

マニピュレーション ロボットのための 3次元物体認識研究の現状と展望

～ 生産・物流から生活支援まで ～



中京大学 工学部
橋本 学

mana@isl.sist.chukyo-u.ac.jp
<http://isl.sist.chukyo-u.ac.jp/>



本日の話題

1. ロボットビジョンの課題
2. 現状：3次元センサ
3. 現状：特定物体ハンドリングのための物体認識
4. 現状：柔軟なロボット動作のための物体認識
5. ARCとWRSにみる実課題への対応
6. 展望：タスク生成のための物体認識
7. 残された課題

1. ロボットビジョンの課題

2. 現状：3次元センサ

3. 現状：特定物体ハンドリングのための物体認識

4. 現状：柔軟なロボット動作のための物体認識

5. ARCとWRSにみる実課題への対応

6. 展望：タスク生成のための物体認識

7. 残された課題

ロボットビジョンの利用分野



生産・物流・家庭分野における物体認識の課題



物体モデル利用 可	物体モデル利用 難
品種 複数だがほぼ既知	品種 多量かつ未知
Pick and Place	Pick and Operation
動作パターン種類 少（事前に生成）	動作パターン種類 多（その場で生成）

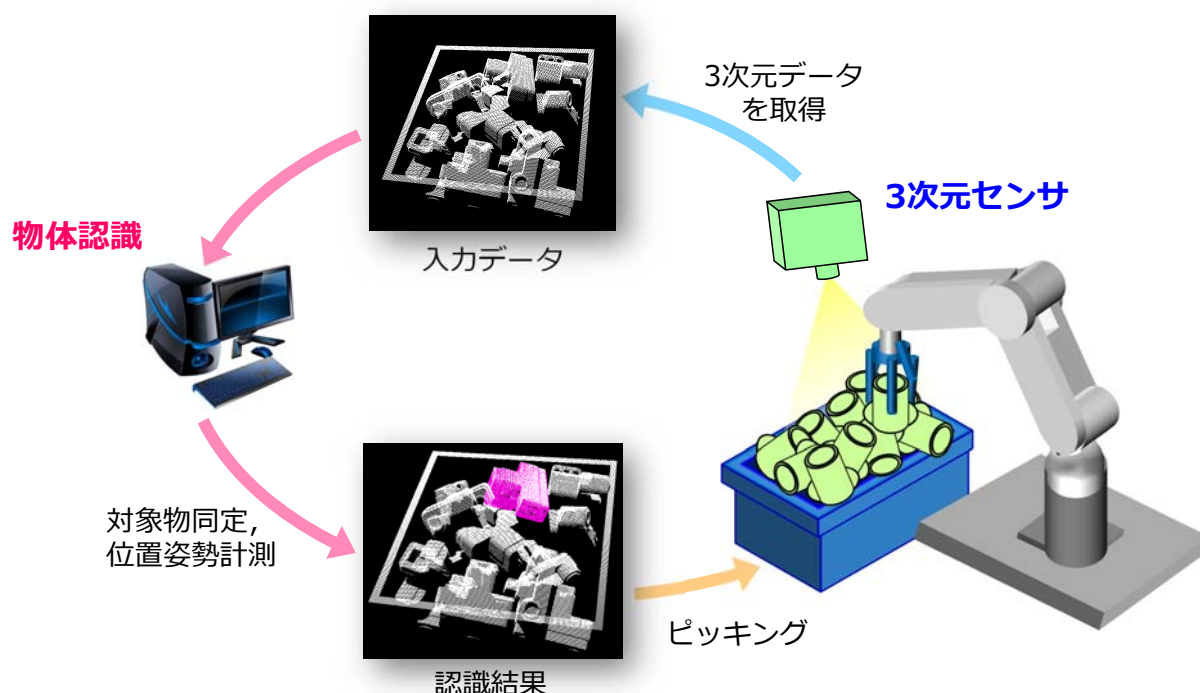


本日の話題

1. ロボットビジョンの課題
2. 現状：3次元センサ
3. 現状：特定物体ハンドリングのための物体認識
4. 現状：柔軟なロボット動作のための物体認識
5. ARCとWRSにみる実課題への対応
6. 展望：タスク生成のための物体認識
7. 残された課題

ロボットビジョンの基本構成

ロボットビジョン = 3次元センサ + 物体認識 (アルゴリズム)



3次元センサの分類 (光学センサ)

青 : 文献例 緑 : 商品例

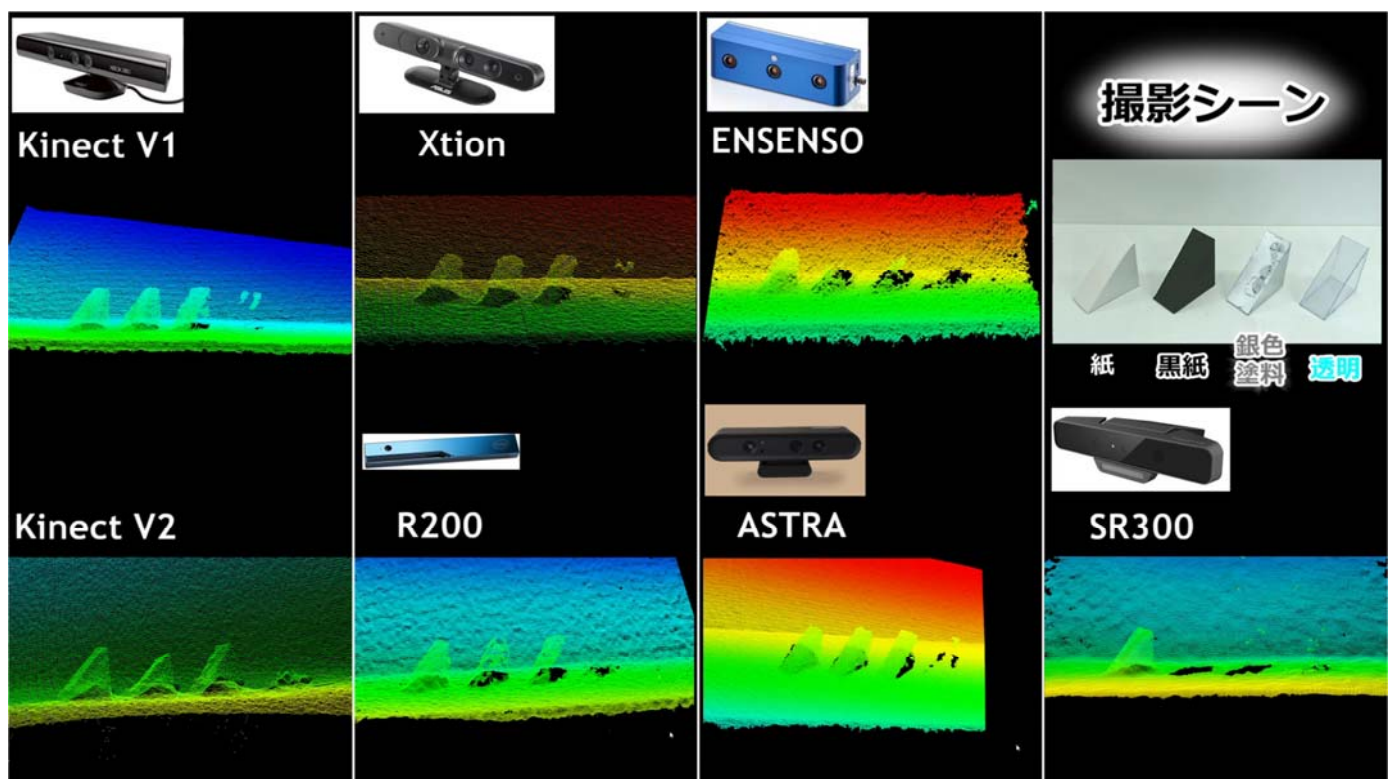
	パッシブ	アクティブ
三角測量	ステレオ視 (2眼, 3眼, 多眼) Bumblebee 2 / XB3 (Point Grey 2006 / 2007) TVS (三次元メディア 2011) SV-M-S1 (リコー) Bumblebee2 TVS SV-M-S1 マルチベースラインステレオ法 [Okutomi1993] 視体積交差法 [Matsuyama2002]	光切断法 (スポット光) TDS-A (パルステック 1997) 光切断法 (スリット光) VIVID9i (KONICA MINOLTA 2004) 空間コード化法 [Posdamer1982], [Sato1985] Cartesia (SPACEVISION 2004) MELFA-3D Vision (三菱電機 2013) RV1100 (キャノン) ランダムドットパターン投光法 [Hashimoto1999] Kinect v1 (Microsoft 2010) Xtion PRO LIVE (ASUS 2011) Leap Motion (2013) ASTORA (Orbbec 2016) SR300 (Intel 2016) ステレオ+パターン投光法 ENSENSO N10 (iDS 2012) R200 (Intel 2015) 位相シフト法 [Halioua1989], [Zhao1994] MELFA-3D Cartesia RV1100 Kinect v1 Xtion PRO LIVE SR300 ASTRA ENSENSO N10 R200
同軸測量	Shape(depth) from (De) Focus [Hiura1999] Shape from Motion	TOF: Time Of Flight Swiss Ranger SR3000 / SR4000 (MESA 2005 / 2008) D-Imager (パナソニック2010) DepthSense325 (SoftKinetic2012) Kinect v2 (Microsoft 2014) 照度差ステレオ [Woodham1980] Kinect v2 SR4000 DepthSense325

3次元センサの比較

販売開始年度順

	Kinect V1	Xtion	ENSENSO	Kinect V2	R200	Astra	SR300
							
メーカー名	Microsoft	ASUS	iDS	Microsoft	Intel	Orbbec	Intel
型番	L6M-00005	XTION PRO	N10-408-18	GT3-00005	82634DSB2P	Orbbec Astra	825351VCQSPL04N
販売時期	2010	2011	2012	2014	2015	2016	2016
概略価格	約1万5000円	約2万5000円	約60万円	2万1578円	約1万9000円 (開発ツール セット料金)	約1万7000円	約1万8000円
デプス解像度	320x480で撮影 →640x480に変換	320x240	752x480	512x424	628x468	640x480	640x480
撮影範囲 (角度・距離)	水平57°× 垂直43° 0.4～3.0m	水平58°× 垂直45° 0.8～3.5m	 0.3～2.0m	水平70°× 垂直60° 0.5～4.5m	 0.5～3.5m	水平60°× 垂直49.5° 0.6～8.0m (最適0.6～5.0m)	 0.2m～1.5m
計測形式	ランダムドット パターン (Light Coding)	ランダムドット パターン (Light Coding)	Stereo + Light Coding	Time of Flight	Stereo + Light Coding	ランダムドット パターン (Light Coding)	ランダムドット パターン (Light Coding)
インター フェイス	USB 2.0	USB 2.0	USB 2.0	USB 3.0	USB 3.0	USB 2.0	USB 3.0
電源	○	×	×	○	×	×	×

3次元センサの比較



ポイントクラウドデータ（3次元点群）

■ xyz 3 次元座標系におけるデータ点の集合

代表的な点群のデータ構造（.pcd）

PCL で定義されたフォーマット

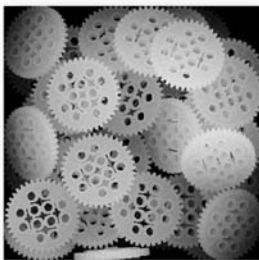
```
1 # .PCD v0.7 - Point Cloud Data file format
2 VERSION 0.7
3 FIELDS x y z
4 SIZE 4 4 4
5 TYPE F F F
6 COUNT 1 1 1
7 WIDTH 307200
8 HEIGHT 1
9 VIEWPOINT 0 0 0 1 0 0 0
10 POINTS 307200
11 DATA ascii
12 -0.600861 0.449636 1.000000
13 -0.598983 0.449636 1.000000
14 : : :
15 : : :
```

ヘッダー情報 { 1-11
点群情報 { 12-15

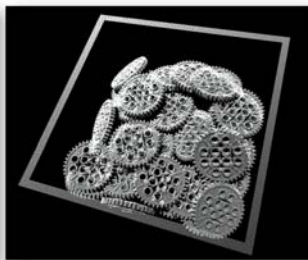
307200 行

Kinect v1, Kinect v2, Xtion は, PCL を用いることによって .pcd ファイル形式で保存可.

■ 距離画像と点群データ



距離画像



3 次元点群

	距離画像	3 次元点群
隣接点へのアクセス方法	データが整列されているので隣接データへの連続アクセスが容易.	データが整列されていないので最近傍探索が必要.
デプス解像度	値が濃度値分解能によって離散化される.	一般には離散化されない. 実空間データが利用可能.
主な表示ツール	画像 Viewer (Irfan View, Windows フォトビューアーなど)	OpenGL, Mesh Lab, PCL の Viewer など
利用可能なオープンソースライブラリ	OpenCV, OpenGL	Point Cloud Library (PCL), OpenGL

本日の話題

1. ロボットビジョンの課題
2. 現状：3次元センサ
3. 現状：特定物体ハンドリングのための物体認識
4. 現状：柔軟なロボット動作のための物体認識
5. ARCとWRSにみる実課題への対応
6. 展望：タスク生成のための物体認識
7. 残された課題

“物体認識”の2つのタイプ

① 特定物体認識

「このコップはどこ？」
(位置・姿勢の認識)



形状・色・模様などのモデルを使用



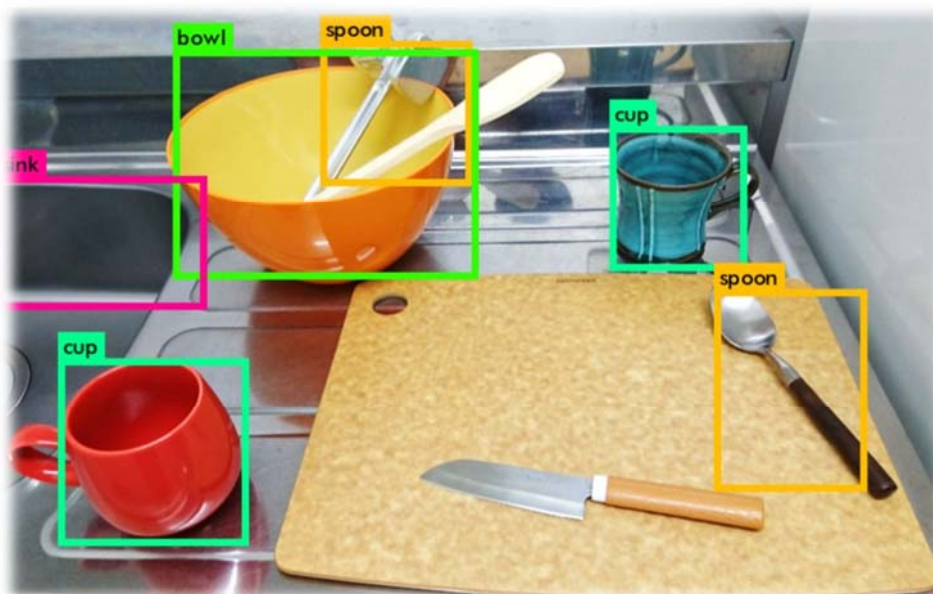
“物体認識”の2つのタイプ

② 一般物体認識

「これは何？」
(一般名称の認識)

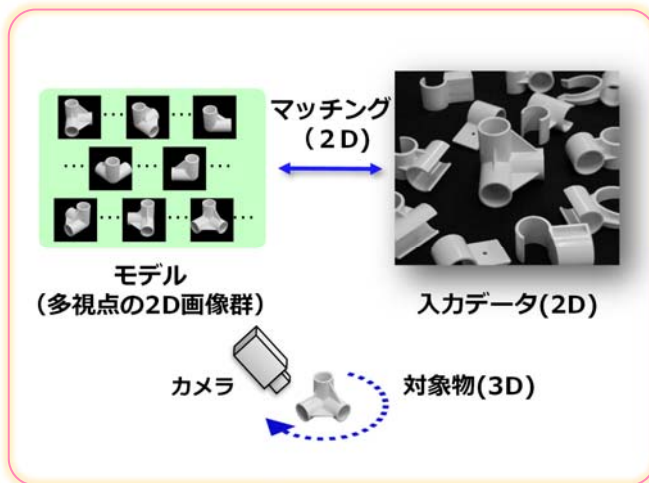


個々のモデルは使用しない（モデルレス）

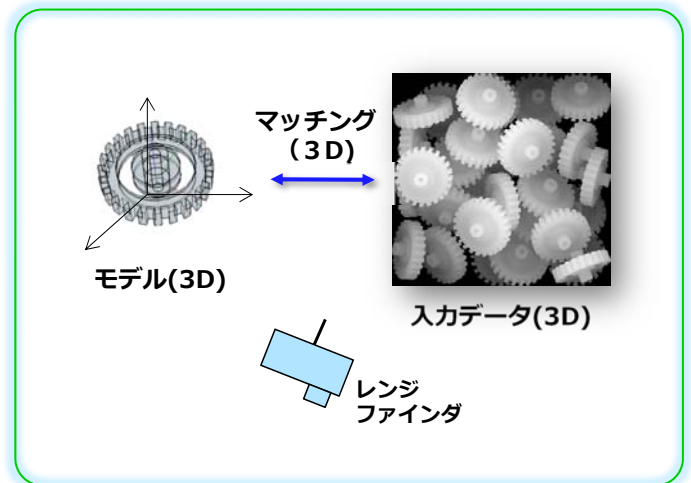


特定物体認識のための主要な2タイプ

アピランスペース物体認識 (2D-2Dマッチング)



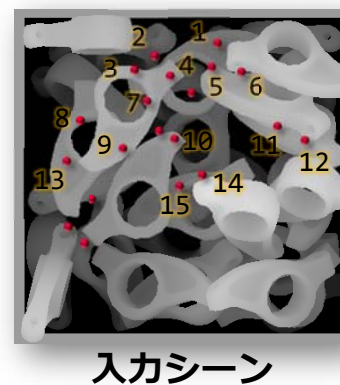
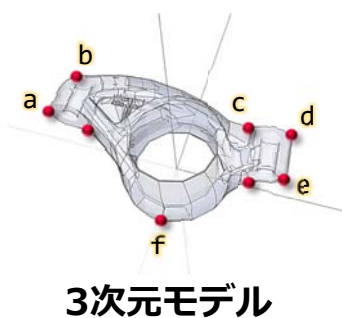
モデルベース物体認識 (3D-3Dマッチング)



	アピランスペース	モデルベース
手法	入力データを多視点画像群と照合	入力データを3次元モデルと照合
メリット	単眼カメラ利用できる	3次元モデルが入手容易に
課題	多量の辞書データが必要	3次元センサはまだ高価・大型

キーポイントマッチングの基本 (特徴量マッチング)

3次元データ同士のマッチング



	座標 (x,y,z)	特徴量 (ベクトル)
a	43.5, 50.2, 73.6	5, 2, 6, ..., 9, 20, ...
b	37.0, 25.3, 43.1	45, 2, 4, ..., 1, 3, ...
:	:	:
d	81.6, 25.2, 33.8	3, 52, 3, ..., 11, 4, ...
:	:	:
g	11.5, 9.5, 234.1	50, 32, 4, ..., 3, 13, ...
j	8.9, 794.4, 212.3	1, 95, 34, ..., 13, 5, ...
:	:	:

	座標 (x,y,z)	特徴量 (ベクトル)
1	53.5, 55.1, 71.4	74, 36, 35, ..., 19, 76, ...
2	64.2, 60.0, 71.4	103, 2, 45, ..., 1, 34, ...
:	:	:
10	101.6, 35.3, 71.8	5, 2, 6, ..., 9, 20, ...
:	:	:
21	25.1, 220.9, 5.1	99, 49, 3, ..., 20, 4, ...
24	734.7, 5.4, 20.1	8, 34, 24, ..., 92, 44, ...
:	:	:
27	21.2, 7.5, 265.6	3, 52, 3, ..., 11, 4, ...
30	95.0, 26.1, 578.8	58, 29, 99, ..., 45, 320, ...
:	:	:

マッチングの指標は「特徴量 (特徴ベクトル)」のみ

→ マッチングを成功させるためには **よい3次元特徴量** の設計が重要

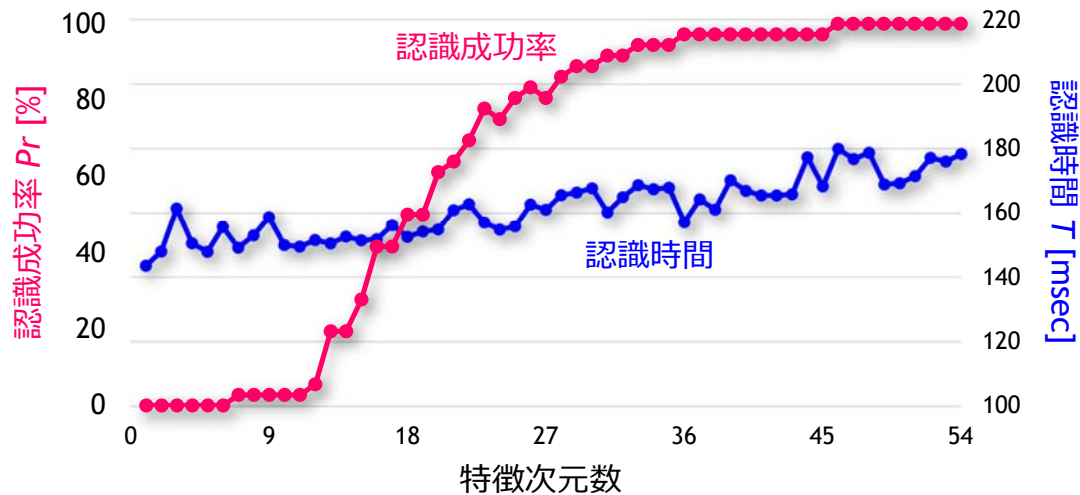
実験① 特徴量の次元数と認識性能

【実験方法】

実験データ：Stanford models の "Dragon" で構成

特徴量：SHORT 特徴量 54 次元（6 シェル x 9 ビンのヒストグラム）からランダムに要素を抽出
（特徴量の次元数 = ランダム抽出数）

【実験結果】



特徴量の次元数が性能を左右する。

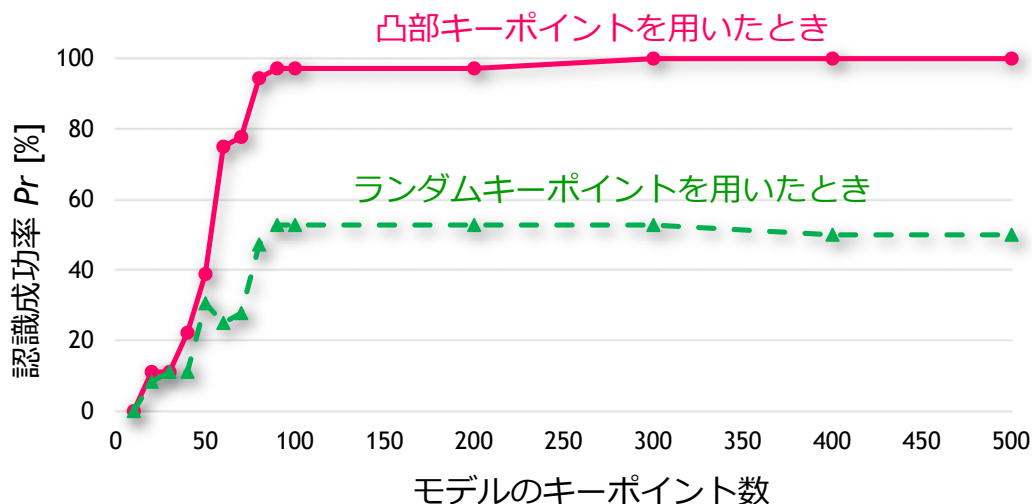
実験② キーポイント検出方法と認識成功率の関係

【実験方法】

SHORT 法：凸形状の点だけをキーポイントとして検出

ランダム法：ランダムにキーポイントを検出

【実験結果】



キーポイントの選び方が性能を左右する。

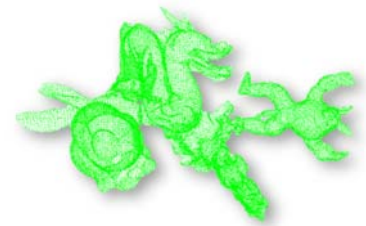
実験③ キーポイント数と認識性能の関係

【実験方法】

実験データ：Stanford models の "Dragon" で構成

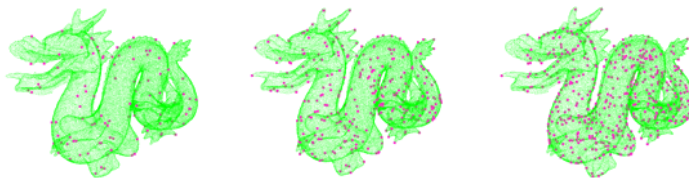
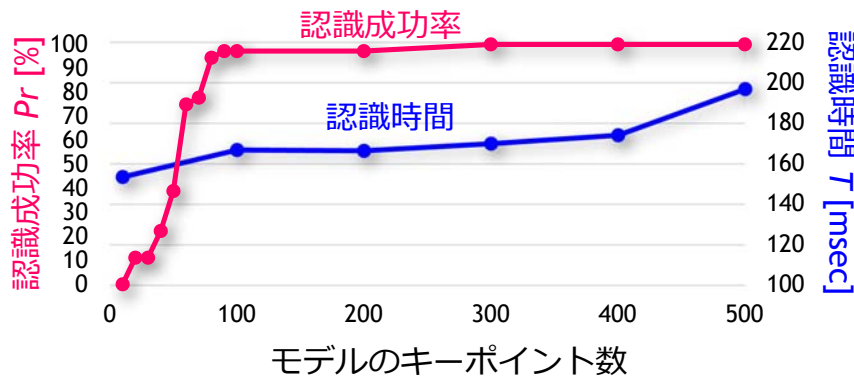
キーポイント：SHORT 法によって検出された点からランダムに選択

S.Takei, S.Akizuki, M.Hashimoto, "SHORT: A Fast 3D Feature Description based on Estimating Occupancy in Spherical Shell Regions", IVCNZ, 2015.



入力データ例

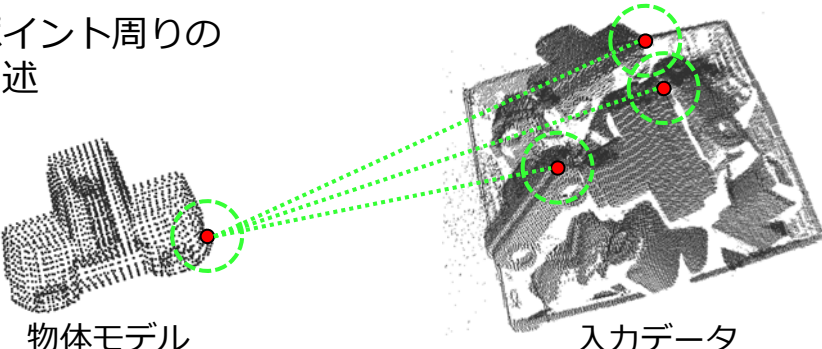
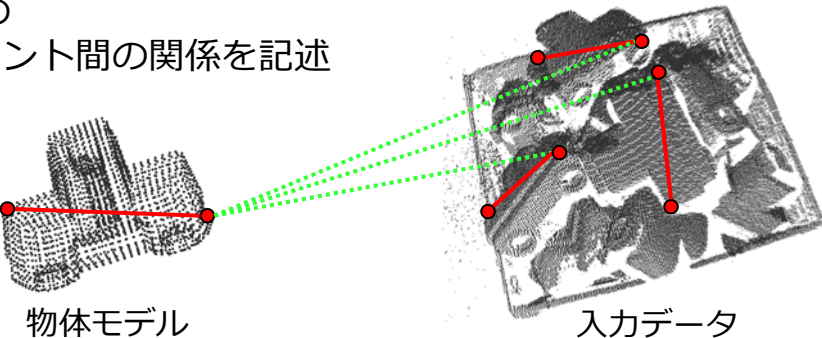
【実験結果】



キーポイント数が性能を左右する。

2つのタイプの3次元特徴量

キーポイントにどのような情報（特徴量）を付加するか？

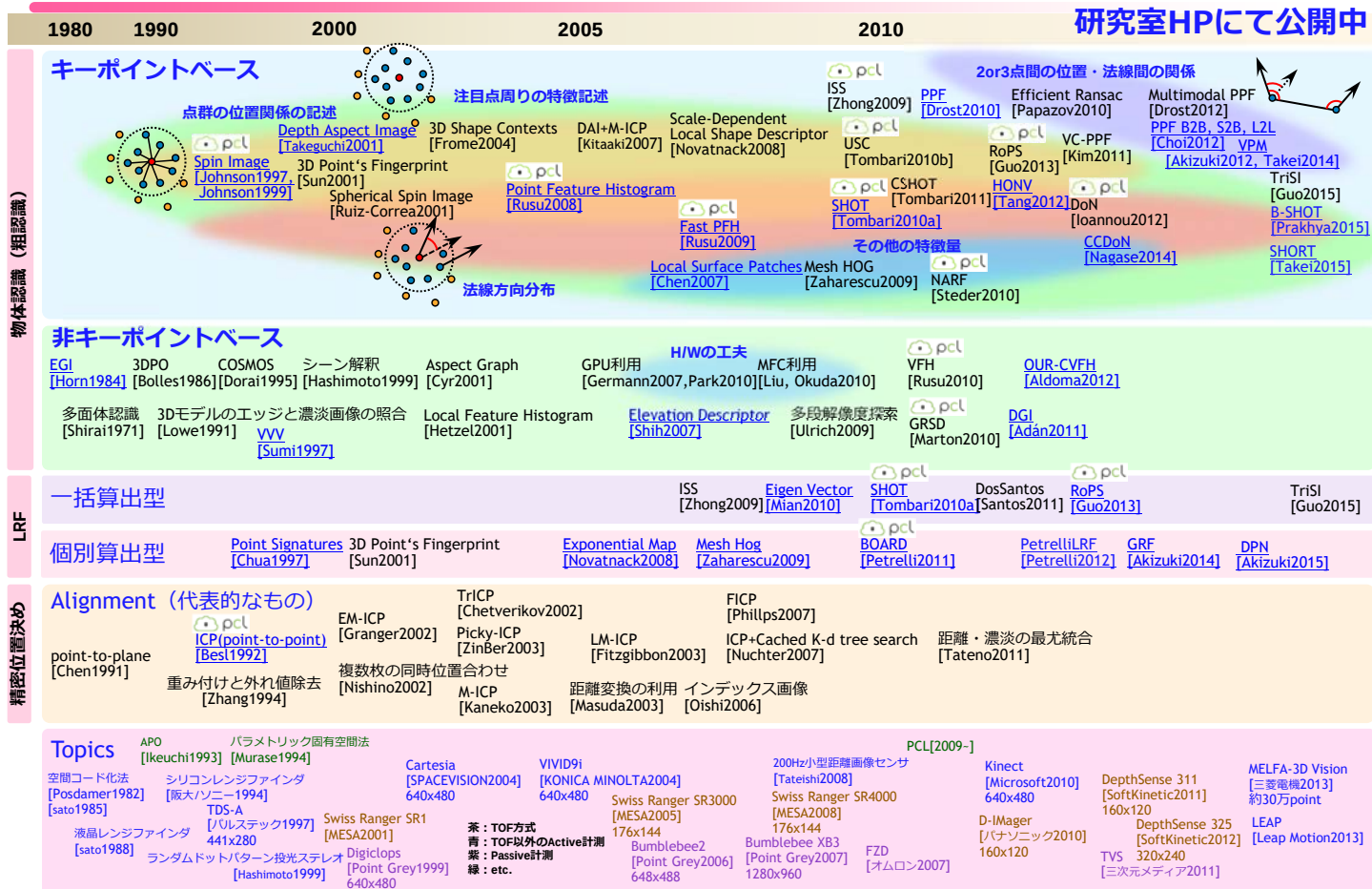
タイプ (A)	各キーポイント周りの 情報を記述  物体モデル 入力データ
タイプ (B)	2～3個の キーポイント間の関係を記述  物体モデル 入力データ

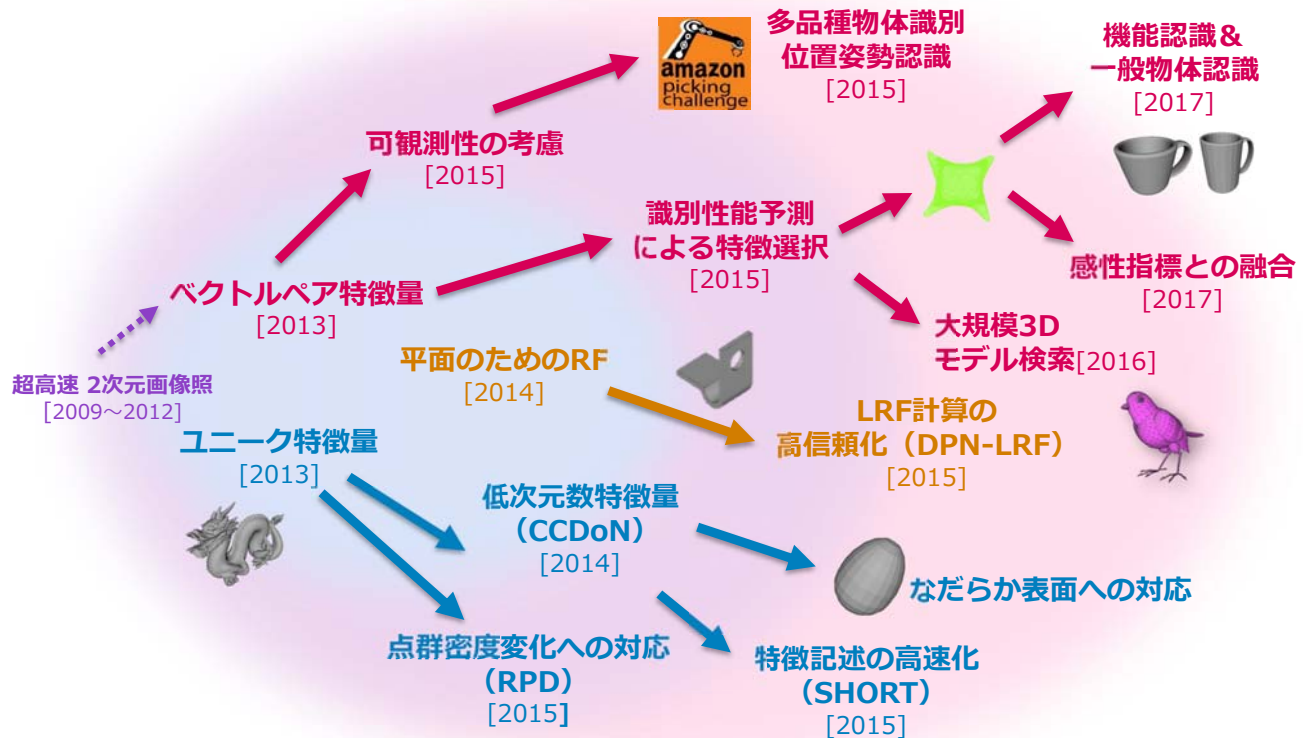
主な 3 次元特徴量

特徴量の記述方法			
(A) キーポイント周りの情報記述			(B) 複数点間の関係記述
座標データ利用	法線ベクトル利用	その他の情報を利用	2点間 or 3点間
<u>SHOT</u> , <u>PFH</u> , <u>CSHOT</u> , <u>B-SHOT</u> , <u>HONV</u>		<u>LSP</u> , <u>CCDoN</u> , <u>NARF</u> , <u>MHOG</u> , <u>RoPS</u>	<u>PPF</u> , <u>ER</u> , <u>VC-PPF</u> , <u>MPPF</u> , <u>PPF B2B</u> or <u>S2B</u> or <u>L2L</u> , <u>VPM</u>
<u>SI</u> , <u>SSI</u> , <u>DAI</u> , <u>3DSC</u> , <u>SDLSD</u> , <u>USC</u>	<u>FPFH</u> , <u>DoN</u>		

21

特定物体認識（モデルベース手法）の研究マップ



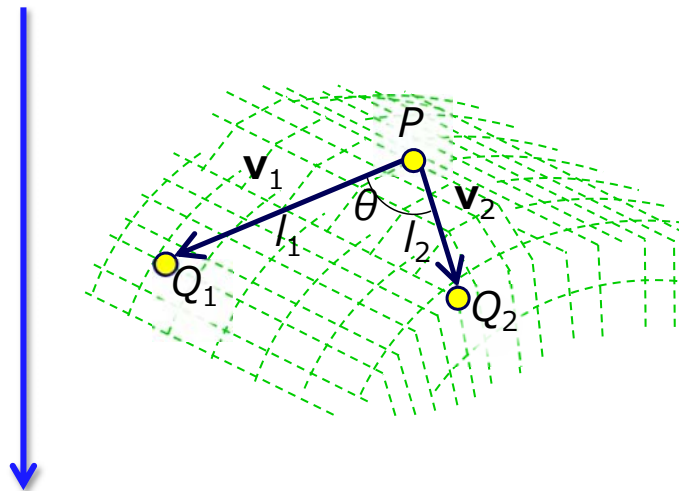


ベクトルペア特徴量による物体認識

3次元ベクトルペア

■ 3つの3次元点

- 3次元姿勢パラメータを決定するためには、3つの3次元点があればよい。



幾何学的パラメータ

$$l_1 = |\mathbf{v}_1| = |PQ_1|,$$
$$l_2 = |\mathbf{v}_2| = |PQ_2|,$$
$$\theta$$

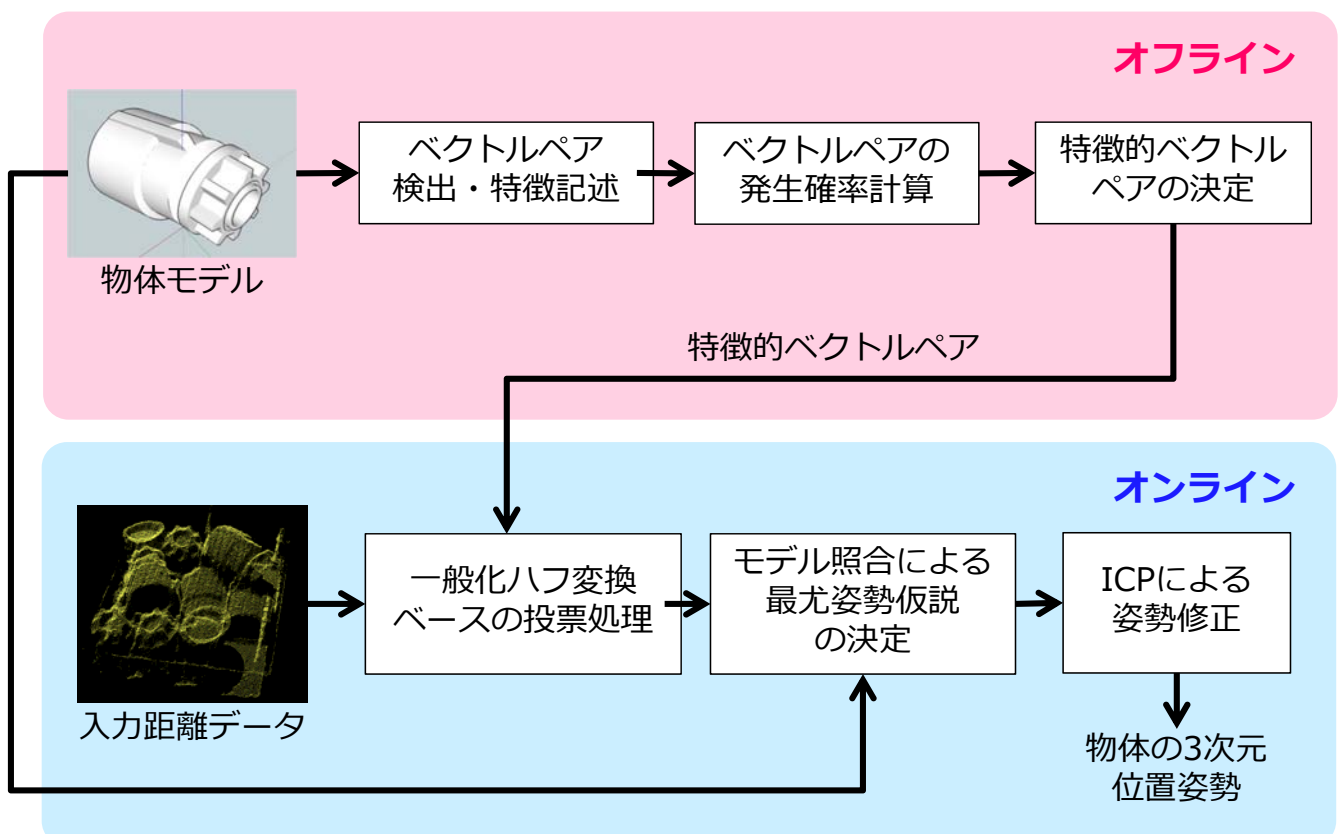
Shape description value

$$s_p, s_{q1}, s_{q2}$$

■ 視点を共有する2本のベクトル = 3次元ベクトルペア

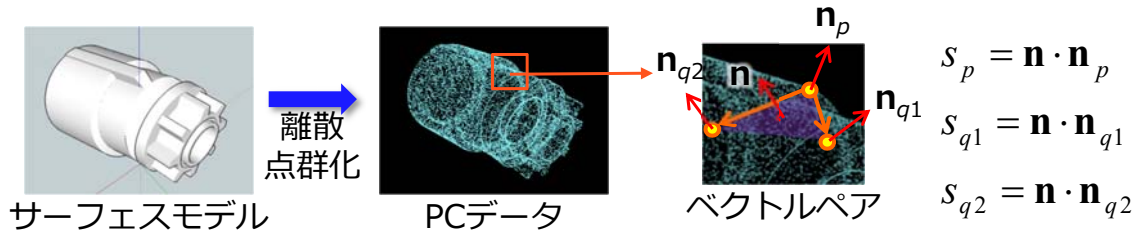
- この表現により、幾何学的な表現がシンプルになる。

ベクトルペアマッチング法：処理の流れ

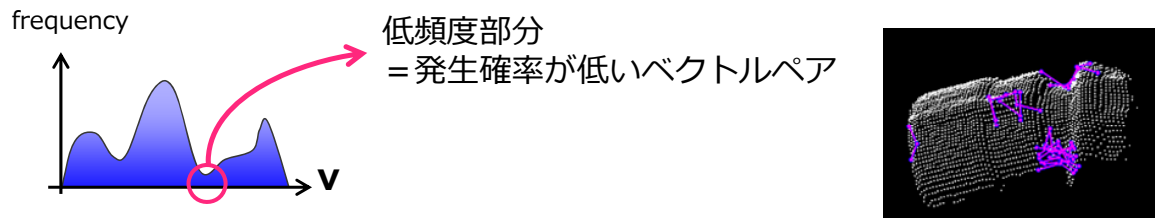


特徴的3-Dベクトルペアの抽出

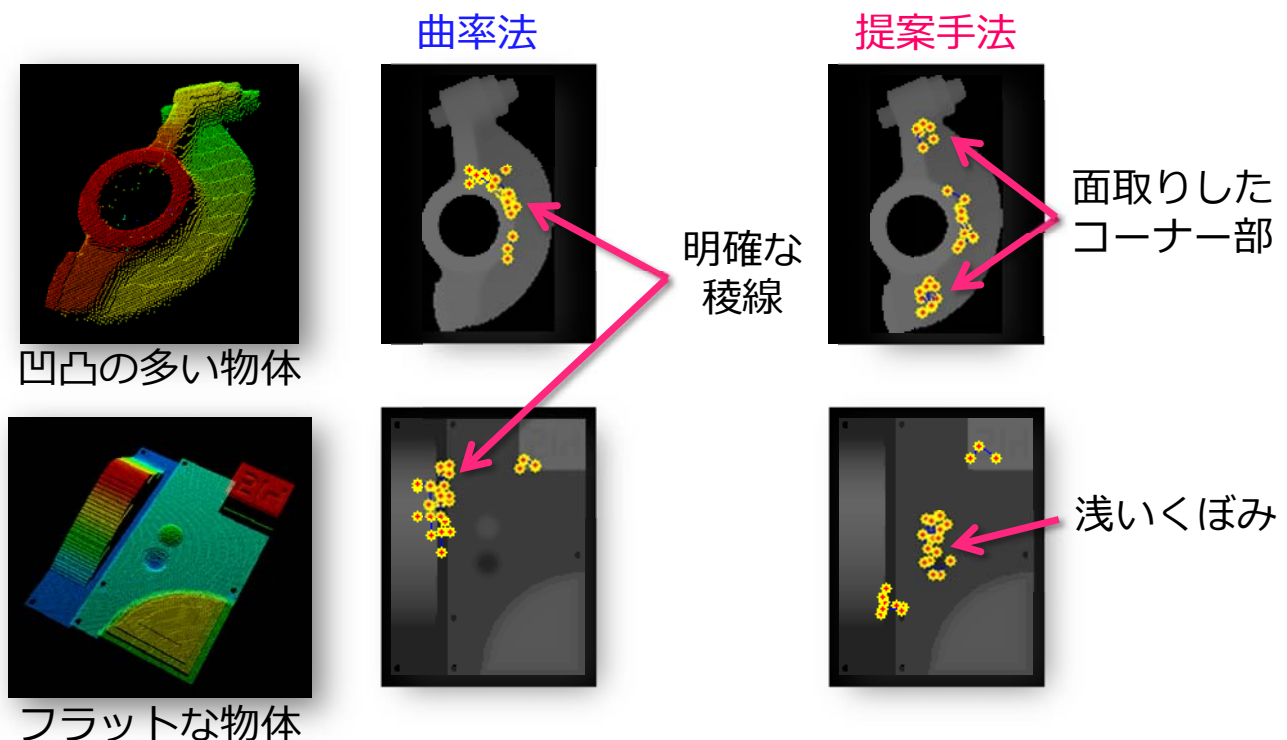
- 物体モデルにおいて「発生確率が低いベクトルペア」→**特徴的**
- ベクトルペアの特徴量 $\mathbf{v}=(s_p, s_{q1}, s_{q2})$ の発生確率を計算
 1. 法線分布を用いた特徴量の計算
 - 三角パッチの法線と、各端点の内積値



2. 特徴量の発生確率分布としての物体モデル表現
 - 物体内全てのベクトルペアを $\mathbf{v}$ の各値を軸とする頻度分布化

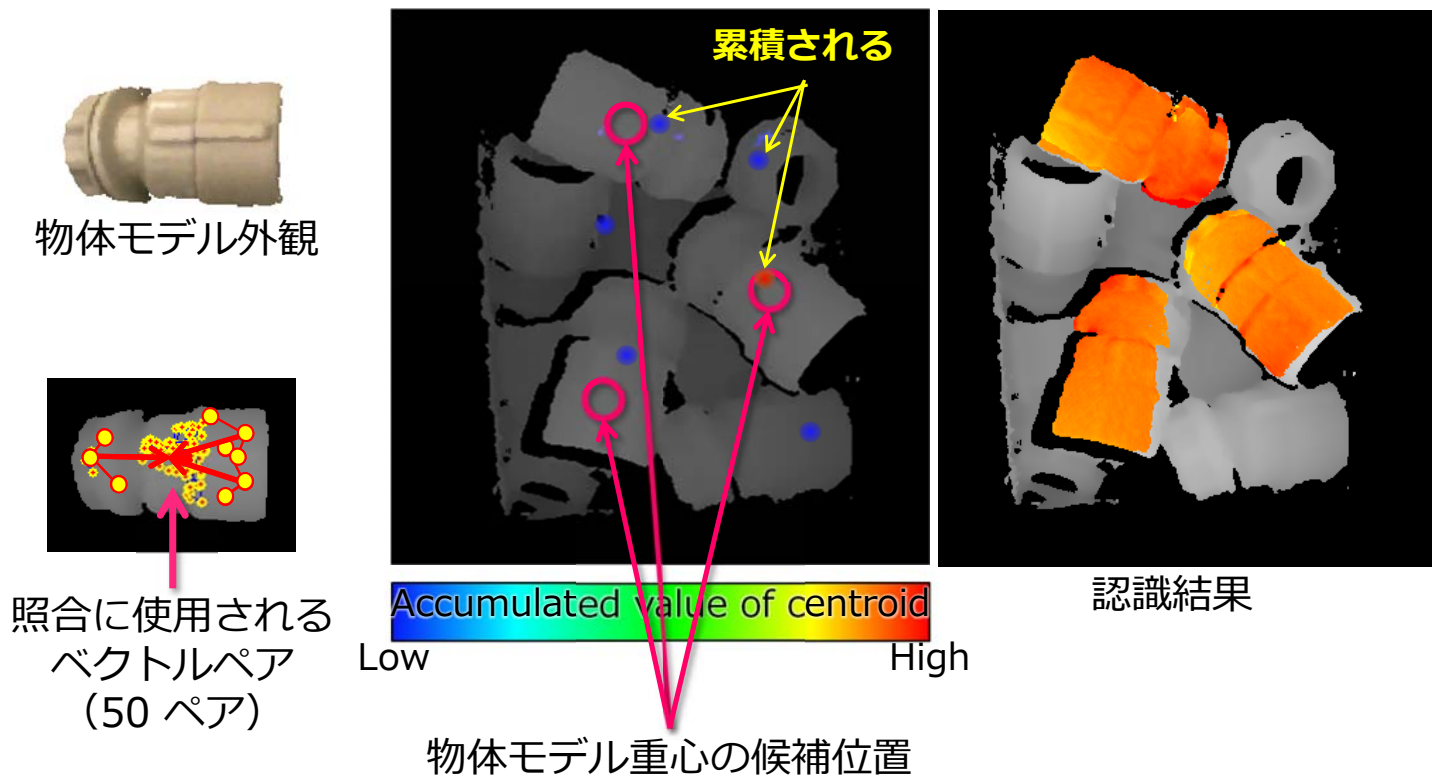


特徴抽出結果の例： 曲率法 v.s. 提案手法



3次元ベクトルペアは独自性の高い形状から抽出される。
この3次元ベクトルペアを照合に用いる。

3次元ベクトルペアマッチング

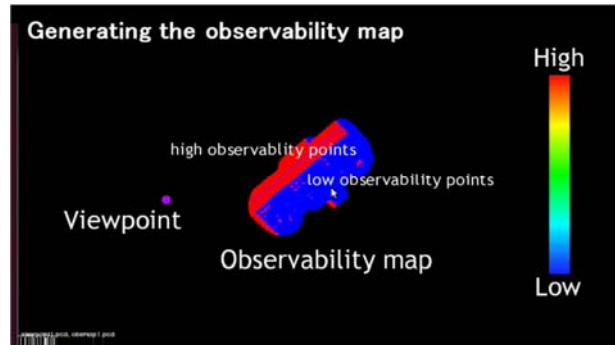
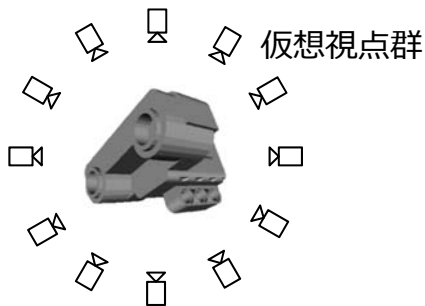


可観測性を考慮した物体認識

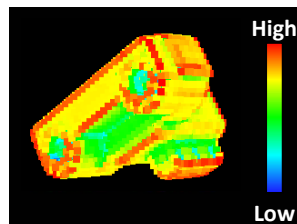
見えない点は使えない

可観測性を考慮した 3D特徴点選択

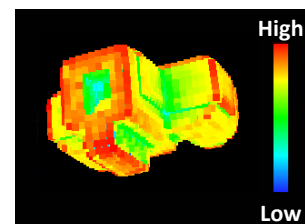
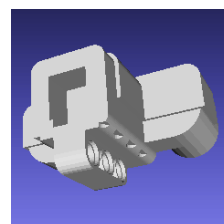
- 入力シーンにおける観測可能性の高い点を自動選択
 - 物体周りに複数の視点を設定し、各点の観測可能性をモニタリング
 - 観測可能性の高い点を特徴点とする。



可観測率マップの生成



可観測率マップ (視点 1)



可観測率マップ (視点 2)

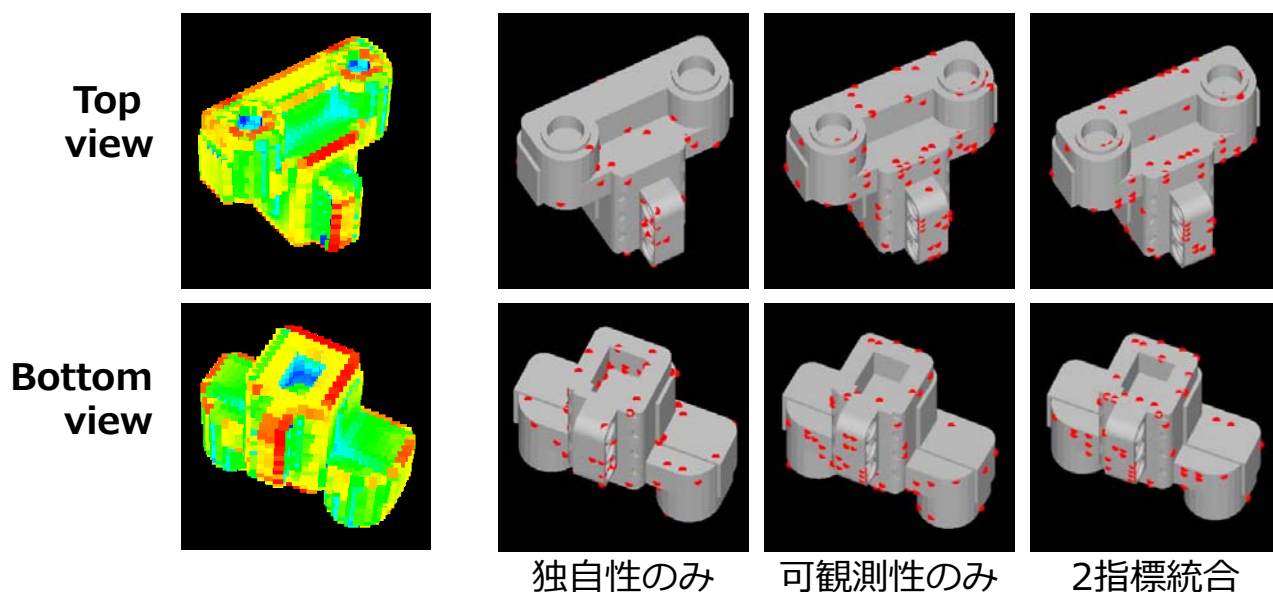
S. Akizuki, M. Hashimoto : "Position and Pose Recognition of Randomly Stacked Objects using Highly Observable 3D Vector Pairs", Proc. IECON, pp.5266-5271, Oct, 2014.

Hashimoto Laboratory, Chukyo University

マニピュレーション冬の学校 Dec. 16, 2018

31

選択された特徴点



さまざまな形状の物体に対する認識結果

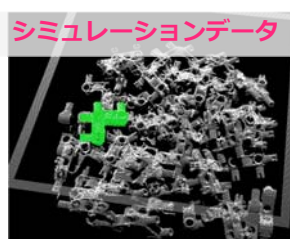
■ 4種類×約113シーンで実験

認識率 Pr [%]と処理時間 T [sec]の関係

		A	B	C	D
SHOT+Correspondence Grouping [Tombari,PSIVT2011]	Pr	51.3	77.9	33.6	20.3
	T	27.78	19.46	33.69	22.48
FPFH+RANSACベース照合 [Buch,ICRA2013]	Pr	54.9	11.5	73.5	31.0
	T	0.67	3.97	1.07	0.91
VPM(独自性のみ)	Pr	40.7	31.0	47.8	33.6
	T	0.27	1.41	0.39	0.72
VPM(可観測性のみ)	Pr	97.3	100.0	94.7	85.0
	T	0.47	1.76	1.76	1.99
VPM(独自性+可観測性)	Pr	99.1	95.6	92.0	85.8
	T	0.44	1.76	1.42	1.81



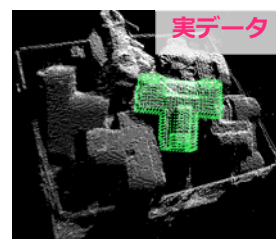
Aの認識結果
(平面形状で構成)



Bの認識結果
(細かい凹凸が多い)



Cの認識結果
(円筒形状)



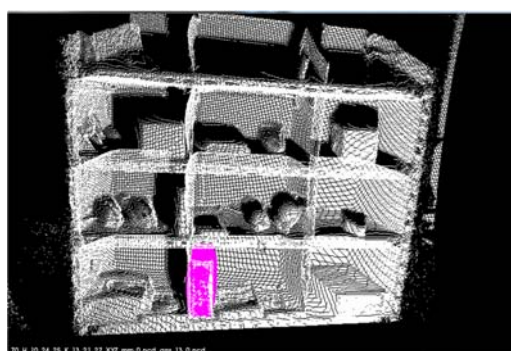
Dの認識結果
(実距離データ)

3次元局所特徴量を用いた物体認識の例 (ARC)

箱状物体



入力シーン外観

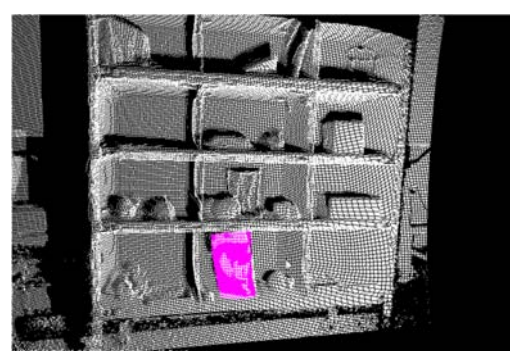


認識結果

形が変形しやすい袋状物体



入力シーン外観



認識結果

3D-CG技術の物体認識への応用

シミュレータを利用した「よい特徴量」の選別

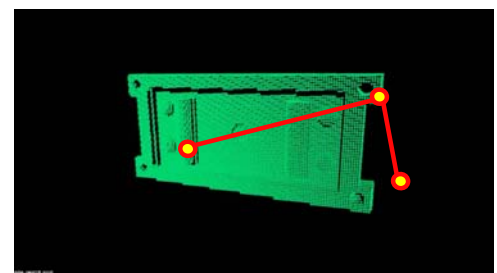
基本アイデア

1. 3点を用いた認識

膨大な3次元データから、3点のみを選択して認識に使用

→ 処理の高速化

3点の選び方が重要



3D-CADデータ（3D点の集合）

2. 3点の選択方法

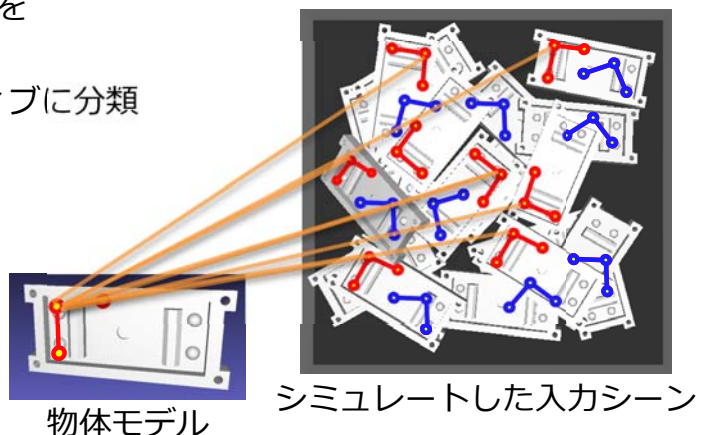
認識性能が高い3点を選択

1. 3D-CGを用いて入力シーンを精密にシミュレート
2. 3点をポジティブとネガティブに分類
3. 3点の性能を数値化

→ 確実性の高い認識

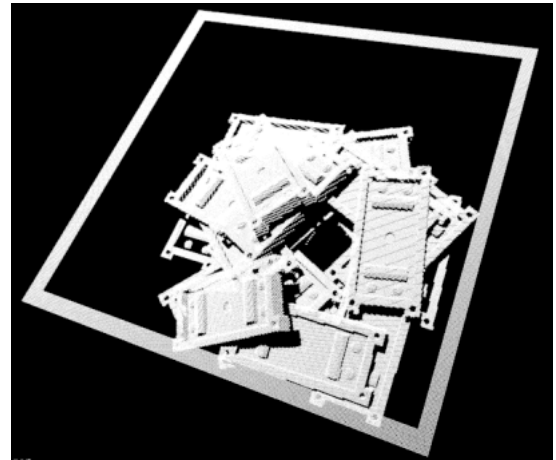
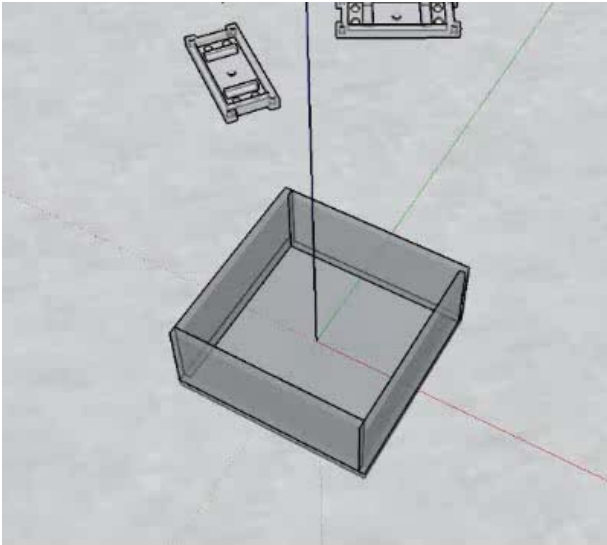
ポジティブ

ネガティブ



アイデア：入力シーンを予測して特徴点の良否判定

シミュレータを用いてリアリティの高いばら積みシーンを自動生成
物理エンジンには Sketchy Physics を使用



ばら積みシーン
(センサ入力を模擬した3次元データ)

さまざまなパターンのばら積み状態をシミュレート

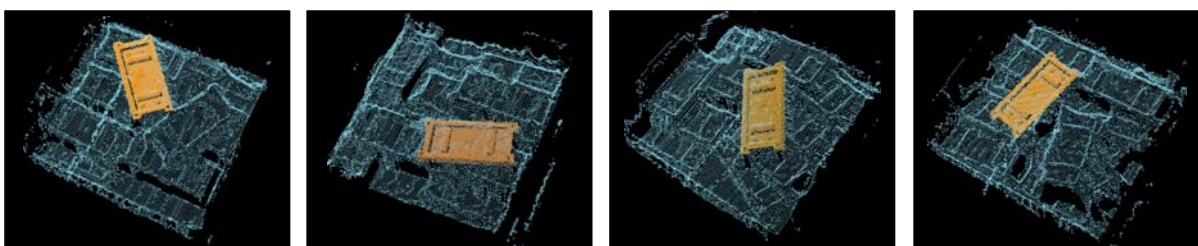
認識性能の評価実験②（本実験）

実際のばら積みシーン140枚に対する認識実験

ライバル手法との性能比較

	認識成功率	処理時間
Spin Image法	71.4%	57.99秒
Correspondence Grouping法	62.1%	16.66秒
曲率法 + VPM法	91.4%	5.28秒
Vector Pair Matching(VPM)法	75.7%	0.61秒
提案手法	94.3%	0.36秒

CPU : Intel®CORE™i7-3770 システムメモリ : 16GB



提案手法はロボットの動作サイクル時間以内に**高信頼な認識を実現**

点群密度変化への対応（RPD特徴量）

センサからの距離に依存しない特徴量

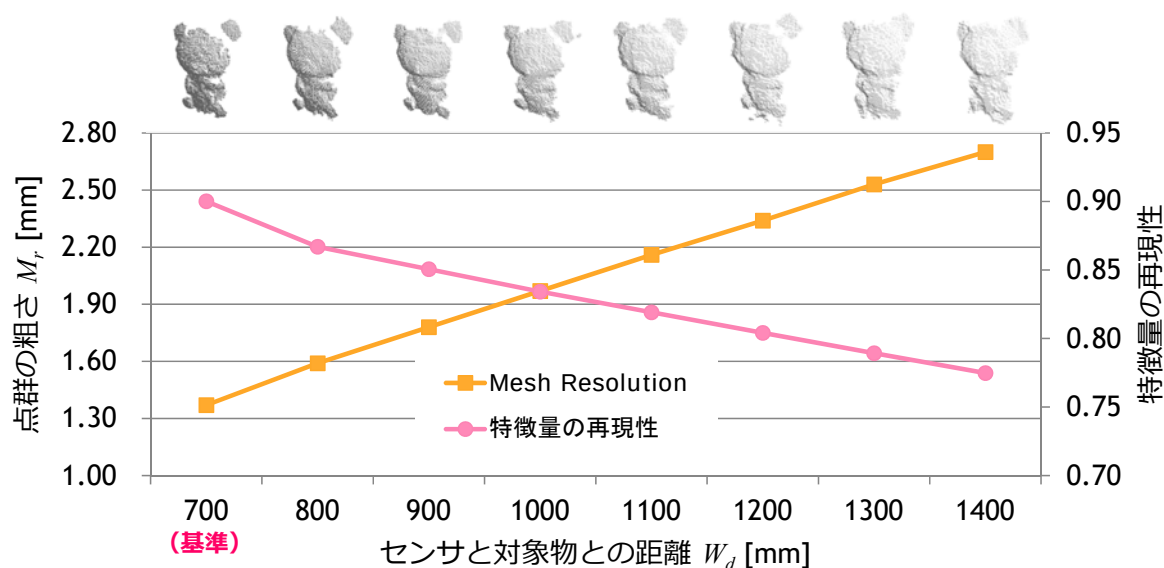
点群密度の差に起因する特徴点の再現性の低下

■ 予備実験

計測距離に起因する点群の密度変化 v.s. 特徴量の再現性

※再現性：2つの特徴ベクトル間の内積値

※ $W_d=700\text{mm}$ 時をモデル（基準）として、遠方から撮影したデータと照合（使用特徴量：SHOT）

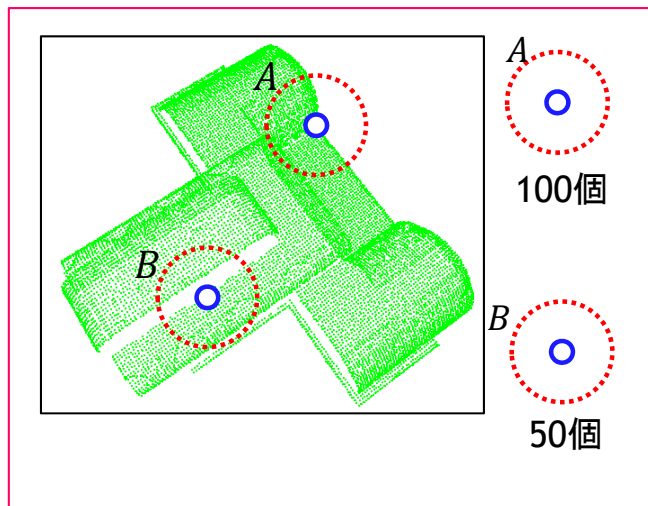


計測距離の変化により、特徴量の再現性が低下する。

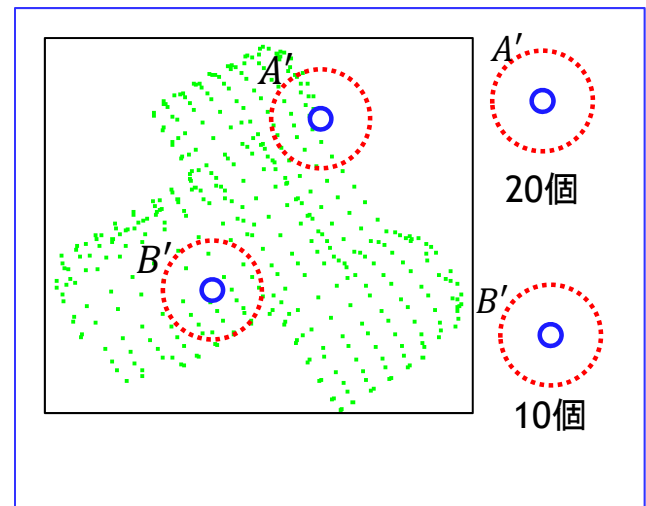
このほか、ノイズの影響もある（ $\therefore W_d=700\text{mm}$ でも1.0にならない）

基本アイデア

- センサからの距離等によって点群の密度は変化するが・・・
局所領域間の相対点数比は不変
 - 複数の局所領域内に存在する点数の相対関係を利用



密な点群



疎な点群

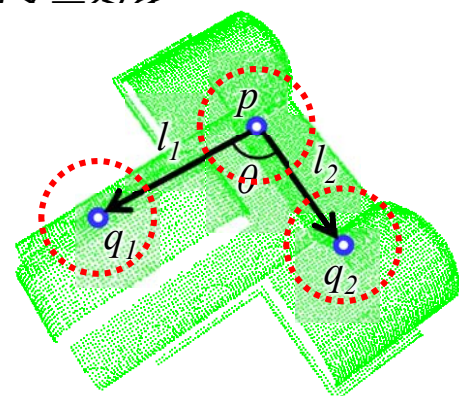
RPD (Relative Point Density) 特徴量

局所領域間の点数比から構成される特徴量

- 3つの3次元点を基準として特徴量を記述
 - 3つの3次元点は位置姿勢変換パラメータを算出するための最小単位
 - Vector Pair

- 特徴量の記述方法
 $C(x)$: 点 x 周辺に存在する点の個数

$$\mathbf{f} = \left(\frac{C(p)}{S}, \frac{C(q_1)}{S}, \frac{C(q_2)}{S} \right)$$
$$S = C(p) + C(q_1) + C(q_2)$$



ベクトルペア幾何パラメータ

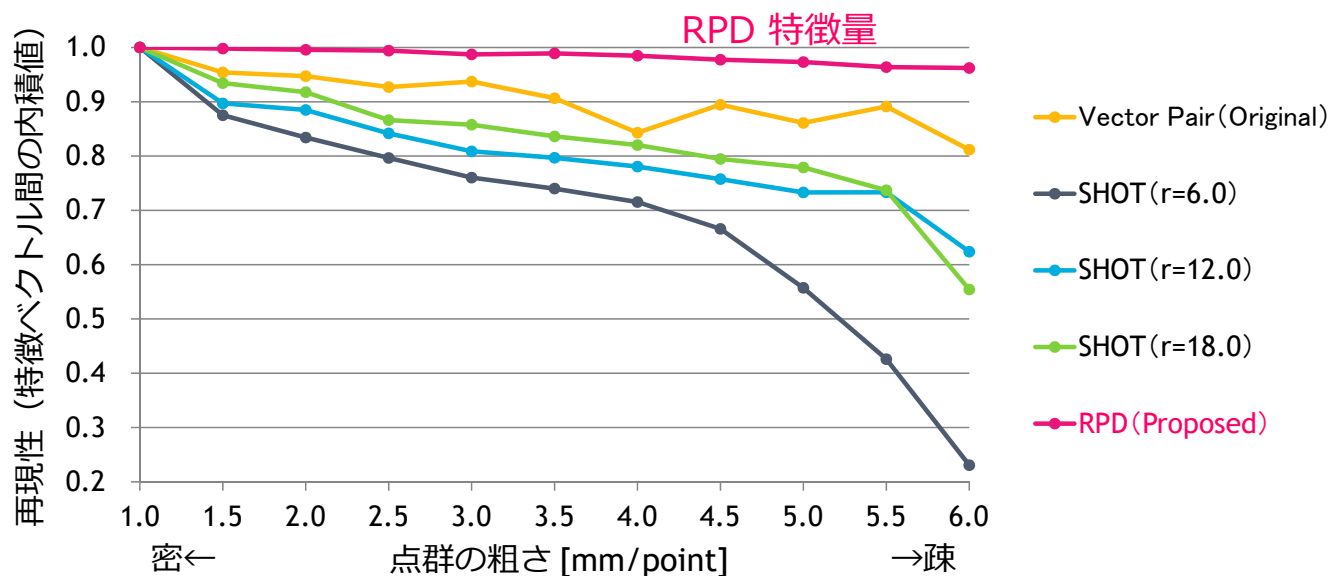
l_1 : p - q_1 間の距離

l_2 : p - q_2 間の距離

θ : p - q_1 と p - q_2 の間の角

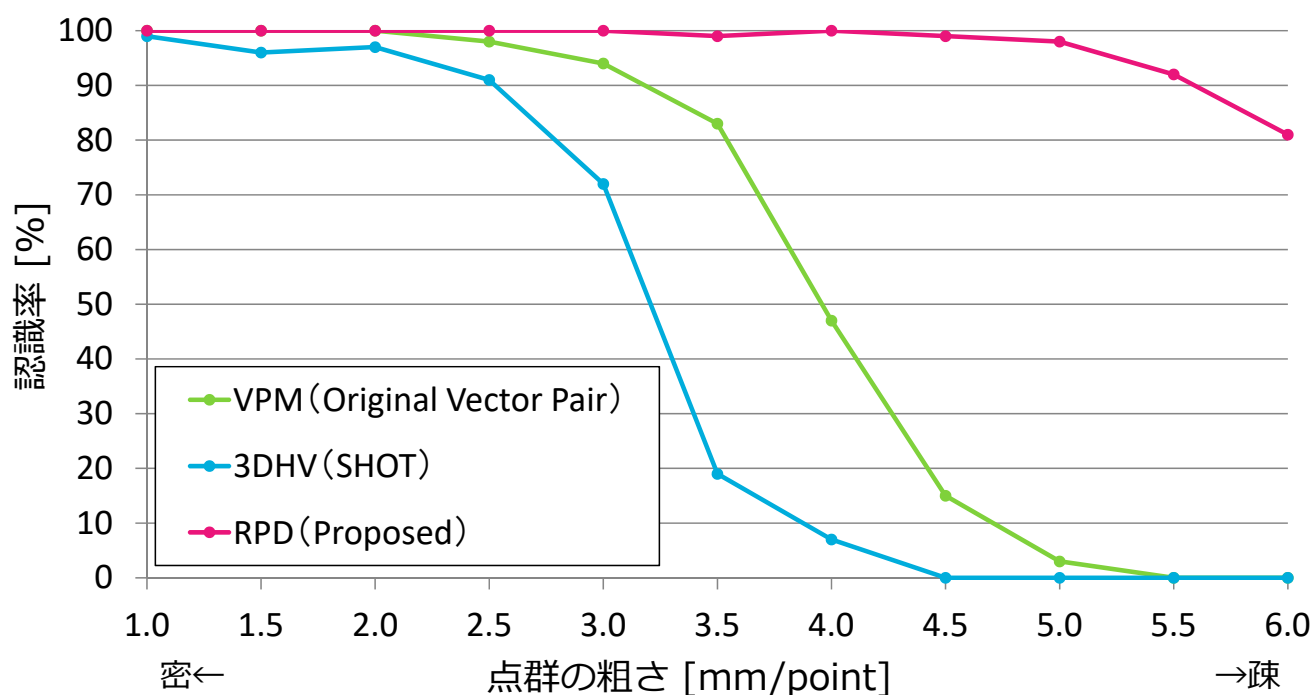
RPD 特徴量の性能①（再現性）

比較特徴量：①オリジナルの Vector Pair 特徴量
②SHOT 特徴量



RPD 特徴量は粗い点群に対しても再現性が高い

RPD 特徴量の性能②（認識率）



RPD 特徴量は点群が粗くても認識率が高い

SHORT: 点群の空間占有率とマルチスケールシェルを用いた高速 3D 物体認識

Shell Histograms and Occupancy from Radial Transform

キーポイント検出と特徴量記述の高速化

マルチスケールシェルを用いた特徴量の記述

マルチスケールのシェル領域に存在する点群の分布を記述

1. 主方向ベクトル $\mathbf{d}_k$ の算出 → 姿勢不変な特徴量記述

キーポイント $\mathbf{p}_k$ を原点とした
重心ベクトルを主方向とする

$$\mathbf{g}_k = \frac{1}{L} \sum_{i=1}^L o_i (\mathbf{p}_i - \mathbf{p}_k) \quad \mathbf{p}_i : \text{領域内の点}$$

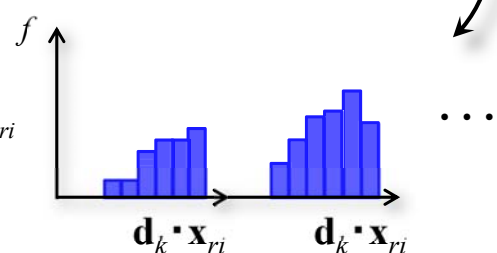
$$\mathbf{d}_k = \frac{\mathbf{g}_k}{\|\mathbf{g}_k\|} \quad o_i : \text{占有率} \quad L : \text{領域内の点数}$$



2. 特徴量（ヒストグラム）の算出

$\mathbf{d}_k$ と、領域内の点への方向ベクトル $\mathbf{x}_{ri}$
との内積ヒストグラムを算出

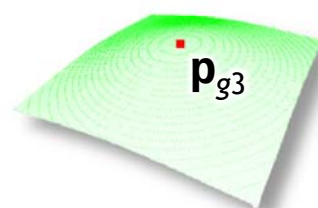
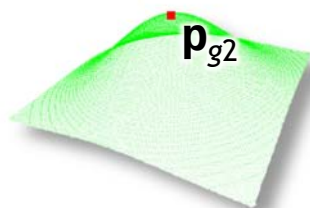
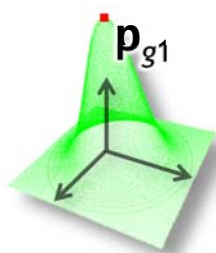
→ 点群の分布を高速に記述



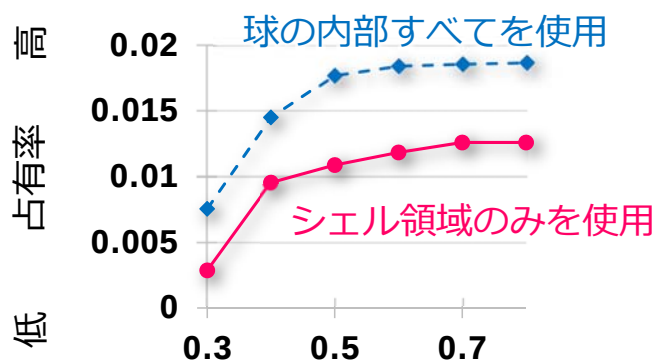
SHORT 特徴量

点群の空間占有率と形状の関係

σ が異なるガウス分布の点群を生成し, p_{gk} に対して占有率を計測



ガウス分布 ($\sigma = 0.3$) ガウス分布 ($\sigma = 0.5$) ガウス分布 ($\sigma = 0.7$)



小 (凸) ガウス分布の σ 大 (平面)

凸は占有率が低く,
平面は占有率が高いことを確認



占有率を形状評価値として,
キーポイント検出と
特徴量記述に利用

認識性能の評価実験

公開データセット [1] に対して物体認識

処理時間 T [msec] と認識率 Pr [%] の関係

キーポイント検出 + 特徴量記述	認識 アルゴリズム	データセット	処理時間 T [msec]	認識率 Pr [%]
ISS 法 [Zhong2009] + SHOT 法 [Salti2014]	3D Hough Voting 法 [Tombari2010]	A	840.2	98.9
		B	4146.9	85.7
SHORT法 (提案手法)		A	211.7	100.0
		B	460.2	91.8

データセット B に対する処理時間の比較

(CPU : Intel®CORE™i7-3770, メモリ : 16GB)

キーポイント検出

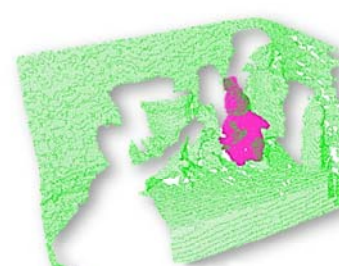
ISS 法 299.0[msec] **約 1/15** 提案手法 20.0[msec]

特徴量記述

SHOT 法 1571.7[msec] **約 1/8** 提案手法 182.6[msec]



データセット A
(模擬データ) の認識結果



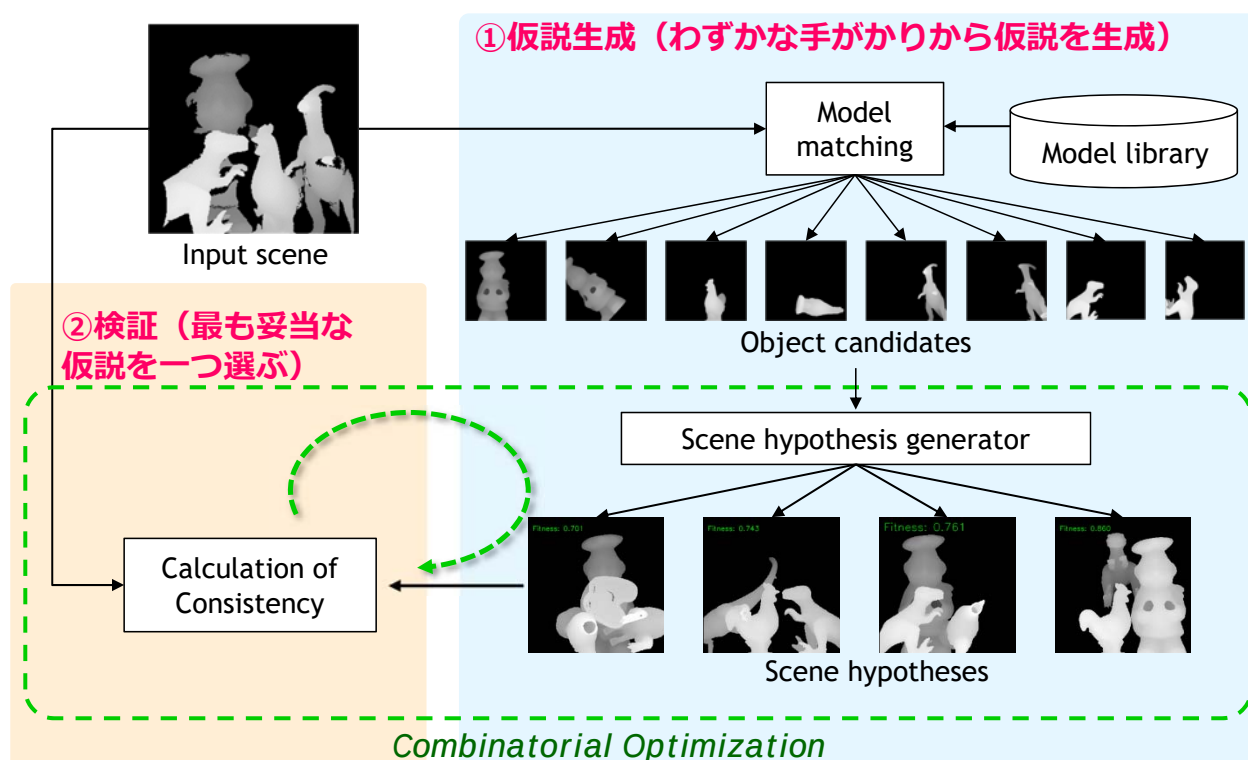
データセット B
(実データ) の認識結果

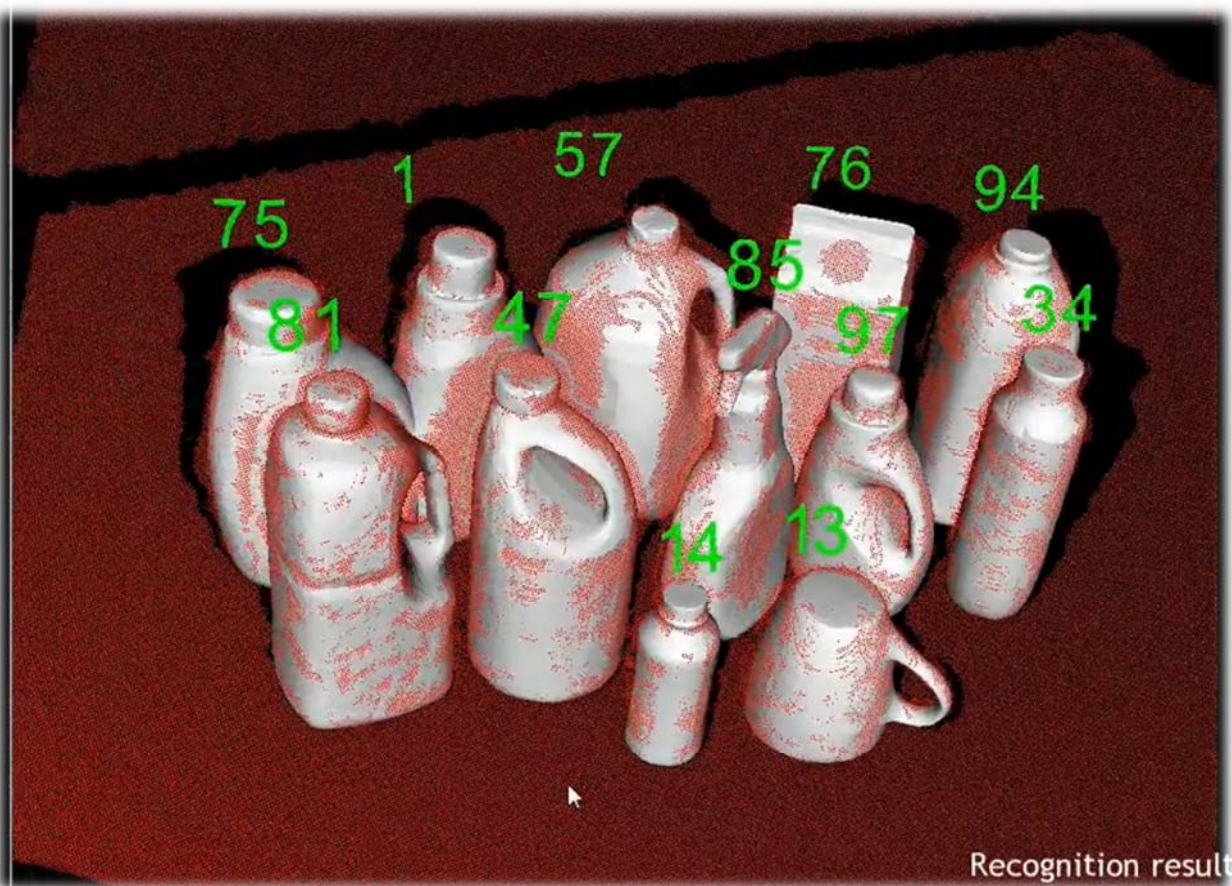
[1] <http://vision.deis.unibo.it/research/78-cvlab/80-shot>

複数物体の同時認識

複雑なシーンを理解する

アイデア：手がかりから仮説生成 → 大局的に検証





本日の話題

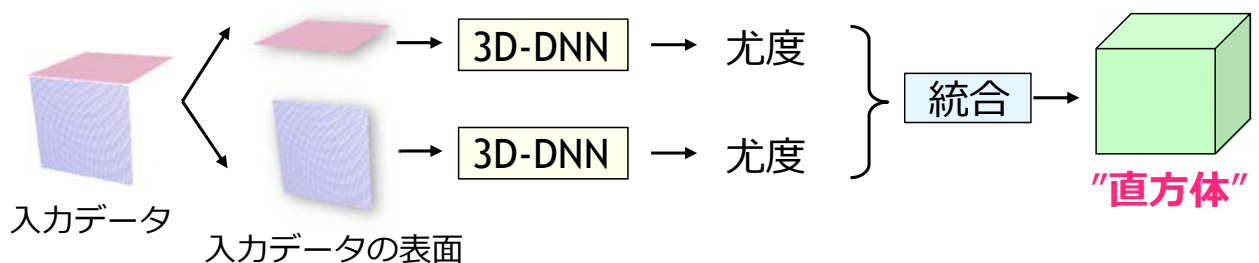
1. ロボットビジョンの課題
2. 現状：3次元センサ
3. 現状：特定物体ハンドリングのための物体認識
4. 現状：柔軟なロボット動作のための物体認識
5. ARCとWRSにみる実課題への対応
6. 展望：タスク生成のための物体認識
7. 残された課題

プリミティブ物体近似に基づく モデルレス把持パラメータ決定

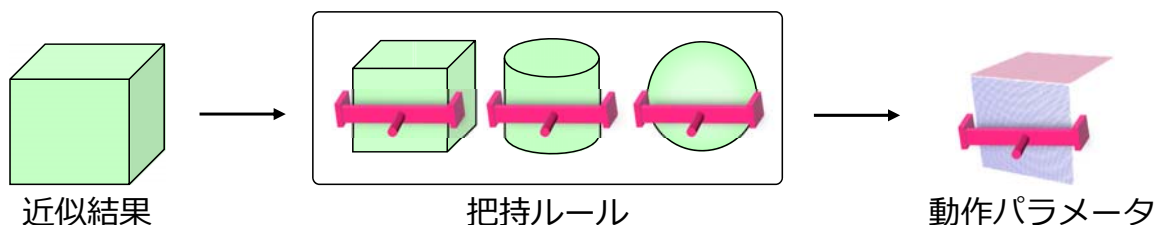
基本アイデア

物体の表面の形状をもとに物体をプリミティブ近似し、把持ルールを用いてロボットの動作パラメータを推定する。

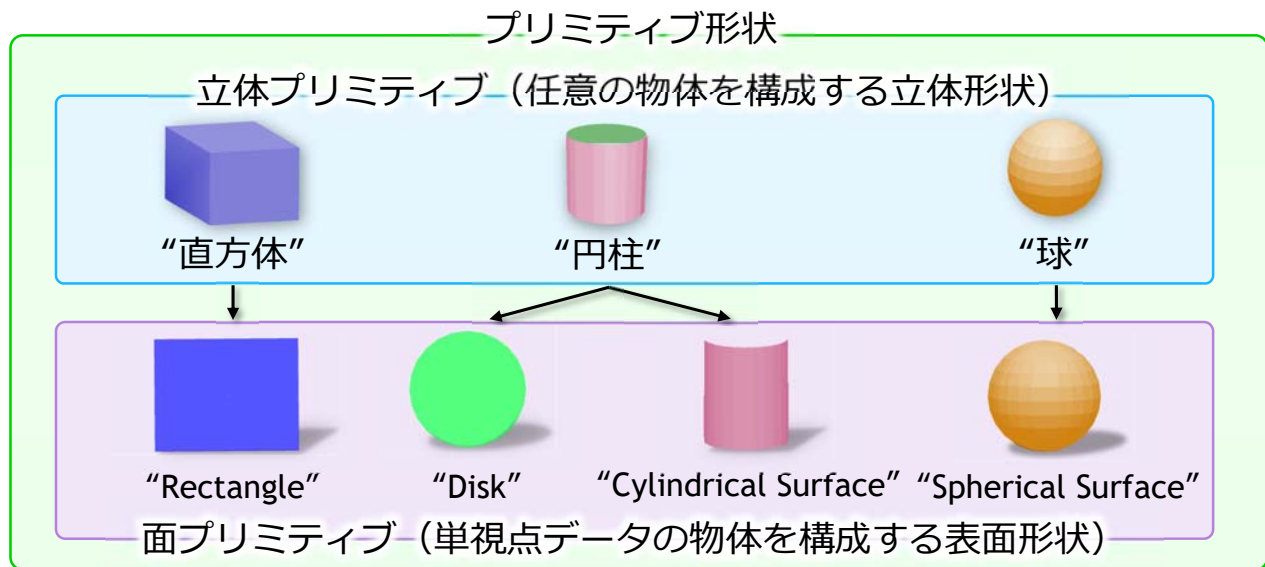
■ プリミティブ物体近似



■ 動作パラメータ推定



プリミティブ形状の種類

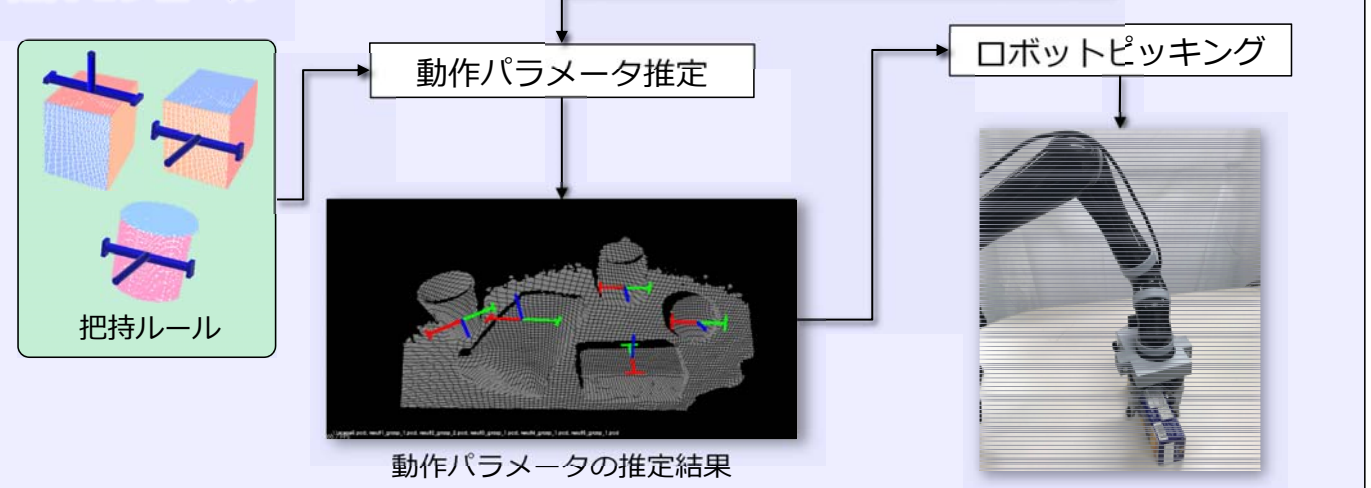


提案手法の流れ

近似モジュール



把持モジュール

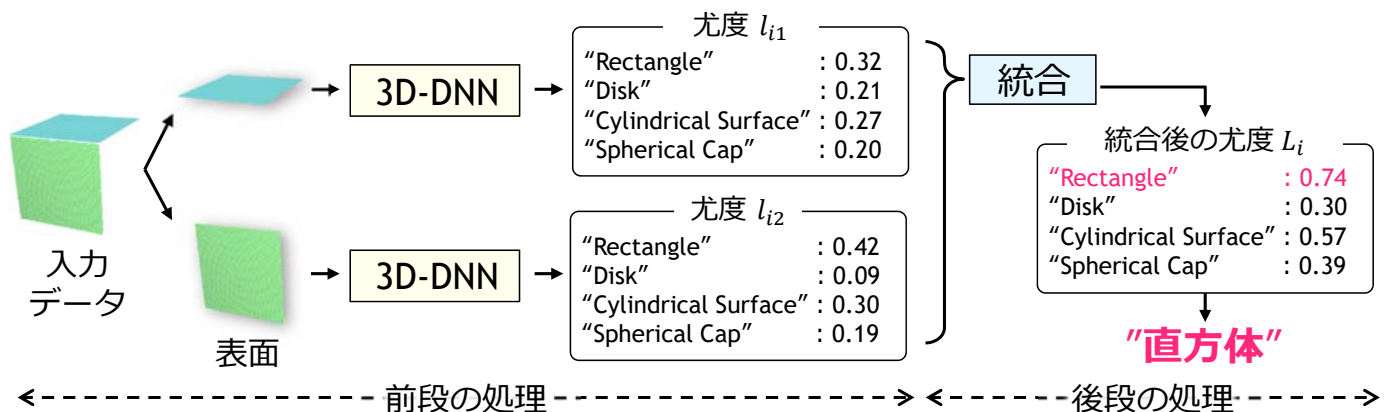


プリミティブ近似方法

物体の表面の形状をもとに 2 段階の処理で物体をプリミティブ近似

前段：3D-DNN を用いて，表面ごとの面プリミティブの尤度推定

後段：各尤度を統合することによって立体プリミティブで近似



尤度の統合式

$$L_i = \sum_{j=1}^N l_{ij}$$

L_i : 統合後の面プリミティブの尤度

l_{ij} : 各セグメントの面プリミティブの尤度

i : 面プリミティブ ID

j : セグメント ID

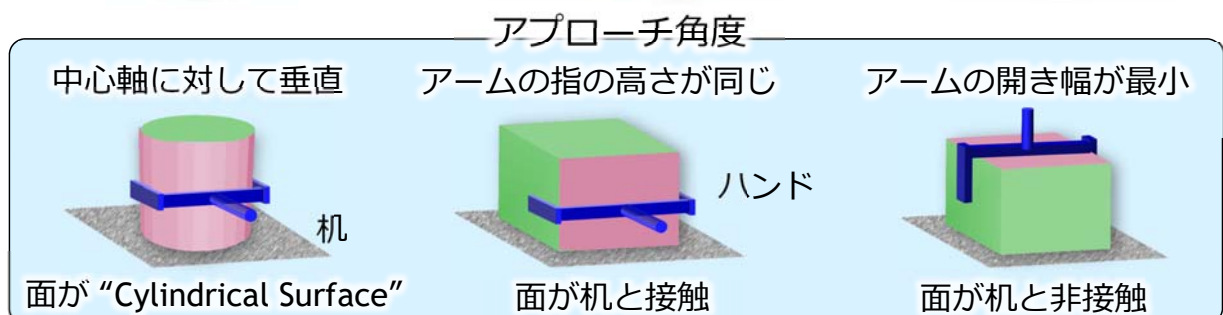
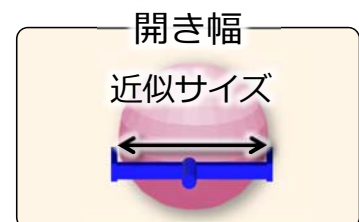
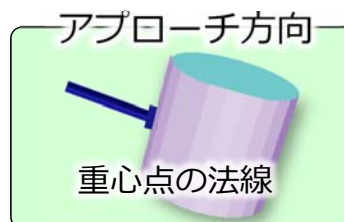
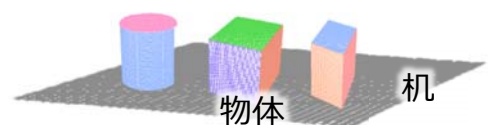
N : セグメント数

動作パラメータの推定

プリミティブ近似結果に対して把持ルールを適用することによりロボットの動作パラメータを推定する。

■ 把持ルール的前提条件

- 物体が平面上に置かれていること



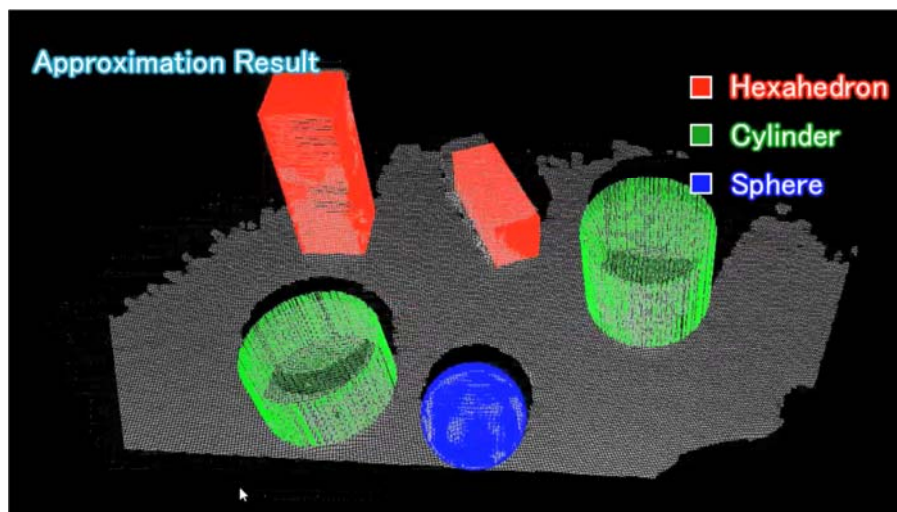
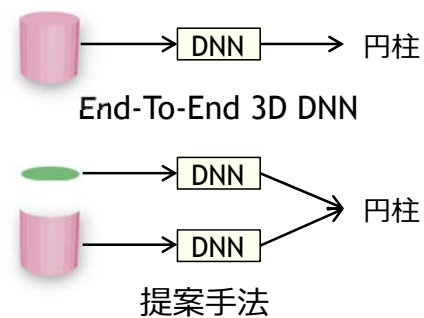
未知物体に対しても動作パラメータの推定が可能

実験① 物体近似性能評価

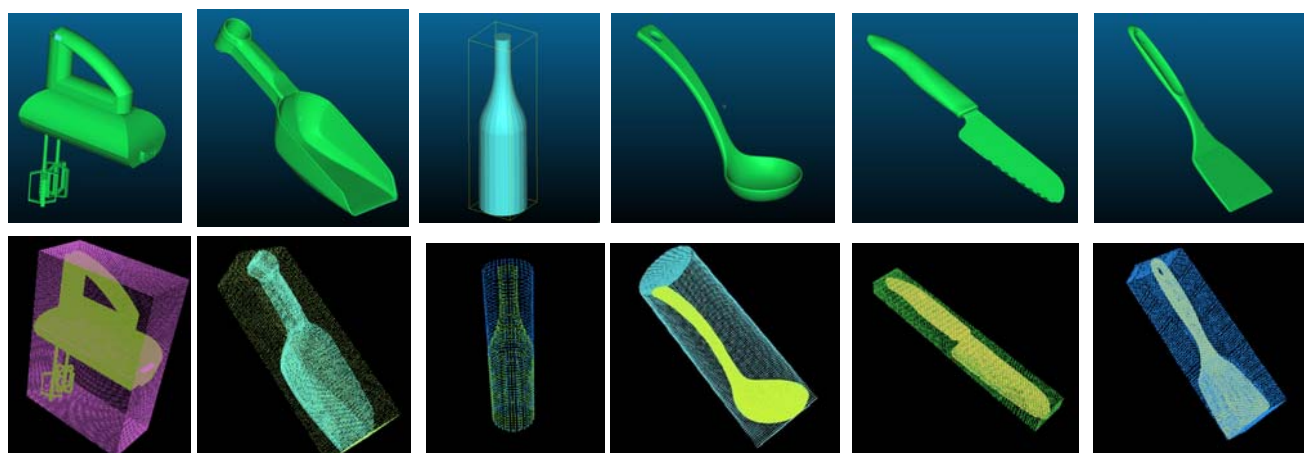
各手法のプリミティブ近似の成功率 [%]

手法 カテゴリ	曲率法	End-To-End 3D DNN		提案手法
		3D ShapeNets	PointNet++	
"直方体"	60.0	75.0	98.0	99.0
"円柱"	28.0	92.0	94.0	95.0
"球"	78.0	97.0	98.0	98.0
平均	55.3	87.7	96.7	97.3

* 提案手法では, PointNet++ を DNN として使用



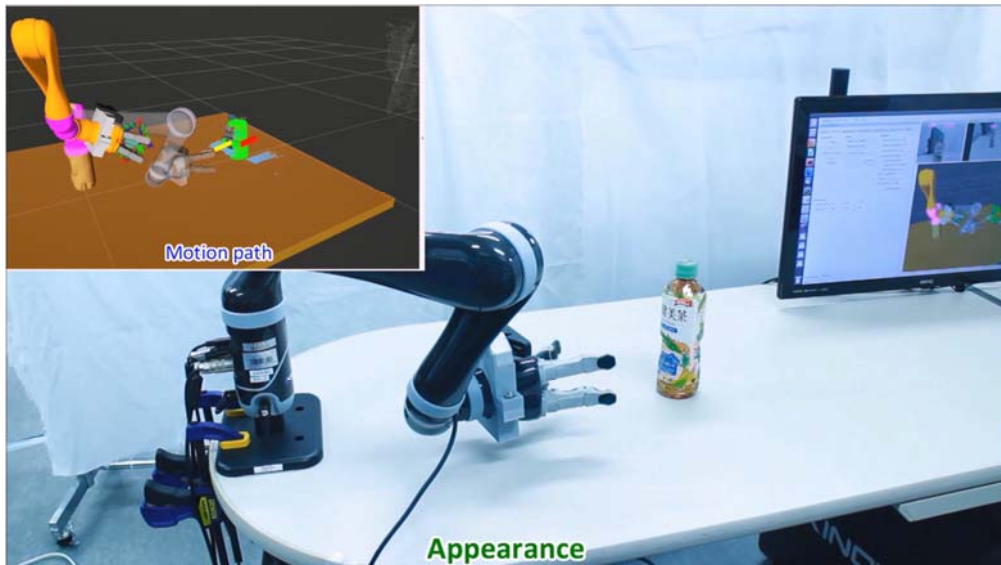
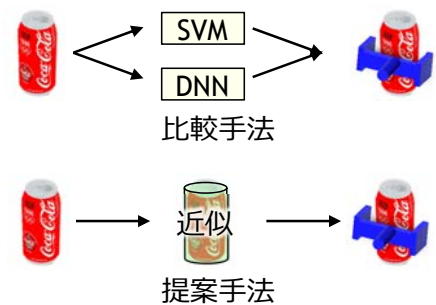
さまざまな対象物に対するプリミティブ近似結果



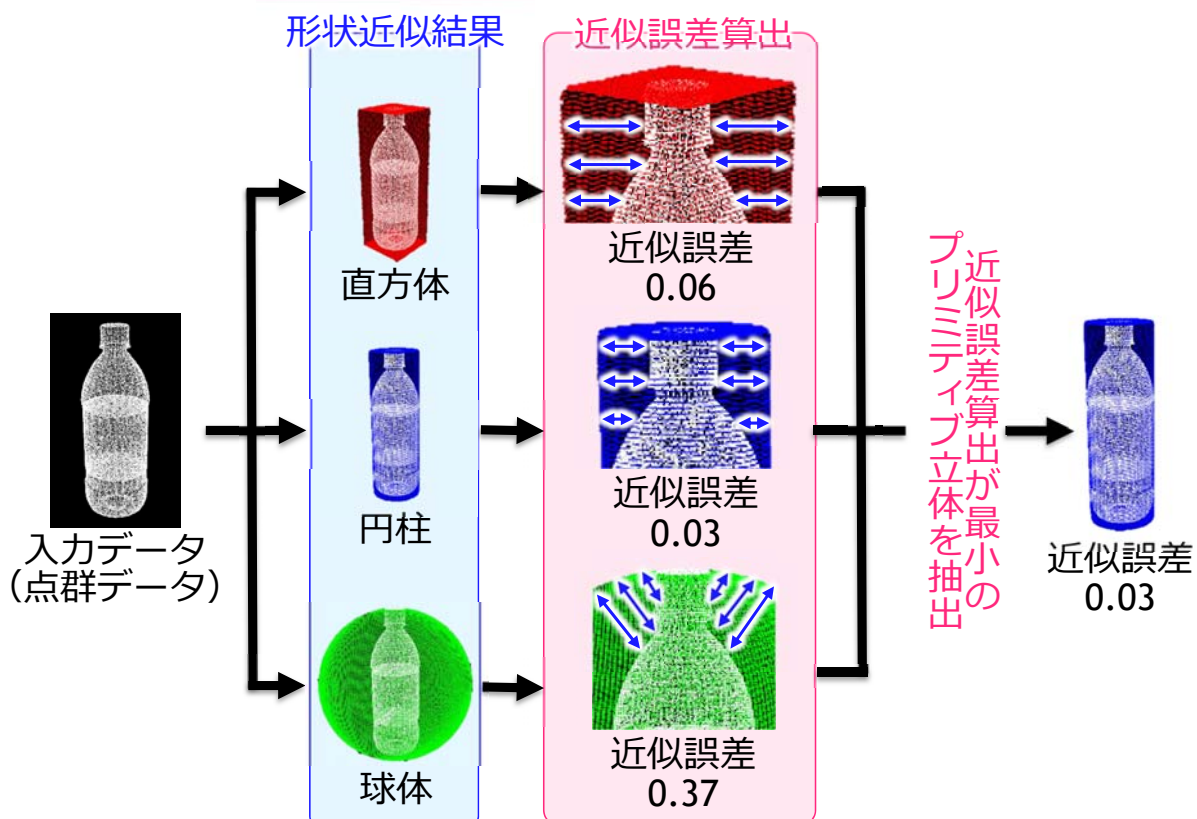
実験② ロボット把持性能評価

各手法の把持成功率 [%]

手法 カテゴリ	Agile grasp (SVM)	GPD (DNN)	提案手法
“直方体”	63.3	73.3	86.7
“円柱”	76.7	76.7	86.7
“球”	66.7	70.0	90.0
平均	68.9	73.3	87.8



実験③ プリミティブ 3 種の有効性（実験的調査）



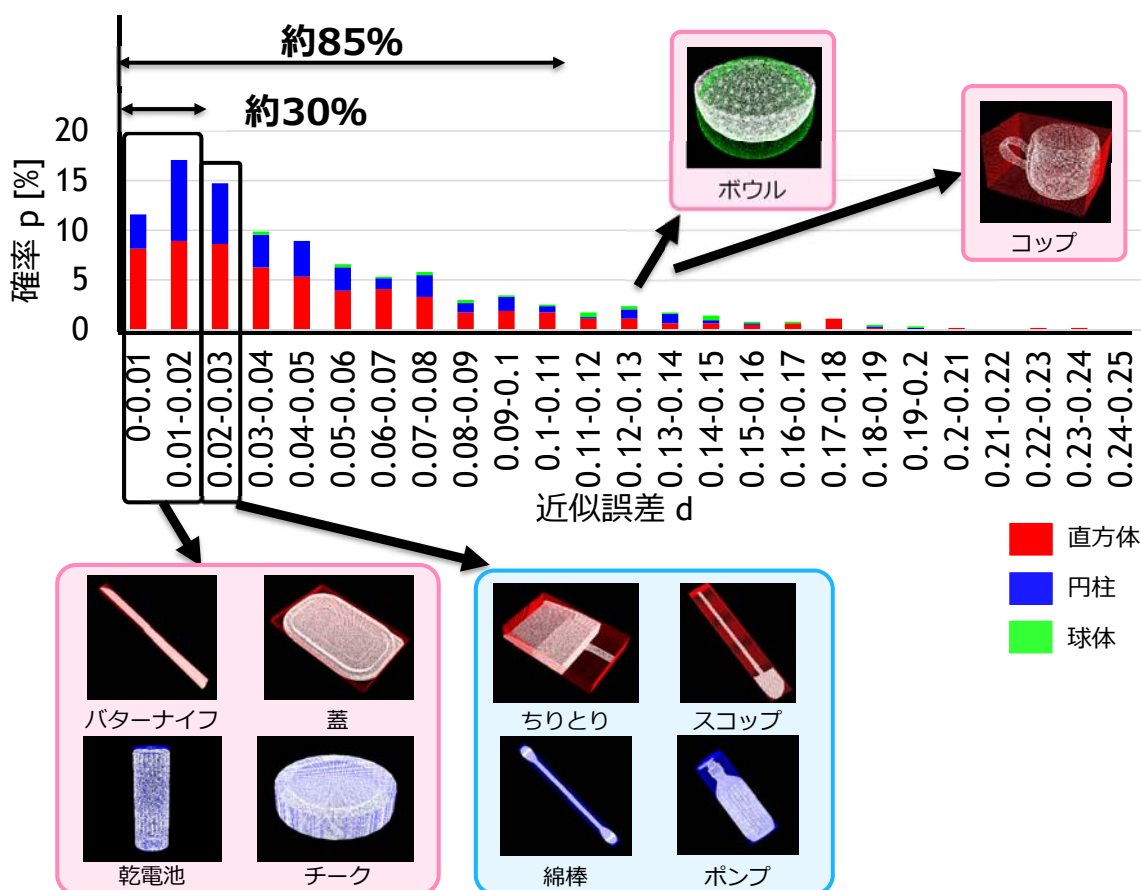
↔ は入力データからプリミティブ立体までの最近傍距離を表す。
近似誤差は複数の↔の平均である。

実験に使用した日用品データ集

- JICFS/IFDB（商品情報データベース）が定義する5カテゴリ
- Web 上の ShapeNet, ObjectNet, ModelNet, 3Dwarehouse, Thingiverse と自身で作成した日用品モデル 638 個を用いた.



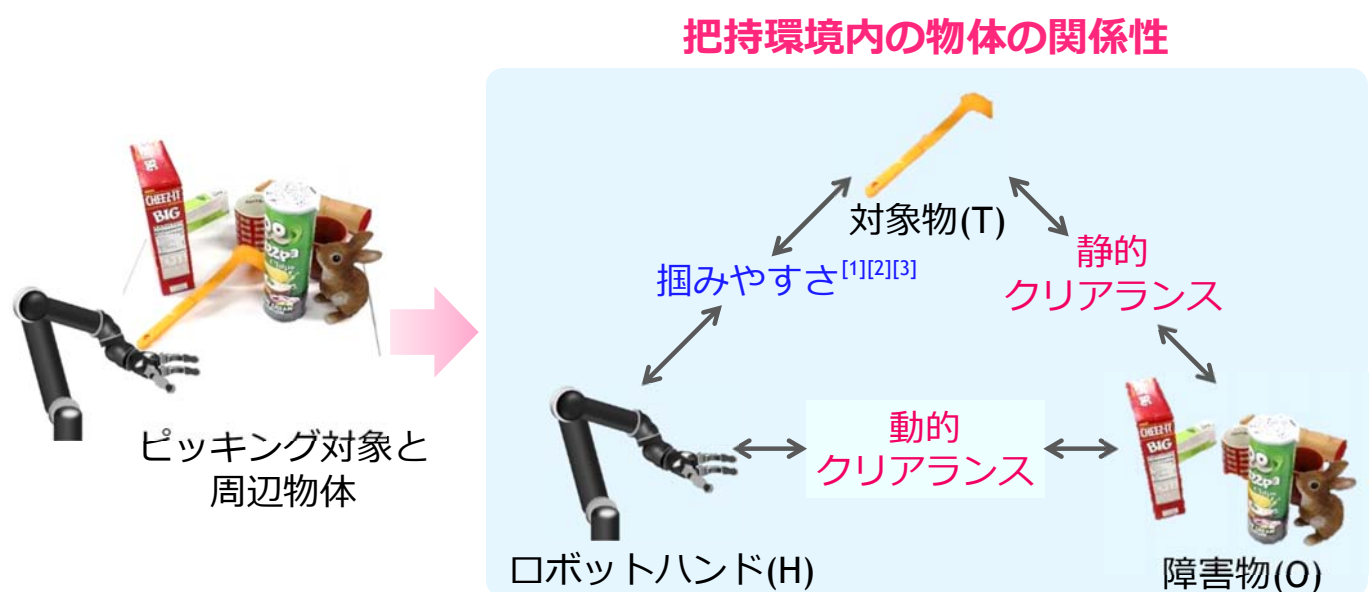
実験結果（速報）



周辺環境を考慮したロボット動作パラメータ決定

基本アイデア

環境内の各物体の相対的な余裕度をもとに動作生成する



把持余裕度推定モデルによる把持パラメータ推定

把持のしやすさを定量的に算出するために、3つの指標を統合した確率モデルを構築する。

把持余裕度推定モデル

$$P_s(\mathbf{p}) = P_{H-O}(\mathbf{p}) \{ \alpha P_{H-T}(\mathbf{p}) + \beta P_{T-O}(\mathbf{p}) \}$$

把持余裕度 ハンドと障害物の動的クリアランス 対象物の掴みやすさ 対象物と障害物の静的クリアランス

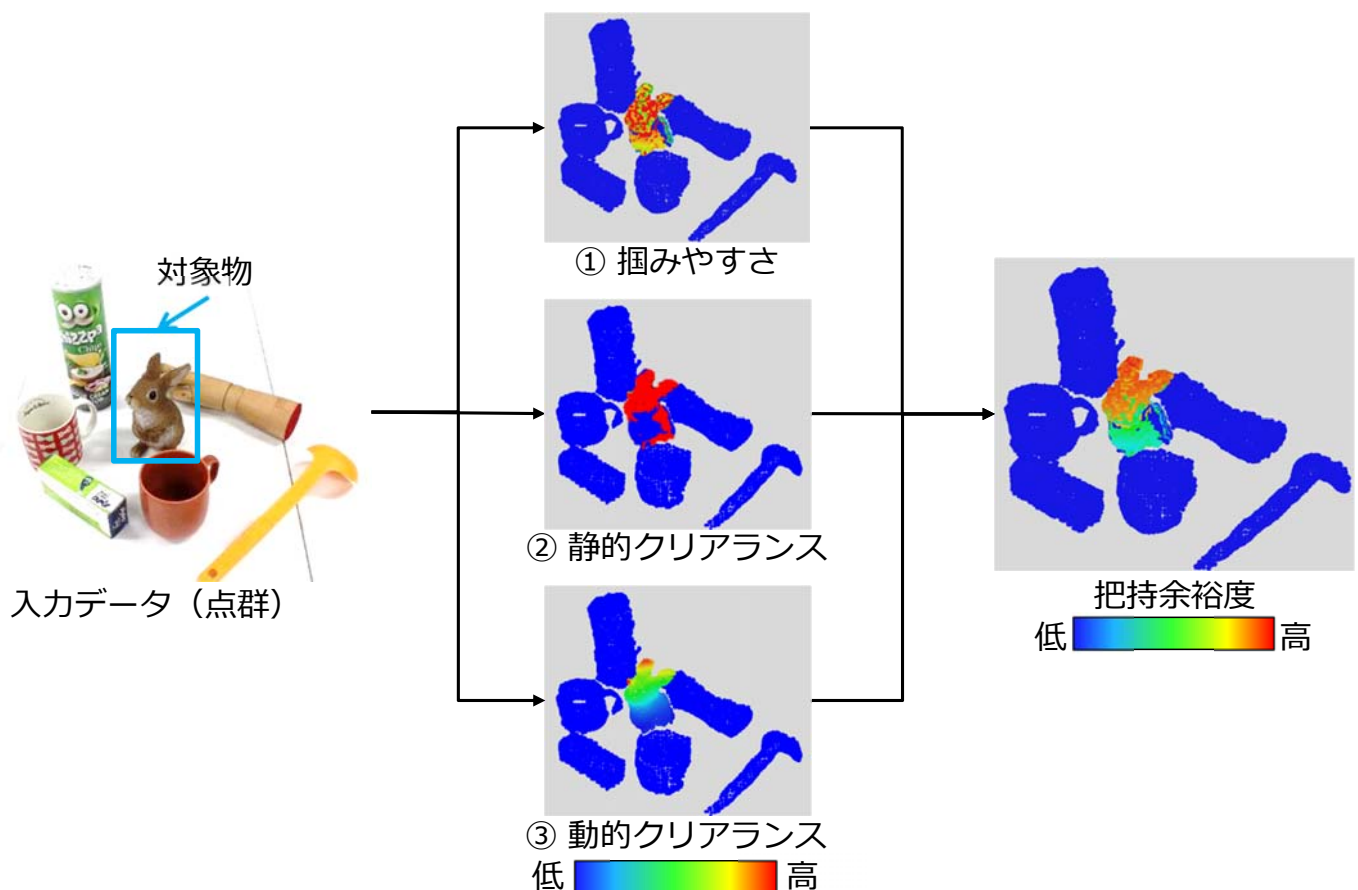
※ $\mathbf{p}$: 把持パラメータ (把持位置, 姿勢, 開口幅)

把持余裕度推定モデルを用いた $P_s(\mathbf{p})$ の最大化による
最適な把持パラメータ $\tilde{\mathbf{p}}$ の推定

$$\tilde{\mathbf{p}} = \underset{\mathbf{p} \in \mathcal{S}}{\operatorname{argmax}} P_s(\mathbf{p})$$

$\mathcal{S}$: $\mathbf{p}$ の候補集合

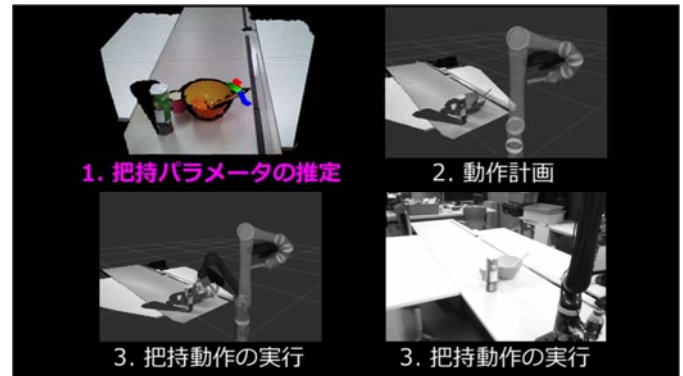
予備実験 – 各相互関係の指標の可視化 –



把持パラメータの推定実験の結果

15 シーン中の 47 個の対象物について把持パラメータを推定し，その非干渉率，把持成功率，処理時間を比較した．

手法	非干渉率[%]	把持成功率[%]	処理時間[s]
Agile_Grasp [Pas 2015]	55	48	5.46
Grasp-Detector [Pint 2017]	64	45	15.0
GPD [Gualtieri 2017]	73	64	3.20
提案手法	91	82	0.42



■ ハンド右指 ■ ハンド左指 ■ 把持姿勢

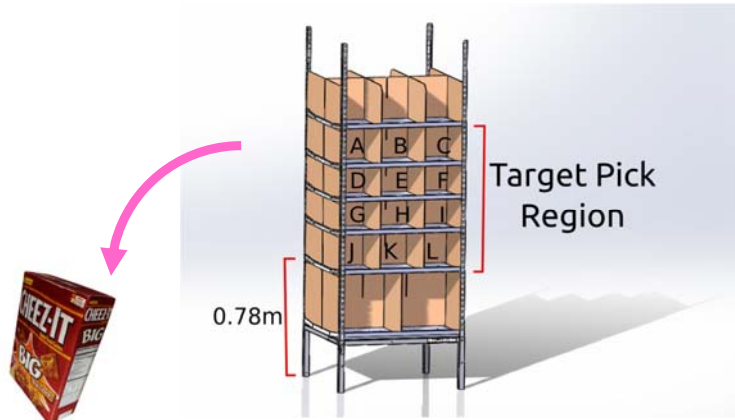
本日の話題

1. ロボットビジョンの課題
2. 現状：3次元センサ
3. 現状：特定物体ハンドリングのための物体認識
4. 現状：柔軟なロボット動作のための物体認識
5. ARCとWRSにみる実課題への対応
6. 展望：タスク生成のための物体認識
7. 残された課題

Amazon チャレンジへの挑戦履歴

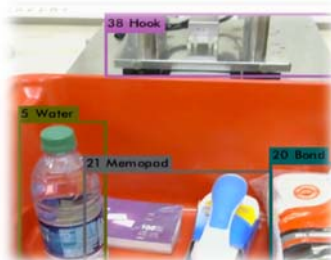
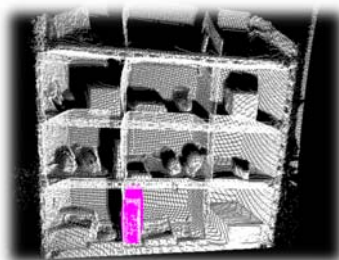
2015年	第1回大会（シアトル）	世界6位
2016年	第2回大会（ライプツィヒ）	世界8位
2017年	第3回大会（名古屋）	世界3位

- 制限時間内に正確にアイテムをピックアップする性能を競う.
- 課題は年々難しくなり, 毎回, 最先端技術の投入が不可欠.

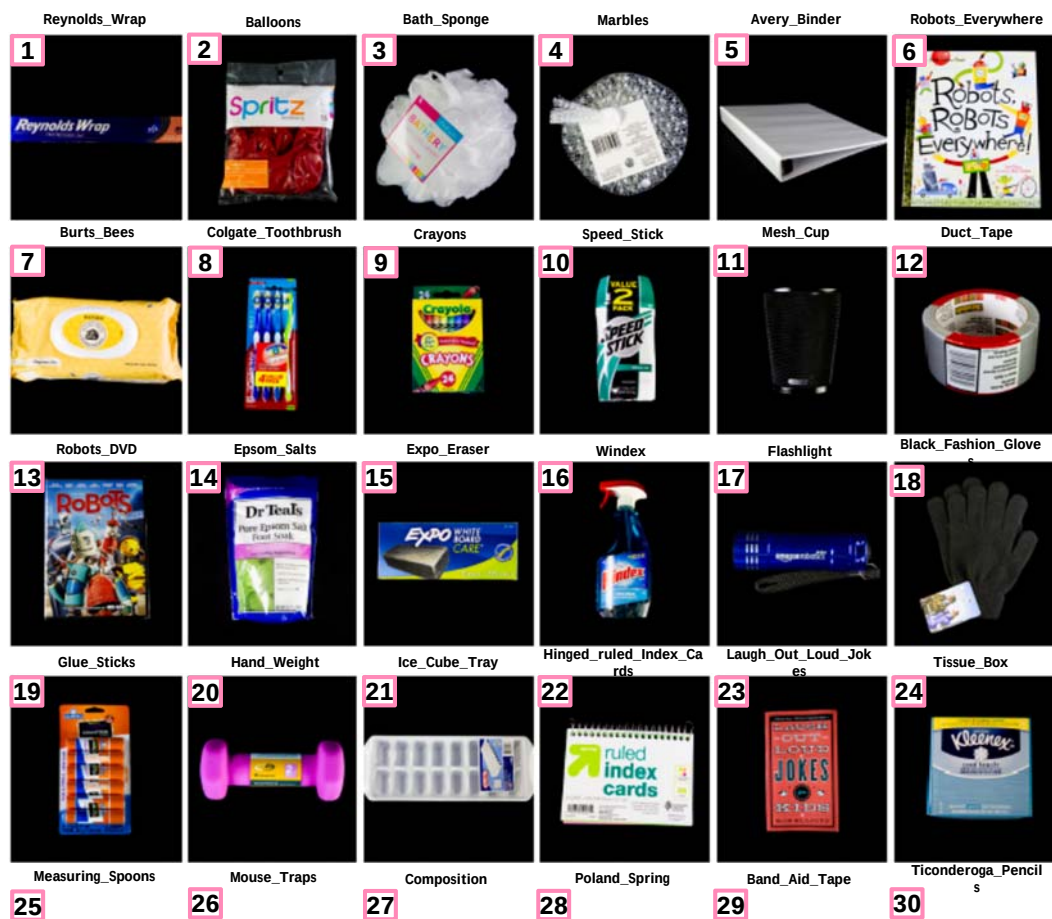


Amazonチャレンジにおけるビジョンの基本課題

どこに, なにがある? → どう掴む?



アイテムリスト（事前配布）



アイテムリスト（事前配布）



アイテムリスト（認識・把持が難しい事例）



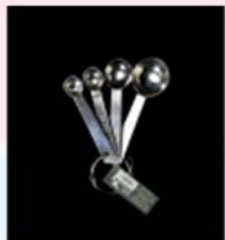
特徴（テクスチャ）が少ない



3D センサで撮影できない



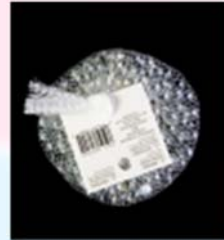
**認識が難しい
アイテム**



小さい, 3D センサで撮影できない



吸着が困難, 3D センサで撮影できない



**把持が難しい
アイテム**



吸着が困難



小さい

アイテムリスト（当日配布物の例）



白い手袋
非剛体のため
多様な見え方が生じる



子供用ストロー
全体的に細く
吸着が困難



ケーキ皿
金属のためサチりやすく
3D センサでも撮影が困難



ラチェット式荷締めベルト
重く, 隙間も多いため
吸着が困難



水槽用砂利
中身が動くため多様な見え方が生じる

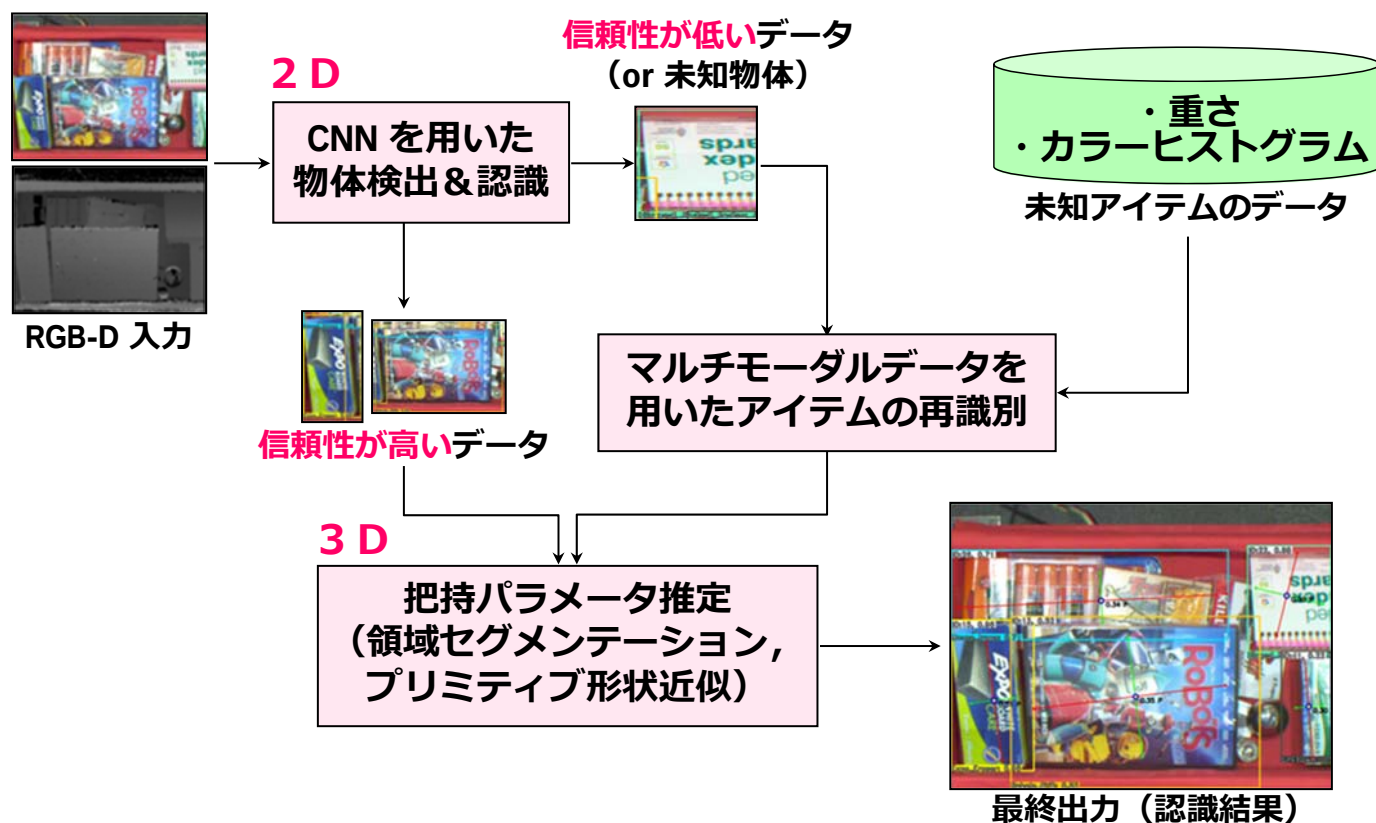


プラスチック製グラス
3Dデータ取得が困難



ブラシ
隙間が多く吸着が困難

物体認識アルゴリズム



当日のようす



サマリー（動画）

（2分21秒）



WRS = World Robot Summit 経産省（NEDO）主催

FCSC = Future Convenience Store Challenge（コンビニロボット）

- 2020年に本大会（2017, 2018, 2019 プレ大会あり）
- オムロン+中京大+中部大 合同チーム
- 戦績： 2017年優勝，2018年2位+学会賞



コンビニを模した競技会場
（セブンイレブン様が協力）



開発したロボット（2017）

サンドイッチの姿勢推定



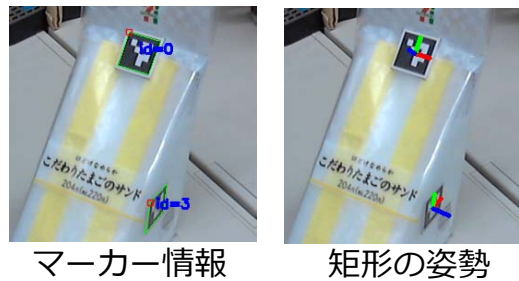
① 画像内の矩形領域を検出

- 輪郭を利用した高速パターン検出
- 照明変動や角度変動にも頑健

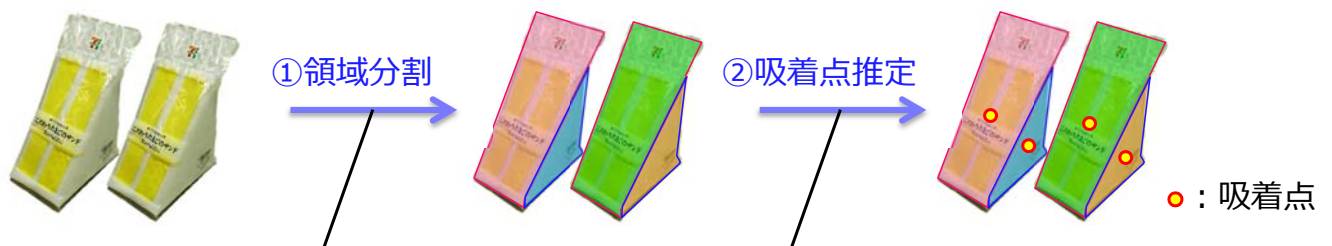
② マーカから 3 次元姿勢を計算

- マーカ情報読み取り
- 3 次元姿勢を計算

ARマーカ-の認識結果例



ロボット吸着点の自動決定



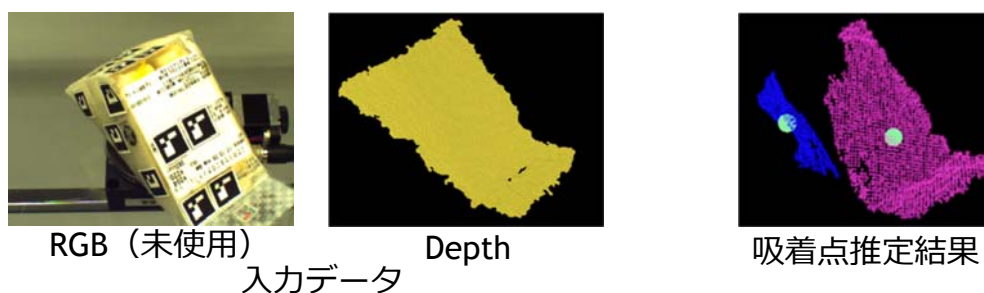
① 面領域を分割する

- 3次元センサ情報を利用して、表面形状を計測。
- 表面の滑らかさ度合い等を利用して分割。

② 領域分割された面ごとに吸着点を推定する

- 面ごとに計算した確率的な安定性指標が最大になる点 (= 重心点) を推定。

吸着点推定の結果例



お知らせ

マーカーレスのサンドイッチ姿勢認識手法

- 2 種のアプローチを開発中
 - 1. 3D面情報に着目した仮説検証アプローチ
 - 2. サイズ可変モデルを利用したアプローチ
- 動的画像処理実利用化ワークショップ（DIA2019）で発表予定

DIA2019

開催日 3月7日(水)～8日(木)

@北九州国際会議場

WRS-FCSCトライアル大会のようす（2017年12月）

WRS FUTURE CONVIENCE STORE CHALLENGE 2017
ROC2 (OMRON, CHUKYO UNIV., CHUBU UNIV.)

STOCKING AND DISPOSING TASK

1. ロボットビジョンの課題
2. 現状：3次元センサ
3. 現状：特定物体ハンドリングのための物体認識
4. 現状：柔軟なロボット動作のための物体認識
5. ARCとWRSにみる実課題への対応
6. 展望：タスク生成のための物体認識
7. 残された課題

対象物の“機能”の認識

「コップにジュースを入れて持ってくる」

①ボトルを持つ



②ジュースを注ぐ



③コップを持つ



ボトルを見つける

持つべき位置を認識
持ち方を生成

ボトルを把持

コップを見つける

注ぐべき位置を認識
注ぎ方を生成

ボトルを傾ける

コップを見つける

持つべき位置を認識
持ち方を生成

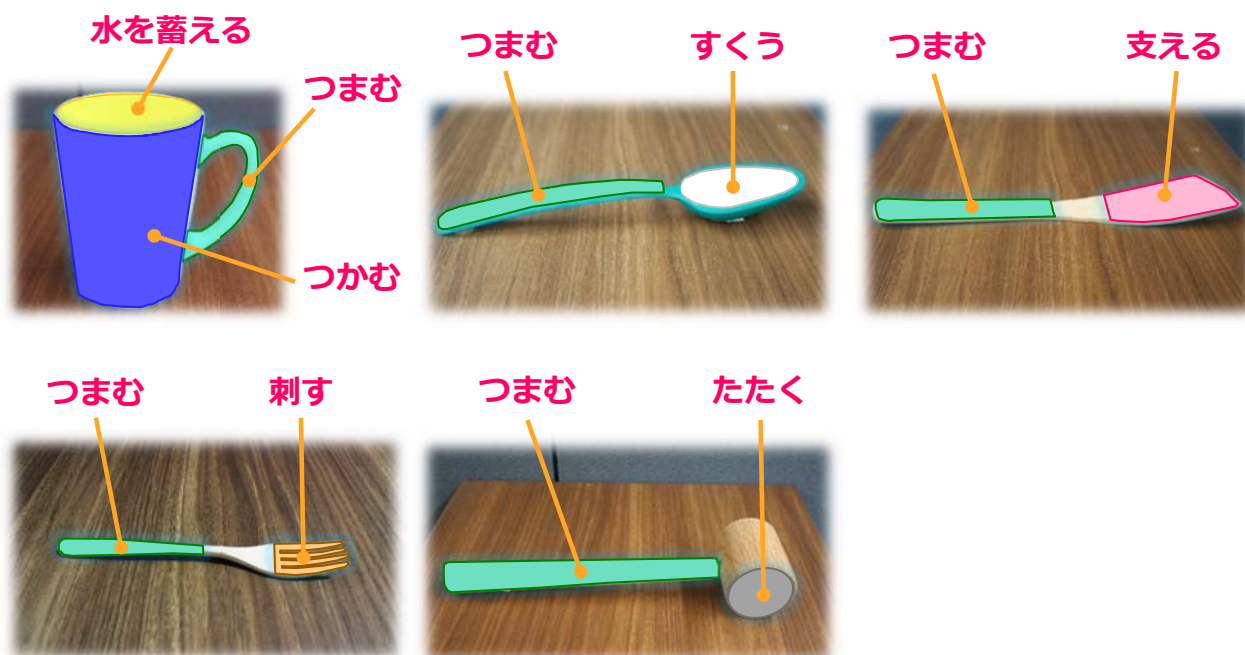
コップを把持

名前を認識
(一般物体認識)

モノの機能を認識
動作の生成

ロボット動作

“形” に作り込まれた “機能”



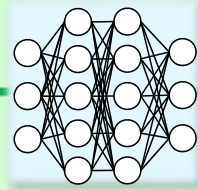
機能の発現に影響を与える要因の例

機能を決定する 主な要因	具体例
形状	いすの support, コップの grasp, コップの contain, ヘラの support, おたまの scoop, スプーンの scoop, 水筒の wrap-grasp, 弁当箱の contain, 洗面器の contain, バケツの contain, アイロン台の support, シャワーカーテンの support, うちわの support, アルミカップの contain, 植木鉢の contain, ペンの grasp, かぎの grasp, お盆の support, ゴミ箱の contain, 泡だて器の grasp, 台車の support, ほうきの grasp, かさの wrap-grasp, 洗剤ボトルの wrap-grasp, ドライバの wrap-grasp, 印鑑の grasp, きりの grasp, メジャーの wrap-grasp, 蛇口の wrap-grasp, フラワースタンドの contain, 脚立の support, 靴べらの support, 灰皿の contain, リモコンラックの contain, はかりの support, カミソリの grasp, ホットプレートの contain, やかんの contain, コースターの support, 水切りかごの contain
素材	いすの support, はさみの cut
大きさ	いすの support, コップの wrap-grasp
重力	コップの contain, ヘラの support, スプーンの scoop, まな板の support, 椀の contain, 皿の support, 鍋の contain, フライパンの contain, スコップの support, 急須の contain, お盆の support, かごの contain, ボウルの contain, S字フックの hang, 袋の contain, バケツの contain, ハンガーの hang, 物干し竿の hang, ペーパーホルダの hang, 簀立ての contain, 楊枝立ての contain, 鍋敷きの support, れんげの scoop, しゃもじの support, タオル掛けの hang, 洗面器の contain, アイロン台の support, アルミカップの contain, シャワーホルダの hang, 植木鉢の contain, ゴミ箱の contain, 台車の support, 脚立の support, 灰皿の contain, リモコンラックの contain, はかりの support, ホットプレートの contain, コースターの support, 水切りかごの contain
動き	ナイフの cut, ナイフの sting, フォークの sting, コップの pound, ハンマーの pound, はさみの cut, 箸の pinch, 箸の sting, スプーンの pound, 鍋の pound, フライパンの pound, 歯ブラシの sweep, のこぎりの cut, スコップの dig, くしの comb, ピンセットの pinch, クリップの pinch, ピーラーの cut, ハンマーの pound, アイスピックの sting, 爪楊枝の sting, 縫い針の sting, ペンチの pinch, カッターの cut, 洗濯バサミの pinch, ピンの sting, トングの pinch, ペンの sting, かぎの sting, ほうきの sting, ほうきの sweep, かさの sting, ドライバの sting, ボタンの pushed, くぎの sting, きりの sting, カミソリの cut

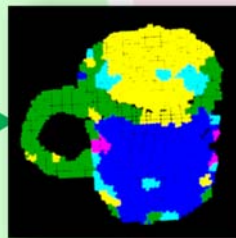
1. 仮ラベル付与



観測データ

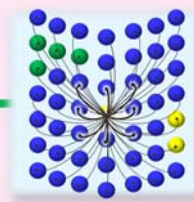


深層学習
Deep Learning

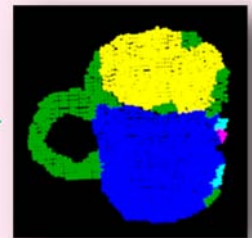


仮ラベル

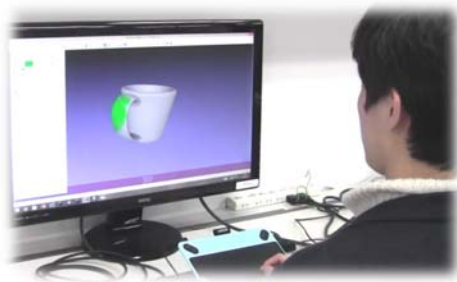
2. 最適化



確率モデル
CRF



最終ラベル



リアルタイム機能属性認識 (ランダムフォレストで学習)

中京大学 橋本研究室

認識結果

機能属性

- 液体を蓄える
- つまむ
- つかむ
- 支える
- すくう

実行環境

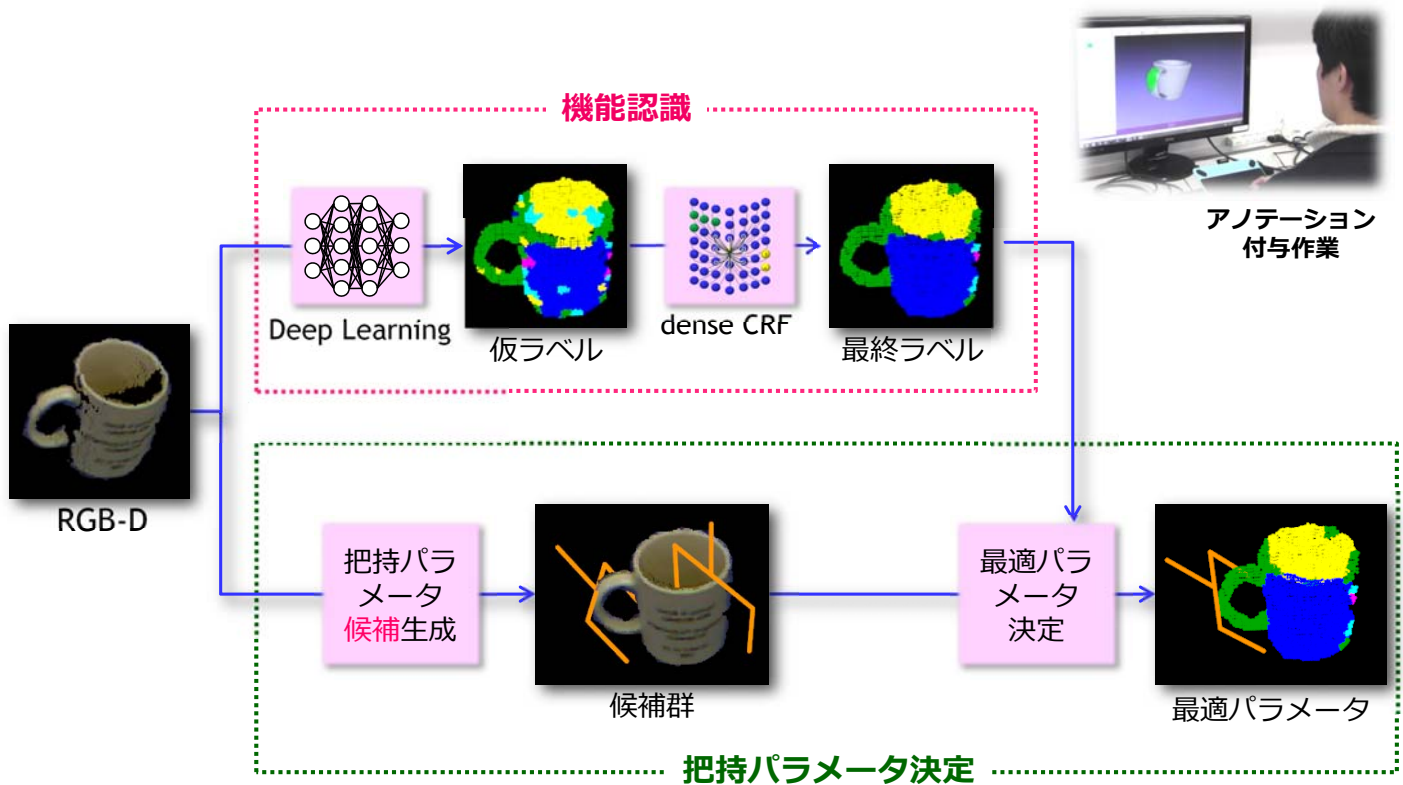
CPU: intel(R) core-i7(TM) 2.5GHz
RAM: 8GB
GPU: none

RGB

Depth

x1

“機能”を利用した最適把持パラメータ決定



生活支援ロボットへの応用

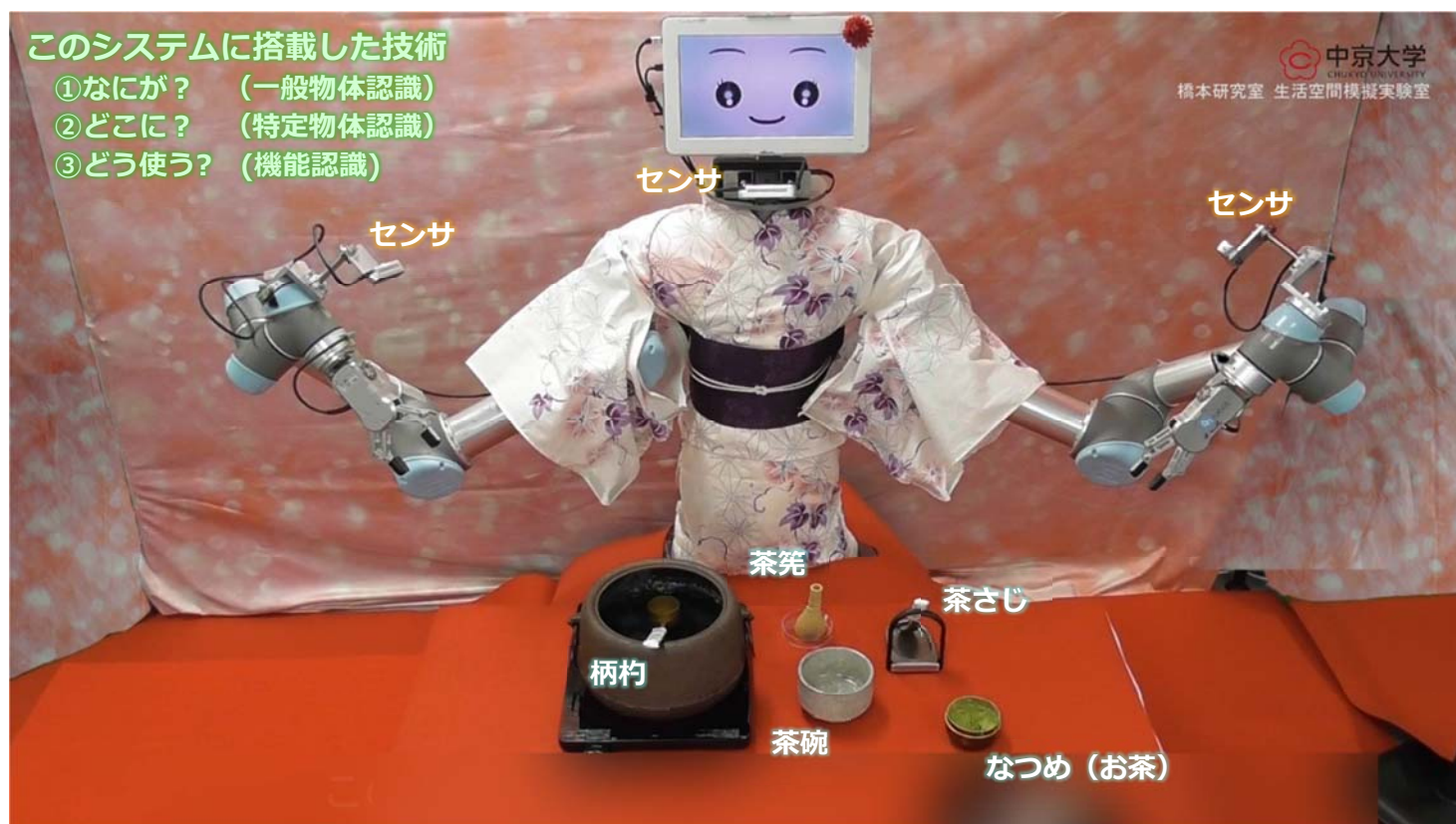
[Click here](#)

機能属性認識と一般物体認識を利用した
生活支援ロボットのプロトタイプ

NEDO プロジェクト成果

2017年 3月
中京大学

機能属性認識と一般物体認識を利用した生活支援ロボットのプロトタイプ



1. ロボットビジョンの課題
2. 現状：3次元センサ
3. 現状：特定物体ハンドリングのための物体認識
4. 現状：柔軟なロボット動作のための物体認識
5. ARCとWRSにみる実課題への対応
6. 展望：タスク生成のための物体認識
7. 残された課題

ロボットビジョンの次なる課題

【認識】

1. 認識の汎用化・抽象化（モデル不要，追加的学習不要）
2. イレギュラー検知（突発的失敗の自覚と対処法生成）

【AIのロボットタスクとの連携】

3. 要求からのブレークダウン（機能認識，言語処理）
4. ロボット動作のための認識（クラス識別⇒その後どうする？）

【関連要素】

5. 見えるものを増やす（センシングデバイス・方式）
6. 視覚だけに頼らない（人間には5感がある．Ex.触覚）

Thank you for your attention!