

1. はじめに

本研究では、Random Forest を用いた物体検出に着目した Weighted Hough Forest を提案し、その応用としてビンピッキングに取り組んだ。本稿では、ピッキングシステムにおける Weighted Hough Forest の追加学習法について述べる。ピッキングシステムは、工場での部品組み立ての自動化や生活支援ロボットに必要な技術であり、入力画像から対象の物体（以下、ワークと記す）の位置・回転を検出してワークの把持を行う。実利用シーンでは、ワーク同士が密着していない把持可能な場合や、ワーク同士が密着している把持不可能な場合があるため、ワークの検出に成功した場合でも把持に失敗する場合がある。そのため、検出したワークが把持可能かどうか判定する必要がある。実利用シーンにおいて把持不可能なワークを判定する基準を獲得する方法として、追加学習が考えられる。追加学習は、ワークの把持に失敗したシーン画像から作成したサンプルに把持不可能というラベルを与え、識別器が把持不可能サンプルを正しく識別できるように学習する。Random Forest は、木構造を用いてサンプルを識別するため、追加学習の際に把持不可能サンプルが辿り着いた末端ノードのみを更新することで、識別器の変更を最小に抑えて把持不可能サンプルに適応できると考えられる。そこで、本研究では Random Forest を用いたピッキングシステムのための追加学習を提案する。

2. Weighted Hough Forest による追加学習

提案手法は、まず把持に失敗したシーン画像から作成した把持不可能サンプルを決定木に入力する。その後、把持不可能サンプルが辿り着いた末端ノードを更新する。まず、追加学習のフレームワークを述べた後に、提案手法のアルゴリズムについて説明する。

2.1 追加学習のフレームワーク

追加学習のフレームワークを図 1 に示す。まず、把持可能な検出対象ワークをポジティブサンプル、非検出対象をネガティブサンプルとして決定木を構築する。構築した決定木を用いてシーン画像から検出対象ワークの位置・回転を検出する。そして、検出結果を用いてワークの把持動作を行い、ワークの把持に成功した場合は把持を続行、ワークの把持に失敗した場合は把持不可能サンプルを作成し、追加学習をする。このとき、把持の成功、失敗はワーク同士の密集状態から目視で判断する。

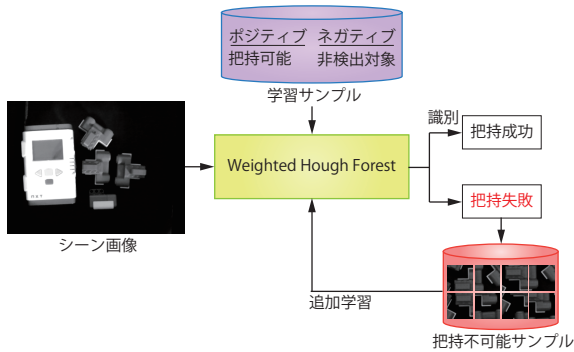


図 1: 追加学習のフレームワーク

2.2 Weighted Hough Forest の構築

Weighted Hough Forest は、Hough Forest[1] の学習サンプルに重みを導入して、ネガティブサンプルに類似するポジティブサンプルの重みを決定木の階層毎に小さくするように更新しながら決定木を構築する。これにより、検出の際に背景やネガティブサンプルと類似した領域の誤検出を低減することが可能である。

2.3 把持不可能サンプルの収集

把持不可能サンプルの収集方法を図 2 に示す。検出したワークの位置、回転に重なるようにワークのモデルを当てはめる。そして、検出対象ワークを含む領域からパッチを

切り出すことで把持不可能サンプルを収集する。上記の処理を 0~360 度方向に 1 度ずつ回転させた画像に行うことで回転したワークに対しても把持判定が可能となる。

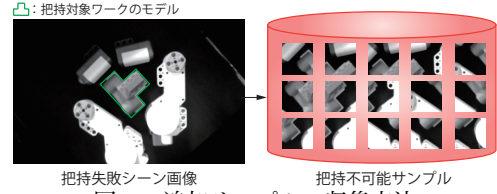


図 2: 追加サンプルの収集方法

2.4 追加学習アルゴリズム

提案する追加学習アルゴリズムは、図 3 に示すように把持可能サンプルと把持不可能サンプルの分けやすさを表す重み w^{c^*} を算出する。このとき、 w^{c^1} は把持可能サンプル、 w^{c^2} は把持不可能サンプルの重みを表しており、 w^{c^*} は把持可能サンプルと把持不可能サンプルで類似していない特徴的なサンプルに対して大きくなる。重みの導入により、把持判定の基準を獲得することが可能となる。

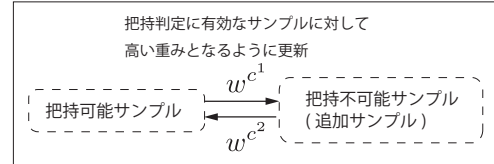


図 3: 追加学習における重み

Step1: 学習、追加サンプルの重み初期化 サンプルの重みの初期化には式 (1) を用いる。ここで、 T は追加学習の回数、 i はサンプルの番号、 $N_T^{c^1}$ は T 回目の追加学習時の把持可能サンプルもしくは把持不可能サンプルの総数を表している。1 回目の追加学習では、重み w^{c^1} と w^{c^2} の総和が 1 になるように初期化する。把持可能サンプルの重み w^{c^1} は学習後に正規化をするため、2 回目以降の追加学習では把持不可能サンプルの重み w^{c^2} のみを初期化する。 w^{c^2} は、新規追加サンプル数とそれまでの追加サンプル数の比率を用いて初期化することで w^{c^1} との均衡を保つ。

$$w_{T,i}^{c^*} = \begin{cases} 1/N_T^{c^2} & T = 1 \\ \frac{w_{T,i}^{c^2}}{\sum_{t=1}^{T-1} w_{t,i}^{c^2} + \sum_{t=1}^{T-1} \frac{N_t^{c^2}}{N_t^{c^1}}} & T \geq 2 \end{cases} \quad (1)$$

Step2: 末端ノードの更新 末端ノードの更新過程を図 4 に示す。まず、把持不可能サンプルを決定木に入力する。その後、把持不可能サンプルが辿り着いた末端ノードを更新する。追加する分岐関数は、事例型の分岐関数 [2] を用いる。事例型の分岐関数では、サンプルの中からテンプレートを選択し、入力サンプル $x_{T,i}$ とテンプレート T の類似度を計算して式 (2) に示すようにサンプルを分岐させる。

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

ここで、 S は入力サンプル $x_{T,i}$ とテンプレート T の類似度を表しており、類似度がしきい値 τ 未満の場合はサンプルを左に分岐、しきい値以上の場合は右に分岐させる。追加の分岐ノードでは、ネガティブサンプルの分岐は行わない。1 回目の追加学習時に更新された末端ノードは、検出過程では末端ノードと分岐ノードの 2 つの役割を持つ。2 回目以降の追加学習において、把持不可能サンプルを正しく識別可能な末端ノードの更新はしない。

Step3: 学習、追加サンプルの重み更新 末端ノードの更新後に追加したノード内のサンプルの重みを更新する。重みは、把持可能と把持不可能クラスのクラス尤度に基づいて式 (3)、式 (4) を用いて算出する。

$$G(x_{T,i}) = \frac{P(k_m|x_{T,i}) + \epsilon}{(1 - P(k_m|x_{T,i})) + \epsilon} \quad \epsilon = 0.000001 \quad (3)$$

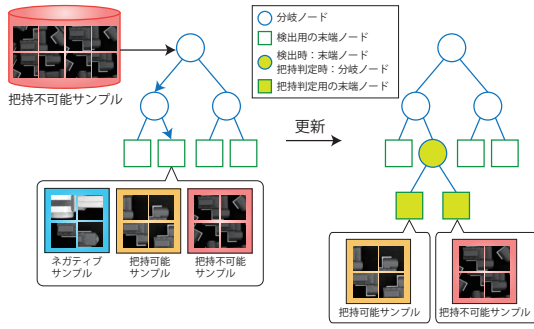


図 4: 末端ノードの更新過程

$$w_{T,i}^* = w_{(T-1)j}^* \exp(G(x_{T,i})) \quad (4)$$

ここで、 G はクラス尤度を算出する関数、 $k = \{0, 1\}$ であり 0 は把持可能クラス、1 は把持不可能クラスを表している。 $P(k_m|x_{T,i})$ は $x_{T,i}$ が k_m に属する重み付き確率を表しているため、クラス尤度が高いサンプルは重み $w_{T,i}^*$ が大きくなり、クラス尤度が低いサンプルは重みが小さくなる。そして、各クラスのサンプルの総数が足して 1 になるように式 (5) を用いて正規化する。このとき、 $\mathcal{I}_T$ は学習サンプル集合を表している。

$$w_{T,i}^* = \frac{w_{T,i}^*}{\sum_{w_{T,i}^* \in \mathcal{I}_T} w_{T,i}^*} \quad (5)$$

Step4: 末端ノードの作成 Step2 により、追加で作成された左右の子ノードを末端ノードとする。このとき、追加された子ノードの把持可能サンプルと把持不可能サンプルのクラス確率の大小関係が追加前のノードと変化がない場合は追加された子ノードを削除することで不要なノードの作成を抑制する。

2.5 検出過程

ワークの検出過程を図 5 に示す。図 5(a) に示すように、1 回目の追加学習で追加された分岐ノードは、検出時には末端ノードとして使用する。上記のノードは、把持判定時には図 5(b) に示すように分岐ノードとして使用する。入力サンプルが検出用の末端ノードに辿り着いた場合、物体中心へのオフセットベクトルを用いて尤度マップへクラス確率を投票、把持判定用の末端ノードに辿り着いた場合は、把持可能クラスと把持不可能クラスの確率を尤度マップへ投票する。そして、尤度マップの極値を探索することでワークの位置・回転を検出する。検出した位置に対する把持可能クラスと把持不可能クラスの投票値が大きいクラスに判定する。

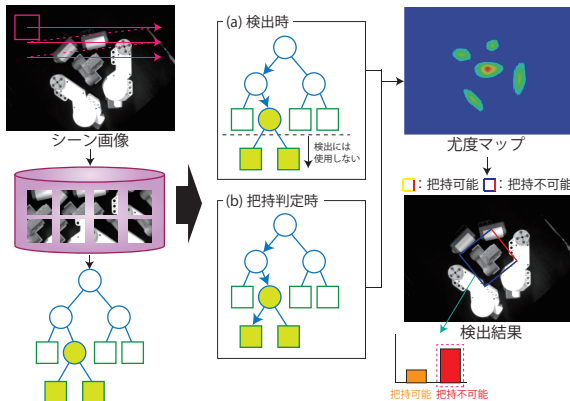


図 5: 検出過程

3. 評価実験

提案手法の有効性を確認するために追加学習前と追加学習後の検出性能を比較する。追加学習用に 100 枚の把持不可能なシーンの画像を用意した。各画像を 0~360 度まで回転させた 36000 枚の画像を追加学習と評価に用いる。評価は、検出率と把持判定の正解率により行う。

図 6 に実験結果を示す。グラフの横軸は追加回数、縦軸は 36000 枚の評価画像を用いて評価した検出率と把持判定の正解率を表している。結果から追加学習をした場合でも、

検出用の末端ノードに変更を加えていないため検出精度を維持していることが確認できる。さらに、追加回数が増加する毎に把持判定の正解率が上昇していることがわかる。以上の結果から、提案手法は追加学習前の検出精度を維持して把持判定の基準を獲得できることを示した。

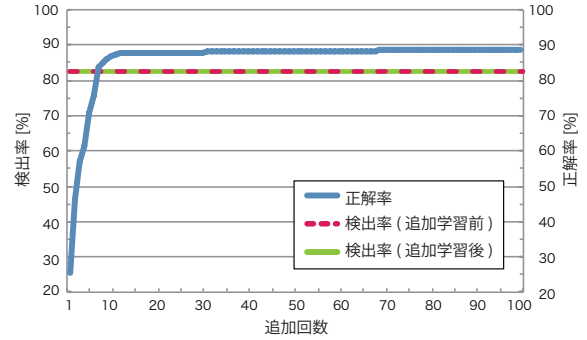


図 6: 検出結果

図 7 にサンプルの重みを可視化した重みマップを示す。赤色の短形は、重みの値が上位 10 個の位置、大きさは学習に用いたパッチサイズを表している。図 7 から把持可能サンプルと把持不可能サンプルで形状が類似していない輪郭領域の重みが大きく、その他の領域の重みが小さいことがわかる。したがって、本実験で用いたワークが把持可能か把持不可能か判定するためには輪郭領域から切り出したパッチが有効であるといえる。図 8 に検出例を示す。検出位置を中心に短形を描画しており、赤色の線がワークの向きを表している。図 8 から検出精度を維持しながら把持判定が可能であることが確認できる。

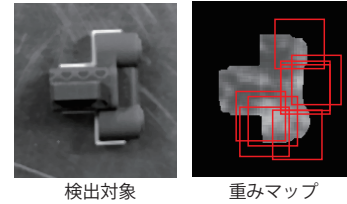


図 7: 重みマップ

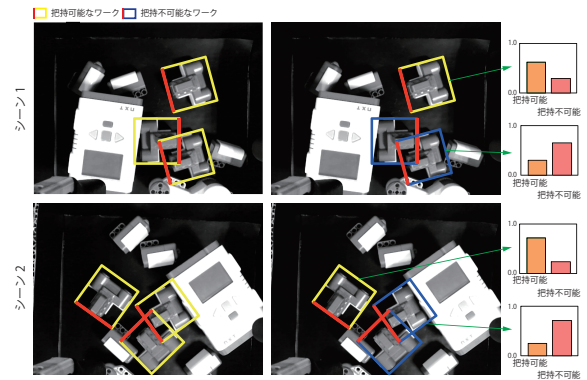


図 8: 検出例

4. おわりに

本研究では、ピッキングシステムにおける Weighted Hough Forest の追加学習法を提案した。決定木を用いることで識別器の更新を最小限に抑えることが可能なため、検出精度を維持して把持判定の基準を獲得することができると示した。今後は、見えの変化が大きいワークの検出方法について検討する予定である。

参考文献

- [1] J. Gall, and V. Lempitsky, "Class-specific hough forests for object detection", CVPR, 2009.
- [2] D. Tang, Y. Liu, and T. K. Kin, "Fast Pedestrian Detection by Cascaded Random Forest with Dominant Orientation Templates", BMVC, 2012.

研究業績

- [1] 村井佑輔, 小関亮介, 松浦康寿, 山内悠嗣, 山下隆義, 藤吉弘巨, "事例型 MI-Hough Forest を用いた物体把持判定", ViEW, OS4-H3(IS2-5), 2014.
(他学会口頭発表 4 件)