



# 埋め込み変形グラフの適応的調整による 非剛体 3次元トラッキングの高速化

邵 乾瀚<sup>\*1</sup> 渡辺 義浩<sup>\*1</sup>

Qianhan Shao<sup>\*1</sup> and Yoshihiro Watanabe<sup>\*1</sup>

**Abstract** – 非剛体 3次元トラッキングは、拡張現実においてニーズが高い。また、拡張現実応用では、高い没入感を得るために低い遅延が要求されている。従来手法では、埋め込み変形グラフを利用し、計算の複雑度を軽減できることが示された。しかし、人間が知覚できない程度の低い遅延には、トラッキング速度のさらなる向上が必要である。そこで本稿では、対象表面の剛体運動であると考えられる領域において、埋め込み変形グラフのノードを適応的に調整する手法を提案する。提案手法は、非剛体 3次元トラッキングの精度を落とすことなく、高速化できる。本稿では、実験を通じて提案手法の有効性を検証する。

**Keywords** : 3次元センシング, トラッキング, 高速化, 埋め込み変形グラフ

## 1 はじめに

近年、LIDAR や構造化光などの技術の発展に伴い、3次元センシングは実世界を計測する重要な技術となっている。また、3次元トラッキングは、3次元センシングによる取得した空間的情報を利用し、対象の姿勢や変形を追跡して、拡張現実において重要な役割を担っている。

さらに、動く物体へ投影を行うダイナミックプロジェクションマッピング (DPM) などの拡張現実応用では、体験の没入感が重要である。より没入感のある体験の実現に向けて、遅延を知覚できない程度まで抑える必要がある [1]。そのため、物体の動きを捉えるトラッキングの高速化が不可欠である。物体が剛体運動するシーンに対して、従来の 3次元トラッキング手法はマーカーの使用に関わらず 1000 fps を達成した [2][3][4]。物体が非剛体変形するシーンに対しては、マーカーを利用する従来手法において 1000 fps が達成された [5]。

このような背景のもと、本稿では DPM の新たな展開に向けて、非剛体 3次元トラッキングに着目する。マーカーベースの非剛体 3次元トラッキングでは、マーカーの貼り付けに時間を要する。さらに、マーカーが対象の見栄えに影響を与えて、没入感は低下する恐れがある。そのため、マーカーレスの高速な非剛体 3次元トラッキングのニーズが高い。

高速なトラッキングを実現するため、従来手法は埋め込み変形 (ED: Embedded Deformation) グラフを導入した。ED グラフは、本来コンピュータグラフィックスで形状を操作するために使われた手法である [6]。近年、3次元トラッキングや 3次元復元における非剛

体変形のスパースな表現に用いられ、計算コストを削減し、トラッキングのリアルタイム実行を可能にした [7][8][9]。しかし、ED グラフを導入した場合でも、非剛体 3次元トラッキングの手法は 200 fps 以上の速度を達成することが困難であった [9][10]。速度が 200 fps 以下のトラッキングを DPM などの拡張現実応用に適用した場合、遅延が知覚される恐れがある [1]。そこで、ED グラフを用いた非剛体 3次元トラッキングをさらに高速化する必要がある。

ED グラフ以外のグラフは、剛体シーンを対象とした Simultaneous Localization and Mapping (SLAM) で多く使われている。このような SLAM に関する研究では、高速化のため、グラフのベクトル場の発散などの指標に基づいて、グラフノードの部分集合をさらにスパース化する手法が提案された [11][12][13]。ED グラフを用いた非剛体シーンの SLAM についても、同様のスパース化で高速化する手法が提案された [14]。しかし、これらの高速化手法は、SLAM の特性に基づいて提案されているため、3次元トラッキングのみを行う DPM などの応用に適していない。さらに、上記の手法は、高速化のためのスパース化によって、トラッキングの精度が著しく低下している。

そこで本稿では、ED グラフのノード数を対象表面の各領域の変形量に応じて適応的に調整することで、非剛体 3次元トラッキングを高速化する手法を提案する。具体的には、フレーム間の変形がそれほど大きくない領域を剛体領域とみなす。その剛体領域内のノード数を適応的に減らし、それに伴う計算コストの削減によって非剛体 3次元トラッキングを高速化する。また、本手法を実験で検証し、トラッキングの精度を落とすことなく、速度を向上できることを示した。

<sup>\*1</sup>東京工業大学

<sup>\*1</sup>Tokyo Institute of Technology

## 2 関連研究

### 2.1 ED グラフによる非剛体変形の表現

ED グラフは当初, アニメのキャラクターの滑らかな動きをデザインするために提案された [6]. ED グラフは, ローカルな剛体運動の情報を持つグラフノードがスパースに相互接続されたものである. グラフの各ノードのローカルな剛体運動に重みを付けて, 対象表面にある各頂点の複数の隣接ノードの重み付き和を計算することで, グローバルな非剛体変形を滑らかにシミュレートしている.

具体的には, まず剛体トラッキングを実行し, グローバルの回転行列  $\mathbf{R}_c$  と並進ベクトル  $\mathbf{T}_c$  を取得する. 次に, 均一サンプリングを用いて, 対象表面のすべての頂点からスパースなノード集合  $G$  を選択し, 各ノード  $\mathbf{g}_k$  に重み  $\omega_k$  を付ける. 変形後の対象表面にある各頂点の座標  $\tilde{\mathbf{v}}_i$  は以下の式で表される:

$$\tilde{\mathbf{v}}_i = \mathbf{R}_c \sum_{j=1}^m \omega_j [\mathbf{A}_j (\mathbf{v}_i - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j] + \mathbf{T}_c \quad (1)$$

$i$  番目の頂点  $\mathbf{v}_i$  の変形後の座標  $\tilde{\mathbf{v}}_i$  は,  $\mathbf{v}_i$  との距離が最も短い  $m$  個のノード  $\mathbf{g}_j$  の回転行列  $\mathbf{A}_j \in \mathbb{R}^{3 \times 3}$  と並進ベクトル  $\mathbf{t}_j \in \mathbb{R}^3$  とそれらのノードの座標で計算する. また,  $\mathbf{v}_i$  と上記の  $m$  個の隣接するノードをエッジで接続する. 最後に, 上記の座標を用いて, フレーム間の対応する頂点の距離によるエネルギー方程式を構築する. エネルギー方程式を最小化するように, 各ノードの回転行列と並進ベクトルを取得し, 非剛体変形を推定する.

このような ED グラフを用いた表現によって, 数万点の頂点を持つ対象の変形を, 数百程度のノードのグラフに軽量化することができる [6].

### 2.2 ED グラフを用いた非剛体 3 次元トラッキング

非剛体 ICP アルゴリズム [15] などの従来手法では, 対象表面の各頂点に対して高密度に非剛体変形を表現するため, 変数が多くなり, 計算コストが高い. そこで, 自由度を減らし, より速いトラッキングを達成するために, 2.1 節で述べた ED グラフをトラッキングにおける非剛体変形表現に用いる手法が提案された [16][17].

また, ED グラフを用いた非剛体 3 次元トラッキングを GPU 上で計算することで, リアルタイムのトラッキングを達成できるようになった [7][8]. その中の先駆的な研究は, Newcombe らが提案した DynamicFusion [7] である. DynamicFusion は ED グラフを用いたトラッキングを導入し, GPU を用いて, リアルタイムに 3 次元トラッキングと 3 次元復元を同時に行い, 非剛体変形する対象の静的モデルの復元を可能にしたものである.

さらに, 続く研究では, 高速なデプスセンサの導入やトラッキングにおける最適化問題の改善などの手法が提案され, 約 200 fps のトラッキング速度が実現された [9][10].

### 2.3 グラフを用いた 3 次元トラッキングの高速化

ED グラフに対する高速化手法は少ないが, 剛体シーンを対象とした ED グラフ以外のグラフの高速化については多くの研究がある. それらの研究では, グラフノードの部分集合をさらなるスパース化する手法が, 広く適用されている [11][12][13]. Song らは, これらの先行研究と, グラフには視点による見えない部分が存在することを踏まえ, 高速化を達成するために, ED グラフの不可視な部分をさらにスパース化することを提案した [14]. しかし, この手法は実験で 1.5 倍の高速化が達成したが, 精度は 30% 低下した.

## 3 埋め込み変形グラフの適応的調整による

### 非剛体 3 次元トラッキングの高速化

#### 3.1 全体概要

従来の ED グラフを用いた非剛体 3 次元トラッキングの欠点である速度問題に対して, 剛体運動に近い対象表面の領域でノード数を適応的に減らすことで計算コストを削減し, トラッキング速度を向上させる手法を提案する. 本手法の概要を図 1 に示す.

まず, 3 次元スキャンなどの手法により対象の 3 次元モデルと, モデルに対応する ED グラフを事前に構築する. 次に, デプスセンサを使用してリアルタイムの深度画像を取得し, その画像から対象の 3 次元点群を生成する. なお, 初期化は別手法で実施されるものとして, 本手法では 1 フレーム前の情報が得られている状態のみを扱う.

次に 2.1 節で述べた手法と同様に, 対象の 3 次元モデルと, 取得された 3 次元点群の間で, グローバルの剛体変換を推定する. 続いて, 対象表面上の剛体領域を 3.2 節で述べる手法によって検出する. 最後に, 剛体領域のノードの数を 3.3 節で述べる手法で減らし, 調整後の ED グラフに基づいて, 非剛体変形を推定する.

#### 3.2 剛体領域の検出

まず, 3 次元点群の各点  $\mathbf{p}_i$  から対象の 3 次元モデルの表面までの最短距離と, その最短距離に対応する対象の 3 次元モデルの頂点  $\mathbf{v}_i$  を求める. 上記の最短距離は  $\|\mathbf{v}_i - \mathbf{p}_i\|$  で定義される. なお,  $\|\cdot\|$  はユークリッド距離である. センサデータのシーケンスにおいて,  $k$  番目のフレームに応じて変形した対象の 3 次元モデルを,  $k+1$  番目のフレームに対応する 3 次元点群に合わせるように変形する場合, フレーム間変形量

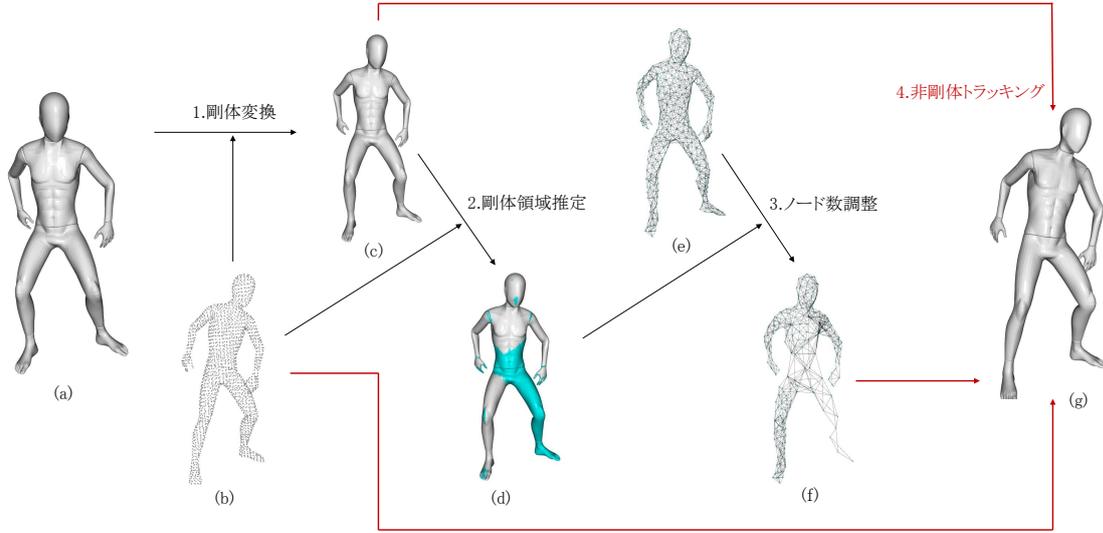


図1 本手法の概要。(a) 事前に構築した対象の 3 次元モデル。(b) リアルタイムの深度画像から生成した 3 次元点群。(c) 剛体変換後の対象の 3 次元モデル。(d) 推定した剛体領域。(e) 対象の 3 次元モデルと同時に生成した対象の ED グラフ。(f) 剛体領域内のノードを削減した ED グラフ。(g) 非剛体変形後の対象の 3 次元モデル。

Fig.1 Overview of the method

を以下の式で定義する:

$$\mathbf{d}_{k,k+1} = \frac{\sum_{i=1}^n \|\mathbf{v}_i - \mathbf{p}_i\|}{n} \quad (2)$$

$n$  は 3 次元点群の点数である。

実世界のシーンでは、デプスセンサデータは連続的に取得されている。そのため、 $k+1$  番目のフレームより前の  $m+1$  フレームの平均フレーム間変形量を次式で計算する:

$$\mathbf{D}_k = \mu \frac{\sum_{j=k-m}^k \mathbf{d}_{j,j+1}}{m+1} \quad (3)$$

$\mu$  は係数であり、 $m$  は  $k$  以下である。 $\|\mathbf{v}_i - \mathbf{p}_i\|$  が  $\mathbf{D}_k$  より小さい場合、頂点  $\mathbf{v}_i$  が剛体領域内にあるとする。

### 3.3 剛体領域におけるノード調整

次に、ED グラフのノード  $g$  と  $g$  に最も近い隣のノード  $\bar{g}$  の距離  $\|g - \bar{g}\|$  を、ノードの影響半径  $\pi$  と定義する。この場合、 $g$  の影響半径  $\pi$  以内に  $g$  以外のノードはない。

続いて、 $g$  が剛体領域内にある場合、影響半径  $\pi$  を  $\tilde{\pi}$  に拡大する。 $g$  の拡大した影響半径  $\tilde{\pi}$  以内にある  $g$  以外のノードは、フレーム間の非剛体変形の推定に利用しない。また、影響半径以内にある  $g$  以外のノードに接続した対象のモデルの頂点も  $g$  に接続し直す。このように、ノードの数を削減する。

しかし、ノード  $g$  が剛体領域の端にある場合や、その剛体領域が小さすぎる場合、非剛体領域にも影響を

与える可能性があるため、 $g$  の周辺のノードを単純に削減することは適切ではない。そこで、剛体領域にあるノード  $g$  に接続するすべての頂点に対して、頂点が剛体領域にある割合を  $\epsilon$  で表示する。 $\epsilon$  を利用し、ノード  $g$  が剛体領域の中心部分、非中心部分、端の部分いずれにあるかを判定する。そのため、次式で示されるように、3つのケースに分けて、ノード  $g$  の影響半径を拡大する。

$$\tilde{\pi} = \begin{cases} \mathbf{k}_\alpha \pi & \epsilon > \alpha \\ \mathbf{k}_\beta \pi & \beta \leq \epsilon \leq \alpha \\ \pi & \epsilon < \beta \end{cases} \quad (4)$$

$\tilde{\pi}$  は拡大した後の影響半径である。

式(4)に示されるように、まず  $\epsilon$  が  $\alpha$  より大きい場合、 $g$  は剛体領域の中心部分にあり、非剛体領域に影響がないと判断し、ノード  $g$  の影響半径を  $\mathbf{k}_\alpha \pi$  に拡大する。次に、 $\epsilon$  が  $\beta$  以上  $\alpha$  未満の場合、 $g$  は剛体領域の非中心部分にあるが、非剛体領域に影響が少ないと判断し、ノード  $g$  の影響半径を  $\mathbf{k}_\beta \pi$  に拡大する。なお、 $\beta$  は  $\alpha$  以下であり、 $\mathbf{k}_\beta$  は  $\mathbf{k}_\alpha$  以下である。ただし、 $\epsilon$  が  $\beta$  より小さい場合、 $g$  が剛体領域端の部分にあり、非剛体領域には影響が大きいと判断し、影響半径の拡大を行わない。

また、削減後のノードの分布が対象表面で不均一になると、トラッキングの精度と速度が大幅に低下することがある。そこで、対象表面の総頂点数に対して、

表1 剛体領域の変数

Table 1 Variables of rigid zone

フレームレート	$k$	$m$
30 fps	10	10
60 fps	20	20
120 fps	40	40
180 fps	60	60
300 fps	100	100

剛体領域にある頂点数の割合を  $\varphi$  で表示する。ノードを削減する時、均一的なノード分布を保つため、異なる  $\varphi$  に対して、異なる  $\mathbf{k}_\alpha$  と  $\mathbf{k}_\beta$  の値を設定する。

例えば、 $\varphi$  が 0.5 に近い場合、半分のノードが影響半径の拡大を行うが、残る半分のノードが影響半径の拡大を行わないため、 $\mathbf{k}_\alpha$  と  $\mathbf{k}_\beta$  を高い値に設定すると、ノードの分布が不均一になる。その場合、 $\mathbf{k}_\alpha$  と  $\mathbf{k}_\beta$  を低い値に設定する必要がある。一方、 $\varphi$  が 1 に近い場合、ほぼすべてのノードが剛体領域の中心部分にあり、 $\mathbf{k}_\alpha$  を高い値に設定しても、ノードの分布が不均一にならない。

## 4 実験

### 4.1 精度の推定

本実験では、提案手法の精度を推定するため、リアルタイム 3 次元点群に応じて変形した後の対象の 3 次元モデルのすべての頂点と、それらの頂点に対応する 3 次元点群内の点のユークリッド距離を計算した。その距離が 0.0005 より小さい比率  $\eta$  を、精度として定義した。

ノード調整を行うトラッキングの精度を  $\eta$ 、ノード調整を行わないトラッキングの精度を  $\bar{\eta}$  とする。 $\eta, \bar{\eta} \in [0, 1]$  である。本稿では、精度  $\bar{\eta}$  に対する精度  $\eta$  の精度変化率  $R$  を下式で定義した:

$$R = \frac{\bar{\eta} - \eta}{\bar{\eta}} \quad (5)$$

なお、精度が向上した場合、 $\eta$  が  $\bar{\eta}$  より大きいため、精度変化率がマイナスである。

### 4.2 実験条件

本実験では、実世界でのセンサデータの代わりに、Li らが作成した合成データセットを用いた [18]。このデータセットから、頂点分布が実世界でのセンサデータのように均一的なデータ mannequin\_StandingCoverTurn を用いた。データのフレーム間時間を  $\frac{1}{30}$  秒に設定して、デフォルトのフレームレートが 30 fps とした。

上記の合成データから、フレーム補間によって 60 fps, 120 fps, 180 fps, 300 fps のシーケンスを生成した。各フレームレートに対する式 (3) の  $k$  と  $m$  の値を表 1 に示す。なお、式 (4) の  $\alpha$  は 0.8,  $\beta$  は 0.5 に設定した。異なる  $\varphi$  に対する式 (4) の  $\mathbf{k}_\alpha$  と  $\mathbf{k}_\beta$  の値を

表2 ノード調整の変数

Table 2 Variables of node adjustment

剛体領域の割合	$\mathbf{k}_\alpha$	$\mathbf{k}_\beta$
$\varphi > 0.8$	4	2
$0.5 \leq \varphi \leq 0.8$	3	2
$\varphi < 0.5$	2	2

表 2 に示す。

また、本実験では Yao らが提案した非剛体 3 次元変形推定手法 Fast\_RNRR [19] を利用して非剛体変形を推定した。実験には Intel Core i9-11900H @ 2.50GHz を搭載した計算機を用いた。

### 4.3 結果

剛体領域内のノードを適応的に調整する非剛体 3 次元トラッキングと、ノード調整を行わないトラッキングの精度や速度を比較した。各フレームレートでの処理速度の向上比と精度変化率を表 3 と表 4 に示す。

ただし、 $\varphi$  が 0.1 以下の場合や、精度低下が 40% 以上の場合、表に None で表示する。各フレームレートで、精度が向上したまたは精度低下が 10% 以下の最も高い速度向上比と、その速度向上比に対応する係数  $\mu$  を表 5 に示す。

表 3, 表 4, 表 5 に示されるように、データのフレームレートが 120 fps 以上の場合、約 2 倍に高速化できた。さらに、各フレームレートに対して、精度の変化は 10% 以下となった。また、180 fps の場合、13.67% の精度低下の代わりに、約 3.4 倍に高速化できることを確認した。

## 5 まとめ

本稿では、対象の変形に応じて、ED グラフのノードを適応的に調整することで、非剛体 3 次元トラッキングを高速化する手法を提案した。実験より、トラッキングの速度が約 2 倍になることを確認した。

今後は本手法を GPU ベースで実装することで、DPM などの拡張現実応用を高速化できると考えられ、より没入感の高い体験を実現することが期待される。

### 参考文献

- [1] Albert Ng, Julian Lepinski, Daniel Wigdor, Steven Sanders, and Paul Dietz. Designing for low-latency direct-touch input. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, p. 453–464, 2012.
- [2] Yoshihiro Watanabe, Toshiyuki Kato, and Masatoshi ishikawa. Extended dot cluster marker for high-speed 3D tracking in dynamic projection mapping. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 52–61, 2017.

表 3 速度向上比  
Table 3 Ratio of speed up

	$\mu = 0.5$	$\mu = 1$	$\mu = 2$	$\mu = 3$	$\mu = 4$	$\mu = 5$
30 fps	1.004	1.216	1.257	1.922	None	None
60 fps	1.177	1.102	1.228	1.149	None	None
120 fps	1.051	1.062	1.057	1.988	1.454	None
180 fps	None	1.200	1.666	1.925	2.659	3.390
300 fps	None	None	1.145	1.925	1.511	1.717

表 4 精度変化率  
Table 4 Ratio of accuracy change

	$\mu = 0.5$	$\mu = 1$	$\mu = 2$	$\mu = 3$	$\mu = 4$	$\mu = 5$
30 fps	-5.37%	-6.58%	16.51%	33.88%	None	None
60 fps	1.34%	3.07%	18.80%	39.38%	None	None
120 fps	-0.21%	1.04%	2.02%	9.22%	1.56%	None
180 fps	None	0.74%	3.14%	6.39%	11.39%	13.67%
300 fps	None	None	1.03%	2.30%	-1.30%	-2.74%

表 5 高速化結果  
Table 5 Speeding up results

フレームレート	$\mu$	精度変化率	速度向上比
30 fps	1	-6.58%	1.216
60 fps	0.5	1.34%	1.177
120 fps	3	9.22%	1.988
180 fps	3	6.39%	1.925
300 fps	3	2.30%	1.925

- [3] 田畑智志, 天野光, 渡辺義浩, 石川正俊. 高速 3 次元トラッキングを用いたマーカレスダイナミックプロジェクションマッピングの検証. 電子情報通信学会技術研究報告, Vol. 117, No. 392, pp. 147–152, 2018.
- [4] 久一空, 渡辺義浩. 500fps デプスセンシングを用いたリアルタイムモデルベーストラッキング. 映像情報メディア学会創立 70 周年記念大会, 31B-3, 2020.
- [5] Gaku Narita, Yoshihiro Watanabe, and Masatoshi Ishikawa. Dynamic projection mapping onto deforming non-rigid surface using deformable dot cluster marker. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 23, No. 3, pp. 1235–1248, 2017.
- [6] Robert W. Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Trans. Graph.*, Vol. 26, No. 3, p. 80–es, 2007.
- [7] Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 343–352, 2015.
- [8] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. Fusion4d: Real-time performance capture of challenging scenes. *ACM Trans. Graph.*, Vol. 35, No. 4, 2016.
- [9] Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. Motion2fusion: Real-time volumetric performance capture. *ACM Trans. Graph.*, Vol. 36, No. 6, 2017.
- [10] Adarsh Kowdle, Christoph Rhemann, Sean Fanello, Andrea Tagliasacchi, Jonathan Taylor, Philip Davidson, Mingsong Dou, Kaiwen Guo, Cem Keskin, Sameh Khamis, David Kim, Danyang Tang, Vladimir Tankovich, Julien Valentin, and Shahram Izadi. The need 4 speed in real-time dense visual tracking. *ACM Trans. Graph.*, Vol. 37, No. 6, 2018.
- [11] Nicholas Carlevaris-Bianco, Michael Kaess, and Ryan M. Eustice. Generic node removal for factor-graph SLAM. *IEEE Transactions on Robotics*, Vol. 30, No. 6, pp. 1371–1385, 2014.
- [12] Kevin Eickenhoff, Liam Paull, and Guoquan Huang. Decoupled, consistent node removal and edge sparsification for graph-based SLAM. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3275–3282, 2016.
- [13] Jingwei Song, Jun Wang, Liang Zhao, Shoudong Huang, and Gamini Dissanayake. MIS-SLAM: Real-time large-scale dense deformable SLAM system in minimal invasive surgery based on heterogeneous computing. *IEEE Robotics and Automation Letters*, Vol. 3, No. 4, pp. 4068–4075, 2018.
- [14] Jingwei Song, Fang Bai, Liang Zhao, Shoudong Huang, and Rong Xiong. Efficient two step optimization for large embedded deformation graph based SLAM. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9419–9425, 2020.
- [15] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.
- [16] Hao Li, Robert W. Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Proceedings of the Symposium on Geometry Processing, SGP '08*, p.

1421–1430, 2008.

- [17] Hao Li, Bart Adams, Leonidas J. Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.*, Vol. 28, No. 5, p. 1–10, 2009.
- [18] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4DComplete: Non-rigid motion estimation beyond the observable surface. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12686–12696, 2021.
- [19] Yuxin Yao, Bailin Deng, Weiwei Xu, and Juyong Zhang. Quasi-newton solver for robust non-rigid registration. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7597–7606, 2020.

© 2023 by the Virtual Reality Society of Japan (VRSJ)