MEASUREMENT SCIENCE REVIEW, **18**, (2018), No. 3, 86–93

DE GRUYTER OPEN

ISSN 1335-8871

MEASUREMENT SCIENCE REVIEW

Journal homepage: http://www.degruyter.com/view/j/msr

Proportional Error Back-Propagation (PEB): Real-Time Automatic Loop Closure Correction for Maintaining Global Consistency in 3D Reconstruction with Minimal Computational Cost

Morteza Daneshmand[1], Egils Avots[1], Gholamreza Anbarjafari[1,2]

[1]*iCV Research Group, Institute of Technology, University of Tartu, Tartu, Estonia, {ea,md,shb}@icv.tuit.ut.ee*
[2]*Department of Electrical and Electronic Engineering, Hasan Kalyoncu University, Gaziantep, Turkey*

This paper introduces a robust, real-time loop closure correction technique for achieving global consistency in 3D reconstruction, whose underlying notion is to back-propagate the cumulative transformation error appearing while merging the pairs of consecutive frames in a sequence of shots taken by an RGB-D or depth camera. The proposed algorithm assumes that the starting frame and the last frame of the sequence roughly overlap. In order to verify the robustness and reliability of the proposed method, namely, Proportional Error Back-Propagation (PEB), it has been applied to numerous case-studies, which encompass a wide range of experimental conditions, including different scanning trajectories with reversely directed motions within them, and the results are presented. The main contribution of the proposed algorithm is its considerably low computational cost which has the possibility of usage in real-time 3D reconstruction applications. Also, neither manual input nor interference is required from the user, which renders the whole process automatic.

Keywords: 3D reconstruction, global consistency, loop closure correction, Iterative Closest Point, Proportional Error Back-propagation.

1. Introduction

Reconstruction of 3D objects and scenes has various applications in research and industry contexts, examples of which include virtual reality [1–3], 3D Scanning [4], and Simultaneous Localization and Mapping (SLAM) [5] by autonomous systems [6]. The latter goal is usually achieved through taking multiple shots of RGB, depth or RGB-D frames and registering each of them onto the previous one through finding the relevant transformations, using algorithms such as Iterative Closest Point (ICP) [7, 8]. Nevertheless, each individual transformation usually entails a certain level of error, which when accumulated throughout a large sequence, will cause a noticeable misalignment between the two ends.

The problem of correctly closing the loop for achieving global consistency in 3D reconstruction has been investigated and approached through different techniques in the existing literature. In most cases, especially in the presence of large sequences of frames, the latter is necessary. Incrementally tracking the motion by accumulating the drift throughout the frame transformations [9] is one of the earliest examples that has been utilized along with the Structure from Motion (SfM) [10]. The offline optimization procedure proposed in [11] is another early example from the foregoing list.

One of the first algorithms accomplishing real-time performance in creating globally consistent 3D representations of objects based on sequences of frames taken by a handled camera was devised on the basis of probabilistic analysis of feature position approximations [12], which was not capable of dealing with sequences larger than a certain amount. More clearly, due to the high computational cost involved, the latter method will fail to demonstrate real-time performance if a large scene is going to be reconstructed, which demands creating great feature vectors and incorporating them into the calculations. Even with smaller scenes, the amount of data to be handled is larger than what could be sustained along with a dense filter map, which incurs having to ignore some of the features, which is tantamount to reducing the accuracy.

Apart from detecting the loop, the majority of the loop closure correction methods proposed in the literature relies on complex and time-consuming statistical and mathematical algorithms and operations, including computationally expensive optimization procedures, which often require intense manual input and interference from the user as well, practically preventing a real-time functionality, even if other elements of the pipeline comply with it. In [13] the closing of loop is achieved using a pose graph optimization algorithm based on the features extracted using the RGB data, for instance. The foregoing approach is an example of the long list of methods possibly leading to impressive global alignment, but in the case of being exposed to large databases, either delaying the whole reconstruction process or loosing the potential precision and accuracy due to implementing a global fu-

DOI: 10.1515/msr-2018-0013

sion algorithm, which is responsible for reducing the frames to representations that are based on dense patches.

In the article proposed by Steinbrucker et al. [14] the loop closure is estimated based on key-frames, where each new key frame is matched against all previous key frames. The loop closure is detected based on entropy ratio, where a small error between frames coincides with high entropy value.

To reduce the computational cost in frame feature matching, an efficient mechanism detecting loop closures via landmarks is presented in the article by Liu et al. [15]. The landmarks are compared between the incoming images and all landmarks. According to the match information, the loop closure can be detected.

Article by Shiratori et al. [16] describes a method for aligning very large sets of 3D point clouds. From an initial estimate of the sensor paths, a 3D graph is constructed and the alignment problem is decomposed into smaller ones based on the loop closures that exist in this graph. Data is aligned with Simultaneous GICP (S-GICP) that exploits the loop closure property to produce highly accurate intra-loop registration results. The individual loops are then combined into a single, consistent point cloud via an inter-loop alignment step that reconnects the graph of loops, according to a least squares optimization.

The loop-closure problem is most widely explored int the SLAM (simultaneous localization and mapping), where the absence of loop-closure detection and error correction can cause large errors, as it accumulates over the frames. In SLAM problems, usually, a pose graph is built and then corrected using the loop-closure constraint.

A popular approach [17] for this problem is the iSAM [18] algorithm that is based on fast incremental matrix factorization for the correction of transformation matrices. With the help of QR decomposition of the matrix only the values that change are updated, resulting in fast performance. The information matrix was also used for error estimation.
Similar to this solution is the iSAM2 [19] system, where Keiss proposed to use Bayesian trees, a data structure that provides a better understanding of the matrix factorization in terms of probability densities. It was shown how the fairly abstract updates to a matrix factorization translated to a simple editing of the Bayes tree and its conditional densities. As a result this system was faster and more accurate than the previous one.

Another commonly used approach for loop-closure is using RANSAC and keyframes as proposed in [20]. Once detected, to minimize the conflict between sequential constraints and loop closure constraints, TORO [21] was employed. TORO provides a gradient descend based error minimization solution for constraint-networks. The authors ran TORO each time a loop-closure was detected, using the output of the previous run as initial guess.

Even though iSAM, TORO and iSAM2 are a great solution for the SLAM problem, they are overly complicated for simpler and more controlled environments, where building a location graph would be unnecessary.

In this paper, a loop closure correction algorithm with a negligible computational load is proposed, which is referred to as Proportional Error Back-Propagation (PEB), and aims at applications where the sequence always possesses similar first and last frames, whose examples, among others, include scanning a room while stopping at a pose similar to that of the starting frame. The main virtue of the PEB is its unparalleled fastness, i.e. it usually takes a fraction of a second for it to correct the transformations throughout the whole sequence. Such algorithm can be used in producing more realistic models subject to use in virtual fitting rooms or virtual reality applications [22–28].

The remainder of the paper is organized as follows: The proposed method is introduced in the next section. Afterwards, the experimental results are presented and discussed. Finally, the paper is concluded.

2. The Proposed Method

In this section, the underlying idea of the PEB is described, along with the associated mathematical framework. It is worth noticing that in a 3D reconstruction context, various modules shall either precede or succeed the loop closure system, none of which is investigated in this paper, where only a selection of the existing solutions is considered for verifying the efficiency of the PEB. From a broad perspective, the items from the foregoing list may include preprocessing and registering the depth frames and post-processing the resulting point cloud.

The input to the PEB algorithm is a sequence of depth frames, containing the 3D coordinates of the corresponding points in the associated systems, and a set of homogeneous transformations each of which supposedly maps every point in a frame to the system of coordinates through which the points from the one preceding it within the sequence are represented. More clearly, the foregoing transformations have been calculated by a registration algorithm which is expected to find the camera poses for all the frames, based on which it obtains transformations that approximately map every point in a frame to the one matching it in the previous frame.
However, the above transformations are usually not totally accurate, and the negligible error entailed by each of them still contributes to a considerable overall error, which causes overall inconsistency and prevents the reconstruction loop from closing. More clearly, all the points in each frame are supposed to be mapped to their locations in the reference coordinate system, which is tantamount to that of the first frame, by means of a homogeneous transformation resulted from accumulating a sequence of transformations each of which maps them one step backward, i.e. from the coordinate system associated with a frame to that of the one preceding it, where although no outstanding misalignment might show up at every step, the aggregated error may be considerable. The latter errors may have been caused or compounded by a variety of factors, including vibration of the camera or its movement in

directions, or under axes, other than the intended ones and measurement noise.

The purpose of the PEB is to overcome the above error, and correct the loop closure, taking the following principle into account as the criterion: If the first and the last frames are exactly the same, the cumulative homogeneous transformation taking the latter to the former must be equal to the identity transformation. If the aforementioned condition is met, i.e. if the first and the last frames are the same, the overall error is equivalent to the existing cumulative transformation supposedly mapping the last frame to the first one. By the PEB, to correct the foregoing error, it is back-propagated throughout the chain of transformations based on their proportional contribution to the overall transformation.

In order to do so, the rotation and translation components of the transformations are modified separately and respectively, where an extra module mediating between them compensates for the effect of the modification of the rotation on the translation. It should be noted that the condition that the first and the last frames input into the PEB must be the same necessitates making a copy of the first frame and inserting it at the end of the sequence before performing registration, where the difference between the poses of the camera between the first frame and the original last one needs not to be larger than the threshold that could be tolerated by the registration algorithm when finding the transformation mapping one to the other, meaning that the scanning process should finish at a pose close enough to its starting one.
The underlying methodology of the PEB will be described in mathematical terms in what follows. Assuming that n distinct frames, being each stood for by a set F_i, $i = 1,2,\dots,n$, exist in the whole sequence, the points in the i^{th} one, i.e. F_i, are represented through a Cartesian coordinate system $\mathscr{F}_i$ which is defined by the origin O_i and the axes X_i, Y_i, and Z_i, as follows:

$$\forall j,\, j \in \{1,2,\dots,n_i\} \Longrightarrow \left[\boldsymbol{p}_{i_{j(3\times1)}}\right]_{\mathscr{F}_i} \in F_i, \tag{1}$$

where $\boldsymbol{p}_{i_j}$ represents the position vector of the j^{th} point in F_i, namely, P_{i_j}, and n_i is the total number of the points in F_i. It should be noted that i and j are dummy variables to be changed throughout the paper. Then, having in mind that a copy of the first frame, $F_{n+1} = F_1$, has been added to the end of the sequence, i.e. there are now $n+1$ frames in the sequence, upon constructing the homogeneous coordinates of P_{i_j}, namely, $\left\{\boldsymbol{p}_{i_j}\right\}_{\mathscr{F}_i}$, as follows:

$$\begin{gathered}\forall i \forall j, i \in \{1,2,\dots,n+1\} \wedge j \in \{1,2,\dots,n_i\} \Longrightarrow \\ \left\{\boldsymbol{p}_{i_j}\right\}_{\mathscr{F}_i} = \left[\left[\boldsymbol{p}_{i_j}\right]_{\mathscr{F}_i}^{\mathrm{T}} \quad 1\right]^{\mathrm{T}},\end{gathered} \tag{2}$$

the homogeneous transformation matrix $\boldsymbol{T}_{i_{(4\times4)}}$, which has been obtained by the registration algorithm, maps the homogeneous coordinates of every point in the $(i+1)^{\text{th}}$ frame, $\left\{\boldsymbol{p}_{i+1_j}\right\}_{\mathscr{F}_{i+1}}$, from its native coordinate system, being $\mathscr{F}_{i+1}$, to that of a point $\left\{\boldsymbol{p}^*_{i+1_{j(3\times1)}}\right\}_{\mathscr{F}_i}$ supposed to match it in the preceding one, namely, $\mathscr{F}_i$, meaning that:

$$\begin{gathered}\forall i \forall j, i \in \{1,2,\dots,n\} \wedge j \in \{1,2,\dots,n_{i+1}\} \Longrightarrow \\ \left\{\boldsymbol{p}^*_{i+1_j}\right\}_{\mathscr{F}_i} = \boldsymbol{T}_i \left\{\boldsymbol{p}_{i+1_j}\right\}_{\mathscr{F}_{i+1}},\end{gathered} \tag{3}$$

where:

$$\boldsymbol{T}_i = \begin{bmatrix} \boldsymbol{Q}_{i(3\times3)} & \boldsymbol{t}_{i(3\times1)} \\ \boldsymbol{0}_{(1\times3)} & 1 \end{bmatrix}, \tag{4}$$

in which $\boldsymbol{Q}_i$ and $\boldsymbol{t}_i$ stand for a rotation matrix and a translation vector, respectively, and $\boldsymbol{0}$ denotes a vector of all-zeros. If the camera poses have been calculated flawlessly, the latter transformation will map the homogeneous coordinates of every point from the corresponding coordinate system to its own representation in the coordinate system associated with the previous frame, i.e. ideally, $\left\{\boldsymbol{p}^*_{i+1_j}\right\}_{\mathscr{F}_i}$ should be equivalent to $\boldsymbol{T}_i \left\{\boldsymbol{p}_{i+1_j}\right\}_{\mathscr{F}_{i+1}} = \left\{\boldsymbol{p}_{i+1_j}\right\}_{\mathscr{F}_i}$, which is usually not the case, due to the errors having, as aforementioned, arisen because of a variety of reasons. The foregoing inconsistency explains the cause of the loop closure error, i.e. the accumulation of the error throughout the transformations prevents the ends of a closed loop of the frames from coinciding with each other at the pose they are supposed to do.
In order to define measures describing the overall error, which is tantamount to the loop closure error and should be back-propagated so as to correct the loop closure by modifying the transformations, one could find the overall transformation supposedly mapping the points from the coordinate system associated with the newly inserted last frame to that of the first frame, which are in fact the same, and upon noticing that ideally it has to become an identity homogeneous transformation, deriving the loop closure pose error from it. In other words, the accumulation of the first to the n^{th} homogeneous transformations, namely, $\boldsymbol{T}_{T_{(4\times4)}}$, which can be found as follows:

$$\boldsymbol{T}_T = \prod_{i=1}^{n} \boldsymbol{T}_i, \tag{5}$$

can be considered as a homogeneous transformation constructed on the basis of the parameters standing for the loop closure error, such that with the following representation:

$$\boldsymbol{T}_T = \begin{bmatrix} \boldsymbol{Q}_{T(3\times3)} & \boldsymbol{t}_{T(3\times1)} \\ \boldsymbol{0} & 1 \end{bmatrix}, \tag{6}$$

$\boldsymbol{Q}_T$ and $\boldsymbol{t}_T$ denote a cumulative rotation matrix and a cumulative translation vector, respectively, which could be utilized to extract the rotation and translation loop closure errors.
For extracting the loop closure error correction terms based on the overall homogeneous transformation, i.e. $\boldsymbol{T}_T$, first, Eq. (5) can be expanded through substituting each individual

homogeneous transformation by the expression describing it from Eq. (4) in order to find $\boldsymbol{Q}_T$ and $\boldsymbol{t}_T$ in Eq. (6), as follows:

$$\boldsymbol{Q}_T = \prod_{i=1}^{n} \boldsymbol{Q}_i, \quad \boldsymbol{t}_T = \sum_{i=1}^{n} \left(\prod_{j=0}^{i-1} \boldsymbol{Q}_j \right) \boldsymbol{t}_i, \tag{7}$$

where $\boldsymbol{Q}_{0_{(3\times3)}} = \boldsymbol{I}_3$ is an identity matrix.

In fact, the goal of the PEB is to find modified rotation matrices $\hat{\boldsymbol{Q}}_{i_{(3\times3)}}$ and translation vectors $\hat{\boldsymbol{t}}_{i_{(3\times1)}}$, according to the original ones $\boldsymbol{Q}_i$ and $\boldsymbol{t}_i$, respectively, $i = 1,2,\ldots,n$, based on their proportional contributions to the overall rotation matrix $\boldsymbol{Q}_T$ and the overall translation vector $\boldsymbol{t}_T$, respectively, such that the resulting homogeneous transformation matrices $\hat{\boldsymbol{T}}_{i_{(4\times4)}}$ constructed as follows:

$$\hat{\boldsymbol{T}}_i = \begin{bmatrix} \hat{\boldsymbol{Q}}_i & \hat{\boldsymbol{t}}_i \\ \boldsymbol{0} & 1 \end{bmatrix}, \tag{8}$$

would overall accumulate the identity homogeneous transformation, represented by the 4×4 identity matrix $\boldsymbol{I}_4$, meaning that:

$$\hat{\boldsymbol{T}}_T = \begin{bmatrix} \hat{\boldsymbol{Q}}_T & \hat{\boldsymbol{t}}_T \\ \boldsymbol{0} & 1 \end{bmatrix} = \prod_{i=1}^{n} \hat{\boldsymbol{T}}_i = \boldsymbol{I}_4, \tag{9}$$

where $\hat{\boldsymbol{Q}}_T$ and $\hat{\boldsymbol{t}}_T$ denote the corrected cumulative rotation matrix and the corrected cumulative translation vector, respectively.

In the context of the transformation correction procedure of the PEB, first, the rotation matrices are corrected. The purpose is to modify each rotation matrix $\boldsymbol{Q}_i$, $i = 1,2,\ldots,n$, such that the corrected overall rotation would become an identity rotation. In order to do so, a set of rotation correction matrices $\boldsymbol{Q}_{e_{i_{(3\times3)}}}$, $i = 1,2,\ldots,n$, should be calculated to be incorporated into the construction of the corresponding corrected rotation matrices. In what follows, the proposed mathematical framework for achieving the latter goal is explained, where the virtue of the fact that the inverse of every rotation matrix is equal to its own transpose has been resorted to for the sake of reducing the consequent computational cost.

In order to format the structure of the rotation correction procedure, an expression for each corrected rotation matrix, $\hat{\boldsymbol{Q}}_i$, $i = 1,2,\ldots,n$, in terms of the rotation and rotation correction matrices, is first derived in a way that would enable the algorithm to manipulate the corrected overall rotation, which must become identity, by adjusting the rotation correction matrices. To this end, the cumulative rotation matrices $\boldsymbol{Q}_{c_{i_{(3\times3)}}}$, $i = 1,2,\ldots,n$, and their corrected counterparts $\hat{\boldsymbol{Q}}_{c_{i_{(3\times3)}}}$ are defined as follows:

$$\boldsymbol{Q}_{c_i} = \prod_{j=1}^{i} \boldsymbol{Q}_j, \quad \hat{\boldsymbol{Q}}_{c_i} = \prod_{j=1}^{i} \hat{\boldsymbol{Q}}_j. \tag{10}$$

Then by noticing that each corrected cumulative rotation matrix should assimilate all the associated rotation correction matrices, meaning that:

$$\forall i,\, i \in \{0,1,\ldots,n\} \Longrightarrow \hat{\boldsymbol{Q}}_{c_i} = \left(\prod_{j=0}^{i} \boldsymbol{Q}_{e_j} \right) \boldsymbol{Q}_{c_i}, \tag{11}$$

where $\boldsymbol{Q}_{e_0} = \boldsymbol{I}_3$, a closed-form expression can be found for the corrected rotation matrices $\hat{\boldsymbol{Q}}_i$, $i = 1,2,\ldots,n$, as follows:

$$\begin{gathered} \forall i,\, i \in \{1,2,\ldots,n\} \Longrightarrow \hat{\boldsymbol{Q}}_{c_i} = \hat{\boldsymbol{Q}}_{c_{i-1}} \hat{\boldsymbol{Q}}_i \Longrightarrow \hat{\boldsymbol{Q}}_i = \hat{\boldsymbol{Q}}_{c_{i-1}}^{\mathrm{T}} \hat{\boldsymbol{Q}}_{c_i} = \\ \left(\left(\prod_{j=0}^{i-1} \boldsymbol{Q}_{e_j} \right) \boldsymbol{Q}_{c_{i-1}} \right)^{\mathrm{T}} \left(\prod_{j=0}^{i} \boldsymbol{Q}_{e_j} \right) \boldsymbol{Q}_{c_i} = \\ \boldsymbol{Q}_{c_{i-1}}^{\mathrm{T}} \left(\prod_{j=0}^{i-1} \boldsymbol{Q}_{e_j} \right)^{\mathrm{T}} \left(\prod_{j=0}^{i-1} \boldsymbol{Q}_{e_j} \right) \boldsymbol{Q}_{e_i} \boldsymbol{Q}_{c_i} = \boldsymbol{Q}_{c_{i-1}}^{\mathrm{T}} \boldsymbol{Q}_{e_i} \boldsymbol{Q}_{c_i}, \end{gathered} \tag{12}$$

where $\boldsymbol{Q}_{c_0} = \hat{\boldsymbol{Q}}_{c_0} = \boldsymbol{I}_3$.

Assuming that the accumulation of the above rotation correction matrices is represented as $\boldsymbol{Q}_{e_{T_{(3\times3)}}}$, which is calculated as follows:

$$\boldsymbol{Q}_{e_T} = \prod_{i=1}^{n} \boldsymbol{Q}_{e_i}, \tag{13}$$

one has:

$$\hat{\boldsymbol{Q}}_T = \left(\prod_{i=1}^{n} \boldsymbol{Q}_{e_i} \right) \boldsymbol{Q}_T = \boldsymbol{I}_3 \Longrightarrow \prod_{i=1}^{n} \boldsymbol{Q}_{e_i} = \boldsymbol{Q}_T^{\mathrm{T}}, \tag{14}$$

meaning that $\boldsymbol{Q}_{e_T}$ shall become the inverse of the overall rotation matrix, i.e. $\boldsymbol{Q}_T$, meaning that the individual rotation correction matrices can be constructed such that they stand for rotations around the same axis as that of $\boldsymbol{Q}_T$, but lead to rotation angles which accumulate the negation of that of $\boldsymbol{Q}_T$. Therefore, assuming that $\boldsymbol{Q}_T$ is represented by $\boldsymbol{e}_T$ and ϕ_T as the unit vector along the rotation axis and the rotation angle, respectively, using the same rotation axis and the following rotation angles:

$$\forall i,\, i \in \{1,2,\ldots,n\} \Longrightarrow \phi_{e_i} = -\frac{|\phi_i|}{\sum\limits_{i=1}^{n} |\phi_i|} \phi_T, \tag{15}$$

meaning that:

$$\sum_{i=1}^{n} \phi_{e_i} = -\sum_{i=1}^{n} \frac{|\phi_i|}{\sum\limits_{i=1}^{n} |\phi_i|} \phi_T = -\phi_T, \tag{16}$$

the corresponding rotation correction matrices, i.e. $\boldsymbol{Q}_{e_i}$, can be obtained, where ϕ_i stands for the rotation angle associated with $\boldsymbol{Q}_i$, the ratio $\frac{|\phi_i|}{\sum_{i=1}^{n} |\phi_i|}$ being meant to make each rotation correction matrix proportional to the contribution of the corresponding original rotation matrix to the overall one.

However, the correction of the rotation matrices affects the translation vectors as well. More clearly, while the rotations are being fixed, further drift will be introduced into the alignments, which appears as a higher level of error in the translations. Therefore, the translations are first revised such that the effect of the changes in the rotation would be minimized. In order to do so, it is assumed that the average position of the points from a given frame must be affected in the same manner before and after revising the corresponding rotation, which can be mathematically represented as follows:

$$\begin{gathered}\forall i,\, i \in \{1,2,\dots,n\} \Longrightarrow \boldsymbol{Q}_i \frac{\sum_{j=1}^{n_{i+1}} \boldsymbol{p}_{i+1_j}}{n_{i+1}} + \boldsymbol{t}_i = \\ \hat{\boldsymbol{Q}}_i \frac{\sum_{j=1}^{n_{i+1}} \boldsymbol{p}_{i+1_j}}{n_{i+1}} + \boldsymbol{u}_i \Longrightarrow \boldsymbol{u}_i = \boldsymbol{t}_i + \left(\boldsymbol{Q}_i - \hat{\boldsymbol{Q}}_i\right) \frac{\sum_{j=1}^{n_{i+1}} \boldsymbol{p}_{i+1_j}}{n_{i+1}},\end{gathered} \tag{17}$$

where $\boldsymbol{u}_i$, $i = 1,\dots,n$, is the revised translation vector. Thus, the overall translation based on the newly obtained rotation matrices and translation vectors could be found based on Eq. (7), as follows:

$$\boldsymbol{v}_T = \sum_{i=1}^{n} \left(\prod_{j=0}^{i-1} \hat{\boldsymbol{Q}}_j \right) \boldsymbol{u}_i = \hat{\boldsymbol{Q}}_{c_{j-1}} \boldsymbol{u}_i, \tag{18}$$

Similarly to what preceded regarding correcting the rotations, when it comes to doing so on the translations, the above vector, i.e. $\boldsymbol{v}_T$, can be considered as the new overall translation error, since given the fact that the sequences consisting of $n+1$ frames stands for a fully closed loop with identical first and last frames, if the transformations had been calculated perfectly, then it would need to become zero.

Thus the task of correcting the translations will consist of distributing the aforementioned overall translation error to the individual translation vectors, $\boldsymbol{u}_i$, proportionally to their contributions. The latter are represented as follows:

$$\begin{gathered}\forall i,\, i \in \{1,2,\dots,n\} \Longrightarrow \boldsymbol{v}_i = \begin{bmatrix} v_{i_1} & v_{i_2} & v_{i_3} \end{bmatrix}^{\mathrm{T}} = \\ \hat{\boldsymbol{Q}}_{c_{j-1}} \boldsymbol{u}_i.\end{gathered} \tag{19}$$

Thus the translation correction vectors can be constructed as follows:

$$\begin{gathered}\forall i,\, i \in \{1,2,\dots,n\} \Longrightarrow \boldsymbol{t}_{e_i} = \\ -\begin{bmatrix} |v_{i_1}| & |v_{i_2}| & |v_{i_3}| \end{bmatrix}^{\mathrm{T}} \oslash \sum_{j=1}^{n} \left(\begin{bmatrix} |v_{j_1}| & |v_{j_2}| & |v_{j_3}| \end{bmatrix}^{\mathrm{T}} \right) \odot \boldsymbol{v}_T,\end{gathered} \tag{20}$$

which are proportional to the corresponding contributions $\boldsymbol{v}_i$ to the overall error, i.e. $\boldsymbol{v}_T$, and their cumulative value is its negation, being realized as follows:

$$\begin{gathered}\sum_{i=1}^{n} \boldsymbol{t}_{e_i} = \\ -\sum_{i=1}^{n} \left(\begin{bmatrix} |v_{i_1}| & |v_{i_2}| & |v_{i_3}| \end{bmatrix}^{\mathrm{T}} \oslash \sum_{j=1}^{n} \left(\begin{bmatrix} |v_{j_1}| & |v_{j_2}| & |v_{j_3}| \end{bmatrix}^{\mathrm{T}} \right) \odot \boldsymbol{v}_T \right) = \\ -\sum_{i=1}^{n} \left(\begin{bmatrix} |v_{i_1}| & |v_{i_2}| & |v_{i_3}| \end{bmatrix}^{\mathrm{T}} \right) \oslash \sum_{j=1}^{n} \left(\begin{bmatrix} |v_{j_1}| & |v_{j_2}| & |v_{j_3}| \end{bmatrix}^{\mathrm{T}} \right) \odot \boldsymbol{v}_T = \\ -\boldsymbol{v}_T.\end{gathered} \tag{21}$$

Subsequently, the revised contributions to the overall translation are obtained as follows:

$$\forall i,\, i \in \{1,2,\dots,n\} \Longrightarrow \hat{\boldsymbol{v}}_i = \boldsymbol{v}_i + \boldsymbol{t}_{e_i} = \hat{\boldsymbol{Q}}_{c_{j-1}} \boldsymbol{u}_i + \boldsymbol{t}_{e_i}. \tag{22}$$

Lastly, in order to incorporate the above conclusion into the calculation of the corrected translation vectors, Eq. (19) is recalled, and the relationship between the translation contribution vectors found through Eq. (22) and the corrected translation vectors is establishes as follows:

$$\forall i,\, i \in \{1,2,\dots,n\} \Longrightarrow \hat{\boldsymbol{v}}_i = \hat{\boldsymbol{Q}}_{c_{j-1}} \hat{\boldsymbol{t}}_i, \tag{23}$$

based on which, utilizing Eq. (22), the corrected translation vectors can be obtained as follows:

$$\begin{gathered}\forall i,\, i \in \{1,2,\dots,n\} \Longrightarrow \hat{\boldsymbol{t}}_i = \hat{\boldsymbol{Q}}_{c_{j-1}}^{\mathrm{T}} \hat{\boldsymbol{v}}_i = \\ \hat{\boldsymbol{Q}}_{c_{j-1}}^{\mathrm{T}} \left(\hat{\boldsymbol{Q}}_{c_{j-1}} \boldsymbol{u}_i + \boldsymbol{t}_{e_i} \right) = \boldsymbol{u}_i + \hat{\boldsymbol{Q}}_{c_{j-1}}^{\mathrm{T}} \boldsymbol{t}_{e_i}.\end{gathered} \tag{24}$$

3. Experimental Results and Discussion

As mentioned before, the PEB has been devised such that given the assumption that the sequence showing a sequence or object possesses first and last frames which have been taken from similar poses, the raw transformations which have been found using an alignment method could be revised, thereby removing the apparent overall misalignment from the point cloud resulting from merging the individual depth maps.

A typical task to be left upon the PEB could be to modify the preliminary outcome of a standard 3D reconstruction pipeline consisting of filming an object while being rotated on top of a turntable, using an RGB-D sensor such as the Microsoft Kinect 2 RGB-D camera [29]. In such a scenario, although the individual transformations may appear to be reasonable, the slight misalignments present in them usually accumulate, and appear as a noticeable diversion between the parts of the reconstructed point cloud corresponding to the initial and final images from the sequence.

Thus, the aim of the PEB would be to distribute the overall misalignment to the individual transformations, so that the structure of the object would be maintained. Similarly, if a scene, e.g., a rectangular room, has been filmed instead, the PEB can be employed to modify an initial reconstructed point cloud.

However, filming scenes, as opposed to objects, usually takes a higher number of frames, which is due to the fact that every pair of consecutive frames fed into an alignment algorithm need to have been taken such that the pose of the camera in the second frame relatively to that of the first one would lead to a reasonable difference, in order for the optimization routine to converge with a tolerable level of error. More clearly, if the pairs of consecutive frames are too different from each other, then the transformation returned by the alignment algorithm may be wrong enough for the PEB to perform weakly in terms of compensating for the present misalignments.

Thus due to the higher number of frames, and as a result, transformations reconstructing a scene are usually associated with higher levels of misalignments, handling which would be more challenging for the PEB. Therefore, in this paper, it is assumed that if the capabilities of the PEB are verified in modifying reconstructions of scenes, its reliability in reconstructing objects would be implied as well. Based on the latter

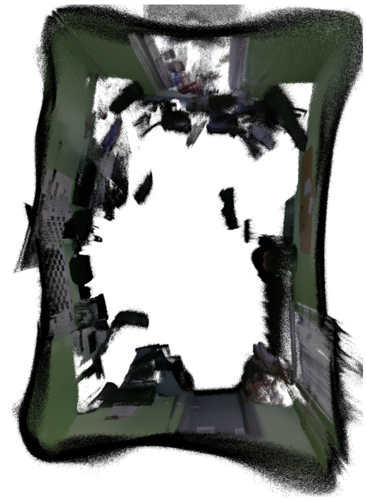

Fig.1. The first initial reconstruction result.

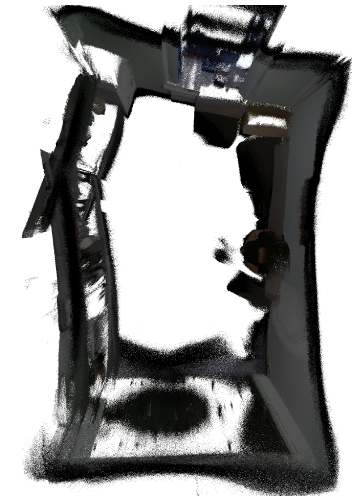

Fig.2. The second initial reconstruction result.

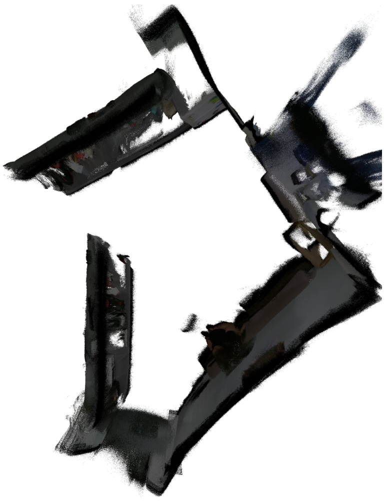

Fig.3. The third initial reconstruction result.

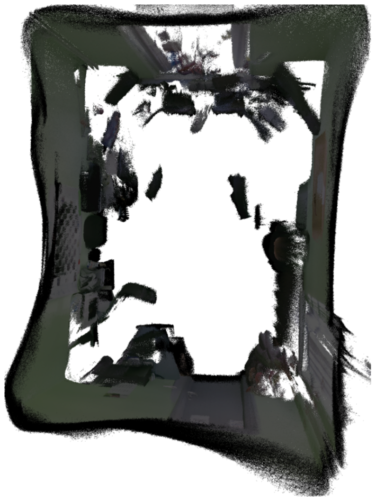

Fig.4. The corrected counterpart of the result shown in Fig. 1.

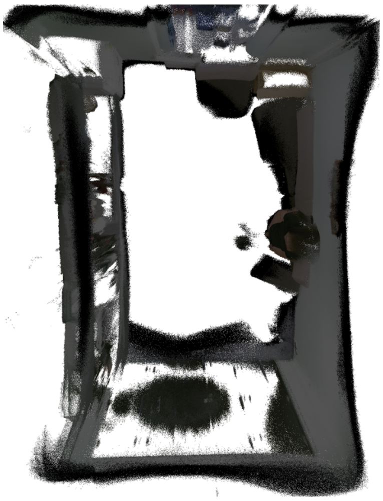

Fig.5. The corrected counterpart of the result shown in Fig. 2.

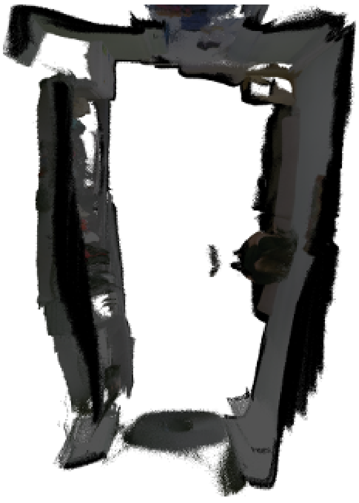

Fig.6. The corrected counterpart of the result shown in Fig. 3.

reasoning, in this paper, the performance of the PEB is examined only in the context of reconstructing scenes, where rectangular rooms are considered as case-studies. The sequences have been taken using Kinect 2.

While filming, it is ensured that the camera will stop at a pose which is close to the starting one, being demanded by the PEB. In order to evaluate the performance of the PEB under arbitrary conditions, and examine its robustness, various trajectories have been considered for the motions of the camera, which include wavy patterns and movements of the camera in the opposite direction of the general trend. The latter is necessary for realizing whether the PEB could handle cases where the experimental setup requires the user to perform indisciplined movements, e.g., due to the restrictions caused by the lengths of the cables connecting the camera and the computer to each other, as well as to the electricity outlet.

The sequences considered for the purpose of evaluating the PEB consist of series of RGB-D frames taken while the user moves throughout the room and holds the camera such that it is facing a part of one of the walls at all of the timestamps, which leads to around 500 frames for a 3×4 m rectangular room, using a frame-rate of 30 Hz. The foregoing frames have all been intentionally kept and fed into the reconstruction pipeline, i.e. downsampling has been avoided, for the sake of introducing a strong amount of misalignment, thereby verifying the robustness of the PEB.

The initial results of reconstructing the sequences using the Iterative Closest Point (ICP) [30] algorithm are shown in Figs. 1 through 3, whose counterparts which have been improved through applying the PEB can be found in Figs. 4 through 6, in the same order.

It is noteworthy that the end of the sequence is determined by taking the first frame of the sequence and iteratively comparing it to the frames from the end of the sequence, starting from the last frame and going backwards. The frame similarity is determined by SSIM and once a frame with highest similarity is found, the process is stopped. If the first frame is placed at the end of a partial loop, the algorithm will still consider the sequence as a full loop, meaning that the final results will be incorrect. Currently the implementation, does not check for such cases.

As it could be seen from the results shown in the aforementioned figures, although the PEB incurs a negligible computational load, it provides a reliable platform for revising the transformations returned by a typical alignment algorithm such as ICP, which leads to smoothly distributing the overall error to the relative poses, thereby obtaining a visually appealing representation of the scene that believably corresponds to the expected 3D structure.

4. Conclusion

In this paper, a fast loop closure correction algorithm, namely, Proportional Error Back-Propagation (PEB), was proposed, which performs the task in a fraction of a second on a sequence of frames meant to reconstruct a 3D representation of a scene, where the overall transformation error is distributed to the individual relative poses proportionally to their contributions to the cumulative transformation. The underlying assumption was that the initial and final frames from the sequence need to be taken at similar poses of the camera, which makes it a suitable choice for reconstructing a room, where the camera films its surroundings trying to stop at a similar pose as it had started. The proposed method was verified in terms of visual and computational efficiency through applying it to a variety of sequences. The future works may involve incorporating the possibility of correcting the transformations in case multiple closures appear in a given sequence of frames.

Acknowledgement

This work has been partially supported by Estonian Research Council Grants (PUT638), The Scientific and Technological Research Council of Turkey (TÜBiTAK) (Proje 1001 - 116E097) , the Estonian Centre of Excellence in IT (EXCITE) funded by the European Regional Development Fund and the European Network on Integrating Vision and Language (iV&L Net) ICT COST Action IC1307.

References

[1] Berg, L.P., Vance, J.M. (2017). Industry use of virtual reality in product design and manufacturing: A survey. *Virtual Reality* 21(1), 1–17.

[2] Avots, E., Daneshmand, M., Traumann, A., Escalera, S., Anbarjafari, G. (2016). Automatic garment retexturing based on infrared information. *Computers & Graphics*, 59, 28–38.

[3] Anbarjafari, G., Haamer, R.E., Lusi, I., Tikk, T., Valgma, L. (2018). 3D face reconstruction with region based best fit blending using mobile phone for virtual reality based social media. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 66, 1–11.

[4] Daneshmand, M., Helmi, A., Avots, E., Noroozi, F., Alisinanoglu, F., Arslan, H.S., Gorbova, J., Haamer, R.E., Ozcinar, C., Anbarjafari, G. (2018). 3D scanning: A comprehensive survey. *arXiv:1801.08863 [cs.CV].*

[5] Bailey, T., Durrant-Whyte, H. (2006). Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine* 13(3), 108–117.

[6] Sim, R., Roy, N. (2005). Global a-optimal robot exploration in SLAM. In *IEEE International Conference on Robotics and Automation (ICRA 2005).* IEEE, 661–666.

[7] Tomono, M. (2009). Robust 3d SLAM with a stereo camera based on an edge-point ICP algorithm. In *International Conference on Robotics and Automation (ICRA'09).* IEEE, 4306–4311.

[8] Valgma, L., Daneshmand, M., Anbarjafari, G. (2016). Iterative closest point based 3D object reconstruction using RGB-D acquisition devices. In *24th Signal Processing and Communication Application Conference (SIU).* IEEE, 457–460.

[9] Beardsley, P.A., Zisserman, A., Murray, D.W. (1997). Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision*, 23(3), 235–259.

[10] Turner, D., Lucieer, A., Watson, C. (2012). An automated technique for generating georectified mosaics from ultra-high resolution unmanned aerial vehicle (UAV) imagery, based on structure from motion (SFM) point clouds. *Remote Sensing* 4(5), 1392–1410.

[11] Fitzgibbon, A.W., Zisserman, A. (1998). Automatic camera recovery for closed or open image sequences. In: *Computer Vision – ECCV'98*. Springer, 311–326.

[12] Curless, B., Levoy, M. (1996). A volumetric method for building complex models from range images. In: *23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96). ACM*, 303–312.

[13] Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D. (2010). RGB-D mapping: Using depth cameras for dense 3d modeling of indoor environments. In: *Experimental Robotics: 12th International Symposium on Experimental Robotics.* Springer, STAR 79, 477-491.

[14] Frank Steinbrucker, Christian Kerl, J.S., Cremers, D. (2013). Large-scale multi-resolution surface reconstruction from RGB-Dsequences. In: *IEEE International Conference on Computer Vision (ICCV).* IEEE, 3264-3271.

[15] Liu, T., Zhang, X., Wei, Z., Yuan, Z. (2013). A robust fusion method for RGB-D SLAM. In: *Chinese Automation Congress (CAC).* IEEE, 474–481.

[16] Shiratori, T., Berclaz, J., Harville, M., Shah, C., Li, T., Matsushita, Y., Shiller, S. (2015). Efficient large-scale point cloud registration using loop closures. In: *International Conference on 3D Vision (3DV).* IEEE, 232–240.

[17] Whelan, T., Kaess, M., Johannsson, H., Fallon, M., Leonard, J.J., McDonald, J. (2015). Real-time large-scale dense RGB-D SLAM with volumetric fusion. *The International Journal of Robotics Research*, 34(4-5), 598–626.

[18] Kaess, M., Ranganathan, A., Dellaert, F. (2008). ISAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6), 1365–1378.

[19] Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J.J., Dellaert, F. (2011). ISAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2), 216-235.

[20] Wang, Y., Zhang, Q., Zhou, Y. (2015). Dense 3D mapping for indoor environment based on kinect-style depth cameras. In: *Robot Intelligence Technology and Applications 3.*. Springer, 317–330.

[21] Grisetti, G., Stachniss, C., Grzonka, S., Burgard, W. (2007). TORO - Tree-based netwORk Optimizer. `https://openslam.org/toro.html`.

[22] Wu, J., Cui, Z., Sheng, V.S., Zhao, P., Su, D., Gong, S. (2013). A comparative study of SIFT and its variants. *Measurement Science Review*, 13(3), 122–131.

[23] Daneshmand, M., Aabloo, A., Ozcinar, C., Anbarjafari, G. (2016). Real-time, automatic shape-changing robot adjustment and gender classification. *Signal, Image and Video Processing*, 10(4), 753–760.

[24] Kim, K., Lawrence, R.L., Kyllonen, N., Ludewig, P.M., Ellingson, A.M., Keefe, D.F. (2017). Anatomical 2D/3D shape-matching in virtual reality: A user interface for quantifying joint kinematics with radiographic imaging. In *IEEE Symposium on 3D User Interfaces (3DUI).*, IEEE, 243–244.

[25] Lüsi, I., Anbarjafari, G., Meister, E. (2015). Real-time mimicking of estonian speaker's mouth movements on a 3D avatar using Kinect 2. In *International Conference on Information and Communication Technology Convergence (ICTC)*, IEEE, 141–143.

[26] Kühnapfel, U., Cakmak, H.K., Maaß, H. (2000). Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computers & Graphics*, 24(5), 671–682.

[27] Traumann, A., Daneshmand, M., Escalera, S., Anbarjafari, G. (2015). Accurate 3D measurement using optical depth information. *Electronics Letters*, 51(18), 1420–1422.

[28] Daneshmand, M., Aabloo, A., Anbarjafari, G. (2015). Size-dictionary interpolation for robot's adjustment. *Frontiers in Bioengineering and Biotechnology*, 3, 63.

[29] Microsoft Corporation. Kinect for Windows. `https://developer.microsoft.com/en-us/windows/kinect`.

[30] Besl, P.J., McKay, N.D. (1992). Method for registration of 3-D shapes. In *Robotics-DL tentative*, SPIE, 586–606.

Received October 01, 2017.
Accepted April 25, 2018.