Online 3D Reconstruction and 6-DoF Pose Estimation for RGB-D Sensors

Hyon Lim*, Jongwoo Lim†, and H. Jin Kim*

Seoul National University*, Hanyang University†
Seoul, Korea

Abstract. In this paper, we propose an approach to Simultaneous Localization and Mapping (SLAM) for RGB-D sensors. Our system computes 6-DoF pose and sparse feature map of the environment. We propose a novel keyframe selection scheme based on the Fisher information, and new loop closing method that utilizes feature-to-landmark correspondences inspired by image-based localization. As a result, the system effectively mitigates drift that is frequently observed in visual odometry system. Our approach gives lowest relative pose error amongst any other approaches tested on public benchmark dataset. A set of 3D reconstruction results on publicly available RGB-D videos are presented.

Keywords: Simultaneous Localization and Mapping, RGB-D SLAM

1 Introduction

The goal of *online 3D reconstruction and 6-DoF pose estimation*, also known as Simultaneous Localization and Mapping (SLAM) is to incrementally build a 3D model of the surrounding environment while concurrently localizing the camera. This has been a key technology for autonomous navigation of robots and many useful applications [6, 10, 15, 22, 11, 26, 12]. To this end, selection of a keyframe and finding inter-keyframe geometric relationships, are one of the most important parts. However, most existing systems select keyframe based on heuristics, such as fixed time or distance intervals, and find geometric relation only between adjacent keyframes.

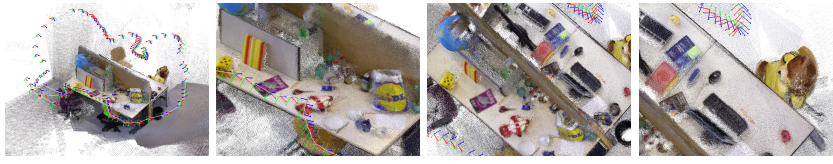


Fig. 1: The 3D model of fr3/office reconstructed by the proposed method. This reconstruction is obtained by registering the full point clouds of all keyframes transformed according to their keyframe poses computed by the proposed method. Position and orientation of keyframes are denoted by coordinate axis that is color coded by red, green and blue (X-Y-Z order) in below four images.

In this paper, we discuss two essential problems of online 3D reconstruction and camera tracking, which critically affect the quality and speed of the reconstruction. The first problem is the online *keyframe selection*. The keyframes are the representative images of the scene chosen among the input frames, and they are used in building the model and optimizing the structure [10, 22]. Since the reconstruction is performed on top of the selected keyframes, selecting proper keyframes is a critical task, but how to choose good keyframes has not been studied extensively so far. Several heuristic methods have been widely used, such as using fixed time intervals or using fixed distance or rotation threshold. These rule-of-thumb methods have introduced somewhat ad-hoc parameters, and they tend to generate more than necessary keyframes to model the scene. Instead, we propose an information theoretic approach to measure informativeness of the current estimate to decide whether to put it as a keyframe or not. Detailed method is described in Section 4.1.

The other problem we are tackling in this paper is the *loop closing*. The loop closing is the task of finding new geometric relationships between keyframes, which was not available from temporal incremental motion estimation. By closing loops, the uncertainty of reconstruction can be reduced and the model quality can be improved. However, the existing methods using visual features [10, 11, 4, 15, 22] search only keyframe-to-keyframe loops. However, these existing approaches have been overlooked normal frames between keyframes. As the number of normal frames is significantly greater than the number of keyframes, finding a loop on normal frames will increase the chance of finding of better loops. Other approaches such as [10, 22] search metric loop closures by assuming some motion prior to be available. However, such metric loop closure is less likely to be successful in most online 3D reconstruction scenarios due to the measurement noise and pose drift.

To address the problems above, we propose a method which is not only utilize an appearance-based method [3] but also bridges disconnected feature-to-landmark relations inspired by image-based localization. The key distinction of proposed method compared to the existing online 3D reconstruction methods [10, 11, 4, 15, 22] is that we seek direct link between the 2D features and 3D landmarks at *every input image*, and add the image as a new keyframe on-the-fly when a good match is found. Most existing methods ignores the non-keyframe images (i.e., normal frames) mainly due to computational overhead, and only utilized the keyframe-to-keyframe matching. However, we address this issue by distributing computational load by extracting specified number of descriptors on keypoints at each frame. This is inspired by work of real-time image-based localization [14]. The difference between the existing approaches and the proposed method is visualized in Fig. 2.

We developed a full 6-DoF online 3D reconstruction system with proposed methods to demonstrate the effectiveness of the proposed approaches. An example input and output of developed system is shown in Fig 1. We have evaluated our approach on public RGB-D datasets recorded in various environment [24]. We are able to achieve high quality of 3D reconstruction with less number of

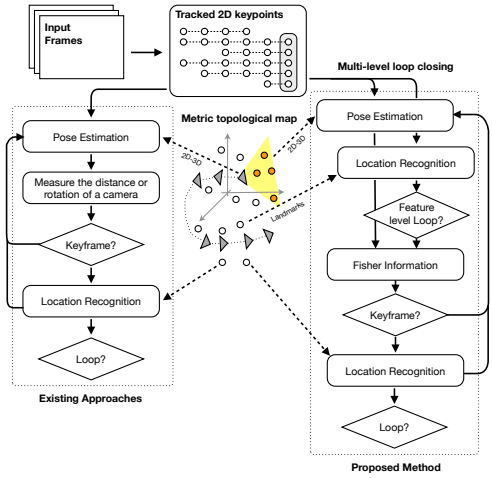


Fig. 2: The block diagram of the core steps in the existing methods (left side) and the proposed method (right side). Note that loop closure test is performed at every frame (while existing approaches perform at keyframe creation) and keyframe selection is based on the Fisher information in our method. Refer to the text for more detail.

keyframes than existing keyframe selection scheme. Most of evaluation results show less than 10 cm in root mean square error (RMSE) of absolute trajectory error. To summarize, our key contributions are:

- A novel keyframe selection scheme based on Fisher information of the tracked camera pose. It measures informativeness of the given frame. As a result, a keyframe is added only if it has enough information, thus it can select keyframes adaptively and creates less number of keyframes to build similar quality reconstructions as compared in Section 5.
- A new loop closing strategy for online 3D reconstruction. This consists of two types of match. The one matching process seeks best similar view of current keyframe which only happens at keyframe creation process (keyframe-to-keyframe), while the another process searches best 3D landmarks in the 2D keypoints in an image (feature-to-landmark) at every input frame. The effectiveness of proposed method is tested our own built online 3D reconstruction system with real experiments with ground truth comparison. We also evaluated the proposed approach with other method.

2 Related Work

2.1 Online 3D Reconstruction

The online 3D reconstruction is a task that reconstructs surrounding environments in 3D by utilizing input video sequences. A common components of online 3D reconstruction are feature point tracking, which takes care of the temporal keypoint relationship between frames, and bundle adjustment for accurate 3D

reconstruction [25]. In the existing video-based reconstruction [11, 26, 12], the quality of frame-by-frame feature track seriously affects the quality of reconstruction. This is prone to image noise, object occlusions, illumination change, large-motion, which easily causes occasional feature dropout and distraction. If a feature track is interrupted by these disturbances, the existing methods create a new feature track even though it is a same physical point of lost feature. We argue that this non-consecutive track should be bridged correctly for accurate 3D reconstruction. However, most of approaches have no explicit answer to this question because it is not obvious how to insert this discontinuous track as observations. We will discuss this further in Section 4.3.

2.2 Keyframe Selection

Several selection schemes have been proposed to build a map with a smaller set of keyframes. The naïve choice is to choose every n-th frame. However, this will create unnecessary keyframes during stationary motion. Another popular option is to choose keyframe based on *distance or orientation change threshold* since last keyframe [10, 11, 15, 22]. This method will sample Euclidean space evenly, but it does not consider sensor accuracy. For example, if a sensor quality is good enough to measure wide range of the environment, this method will create redundant keyframes. Henry *et al.* [6] selects keyframe based on the number of inliers during RANSAC procedure of current pose. But we observed that an information gain is different even when the same number of inliers is given. Snavely *et al.*[21] propose skeletal graphs to select informative frames. They solved *offline* problem by utilizing trace of translational covariance as an uncertainty measure. In offline problem, maximum and minimum boundaries of uncertainty are bounded in the problem as all the images are provided prior to run algorithm. However, our problem is *online 3D reconstruction and camera tracking*, and such boundary cannot be determined prior to run the system. Therefore, we consider Fisher information matrix and its summarized statistics to measure informativeness of the current frame and initialize the new keyframe based on this metric. Recently, similar work is proposed by Kerl et al [8]. They have utilized relative entropy that is somewhat heuristic method based on their observation, while ours utilized summarized statistics of Fisher information which represents theoretically lower bound of the variance of estimator. Furthermore, we show that simple thresholding works well as opposed to [8].

2.3 Image-based Localization

Image-based localization in this paper is a problem of computing the 6-DoF location of given image with respect to existing set of images. In early period of research, only approximated location was obtained [20] by solving large-scale image retrieval approach [18]. Recently, a complete 6-DoF pose is obtained based on visual 3D maps built by structure from motion technique [7, 13, 26, 19]. In [7, 26], synthesized views are created to group the set of features as same document in vocabulary tree [18]. Most relevant top-k views are considered for expensive

SIFT [16] matching. In [13], prioritized 3D points are matched to input image using approximate nearest neighbor search. Unlike other approaches, it queries 3D point first rather than 2D features in an input image. In [19], direct 2D-3D matches are used to localize a frame. In [19] visual vocabulary-based prioritized search has been proposed to match the number of features required for geometric verification with less computation. We are particularly interested in direct 2D-3D match [19, 14] with vocabulary tree [18] which introduces predictable amount of computation.

3 Problem Formulation

3.1 Metric-Topological Map Representation

The internal representation of the map can be *metric* or *topological*:

- The *metric representation* is the most common for robot navigation and considers a three-dimensional space in which it places the objects. The objects are placed with precise coordinates. This representation is very useful, but is sensitive to noise and it is difficult to close a loop because the location of all vertexes should be adjusted at the moment of the loop closure detection.
- The *topological representation* only considers relative relations between them. No exact global coordinates defined to describe the position of vertexes (i.e., keyframes). The map is then a graph, in which the vertexes corresponds to keyframes (or landmarks) and edges correspond to the relations (e.g., relative pose, image measurements).

The topological map can be transformed to metric map by compositing relative poses assigned on edges between vertexes. To this end, we search a topological graph starting from reference keyframe by breadth-first search (BFS). In many cases, current keyframe is set to origin and remaining keyframes pose are described based on the origin.

3.2 The Graph Representation of Topological Map

Keyframes and 3D points are represented as graph $\mathcal{G}$ consists of a set of keyframe vertexes $\mathcal{K}$, a set of homogeneous 3D points $\mathcal{M}$, and a set of edges represent the relative pose $\mathcal{P}$ between two keyframes or image projection $\mathcal{Z}$. Each keyframe vertex saves unique identification number, list of 3D points observed and normalized image coordinates of associated 3D points. Note that we do not save absolute pose in the keyframe vertexes.

Let $\mathcal{P} = \{p_{jk} \mid j, k \in [0, \ldots, m-1]\}$ be a set of m 6-DoF camera poses (i.e., keyframe poses) where each $p_{jk} \in SE(3)$ is a 6×1 vector contains orientation (i.e., angle-axis representation) and position that defines a 4×4 homogeneous transformation matrix $P^{jk} \in \mathbb{R}^{4\times4}$, a pose j defined in frame k. The transformation matrix P^{jk} and its inverse is defined as

$$P^{jk} = \left(P^{kj}\right)^{-1} = \begin{bmatrix} R^{jk} & T^{jk} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times4}, \quad \text{with} \quad R \in SO(3), \quad T \in \mathbb{R}^3. \tag{1}$$

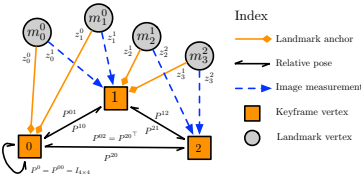


Fig. 3: A mini example of topological map. The i-th 3D point with anchor keyframe j is denoted by m_i^j. The anchor keyframe is a keyframe which observes a 3D point at first. The image measurement of m_i in j-th keyframe is defined z_i^j. The pose between keyframe i and j is denoted by P^{ij}.

Let $\mathcal{M} = \{m_i^k \mid i \in [1, \ldots, n], k \in [0, \ldots, m-1]\}$ be a set of n landmarks described relative to anchor keyframe k. Each $m_i^k = [\bar{m}_i^k, 1]^\top = [\bar{m}_{i,1}^k, \bar{m}_{i,2}^k, \bar{m}_{i,3}^k, 1]^\top$ is a 4×1 homogeneous point where the bar notation selects the 3D component (See Fig. 3). A landmark m_i^k defined in keyframe k can be transformed to keyframe j by transformation matrix P^{jk} as Equation (2).

$$m_i^j = P^{jk} m_i^k = [R^{jk}\bar{m}_i^k + T^{jk}, 1]^\top = \begin{bmatrix} R^{jk} & T^{jk} \\ 0 & 1 \end{bmatrix} [\bar{m}_{i,1}^k, \bar{m}_{i,2}^k, \bar{m}_{i,3}^k, 1]^\top \tag{2}$$

In this paper, we assume that the camera is calibrated (i.e., intrinsic parameters are known a priori). Image projection using homogeneous coordinates $\pi : \mathbb{R}^4 \rightarrow \mathbb{R}^3$ is modeled as a pinhole camera:

$$\pi\left(P^{jk} m_i^k\right) = \left(\frac{\bar{m}_{i,1}^j f}{\bar{m}_{i,3}^j} + o_x, \frac{\bar{m}_{i,2}^j f}{\bar{m}_{i,3}^j} + o_y, \ 1 \right)^\top, \tag{3}$$

where f is the focal length, $\mathbf{o} = [o_x, o_y]^\top$ is the principal point. The normalized image coordinates is defined by $\pi\left(P^{jk} m_i^k\right) / \|\pi\left(P^{jk} m_i^k\right)\|$. In a monocular camera, the observation of a 3D point m_i in keyframe j is denoted by (3) where $z_i^j \in \mathbb{R}^3$. In a binocular camera (e.g., stereo or RGBD camera), the observation is

$$z_i^j = \left[\pi(P^{jk} m_i^k), \quad f\frac{\bar{m}_{i,1}^j - b}{\bar{m}_{i,3}^j} + o_x \right]^\top \in \mathbb{R}^4, \tag{4}$$

where b is the baseline between two cameras. All images are assumed to be undistorted and rectified.

3.3 Optimization of Pose Graph

In order to achieve constant-time operation, we define a sub set of keyframes $\mathcal{A}$ which is called *active keyframes* from set $\mathcal{K}$. The set $\mathcal{A}$ usually constructed with most recent-w keyframes from the current keyframe. The w is called window-size, typically 5 to 10 in our implementation. Local optimization is performed within the set $\mathcal{A}$.

Algorithm 1: Local bundle adjustment: Bundle adjust keyframes and landmarks within the sliding window w. Reference frame a_0 usually set to most recent frame.

Input : Reference keyframe a_0, Window size w
Output: Bundle adjusted active keyframes $\mathcal{A} \in \mathcal{G}$ and periphery keyframes $\mathcal{A}' \in \mathcal{G}, \mathcal{A} \neq \mathcal{A}'$
Initialize active keyframe set $\mathcal{A}$, periphery keyframe set $\mathcal{A}'$ and active landmark set $\mathcal{M}_\mathcal{A}$ as $\emptyset$;
Perform metric embedding from a_0 only w number of frames. Insert embedded keyframes to $\mathcal{A}$;
// Construct active set $\mathcal{A}$ and periphery keyframe set $\mathcal{A}'$
forall the *keyframe* $k \in \mathcal{A}$ **do**
  $\mathcal{L} \Leftarrow$ All landmarks associated to a keyframe k ;
  forall the *landmark* $m \in \mathcal{L}$ **do**
    $\mathcal{M}_\mathcal{A} \Leftarrow \mathcal{M}_\mathcal{A} \cup \{m\}$;
    $\mathcal{N} \Leftarrow$ Set of keyframes that observed the landmark m;
    // Add periphery keyframes for pose optimization
    forall the *keyframe* $k' \in \mathcal{N}$ **do**
      if $k' \notin \mathcal{A}$ **then**
        $\mathcal{A}' \Leftarrow \mathcal{A}' \cup \{k'\}$
      end
    end
  end
end
Perform bundle adjustment using cost function (5) with $\mathcal{A}, \mathcal{A}', \mathcal{M}_\mathcal{A}$;
// De-embedding : Update topological map using optimized poses
forall the *Optimized keyframe* $i \in \{\mathcal{A} \cup \mathcal{A}'\}$ **do**
  $\tilde{P}^{ia_0} \leftarrow$ Pose of the keyframe i with respect to a_0 ; /* bundle adjusted pose */
  $\mathcal{N} \leftarrow$ Find neighbor keyframes of i;
  forall the *neighbor keyframe* $j \in \mathcal{N}$ **do**
    $\tilde{P}^{ja_0} \leftarrow$ Pose of the neighbor keyframe j with respect to a_0;
    $P^{ij} \leftarrow \tilde{P}^{ia_0} \tilde{P}^{ja_0\top}$; /* Update an edge $j \rightarrow i$ */
    $P^{ji} \leftarrow \tilde{P}^{ja_0} \tilde{P}^{ia_0\top}$; /* Update an edge $i \rightarrow j$ */
  end
end
// Update landmark position in topological map
forall the *Optimized landmark* $\tilde{m}_i^{a_0} \in \mathcal{M}_\mathcal{A}$ **do**
  $\tilde{P}^{ka_0} \leftarrow$ Pose of anchor keyframe k of a landmark $m_i^{a_0}$ with respect to a_0;
  $m_i^k = P^{ka_0} \tilde{m}_i^{a_0}$; /* $\tilde{m}_i$ Bundle adjusted 3D point */
end

As we preserve topological relations between map and poses, we further seek keyframes that are worthy of inclusion in the optimization. These keyframes are denoted $\mathcal{A}'$ which are selected based on the existence of covisible features between the set $\mathcal{A}$ and a neighbor keyframe set $\mathcal{N}$ of $\mathcal{A}$, but $\mathcal{N} \notin \mathcal{A}$. The original active keyframes include all measurement and poses to be optimized (i.e., poses and landmarks are optimized). However, only relative pose information of keyframes in $\mathcal{A}'$ are optimized (i.e., pose-to-pose only optimization). This approach is motivated by [22]. In contrast to [22] which used single global coordinates, our approach is purely based on relative formulation, so it shows better metric consistency in local window in large-scale loop closures. Therefore, our algorithm can handle complex multi level loop closures which will be described in detail in Section 4.3. Following cost function J is considered to optimize both 3D point and pose-to-pose constraint of local metric embeddings constructed by Algorithm 1:

$$J^2(\mathcal{G}, a_0) = \sum_{i \in \mathcal{M}_\mathcal{A},\, j \in \mathcal{A}} v_{ij} \left(z_i^j - \pi \left(P^{ja_0} m_i^{a_0} \right) \right)^2 + \sum_{i,j \in \mathcal{A}'} v_{ij} \left(P^{ij\top} \Lambda_{ij} \left(P^{ia_0} P^{ja_0\top} \right) \right) \tag{5}$$

where $\mathcal{G}$ is a pose graph, a_0 is an reference frame of local window, $v_{ij\ (i \neq j)}$ is a function where 1 if a direct path available in topological map, otherwise 0 and Λ_{ij} is a covariance matrix.

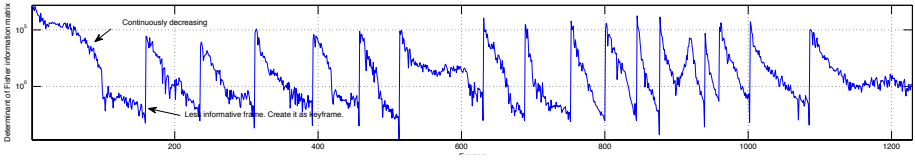


Fig. 4: Summarized statistics of Fisher information matrix. The time history of determinant of Fisher information matrix in FR2/SITTING_XYZ sequence. The sequence information is shown in Table. 1.

4 Proposed Method

4.1 Fisher Information for Uncertainty Measure

Current scheme to determine keyframe usually relies on a norm of translation or rotation part of the pose with respect to previous keyframe. This results uniform sample of poses and landmarks in the Euclidean space as proposed by existing approaches [10, 11]. However, determination of this threshold is somewhat vague, also it is prone to pose estimation error or spiky motion. The distance or rotation threshold also depends on scene characteristic. For example, if a camera travels in a large outdoor environment, the same distance threshold as indoor environment will create keyframe too frequently. Therefore, we need a more intelligent metric to determine whether current frame should be registered as keyframe or not. In the following subsection, we describe detailed metric used for keyframe generation.

In [21], the uncertainty is modeled by a trace of covariance matrix. It is well suited for offline problem such as structure from motion (SfM) by calculation of uncertainty amount along given keyframe path. However, it is unclear to determine the threshold of trace of the covariance to represent uncertainty to be kept for the keyframe creation that is online process. To address this problem, we propose a novel keyframe selection scheme based on Fisher information matrix.

The Fisher information matrix can be approximated by the inverse of the covariance matrix of maximum-likelihood estimators [1]. As bundle adjustment is a maximum likelihood problem, estimators asymptotically have zero bias and the lowest-variance that any unbiased estimator can have [25]. Obtaining covariance is not straightforward, since the bundle adjustment problem is large. Filters like Kalman filter always keep covariance of the system, but they are not practical for such large problems.

Let pose $p^\star$ is a solution of maximum likelihood estimate of p obtained by bundle adjustment :

$$p^\star = \arg\min_p \|z - f(p)\|^2 \quad \text{where} \quad z = f(p), \tag{6}$$

where $f(\cdot)$ is a nonlinear measurement process (3) and z is a measurement. If measurement process $f(\cdot)$ is differentiable, the covariance matrix of pose p can be estimated by following equation:

$$\Sigma_{\mathbf{p}^\star} = (\mathrm{J}_{p^\star}^\top \Sigma_{\mathbf{z}} \mathrm{J}_{p^\star})^{-1} \tag{7}$$

where $\mathtt{J}_{p^\star}$ is Jacobian matrix evaluated at estimated pose $p^\star$, and $\Sigma_{\mathbf{z}}$ is covariance matrix of observations. We assume that the measurement process is independent and equal covariance which means that $\Sigma_{\mathbf{z}} = I$. The Fisher information matrix F is defined as

$$F = \Sigma_{\mathbf{p}^\star}^{-1}. \tag{8}$$

See Fig. 4 for F of FR2/SITTING_XYZ sequence. For online algorithm, computing the covariance of current frame with respect to all other existing keyframe is not feasible. As our representation separates global and local map, we only compute current pose uncertainty within active keyframes. This naturally models current frame uncertainty within a meaningful physical region. For example, if we consider whole pose graph to compute covariance, the distant keyframe might increase actual uncertainty we are interested in. In [21], only translational uncertainty was considered for offline structure from motion. However, the rotational uncertainty should be considered in online video-based reconstruction. We estimate full 6×6 covariances of 6-DoF pose with considering all landmarks currently observing and relative keyframes in pose graph.

4.2 Keyframe Selection Scheme

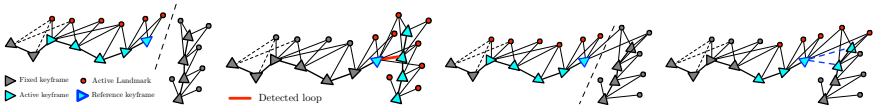
The Fisher information is inverse of covariance in our problem. We consider the determinant of Fisher information matrix τ_F:

$$\tau_F = \det(F) = \det\left(\Sigma_{\mathbf{p}^\star}^{-1}\right). \tag{9}$$

As the amount of variance can be approximated by determinant of covariance matrix, if estimator variance becomes large, proposed metric τ_F naturally converges to zero, but never reaches zero in theory. Fig. 4 shows actual value of proposed metric for RGB-D sequence FR2/XYZ along with time (See Table. 1 for details). Setting threshold of τ_F as moderately small number (e.g., 0.001 in our case) will adaptively selects keyframes based on surrounding environment and feature qualities. The proposed method creates keyframe based on uncertainty bound of pose. Naturally it considers both position and rotational uncertainty in a single framework. In the previous approach, distance or rotation threshold was used to keep feasible problem size for optimization. In other words, the propose approach is naturally considers the quality of measurements which depends on scene size and the speed of moving camera. In summary, it creates even sample of keyframe in information wise, not heuristically determined distance threshold. Although a user still have to determine uncertainty boundary, however, the tuning parameter itself more informative. Tuning the threshold is easier than determination of 0.2 m or 20 degrees which depends on scene characteristics for example.

4.3 Multi Level Loop Closing

Structure-from-motion or keyframe-based visual SLAM problem largely rely on the quality of feature tracking [27]. If feature tracks are disjointed caused by



(a) Right before the loop (keyframe-to-keyframe) (b) Right after (keyframe-to-keyframe) (c) Right before the loop (feature-level) (d) Right after (feature-level)

Fig. 5: An illustrative example of two loop detection cases. (a) local bundle adjustment is performed past five frames. (b) If loop is found, corresponding relative pose and landmark observations are added to the bundle adjustment framework, then optimized. (c) Right before the loop, features in current frame and 3D landmarks are linked. (d) After the detection, pose links are registered.

objects moving in an out of the view, or pure rotation motion, are not included as constraint in the bundle adjustment. Observing a 3D point in many views definitely helps to obtain accurate map and pose within bundle adjustment framework. We will call this constraint namely *feature-level loop* throughout this paper.

However, obtaining a feature-level loop is not a straightforward problem. In PTAM [10], they projected all 3D features in the map which are visible by the previous motion, to obtain non-consecutive feature-level loop. But, the PTAM requires strong motion prior to this end. Once a camera motion is lost, they compare small image patches of all keyframes to measure similarity then perform exhaustive matching that takes time. It is also prone to errors and not scalable. In [27], they proposed two-pass algorithm to find non-consecutive feature tracks which is the same as the feature-level loop. However, it is very slow and not an online algorithm.

In this paper, we propose feature-level loop closing motivated by recent development of image-based localization [19, 7]. The main difference compared to image-based localization, we update database in real-time and utilize linked motion prior of the matched 3D landmark, to reject false positives. Also we use videos, not a discrete images as in [19, 7], we utilize both temporal relationship among frames and feature-level loops. The two types of loop are shown in Fig. 5.

Keyframe-to-keyframe Loop Closing We first utilize existing keyframe-to-keyframe style loop closing. Our feature tracker will provide the most recent SURF descriptor tracked until the current frame. These descriptors are provided to vocabulary tree when keyframe is created. The SURF descriptors of a new keyframe are queried to the existing keyframes, and 2D-3D correspondences are obtained from query result. Further geometric verification is performed with RANSAC. Outliers will be removed during the RANSAC process. If enough inliers are found, nonlinear optimization minimizing reprojection error is performed to obtain accurate relative pose with respect to matched frame. We add this relative pose and 2D-3D observations to the graph, further local bundle adjustment will be performed.

Algorithm 2: Feature-level loop detection. $\tau = 0.7, \epsilon = 0.001$ in this paper.

Data: Vocabulary tree V, trained by using descriptors associated to 3D points.
Input : SURF descriptors Q, of current input frame.
Output: Loop constraint P^{ka_0} between current frame a_0 and matched keyframes k.

$M, K \leftarrow \emptyset$; /* Initialize M:2D-to-3D matches, K:votes */
$W \leftarrow$Find visual words of query descriptors Q using vocabulary tree V;
forall the *visual word* $w \in W$ **do**
  $L \leftarrow$ Get landmark ids of w;
  $q \leftarrow$ Get SURF descriptor of w;
  $d_{first}, d_{second} = \infty$;
  forall the *landmark* $l \in L$ **do**
    $q' \leftarrow$ Get SURF descriptor of a landmark l;
    $d \leftarrow$ Compute distance between descriptors q and q';
    if $d < d_{first}$ **then**
      $d_{second} = d_{first}$;
      $d_{first} = d$;
      $m \leftarrow$ A 2D feature related to the visual word w and a 3D landmark l are set as best match;
      $k \leftarrow$ Find anchor keyframe number of a landmark l_j;
    end
  end
  if $\frac{d_{first}}{d_{second}} > \tau$ **then**
    $M = M \cup \{m\}$;
    $K = K \cup \{k\}$;
  end
end
Perform RANSAC with P3P using 2D-3D matches M;
/* Compute Fisher information and optimize pose. λ (= 15) */
if *the number of inliers* $> \lambda$ **then**
  Optimize pose P using inliers. The determinant of Fisher information, J is returned.;
  if $J > \epsilon$ **then**
    forall the *voted keyframe* $k \in K$ **do**
      Compute relative pose P^{ka_0} between current a_0 and target keyframe k;
      Add relative pose constraint to the pose graph.
    end
  else
    Create a new keyframe using current frame.
  end
end

Feature-Level Loop Closing The majority of existing online 3D modeling or visual SLAM systems have been used keyframe-to-keyframe loop closure scheme [15, 22, 11] which seeks corresponding keyframe among existing keyframes. Loop closure operation (*i.e.,* querying current keyframe to the database) only happens when the keyframe insertion is performed. The existing approaches overlooked the fact that it is possible to have better loop closure from input frames between keyframes.

In this paper, we directly search 2D-3D correspondences efficiently to perform every-frame loop seeking by utilizing recent results of image-based localization [19, 7]. In [4], similar view recognition is performed by using kd-trees. The system builds kd-tree incrementally, and all descriptors in current frame are queried to the database. They kept top-k views among registered views for geometric verification. However, this approach does not scale well due to the fact that the metric tree is required to store all descriptors on memory, also search time grows significantly as the number of features in the tree increases.

Instead computing pose from these 2D-3D matches directly, we create keyframe based on a pose between current and last keyframes that is most accurate pose in the graph. Then we add these 2D-3D matches to the keyframe. See Algorithm 2 in detail.

Dataset	Travel Length [m]	Duration [s]	# images	# of keyframes	% of frames used
fr1/xyz	7.112	30.09	798	53	6.6
fr1/room	15.989	48.90	1362	138	10.1
fr1/desk	9.263	23.40	613	74	12.1
fr1/desk2	10.161	24.86	640	121	18.9
fr1/rpy	1.664	27.67	723	83	11.4
fr2/desk	18.880	99.36	2965	99	3.4
fr2/xyz	7.029	122.74	3669	53	1.5
fr3/office	21.455	87.09	2585	118	4.6
fr3/sitting_xyz	5.496	42.50	1261	17	1.4

Table 1: Evaluation datasets. Evaluation sequences and its scene characteristics are shown. The length of camera traveled, duration, the number of images and keyframes reconstructed are listed. As you can see, relatively stationary motion shows low percentage of frames selected as keyframes.

	RMSE ATE in $[m]$					RMSE RPE in $[m]$	
Dataset	Ours	Stuckler *et al.*[23]	Bylow *et al.*[2]	Kinfu	RGB-D SLAM [5]	Ours	Stuckler *et al.*[23]
fr1/xyz	0.015	**0.013**	0.021	0.026	0.014	0.021	**0.02**
fr1/room	0.101	**0.069**	0.078	0.313	0.101	**0.056**	0.139
fr1/desk	0.059	0.043	0.046	0.057	**0.026**	**0.014**	0.075
fr1/desk2	0.108	**0.049**	0.069	0.420	0.059	**0.067**	0.09
fr1/rpy	0.031	0.027	0.042	0.133	**0.026**	**0.037**	0.04
Average	0.062	**0.040**	0.051	0.189	0.045	**0.032**	0.077
fr2/xyz	**0.013**	0.020	-	-	-	**0.002**	0.030
fr2/desk	0.072	0.052	-	-	-	**0.047**	0.099
fr3/office	**0.025**	-	0.039	0.064	-	0.011	-
fr3/sitting_xyz	0.017	-	-	-	-	0.005	-

Table 2: Comparisons to other approaches. We have compared our method to four different approaches. The nine sequences were evaluated. The reader should note that all the methods compared in this table are using pose estimation based on dense iterative closed point (ICP) using depth image. In contrast, our algorithm considers only a depth of each keypoint which is essentially same as a stereo camera. Without use of ICP algorithm as main pose estimator, we achieved few ($2.2cm$) difference compared to the state-of-the-art approaches in ATE. Furthermore, our approach has better performance in terms of relative pose error (RPE). This is due to the fact that our algorithm successfully close the loop which is a key of minimizing drift. We compared RPE with [23] because the RPE is only reported in [23].

5 Experimental Result

The evaluation datasets are summarized in Table 1. For evaluation we chose 9 sequences across Freiburg1 to 3 (FR1–3) datasets that are especially recorded by handheld camera. Those datasets also have been used by several other authors for evaluation [5, 2, 9, 23]. We have compared our approach with those authors (See Fig. 2). The experiments were performed on a laptop with an Intel Core i7 with 2.3 Ghz processor. But only single core is utilized for the proposed method to make the proposed method work on embedded system in the future.

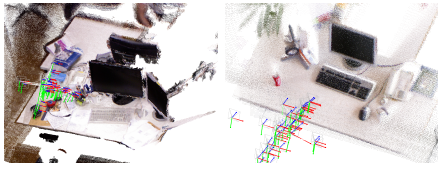
We utilize the evaluation software that measures root mean square error (RMSE) of the translation drift (RPE) in m/s and absolute trajectory error (ATE) in meters. Detailed computation of those metrics is shown in [24].

Unlike existing RGB-D online mapping approaches which use all dense 3D point cloud to obtain relative pose between frames [17, 5], we only utilize depths of tracked 2D features (usually 200 in a frame) and the relative pose is computed by P3P algorithm, essentially the same as stereo camera. Thus the RGB-D camera can be replaced with a stereo camera without changing the system.

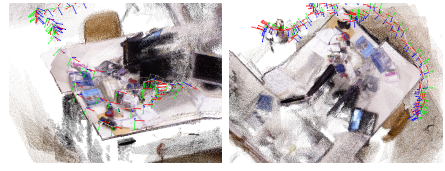
We have compared our method with four other approaches: multi-resolution surfel maps (MRSMap) [23], KinectFusion [17], RGBD-SLAM [5] and Bylow *et al.*[2]. As shown in Table 2, our approach is superior in terms of relative pose error (RPE) as two types of loops are thoroughly considered. However, absolute trajectory error (ATE) is slightly inferior to existing approach (by less than a few centimeters). One of the main reasons is that we did not use dense depth map for motion estimation that is likely to be more accurate than image only pose estimation when the motion is slow. Another reason is that the ATE metric divides number of keyframes used. In most cases, our algorithm produces fewer numbers of keyframes than other approaches which used all frames as keyframe. The accuracy of our method can be shown by RPE as other approaches are less accurate despite they have used ICP and all frames. The reconstructed 3D environment map of evaluation sequences is shown in Fig. 6.

Fig. 7 shows the difference between the proposed and existing keyframe selection method. Our method creates keyframes adaptively as shown in the screenshot above according to the uncertainty amount of current pose computed with landmarks.

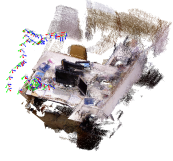
A video of our method is available on http://youtu.be/gnbnFEjy8wU.



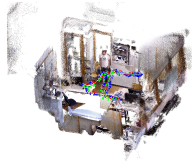
(a) **fr1/xyz (left), fr2/xyz (right) sequences reconstructed.** Both sequeces have a specific camera motion as shown in the figures (i.e., the camera was moved only in axial direction). The axial trajectory is clearly seen in the figures.



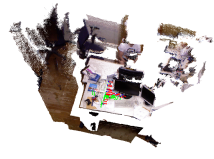
(b) **fr1/desk (left), fr1/desk2 (right) sequence reconstruction.** Two sequences were captured in the same physical place, but captured twice. Dense 3D point cloud model obtained by proposed method. Red, gree, red color bars mean X,Y,Z axes of the keyframes respectively.



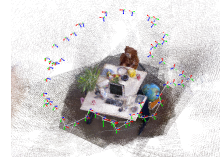
(c) Sequence FR1/DESK. 74 keyframes.



(d) Sequence FR1/ROOM. 138 keyframes.

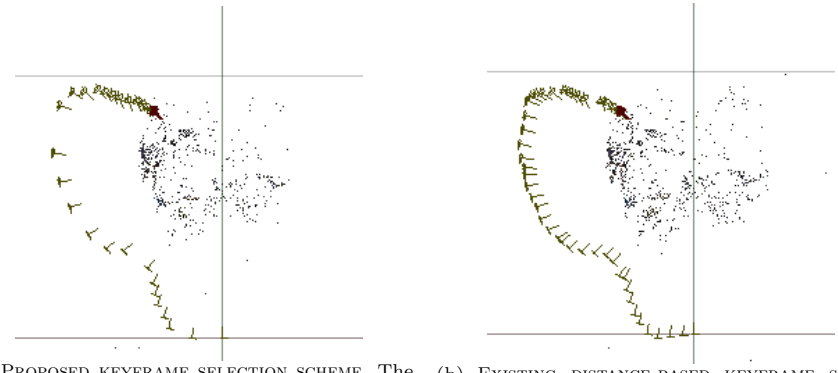


(e) Sequence FR1/XYZ. 53 keyframes.



(f) Sequence FR2/DESK. 99 keyframes.

Fig. 6: Reconstruction results. Once all frames are processed, bundle adjustment with all keyframes is performed. Then the full-resolution depth image of each keyframes are projected by the assigned keyframe pose. Currently, colored points are rendered.



(a) Proposed keyframe selection scheme. The proposed method utilizes Fisher information matrix to select keyframe from 30fps video sequence. In fast or high angular velocity motion, our method automatically creates more dense keyframes compared to normal camera movement.

(b) Existing distance-based keyframe selection scheme. The keyframes are generated with 0.2m constant distance threshold with orientation norm 0.1 radian. This method creates regular sample of Euclidean 3D environment, but these keyframes are very redundant.

Fig. 7: Comparison between keyframe selection methods. The propose method and existing method are compared with same video sequence. Our method (left) adaptively creates sparse or dense keyframe compared to the right (distance-based approach).

6 Conclusion and Future Work

We have proposed a novel and efficient framework for online 3D reconstruction and camera tracking. The information theoretic keyframe selection scheme can adaptively selects the keyframe, and the feature level loop detection successfully closes the loops with the same uncertainty metric. We demonstrated the proposed approach with ground truth, where it performs better than or at least as good as the state of the art methods. Considering that only sparse keyframes are used for reconstruction, the proposed algorithm has many benefits over the dense RGB-D reconstruction algorithms.

One interesting problem for the future work is to keep the number of keyframes and landmarks as sparse as possible while it preserves essential information of the environment. This is also connected to life-long visual mapping or skeletal graph construction. This would allow the map size to be proportional to the characteristics (e.g., size, a degree of clutter) of the environment not to the number of images taken in the environment.

Acknowledgment

This work was supported partly by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST), ICT R&D programs of MSIP/IITP (No. 14-824-09-006 and No. 2014-10047078) and Defense Research Grant, funded by the Agency for Defense Development, under the contract UD120013JD.

References

1. Abt, M., Welch, W.J.: Fisher information and maximum-likelihood estimation of covariance parameters in gaussian stochastic processes. Canadian Journal of Statistics 26(1), 127–137 (1998)
2. Bylow, E., Sturm, J., Kerl, C., Kahl, F., Cremers, D.: Real-time camera tracking and 3d reconstruction using signed distance functions. In: RSS. RSS (2013)
3. Cummins, M., Newman, P.: Fab-map: Probabilistic localization and mapping in the space of appearance. The International Journal of Robotics Research 27(6), 647–665 (2008)
4. Eade, E., Fong, P., Munich, M.E.: Monocular graph slam with complexity reduction. In: IROS. pp. 3017–3024. IEEE (2010)
5. Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., Burgard, W.: An evaluation of the rgb-d slam system. In: ICRA. pp. 1691–1696. IEEE (2012)
6. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. The International Journal of Robotics Research 31(5), 647–663 (2012)
7. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From structure-from-motion point clouds to fast location recognition. In: CVPR. pp. 2599–2606. IEEE (2009)
8. Kerl, C., Sturm, J., Cremers, D.: Dense visual slam for rgb-d cameras. In: Proc. of the Int. Conf. on Intelligent Robot Systems (IROS) (2013)
9. Kerl, C., Sturm, J., Cremers, D.: Robust odometry estimation for rgb-d cameras. In: ICRA. ICRA (2013)
10. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: ISMAR. pp. 225–234. IEEE (2007)
11. Konolige, K., Agrawal, M.: Frameslam: From bundle adjustment to real-time visual mapping. Robotics, IEEE Transactions on 24(5), 1066–1077 (2008)
12. Konolige, K., Bowman, J.: Towards lifelong visual maps. In: IROS. pp. 1156–1163. IEEE (2009)
13. Li, Y., Snavely, N., Huttenlocher, D.P.: Location recognition using prioritized feature matching. In: ECCV, pp. 791–804. Springer (2010)
14. Lim, H., Sinha, S.N., Cohen, M.F., Uyttendaele, M.: Real-time image-based 6-dof localization in large-scale environments. In: CVPR. pp. 1043–1050. IEEE (2012)
15. Lim, J., Frahm, J.M., Pollefeys, M.: Online environment mapping. In: CVPR. pp. 3489–3496. IEEE (2011)
16. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV. vol. 2, pp. 1150–1157. Ieee (1999)
17. Newcombe, R.A., Davison, A.J., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: ISMAR. pp. 127–136. IEEE (2011)
18. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR. vol. 2, pp. 2161–2168. IEEE (2006)
19. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2d-to-3d matching. In: ICCV. pp. 667–674. IEEE (2011)
20. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: CVPR. pp. 1–7. IEEE (2007)
21. Snavely, N., Seitz, S.M., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: CVPR. CVPR (2008)
22. Strasdat, H., Davison, A.J., Montiel, J., Konolige, K.: Double window optimisation for constant time visual slam. In: ICCV. pp. 2352–2359. IEEE (2011)

23. Stuckler, J., Behnke, S.: Integrating depth and color cues for dense multi-resolution scene mapping using rgb-d cameras. In: IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems. pp. 162–167. IEEE (2012)
24. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: IROS (Oct 2012)
25. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustmenta modern synthesis. In: Vision algorithms: theory and practice, pp. 298–372. Springer (2000)
26. Wendel, A., Irschara, A., Bischof, H.: Natural landmark-based monocular localization for mavs. In: ICRA. pp. 5792–5799. IEEE (2011)
27. Zhang, G., Dong, Z., Jia, J., Wong, T.T., Bao, H.: Efficient non-consecutive feature tracking for structure-from-motion. In: ECCV, pp. 422–435. Springer (2010)