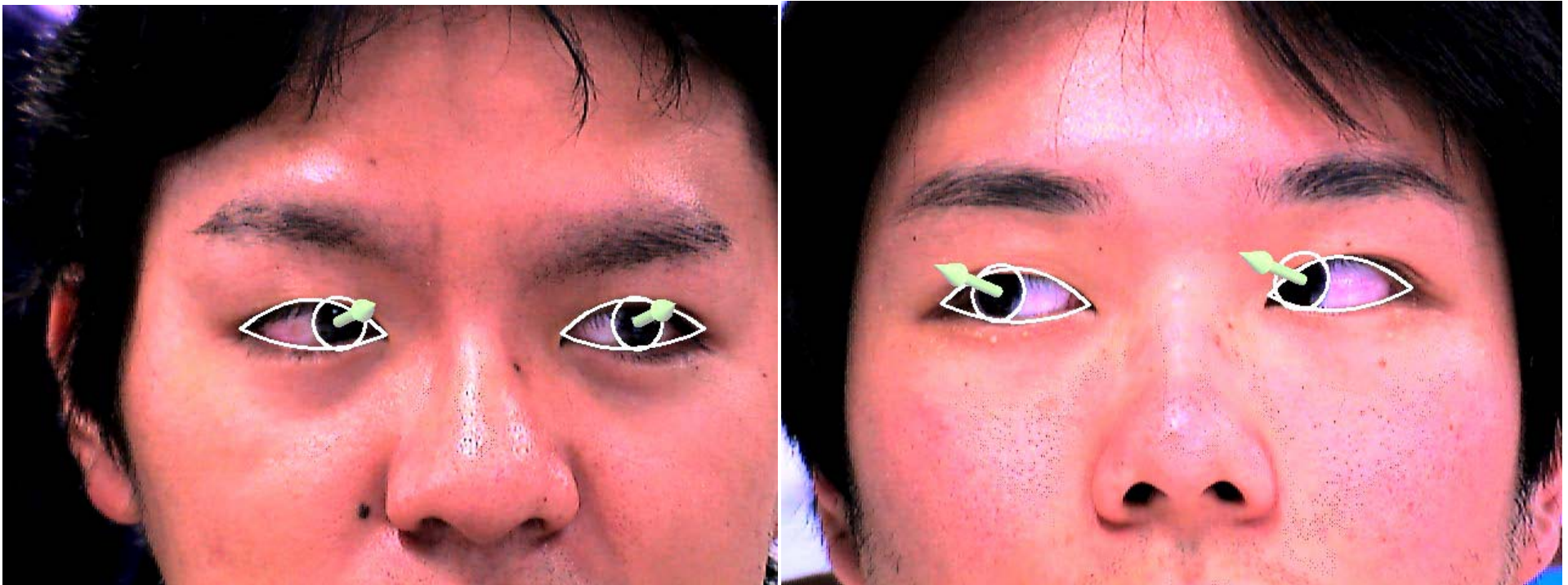
11.8

6.1

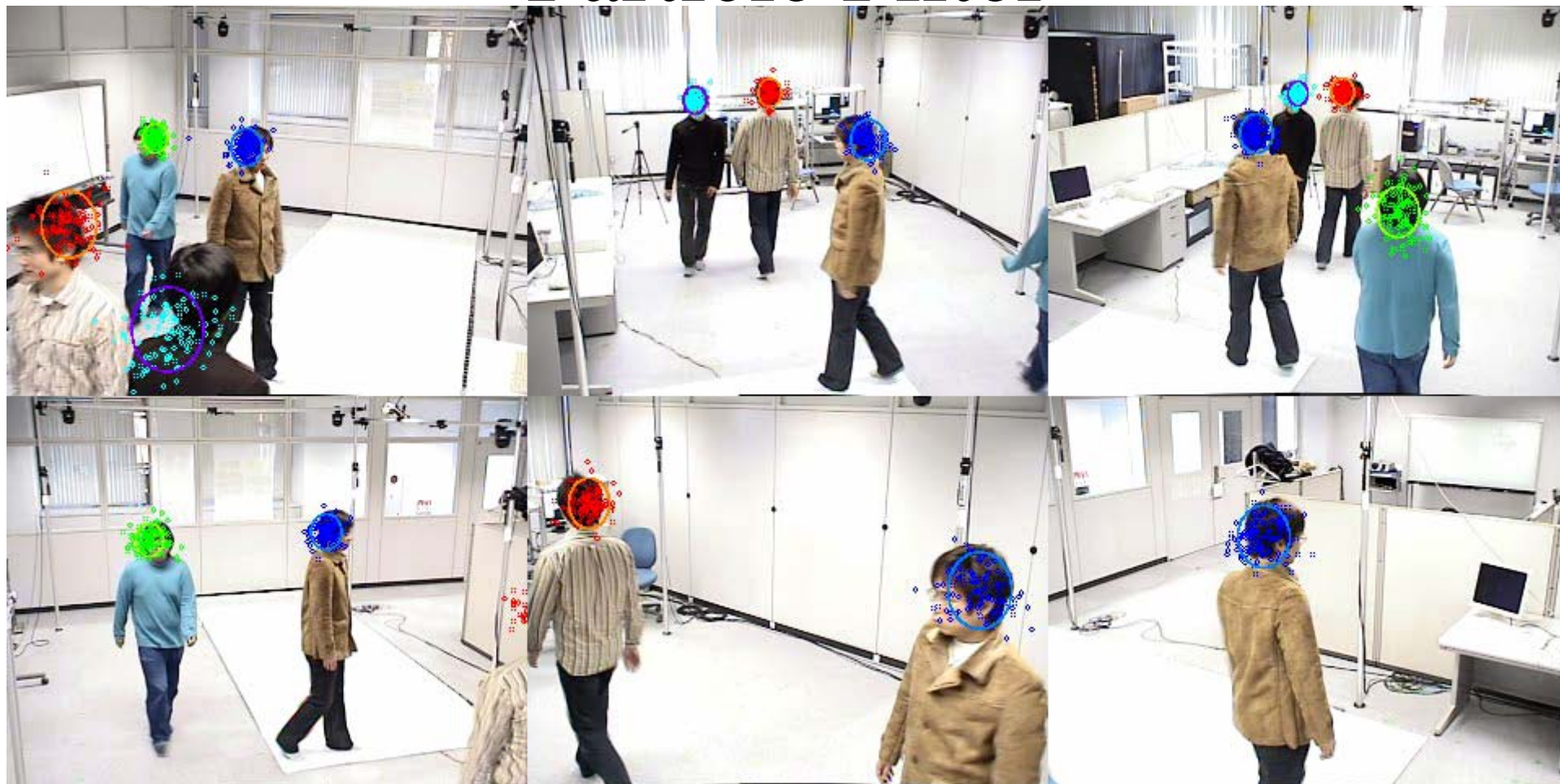


# 視線方向の検出: Particle Filter

- 目のモデルを用いた視線方向の推定



# マルチカメラトラッキング： Particle Filter



# まとめ

- 画像の問題も，認識の問題も，どちらも探索の問題です．
- 画像の中から対象を探索する問題は連続空間内でのサーチになるケースが多い
- 認識の問題は，カテゴリやクラスなどの離散位相を持った空間への写像になるケースが多い
- 現在のところ，これらを包含する枠組みはない．
- 良いアイデアがある人は，是非挑戦していただきたい． [ICCV2009](#)へ良い論文の投稿を！

ご清聴ありがとうございました。