工場内の台車に取り付けた ARマーカ認識システムの提案

橘川 拓実† 高橋 正裕† モロ アレサンドロ† 原田 佳周‡
西川 英雄‡ 野口 稔‡ 濱谷 章史‡ 梅田 和昇†
†中央大学 ‡日立ハイテクソリューションズ
E-mail: kitsukawa@sensor.mech.chuo-u.ac.jp

1 序論

近年，製造業などの工場内でも IoT の導入が進められている．それに伴い，様々な作業の自動化が進められている．しかし，多品種少量生産を行っている工場では，Fig.1 のような台車を利用して，部品を一つ一つ棚からとり，加工する場所まで運び加工して組み立てるといった作業を行っている．これらの作業を自動化することは難しく，多くの工程を人手で行っている．このような台車を利用した工場では，生産ラインの可視化ができておらず，主に二つの解決すべき課題があると考えられる．

一つ目の課題は，作業者一人一人の作業工程や進捗の把握である．工場内の作業員の作業内容や作業にかかる時間を把握できておらず，その作業にかける人員数が適正人数なのかがわからない．また，個人の作業時間が数値化できていないため，作業が的確に行われているのかがわからないという問題もある．このような問題により，作業員による進捗のばらつきが発生し，工場全体の作業効率が悪くなる．

二つ目の課題は，台車や部品の所在把握である．台車は工場や倉庫内に点在し，希望の台車を探すのに手間がかかる．また，一般的に台車は工場内でデザインが統一されており，台車に乗せた部品も似ているものが多く，台車の取り間違いが起こる可能性もある．このような問題点を踏まえ台車や部品の所在把握は重要な課題である

解決方法としては，IC チップなど電波を発するものを台車に取り付け，台車と作業員の位置を把握することできると考えられる [2]．しかし，この解決策は導入にコストと手間がかかるという問題点がある．

そこで, コストのかからない AR マーカを台車に取り付け，それを工場内の定点カメラで撮影し読み取ることで工場内の台車の位置を把握するという方法が効果的であると考えられる．

工場内台車に AR マーカを取り付けることを想定すると，マーカが作業の邪魔にならないよう小さいサイズのマーカを利用することになる．しかしそうすると認識性能は低下するため，実際に利用するためにはマーカの性能を向上させる必要がある．

AR マーカの認識性能を向上させるために，従来の AR マーカのデザインを改良したり，新たなデザインのマーカの提案が行われている [4][5]．これらのマーカは認識精度は向上しているが，認識可能距離の伸びは小さい．また，従来のマーカと比較してマーカサイズの拡大や作成可能なマーカの ID 数の減少といった問題もある．

そこで，本研究では既存の AR マーカを利用し，物体検出やブレ除去などの手法を組み合わせることで認識性能を向上させることを目的とする．既存の AR マーカには Fig.2 に示す ArUco マーカ [3] を利用する．提案システムについて述べ，実験により有効性を検証する．

Fig. 1 工場で利用される台車 [1]

Fig. 2 ArUco マーカ [3]

2 提案システム

2.1 認識システムの概要

提案システムの流れを Fig.3 に示す．取得した画像から深層学習を用いた手法でマーカ検出し，マーカ周辺を切り出す．その切り出し画像に対して前処理を行い，AR マーカが認識しやすいようにする．切り出した画像を OpenCV のライブラリ ArUco に認識させる [6]．この手法を用いることでマーカの認識可能距離が長くなることが期待できる．

また，工場内の台車に AR マーカを取り付けるということは，マーカ自体が動いており取得画像にブレが生じることが考えられる．そこで，深層学習を用いたブレ除去処理を加えることでブレ画像にも対応したシステムを構築する．

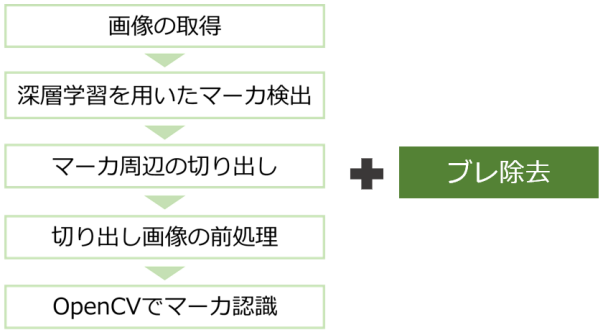

Fig. 3　提案システムの流れ

2.2 深層学習を用いたマーカ検出

深層学習を用いたマーカ検出には物体検出手法である Faster R-CNN[7] を利用する．Faster R-CNN は，ある矩形の中身が物体なのか背景なのかを識別し，検出された領域をクラス分類するというモデル構造になっている．R-CNN[8] や Fast R-CNN[9] といった従来の物体検出手法と比べて，物体領域の候補の抽出の際に RPN(Region Proposal Network)[10] と呼ばれる CNN 構造を利用しているので，処理時間が大幅に向上している．この Faster R-CNN の学習済みモデルをマーカ用に作成したカスタムデータセットでファインチューニングすることでマーカ検出を可能にする．

マーカ検出器を作成するためのカスタムデータセットの作成について述べる．データセットを作成する際，効率よく大量の教師データを作成するためにマスク単位の物体追跡手法である SiamMask[11] を利用する．動画の最初のフレームに追跡するべき物体の位置を与えれば，それ以降のすべてのフレームで対象の位置を推定する．この方法は，他の深層学習を用いた物体追跡手法に比べて高速であり，リアルタイムでの運用が可能である．

2.3 切り出し画像の前処理

マーカ周辺の切り出しについて述べる．2.2 節で述べたマーカ検出に成功し 4 点の座標が取得出来たら，入力画像からその周辺を切り出す．次に切り出し画像のサイズの正規化を行う．こうすることで，カメラとマーカの距離が近くても遠くても同じサイズの切り出し画像を生成することができる．

切り出し画像に対し，鮮鋭化処理を行うことでマーカの認識精度が向上することが考えられる．Fig.4 に切り出し画像に 8 近傍鮮鋭化フィルタで鮮鋭化処理をしたものを示す．元画像よりマーカの明暗がはっきりとし，形状が読み取りやすくなっているのが確認できる．しかし，元画像に含まれるノイズも鮮鋭化されてしまっているので白と黒の境界線がはっきりしない部分もある．

そこで，切り出し画像にまず平滑化処理を行い，その後鮮鋭化処理を行う．実際に処理したものを Fig.5 に示す．平滑化処理には 3×3 のガウシアンフィルタを利用している．Fig.4 に示す鮮鋭化処理のみのものと比較すると，ノイズがなくなり明暗もはっきりとして，マーカの形状がより読み取りやすくなっているのが確認できる．

これらの結果より，平滑化処理と鮮鋭化処理を組み合わせたものを切り出し画像の前処理として利用する．

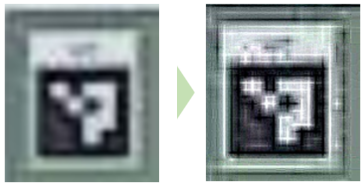

Fig. 4　鮮鋭化処理のみ

Fig. 5　平滑化処理後に鮮鋭化処理

2.4 深層学習を用いたブレ除去

深層学習を用いたブレ除去には敵対的生成ネットワーク (GAN:Generative Adversarial Networks)[12] をブレ除去適応させた手法である DeblurGAN-v2[13] を利用する．

DeblurGAN-v2 のモデル構造は，Generator では高次元特徴マップをアップサンプリングしたものと低次

元特徴マップを足し合わせながら生成画像を作成しており，Discriminator では Patch GAN を導入し，生成画像をパッチレベルで分割して本物か偽物か判別する．こうすることで，画像内の細かい部分のブレまで着目したモデルとなっている．また，学習データにないブレ画像に対してもブレが除去が可能である．

3 実験

3.1 マーカ検出器の作成実験

2.2 節で述べたマーカ検出器を作成をする際，工場内で利用する ID の形状すべてを学習させて検出器を作成することは困難である．そこで，学習させるマーカの種類の数を変えて検出器を作成することでどのくらいの種類のマーカを学習する必要があるかを評価した．

3.1.1 実験条件

カメラはアイ・オー・データ機器社のネットワークカメラ Qwatch（クウォッチ）の TS-WRLP[15] を利用し，画像を取得した．AR マーカのサイズはすべて 3[cm]×3[cm] のものを利用し，マーカの ID が一目でわかるようにマーカの上に ID を表示しているものを利用した．

検出器は COCO dataset[16] で事前学習済みの Faster R-CNN モデルをファインチューニングすることで作成した．作成には物体検出アルゴリズムを実装している深層学習ライブラリ Detectron2[17] を利用した．

学習には ID=00 から 15 までの 16 種類と ID=20，30，40，50 の 4 種類を利用した．テストには ID=00，01，98，99 の 4 種類を利用した．ID=00，01 は教師データに含まれているが，ID=98，99 は教師データに含まれていない ID である．

学習に利用したマーカの小さい数字から，1，2，3，4，5，10，15，20 種類のマーカを学習させた検出器をそれぞれ作成した．

教師データの作成には 2.2 節で述べた SiamMask を利用した．学習とテストに使用した動画はどちらも解像度が 1280×720，マーカが約 0.3[m] から 3[m] の距離を前後に移動している動画で，室内の同じカメラ位置で撮影されたものである．学習に使用した動画は 3 分で約 5200 フレーム，テストに使用した動画は 30 秒で約 900 フレームである．

学習させるときのハイパーパラメータは，learning rate を 0.005，epoch 数を 300，batch size を 64 に固定してそれぞれの検出器を作成した．1，2，3，4，5，10，15，20 種類のマーカを学習させた計 8 個のマーカ検出器を ID=00，01，98，99 の ID のテスト動画でマーカ検出させ検出成功率を求めた．

3.1.2 実験結果と考察

Table 1 に学習させたマーカの種類の数による検出成功率を示す．縦軸が学習させたマーカの種類の数，横軸がテストに利用したマーカの ID を示す．また，この結果を縦軸が検出成功率，横軸が学習させたマーカの種類の数でグラフ化したものを Fig.6 に示す．学習させるマーカの種類が少なくても精度の高い検出器を作成することができているときもある．また，15 種類 20 種類と種類を増やせば検出率が上がり，学習させていないマーカーの検出がより安定している．

ID=98，99 のマーカは教師データに含まれていないにもかかわらず，ID=00，01 と同様に高い検出率であった．これは，Faster R-CNN の事前学習済みモデルを利用していることにより，少ない偏った教師データでも精度の高い検出器が作成できているためと考えられる．

Table 1　学習させた種類の数による成功率

種類の数	id=00	id=01	id=98	id=99
1	0.89	0.82	0.96	0.79
2	0.98	1.00	0.94	0.96
3	0.70	0.82	0.62	0.72
4	0.97	0.92	0.81	0.99
5	0.98	1.00	0.99	0.99
10	0.87	0.94	0.93	0.95
15	0.97	0.98	0.98	0.92
20	1.00	1.00	1.00	1.00

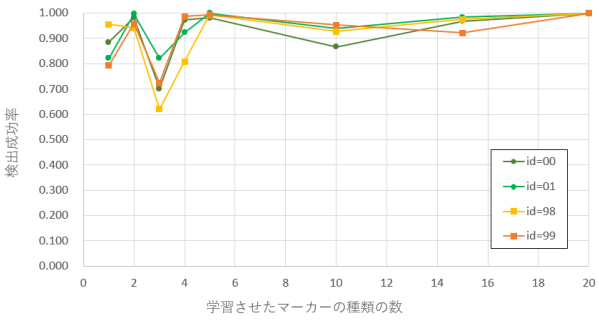

Fig. 6　マーカの種類の数と成功率の関係

3.2 提案システムの認識可能距離に対する実験

2.1 節の Fig.3 の左に示した，マーカ検出後にマーカ周辺を切り出し，前処理して OpenCV で認識させるという提案手法で認識可能距離が伸びることを検証した．

3.2.1 実験条件

カメラと AR マーカは 3.1.1 項と同じものを利用し，マーカの ID は 0 から 9 の 10 種類を用いた．

入力画像の解像度は 720×400 と 1280×720 で実験を行った．距離は 0.5[m] から 0.1[m] ずつカメラから離して撮影した．各距離で 10 フレーム画像を取得し認識成功率を求めた．

マーカ検出には 2.2 節で述べたマーカ検出器を利用した．検出器は SiamMask アノテーションツールを用いて作成した合計約 12000 フレームのカスタムデータセットでファインチューニングして作成した．

切り出し画像の前処理は，2.3 節で述べた平滑化処理後に鮮鋭化処理を行う手法を利用した．検出したマーカ周辺の切り出し後の平滑化処理は 2.3 節で述べた 3×3 のガウシアンフィルタ，鮮鋭化処理には 8 近傍鮮鋭化フィルタを利用した．

3.2.2　実験結果と考察

Fig.7 と Fig.8 に解像度 720×400 と 1280×720 の実験結果を示す．各距離でのマーカ検出の成功率も図中に示している．

720×400 では 0.5[m] から 0.8[m] と 0.17[m] では AR マーカ認識手法単体よりも認識成功率が低くなったがそれ以外の距離では認識成功率が向上している．認識不可となった距離は 0.23[m] から 0.30[m] まで伸びている．

1280×720 では 0.27[m] まで認識成功率が 80 %を超えており，各距離で認識成功率が向上している．AR マーカ認識手法単体では認識不可であった 3.0[m] でも認識成功率が 66 %となった．また，認識不可となった距離は 3.0[m] から 4.0[m] まで伸びている．

以上の結果より，深層学習を用いたマーカ検出，周辺の切り出しと前処理を組み合わせることで AR マーカの認識可能距離が伸びることが確認できた．

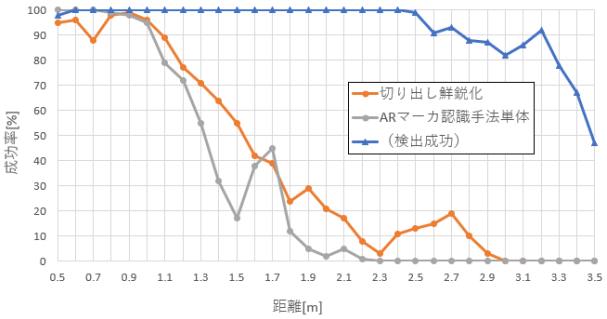

Fig. 7　720×400 の距離と認識成功率の関係

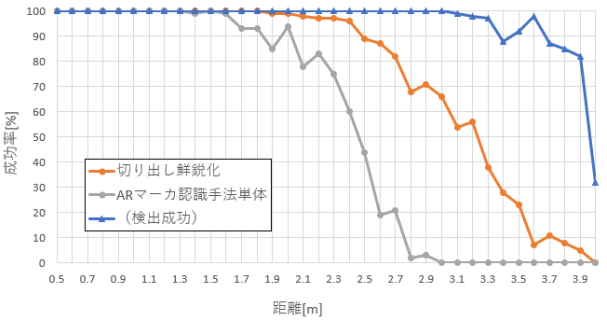

Fig. 8　1280×720 の距離と認識成功率の関係

3.3　提案システムのブレ画像に対する実験

本実験では AR マーカを一定速度で横にスライドさせることでマーカにブレを発生させた．得られたブレ画像に対しブレ除去を含めた提案手法で認識性能が向上することを検証した．

3.3.1　実験概要

実験の目的は二つある．一つ目はマーカがブレた画像に対し深層学習を用いたブレ除去処理を行うことでブレ除去処理の有効性を評価する．二つ目は切り出し処理，前処理である平滑化・鮮鋭化処理をブレ除去と組み合わせた時の認識性能を評価する．

また，ブレ除去処理を行う箇所は，入力画像全体に対してとマーカ検出後の切り出し画像に対しての二か所が考えられる．以上を踏まえて本実験で比較する手法の流れを Fig.9 に示す．図中の番号は各手法の番号を示し，それぞれ手法 1，手法 2，手法 3，手法 4，手法 5，手法 6 と呼ぶ．

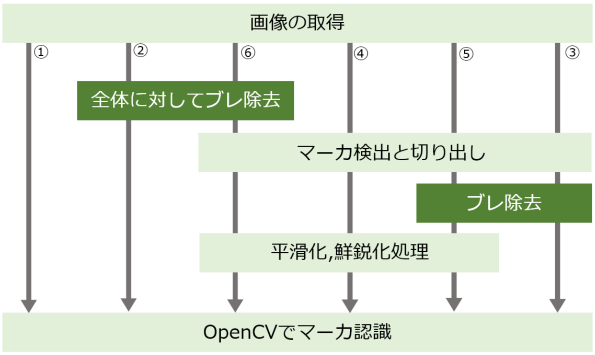

Fig. 9　比較する手法の流れ

3.3.2　実験条件

カメラと AR マーカは 3.2.1 項と同じものを利用した．

画像の解像度は 1280×720 で，距離は 1m と 2m で実験を行った．速度は 0.05[m/s]，0.10[m/s]，0.15[m/s] と 0.05[m/s] ずつ上げていき，各速度で 10 フレームずつ画像を取得して認識成功率を求めた．

マーカ検出器，前処理である平滑化・鮮鋭化処理は 3.2.1 項と同じ条件とした．ブレ除去処理には 2.4 節で述べた DeblurGAN-v2 の学習済みモデルを利用した．

3.3.1 項で述べた実験の目的より，まず手法 1，2，3 を比較しブレ除去処理の有効性を評価した．この実験の結果を 3.3.3 項のブレ除去実験の評価で示す．次に手法 4，5，6 を比較し，切り出し処理，平滑化・鮮鋭化処理をブレ除去と組み合わせた時の認識性能を評価した．この実験の結果を 3.3.4 項の組み合わせ実験の評価で示す．

3.3.3　ブレ除去の有効性の評価

Fig.10 と Fig.11 に距離 1[m] と 2[m] での速度と認識成功率の関係を示す．1[m]，2[m] ともにブレ除去処理を利用することで認識成功率が向上している．

距離 1[m] では，入力画像全体にブレ除去処理をする手法 2 のほうが，切り出し後にブレ除去処理をする手法 3 より認識性能が高くなった．一方，距離 2[m] では手法 3 のほうが手法 4 より認識性能が高くなった．これは，ブレ除去処理に利用している DeblurGAN-v2 の特性によるものだと考えられる．

Fig.12 に実際のブレ除去画像を示す．図中の上の 2 枚は手法 3 を用いて切り出し後にブレ除去を行った結果である．図中の下の画像は手法 2 を用いて全体ブレ除去をしたものである．これを手法 3 の結果と比較するために拡大した．比べるとどちらもブレ除去ができているが，全体ブレ除去をしているもののほうが鮮明に処理できている．

以上の結果から，ブレ画像に対するブレ除去処理の有効性が確認できた．

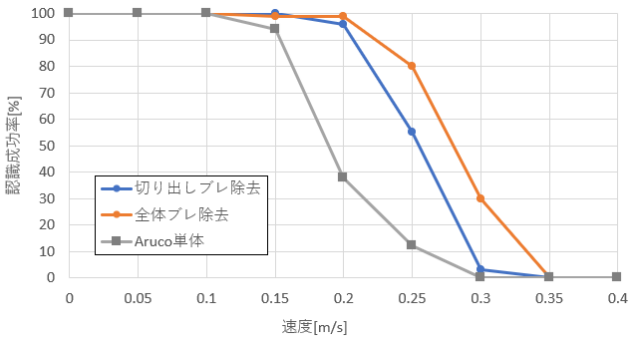

Fig. 10　距離 1[m] での速度と認識成功率の関係

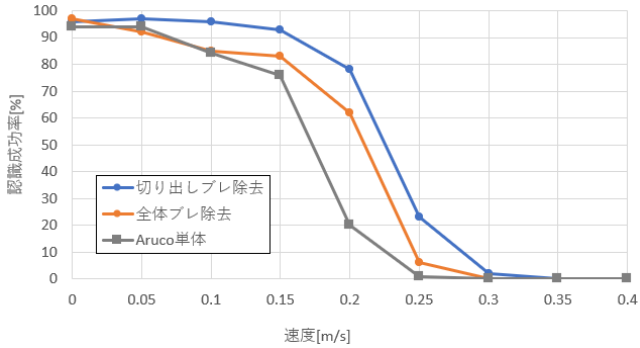

Fig. 11　距離 2[m] での速度と認識成功率の関係

Fig. 12　実際のブレ除去画像　上：切り出し後にブレ除去，下：全体にブレ除去

3.3.4　組み合わせの有効性の評価

Fig.13 と Fig.15 に距離 1[m] と 2[m] での速度と認識成功率の関係を示す．切り出し後に平滑化・鮮鋭化処理を行う手法 4 と比較して，切り出し後にブレ除去処理を加える手法 5 と全体にブレ除去処理を加える手法 6 は認識性能が向上している．距離 1[m] では，手法 6 のほうが手法 5 より認識性能が高くなった．一方，距離 2[m] では手法 5 のほうが手法 6 より認識性能が高くなった．

Fig.14 に距離 1[m]，速度 0.25[m/s] での実際の画像を示す．左から元画像，手法 4，手法 5，手法 6 の処理後の結果を示している．同様に，Fig.16 に距離 2[m]，速度 0.25[m/s] での実際の画像を示す．距離 1[m]，2[m] ともに，ブレ除去処理をしていない手法 4 はブレ部分も平滑化・鮮鋭化処理されており，読み取りづらい．一方，ブレ除去処理をしてから平滑化・鮮鋭化処理を行う手法 5 と手法 6 はマーカがより読み取りやすくなっている．手法 5 と手法 6 を画像で比較すると距離 1[m] では手法 6 のほうが手法 5 よりきれいに処理されている．距離 2[m] はどちらも同様の結果となっている．

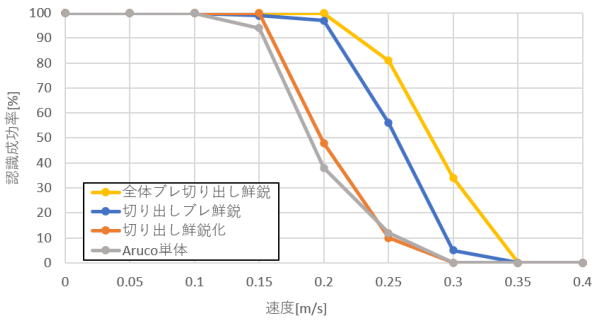

Fig. 13　距離 1[m] での速度と認識成功率の関係

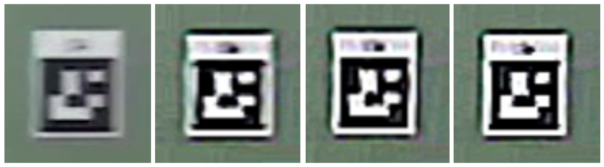

Fig. 14　距離 1[m] での実画像比較　左から (1) 元画像，(2) 手法 4，(3) 手法 5，(4) 手法 6

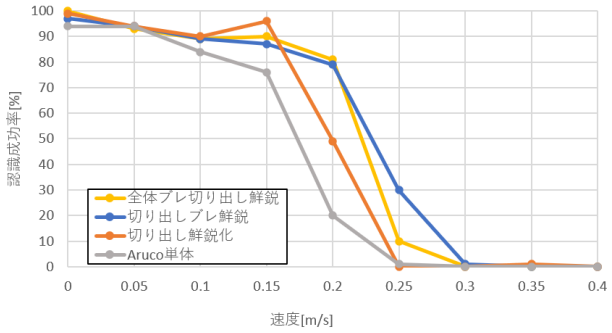

Fig. 15　距離 2[m] での速度と認識成功率の関係

Fig. 16　距離 2[m] での実画像比較　左から (1) 元画像，(2) 手法 4，(3) 手法 5，(4) 手法 6

3.3.5　処理速度の評価

1280×720 の画像を用いて手法 1 から手法 6 の処理速度を計測した．Table 2 に各手法の処理速度を示す．手法 3 と手法 5 は切り出し後にブレ除去処理を行うため検出できたマーカ数によって処理速度が変わる．検出できたマーカ数が増えるほど処理時間がかかり，マーカ数が 0 個なら 4.0[fps]，1 個なら 2.9[fps]，2 個なら 2.3[fps] となった．手法 6 は入力画像全体に対してブレ除去処理した後マーカ検出を行うため処理時間がかかると考えられるが，処理速度は 2.2[fps] となった．

全体的に数 fps の処理速度という結果になり，オンラインでの利用が可能な処理速度であることが確認できた．

Table 2　各手法の処理速度

	手法1	手法2	手法3	手法4	手法5	手法6
処理速度	100fps	4.6fps	4.0fps 2.9fps(1個) 2.3fps(2個)	4.0fps	4.0fps 2.9fps(1個) 2.3fps(2個)	2.2fps

3.4　実際の工場内での動作検証実験

工場内で実際に使用している台車に AR マーカを取り付けて動画を取得し，提案システムの有効性を検証した．

3.4.1　実験条件

カメラと AR マーカは 3.2.1 項と同じものを利用した．画像の解像度は 1920×1080 で動画を取得した．動画のフレームレートは 30[fps] である．

Fig.17 に実験環境の平面図を示す．Fig.18 に台車に取り付ける AR マーカの ID と位置を示す．図中の番号は AR マーカの ID を示し，台車の支柱各 4 箇所に貼り付けた．Fig.19 にカメラの設置部分の現場写真を示す．カメラは柱に取り付けてあり，地面と水平方向に向けた．また，カメラを取り付ける高さは台車に付けた AR マーカと同じにした．

台車に異なる動きをさせた 2 パターンの動画，動画 1 と動画 2 を撮影し実験を行った．動画 1 は，通路からカメラ前の青く囲まれたピット内に台車を侵入させカメラに近づけ，近づけ終えたら台車を後ろに引いて通路まで戻し通路を通過させた．動画 2 は，通路を通過する際，ピットに侵入させずそのまま通り過ぎ，通り過ぎた後反対方向から折り返してもう一度カメラ前の通路をそのまま通過させた．

この動画 1 と動画 2 に対して AR マーカ認識手法を単体で利用する手法 1 と提案システムである手法 5,6 でマーカを認識させ，マーカの認識成功数を求めた．なお，本実験はオフラインで行った．

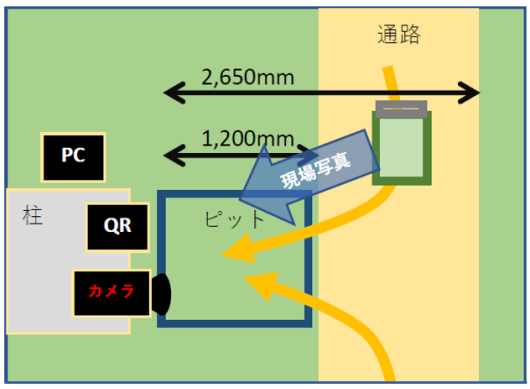

Fig. 17　実験環境の平面図

3.4.2　実験結果と考察

Table 3 に動画 1,2 における手法 1,5,6 の認識成功数を示す．

動画 1 では AR マーカ認識手法単体である手法 1 より提案システムである手法 5 と手法 6 で認識成功数が

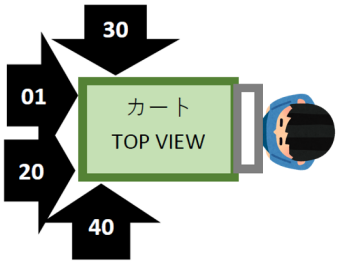

Fig. 18　台車に取り付ける AR マーカの ID

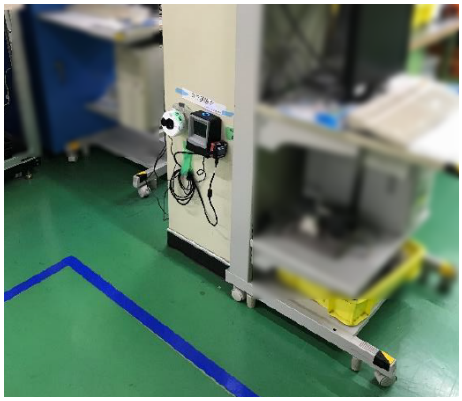

Fig. 19　設置カメラの現場写真

増加している．手法 5 と手法 6 を比較すると取得画像全体に対してブレ除去を行う手法 6 のほうが認識成功数が増加した．

Fig.20 に手法 6 での実験において認識に成功している様子を示す．黄色のバウンディングボックスがマーカの検出成功を示し，認識できた ID を青文字で示している．

動画 2 では AR マーカ認識手法単体と提案システムどちらでもマーカ認識ができなかった．これは台車の移動速度が速すぎてブレが除去しきれなかったことが原因だと考えられる．しかし，ID の認識には至らなかったが本手法を用いることでマーカ検出には成功しているフレームが複数確認できた．よって提案システムのブレ除去処理とマーカ検出は動画 2 に対しても効果があると言える．

以上の結果より，実際の工場内の台車に AR マーカを取り付けた動画に対しても提案システムは有効であることを検証できた．

Table 3　各手法の認識成功数

	手法1	手法5	手法6
動画1	608	689	708
動画2	0	0	0

Fig. 20　認識に成功している様子（動画 1, 手法 6）

4　結論

本研究では，工場内の台車に取り付けた AR マーカの認識システムを提案した．具体的には AR マーカ認識手法に深層学習を用いたマーカの検出と切り出し処理，ブレ除去処理を加えた手法を提案した．また，AR マーカの認識手法単体での認識性能の評価実験を行い，その後マーカとカメラの距離に対する実験とマーカを動かす速度に対する実験で本システムの有効性を示した．

具体的には，距離に対する実験では，AR マーカの認識可能距離が解像度 1280×720 では 2.9[m] から 3.9[m] に，解像度 720×400 では 2.2[m] から 2.9[m] に伸びた．速度に対する実験では，距離 1[m]，速度 0.25[m/s] の実験で認識成功率が 0.12 から 0.81 に向上した．

処理速度は 2〜4[fps] であり，オンライン環境での実際の利用が可能な処理速度であることが確認できた．さらに，工場で実際の利用を想定した実験を行い提案システムの有効性を検証した．

今後の課題としては，処理速度の向上が挙げられる．3.3.5 項の結果よりオンラインでの実装が可能な処理速度ではあるが，システム全体の認識性能を向上させることを考えると処理速度の改善は重要である．そこで，ブレ除去とマーカ検出を別で学習させるのではなく，一つにして学習させることで処理速度が改善できると考えられる．

参考文献

[1] 株式会社サカエ: ”サカエ総合カタログ 2021”, https://skebook.com/sakae_webcatalog2021/book/#target/page_no=367.

[2] リコー: ”屋内位置情報サービス（製造向け）”, https://www.ricoh.co.jp/sensing/factory/.

[3] S. Garrido-Jurado, R. Muoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marn-Jimnez. ”Automatic generation and detection of highly reliable fiducial

markers under occlusion", Pattern Recognition, pp.2280-2292, 2014.

[4] Y. Wang, Z. Zheng, Z. Su, G. Yang, Z. Wang, Y. Luo, "An Improved ArUco Marker for Monocular Vision Ranging", Chinese Control And Decision Conference (CCDC), 2020.

[5] 石井裕剛，藤野秀則，卞志強，関山友輝，中井俊憲，下田宏："拡張現実感用広域トラッキングシステムの開発"，ヒューマンインタフェースシンポジウム 2006 論文集，pp.387-392，2006.

[6] OpenCV: "Detection of ArUco Marker", https://docs.opencv.org/master/d5/dae/tutorial_aruco_detection.html.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Neural Information Processing Systems (NIPS), 2015.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," Computer Vision and Pattern Recognition (CVPR), 2014.

[9] Girshick, Ross. "Fast R-CNN," International Conference on Computer Vision (ICCV), 2015.

[10] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. "High performance visual tracking with siamese region proposal network," Computer Vision and Pattern Recognition, 2018.

[11] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H.S. Torr. "Fast Online Object Tracking and Segmentation: A Unifying Approach," Computer Vision and Pattern Recognition (CVPR), 2019.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative Adversarial Networks," arXiv:1406.2661 [stat.ML], 2014.

[13] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-v2: Deblurring (Orders-of-Magnitude) Faster and Better", International Conference on Computer Vision (ICCV), 2019.

[14] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. "DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks," Computer Vision and Pattern Recognition (CVPR), 2018.

[15] 株式会社アイ・オー・データ機器： "ネットワークカメラ Qwatch （クウォッチ） TS-WRLP", https://www.iodata.jp/product/lancam/lancam/ts-wrlp/index.html.

[16] Tsung-Yi Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollr. "Microsoft COCO: Common Objects in Context," Computer Vision and Pattern Recognition (CVPR), 2015.

[17] Facebook AI: "Detectron2", https://ai.facebook.com/tools/detectron2/.