

# 合成データセットを用いたブラストマシンの未知ワークに対する 3D 姿勢推定方式 A 3D posture estimation method for unknown workpiece of blast machine using synthetic dataset

阿倍 博信<sup>†</sup> 松山 蓮<sup>†</sup>  
Hironobu Abe Ren Matsuyama

## 1. はじめに

近年、製造分野において、高度な計測・観測により収集されたデータを基に、大規模データ処理と現象のモデリングを通じて、実空間内の現象や人工物を仮想空間内で仮想的に再現・複製するシミュレーション技術であるデジタルツインが注目されている[1].

ブラストマシンは、ワークと呼ばれる金属部品の表面に付着している錆を機械的に除去する表面加工処理を行う工作機械であり、回転処理により投射室内に投入したワークを攪拌し、ワークに対して鉄球などの研削材を吹き付けることで表面加工処理を行う[2].

通常、ブラストマシンは閉空間で処理が行われるため、ワークの状態を外部から把握することは困難である上に、一般的にブラストマシンで扱うワークの種類は設計時に定義することができないため、ワーク毎にブラスト処理(本論文では、ワークの攪拌処理と投射処理を組み合わせてブラスト処理と定義する)に最適な動作パラメータを調整する際は、実機を用いた試行錯誤が必要となっていた。

そこで、我々は製造分野における工作機械の一つであるブラストマシンを対象として工作機械のデジタルツイン環境を構築に取り組んでいる。

本研究では、その一環として実施したデータ駆動アプローチに基づくブラスト処理のモデル化の取り組みについて報告する。具体的には、開発中のエプロン式のショット型ブラストマシンシミュレータ[3]を活用し、投射室のカメラ映像からワークの 3D 姿勢推定の機械学習に必要なデータを自動生成し、投射室画像からワークの 3D 姿勢推定を目的としたデータ駆動モデルを構築する。次に、構築したモデルを用いて未知のワークに対する 3D 姿勢推定精度についての有効性評価を行う。

以下、関連研究、合成データセットの生成、3D 姿勢推定モデルの構築、評価、考察を行い、最後にまとめを行う。

## 2. 関連研究

Adil らはデジタルツインの構成要素として、大きくデータ、モデル、インタフェースの 3 つの要素に分類しており、その中でもモデルは仮想空間における大規模データ処理と現象のモデル化・シミュレーションによる分析・解析・知能化を可能とする要素であり、最も重要な技術領域とされている[4].

デジタルツインにおけるモデルの課題の一つとしてサイバー空間でのシミュレーション結果に基づく物理ベースモデリングと実空間での計測結果であるデータ駆動型モデリングを組み合わせたハイブリッドモデリングの確立があげられる。

本研究では、モデルの課題に着目し、将来的に物理ベー

スモデリングとデータ駆動型モデリングの融合によるハイブリッド解析・モデリングの確立を目的として、データ駆動モデリングに基づくブラスト処理モデルの構築に取り組む。

また、工作機械を対象としたデジタルツインやシミュレータの研究は、NC 工作機械を対象とした旋削加工や切削加工を対象とした取り組みが一般的であり[5][6][7][8]、本研究で対象とする表面加工処理の一方式であるショットブラストを対象とした取り組みは存在しない。

3D ゲームエンジンを用いた機械学習用の合成データセットの生成に関する研究は広く取り組まれており[9]、本研究でも 3D ゲームエンジンを用いたブラストマシンシミュレータ[3]を用いて機械学習用データを生成する方針とする。

## 3. 合成データセットの生成

### 3.1 ブラスト処理のモデリング

ブラスト処理のモデリングの際に、処理全体を(1)攪拌処理、(2)投射処理の 2 個に分割し、それぞれ設計していく方針とした。(1)攪拌処理では、3D スキャンして 3D 化したワークを投射室内に配置し、投射室を回転することで、ワークを攪拌する。(2)投射処理では、攪拌処理の実行中に投射室の上部に設置したカメラからワークの回転状況を把握する。

本研究では、この中から(1)攪拌処理に対するモデリングを対象として、ワークのモデリング、ブラストマシンシミュレータを用いた合成データセット生成についてそれぞれ説明する。

### 3.2 ワークのモデリング

ブラストマシンで実際に使用しているワークを 3D スキャンし、3D 姿勢推定用のマーカーを付与したものをブラスト処理のシミュレーション用のワークとして使用する。

以下に、実際のワークをシミュレータで使用するための流れについて説明する。

(Step1) 3D スキャン：まず、ブラストマシンで実際に使用しているワークを 3D スキャンし、モデリングソフトである Blender で扱うことができるデータ形式(OBJ 形式)に変換する。今回、3D スキャナとしては、Revopoint MINI 3D Scanner を使用した。

(Step2) モデルの加工：次に、Blender でメッシュに凹面が存在しないように 3D オブジェクトを分割する。前述のとおり、攪拌処理において、Unity の物理演算を用いるため、使用する 3D オブジェクトにはメッシュコライダーの衝突判定を設定する必要がある。しかし、メッシュコライダーは凹面に対して適用することができない。よって、凹面による 3D オブジェクトの分割により凸面のみの 3D オブジェクト群を作成し、一つ一つにメッシュコライダーを適用

<sup>†</sup> 東京電機大学 Tokyo Denki University

して一個の 3D オブジェクトとして構成し直すことでメッシュとの誤差が少ない衝突判定を設定することができる。

(Step3)マーカの付与：Step2 の結果に対し、3D 姿勢推定用マーカを付与する。今回、マーカとして OpenCV の ArUco ライブラリ[10]を採用した。このマーカは QR コードよりも画素数が低く、認識が容易なことが特徴である。ワーク毎に左右非対称のマーカを 2 個作成し、ワークの表裏にそれぞれ別のマーカを付与することで姿勢を判別できるようにした。

図 1 にマーカを付与したワークのイメージについて示す。また、シミュレータにおいて、ワークの仕様は実測値に応じて、直径：0.7m、質量：1.1kg に設定した。

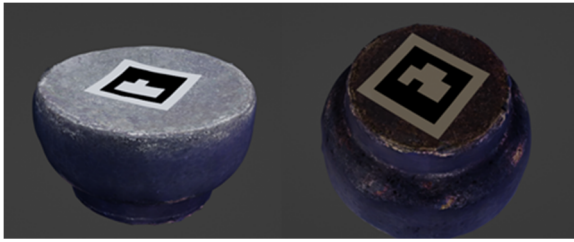


図 1 マーカを付与したワークのイメージ

### 3.3 ブラストマシンシミュレータを用いた合成データセットの生成

3D ゲームエンジンである Unity を用いて開発中のブラストマシンシミュレータ[3]にモデリングしたワークを読み込んで、実際に、ワークの攪拌処理を行い、投射ロカメラから投射室の映像を撮影するとともに、ワークの 3D 姿勢データを記録することで合成データセットの自動生成を行った。図 2 にブラストマシンシミュレータのイメージについて示す。

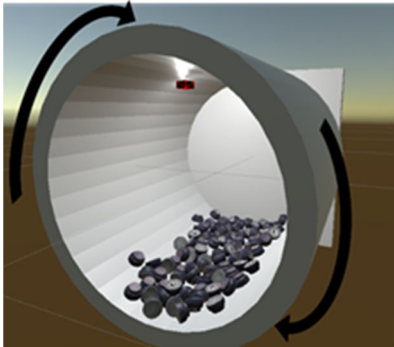


図 2 ブラストマシンシミュレータのイメージ

図 2 において、シミュレータにおけるブラストマシンの投射室は、回転運動によりワークを攪拌する筒型のモデルを採用し、その仕様としては、投射室の幅は 1.0m、直径は 0.9m に設定した。

図 3 に、ワークを 300 個設置した場合の、投射ロカメラから撮影したブラストマシンの投射室内画像(合成データセット)のイメージについて示す。投射ロカメラに採用したカメラのセンサーサイズと焦点距離は GoPro HERO11 Black[11]のデータに準拠する形とした。

合成データセットの生成において、まず、マーカを付与していない通常のワークとマーカを付与したワークをそれぞれブラストマシンシミュレータに読み込んで、ブラスト処理のシミュレーションを実行する。次に、指定時間ごと



図 3 合成データセットのイメージ

に、投射ロカメラから投射室内の画像を撮影して画像データとして記録するとともに、画像データに登場するマーカ付きワークの X,Y,Z 座標と回転(クォータニオン)を 3D 姿勢データとして取得して、CSV 形式のメタデータとして記録することで、合成データの自動生成機能を実現した。

図 4 に 3D 姿勢メタデータの概要について示す。

	A	B	C	D	E	F	G	H	I
1	file_name	time	r_x	r_y	r_z	r_w	p_x	p_y	p_z
2	0	0	-0.58356	-0.02214	0.280699	0.761691	-0.00854	-0.41738	0.016792
3	1	0.02	-0.57309	-0.02866	0.277351	0.770602	-0.00846	-0.41794	0.012872
4	2	0.04	-0.5663	-0.02893	0.273214	0.777706	-0.00821	-0.41819	0.00857
5	3	0.06	-0.56061	-0.0202	0.269599	0.782705	-0.00786	-0.41845	0.00413
6	4	0.08	-0.55367	-0.01504	0.261195	0.790569	-0.00725	-0.41872	-0.00019
7	5	0.1	-0.54538	-0.00472	0.259438	0.797014	-0.00684	-0.41894	-0.00373

図 4 3D 姿勢メタデータの構成

図 4 において、3D 姿勢メタデータの構成は、

- ・ A 列：画像名
- ・ B 列：撮影時刻 (1 枚目を 0 秒とする)
- ・ C~F 列：ワークのクォータニオン
- ・ G~I 列：ワークの位置座標 (投射室中央からの距離

(m)

である。

また、データ取得効率化のため、画像の記録時に各画像には映るマーカ付きワークは 1 個のみとし、他のマーカ付きワークは映らないようにマーカ付きワークを削除する処理を入れることにした。この処理を追加することで、例えば、投射室内に 20 個のマーカ付きワークが存在する場合、1 回の撮影により 20 通りの画像の生成が可能となる(図 5)。

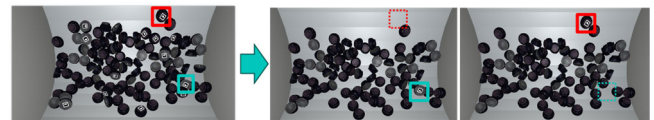


図 5 データ取得効率化の概要

以上の条件の元に、20 種類のマーカ付きワークを準備し、下記の条件にて合成データセットの生成を行った。

- (1) データセット A：データセット総数：300 個
- (2) データセット B：データセット総数：100 個

また、記録時の画像サイズは 1920 ドット×1080 ドットの条件にて撮影を行った。

以下、表 1 に今回生成した合成データセットの詳細について示す。

その結果、データセット A では、54,000(20×270×10)枚、データセット B では、60,000(20×60×50)枚の画像を準備することができた。

表 1 生成した合成データセットの詳細

	マーカ付き ワーク数	マーカ無し ワーク数	撮影時間 (s)	撮影間隔 (枚/s)
A	20	280	270	10
B	20	80	60	50

## 4. 3D 姿勢推定モデルの構築

### 4.1 合成データセットの前処理

3D 姿勢推定モデル構築に向けた合成データセットの前処理として、下記の手順を実施した。

(Step1) マーカの認識：合成データセットで撮影された画像データから OpenCV を用いてマーカ認識を行い、マーカが認識できた場合は、マーカ ID とマーカ位置を画像上の X, Y 座標で取得して 3D 姿勢メタデータに追加した(図 6)。

メタデータ CSV に追加

	A	B	C	D	E	F	G	H	I	J	K	L
1	file_name	time	r_x	r_y	r_z	r_w	p_x	p_y	p_z	clip_x	clip_y	aruco_id
2	6	0.12	-0.53286	0.003337	0.248767	0.808806	-0.00592	-0.41931	-0.00773	947.25	415.75	37
3	8	0.16	-0.51579	-0.00146	0.231947	0.82472	-0.00483	-0.41923	-0.01696	948.75	424	37
4	39	0.78	-0.15622	0.099582	0.347408	0.919231	0.004283	-0.39943	-0.1352	952.75	528	37
5	40	0.8	-0.14297	0.099505	0.347906	0.921483	0.004797	-0.3983	-0.13806	953.75	530.75	37
6	41	0.82	-0.14552	0.096746	0.348217	0.920982	0.005131	-0.39714	-0.14135	953.75	533.75	37
7	42	0.84	-0.14204	0.091382	0.353851	0.919926	0.006362	-0.39611	-0.14387	955	536.5	37

図 6 前処理後のメタデータの概要

(Step2) 画像データのトリミング：次に、画像データのグレースケール化を行い、その後、画像データをマーカ付きワークが映るように 150 ドット×150 ドットにトリミングを行った。

(Step3) 不要データの削除：最後に、マーカが認識できなかった画像データは画像とメタデータの削除を行った。

Step1~3 まで処理を行った結果、データセット A が、20,897 枚、データセット B が、40,231 枚となった。

### 4.2 学習

機械学習フレームワークとして PyTorch を採用し、前処理後のデータセットを用いた CNN モデルと MLP モデルを機械学習により、画像データからマーカの 3D 姿勢推定を行うモデルを構築した。

入力データは前処理後の画像データ(150 ドット×150 ドット)、ArUco によるマーカ ID、マーカの画像データ上の X, Y 座標とした。出力データは、3D 姿勢メタデータのワークのクォータニオン 4 個、と位置座標 X, Y, Z の計 7 個とした。また、共通項目としては、評価関数は MSE(平均二乗誤差)を採用し、エポック数は 30 とした。

#### 4.2.1 CNN モデルによる学習

CNN モデルに基づき、データセット A, B を用いて 3D 姿勢推定を行う機械学習モデルの構築を行った。図 7 に CNN モデルの構造について示す。

CNNモデル	[in, out]
Conv2d: 1-1	[[1, 150, 150], [64, 150, 150], kernel_size=[5, 5], stride=[1, 1]]
Dropout2d: 1-2	[p=0.25, inplace=False]
MaxPool2d: 1-3	[kernel_size=[2, 2], stride=[2, 2], padding=0, dilation=1]
Conv2d: 1-4	[[64, 150, 150], [128, 75, 75], kernel_size=[3, 3], stride=[1, 1]]
MaxPool2d: 1-5	[kernel_size=[2, 2], stride=[2, 2], padding=0, dilation=1]
Conv2d: 1-6	[[128, 75, 75], [256, 37, 37], kernel_size=[2, 2], stride=[1, 1]]
MaxPool2d: 1-7	[kernel_size=[2, 2], stride=[2, 2], padding=0, dilation=1]
Conv2d: 1-8	[[256, 37, 37], [512, 18, 18], kernel_size=[2, 2], stride=[1, 1]]
Dropout2d: 1-9	[p=0.5, inplace=False]
Flatten: 1-10	[[512, 18, 18], [165888]]
*torch.cat([165888, 3], 1)	[165888, 3], [165891]
Linear: 1-12	[165891, 7]

図 7 構築した CNN モデルの構造

学習を行う際には、ハイパーパラメータとして下記の項目を変更する形で学習を行った。

- 学習率：0.0002 から 0.00005 刻みで 0.0007 まで変更した。

- バッチサイズ：大きくするとメモリ不足になってしまったため、64 と 128 の 2 パターンで学習した。

その結果、データセット A, B について、最も誤差が小さくなったモデルについて表 2 に示す。

表 2 CNN モデルでの学習結果

	学習率	バッチサイズ	MSE
A	0.00055	128	0.04386
B	0.00045	128	0.05543

表 2 において、データセット A で最も誤差が少なかったモデルを CNN\_A、データセット B で最も誤差が少なかったモデルを CNN\_B とした。

#### 4.2.2 MLP モデルによる学習

CNN モデルと同様、MLP モデルに基づき、3D 姿勢推定を行う機械学習モデルの構築を行った。図 8 に MLP モデルの構造について示す。

MLPモデル	[in, out]
Linear: 1-1	[150*150+3 = 22503, 4000]
Linear: 1-2	[4000, 3000]
Linear: 1-3	[3000, 2000]
Dropout: 1-4	[p=0.5, inplace=False]
Linear: 1-5	[2000, 1000]
Linear: 1-6	[1000, 200]
Linear: 1-7	[200, 7]

図 8 構築した MLP モデルの構造

学習を行う際には、ハイパーパラメータとして下記の項目を変更する形で学習を行った。

- 学習率：0.0002 から 0.00005 刻みで 0.0007 まで変更した。

- バッチサイズ：64, 128, 256, 512 の 4 パターンで学習した。

その結果、データセット A, B について、最も誤差が小さくなったモデルについて表 3 に示す。

表 3 MLP モデルでの学習結果

	学習率	バッチサイズ	MSE
A	0.00020	256	0.05063
B	0.00055	256	0.03061

表 3 において、データセット A で最も誤差が少なかったモデルを MLP\_A、データセット B で最も誤差が少なかったモデルを MLP\_B とした。

## 5. 評価

### 5.1 基本評価

前章にて構築した 3D 姿勢推定モデル(CNN, MLP)の基本評価を実施した。方針としては、表 4 に示した方針に基づき評価用データセットを構築した。

表 4 評価用データセットの構築

	マーカ付き ワーク数	マーカ無し ワーク数	撮影時間 (s)	撮影間隔 (枚/s)
$\alpha$	1	299	50	10
$\beta$	1	99	10	50

データセット  $\alpha$ ,  $\beta$  ともそれぞれ 500 枚の画像を生成した。

次に、前章にて構築した CNN\_A, CNN\_B, MLP\_A, MLP\_B に対して、評価用データセット  $\alpha$ ,  $\beta$  を用いて 3D 姿勢推定の性能評価を行った。評価関数としては、モデル構築時と同様 MSE を使用した。評価結果を表 5 に示す。

表 5 基本評価結果

	MLP_A	MLP_B	CNN_A	CNN_B
$\alpha$	0.2010	0.1816	0.2303	<b>0.1416</b>
$\beta$	0.2616	0.1981	0.2237	<b>0.1544</b>

表 5 を確認した結果、データセット  $\alpha$ ,  $\beta$  とも CNN\_B を使用して構築したモデルが最も良い性能を得ていることが分かった。

## 5.2 未知ワークに対する評価

続いて、構築したモデルのロバスト性評価として、未知のワークが入力された場合の 3D 姿勢推定について性能評価を行った。

今回使用した未知ワークの 3D モデルの概要について図 9 に示す。

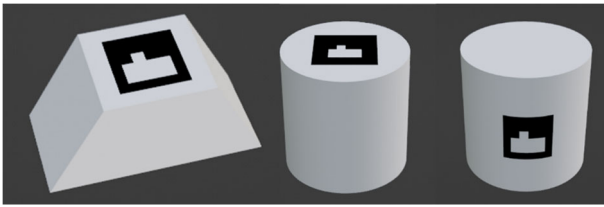


図 9 未知ワークの 3D モデルの概要

図 9 において、マークは同じものを使用する方針とし、左から台形、パイプ(上下)、パイプ(側面)として 3D モデルを準備した。

基本評価時に使用した表 4 のデータを参考にして、 $\alpha 1$  として 300 個のワークを、 $\beta 1$  として 100 個のワークを準備し、シミュレータを用いてデータセット  $\alpha 1$ ,  $\beta 1$  ともそれぞれ 500 枚の画像を生成した。

次に、各ワークについて 5 回のデータを撮影し、その結果として MSE の平均値を算出することで評価を行った。

表 6 に未知のワークに対する評価結果について示す。

表 6 未知のワークに対する評価結果

		MLP_A	MLP_B	CNN_A	CNN_B	平均
台形	$\alpha 1$	0.1921	0.1744	0.2283	<b>0.1681</b>	0.1907
	$\beta 1$	0.2080	0.1882	0.2000	<b>0.1771</b>	0.1933
パイプ 上下	$\alpha 1$	0.1937	0.1744	0.2277	<b>0.1585</b>	0.1885
	$\beta 1$	0.1869	<b>0.1753</b>	0.2344	<b>0.1753</b>	0.1929
パイプ 側面	$\alpha 1$	0.1854	0.1785	0.2247	<b>0.1691</b>	0.1894
	$\beta 1$	0.1905	<b>0.1766</b>	0.2571	0.1813	0.2013
平均		0.1927	0.1779	0.2287	<b>0.1715</b>	

表 6 を確認した結果、未知ワークの形状に関係なく、データセット  $\alpha 1$ ,  $\beta 1$  とも CNN\_B を使用して構築したモデルが全体的に良い性能を得ていることが分かった。

## 6. 考察

表 5, 表 6 について考察を行った結果、CNN\_B が最も性能が高く、表 5 で 0.14~0.15, 表 6 で 0.15~0.18 という結果が得られた。また、表 6 において、未知のモデルの形状の違いに対しても性能差が認められなかったため、現行方式でワークの形状に対する 3D 姿勢推定機能のロバスト性について有意性があると考えられる。

また、基本評価と未知ワークに対する評価の共通点として、 $\alpha$  (300 個) と  $\beta$  (100 個) の両方のデータセットで類似した性能が示されている。わずかではあるが、データセット  $\alpha$  の方がやや優れた結果を示していることが分かった。

また、MLP と CNN の両方のモデルにおいて、どちらも B モデルの方が良好な結果が得られた。これは、データセット B では、A と比較してワーク数が 1/3, FPS が 5 倍だったため結果的にマーカの認識率が高くなり、学習データ量が 1.68 倍に増えたことが原因と考えられる。

今回、最も優れた結果を示したのは CNN\_B であった。しかし、CNN\_A では MLP\_A と比較して性能の低下傾向が認められるため、継続した調査が必要である。

## 7. おわりに

本研究では、プラストマシンを対象としたデジタルツイン環境構築におけるデータ駆動アプローチに基づくプラスト処理のモデル化への取り組みとして、エブロン式のショット型プラストマシンシミュレータを活用して、投射室のカメラ映像からワークの 3D 姿勢推定の機械学習に必要なデータセットの自動生成を行うとともに、機械学習に基づき 3D 姿勢推定を行うモデルを構築した。その後、構築したモデルを用いて未知ワークに対する 3D 姿勢推定精度について有効性評価を行った。

実際には、生成した合成データセットを用いて、MLP と CNN の 2 種類のモデルを構築した。その結果、CNN のデータセット B で学習したモデルの 3D 姿勢推定精度が最良の結果となった。

## 参考文献

- [1] 科学技術振興機構, “デジタルツインに関する国内外の研究開発動向”, <https://www.jst.go.jp/crds/report/CRDS-FY2021-RR-09.html> (2022).
- [2] 田中和明, “最新金属の基本がわかる事典”, 秀和システム[第 2 版] (2021).
- [3] 阿倍博信, 他 1 名, “3D ゲームエンジンを用いたショット型プラストマシンシミュレータ”, 情報処理学会マルチメディア、分散、協調とモバイルシンポジウム DICOMO2024, DS-4 (2024).
- [4] Adil Rasheed et. al, “Digital Twin: Values, Challenges and Enablers From a Modeling Perspective”, IEEE Access, Volume 8, pp.21980 – 22012 (2020).
- [5] 伊東隆充, 他 1 名, “工作機械の衝突時の挙動と負荷軽減処理の効果確認”, 精密工学会学術講演会講演論文集, 2023 年度精密工学会春季大会, pp.78-79 (2023).
- [6] 藤田智哉, 他 5 名, “Model-Based Design を用いた工作機械の設計・制御最適化手法の開発 (第一報)”, 精密工学会学術講演会講演論文集, 2022 年度精密工学会春季大会, pp.542-543 (2022).
- [7] DMG 森精機, “工作機械のデジタルツインを活用した「デジタルツインテストカット」を開始”, [https://www.dmgmori.co.jp/corporate/news/pdf/20210216\\_testcut.pdf](https://www.dmgmori.co.jp/corporate/news/pdf/20210216_testcut.pdf) (2021).
- [8] 東俊博, 他 2 名, “製造業のスマートファクトリー化を支援するシミュレータ” NC Virtual Simulator “”, 三菱電機技報, Volume 96, No.4, pp.188-191 (2022).
- [9] 秋田悠河, 他 2 名, “合成データセットを用いた”, 情報処理学会第 30 回デジタルコンテンツクリエイション研究会, Vol.202 2-DCC-30, No.1, pp.1-7 (2022).
- [10] Open CV, “ArUco”, [https://dps.opencv.org/4.x/d9/d6d/tutorial\\_table\\_of\\_content\\_aruco.html](https://dps.opencv.org/4.x/d9/d6d/tutorial_table_of_content_aruco.html) (2024).
- [11] Go Pro Inc., “GoPro HERO11 Black”, <https://gopro.com/ja/jp/shop/cameras/hero11-black/CHDHX-111master.html> (2024).