

工場内巡回ロボットにおける広域実環境再構築システムのための自己位置回復手法

古都 知哉^{*1} 阿部 哲朗^{*1} プロプスキ アレクサンダー^{*1} 間下 以大^{*1*2}
清川 清^{*1*2} 竹村治雄^{*1*2} 福田 登仁^{*3}

Relocalization method of Real-time Wide-Area Reconstruction System Using RGB-D Camera
for a Patrol Robot

Tomoya Furuichi^{*1}, Tetsuro Abe^{*1}, Alexander Plopski^{*1}, Tomohiro Mashita^{*1*2},
Kiyoshi Kiyokawa^{*1*2}, Haruo Takemura^{*1*2} and Takahito Fukuda^{*3}

Abstract – 我々は、工場内を巡回する移動ロボットに搭載することを前提とした RGB-D カメラを用いた広域実環境再構築システムについての研究開発を行なっている。広域実環境再構築システムは ICP アルゴリズムによるカメラ移動量推定を行うことでカメラ位置を把握し、実環境の再構築を行っている。しかし、カメラ移動量推定に失敗した場合、実環境再構築を続けることができない問題がある。カメラ移動量の推定の失敗は、カメラの前を人や障害物が横切るだけでも発生する可能性があるため、巡回ロボットが安定して稼働する上で大きな問題となる。本研究では、現実の三次元空間に対して、カメラの大まかな位置を求めることができる自己位置回復手法を提案する。この手法を用いることで、ICP アルゴリズムによるカメラ移動量推定を回復でき、実環境再構築を続けることが可能となる。本稿では、自己位置回復手法を実現する上での問題点を明らかにし、実装方針について述べる。また、試作システムを用いて実際に自己位置回復が行えることを確認した予備実験について述べる。

Keywords : 巡回ロボット, RGB-D カメラ, 自己位置回復, SURF, GAFD

1 はじめに

近年、工場内において作業の自動化、高速化を目的としたロボットが広く用いられている。しかし、部品の選別といった作業においては現状では人手による確認が必要であり、十分な自動化が行われていない。現状よりもさらに作業の自動化、高速化を推進し、人手による作業を削減するため、より高度な作業を自律的に行うことのできるロボットが工場において求められている。

現在産業界で利用されている自動巡回ロボットは、事前の準備が必要である。例えば、一部のロボットは、工場内のロボットを巡回させる経路に沿って磁気テープを貼付けないといけない。この点は、特に広い工場において事前準備が煩雑になるという問題がある。また、ロボットが巡回している経路上に障害物が存在していると、ロボットの巡回に支障をきたす可能性がある。このため、工場の作業員はロボットに対して注意を払わなければならないという問題がある。

我々は巡回ロボットのための、事前準備が不要である RGB-D カメラを用いたシステムの研究開発を行っ

ている [1]。100m×100m 程度の工場を想定し、広域で実環境の再構築、障害物認識及び部品認識を行うことを目標としている。システム実現のために、Newcombe らが提案した RGB-D カメラを用いた三次元実環境構築手法である KinectFusion [2] を利用している。KinectFusion は、市販の RGB-D カメラを用いて、現実のシーンを様々な方向から眺めることにより、リアルタイムで実環境の三次元再構築を行うことができる。リアルタイムの実環境三次元再構築を行うために、ICP アルゴリズムを用いたフレーム間のカメラの移動量推定と三次元ボリュームの更新をリアルタイムに行っている。

KinectFusion の問題点として、カメラの移動量推定に失敗した場合に、実環境再構築を続けることができないという問題がある。カメラの移動量推定の失敗は、カメラを速く動かした場合やカメラの目の前を人や障害物が横切った場合などに発生する。工場において、狭い通路を通過している場合等、カメラの目の前を人や障害物が通過することは通常考えられることであり、工場において KinectFusion システムを使う上で、カメラの移動量推定に失敗した場合でも三次元構築を続ける仕組みは必要不可欠である。

上記の問題を踏まえ、本研究では現実の三次元空間に対して、現在の大まかなカメラ位置を求めることのできる自己位置回復手法を提案する。この手法を

^{*1}大阪大学 大学院情報科学研究科

^{*2}大阪大学 サイバーメディアセンター

^{*3}ウエストユニティス (株)

^{*1}Graduate School of Information Science and Technology, Osaka University

^{*2}Cybermedia Center, Osaka University

^{*3}WestUnitis CO.LTD.

KinectFusion に対して用いることで、カメラの移動量推定に失敗した場合でも現在の大きなカメラ位置を求めることができ、フレーム間のカメラ移動量推定に失敗した場合でも復帰を行い、三次元再構築を続けることが可能となる。

自己位置回復手法のもう一つの利点として、システムの再稼働時における自己位置の推定がある。今回の研究において想定されるロボットの移動範囲は100m×100mの広さがあり、リアルタイムで三次元再構築が行える KinectFusion であっても全環境の再構築は時間がかかってしまう。このため、システムにおいては一度復元した三次元ボリュームデータを保存し、次回起動時にも再利用できるようにするのが望ましい。この状況において、次回起動時に保存した三次元ボリュームに対してカメラがどの位置、方向であるかを求める必要がある。

2 関連研究

三次元空間内のカメラ位置の推定は、2つのカテゴリに大きく分けられる。一つ目は特徴点ベースの位置推定手法 [3] で、もう一つは画像ベースの位置推定手法 [4, 5] である。

特徴点ベースの手法は、シーン全体を画像特徴量で識別できる多数の特徴点のデータベースとして保持する。画像特徴量の例としては、SIFT[6] や SIFT を高速化した SURF[7] が存在する。トラッキングに失敗した際には、現在見ているシーンに存在する特徴点のうち、データベース内の特徴点に似ているものを探し、カメラ姿勢を求める。

一方で画像ベースの手法は、画像全体から少量の画像データを選出し、データとその画像を撮影した際のカメラ姿勢の組のデータベースを作成する。格納されるデータ組はキーフレームと呼ばれる。トラッキングに失敗した際には、現在見ているシーンのキーフレームとデータベース内の全キーフレームを比較し、最も似ているキーフレームを計算、そのキーフレームのカメラ姿勢を使用し位置を復元する。

処理時間に関して比較すると、特徴点ベースの手法は特徴点の抽出、マッチング、及びカメラ姿勢の計算に時間を要し、リアルタイムでの動作は一般に難しい。一方で、画像ベースの手法については、キーフレームの生成方法や比較方法により処理時間は異なるが、リアルタイムで動作可能な手法 [5] も存在する。一方で、画像ベースの手法に関しては、トラッキングに失敗した場合、カメラをトラッキングに成功している間に動いていた位置姿勢に戻さなければカメラ位置の推定を行えないという問題がある。すなわち、画像ベースの手法を用いると、カメラが一方向に移動する状況においてトラッキングに失敗した場合、カメラを逆の方向

に一度戻してカメラ位置を推定、回復させなければならないという問題がある。

画像ベースの手法に上記のような問題があることから、本研究では特徴点ベースの手法を採用する。工場において、ロボットが移動中にトラッキングに失敗した場合、逆方向に戻らなければいけないことを避けるためである。

3 自己位置回復手法の設計

本節では、始めに三次元再構築システムの構成について概観する。その後自己位置回復機能を実装する上で考慮すべき問題を議論し、続いて具体的な設計を示す。

3.1 三次元再構築システムの構成

我々の KinectFusion を利用した、RGB-D カメラを用いた三次元再構築システム [1] は、以下の機能を備えることを目指して開発を進めている。

- 三次元実環境の再構築
- 移動経路計算
- 障害物認識
- 部品の識別と位置及び向き

三次元実環境再構築は、KinectFusion のアルゴリズムに基づいて行われている。具体的には、デプス画像を RGB-D カメラから読み込み、カメラ移動量推定によるカメラ位置を用いてデプス画像を位置合わせし、実環境三次元ボリュームを更新する。実環境三次元ボリュームは、移動経路計算 (ICP アルゴリズム)、障害物認識、部品識別のために利用される。また、再構築された三次元ボリュームは保存され、次回起動時には保存された三次元ボリュームを再利用する。この時、保存された三次元ボリュームに対するカメラ位置を初期推定する。以下、再構築される実環境三次元ボリュームの座標を世界座標と定義する。

3.2 自己位置回復を実装する上での問題

特徴点ベースの手法を用いて自己位置回復機能を実装する際の問題点としては、先述したように計算時間の問題がある。また、自己位置回復を実装する上での環境 (工場内) に特有の問題として、広域環境 (100m×100m 程度) であることがある。広域環境であることは、すなわち特徴点データベースが最終的に非常に大きくなることを意味し、特徴点のマッチングに時間がかかってしまう。特徴点のマッチング及びカメラの姿勢行列の推定の計算に時間がかかることはすなわち、ロボットの応答性が悪くなることを意味し、極力計算時間を抑える必要がある。

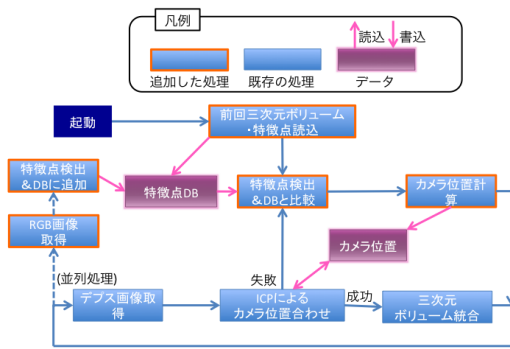


図 1: 自己位置回復機能を組み込んだ広域実環境再構築システムの処理フロー

3.3 自己位置回復の実装方針

自己位置回復機能を組み込んだ三次元再構築システムの処理フローを図 1 に示す。本システムは、ICP アルゴリズムによるカメラ移動量推定に成功している間は、KinectFusion のアルゴリズムを用いた実環境三次元ボリュームの再構築を行いつつ、RGB 画像から特徴点を抜き出し、特徴点データベースに追加を行う。本システムは、カメラを速く動かしたり、カメラの目の前に障害物が現れることにより、ICP アルゴリズムによるカメラ移動量推定が失敗した場合、現在の RGB 画像から特徴点を抽出し、特徴点データベースとのマッチングを行い、マッチング結果から現在のカメラ位置姿勢行列を推定する。ここで、カメラ位置姿勢行列の推定に成功し、その姿勢を初期値として用いた ICP アルゴリズムによるカメラ位置推定が成功すれば、自己位置が回復したと言える。ICP アルゴリズムが失敗した場合は、再び現在の RGB 画像から得られた特徴点を用いた特徴点データベース、マッチング、及びカメラ位置姿勢行列推定を成功するまで繰り返す。以下、各処理項目について詳しく説明する。

特徴点の取得

特徴点の抽出には、二次元画像における特徴量である SURF[7] と、GAFD[8] を組み合わせた SURF+GAFD 特徴量を利用する。GAFD 特徴量は、重力加速度の方向を表す量である。本研究では、巡回ロボットに搭載するシステムを想定しているため、カメラは常に正立していると仮定する。そのため、カメラ画像の下向きが重力方向であると判断し、SURF 特徴量の向きと、重力加速度の方向の間の角度を GAFD 特徴量とする。

SURF により検出される特徴点は輝度が大きく変化する点のため、二次元画像における、三次元的に離れた二つの物体の境界 (ジャンプエッジ) において SURF 特徴点が検出されてしまう。ジャンプエッジ上に検出される特徴点は三次元位置が不安定であるため、自己

位置回復の目的には適さない。この問題を解決するため、本研究では RGB-D カメラから得られるデプス画像を用いてジャンプエッジ上の SURF 特徴点のフィルタリングを行う。以下、このフィルタリングをジャンプエッジフィルタリングと呼ぶ。

また、このフェーズにおいてカメラ位置姿勢行列の推定で使用する、特徴点の三次元座標を求めておく。この三次元座標は特徴点の二次元座標とデプスで表されるクリップ座標系の点から、世界座標系の点に変換することで求められる。なお、ICP アルゴリズムが失敗している場合は、現在のカメラ位置を原点とする座標系における特徴点の三次元座標を用いる。

特徴点データベースの構築

特徴点データベースの更新は、ICP アルゴリズムによるカメラ移動量推定が成功している間、逐次行われる。特徴点データベースに新たな特徴点を追加する際には、特徴点データベースの肥大化を防ぐために同じ特徴点が追加されないようにする。このため、特徴点データベース内に、新しい特徴点とユークリッド距離が近く、かつ SURF+GAFD 特徴量が類似している特徴点が既に存在している場合、その特徴点はデータベースに追加しない。SURF+GAFD 特徴量の類似度の比較は、GAFD 特徴量の比較と SURF 特徴量の比較の二段階で行う。まず、GAFD 特徴量の比較は、2 つの GAFD 特徴量の間の角度を求め、その角度が大きい (例えば 30° 以上) 場合、その 2 つの特徴点是对応していないと判断する (SURF 特徴量の比較は行わず、類似度無しとする)。GAFD に対応していないと判断されない特徴点は、続いて SURF 特徴量の比較を行う。SURF 特徴量の類似度としては、特徴量間のユークリッド距離を使用する。ユークリッド距離が小さいほど、特徴量が類似していると判断する。

特徴点のマッチング

特徴点のマッチングは、総当たりで行う。すなわち、現在の RGB 画像から抽出された各 SURF+GAFD 特徴点に対し、特徴点データベース内の最も類似度が大きい SURF+GAFD 特徴点とマッチさせる。

カメラ位置姿勢行列の推定

上述の特徴点マッチングにより、対応する特徴点の組の集合が得られる。この特徴点組集合の三次元座標の関係に対応する三次元剛体変換を RANSAC アルゴリズム [9] を用いて求める。三次元剛体変換を求める具体的なアルゴリズムをアルゴリズム 1 に示す。アルゴリズム 1 は、推定に成功した場合に推定結果の三次元剛体変換を返し、失敗した場合には何も返さない。アルゴリズム 1 内で用いられている、点群間の三次元剛体変換を推定する関数 RigidBodyTransEstimate のアルゴリズムをアルゴリズム 2 に示す。推定された三次元剛体変換は、ICP 失敗前のカメラの位置と現在の

Algorithm 1 特徴点組集合間の三次元剛体変換の RANSAC アルゴリズムを用いた計算

```

1: constant:
   M: 仮の三次元剛体変換を推定する際に使用する
      点対の数
   T: 再投影誤差の外れ値判断のための閾値
2: input: Pair =  $\{(p_{si}, p_{di})\}, i = 1..N, p_{si}, p_{di} \in R^3$ 
3: iter  $\leftarrow$  0
4: max_inliners_count  $\leftarrow$  0
5: while iter < MAX_ITER do
6:   SubPair  $\leftarrow$  Pair からランダムに選ばれた M 個
      の要素
7:   Trans  $\leftarrow$  RigidBodyTransEstimate(SubPair)
8:   inliners_count  $\leftarrow$  Pair について Trans の再投影
      誤差が閾値 T 以下だった要素数
9:   if inliners_count  $\geq$  M and
      inliners_count > max_inliners_count then
10:    max_inliners_count  $\leftarrow$  inliners_count
11:    best_Trans  $\leftarrow$  Trans
12:    iter  $\leftarrow$  iter + 1
13:   end if
14: end while
15: if max_inliners_count > 0 then
16:   SubPair  $\leftarrow$  Pair のうち best_Trans の変換の誤
      差が閾値 T 以下だった要素の集合
17:   Trans  $\leftarrow$  RigidBodyTransEstimate(SubPair)
18:   return Trans
19: else
20:   return (failed)
21: end if

```

Algorithm 2 三次元剛体変換の推定

```

1: Procedure: RigidBodyEstimate(Pair) 関数: 三
   次元剛体変換を推定する
2: input: Pair =  $\{(p_{si}, p_{di})\}, i = 1..M$ 
3: centroid1  $\leftarrow$   $\frac{1}{M} \sum_{i=1}^M p_{si}$ 
4: centroid2  $\leftarrow$   $\frac{1}{M} \sum_{i=1}^M p_{di}$ 
5:  $H = \sum_{i=1}^M (p_{si} - \text{centroid1})(p_{di} - \text{centroid2})$ 
6:  $[U, S, V] = \text{SVD}(H)$ 
7:  $R = VU^T$ 
8:  $t = \text{centroid2} - R * \text{centroid1}$ 
9: return  $\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$ 
10: end

```



図 2: 実験の様子。

カメラ位置の関係を表しているため、世界座標系における現在のカメラの位置を求めることができる。

4 予備実験

前節の方法と KinectFusion を用いた自己位置回復機能を有する実環境再構築システムを試作し、システム起動直後の初期推定における、GAFD やジャンプエッジフィルタリングを用いることによる計算時間、推定精度の違いを確認するための予備実験を行った。

4.1 実験環境

実験環境の様子を図 2 に示す。図の右側が撮影対象のシーンである。図の左側に示されるように、RGB-D カメラは、巡回ロボットに見立てた台車上に設置され、シーンを撮影する。RGB-D カメラは常に地面に対して正立している。

4.2 実験方法

実験では、予めシーンの RGB-D 動画を撮影しておき、その RGB-D 動画を入力として自己位置回復機能を組み込んだ実環境再構築システムを動かした。これは GAFD 及びジャンプエッジフィルタリングが有用であるか比較するため、これらの手法を利用した場合としていない場合の比較を、同一の RGB-D 動画を用いて行うためである。

具体的な実験方法について説明する。まず、ロボットを模倣した台車を動かし、シーンの周りを 3 周し、RGB-D 動画を撮影する。撮影後、3 周したシーンを 3 分割する (分割された RGB-D 動画をそれぞれ動画 a、動画 b、動画 c とする)。まず、動画 a を入力として自己位置回復機能を組み込んだ実環境再構築システムを動作させ、SURF 特徴点データベースを作成、保存する。続いて、保存された特徴点データベースを初期入力とし、動画 b 及び動画 c をそれぞれ入力して自己位置回復機能を組み込んだ実環境再構築システムを動作させ、自己位置の推定結果、及び ICP アルゴリズムによるカメラ位置推定が回復するかどうかを確認する。

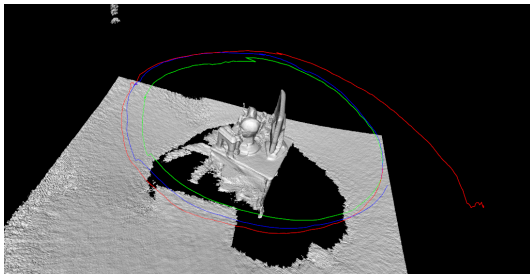


図 3: カメラ位置の軌跡: 赤色が 1 周目, 緑色が 2 周目, 青色が 3 周目を表す.

以上の動画 a について行った上記の作業を同様に動画 b,c についても行う (合計 6 回). 上記の計 6 回の実験を, 特徴点の種類と取得方法により, ジャンプエッジフィルタリングと GAFD を用いる方法, ジャンプエッジフィルタリングのみを行う方法, GAFD のみを用いる方法, どちらも用いない方法の 4 種類, 計 24 回のシステム動作を行い, 結果の比較を行う.

自己位置回復機能を組み込んだ実環境再構築システムを動作させた計算機環境を表 1 に示す.

4.3 実験結果と考察

予備実験の結果を表 2 及び表 3 に示す. 実験に使用した RGB-D 動画は毎秒 30 フレームであり, 全体で 21.3 秒, 動画 a, b, c はそれぞれ 8.5 秒, 6.2 秒, 6.3 秒である. 実験中のカメラの位置の軌跡を図 3 に示す. 図 4 に, 各方法で動画 a を動作させた場合に構築された特徴点データベースの特徴点の分布を示す. ここ

表 1: 実環境再構築システムを動作させた計算機環境.

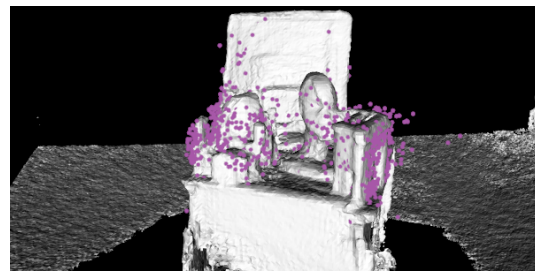
CPU	Core i7-2600 3.40GHz
メインメモリ	4GB
GPU	GeForce GTX 670 4GB

表 2: 推定結果.

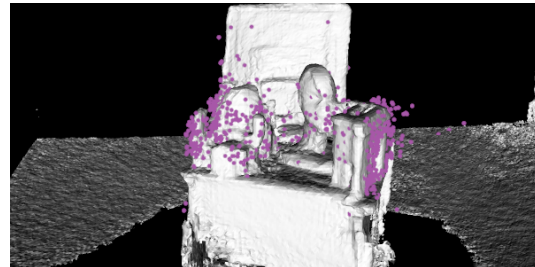
	JEF+GAFD	GAFDのみ	JEFのみ	双方無し
MOV:a, DB:b	3 回目で成功	1 回目で成功	2 回目で成功	4 回目で成功
MOV:a, DB:c	1 回目で成功	1 回目で成功	2 回目で成功	1 回目で成功
MOV:b, DB:a	1 回目で成功	1 回目で成功	1 回目で成功	2 回目で成功
MOV:b, DB:c	1 回目で成功	1 回目で成功	1 回目で成功	1 回目で成功
MOV:c, DB:a	1 回目で成功	2 回目で成功	3 回目で成功	2 回目で成功
MOV:c, DB:b	1 回目で成功	1 回目で成功	1 回目で成功	成功せず (4 回)

表 3: 特徴点データベースとのマッチング及びカメラ位置推定平均時間.

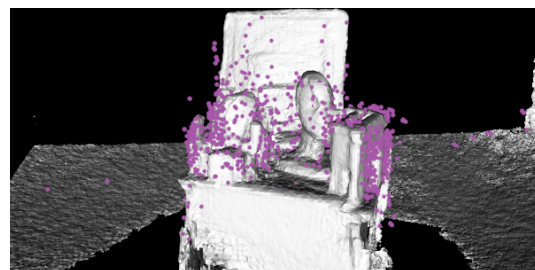
	JEF+GAFD	GAFDのみ	JEFのみ	双方無し
MOV:a, DB:b	0.50 秒	2.01 秒	0.47 秒	1.88 秒
MOV:a, DB:c	0.34 秒	2.00 秒	0.46 秒	1.94 秒
MOV:b, DB:a	1.19 秒	3.12 秒	1.17 秒	2.96 秒
MOV:b, DB:c	1.17 秒	3.31 秒	1.09 秒	3.24 秒
MOV:c, DB:a	0.93 秒	2.71 秒	0.70 秒	2.54 秒
MOV:c, DB:b	1.05 秒	3.04 秒	0.94 秒	2.84 秒



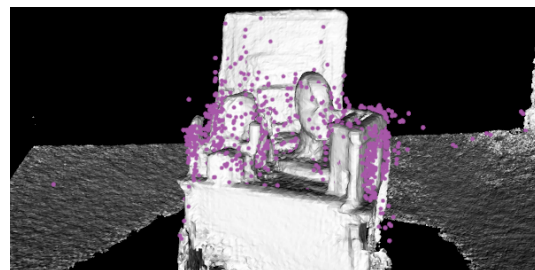
(a) ジャンプエッジフィルタリング+GAFD



(b) ジャンプエッジフィルタリングのみ



(c) GAFD のみ



(d) 双方無し

図 4: 動画 a により構成された特徴点データベース内の特徴点

で, 表 2 及び表 3 における, 表 1 行目の JEF はジャンプエッジフィルタリングを意味し, 表 1 列目における MOV:a, DB:b とは, 動画 b を用いて作成された特徴点データベースを初期入力として動画 a を自己位置回復機能を組み込んだ実環境再構築システムに入力した場合を表し, その他も同様である.

表 2 及び表 3 より, ジャンプエッジフィルタリングを用いることで, 精度を損なわずマッチングの計算時間を改善できることが期待できる. また, GAFD を用いることで, 推定精度が向上することが期待できる. GAFD を用いることで特徴点データベースのマッチング及びカメラ位置推定平均時間が増加している. こ

れは、GAFDを導入することで、SURF特徴量だけでは類似していると判断されるがGAFDが異なるために類似していないと判断される特徴点組の存在のため、特徴点データベース内の特徴点の数が増加したからであると考えられる。

5 おわりに

本稿では、工場内巡回ロボットのための三次元再構築システムのための自己位置回復手法の設計を説明し、予備実験について報告した。今後、予備実験で得られた知見を元に、手法の改善を行い、広域な環境において手法の有用性を確認する実証実験を行う予定である。

参考文献

- [1] 阿部, 古都, プロブスキ, 間下, 清川, 竹村, 福田. 工場内巡回ロボットのための RGB-D カメラを用いた広域実環境再構築システムの設計. サイバースペースと仮想都市研究会, 2013.
- [2] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time Dense Surface Mapping and Tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, Washington, DC, USA, 2011.
- [3] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Proceedings of the International Symposium on Robotics Research*, Flagstaff, AZ, USA, 2011.
- [4] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. W. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, 2013.
- [5] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi. Real-Time RGB-D Camera Relocalization. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, Adelaide, SA, Australia, 2013.
- [6] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision*, Washington, DC, USA, 1999.
- [7] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 2008.
- [8] D. Kurz and S. BenHimane. Inertial Sensor-Aligned Visual Feature Descriptors. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, USA, 2011.
- [9] M. A. Fischler and R. C. Bolles. Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 1981.