研究論文

ディープラーニングを利用した バラ積み部品のピッキングシステム開発

木嶋　祐太[*]　大野　宏[*]　山田　敏浩[**]　森田　渉[***]　阿部　淑人[****]
金泉　亮治[*****]　田中　仁[*****]　棚橋　広貴[*****]　田村　健太郎[*****]

Development of Bin Picking System using Deep Learning

KIJIMA Yuta[*], OHNO Hiroshi[*], YAMADA Toshihiro[**], MORITA Wataru[***], ABE Yoshito[****],
KANAIZUMI Ryoji[*****], TANAKA Hitoshi[*****], TANAHASHI Hiroki[*****] and TAMURA Kentaro[*****]

抄　録

本研究ではディープラーニングを利用したバラ積み部品のピッキングシステムを開発した。ピッキングシステムに必要な，ピッキングする部品の位置・姿勢がわかる機能，部品の情報を簡単にピッキングシステムへ登録できる機能，ピッキング前後のロボットの動作をプログラミング言語のPythonで記述できる環境を作成した。また，安価な3次元センサを使ったピッキングシステムを作成した。

1. 緒　言

業務の効率化，コロナ禍における無人化が課題の現状において，ロボットの活用は大いに期待される。ロボットは汎用的なアクチュエーターの役割を果たすので，特に多品種小ロットの製品を製造する工場では有用である。しかし，その汎用性が高いために，ロボットを取り扱うことが難しいことが課題となっている。この課題を解決するため，ディープラーニングの活用で，なるべく簡単にロボットを動作させることができるシステムを開発した。

本研究では，バラ積み部品のピッキングシステムを対象に，それを実現するための様々な機能を開発した。プログラムレスでの部品の切り替えやわかりやすいプログラミング言語の活用など，簡単に運用できる工夫をした。

*　下越技術支援センター
**　研究開発センター
　　レーザー・ナノテク研究室
***　素材応用技術支援センター
**** 研究開発センター
***** ケミコン長岡（株）

2. ピッキングシステムの概要

ピッキングシステムは，図 1 に示すように 3 次元センサとロボットによって構成される。バラ積み部品を 3 次元センサで撮影し，その点群を取得する。その点群からピッキングする一つの部品の位置・姿勢を取得する。その部品の位置・姿勢によりロボットを動かして，部品をピッキングする。ピッキングした後は，それぞれの部品や工程によって異なるが，部品の向きを

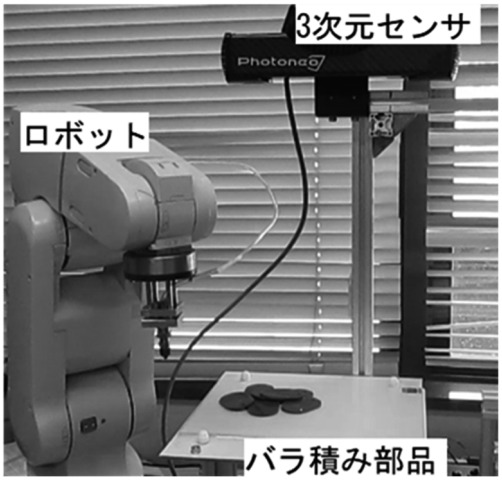

図１　ピッキングシステム

そろえたり，部品を所定の場所へ配置させたりする。

3. バラ積み部品の点群から部品の位置・姿勢の取得方法

バラ積み部品の点群から部品の位置・姿勢を取得する方法は表 1 のとおりである。

3.1 バラ積み部品の点群から距離画像の生成

距離画像は高さを輝度値に変換した画像である。本研究では，高さが高いほど輝度値が大きくなる（白くなる）画像となるようにした。バラ積み部品を 3 次元センサで撮影すると，図 2 左のような点群が得られる。この点群を距離画像に変換すると図 2 右の画像になる。図 2 右では下にある部品ほど暗くなっている。表 1 の①ではこの変換が行われる。

3.2 物体検出 DNN による部品の位置検出

物体検出 DNN（Deep Neural Network）は，画像から物体検出をするためのニューラルネットワークである。図 3 左下を対象の物体としたときに，図 3 左の画像を DNN に入力すると，図 3 右のように対象の物体の位置が検出される。

表 1 部品の位置・姿勢の取得方法

① 点群から距離画像を生成
② 距離画像から物体検出 DNN にて部品の位置を検出
③ 点群の切り出し
④ 切り出した点群と部品の形状データ（点群）に対して ICP を実行し，部品の位置・姿勢を取得
⑤ ピッキングする部品の決定

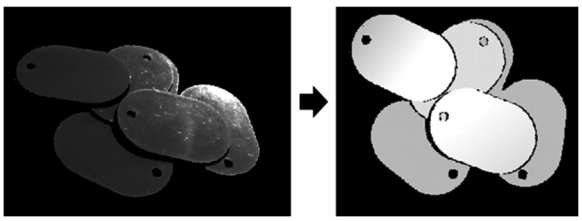

図 2 点群から距離画像への変換
（左：点群，右：距離画像）

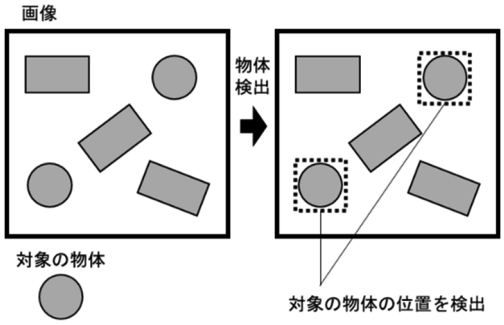

図 3 物体検出 DNN

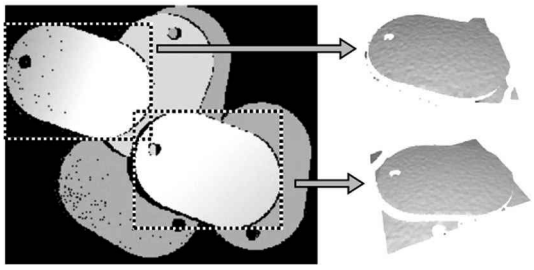

図 4 点群の切り出し

画像と対象の物体の位置からなる正解データにより学習することで，物体検出 DNN を作成できる。表 1 の②ではピッキングしたい部品を対象として学習した物体検出 DNN を用いて，距離画像から部品の位置を検出する。

3.3 点群の切り出し

表 1 の③では，バラ積み部品の点群から物体検出 DNN で検出された位置の点群を切り出す。図 4 にその例を示す。図 4 左の距離画像から点線の枠線で示されているように，2 つの部品が検出されている。この検出結果それぞれの位置の点群を切り出すと，図 4 右のような点群が切り出される。

3.4 ICP による部品の位置・姿勢の取得

ICP（Iterative Closest Point）は，2 つの点群が重なり合うように，その位置姿勢の関係を繰り返し計算で調整する方法である。図 5 左の色の薄い点群が部品の形状データで，色の濃い点群がバラ積み部品の点群から切り出した点群である。

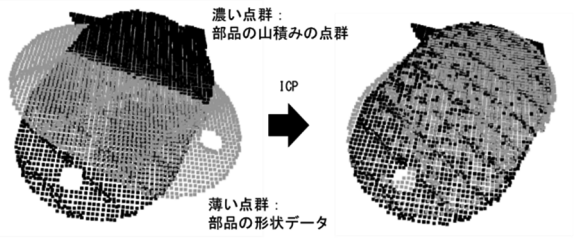

図 5　ICP

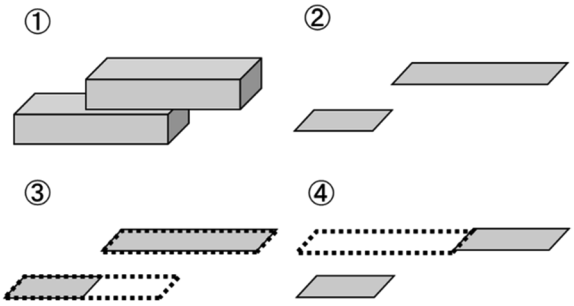

図 6　部品上の点群数の計数

これらの点群に対し ICP を実行すると，図 5 右のように両方が一致する位置・姿勢が取得できる。表 1 の④ではこの方法で部品の位置・姿勢を取得する。

3.5　ピッキングする部品の決定

図 4 のように表 1 の②で検出される部品は複数となる場合がある。その場合，表 1 の④で得られる位置・姿勢は複数となるが，ロボットは一つの部品しかピッキングできないので，どの部品をピッキングするか決める必要がある。ピッキングシステムでは，部品上にある点群数を計数し，その数が最も少ない部品を部品上に他の部品が乗っていない部品とし，ピッキングする。図 6 に部品上の点群数を計数する方法を示す。図 6 の①のバラ積み部品を 3 次元センサで撮影すると，図 6 の②に示す点群が得られる。ICP の結果，図 6 の③に点線で示す 2 つの部品の位置・姿勢が得られる。図 6 の③の下にある部品の位置・姿勢に形状データを配置し，それを持ち上げていくと，図 6 の④に示す点線の範囲が作成される。この範囲内の点群数を部品上の点群数とする。図 6 の③の 2 つの部品の位置・姿勢に対し計数し，点群数が少ない上の部品からピッキングが行われる。

4.　ピッキングしたい部品の登録

3 で説明した方法で部品の位置・姿勢を取得するために，この部品の情報をピッキングシステムに登録する必要がある。登録する流れは表 2 のとおりで，マウスクリックと数値入力だけで登録できるようにした。

4.1　部品の形状データの作成

3.4 で使用する部品の形状データを表 2 の①と②で作成する。部品一つを 3 次元センサの下に置き，撮影ボタンを押すだけで部品の形状データが取得できる。この状態だと図 7 左のように床面の点群が存在するので，しきい値となる高さを数値入力し，高さがしきい値以下の点群を削除することで，図 7 右のように床面の点群が削除できる。

4.2　ピッキング位置，部品の向き，面の向きの入力

ロボットで部品をピッキングする場合，その部品のどこをロボットが持つのかは，ピッキングを成功させるうえで非常に重要である。部品によって最適な位置は変わってしまうので，ピッキングシステムでは，ユーザーがこの位置を

表 2　部品の登録

① 部品一つを 3 次元センサで撮影し，点群を取得
② 床面の除去
③ 部品のピッキングする位置，向き，面の向きを入力
④ 物体検出 DNN の学習

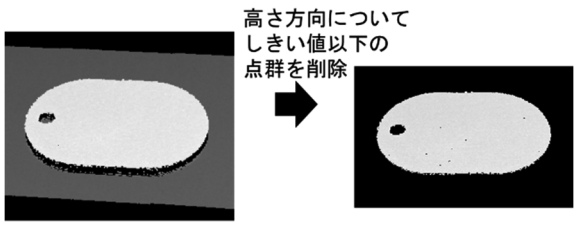

図 7　床面の除去

入力するようにした。また，図 8 に示す部品の向きや面の向きも，ピッキングやその後のハンドリングに必要な場合が多い。

そのため，これらの情報を簡単なマウス操作だけで入力できるようにした。その方法は次のとおりである。最初に，この部品の形状データから距離画像を作成して表示する。図 9 は図 8 の部品の距離画像を想定して表示したものである。表示された距離画像上で，ピッキング位置を指定したい場合は，その位置をクリック，部品の向きを指定したい場合は，その方向にドラッグをする。面の向きについては面の向きを検出する位置をクリックすることとし，クリックした付近の部品の形状データを用いて，最小二乗法により面の向きを計算する。

4.3　物体検出 DNN の学習

3.2 に記載したとおり，物体検出 DNN を学習するためには，画像と対象の物体の位置が必要になる。表 1 の②では距離画像から部品を検出することになるので，学習用の距離画像が必要となる。バラ積みを実際に撮影し，撮影されている物体の位置を指定していく方法は非常に手間がかかるので，4.1 で作成した部品の形状データを用いて，バラ積み部品を模擬した距離画像を自動生成することにした。距離画像を模擬的に作成するので，学習に必要な部品の位置も同時に作成される。

何もない空間を用意して，図 10 のように部品の形状データを一つ一つ配置していく。配置する際には，配置する部品の位置・姿勢をランダムで設定し，互いにぶつからないように配置する。一定数の配置が終わったら，その点群を距離画像に変換すると図 11 のような学習用の画像が生成できる。

この画像を 100 枚～200 枚程度作成して，学習を行うと，この部品を検出する物体検出 DNN が作成できる。学習データの作成，学習はすべて自動で行われるので，ユーザーの操作はボタンを押して 10～20 分待つだけである。

5.　Python による運用環境の作成

別の部品をピッキングするときは，4 の方法でその部品を登録するだけでよい。

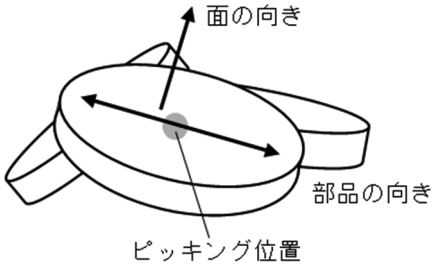

図 8　ピッキング位置，部品の向き，面の向き

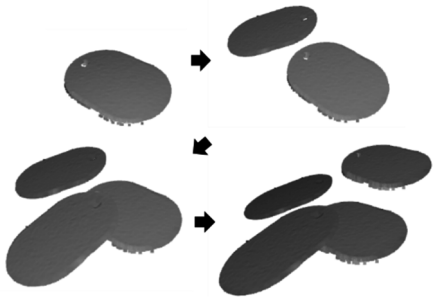

図 10　形状データの配置

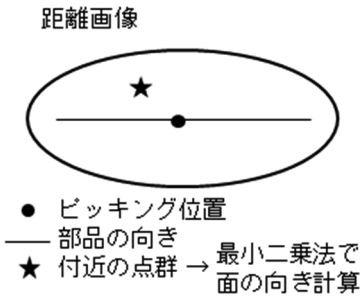

図 9　図 8 の部品の距離画像

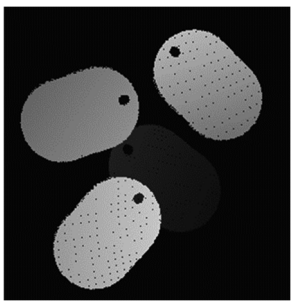

図 11　学習用の距離画像

しかし，ピッキングした後の部品の運搬，部品の整列，裏面の対応など，その他の動作は部品によって多岐に及ぶ。

そこで，ピッキングシステムを運用しやすくするために，比較的わかりやすいプログラミング言語の Python を使用して動作を記述できる環境を作成した。3 次元センサによる点群の取得，部品の位置・姿勢検出，ロボットの動作を Python の関数で実施でき，それらをうまく組み合わせることで，複雑な動作も実行できる。

①点群の取得，②部品の位置・姿勢検出，③位置・姿勢にロボットを動作，と命令を 3 行書くと，図 12 のようにロボットが移動する。ロボットの姿勢も面の向きと合っており，複雑な動作が 3 行の命令で実行できる。

6. 安価な 3 次元センサによるピッキング

図 1 に示した 3 次元センサはパターン光を投影する方式で部品の形状まで取得できるが，非常に高価である。一方，アクティブステレオ方式の数万円程度で入手可能な 3 次元センサが市販されているので，これを使用したピッキングシステムを作成した。

この 3 次元センサは分解能が低いため，今回対象とした小さな部品の図 8 に示す面の向きを検出することは難しい。部品が斜めでもピッキングできるように，先端がフレキシブルになっている吸着ハンドを使用した。これによりロボットは面の向きに合わせて傾きながらピッキングする必要が無いので，ロボットのコストも抑えることができる。図 1 のシステムと比べて 1/10 程度のコストで構成することも可能であった。ただし，ピッキングの成功率は，図 1 のシステムのほうが高いと思われる。

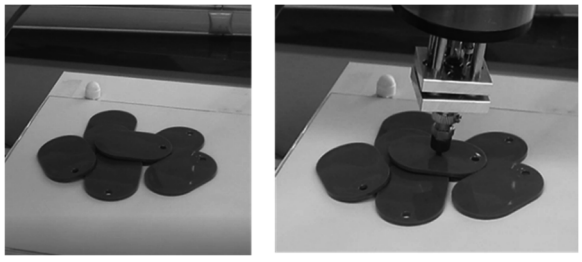

図 12　ロボットの動作

表 3　ピッキング位置と部品の向きの取得方法

① 取得した画像からインスタンスセグメンテーション DNN にて個別の部品の画素を取得
② 画素からピッキング位置や部品の向きを取得
③ ピッキング位置や向きを 3 次元座標に変換

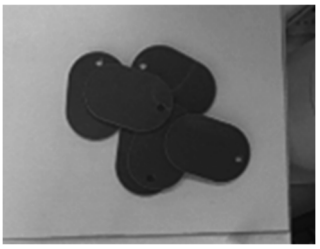

図 13　バラ積み部品の画像

6.1 ピッキング位置と部品の向きの計算

表 3 にピッキング位置と部品の向きの計算方法を示す。この 3 次元センサは 3 次元点群データに加え 2 次元画像を取得できる。表 3 の①，②では，3 次元センサで撮影した図 13 のようなバラ積み部品の画像から個別の部品のピッキング位置や部品の向きを計算する。画像から個別の部品を検出する精度の向上や部品のピッキング位置や向きを特定するため，ここではインスタンスセグメンテーション DNN を活用した。

インスタンスセグメンテーション DNN は，3.2 で説明した物体検出 DNN で検出できる物体の位置だけでなく，その物体が撮影されている画素も取得する。図 14 左は，図 3 で検出された

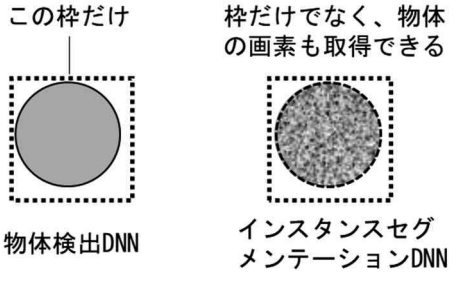

図 14　インスタンスセグメンテーション DNN

物体であるが，インスタンスセグメンテーションでは，図14右のようにその物体が撮影されている画素も取得する。

表 3 の②では，インスタンスセグメンテーション DNN で取得した部品の画素の重心をピッキング位置とした。部品の向きもこの画素から計算した。図15に取得結果を示す。

6.2 3次元への変換

図 16 が図 13 を撮影したときの 3 次元センサで取得した 3 次元点群データである。3 次元点群データのそれぞれの点は画像の明るさデータを持っている。表 3 の③では，この関係を利用して6.1で計算した2次元の位置や向きを，3次元に変換する。変換後のデータをロボットに送り部品がピッキングできることを確認した。

6.3 学習データの自動生成

表3の②で使用するインスタンスセグメンテーション DNN を使用するためには，物体検出DNNのときと同様に学習する必要がある。学習データとなる画像を作成するために，図17左の部品一つを撮影した画像と図17右のその画像の中のどこに部品が写っているかを示すマスク画像の二つを作成する。マスク画像では白い部分が部品の位置となる。この二つの画像から，4.3と同じような方法で図18のような学習用画像を自動生成した。この学習用画像で，図15の結果が得られるインスタンスセグメンテーションDNNを構築できた。

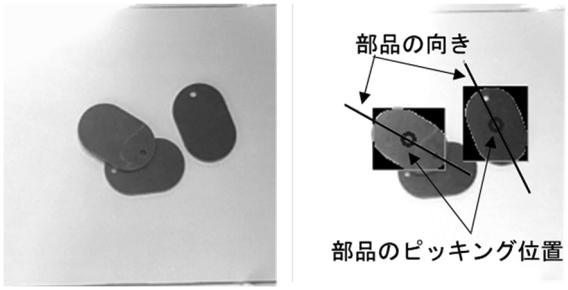

図 15 ピッキング位置と部品の向きの取得結果
（左：入力画像，右：取得結果）

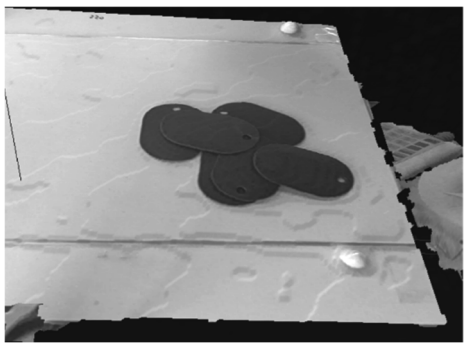

図 16 安価な 3 次元センサで取得した 3 次元データ

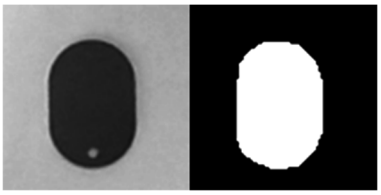

図 17 画像
（左：部品の画像，右：マスク画像）

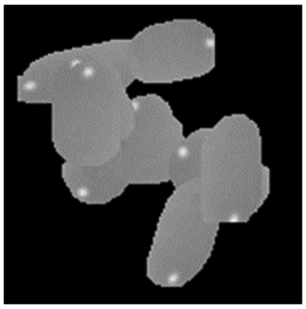

図 18 学習用画像

7. 結　　言

(1) ディープラーニングを利用したバラ積み部品のピッキングシステムを開発した。
(2) マウスクリックなどの簡単な操作でピッキングの設定ができるようにした。
(3) ピッキング後のロボットの動作を Python で記述できる環境を作成した。
(4) 安価な3次元センサを使ったピッキングシステムを作成した。