

# オブジェクト分類を利用した単眼カメラ画像からのクレーン周辺の三次元地図高速作成

繁森 央一<sup>1</sup>・須崎 純一<sup>2</sup>・松島 格也<sup>2</sup>・小谷 仁務<sup>2</sup>  
小林 知生<sup>1</sup>・米田 瑞生<sup>3</sup>・Marek Ososinski<sup>3</sup>

<sup>1</sup>学生会員 京都大学大学院 工学研究科 (〒 615-8540 京都市西京区京都大学桂)  
E-mail: shigemori.hisakazu.82x@st.kyoto-u.ac.jp (Corresponding Author)

E-mail: kobayashi.tomoki.28s@st.kyoto-u.ac.jp

<sup>2</sup>正会員 京都大学大学院 工学研究科 (〒 615-8540 京都市西京区京都大学桂)  
E-mail: {susaki.junichi.3r;matsushima.kakuya.7u;kotani.hitomu.5c} @kyoto-u.ac.jp

<sup>3</sup>非会員 株式会社タダノ 技術研究所 (〒 761-0301 香川県高松市林町 2217-13)  
E-mail: {mizuki.yoneda;marek.ososinski} @tadano.com

昨今では建設現場における人手不足が深刻化し、労働生産性の向上が課題であり、その解決策の一つにクレーンの自動操縦化が挙げられている。実現の際には低コストで実装できることが望ましく、そのためにはクレーンブーム先端に取り付けた単眼カメラで動画像を取得して、クレーン周辺の三次元地図を精確かつ高速に作成する手法の実現が期待されている。しかし、その条件下で撮影した動画像にはクレーンのフックが映り込むことが多く、三次元地図の精度を低下させる要因の一つに挙げられる。そこで本研究では、上記条件下におけるステレオマッチングや視差画像を用いた三次元地図の生成精度向上手法の開発に併せて、動画像中に映るクレーンフック等の三次元地図生成の障壁となりうる対象物を画像上から除去する手法の開発を試みた。

**Key Words:** 3D-mapping, Depth map, Optical flow, Object classification, Computer vision

## 1. 背景と目的

昨今の建設現場では人手不足が問題となっており、労働生産性の向上が喫緊の課題となっている。そこでその解決策の一つとしてクレーン操縦の簡略化あるいは自動化が挙げられている。クレーンの自動操縦にはクレーンの吊荷の経路決定の際に必要なクレーン周辺の三次元地図の生成が必要とされている。現状では、クレーンブーム先端に鉛直下向きに取り付けられた単眼カメラから取得した動画像を用いてクレーン周辺の三次元地図を生成する手法が製造コストおよび実装上の観点から最も望ましいと考えられている。

前述の条件で撮影された動画像を用いた三次元地図生成にはさらに、1) 動画像上の物体から静止物体のみを抽出し三次元地図生成に用いること、2) 地図生成は高速に行われる必要があること、の二つの要件を満たす必要がある。

一つ目の要件について、三次元地図生成には静止物体から得られた特徴点を用いることで画像間のホモグラフィを計算する必要がある。しかし、ブーム先端に取り付けられたカメラから得られる画像には揺動するクレーンフックが常に映り続けるという特徴がある。そのため

クレーンフックから取得された特徴点を用いてホモグラフィが計算されると、取得された三次元地図生成の精度が著しく損なわれてしまう。また、クレーンアームの伸び次第では画像中にアームと共に旋回する操縦室が映るため、上記と同様の理由で三次元地図精度が損なわれる。そのため、静止物体とその他の物体を判別することが要求される。更に、仮にフックと操縦室のうち、フックの判別ができれば、それをフックの位置の推定に用いることができる。また後述する要件 2 に同じく、物体の判別手法も高速である必要があるため、計算コストの大きな Semantic Segmentation<sup>1)</sup> のような判別よりも計算コストの小さな特徴点ベースで判別が行われることが望ましい。

二つ目の要件について、建設現場の状況は刻一刻と変化する。そのため地図生成の高速性も要求される。高速性に長けた三次元地図生成手法には、Simultaneous Localization And Mapping (SLAM)<sup>2)</sup> が知られる。SLAM アルゴリズムの一種である Oriented FAST and Rotated BRIEF-SLAM2 (ORB-SLAM2)<sup>3)</sup> や Oriented FAST and Rotated BRIEF-SLAM3 (ORB-SLAM3)<sup>4)</sup> を用いた既存の研究では、カメラがブームと共に操縦室を中心とした旋回運動をするという条件で地図生成が試みられた。し

かし、航空写真測量やドローン測量で用いられる直線軌道上を動くカメラから撮影される動画像とは異なり、円軌道で動くカメラから撮影される動画像による ORB-SLAM2 での三次元地図生成は得られる点数や安定性に限界があることが判明している。

そこで上記二つの要件を考慮し本研究では、1) 動画像から静止物体、フック、操縦室の特徴点を判別する手法、2) 近いフレーム間で視差画像を作成する Semi-Global Matching (SGM) を用い、作成された視差画像同士を適切に連結することで高速な三次元復元を行う手法、の開発を試みる。

## 2. 使用データ

今回、物体判別手法の実験には実際にクレーン実機から取得した動画像を用い、三次元復元の実験には Gazebo<sup>5)</sup> シミュレータで取得した動画像を使用する。シミュレータ動画像では地物の大きさやテクスチャ、配置などを任意に調節できるため三次元復元手法の各種精度評価にはシミュレータ動画像が適していると考えたからである。また、クレーン実機およびシミュレータ動画の解像度、fps、動画時間の情報を以下の表-1 に示す。

表-1 クレーン実機とシミュレータによる動画情報

動画	解像度 (pix)	fps	フレーム (枚)
クレーン実機	1280 × 720	30	1028
シミュレータ	1280 × 720	30	432

## 3. 特徴点の分類

この章では、1. で述べた動画中から静止物体、フック、操縦室をオプティカルフローを用いて分類する手法について述べる。

### (1) オプティカルフローを用いた分類手法

1. で述べたように、高速な物体判別を実現するために、本研究の物体の分類手法にはオプティカルフローを用いる。オプティカルフローとは、カメラの隣接フレーム間での物体の動きを追跡する手法であり、画像から抽出した特徴点およびその周辺の画素情報をもとに特徴点を追跡する。この手法は画像の一部画素のみを扱うため、計算コストが低い利点がある。代表的な手法には Lucas-Kanade 法<sup>6)</sup> があり、本研究ではこの手法を用いる。

### (2) オプティカルフローの特徴

使用データから抽出される特徴点の対象物には、フック、静止物体、操縦室の三種類が存在する。また、各対象物のオプティカルフローにはそれぞれ異なった次の特徴がある。まずフックのオプティカルフローはフックが揺動するため、左右方向に細かく動く軌跡を描く。次に静止物体のオプティカルフローはクレーンの旋回に合わせて、画像下側を回転中心とした円弧の軌跡を描く。最後に操縦室のオプティカルフローは、操縦室がクレーン旋回と連動して動くため画像上では固定点として軌跡が描かれる。

上記の特徴に加え、オプティカルフロー自体の特徴として追跡対象物に他の物体が重なった場合、誤追跡が生じるなどの特徴もある。そのため今回の動画では揺動するフックが静止物体と重なることによる誤追跡や操縦室が静止物体と重なることによる誤追跡が観測されている。今回は前者をフック、後者を操縦室の分類に含めることとする。以上より、本研究ではこれらの3種類の対象物から得られたオプティカルフローを以下の図-1 のように分類する手法を提案する。

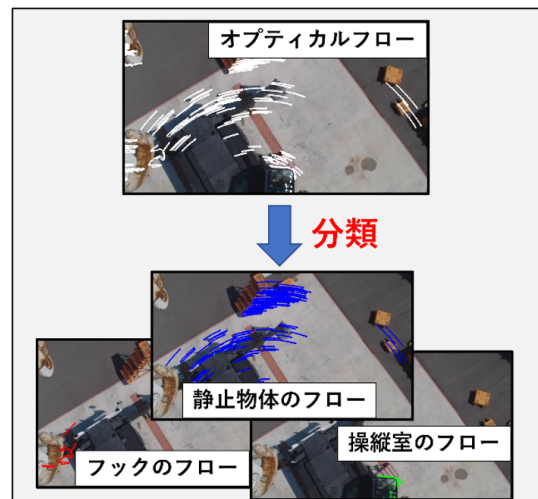


図-1 分類のイメージ

## 4. 特徴点分類の提案手法

### (1) 概要

本研究ではオプティカルフローを用いたフック、静止物体、操縦室の判別手法として、図-2 の分類フローを提案し、次節から各過程について説明する。

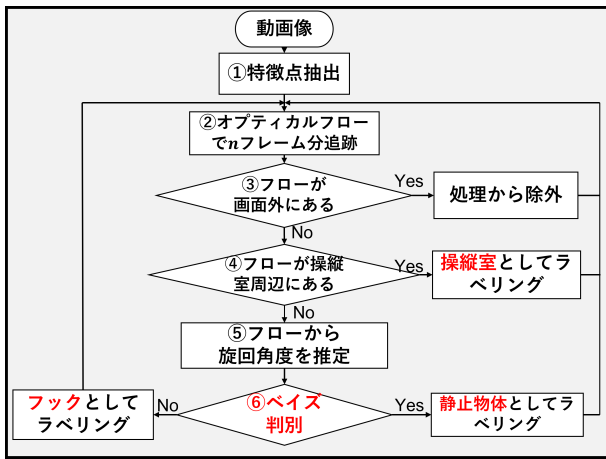


図-2 分類提案手法

(2) 特徴点抽出

画像中からオブティカルフローの初期点である特徴点を抽出する。

(3) オブティカルフロー n フレームの軌跡を取得

オブティカルフローの軌跡を利用して特徴点の分類を行うために、ある程度長い軌跡が情報として必要と考えた。そのため動画を n フレームごとに分けて分類を実行することで n フレーム分のオブティカルフローを分類に利用する。

(4) 画面外のオブティカルフローを除外

特徴点の追跡時に、追跡対象物が画面外へ出るとオブティカルフローも画面外へと出てしまう。この場合、ホモグラフィの計算にこの特徴点を使うことができないため、あらかじめ分類処理から除外する。

(5) 操縦室周辺のオブティカルフローを分類

今回使用した動画像には、画像の下端中央部にアームの旋回と連動して動く操縦室が常に映り込んでいる。しかし操縦室は画像内の特定の領域にしか存在しないため、Region of Interest (ROI) を画像下部に設定し、ROI 内の領域とオブティカルフローが被っているものに関してこれを操縦室の特徴点としてラベリングする。

(6) オブティカルフローから旋回角度を推定

ここまでで未分類のオブティカルフローにはフック、静止物体それぞれから得られたものが含まれる。そこでこれらを分類するために各オブティカルフローによって旋回中心を推定し、それを用いてクレーンの旋回角度を推定する。ここで旋回中心とは、クレーンの旋回運動の中心を指し、旋回中心は以下の手順で求める。

1. n フレーム分のオブティカルフローを、 $\frac{n}{2}$  フレーム目を境に二等分する。
2. 分割したオブティカルフローそれぞれの端点同士で線分を作り、その線分に対する垂直二等分線を引く。
3. 最後に、引いた二本の垂直二等分線の交点を求め、その点を旋回中心の推定値とする。

(7) ベイズ判別

6 節で求めた静止物体、フックそれぞれの旋回角度の推定値を用いてベイズ判別<sup>7)</sup>を行う。n フレーム分を 1 セットとするオブティカルフローを用いて分類を行う際、前のセットで静止物体を追跡しているオブティカルフローは今のセットでも静止物体を追跡している可能性の方が高い。そこで本提案手法ではベイズ判別を用い、分類結果の事前確率を更新していくことで前のセットの分類結果を利用する。ベイズ判別に用いるデータには、以下の式 (1) のように CAN データから得られる旋回角度  $\theta_{CAN}$  と推定値  $\hat{\theta}$  との差分  $\varphi$  を用いる。ここで CAN データとはクレーンのセンサから出力されるクレーンの運動データであり、旋回角度の他にブーム長、ブーム起伏角度、フックワイヤー長などが出力される。

$$\varphi = \hat{\theta} - \theta_{CAN} \quad (1)$$

また今回の場合、各セット毎に動画中から複数のオブティカルフローが得られ、それぞれのオブティカルフローを静止物体 ( $\Pi_1$ ) とフック ( $\Pi_2$ ) に分類することを考える必要がある。そこで t セット目における、k 番目 ( $k = 1, 2, \dots, m$ ) のオブティカルフローから計算される旋回角度の推定値  $\hat{\theta}_t^{(k)}$  と CAN データ  $\theta_{CAN,t}$  との差分を  $\varphi_t^{(k)}$ 、そしてそのオブティカルフローの静止物体に対する事前確率を  $w_{1t}^{(k)}$ 、フックに対する事前確率を  $w_{2t}^{(k)}$  とする。このとき以下の式 (2) が成り立つ。

$$w_i'^{(k)}(\theta_i|\varphi_t^{(k)}) = \frac{w_i^{(k)}(\theta_t^{(k)}|\varphi_t^{(k)})l_i(\varphi_t^{(k)}|\theta_t^{(k)})}{\sum_{j=1}^2 w_j^{(k)}(\theta_t^{(k)}|\varphi_t^{(k)})l_j(\varphi_t^{(k)}|\theta_t^{(k)})} \quad (2)$$

ここで、 $w_i'^{(k)}(\theta_i|\varphi_t^{(k)})$  は t セット目における、k 番目のオブティカルフローの分布  $\Pi_i (i = 1, 2)$  に対する事後確率であり、 $l_i(\varphi|\theta)$  は分布  $\Pi_i$  の尤度である。この式 (2) により事後確率が事前確率と尤度関数で更新される。

更にここで観測値  $\varphi$  を  $\Pi_i$  に分類する行動を  $a_i$  とし、 $\Pi_i$  に分類されるデータ  $\theta_i$  を  $\Pi_j$  に分類した時のコスト  $L_{ij}$  をそれぞれ以下の表-2 のように設定する。

表-2 行動  $a$  によるコスト  $L$

$\theta \backslash a$	$a_1$	$a_2$
$\theta_1$	$L_{11}$	$L_{12}$
$\theta_2$	$L_{21}$	$L_{22}$

この時、静止物体 ( $\Pi_1$ ) に分類する行動  $a_1$  を選択した時の損失は以下のように計算される。

$$w_1 L_{11} + w_2 L_{12}$$

同様にフック ( $\Pi_2$ ) に分類する行動  $a_2$  を選択した時の損失は  $w_1 L_{21} + w_2 L_{22}$  と求まるため、以上より行動  $a_1$  と行動  $a_2$  との間に

$$w_1 L_{21} + w_2 L_{22} \geq w_1 L_{11} + w_2 L_{12} \quad (3)$$

の関係が成り立つとき、静止物体に分類する。そこで、式 (3) に式 (2) を代入すると、以下の式 (4) が導かれ

$$\begin{aligned} & w_1^{(k)} (\theta_t^{(k)} | \varphi_t^{(k)}) l_1(\varphi_t^{(k)} | \theta_t^{(k)}) L_{11} \\ & + w_2^{(k)} (\theta_t^{(k)} | \varphi_t^{(k)}) l_2(\varphi_t^{(k)} | \theta_t^{(k)}) L_{12} \\ & \geq w_1^{(k)} (\theta_t^{(k)} | \varphi_t^{(k)}) l_1(\varphi_t^{(k)} | \theta_t^{(k)}) L_{21} \\ & + w_2^{(k)} (\theta_t^{(k)} | \varphi_t^{(k)}) l_2(\varphi_t^{(k)} | \theta_t^{(k)}) L_{22} \end{aligned} \quad (4)$$

式 (4) の左辺と右辺の差分をとった以下の式 (5) 定式化される  $D_{12}^{(k)}$  を  $k$  番目のオプティカルフローに対する静止物体とフックとの間の分類の判別式とする。

$$\begin{aligned} D_{12}^{(k)} = & \{w_1^{(k)} (\theta_t^{(k)} | \varphi_t^{(k)}) l_1(\varphi_t^{(k)} | \theta_t^{(k)}) L_{11} \\ & + w_2^{(k)} (\theta_t^{(k)} | \varphi_t^{(k)}) l_2(\varphi_t^{(k)} | \theta_t^{(k)}) L_{12}\} \\ & - \{w_1^{(k)} (\theta_t^{(k)} | \varphi_t^{(k)}) l_1(\varphi_t^{(k)} | \theta_t^{(k)}) L_{21} \\ & + w_2^{(k)} (\theta_t^{(k)} | \varphi_t^{(k)}) l_2(\varphi_t^{(k)} | \theta_t^{(k)}) L_{22}\} \end{aligned} \quad (5)$$

すなわち、 $D_{12}^{(k)}$  が 0 以上であれば静止物体に分類し、0 より小さければフックに分類する。以上のベイズ判別の流れをまとめたものを以下の図-3 に示す。

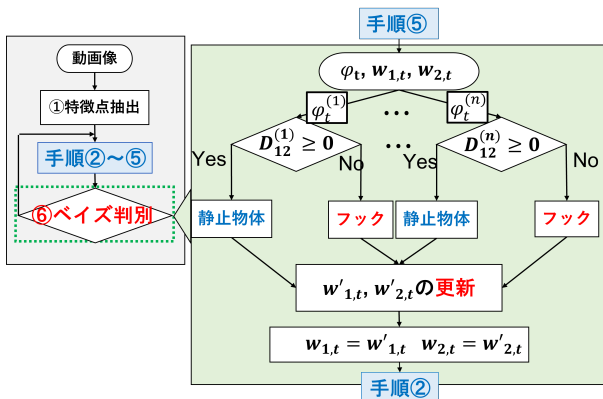


図-3 ベイズ判別

5. に、実際にこの手法をクレーン実機から取得した動画に適用した結果を示す。

## 5. 特徴点分類の結果およびその考察

### (1) 結果

4. で提案した手法を CPU: Intel(R) Core(TM) i7-9700 3.0 GHz, OS: Windows10 Pro, メモリ: 32GB の環境のもと python3.9.5 と opencv4.5.3 で実装した。なお、今回は 4.-(2) の特徴点抽出には Shi-Tomashi 法<sup>8)</sup> が実装された goodFeaturesToTrack 関数を用い、そのパラメータを maxCorners: 255, qualityLevel:0.1, minDistance: 7, blockSize: 7 と設定した。また、4.-(3) における  $n$  は 90 に設定した。そのため 1028 枚のフレームに対して、オプティカルフローは合計で 11 セット取得された。そして同じく 4.-(3) における特徴点の追跡手法には Lucas-Kanade 法<sup>6)</sup> が実装された calcOpticalFlowPyrLK 関数を用い、パラメータには winSize: (15,15), maxLevel: 4, criteria の終了条件として回数を 10, 精度を 0.03 と設定した。さらに、4.-(5) における操縦室の領域を左上頂点座標:(500,590), 右下頂点座標:(780,720) とする矩形に手で設定した。そして最後に各分類に対するコストは以下の表-3 のように設定し、各分布の尤度  $l_i(\varphi|\theta)$  は正解データをもとに図-4 と図-5 のように確率密度分布を作成し、分布系を仮定することで求めた。その結果静止物体は正規分布、フックは一様分布と仮定され、以下のように定まった。ここで、 $p(\cdot)$  は標準正規分布である。以上のように設定したときの結果の例として、2, 4 セット目の時の分類結果をそれぞれ図-6 と図-7 に示す。そして全体のセットにおける分類結果を表-4 に示す。今回の手法の処理速度は 96 fps となり、リアルタイムでの処理を実現した。

$$l_1(\varphi|\theta) = \frac{1}{4.32} p\left(\frac{\varphi - 1.81}{4.32}\right)$$

$$l_2(\varphi|\theta) = \frac{1}{360}$$

表-3 実験時のコスト  $L$

$\theta \backslash a$	$a_1$	$a_2$
$\theta_1$	0	1
$\theta_2$	1	0

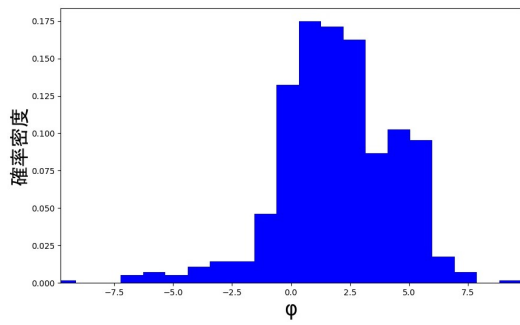


図-4 静止物体の確率密度分布

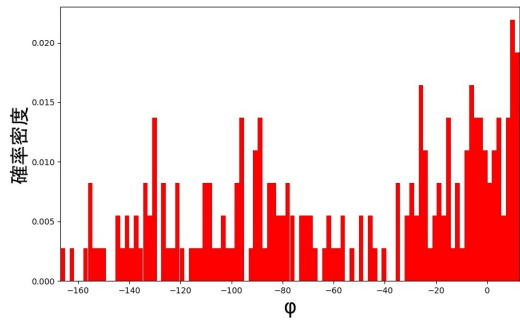


図-5 フックの確率密度分布

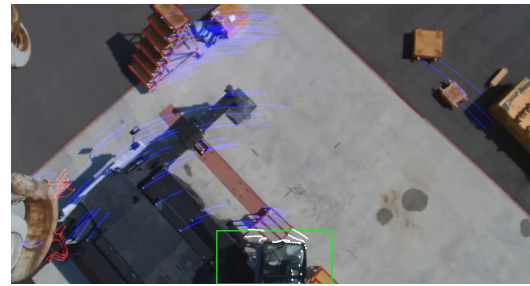


図-6 セット 2 の分類結果。  
青は静止物体, 赤はフック, 白は操縦室を表し  
緑色の矩形は今回設定した操縦室領域を指す。



図-7 セット 4 の分類結果

表-4 分類結果の表

		予測 (セット 1)			予測 (セット 2)			予測 (セット 3)			予測 (セット 4)		
		静止物体	フック	操縦室	静止物体	フック	操縦室	静止物体	フック	操縦室	静止物体	フック	操縦室
正解	静止物体	112	1	3	105	0	1	45	0	5	140	0	5
	フック	7	23	0	5	32	0	8	41	0	3	2	0
	操縦室	70	0	14	0	1	13	0	1	10	0	0	15
Precision		0.94	0.96	0.82	0.95	0.97	0.93	0.85	0.98	0.67	0.98	1.00	0.75
Recall		0.97	0.77	1.00	0.99	0.86	0.93	0.90	0.84	0.91	0.97	0.40	1.00
F1 Score		0.95	0.85	0.90	0.97	0.91	0.93	0.87	0.90	0.77	0.97	0.57	0.86
Accuracy		0.93			0.96			0.87			0.95		

		予測 (セット 5)			予測 (セット 6)			予測 (セット 7)			予測 (セット 8)		
		静止物体	フック	操縦室	静止物体	フック	操縦室	静止物体	フック	操縦室	静止物体	フック	操縦室
正解	静止物体	135	0	3	127	0	4	22	3	3	13	2	3
	フック	5	7	0	5	14	0	4	45	0	2	40	0
	操縦室	0	0	15	0	0	14	0	0	7	1	1	6
Precision		0.96	1.00	0.83	0.96	1.00	0.78	0.85	0.94	0.70	0.81	0.93	0.67
Recall		0.98	0.58	1.00	0.97	0.74	1.00	0.79	0.92	1.00	0.72	0.95	0.75
F1 Score		0.97	0.74	0.91	0.97	0.85	0.88	0.81	0.93	0.82	0.76	0.94	0.71
Accuracy		0.95			0.95			0.88			0.87		

		予測 (セット 9)			予測 (セット 10)			予測 (セット 11)		
		静止物体	フック	操縦室	静止物体	フック	操縦室	静止物体	フック	操縦室
正解	静止物体	7	3	2	0	10	0	0	13	0
	フック	0	29	0	0	29	0	0	30	0
	操縦室	0	0	6	0	2	6	0	3	6
Precision		1.00	0.91	0.75	-	0.71	1.00	-	0.65	1.00
Recall		0.58	1.00	1.00	0.00	1.00	0.75	0.00	1.00	0.67
F1 Score		0.74	0.95	0.86	-	0.83	0.86	-	0.79	0.80
Accuracy		0.89			0.74			0.69		

## (2) 精度評価

精度評価には各クラスの予測結果の正解との適合率を示す *Precision*, 各クラスの予測結果がどれくらい正解データを網羅できているかを示す *Recall*, そしてそれらの調和平均を示す *F1 Score* を用いた。

また、各クラスで得られるデータ数に差があることを考慮し、精度評価には多クラス分類における精度評価指標として *Accuracy* を用いた。また正解データには使用データから取得したオプティカルフローを手動で分類したものをを用いた。

## (3) 考察

結果では、1-9セットにおける *Accuracy* が 0.87 以上だったのに対し、クレーンがほぼ停止状態にあった 10-11セットにおける *Accuracy* は 0.74 以下となっており、精度に差がみられた。この結果からは、1-9セットにおいてはバイズ判別を用いた本提案手法の有効性が示唆されると考えられる。しかし、10-11セットにおいてはオプティカルフローを用いて旋回角を推定する手法が適用できなかったため分類精度が落ちてしまったと考えられ、クレーンが停止状態に移行したときの分類方法を考える必要がある。ただこの場合の分類に関しては CAN データを用いてクレーンの停止を検知できるため、その場合のみ事前確率を更新せず維持する仕組みを加えることで対処可能と考えており、今後の課題としたい。

他にも 1-8セットに見られたように、結果からは本来フックと分類されるべきものを静止物体へと分類してしまうケースが目立った。この結果は千次元地図生成の際に誤ってフックに特徴点を使用してしまう可能性が増えるということを示唆する。したがってこの誤分類を防ぐ必要性があり、分類コストを調整するといった対処法を考えているが、こちらも併せて今後の課題とする。

ここまでを特徴点の分類についての検討とし、6. から三次元復元手法について述べる。

## 6. 三次元復元手法

1. で述べたように、クレーン周辺の三次元地図生成は高速に行う必要がある。高速性に長けた三次元復元手法には、Simultaneous Localization And Mapping (SLAM) が知られる。代表的な SLAM アルゴリズムの一つである ORB-SLAM2 や ORB-SLAM3 を用いて、車載カメラで撮影された動画像から高速で地図を作成する事例<sup>9)</sup> など幅広く報告されている。

これまでに我々は、クレーンのブーム先端に真下に向けて取り付けられた単眼カメラで撮影された動画像に ORB-SLAM2 を適用し、三次元地図の作成を試みた。<sup>10)</sup> 既存研究でのカメラの位置は自由で不規則であり、また

航空写真測量やドローン測量では画像が重なるように直線を往復して撮影が行われる。しかし、クレーン先端に取り付けられたカメラは同じ半径の円軌道上を動く。この独特な軌跡のために ORB-SLAM2 での三次元地図作成には得られる点数や安定性に限界があることが判明している。

## (1) SGM を用いた分類

そこで本研究では近いフレーム間で視差画像を作成する Semi-Global Matching (SGM)<sup>11)</sup> の手法に注目し、作成された視差画像同士を適切に連結することで三次元復元を行った。ここで、視差画像とは、ステレオ平行化された 2 枚の左右画像において、左画像上の各画素に対応する右画像上の画素を検出し (ステレオマッチング)、それらに対応する画素の左右方向の座標のずれ (視差) を基にして、物体とカメラとの距離を色の濃淡などで表した画像である。用いる画像の全ての画素に対して視差値が計算されるため SLAM などを用いた手法に比べて密な点群が得られるというメリットがある。さらに、同じ高度から撮影された画像を基に生成された視差画像はお互いを連結しやすいため、今回の (地面にほぼ水平な) 円軌道からの動画像に適していると考えた。なお株式会社タダノによると、クレーンの自動操縦に必要とされる三次元地図の精度は 5-10 cm, 要求される処理速度は 10 分以内であるので、本研究ではこれを目標とした。

## 7. 三次元復元の提案手法

### (1) 概要

本研究では SGM を用いた三次元地図生成手法として、以下の図-8 の流れで地図生成を行うことを提案する。次節からこの提案手法について説明する。

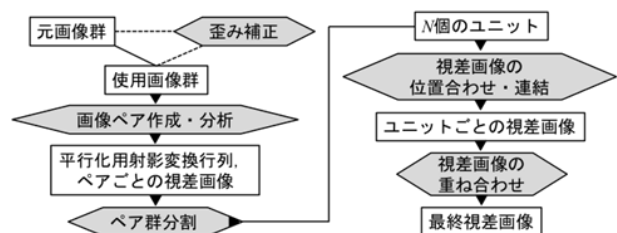


図-8 SGM 提案手法のフローチャート

### (2) 使用画像およびユニットの生成

まず、元動画像から 1 秒に  $n$  フレームの割合でフレームを抽出し元画像群とする。今回、レンズに歪みは与えなかったため、元画像をそのまま使用画像群とした。今回は、全画像から直接 1 枚の視差画像を生成するのではなく、まず全画像から複数枚の全体視差画像を生成し、

それらを各地点の視差値について平均値を取りながら重ね合わせるといった手法を用いた。複数枚の視差画像を重ね合わせる段階でノイズを除去できると考えたためである。本手法では、全画像を  $N$  個の小さな画像群（ユニットと定義）に分割して考えた。具体的には、全使用画像の通し番号を  $n$ 、ユニットの通し番号を  $m$  としたとき、 $n \equiv m(\text{mod}N)$  となるように各ユニットに使用画像群を分配した。

### (3) 画像ペアの作成・分析

視差画像を算出する際の入力画像である 2 枚の画像（時系列的に先の画像を左画像、後の画像を右画像とする）はお互いにステレオ平行化されている必要がある。ここで、ステレオ平行化とは、左画像上の任意の特徴点の  $y$  座標が、それに対応する右画像上の特徴点の  $y$  座標と等しくなるよう右画像を変換する処理のことである。この処理にはホモグラフィ行列による射影変換を利用した。今回扱う画像では、画像中での地面部分の占有面積が大きい。そのため、地面部分の対応点の座標を左右画像で一致させ、高さのある地物上の対応点には画像水平方向（ $x$  軸方向）に視差を生じさせるようなホモグラフィ行列が計算される。元の左画像と、算出したホモグラフィ行列を用いて右画像を射影変換した画像について、改めて対応点を検出および描画した結果の例が図-9 である。一連の処理によってペア画像がステレオ平行化されていることが分かる。

次に左右画像より視差画像を算出する。今回、視差画像を算出する手法としては、前述の SGM の上位互換である Semi-Global Block Matching (SGBM) を用いた。ペアごとに作成した視差画像の例が図-10 である。ここで、使用画像はある一点から撮影した中心投影の画像であり、それに伴って視差画像も中心投影の画像となっている。しかし、理想の最終視差画像は全ての地点を真上から見たような画像、すなわち正射投影の画像である。よって、視差画像同士の連結前に、各視差画像を中心投影から正射投影に変換する必要がある。再投影時に視差

値を画像中心に向けてずらす再投影幅  $x$  は以下の式 (6) で求まる。ただし、 $l$  は画像主点からの距離 [ $px$ ]、 $d$  は視差 [ $px$ ]、 $b'$  は主点間距離 [ $px$ ] である。図-10 で示した中心投影の視差画像について、各視差値を再投影した正射投影の視差画像が図-11 である。

$$x = \frac{dl}{b' + d} \quad (6)$$

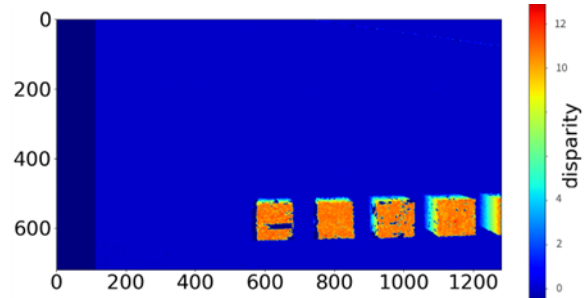


図-10 SGBM による視差画像の例

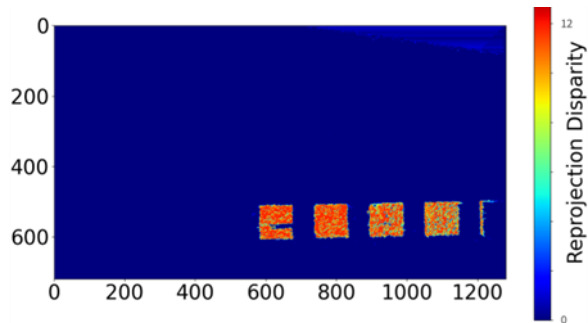


図-11 再投影した視差画像の例

### (4) 視差画像の連結

各視差画像を連結するためには、それぞれの視差画像の位置合わせを行う必要がある。これにはペア画像の平行化時に算出したペア画像間のホモグラフィ行列を用いた。この変換行列によって高さが 0 の地点に関し

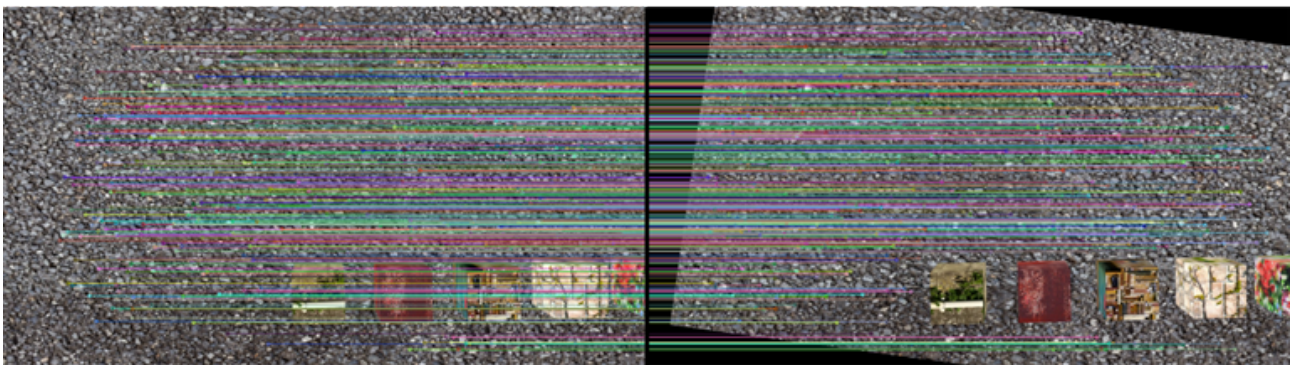


図-9 ステレオ平行化後のペア画像対応点描画の例

ては左右画像における y 座標が同じで、かつ視差が 0 となるように変換されていることを考えれば、このホモグラフィ行列は視差画像同士の位置合わせに利用できる。これを利用してユニット内の全ての視差画像をそのユニットの先頭の視差画像に位置合わせした。

次に、位置合わせした視差画像を重ね合わせてユニットごとに 1 枚の視差画像を算出する。このとき視差画像同士が複数枚重なる地点の画素については複数の視差の候補値が存在することになるが、閾値を設けながら平均値を計算し、当該画素の視差値とした。

最後に、ユニットごとの視差画像を  $N$  枚重ね合わせることで最終視差画像を算出した。この処理により、ユニットごとの視差画像を算出した際に閾値と平均値によって取り除けなかったノイズを取り除く、あるいは軽減することができると考えた。8. からシミュレータを用いて取得した動画に本提案手法を適用した結果を示す。

## 8. 提案手法による三次元復元の結果及びその考察

### (1) 結果

ここでは元動画画像 1 秒あたり 5 枚のフレームを使用し、それらを 5 つのユニットに分割した際の結果を示す。提案手法にて算出した最終視差画像を図-12 に示す。この視差画像の作成に要した時間は約 379 秒であった。

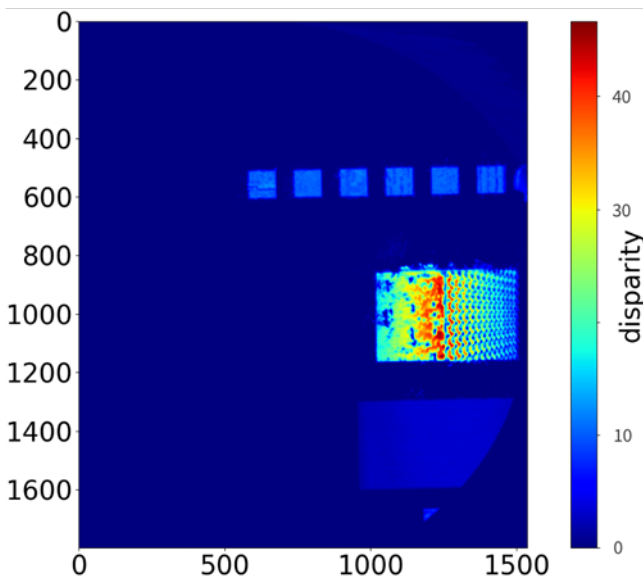


図-12 最終視差画像

また、最終視差画像を基にして三次元復元を行った三次元点群を図-13 に示す。ただし、地面からの距離が大きい点が明るい色になるように色を調節した。

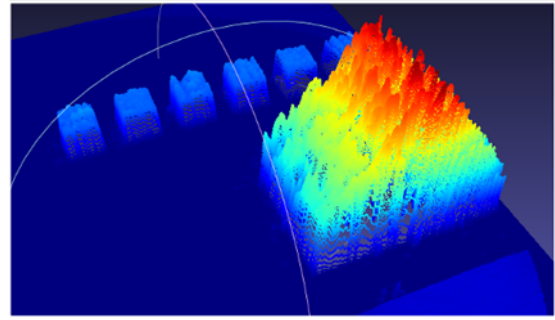


図-13 三次元復元の結果

### (2) 精度評価

本研究では作成された視差画像の精度を、図-14 に示す箇所注目しながら、水平・鉛直方向に分けて定量的に評価した。なお、画像画素と実世界長の縮尺は未知である。よって視差画像上での立方体地物の水平長さおよび視差と、それらに対応する真値 (3 m) から縮尺  $s_H$  および  $s_V$  を設定し、それを他の二つの地物に適用して計算した推定値と真値の誤差  $E$  を算出した。以下に最終視差画像の精度をまとめた表-5 を示す。

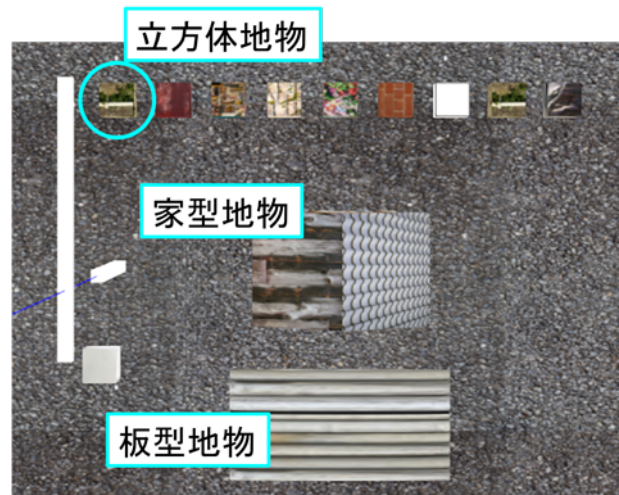


図-14 精度評価の注目箇所

表-5 各精度評価指標

水平 方向	$s_H(mm/px)$	31.25	鉛 直 方 向	$s_V(mm/px)$	284.2
	$E_{2H}(mm)$	31.3		$E_{2V}(mm)$	1485.0
	$E_{3H}(mm)$	592.7		$E_{3V}(mm)$	81.2
	$E_{3H}(mm)$	664.6			

### (3) 考察

まず処理時間については約 379 秒であった。画像ペアの解析に多くの時間を要しているため、使用する画像総

数と全体の処理時間に大きな相関があると考えられる。この処理時間は許容範囲内ではあるものの、今後さらなる短縮を目指して検討を重ねる必要がある。

次に水平方向の精度については、家型地物と板型地物に関して目標の 5-10 cm を達成することはできなかった。三つの精度指標の数値から、最終視差画像のうち、ユニット後半部分（視差画像右側中央から右下部分にかけて）に近づくにつれてスケールが小さくなっていると考えられる。この原因としては、射影変換の繰り返し作用による誤差の蓄積が挙げられる。今回各視差画像を各ユニットの先頭の視差画像に位置合わせしたが、その際ユニット後半の視差画像は、前半部分の視差画像に比べると、射影変換の回数が多い。射影変換とは、そもそも長方形を台形に変形させるような変換であり、特徴点からのホモグラフィ行列の算出の際にわずかなスケールの変化が発生していると考えられ、そのスケール変化が蓄積し、結果としてユニット後半のスケールのずれに繋がったと考えられる。

最後に鉛直方向の精度については、板型地物では目標の 5-10 cm を達成したものの、家型地物では達成できなかった。家型地物での誤差の原因としては、地物のテクスチャ特性による同一平面上での視差値のばらつきが挙げられる。今回、家型地物の屋根部分のテクスチャについては、クレーンから遠い部分（視差画像右側）において瓦の画像を用いた規則的な模様を設定した。すると規則的な模様部分では、特徴量が大きく画像ペア間でマッチングしやすい部分とそうでない部分が存在し、結果として同一平面上であるにも関わらず、視差値についてもパターン模様状になってしまった。そのため、視差値が実際の地物の高さに対してばらついてしまったと考えられる。なお、今回は家型地物の屋根部分の最高地点の視差値を手動で計測しようと試みたため、周辺のバラついた視差値のうち視差値の大きい部分を選択したために、精度指標も大きく計算されてしまったと考えられる。

## 9. 結論

本研究では、建設現場における労働生産性向上のニーズを背景とし、クレーン操縦の自動化のために必要なクレーン周辺の三次元地図の生成に取り組んだ。その際、クレーンブーム先端に取り付けたカメラから得た動画像を用いることによって画像内から静止物体とフックおよび操縦室の分類をする必要があり、またカメラがクレーンに合わせて円軌道を描くことで既存手法では三次元地図生成の安定性に限界があった。そのため本研究では 1) 画像中の特徴点の分類をオプティカルフローによって行う手法、2) SGM を用いて高速かつ要求精度を

満たす三次元地図の生成を行う手法、の 2 つについて検討した。

1) に関してはベイズ判別を利用することでクレーン旋回時における静止物体、フック、操縦室の分類をリアルタイムかつ高い精度で行うことができた一方で、分類に利用した旋回の推定値の算出はクレーンが旋回していることが前提でありクレーン停止時には分類精度が低かったため、クレーン停止時に対する分類手法を考える必要があることが分かった。またフックを静止物体と誤判定してしまうケースも多く見受けられており、三次元地図生成に今回分類した特徴点を用いることを考えると、この誤判定を防ぐ手段を考える必要があることも分かった。そして今回はクレーンの起伏運動については考慮されておらず、実用上クレーンは旋回中に起伏方向にブームを動かすことも多いため、今後は CAN データによるクレーン起伏角の情報を利用することで起伏運動時にも分類ができるように更なる手法の改良も併せて行っていきたい。

2) に関しては、SGM の手法を用いることでクレーン周辺の視差画像の準リアルタイム作成が可能だということ、地物のテクスチャや視差画像同士の位置合わせ精度が作成される視差画像の水平および鉛直方向の精度に大きく影響を与えることが分かった。また、今回は地物のテクスチャは平面であるものを用いたのにもかかわらず波打った視差値が計算されたため、テクスチャに頑強なマッチング手法を探す必要がある。そして、今回はペア画像のステレオ平行化には射影変換を用いたが、ユニット後半の画像に対して繰り返し射影変換を行うことによってスケールの誤差が蓄積することを考慮すると射影変換に換わる平行化の方法を模索する必要がある。

## REFERENCES

- 1) Long, J., Shelhamer, E., and Darrell, T.: Fully convolutional networks for semantic segmentation, CoRR, Vol.abs/1411.4038, 2014.
- 2) MathWorks: SLAM, 2022, <https://jp.mathworks.com/discovery/slam.html>.
- 3) Mur-Artal, R. and Tardós, J. D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras, CoRR, Vol.abs/1610.06475, 2016.
- 4) Campos, C., Elvira, R., Rodrigoíguez, J. J. G., Montiel, J.M. M., and Tardós, J. D.: ORB-SLAM3: an accurate open-source library for visual, visual-inertial and multi-map SLAM, CoRR, Vol.abs/2007.11898, 2020.
- 5) Open Robotics: 2022, <https://gazebosim.org/home>.
- 6) Lucas, B. D. and Kanade, T.: An iterative image registration technique with an application to stereo vision, IJCAI, 1981.
- 7) 栗田 多喜夫: ベイズ決定理論, 2022, <https://home.hiroshima-u.ac.jp/tkurita/lecture/prnn/prnn.html>, [Kurita, T.: *Bayesian decision theory*, 2022, <https://home.hiroshima-u.ac.jp/tkurita/lecture/prnn/>]

- prnn.html].
- 8) Shi, J. and Tomasi: Good features to track, 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600, 1994.
  - 9) Ros, G., Sappa, A. D., Ponsa, D., and Lopez, A. M.: Visual slam for driverless cars : A brief survey, 2012.
  - 10) 繁森央一ほか: ORB-SLAM2 を用いたクレーン周辺の三次元地図のリアルタイム作成, 土木計画学研究委員会, 第 63 回土木計画学研究発表会・講演集, 2021, [Shigemori, H., etc: *Real-time creation of 3D maps around cranes using ORB-SLAM2*, Research Committee on Civil Engineering and Planning, Proceedings of the 63rd Annual Conference on Civil Engineering and Planning, 2021].
  - 11) Hirschmuller, H.: Stereo Processing by Semiglobal Matching and Mutual Information, Vol. 30, 2008.

(Received ?, ?, 2022)

(Accepted ?, ?, 2022)

## HIGH-SPEED 3D MAPPING OF THE CRANE'S SURROUNDINGS FROM MONOCULAR CAMERA IMAGES USING OBJECT CLASSIFICATION

Hisakazu SHIGEMORI, Junichi SUSAKI, Kakuya MATSUSHIMA, Hitomu KOTANI,  
Tomoki KOBAYASHI, Mizuki YONEDA and Marek OSOSINSKI

Recently, labor shortages at construction sites have become more serious, and improving labor productivity has become an issue. One of the solutions to this problem is the automatic operation of cranes. It is needed to be implemented at low cost for sale, therefore it is expected to be implemented by a method to accurately and quickly create a 3D map of the surroundings by acquiring images with a monocular camera attached to the tip of the crane boom. However, the images taken under such conditions often contain the crane's hook, which is one of the factors that reduce the accuracy of the 3D map. Therefore, in this study, in addition to the development of a method to improve the accuracy of 3D map generation using stereo matching and parallax images under the above-mentioned conditions, we have tried to develop a method to remove objects such as crane hooks in the image.