画像の学習と応用

白井 良明

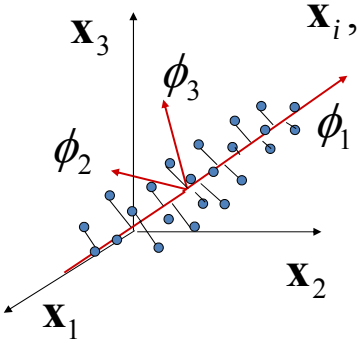
shirai@ci.ritsumei.ac.jp

www.i.ci.ritsumei.ac.jp/~shirai/

主成分分析

(Principal Component Analysis)

$\mathbf{x}_i, i = 1,2,...,n$ ：p次元ベクトル



$$\mathbf{z}_i = \phi^T \mathbf{x}_i$$

$$s^2 = \sum_{i=1}^{n} | \phi^T \mathbf{x}_i |^2 / n$$

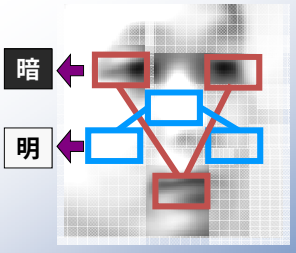
$$s^2 = \sum_{i=1}^{n} | (\mathbf{x}_i^T \phi) |^2 / n = \sum_{i=1}^{n} (\phi^T \mathbf{x}_i \mathbf{x}_i^T \phi) / n$$

$$= \phi^T (\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T / n) \phi$$

顔の状態の変化への対応

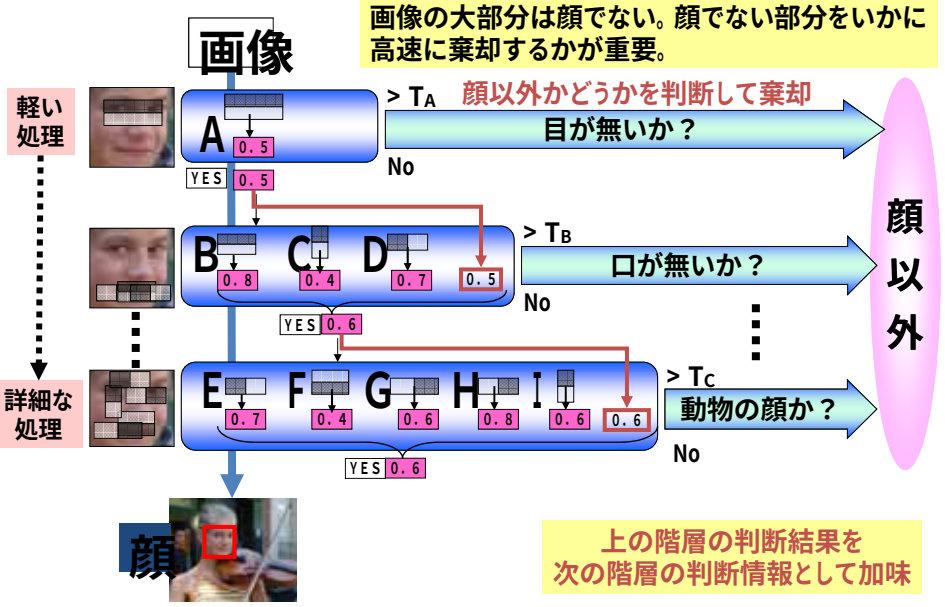
部分的な明暗差で顔をとらえることができる

斜め向きでも



階層化棄却技術

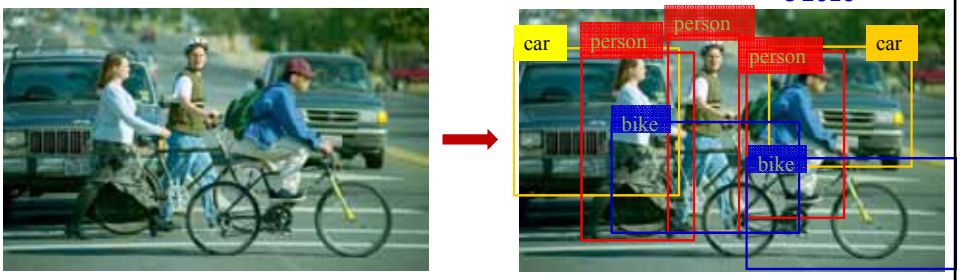
画像の大部分は顔でない。顔でない部分をいかに高速に棄却するかが重要。



上の階層の判断結果を
次の階層の判断情報として加味

Object Extraction from Image Database

PASCALVO C 2010



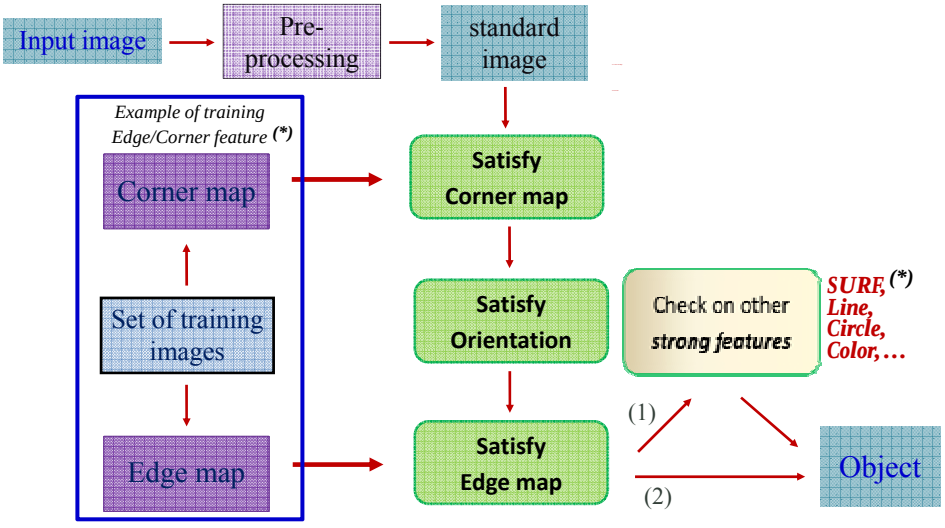
Semantic
not easy to get

Which objects
that an image contain?

Objectives:

Combine **multiple features** to detect
as many objects as possible

Object detection approach



(*)Other features are also trained

6

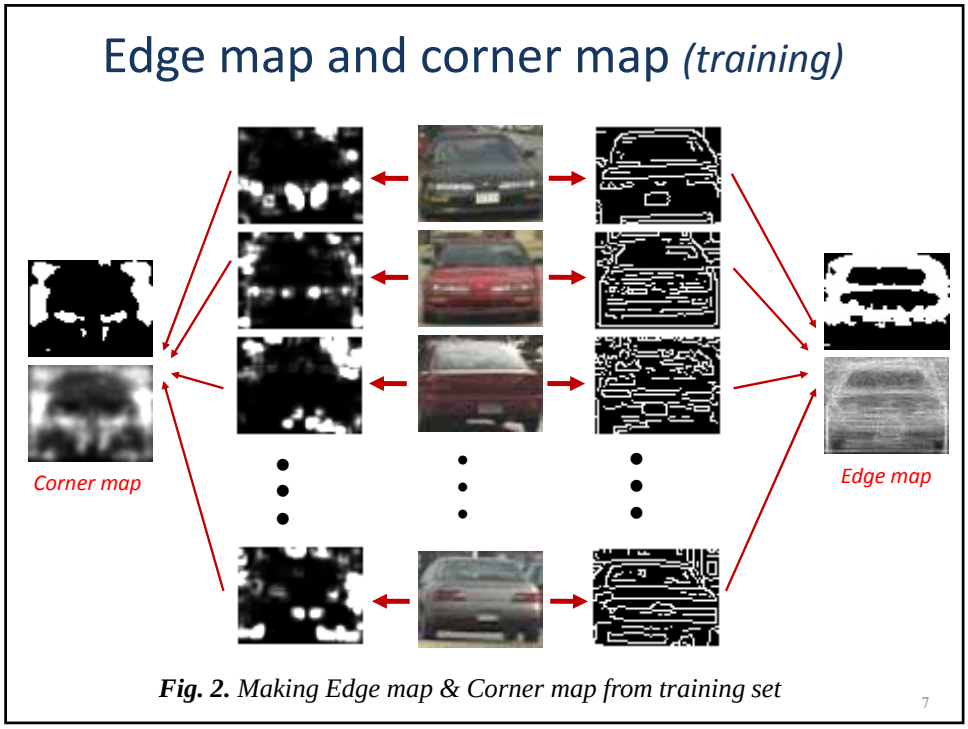
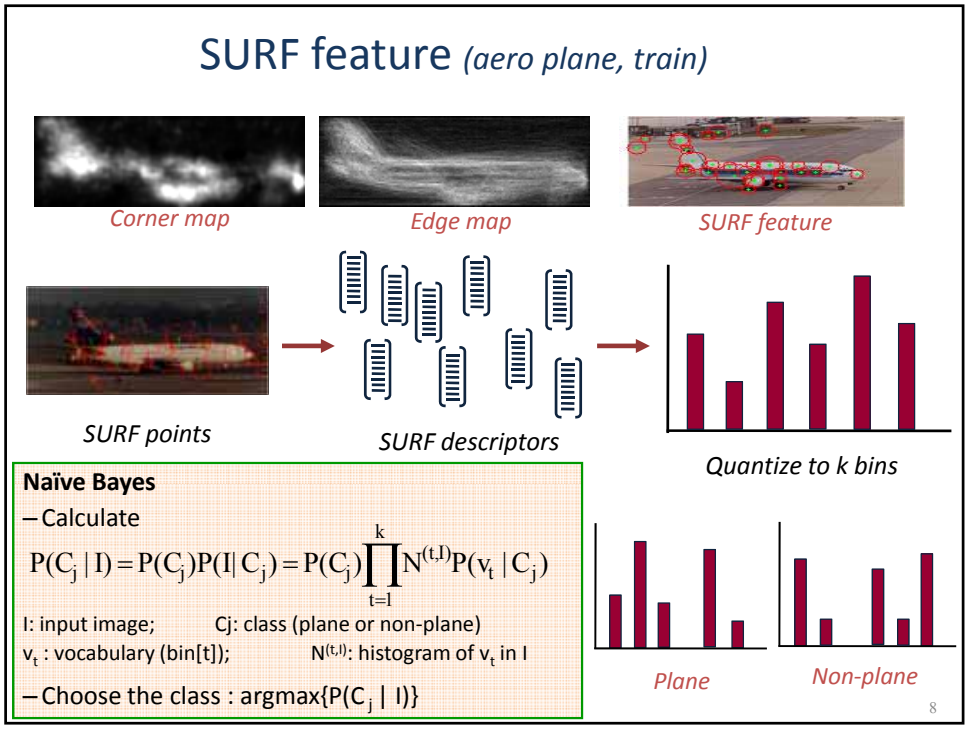
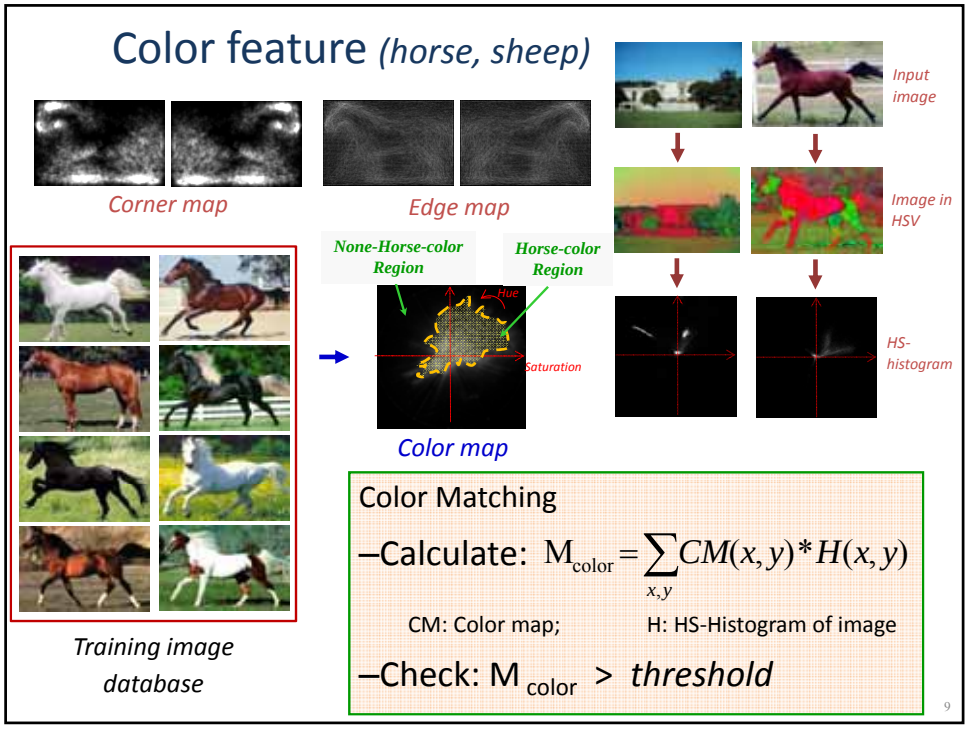


Fig. 2. *Making Edge map & Corner map from training set*





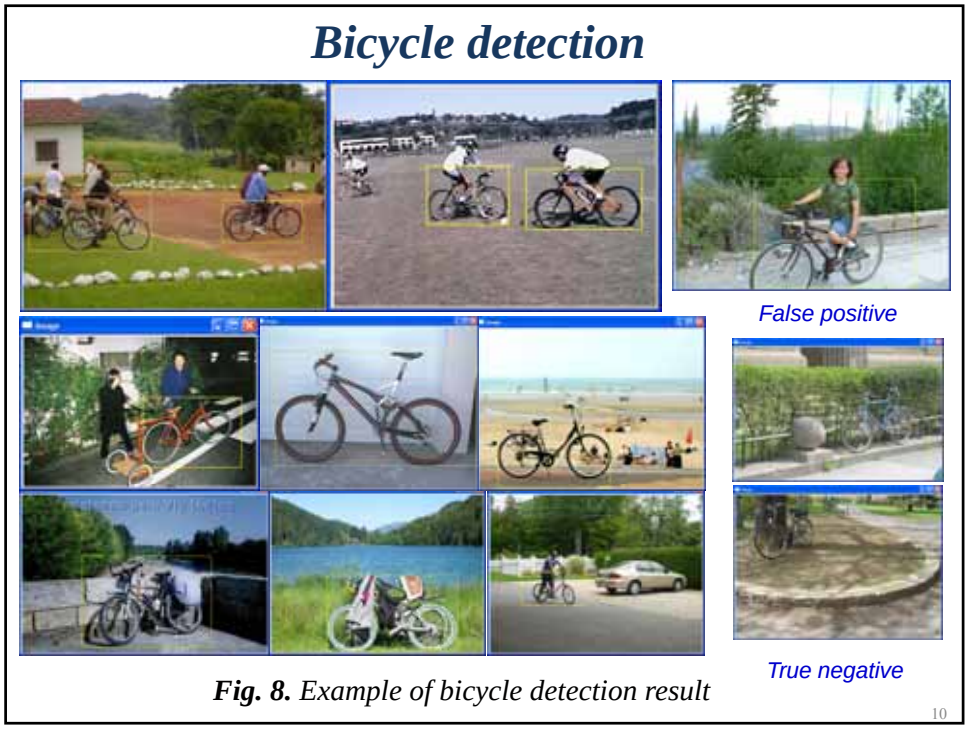
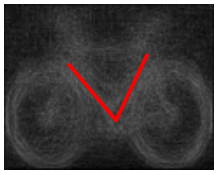
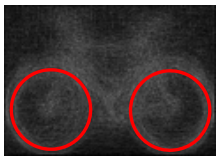


Fig. 8. *Example of bicycle detection result*

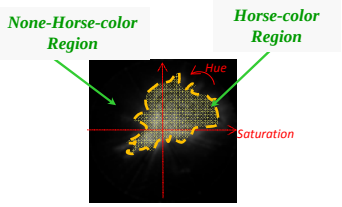
Automatically choose features

Two strongest lines of train



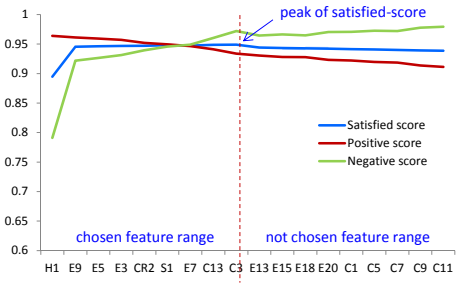


Line and Circle of bicycle

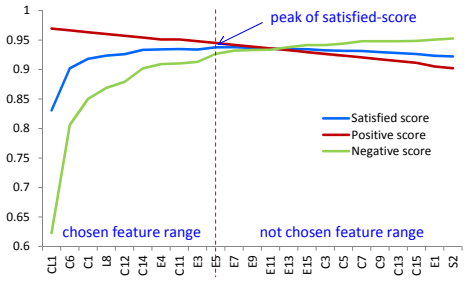


Color map of horse

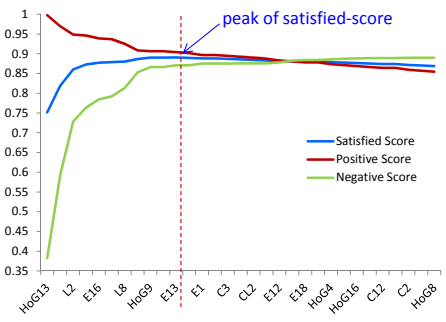
- Which features are *"good"* for a specific object?
- Can system *automatically* choose *"good"* features or not?



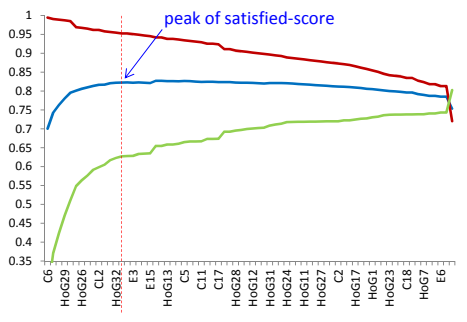
Horse object



Side view car object



Tower object



Flower object

Object with automatically chosen features

Object	Edge	Corner	Line	Circle	HoG	SURF	Color
F/r car	1	1	0	0	1	1	0
Side car	1	1	0	1	1	1	0
Bike	1	1	1	1	0	0	0
Train	1	1	0	0	1	1	0
Aero plane	1	1	1	0	0	1	0
Motorbike	1	1	0	0	0	1	1
Horse	1	1	1	0	0	1	1
Sheep	1	1	0	0	0	1	1
Tower	1	1	1	0	1	0	0
Flower	1	1	0	0	1	1	0

Table 2. Features are good for recognizing object

13

Object	TRAINING STAGE		TESTING STAGE	
	Average Precision	Average Recall	Average Precision	Average Recall
F/r car	96.48%	90.21%	95.12%	90.14%
Side car	97.20%	94.92%	94.21%	91.73%
Bike	85.80%	81.32%	84.02%	79.14%
Train	87.24%	77.16%	83.65%	75.31%
Aero plane	86.65%	84.55%	85.75%	84.51%
Motorbike	90.23%	87.32%	89.38%	85.47%
Horse	88.93%	82.09%	87.81%	75.40%
Sheep	87.32%	75.91%	86.25%	73.24%
Tower	89.07%	90.39%	84.33%	82.64%
Flower	82.71%	75.15%	81.57%	74.42%

Table 3. AP & AR at training/testing stage of automatically choosing features

Fig. 9. *Example of side view car (top) and plane (bottom) detection result*

Evaluation

Object	Our system	PASCAL VOC 2010 [1]	
	Average Precision	Average Precision	Authors
Front/rear car	95.12%	49.10%	UOCTTI_LSVM_MDPM
Side car	94.21%		
Bicycle	84.02%	55.30%	NLPR_HOGLBP_MC_LCEGCHLC
Train	83.65%	50.30%	MITUCLA_HIERARCHY
Aero plane	85.75%	58.40%	UVA_GROUPLOC
Motorbike	89.38%	56.30%	NUS_HOGLBP_CTX_CLS_RESCORE_V2
Horse	87.81%	51.90%	NUS_HOGLBP_CTX_CLS_RESCORE_V2
Sheep	86.25%	37.80%	UVA_DETMONKEY
Tower	84.33%	84.00%	HENA_LU_DU [2]
Flower	81.57%	80.00%	JZU_HONG_CHEN_LI [3]

Table 5. Comparison between our system with PASCAL 10

[1] PASCAL 10 http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/results/ (about 20 categories)
[2] Lu Yang, Du Xiao-wei, 2nd Intl Asia Conference on Informatics in Control, Automation and Robotics 2010, p.p 349-352
[3] Hong et al. / J Zhejiang Univ SCI 2004 5(7):764-772 http://www.zju.edu.cn/jzus

16

Conclusion

- Combine various features for object detection.
- Automatically choose suitable features.

Future works

- $f_i \wedge f_{i+1} \wedge f_{i+2} \rightarrow f_i \wedge (f_{i+1} \vee f_{i+2})$

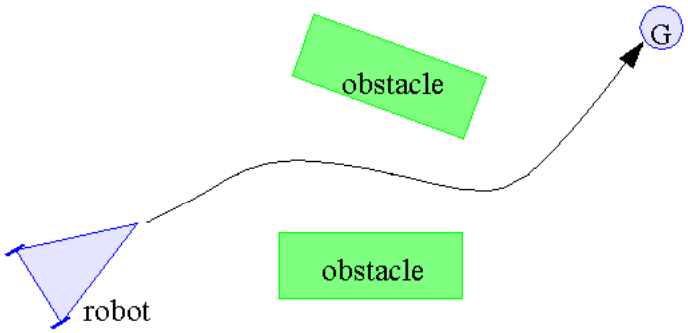
- very small object

Select *"suitable"* scale

Dynamic Motion Planning for Efficient Visual Navigation under Uncertainty

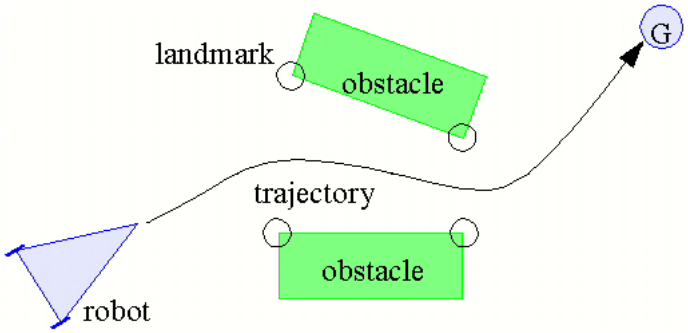
Objective

- Safely and quickly reaching a goal position



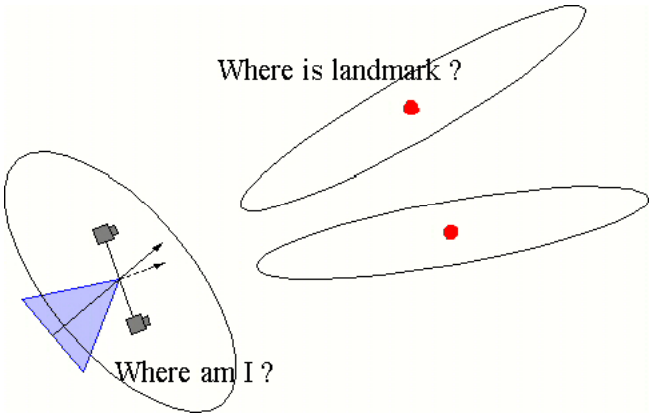
Basic assumption for the first case

- Indoor environment is known
- Landmarks are given
- Use stereo vision for localization



Problem

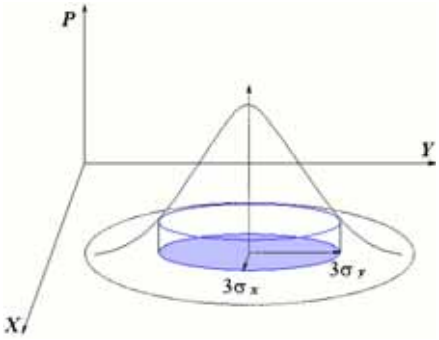
- Motion and vision include uncertainty.
- Vision requires high computational cost.



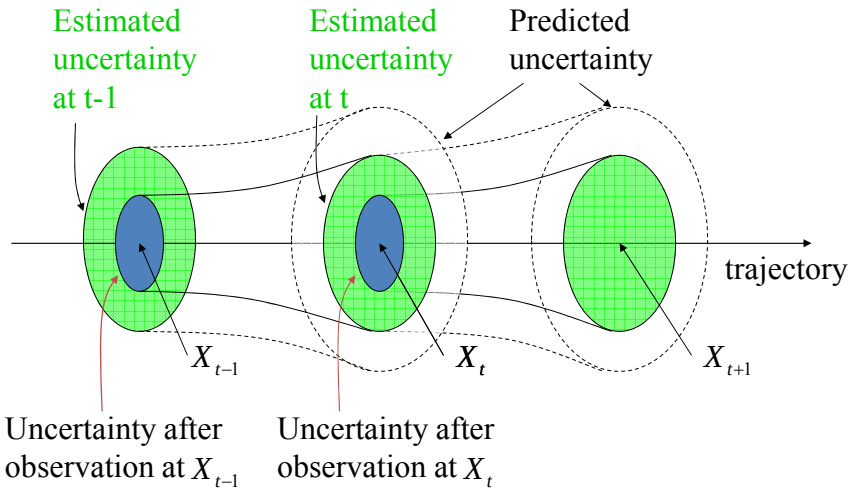
Uncertainty

- Motion uncertainty
  - control error
  - rolling, slippage and etc.
- Vision uncertainty
  - quantization error
  - calibration error

3σ area of covariance matrix

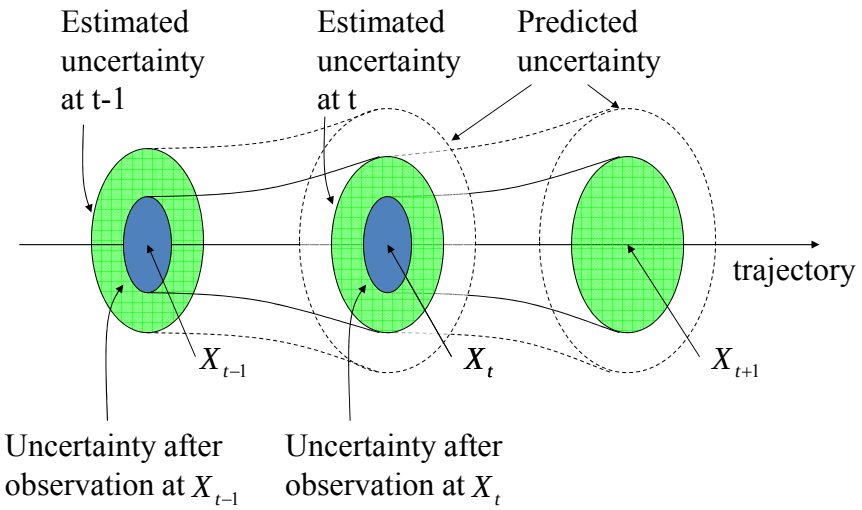


Non-stop and speed controlled navigation strategy



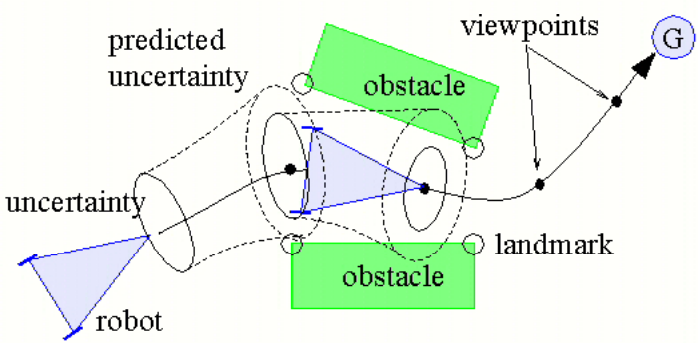
Speed controlled motion between planned viewpoints

Non-stop navigation strategy

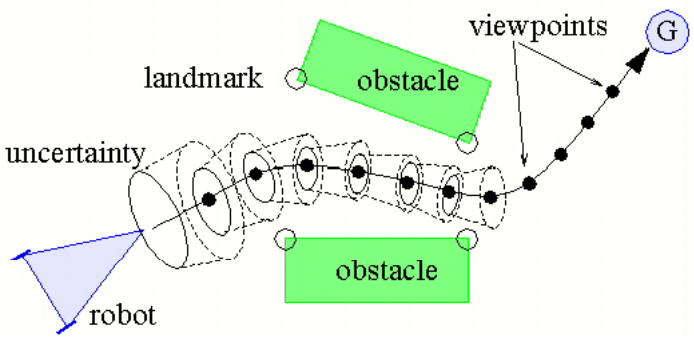


Speed controlled motion between planned viewpoints

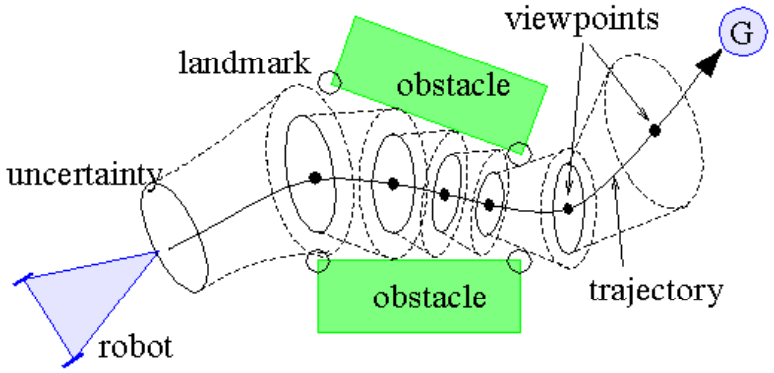
Case 1: fast but dangerous



Case 2: safe but slow



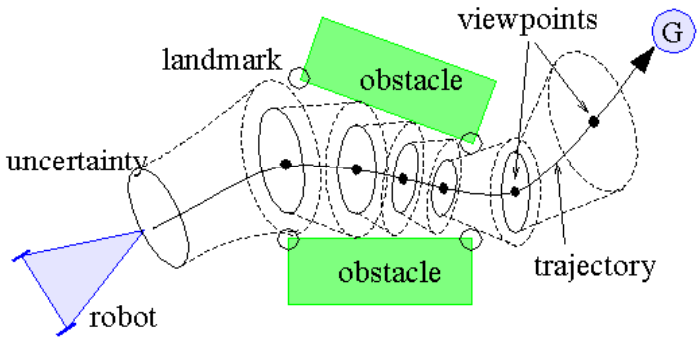
Case 3: adaptive observation



Adaptive viewpoint planning method considering uncertainty

Adaptive viewpoint

1. Observation position to guarantee the safety
2. Observation position to reach the goal position quickly



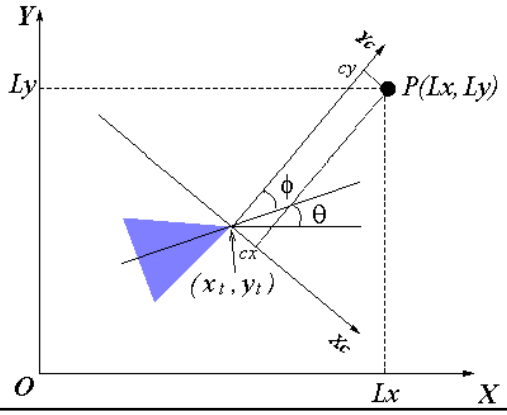
Idea

- Safety
  - considering uncertainty
- Quickly
  - navigation with minimum observation
  - non-stop and speed controlled navigation

Adaptive viewpoint planning under uncertainty

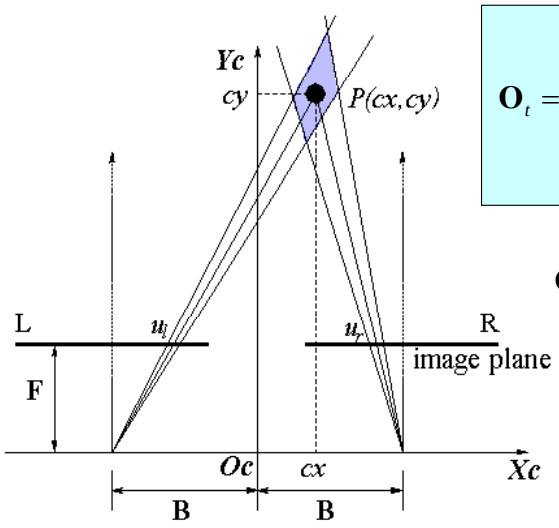
Observation model

$$\mathbf{G}(\mathbf{X}_t, \mathbf{O}_t) = \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \begin{bmatrix} \sin(\theta_t + \phi_t) & \cos(\theta_t + \phi_t) \\ -\cos(\theta_t + \phi_t) & \sin(\theta_t + \phi_t) \end{bmatrix} \begin{bmatrix} cx_t \\ cy_t \end{bmatrix} - \begin{bmatrix} L_x \\ L_y \end{bmatrix} = \mathbf{0}$$



$$\mathbf{X}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} \qquad \mathbf{O}_t = \begin{bmatrix} cx_t \\ cy_t \end{bmatrix}$$

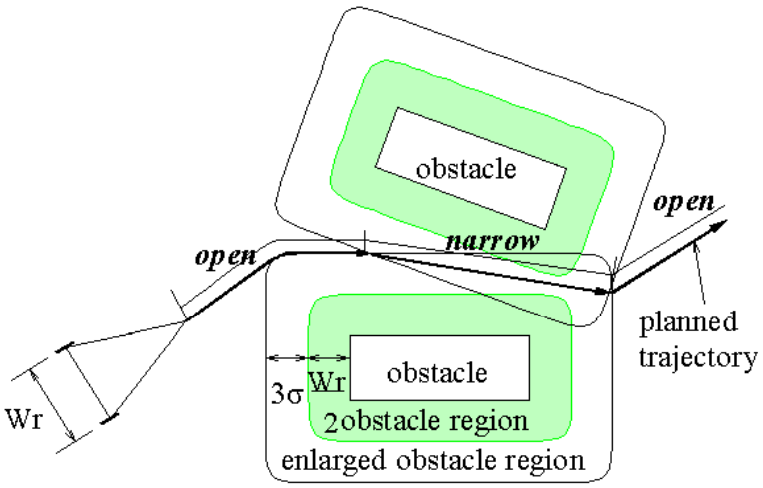
Stereo observation model



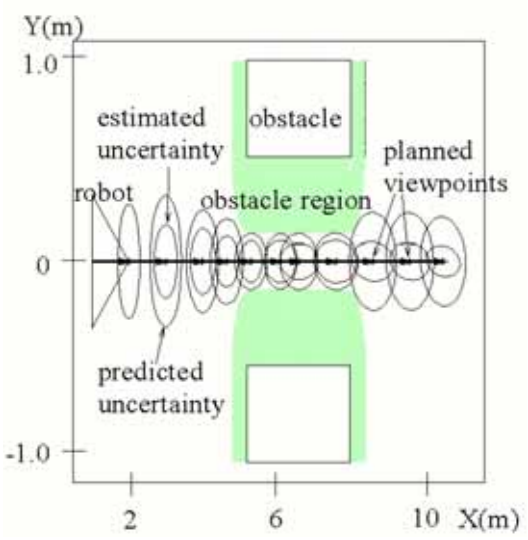
$$\mathbf{O}_t = \mathbf{Z}(\mathbf{I}_t) = \begin{bmatrix} \frac{\mathrm{B}(u_{l_t} + u_{r_t})}{u_{l_t} - u_{r_t}} \\ \frac{2\mathrm{BF}}{u_{l_t} - u_{r_t}} \end{bmatrix}$$

$$\mathbf{O}_t = \begin{bmatrix} cx_t \\ cy_t \end{bmatrix} \quad \mathbf{I}_t = \begin{bmatrix} u_{l_t} \\ u_{r_t} \end{bmatrix}$$

Trajectory planning

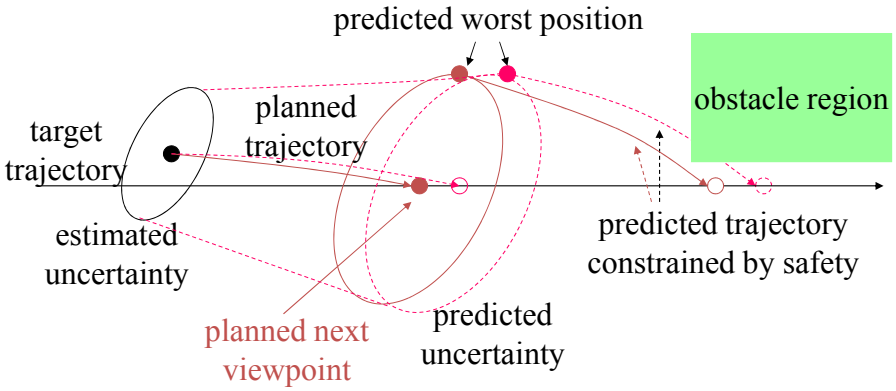


Offline planned viewpoints



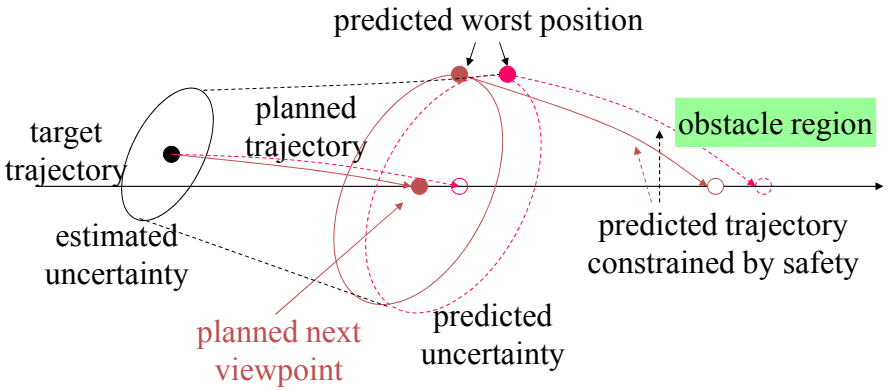
Dynamic planning method

Move to the farthest safe position on the target trajectory

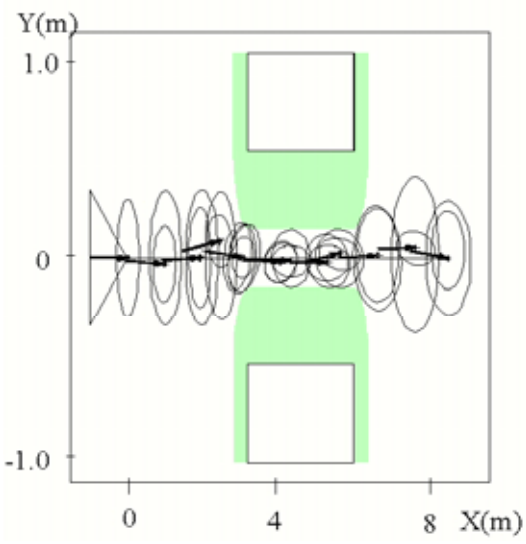


Dynamic viewpoint planning

Move to a position to guarantee safety



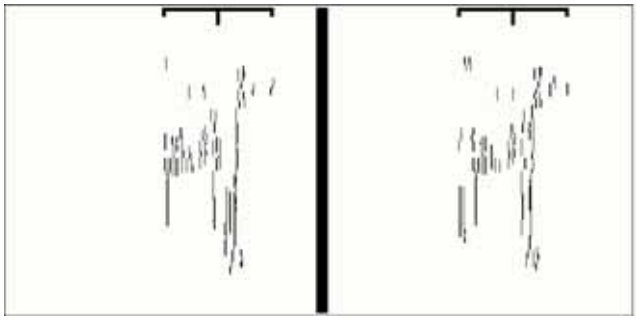
Motion result using online planning



Landmark observation

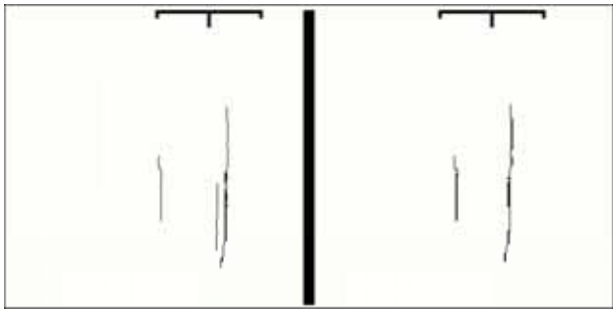
- Vertical segment as landmark
- Selection candidates by position and similarity constraints

Position constraint



Left image Right image

Similarity constraint



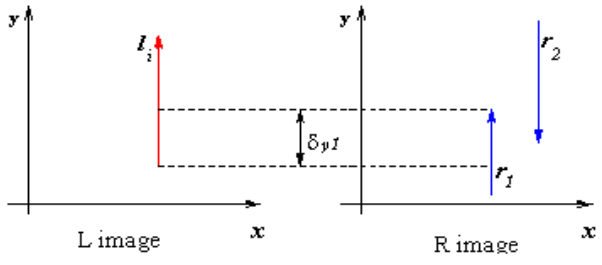
Left image Right image

Similarity of stereo segment pair

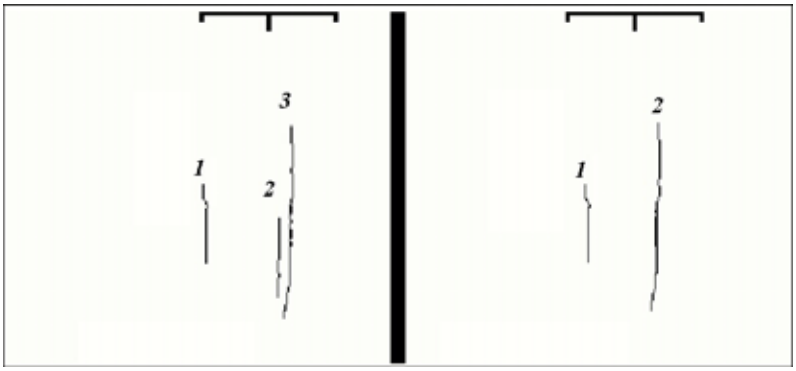
1. Orientation similarity $OS(l_i, r_j) = \begin{cases} 1 & (if\ dir(l_i) = dir(r_j)) \\ 0 & (else) \end{cases}$

2. Overlap length $OL(l_i, r_j) = \frac{1}{2}[\frac{\delta y}{len(l_i)} + \frac{\delta y}{len(r_j)}]$

3. Length similarity $LS(l_i, r_j) = \frac{\min[len(l_i), len(r_j)]}{\max[len(l_i), len(r_j)]}$



Example of matching

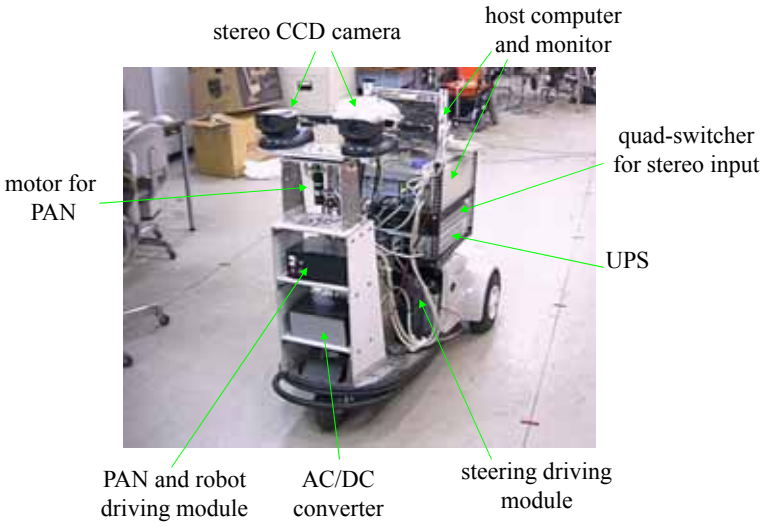


Result of landmark detection

Predicted landmark position on image

Detected landmark position on image

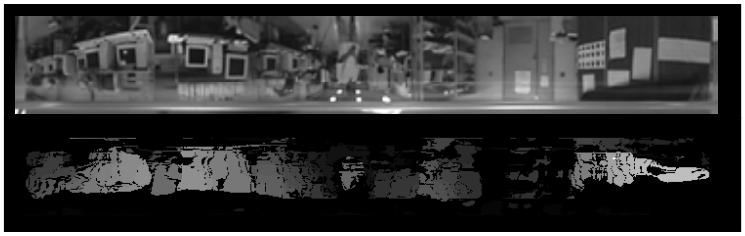
Mobile robot



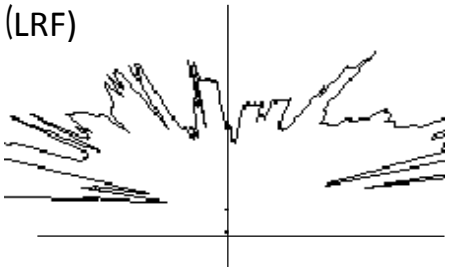
実験　(部分的に早送り)

Range sensors

– Omnidirectional Stereo

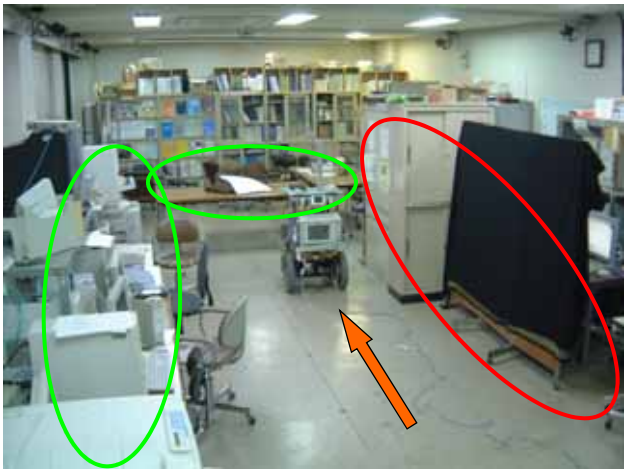


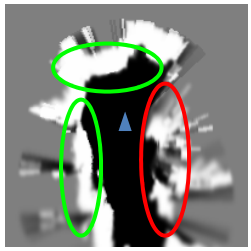
– Laser Range Finder (LRF)



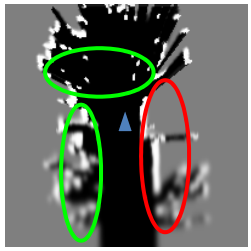
Generating probabilistic occupancy maps

- Representing probabilities of obstacle existence in each grid.





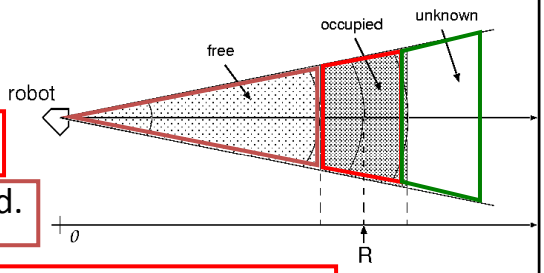
Omnidirectional stereo



LRF

Interpretation of current sensor data

- E: obstacle exists.
- O: obstacle is observed.
- $\overline{O}$: free space is observed.



$$P(E \mid O) = \frac{P(O \mid E)P(E)}{P(O \mid E)P(E) + P(O \mid \overline{E})P(\overline{E})}$$

$$P(E \mid \overline{O}) = \frac{P(\overline{O} \mid E)P(E)}{P(\overline{O} \mid E)P(E) + P(\overline{O} \mid \overline{E})P(\overline{E})}$$

$P(E)$: prior probability (initialized to 0.5)

$P(O \mid E)$, $P(O \mid \overline{E})$: observation models

Map generation
-integration of two sensors-

- Integration by a logical rule.

		Omnidirectional stereo			
		obstacle	undecided*wo*	undecided*wt*	free
L R F	obstacle	obstacle	obstacle	obstacle	obstacle
	undecided*wo*	obstacle	obstacle	obstacle	obstacle
	undecided*wt*	obstacle	obstacle	obstacle	free
	free	obstacle	obstacle	free	free

undecided*wo* : not observed yet

Formulation

The robot moves at a speed to sufficiently observe an undecided region.

N : the number of observations recognizing a free space

d : Distance to undecided region

T : Observation cycle (constant)

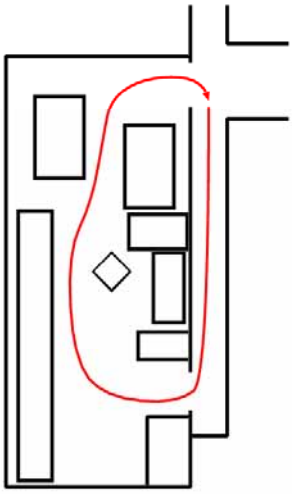
v : Robot speed

$$\frac{d}{vT} \geq N$$

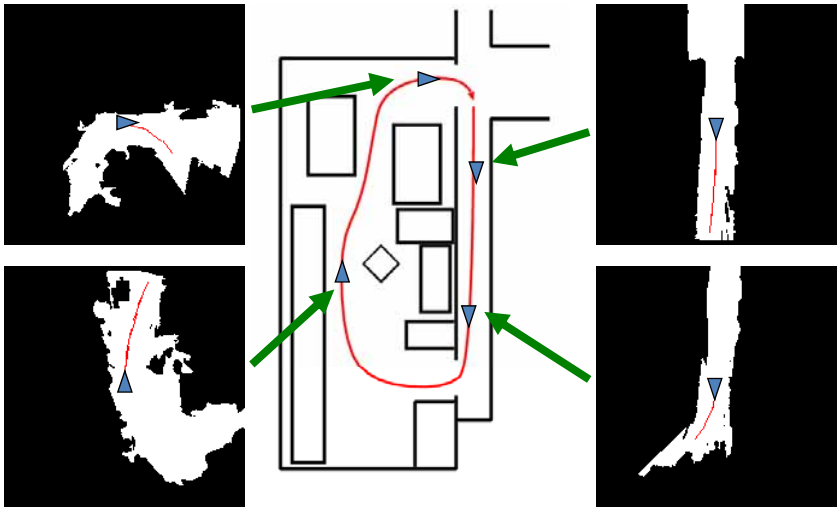
Maximum speed

$$v_{\max} = \frac{d}{NT}$$

Experiment (real-time movie)



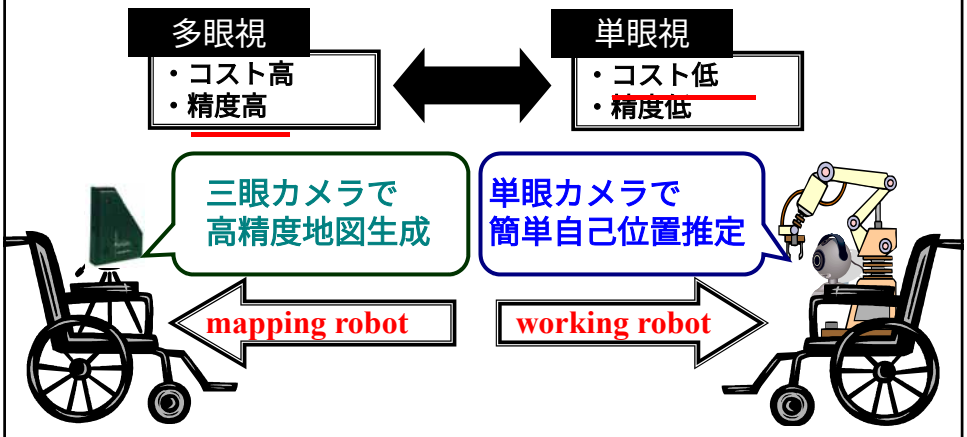
Free space maps and planned paths



Moving distance: 30[m] in 45[sec] (maximum speed: 1[m/s])
(without speed control, 150[sec])

環境地図生成とナビゲーション

三眼視による
ロボットのナビゲーション
＋
単眼視による
ロボットのナビゲーション



特徴点抽出

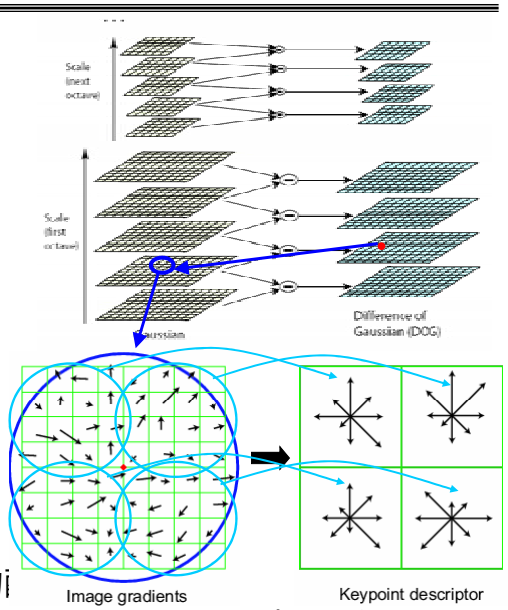
SIFT特徴

Scale
Invariant
Feature
Transform

特徴点のパラメータ

- 画像座標
- スケール
- コントラスト
- SIFT特徴ベクトル

特徴ベクトルは周囲の輝度勾



example : n=2

ステレオ対応付け

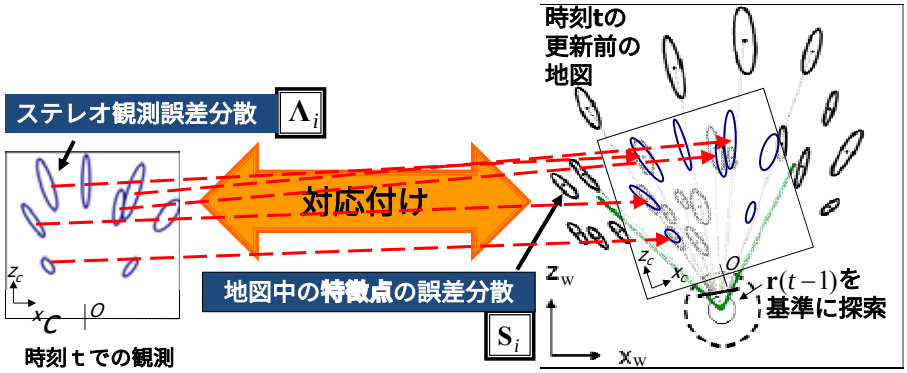
2枚の画像間でユニークな対応を見つける
(T-R間 と R-L間)

水平・垂直共に対応が付いた点
⇒ stable keypoint

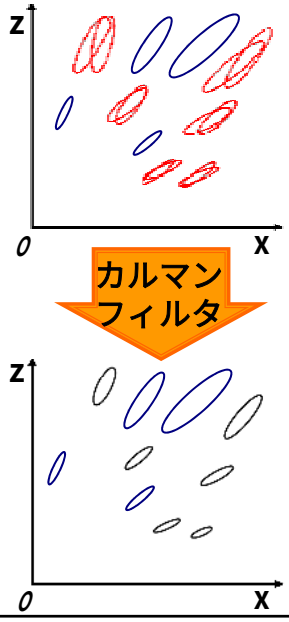
地図に登録

状態推定

時刻tにおける状態 $\mathbf{r}(t) = [X(t)\ Z(t)\ \theta(t)]^T$ を推定する



地図の更新



地図生成結果

地図生成過程

High

特徴点の高さ

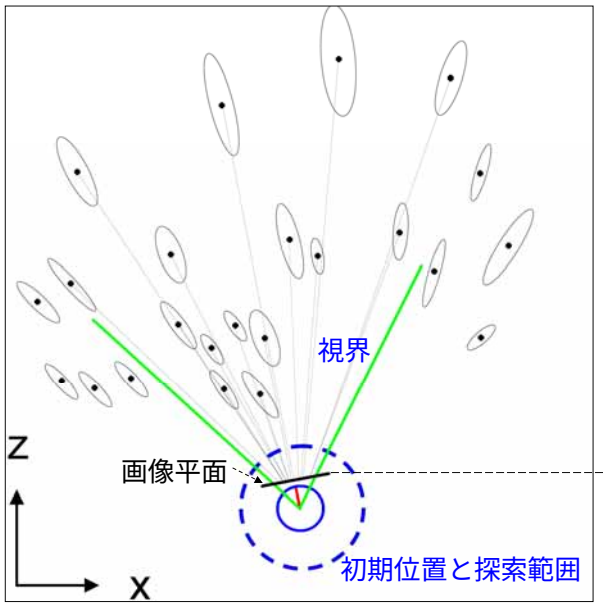
Low

入力画像系列

•：地図と対応した点

+：新しく登録された点

自己位置推定では、特徴点の位置を予測



入力画像

予測

再構成画像

マッチングと状態推定

入力画像

対応付け

再構成画像

・：地図中のkeypointとマッチした点

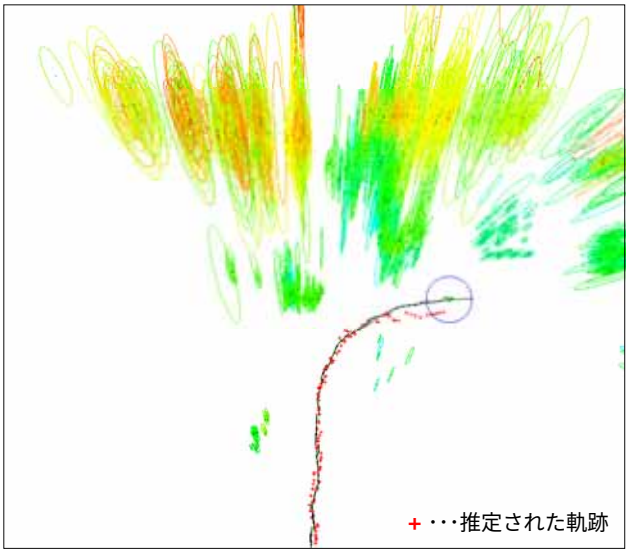
画像平面上での誤差の２乗和を最小化

状態推定結果

状態推定シミュレーション

mapping時と別の画像シーケンスを入力

状態推定に要した時間
0.75 [sec/frame]（平均）



自己位置推定結果

mapping robotの左カメラのシーケンスを入力

	x[mm]	z[mm]	θ[deg]
誤差平均	-0.20	2.07	0.03
分散	157.63	795.04	0.13
標準偏差	**12.55**	**28.20**	**0.36**

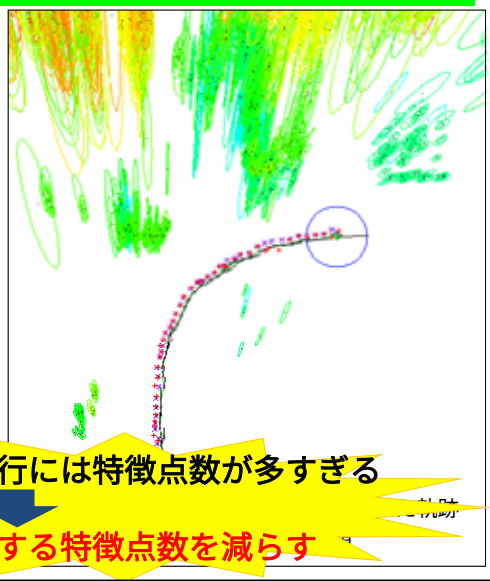
探索範囲

-100 mm≦ x ≦+100 mm

-100 mm≦ z ≦+100 mm

- 10 deg≦θ≦+ 10 deg

(量子化幅…x:10mm, z:10mm, θ:1deg)



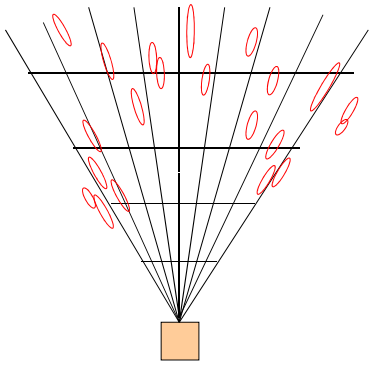
状態推定に要した時間
0.86 [sec/frame]（実時間実行には特徴点数が多すぎる

投影する特徴点数を減らす

投影する特徴点削減

視野空間をセグメンテーション
＆それぞれの区画で代表点を選ぶ

近い点 / 遠い点 は，ロボットの 位置 / 向き を求めるのに有利

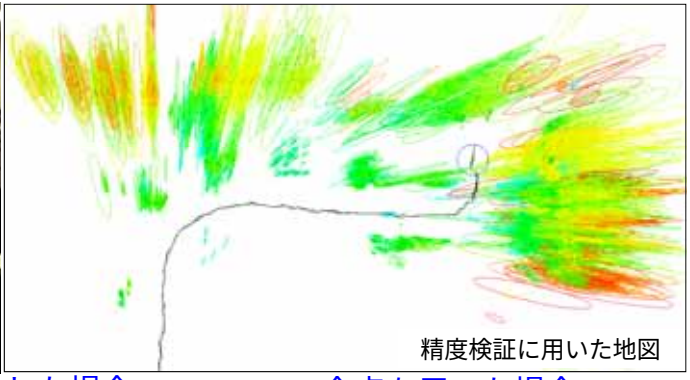


代表点の基準

① 観測回数が多い
② コントラストが高い

画像平面に投影し，
入力画像とマッチング

投影点削減効果



精度検証に用いた地図

投影点を減らした場合

	x[mm]	z[mm]	φ[deg]
誤差平均	-1.90	-1.38	0.16
分散	863.41	713.43	1.38
標準偏差	**29.38**	**26.71**	**1.18**

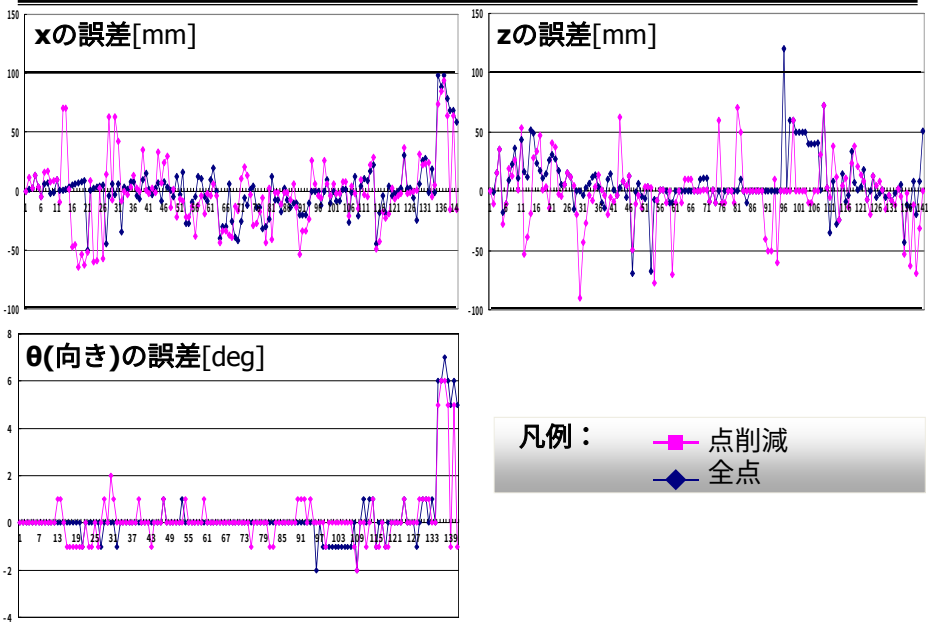
●処理時間　**0.37** [sec/frame]

全点を用いた場合

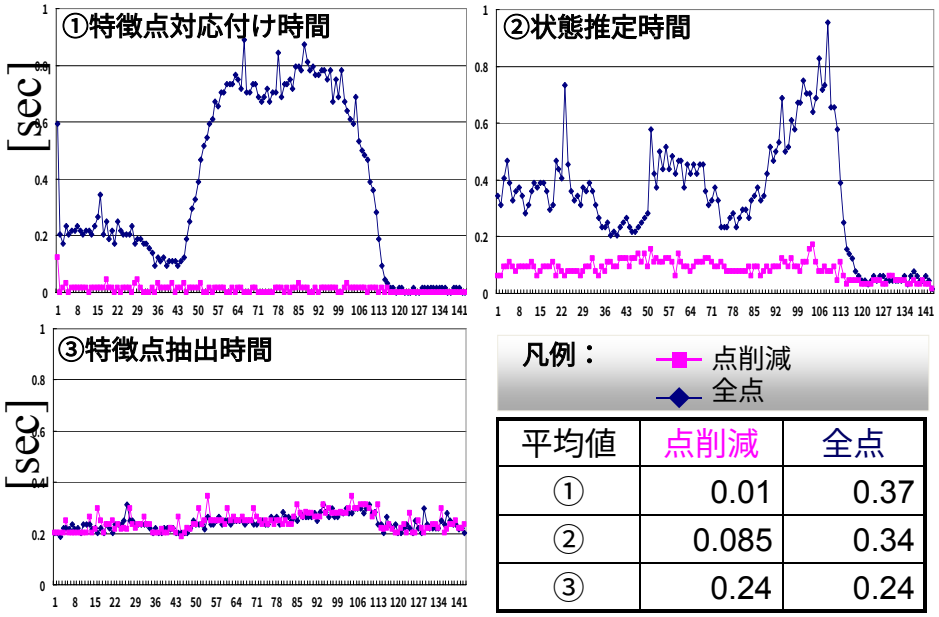
	x[mm]	z[mm]	φ[deg]
誤差平均	0.45	6.99	0.21
分散	558.06	575.12	1.96
標準偏差	**23.62**	**23.98**	**1.40**

●処理時間　**0.98** [sec/frame]

推定値と真値の誤差

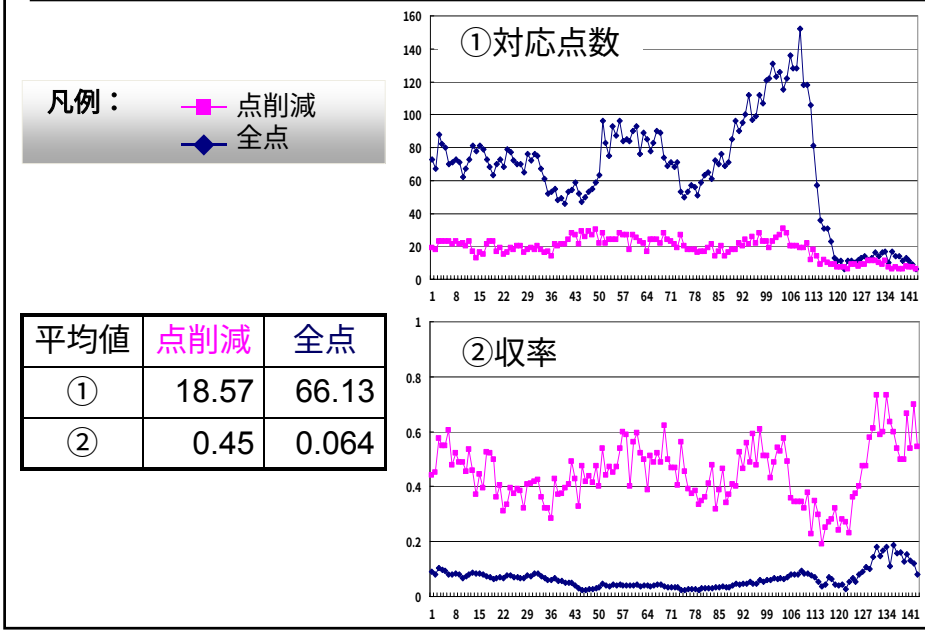


投影点削減効果 ~処理時間~



平均値	点削減	全点
①	0.01	0.37
②	0.085	0.34
③	0.24	0.24

投影点削減効果 ~対応点数と収率~

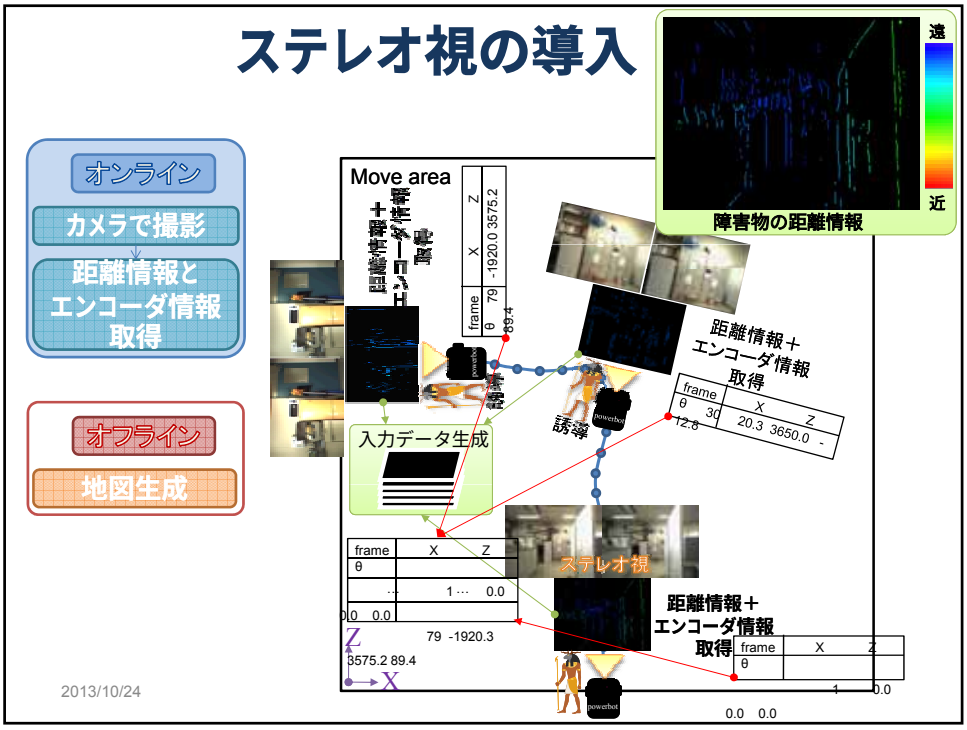


平均値	点削減	全点
①	18.57	66.13
②	0.45	0.064

実機による走行実験結果

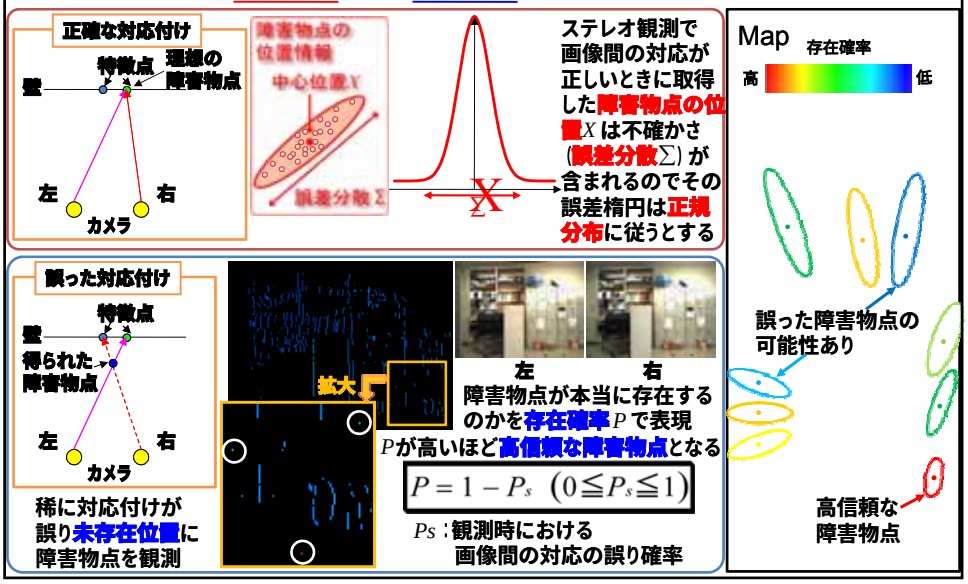
※　時間の都合上3倍速で再生

ステレオ視の導入



生成される地図

- 障害物点の位置情報とその存在確率で表された地図

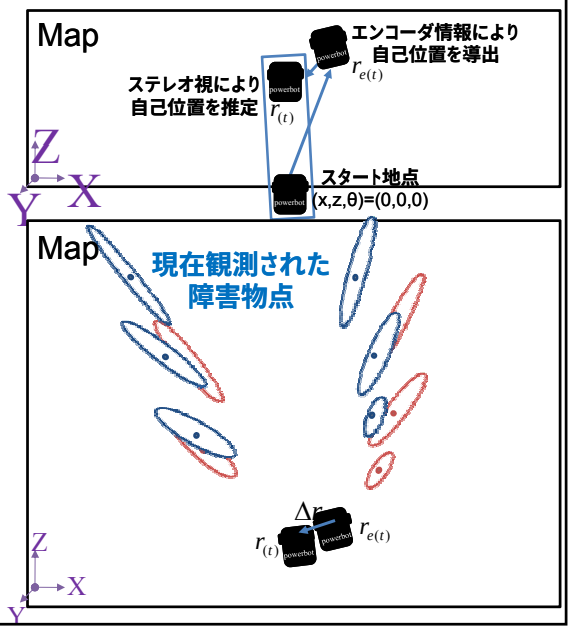


自己位置の推定

$r_{e(\cdot)}$：エンコーダによるロボットの位置(x,z,θ)
$r_{(\cdot)}$：自己位置推定後のロボットの位置(x,z,θ)
t：時刻

- エンコーダ情報から自己位置を導出
- Δr にもとづいて自己位置を修正

$$\Delta r = r_{(t-1)} - r_{e(t-1)}$$
$$r_{(t)} = r_{e(t)} + \Delta r$$

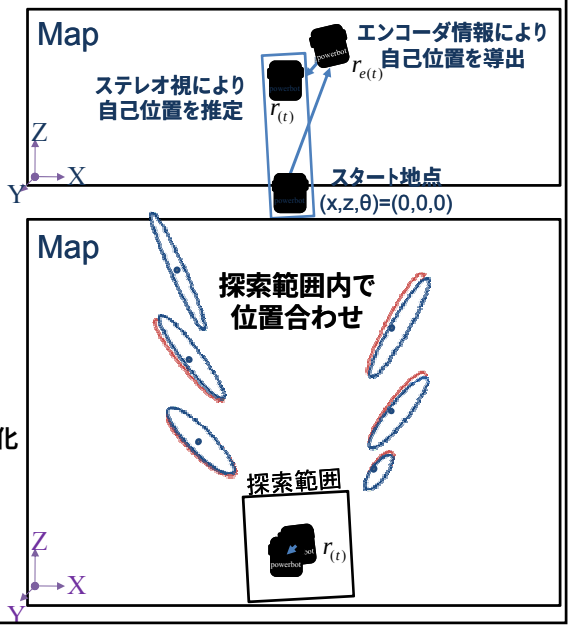


2013/10/24

自己位置の推定

$r_{e(\cdot)}$:エンコーダによるロボットの位置(x,z,θ)
$r_{(\cdot)}$:自己位置推定後のロボットの位置(x,z,θ)
t :時刻

- エンコーダ情報から自己位置を導出
- Δr にもとづいて自己位置を修正
- 探索範囲内で自己位置を推定
  - 地図を仮想的にボクセル化

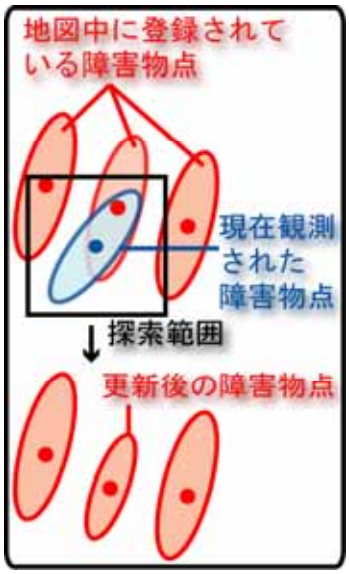


2013/10/24

地図の更新

現在観測された障害物点が
地図中に登録されている障害点と
同じ障害物点を表しているなら
それらを一つに統合して
地図を更新

- 対応点の探索
- 位置情報の更新
- 存在確率の更新

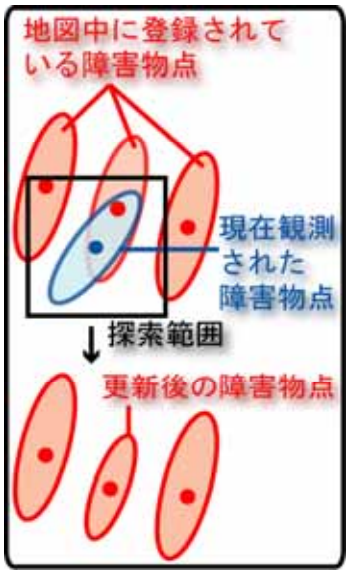


位置情報の更新

- 対応点が発見されれば障害物点の位置 X と誤差分散 Σ を存在確率 P から最尤推定に基づき統合して更新

$$X = \frac{P_n \Sigma_n^{-1} X_n + P_m \Sigma_m^{-1} X_m}{P_n \Sigma_n^{-1} + P_m \Sigma_m^{-1}}$$
$$\Sigma = P_n \Sigma_n^{-1} + P_m \Sigma_m^{-1}$$

n：現在観測された障害物点
m：地図中に登録されている障害物点



存在確率の更新

- 統合された障害物点
  - 障害物点の位置情報が統合されればその存在確率も更新

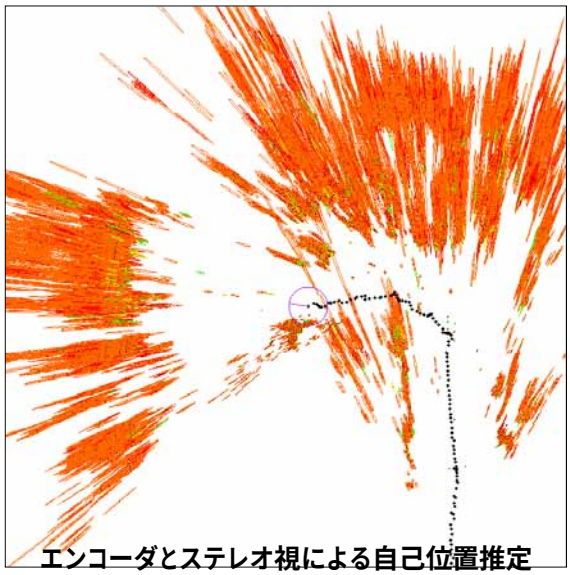
$$P = 1 - (1 - P_n)(1 - P_m)$$

n：現在観測された障害物点
m：地図中に登録されている障害物点

- 空き領域
  - 現在観測された障害物点における空き領域内に地図中に登録されている障害物点があればその存在確率を更新

$$P = (1 - P_n) P_m$$

生成された地図

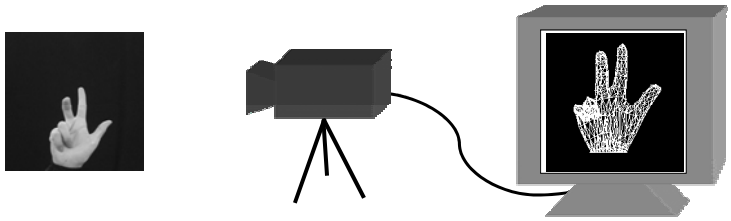


2013/10/24

エンコーダとステレオ視による自己位置推定

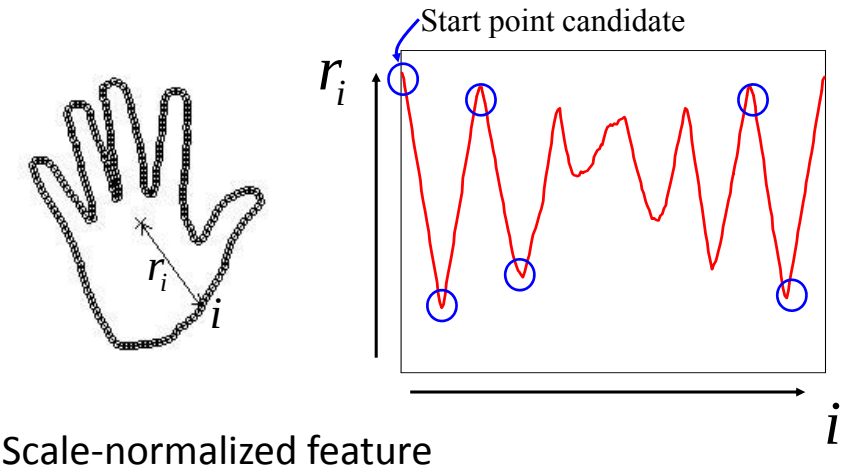
Real-time 3-D hand posture estimation from 2-D appearance

→ Gesture recognition for human interface



Input image　　Camera　　Estimation result

Contour Feature Extraction



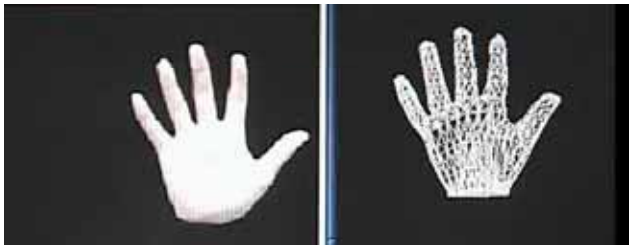
- Scale-normalized feature
- Multiple start points are tried

Previous Approaches of Gesture Estimation

	3-D Model Fitting	Direct Image Matching
Single Image	Arbitrary postures High computation cost	Limited postures Low computation cost
Sequential Image	Motion constraints can be derived from the model	Motion constraint needs to be learned

Real time processing

experimental environment

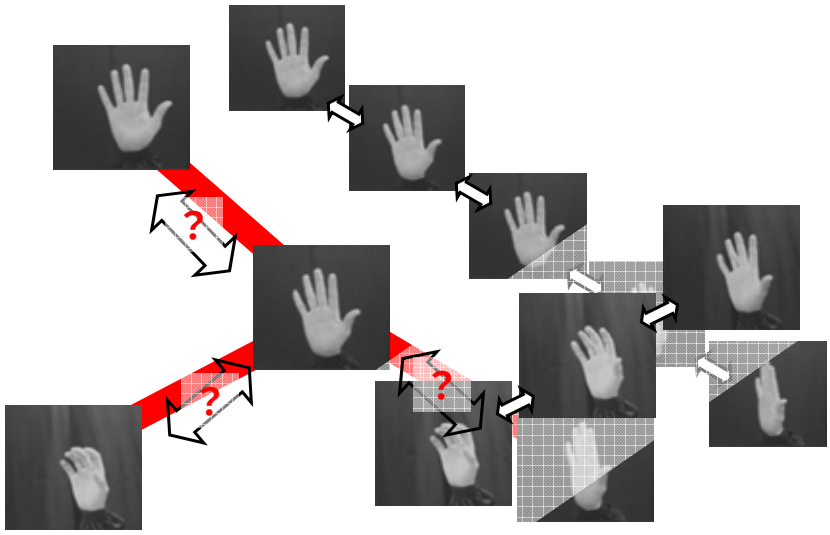


input image result

Previous Approaches of Gesture Estimation

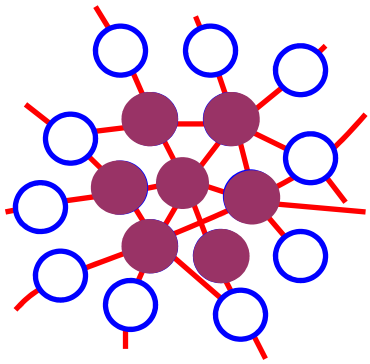
	3-D Model Fitting	Direct Image Matching
Single Image	Arbitrary postures High computation cost	Limited postures Low computation cost
Sequential Image	Motion constraints can be derived from the model	Motion constraint needs to be learned

Learning of Shape Transition



Efficient Matching Using Transition Network

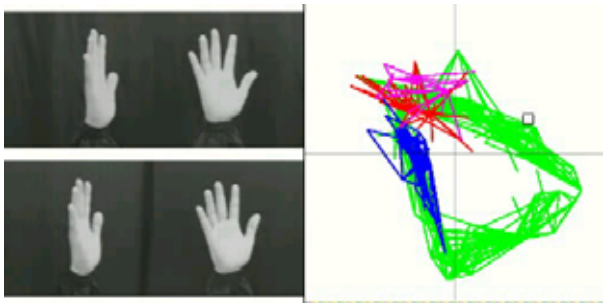
- Try only possible postures which are reached from the current posture

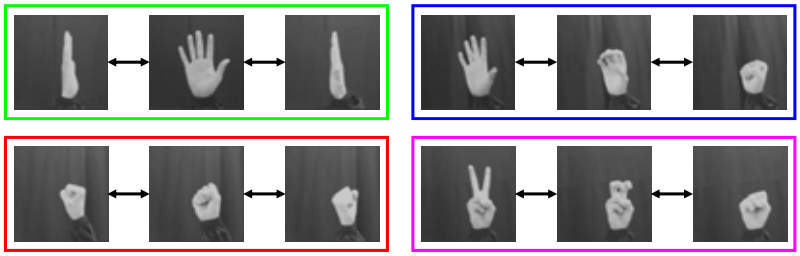


Shape Estimation Result

Input image

Matched model

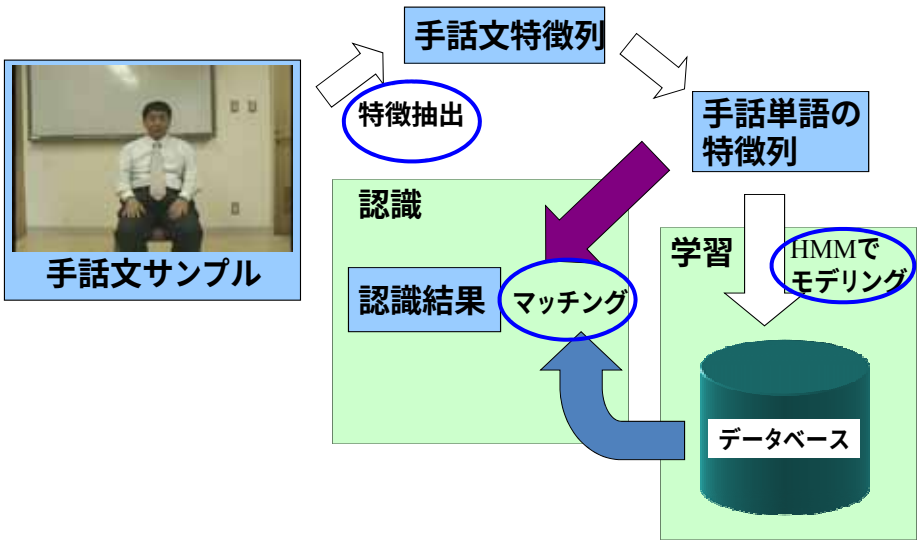




Hand Shape Estimation under Complex Backgrounds for Sign Language Recognition

手話認識

最初は特定領域研究「情報福祉」の中で



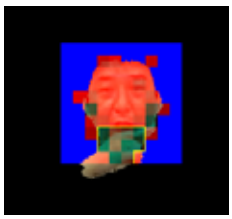
特徴量抽出

手話画像 ➡ 背景差分で人物領域を抽出 ➡ 肌色領域抽出 ➡ 各領域を手、顔に対応させる ➡ 特徴量を計算

隠蔽がある場合は、隠蔽前後の手･顔のテンプレートから各領域を決定

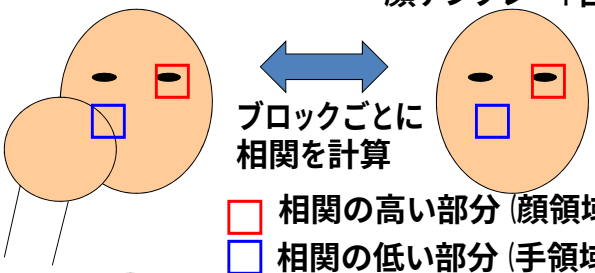
手話画像　人物領域　肌色領域

隠蔽時の処理



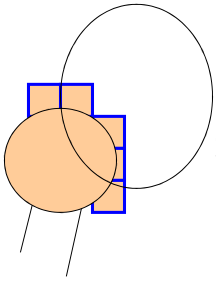
隠蔽状態の肌色領域

保存してある
顔テンプレート画像



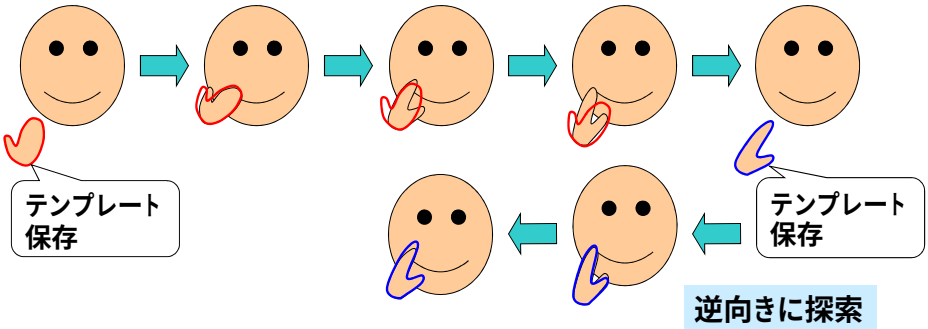
ブロックごとに
相関を計算

□ 相関の高い部分（顔領域）
□ 相関の低い部分（手領域）

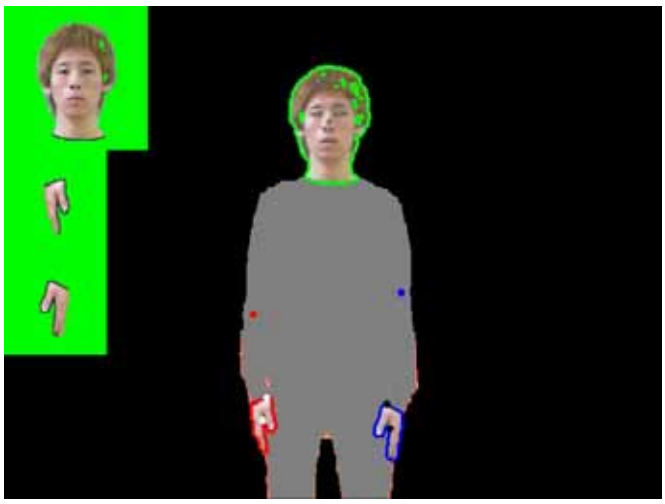


大体の手領域を取得できるため、
・テンプレートマッチング精度の向上
・手の形状が変化する際に利用

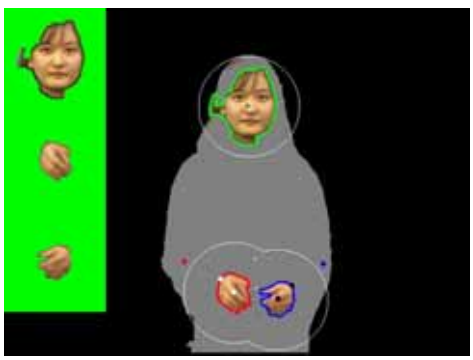
テンプレートマッチングによる分離

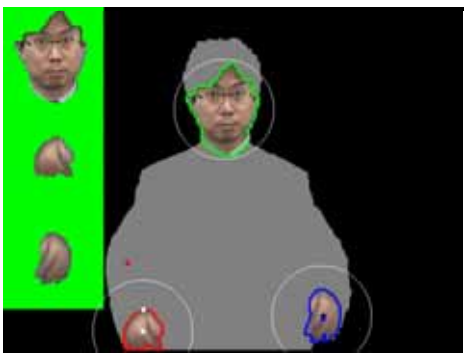


画像処理の結果



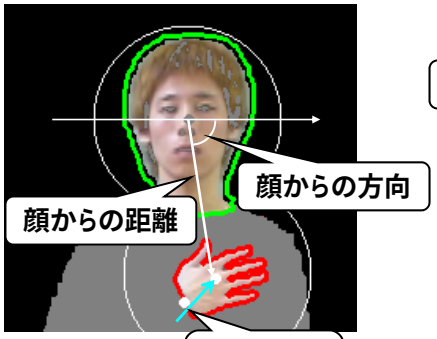
特徴抽出成功例



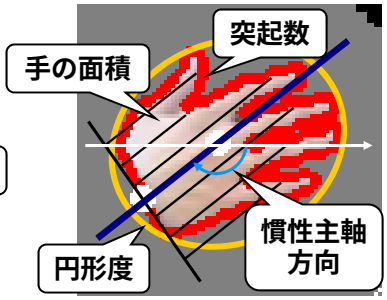


手話特徴量

位置に関する特徴量



形状に関する特徴量



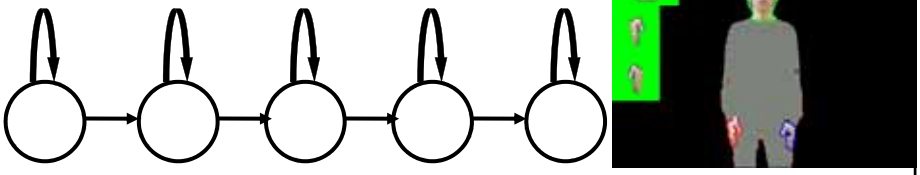
改善結果1

右手と顔の隠蔽

手話認識のための学習 (HMM)

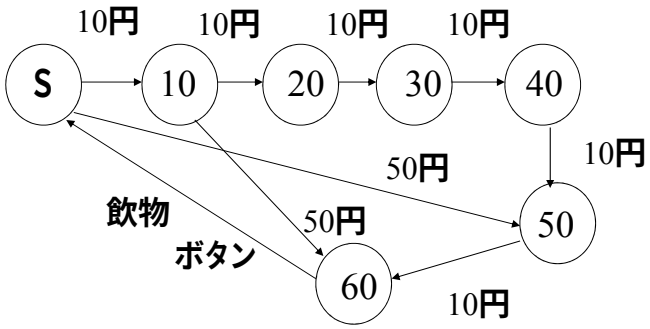
- HMMはLeft-to-Right
- **単語ごとに状態数を設定する必要がある**
- **手の移動中、静止中、手の形を変化時に対して各々状態を1つ割り付ける**

状態数決定の例 (状態数：5)

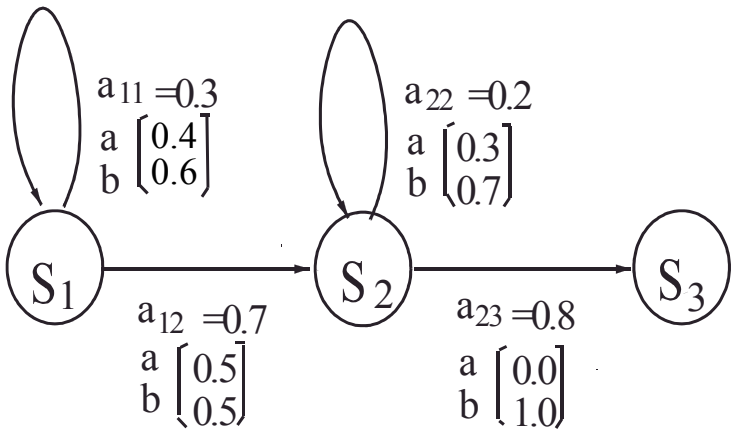


初期状態 移動中 静止中 移動中 最終状態

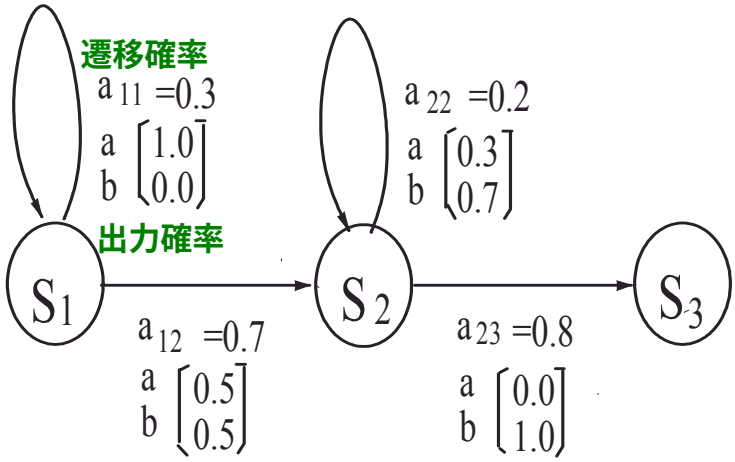
有限オートマトン



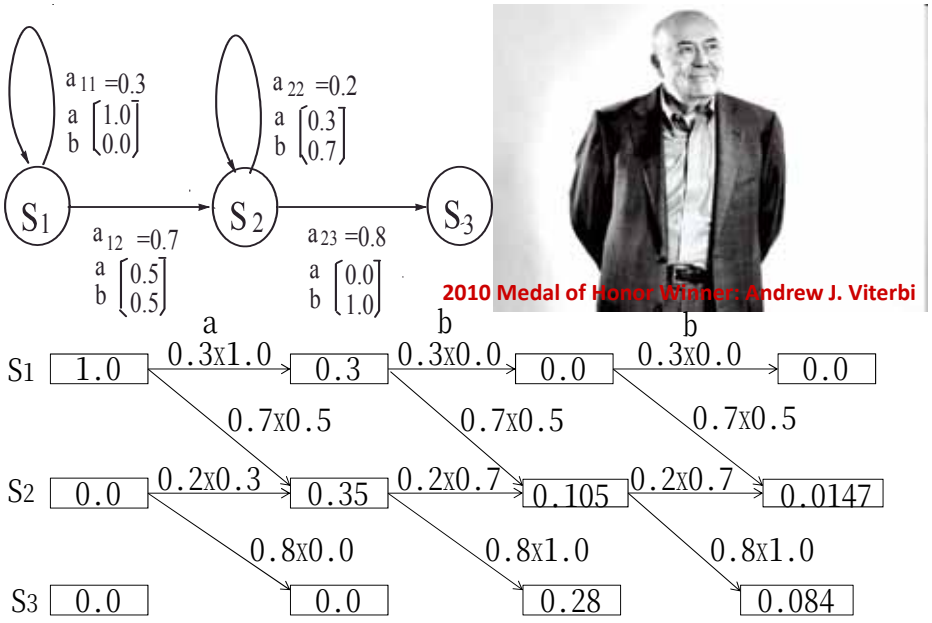
確率的オートマトン



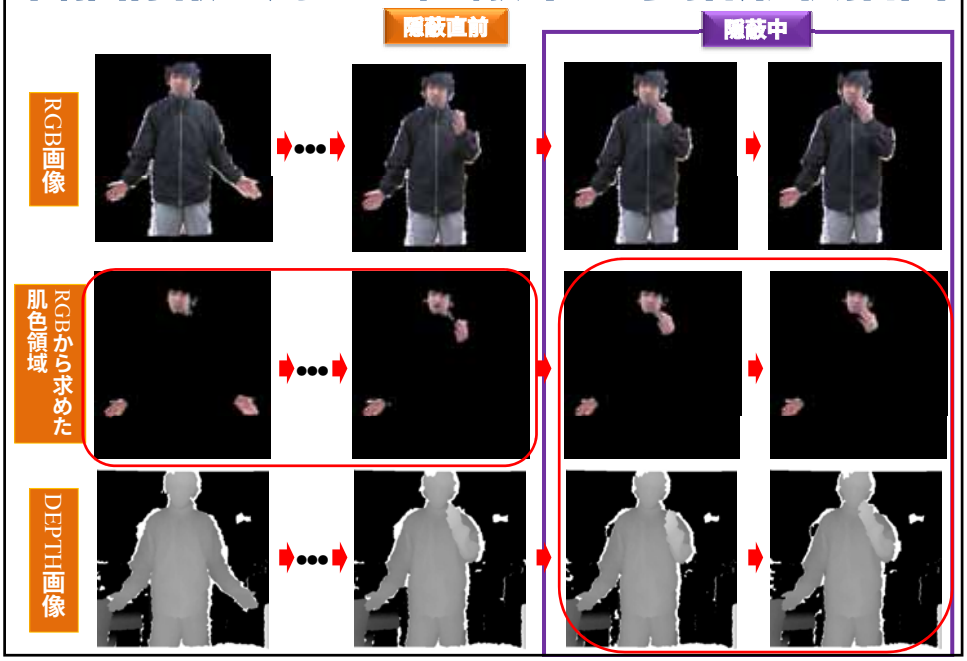
HMMの例



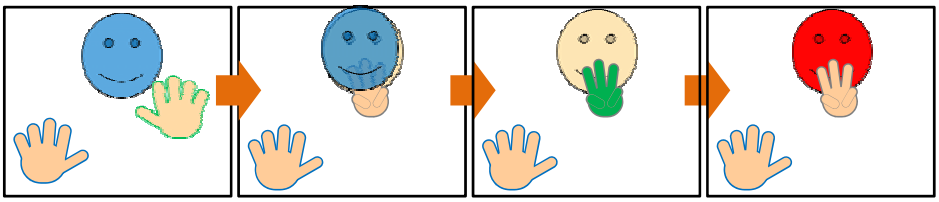
Viterbiアルゴリズムのトレリス上での計算



距離情報を用いた隠蔽中の手指形状抽出



処理の概要



隠蔽前の顔の距離を記憶

顔の距離テンプレートを照合

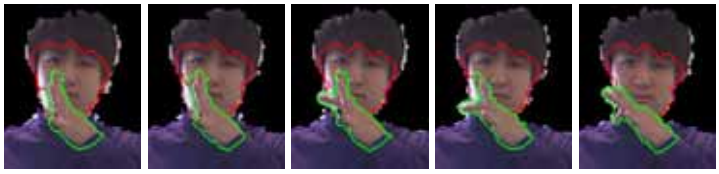
手領域抽出

顔領域抽出

99

色情報のみ

提案手法



100

コンピュータビジョンの今後

- 人間の視覚に迫る、夢を追う
  - 人の能力に近づける
    - 少数の例による学習
    - 生物に埋め込まれた能力
- 応用の拡大
  - 画像処理を部品として研究
  - マルチモーダル処理（インターネットから知識獲得）

レポート課題

shirai@ci.ritsumei.ac.jp
www.i.ci.ritsumei.ac.jp/~shirai/

１．講義した中で重要と思う技術を１つあげ、それを説明するとともに、なぜ重要化をできるだけ具体的に述べなさい。

２．上記以外で、講義した中で自分の研究で応用してみたい技術を説明するとともに、どのような応用であるかをできるだけ具体的に述べなさい。なお、現在の研究でなく、将来行いたい研究あるいは仕事でもかまわない。

３．講義の感想、講義以外で聴きたかったこと、及び来年の講義に参考になることを簡単に述べなさい。