

事例型 MI-Hough Forest を用いた物体把持判定

Grasp Decision Using Exemplar-Based MI-Hough Forest

村井 佑輔[†], 小関 亮介[‡], 松浦 康寿[‡], 山内 悠嗣[†], 山下 隆義[†], 藤吉 弘亘[†]

Yusuke MURAI[†], Ryosuke OZEKI[‡], Yasutoshi MATSUURA[‡],

Yuji YAMAUCHI[†], Takayoshi YAMASHITA[†] and Hironobu FUJIYOSHI[†]

†: 中部大学, {ym@vision., hf@}cs.chubu.ac.jp

‡: 株式会社豊田自動織機

概要: 従来のビンピッキングシステムでは、ワークの認識と把持判定のパイプライン処理からなり、把持のための干渉領域を人手により教示している。本研究では、これらの干渉領域の教示を必要とせず、把持可能な事例と把持不可能な事例を与えることによって、システムがワークの認識と把持判定を同時に実現するアプローチを提案する。提案手法では、ポジティブクラスのサンプルには、把持可能と把持不可能というサブクラスを表現し、このようなサンプルを入力して投票ベースの Random Forest を構築する。このとき、ネガティブサンプルに類似しない識別に有効なポジティブサンプルと把持判定に有効なサンプルを重視した学習を実現するために、学習サンプルに 2 種類の重みを導入する。一つ目は、ネガティブサンプルと類似しない識別に有効なポジティブサンプルに高い重みとなるように更新する。二つ目は、把持可能な事例と把持できない事例から、把持判定に有効なサンプルに高い重みとなるように更新する。重み更新により、識別に有効なサンプルと把持判定に有効なサンプルを重視した決定木の構築ができるため、把持判定の基準を自動的に獲得し、ワークの認識と把持判定を同時に行うことが可能となる。

1. はじめに

工場での部品選別や組み立ての自動化のために、ばら置きした物体(以下、ワークと記述)をロボットアームにより把持をするビンピッキングシステムが導入されている。このビンピッキングシステムにおける処理の流れは、図 1(a)に示すようにワークの認識後、認識したワークの周囲の干渉物体の有無により把持判定[1-7]を行う。把持判定を行うには、把持の容易さを表現するための確率モデル、障害物を予め把握するための 3 次元マップ、ハンドがワークを把持するために必要な開き幅等のルールを人が経験的に設計する。図 1(a)に示すように、把持判定のための干渉領域を人手により教示する必要がある。このような干渉領域は、ハンド機構の種類により異なり、多関節等の複雑な機構となるほど教示が難しいという問題がある。

本研究では、これらの干渉領域の教示を必要とせず、図 1(b)に示すように把持可能な事例と把持不可能な事例を与えることによって、システムがワークの認識と把持判定を同時に実行するアプローチを提案する。これにより、把持領域の教示が不要となるため、教示のコストを削

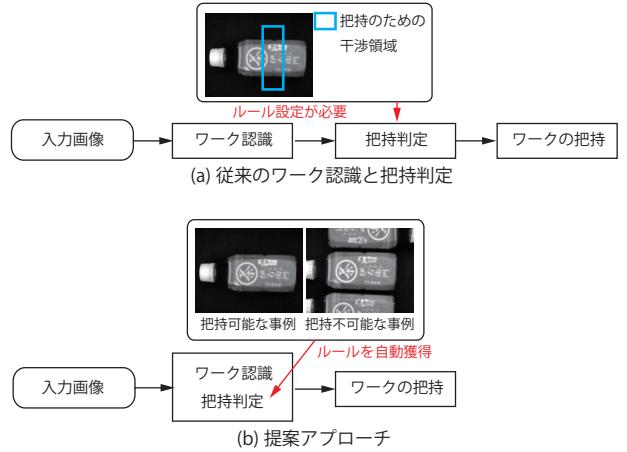


図 1 従来手法と提案アプローチの比較

減ることが期待できる。提案手法では、ポジティブクラスのサンプルには、把持可能と把持不可能というサブクラスを表現し、このようなサンプルを入力して投票ベースの Random Forest[8]を構築する。このとき、ネガティブサンプルと類似しない識別に有効なポジティブサンプルと把持判定に有効なサンプルを重視した学習を実現するために、学習サンプルに 2 種類の重みを導入する。一つ目は、ネガティブサンプルと

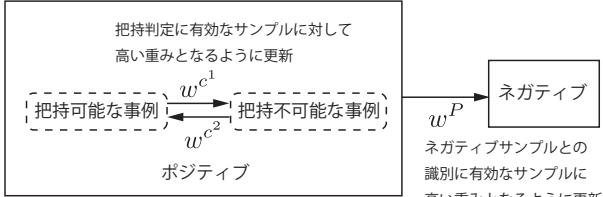


図 2 提案手法の重みの設計

類似しない識別に有効なポジティブサンプルに高い重みとなるように更新する。二つ目は、把持可能な事例と把持不可能な事例から、把持判定に有効なサンプルに高い重みとなるように更新する。このように2種類の重みを用いて学習することで、識別に有効なポジティブサンプルと把持判定に有効なサンプルを重視した決定木を構築する。これにより、把持判定の基準の自動獲得ができるため、ワークの認識と把持判定を同時に行うことが可能となる。

2. 提案アプローチ

提案アプローチは、干渉領域の教示を必要とせず、把持可能な事例と把持不可能な事例を与え、システムがワークの認識と把持判定を同時に実現する。これを実現するために、図 2 に示すように、ポジティブサンプルに対してネガティブサンプルとの判別に有効なサンプルへの重み w^P と、ポジティブクラス内のサブクラスである把持可能サンプルと把持不可能サンプルの把持判定に有効なサンプルへの重み w^{c^*} の2種類を算出する。 w^{c^*} は、把持可能サンプルに対する重みを w^{c^1} 、把持不可サンプルに対する重みを w^{c^2} とする。重みの算出には、MI-Hough Forest[9]を採用する。MI-Hough Forest は、Hough Forest[10]に Multiple Instance Learning[11]の枠組みを導入した手法である。

ここではまず、提案手法の学習サンプルの作成方法について述べ、次に学習法を述べる。

2.1. 学習サンプルの作成と Bag の構成

図 3 に示すように、学習画像から切り出したパッチ画像を基に Bag を構成することで学習サンプルを作成する。まず、学習画像として認識対象を含むポジティブ画像と認識対象を含まないネガティブ画像を定義する。このとき、ポジティブクラスには、把持可能と把持不可能というサブクラスを表現する画像を用意する。サブクラス 1 は把持可能、サブクラス 2 は把持不可能と定義する。これらの画像から切り出したパッチ画像を基に学習サンプルを作成する。回転したワークの認識を可能にするために、学習画像をそれぞれ 360 度方向に 1 度ずつ回転した画像群を作成する。

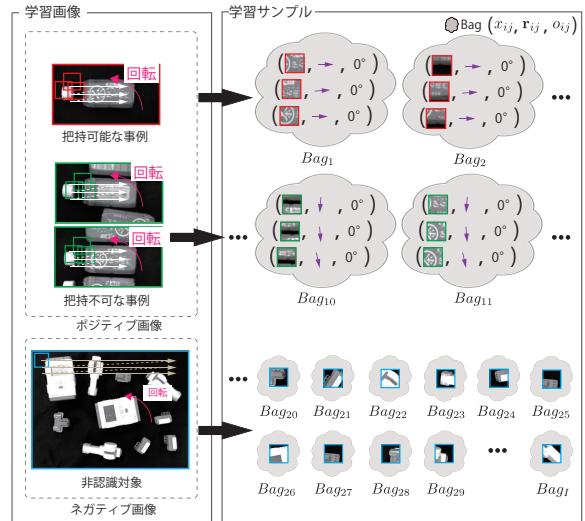


図 3 学習サンプルの作成と Bag の構成

各回転角における画像群からグリッドサンプリングにより切り出したパッチ画像 x_{ij} を用いて $Bag_i = \{x_{ij}, s_{ij}, c_{ij}, r_{ij}, o_{ij}\}$ を I 個作成する。ここで、 i は Bag の番号、 j は Bag_i に含まれるサンプルの番号、 s_{ij} はクラスマルベルでありポジティブサンプルには 1、ネガティブサンプルには 0 を割り当てる。 c_{ij} はポジティブクラスのサブクラスのラベルであり、サブクラス 1(把持可能)には 0、サブクラス 2(把持不可能)には 1 を割り当てるが、ネガティブサンプルには c_{ij} を設定しない。 r_{ij} はパッチ画像の切り出し中心位置から物体中心へのオフセットベクトル、 o_{ij} は画像の回転角である。

ポジティブクラスとネガティブクラスの Bag は異なる方法で作成する。ポジティブクラスの Bag は、画像の回転角 o_{ij} のポジティブ画像における近傍領域から切り出したパッチ画像群を Bag に入れることで作成する。ネガティブクラスの Bag は、重み更新の対象外であるため、ネガティブ画像から切り出した 1 枚のパッチ画像を Bag に入れることで作成する。そして、学習サンプルをランダムサンプリングすることでサブセットを複数作成し、サブセット毎に決定木を構築する。

2.2. 決定木の学習

2 種類の重みを更新する提案手法の学習過程を図 4 に示す。以下に学習の各ステップの詳細について述べる。

Step1: 重みの初期化 ネガティブサンプルとの判別に有効なサンプルへの重み w_{ij}^P 、ポジティブクラス内のサブクラスである把持可能サンプルと把持不可能サンプルの把持判定に有効なサンプルへの重み $w_{ij}^{c^*}$ を 1 で初期化する。

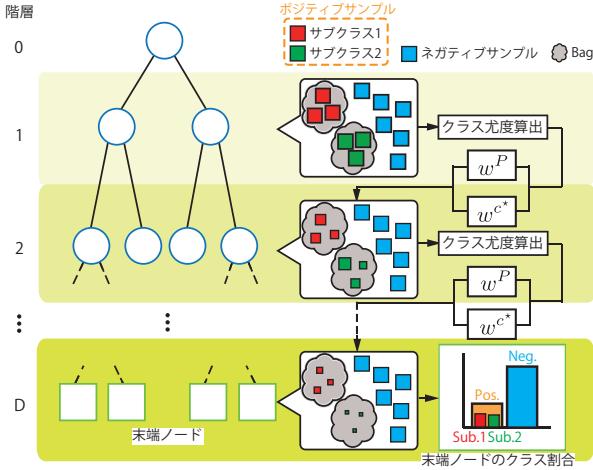


図 4 提案手法の学習過程

Step2: 分岐関数の候補の作成 本研究では, Dominant Orientation Templates(DOT)[12]を基に設計した分岐関数[13]を用いる。DOT は, セルと呼ばれる局所領域毎に作成した勾配方向ヒストグラムと, セルの中心画素の勾配方向により求める。セル毎に 8 つのビンを用意し, 勾配情報を 7 つのビンに投票し, しきい値処理をすることにより 2 値化する。最上位ビットは, セルの中心画素の勾配方向が垂直である場合は 1, それ以外は 0 にする。類似度関数 M を式(1)に定義する。

$$M(x_{ij}, T) = \sum_{\substack{V_l(x_{ij}) \in x_{ij} \\ V_l(T) \in T}} \begin{cases} 0 & \text{if } \delta(V_l(x_{ij}) \otimes V_l(T)) \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

$V_l(x_{ij})$, $V_l(T)$ はそれぞれ入力パッチ画像 x_{ij} と後述するテンプレートパッチ画像 T のセル l における DOT を表している。 δ はデルタ関数, \otimes は AND 演算子であり, $V_l(x_{ij})$, $V_l(T)$ の AND 演算をした結果, 1 ビットでも 1 となる場合に 1 を返す。1 を返したセルの数の総和が x_{ij} , T の類似度となる。

分岐関数の候補 h は, 学習パッチ画像 x_{ij} とテンプレートパッチ画像 T との類似度測定関数 M , しきい値 τ により式(2)のように定義する。

$$h_{T,\tau}(x_{ij}, T) = \begin{cases} 0 & \text{if } M(x_{ij}, T) < \tau \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

テンプレートパッチ画像 T は分岐ノードにおけるポジティブサンプル群から選択される。図 5 に示すように学習パッチ画像 x_{ij} は, テンプレートパッチ画像 T との類似度がしきい値 τ 以下の場合は左の子ノード, それ以外の場合は右の子ノードに分岐させる。テンプレートパッチ画像 T としきい値 τ をランダムで選択する処理を繰り返すことにより複数の候補を作成する。

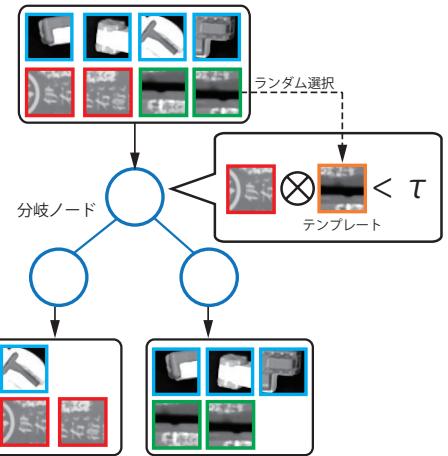


図 5 分岐ノードにおけるサンプルの分岐方法

Step3: 分岐関数の評価 複数の分岐関数候補から最適なものを決定するために, 式(3)の評価関数を用いる。

$$(T, \tau) = \underset{T, \tau}{\operatorname{argmin}} (U_*(\{x_{ij} | h_{T,\tau} = 0\}) + U_*(\{x_{ij} | h_{T,\tau} = 1\})) \quad (3)$$

ここで, $\{x_{ij} | h_{T,\tau} = 0\}$ は左の子ノードに分割されたサンプル集合, $\{x_{ij} | h_{T,\tau} = 1\}$ は右の子ノードに分割されたサンプル集合である。評価関数 U_* は, 以下の 3 つの基準を用意し, 決定木の階層に応じて 1~3 の基準を順番に切り替える。

1 つ目は, 左右の子ノードに分割したサンプルの重み付き確率を用いてエントロピーを算出することでネガティブサンプルとポジティブサンプルの分けやすさを式(4)により評価する。

$$U_1(\mathbf{A}) = |\mathbf{A}|(-a \cdot \log a - (1-a) \cdot \log(1-a)) \quad (4)$$

ここで \mathbf{A} は各子ノードに分岐したサンプル集合, a はノードの重み付きのポジティブサンプルの割合であり, 式(5)により求める。

$$a = \frac{\sum_{\substack{w_{ij}^p \in \mathbf{A} \wedge s_{ij}=1}} w_{ij}^p}{\sum_{w_{ij}^p \in \mathbf{A}} w_{ij}^p} \quad (5)$$

2 つ目は, 左右の子ノードに分割したポジティブサンプル内のサブクラス 1 とサブクラス 2 のサンプルの分けやすさを評価する。 $\mathbf{B} = \{\mathbf{A} | s_{ij} = 1\}$ であり, \mathbf{B} はポジティブサンプルのみの集合となる。

$$U_2(\mathbf{B}) = |\mathbf{B}|(-b \cdot \log b - (1-b) \cdot \log(1-b)) \quad (6)$$

式(6)の評価は, サブクラス間で類似しているサンプルと類似していないサンプルを分けるために用いる。ネガティブサンプルしか含まれていないノードからは

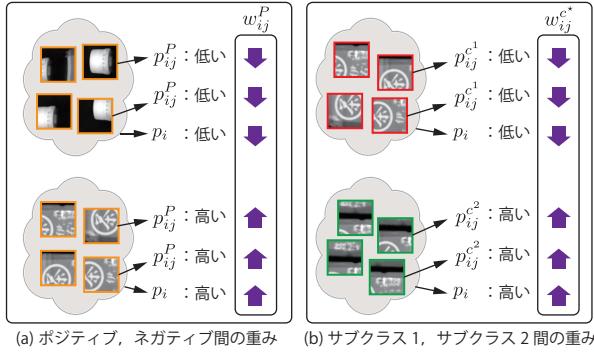


図 6 提案手法による学習サンプルの重み付け
ノードを作成しないため、式(6)の評価が適用されることはない。ここで b はノードの重み付きのサブクラス 1 の割合であり、式(7)により求める。

$$b = \frac{\sum_{\substack{w_{ij}^{c^*} \in \mathbf{B} \wedge c_{ij}=1}} w_{ij}^{c^*}}{\sum_{\substack{w_{ij}^{c^*} \in \mathbf{B}}} w_{ij}^{c^*}} \quad (7)$$

3つ目は、パッチ画像の切り出し位置のばらつきを式(8)により評価する。

$$U_3(\mathbf{B}) = \sum_{\mathbf{r}_i \in \mathbf{B}} (\mathbf{r}_{ij} - \mathbf{r}_{\mathbf{B}})^2 \quad (8)$$

$\mathbf{r}_{\mathbf{B}}$ はオフセットベクトル \mathbf{r}_{ij} の平均値であるため、 U_3 はオフセットベクトルの分散に相当する。

Step4: サンプルの重み更新 重みの更新は決定木の階層 d における全ての分岐ノードを作成した後に行い、更新対象はポジティブサンプルのみとする。重み w_{ij}^P , $w_{ij}^{c^*}$ 共にサンプルのクラス尤度 p_{ij}^P , $p_{ij}^{c^*}$ と Bag のクラス尤度 p_i により求める。 p_{ij}^P , $p_{ij}^{c^*}$ はそれぞれ式(9), (10)により求める。

$$p_{ij}^P = \frac{1}{1 + \exp(1 - F(x_{ij}))} \quad (9)$$

関数 F は $F(x_{ij}) = 2a - 1$ と定義する。サンプルのクラス尤度 p_{ij}^P が高い値であるほど認識対象、低い値であるほど非認識対象である可能性が高いことを表している。

$$p_{ij}^{c^*} = \frac{1}{1 + \exp(1 - G(x_{ij}))} \quad (10)$$

関数 G はサブクラス 1 とサブクラス 2 に対して 2 種類用意する。サブクラス 1 に対して p_{ij}^{c1} を求めるための関数 G_{c1} は $G_{c1}(x_{ij}) = 2b - 1$ 、サブクラス 2 に対して p_{ij}^{c2} を求めるための関数 G_{c2} は $G_{c2}(x_{ij}) = 2b - 1$ と

定義する。 $p_{ij}^{c^*}$ が高い値であるほどサブクラス間で特徴的、低い値であるほどサブクラス間で類似するサンプルが存在している可能性が高いことを表している。

Bag のクラス尤度 p_i は Bag に属するサンプルのクラス尤度を用いて式(11)より計算する。

$$p_i = \prod_{j \in i} p_{ij}^* \quad (11)$$

これにより、クラス尤度が低いサンプルが多く含まれている Bag のクラス尤度が低くなる。そして、重み w_{ij}^P , $w_{ij}^{c^*}$ は式(12)により求める。

$$w_{ij}^* = p_i \cdot p_{ij}^* \quad (12)$$

最後に、決定木の階層を d 、階層 d の全ての分岐ノードに含まれるサンプル集合を $\mathbf{S}^{(d)}$ として、重みの正規化を式(13), (14)により行う。

$$w_{ij}^P = \frac{w_{ij}^P}{\sum_{\substack{w_{ij}^P \in \mathbf{A}}} w_{ij}^P} \quad (13)$$

$$w_{ij}^{c^*} = \frac{w_{ij}^{c^*}}{\sum_{\substack{w_{ij}^{c^*} \in \mathbf{B}}} w_{ij}^{c^*}} \quad (14)$$

図 6(a)に示すようにサンプルのクラス尤度 p_{ij}^P の値が総じて低く、その結果 Bag のクラス尤度 p_i の値が低い場合、その各サンプルはネガティブサンプルとの類似度が高いサンプルとして Bag 中のサンプルの重み w_{ij}^P は減少する。また、図 6(b)に示すようにサンプルのクラス尤度 $p_{ij}^{c^*}$ の値が総じて低く、その結果 Bag のクラス尤度 p_i の値が低い場合、その各サンプルはサブクラス間の類似度が高いサンプルとして Bag 中のサンプルの重み $w_{ij}^{c^*}$ は減少する。一方、特徴的なサンプルの重み w_{ij}^P , $w_{ij}^{c^*}$ は向上し、重点的に学習を行う。

Step5: Step2～Step4の繰り返し Step2～Step4のノードの作成と重み更新を指定した深さ D に達するか、もしくはサンプル数が一定数未満になるまで繰り返す。階層が深くなるにつれ、重みが低いサンプルの影響を低減した分岐構造を持つ決定木が構築される。

Step6: 末端ノードの作成 末端ノード l には、図 7 に示すようにポジティブサンプルの割合 $Q_{t,l}^{(0)} = a$ 、サブクラス 1 の割合 $Q_{t,l}^{(1)} = b$ 、サブクラス 2 の割合

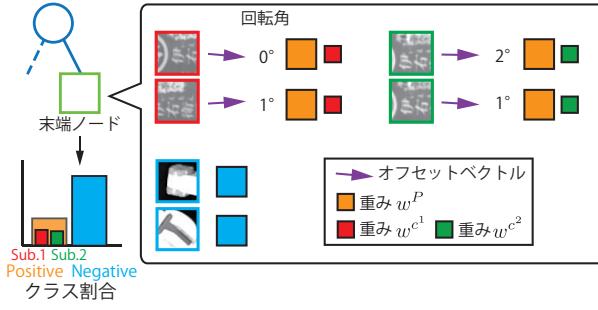


図 7 末端ノードの情報

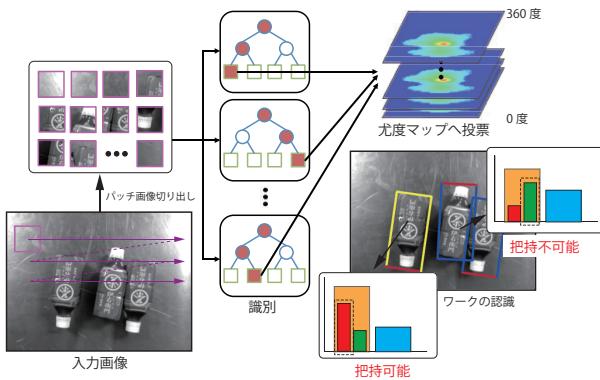


図 8 提案手法の認識過程

$Q_{t,l}^{(2)} = 1 - b$, オフセットベクトル $\mathbf{R}_{t,l} = \mathbf{r}_{ij}$, 回転角 $O_{t,l} = o_{ij}$ を保存する. ここで, t は決定木の番号, l は末端ノードの番号である.

以上により決定木を構築することで、階層が深くなる度にネガティブサンプルやサブクラス間で類似しているサンプルの重みを自動的に低くすることができる.

2.3. 認識過程

ワークの認識と把持判定を行う提案手法の認識過程を図 8 に示す. 以下に詳しい処理を述べる.

2.3.1. 投票処理

学習した決定木を用いた投票処理により行う. 投票の際には、回転角 θ 每に入力画像と同じサイズの投票平面を用意して 3 次元尤度マップを構成する. さらに、尤度マップの 1 つの座標には、ポジティブクラス、サブクラス 1, サブクラス 2 の投票用に 3 つのビン i を用意する. ここで、ポジティブサンプルは $i = 0$, サブクラス 1 は $i = 1$, サブクラス 2 は $i = 2$ のビンへ投票をする. 入力画像の位置 \mathbf{p} から切り出したパッチ画像 x を決定木 t に入力すると、末端ノード l に到達する. そして、末端ノード l に保存されているポジティブサンプルの割合 $Q_{t,l}^{(0)}$, サブクラス 1 の割合 $Q_{t,l}^{(1)}$, サブクラス 2 の割合 $Q_{t,l}^{(2)}$, オフセットベクトル $\mathbf{R}_{t,l}$, 回転角 $O_{t,l}$ を得ることができる. そして、式(15)を用いて、位置 \mathbf{p}

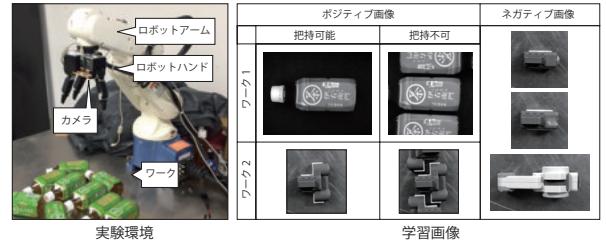


図 9 実験環境と学習に用いた事例

からオフセットベクトル $\mathbf{R}_{t,l}$ に向かい尤度を投票する. θ は、回転角 o に対応する投票面である. 投票処理を決定木の本数分行い統合することで、尤度マップ V を得ることができる.

$$V_{\theta, \mathbf{p} + \mathbf{R}_{ij}}^{(i)} = \sum_t Q_{t,l}^{(i)} \quad (15)$$

作成した尤度マップに対して、以下の手順によりワークの認識を行う.

1. 投票平面 θ において局所領域を走査
2. 局所領域ごとに投票値の総和を求める
3. 投票平面 θ 每に総和の最大値(スコア)を求める
4. スコアが最大の投票平面の θ を回転、局所領域の中心点を位置として認識

2.3.2. 把持判定

ワークの認識後、把持判定を行う. 認識した位置を p' 、回転を θ' としたときに、式(16)により把持判定を行う.

$$c^* = \operatorname{argmax}_{1 \leq i \leq 2} V_{\theta', \mathbf{p}'}^{(i)} \quad (16)$$

認識した回転角 θ' 、位置 p' に対応するサブクラス 1 とサブクラス 2 の投票値を比較し、高い値を持つサブクラスに決定する. したがって、ビン 1(サブクラス 1)の投票値が高い場合は把持可能、ビン 2(サブクラス 2)の投票値が高い場合は把持不可能と判定することができる.

3. 評価実験

3.1. 実験概要

評価画像の撮影環境と学習に用いた事例を図 9 に示す. 評価画像の作成には、まず画像中心に回転角 0 度の状態でワークを配置する. さらに、周囲に複数のワークを配置することで 1 枚の画像を用意する. そして、1 枚の画像を 360 度まで 1 度ずつ回転させることで評価画像を作成する. 図 10 に示す把持可能な画像を 360 度 × 2(シーン) × 2(ワーク数)枚、把持不可能な画像を 360 度 × 2(シーン) × 2(ワーク数)枚用意した. 正解データは、位置に対しては画像の中心位置、

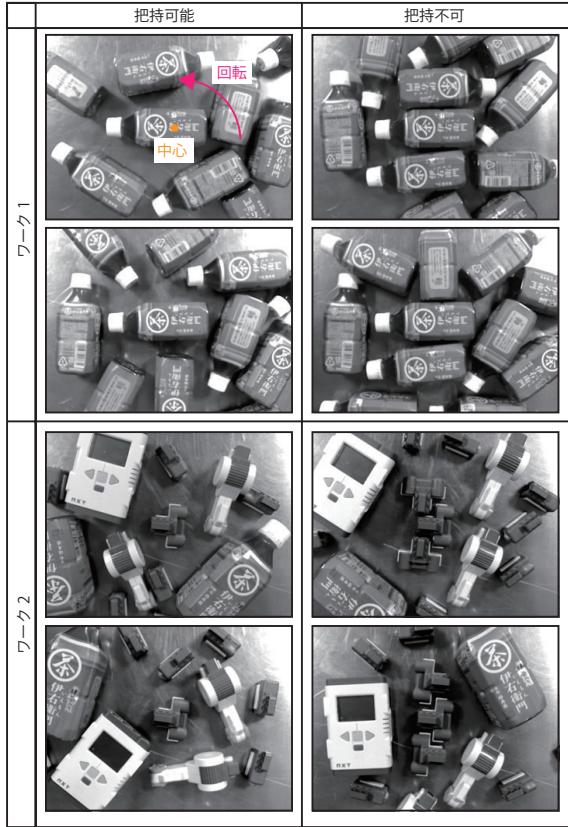


図 10 評価画像

回転に対しては回転変化させたときの回転角、把持判定に対してはロボットハンドの指が約 2cm のため、2cm 以内にワークが存在する場合は把持不可能、それ以外は把持可能として手動で正解データを与えた。評価実験では、画像中心に配置したワークを認識した結果、位置ずれ誤差が 5pixel 以内かつ回転に対する誤差が 5 度以内のときに成功として認識率を算出する。さらに、認識に成功したワークに対して正解と一致、不一致により把持判定の正解率を算出する。

3.2. 実験結果

認識結果を表 1 に示す。

表 1 実験結果[%]

	認識率	正解率	
		把持可能	把持不可
ワーク 1	95.1	100.0	98.1
ワーク 2	85.1	84.0	82.5

ワーク 1 の認識率は 95.1%，把持可能、把持不可能の正解率はそれぞれ 100.0%，98.1% であった。ワーク 2 の認識率は 85.1%，把持可能、把持不可能の正解率はそれぞれ 84.0%，82.5% であった。以上の結果から、ワークの認識と把持判定を同時にを行うことが可能であることが確認できる。

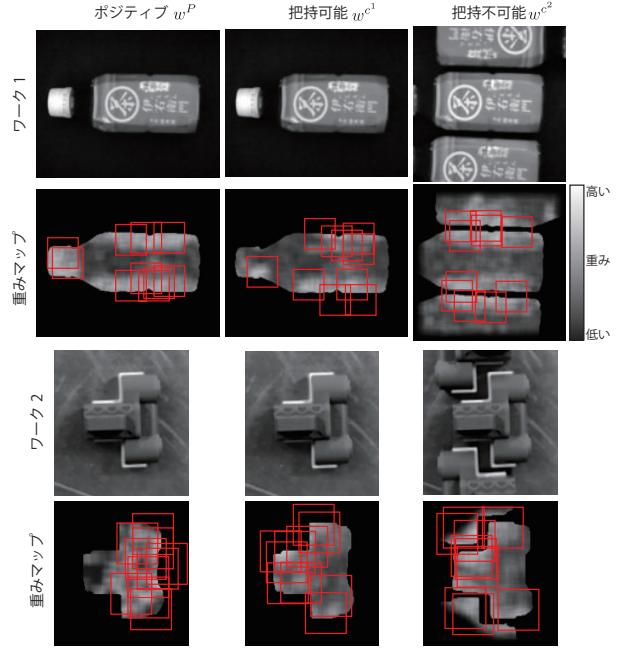


図 11 重みマップの可視化

3.3. 考察

提案手法の認識精度について考察する。まず、図 11 に学習で学習サンプルに付けられた重みの可視化を示す。図中の赤色の短形は、マップ全体で上位 10 個の重みに対応する座標を中心として描画している。ワーク 1 の重みマップから、ポジティブクラスの重みは学習に用いたネガティブサンプルとアピアランスが類似しない輪郭領域が高く、把持可能クラスと把持不可能クラスでは、それぞれのサンプル間で類似しない輪郭領域を含むサンプルの重みが高いことがわかる。ワーク 2 も同様の傾向があり、ポジティブクラスの重みマップはワークの左側の領域のようにネガティブサンプルとアピアランスが類似する領域の重みが低く、類似しない領域の重みが高くなっている。さらに、把持可能クラスと把持不可能クラスでアピアランスが類似しない領域の重みが高いことがわかる。したがって、提案アプローチの学習によって所望の重み付けを実現できたことが確認できた。ワーク認識例を図 12 に示す。図中の短形の位置はワークの位置、短形の赤線はワークの回転、短形が黄色の場合は把持可能、青色の場合は把持不可能なことを表している。図中のヒストグラムは、認識位置に対する把持可能サンプルと把持不可能サンプルの投票の割合を表している。認識例から、シーンに含まれているワークの認識が高精度に可能なことが確認できる。さらに、ワーク 2 の尤度マップをみると、類似したアピアランスを持つ物体領域の投票値を抑制できていることが確認できる。図中のヒストグラムをみると、把持可能なワ

ークに対してはサブクラス 1 において投票値が高く、把持不可能なワークに対してはサブクラス 2 において投票値が高いことが確認できる。しかし、全てのシーンに共通して干渉物体が存在するワークが把持可能と判定されている場合がある。これは、把持不可能の事例に用いた画像に含まれないワークの配置のため、サブクラス 2 からの投票の値が小さくなつたためだと考えられる。

4.まとめ

事例を与えることにより自動で把持判定のルールを獲得することで、ワーク認識と把持判定を同時に手法を提案した。提案手法は、認識率、把持判定正解率から、安定してワークの認識ができるることを示した。今後は、提案手法を追加学習[14]の枠組みに適用し、オンラインで入力されるデータに対して自動でルールを設定し、把持を行うシステムの構築を検討する予定である。

参考文献

- [1] A. Bicchi, and V. Kumar: Robotic grasping and contact: a review, International Conference on Robotics and Automation, 2000.
- [2] M. T. Mason, and J. K. Salisbury: Manipulator grasping and pushing operations, Robot Hands and the Mechanics of Manipulation. Cambridge, MA: The MIT Press, 1985.
- [3] K. Shimoga: Robot grasp synthesis: a survey, The International Journal of Robotics Research, vol. 15, pp.230-266, 1996.
- [4] A. M. Okamura, N. Smaby, and M. R. Cutkosky: An overview of dexterous manipulation: International Conference on Robotics and Automation, 2000.
- [5] J. Glover, D. Rus, and N. Roy: Probabilistic Models of Object Geometry for Grasp Planning, Robotics: Science and Systems IV, 2008.
- [6] A. Saxena, L. Wong, M. Quigley, and A. Y. Ng: A vision-based system for grasping novel objects, Robotics Research. Springer Berlin Heidelberg, 2011, pp.337-348.
- [7] 堂前幸康, 奥田晴久, 北明靖雄, 永谷達也, 野田哲男: 物体形状に依存せず高速なバラ積み物体の取り出し方法, 第 29 回 日本ロボット学会学術講演会, 2011.
- [8] L. Breiman: Random Forests, Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.
- [9] 村井佑輔, 三品陽平, 小関亮介, 松浦康寿, 山内悠嗣, 山下隆義, 藤吉弘亘: MI-Hough Forest による類似物の影響を受けない特定物体検出, 画像センシングシンポジウム, 2014.
- [10] J. Gall, and V. Lempitsky: Class-specific hough forests for object detection, Computer Vision and Pattern Recognition, 2009.
- [11] T. G. Dietterich, and R. H. Lathrop: Solving the multiple-instance problem with axis-parallel rectangles, Artificial Intelligence, vol.89, pp.31-71, 1997.
- [12] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab: Dominant Orientation Templates for Real-Time Detection of Texture-Less Objects, Computer Vision and Pattern Recognition, pp. 2257-2264, 2010.
- [13] D. Tang, Y. Liu, and T.-K. Kim: Fast pedestrian detection by cascaded random forest with dominant orientation templates, British Machine Vision Conference, 2012.
- [14] 三品陽平, 村田隆英, 山内悠嗣, 山下隆義, 藤吉弘亘: Random Forest を用いた事例型追加学習, 画像センシングシンポジウム, 2014.

村井佑輔: 2013 年 中部大学工学部情報工学科修了。同年より同大学大学院博士前期課程に在籍。画像処理、パターン認識に関する研究に従事。

小関亮介: 2007 年、愛知県立大学大学院情報科学専攻博士前期過程修了。現在、株式会社豊田自動織機 技術・開発本部に所属。コンピュータビジョンを応用した技術開発に従事。

松浦康寿: 2003 年、長崎大学大学院電気電子工学専攻博士前期過程修了。現在、株式会社豊田自動織機 技術・開発本部に所属。ロボットハンドの技術開発に従事。

山内悠嗣: 2012 年 中部大学大学院博士後期課程修了。2012 年 中部大学大学院博士研究員。2014 年 中部大学助手。2010 年 独立行政法人日本学術振興会特別研究員 DC2。コンピュータビジョン、パターン認識の研究に従事。

山下隆義: 2002 年 奈良先端科学技術大学院大学博士前期課程修了。2002 年 オムロン株式会社入社。2009 年 中部大学大学院博士後期課程修了(社会人ドクター)。2014 年 中部大学講師。人の理解に向けた動画像処理、パターン認識・機械学習の研究に従事。2009 年 画像センシングシンポジウム高木賞。2013 年 電子情報通信学会 情報・システムソサイエティ論文賞。2013 年 電子情報通信学会 PRMU 研究会研究奨励賞。

藤吉弘亘: 1997 年 中部大学大学院博士後期課程修了。1997~2000 年 米カーネギーメロン大学ロボット工学研究所 Postdoctoral Fellow。2000 年 中部大学講師。2004 年同大准教授を経て 2010 年より同大教授。2005~2006 年 米カーネギーメロン大学ロボット工学研究所客員研究員、計算機視覚、動画像処理、パターン認識・理解の研究に従事。2005 年 ロボカップ研究賞。2009 年 情報処理学会論文誌 コンピュータビジョンとイメージメディア 優秀論文賞。2009 年 山下記念研究賞。2010 年、2013 年、2014 年 画像センシングシンポジウム優秀学術賞。2013 年 電子情報通信学会情報・システムソサイエティ論文賞。

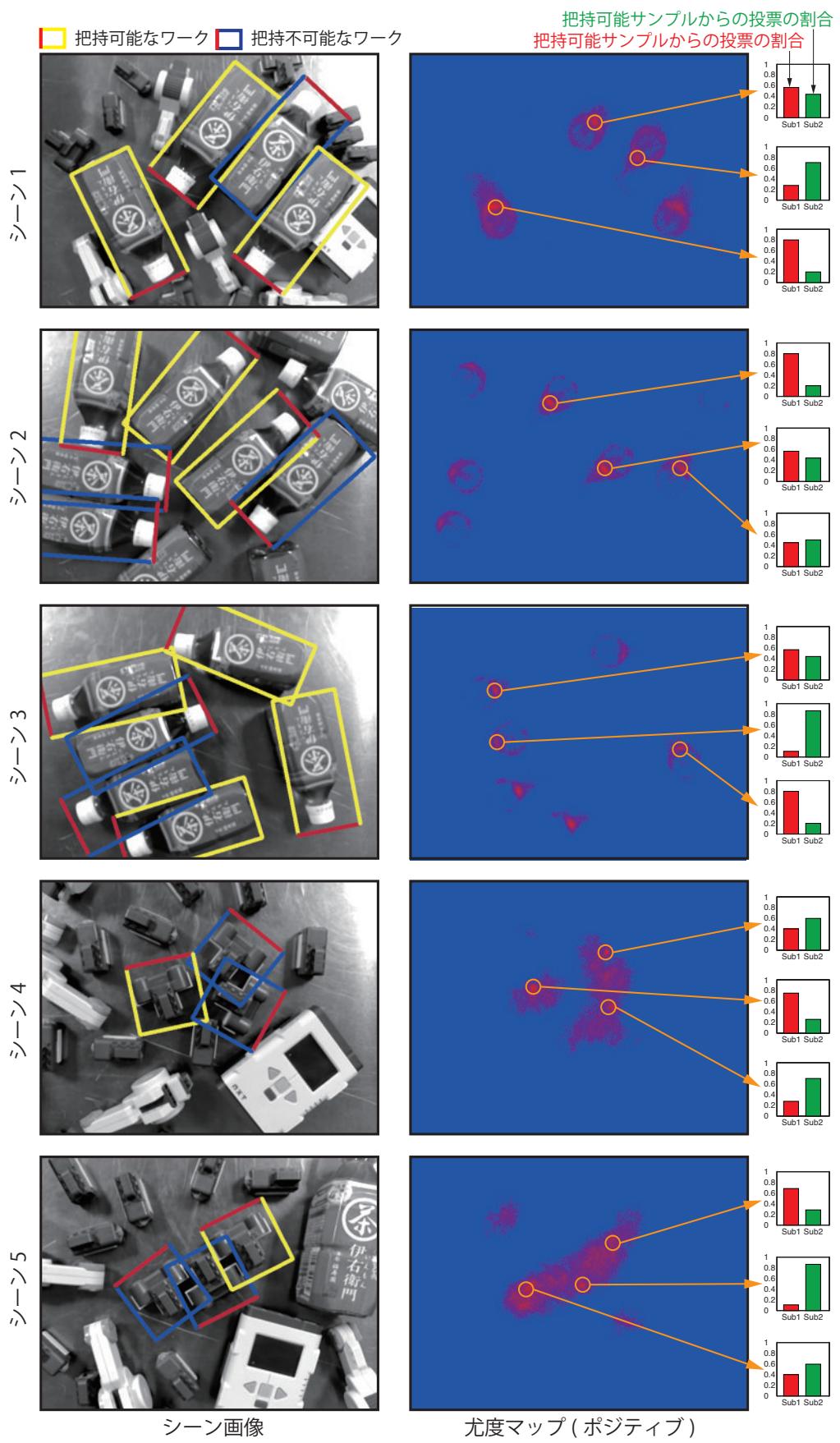


図 12 認識例と尤度マップ