Virtual Insertion: Robust Bundle Adjustment over Long Video Sequences

Ziyan Wu*[1]
ziyan.wu@siemens.com
Zhiwei Zhu[2]
zhiwei.zhu@sri.com
Han-Pang Chiu[2]
han-pang.chiu@sri.com

[1] Siemens Corporate Technology
Princeton, NJ, USA
[2] SRI International
Princeton, NJ, USA

Abstract

Our goal is to circumvent one of the roadblocks of using existing bundle adjustment algorithms for achieving satisfactory large-area structure from motion over long video sequences, namely, the need for sufficient visual features tracked across consecutive frames. We accomplish it by using a novel "virtual insertion" scheme, which constructs virtual points and virtual frames to adapt the existence of visual landmark link outage, namely "visual breaks" due to no common features observed from neighboring camera views in challenging environments. We show how to insert virtual point correspondences at each break position and its neighboring frames, by transforming initial motion estimations from non-vision sensors into 3D to 2D projection constraints of virtual scene landmarks. We also show how to add virtual frames to bridge the gap of non-overlapping field of view (FOV) across sequential frames. Experiments are conducted on several real-world challenging video sequences, collected by multi-sensor based visual odometry systems. We demonstrate our proposed scheme significantly improves bundle adjustment performance in both drift correction and reconstruction accuracy.

1 Introduction

Bundle Adjustment is a key process to enhance the global accuracy of the 3D camera pose and structure estimation in the framework of structure from motion over long video sequences [1, 5, 6, 7, 8]. It is formulated as a nonlinear optimization that minimizes the 2D pixel projection error of each 3D scene feature extracted from the video, resulting optimal camera parameters and landmark positions. In recent years, many bundle adjustment algorithms have been proposed, such as Sparse Bundle Adjustment[10], Parallel Bundle Adjustment[15] and Incremental Bundle Adjustment[12] etc., and a number of reliable bundle adjustment open softwares have become available[2, 10, 15], which has made bundle adjustment a standard and convenient tool for Computer Vision community.

However, most bundle adjustment algorithms require sufficient visual feature correspondences from each camera frame to its neighboring frames in video sequences, which are hard to collect in real environments, especially for indoor real-time navigation applications.

*This work was done while the author was a student associate at SRI International, Princeton, NJ, USA.

A camera may not observe enough common scene points over a long period of time due to occlusions or non-texture background such as the white walls etc.. With the use of video images as the only input, bundle adjustment will easily fail due to the constant link outage of visual landmarks in the scene. We call it the effect of "visual breaks", and the issue of "visual breaks" has hindered the usage of bundle adjustment. It is particularly critical for sequential Structure from Motion (sSfM) applications where motion estimation is from "chaining" neighboring key frames.

On the other hand, to deal with this issue of "visual breaks", vision-based navigation systems, such as Simultaneous Localization and Mapping (SLAM)[13], typically do not rely on the video cameras only for robustness. Different techniques have been proposed to reduce the drift caused by"visual breaks" and other sources (e.g. inaccurate calibration) by fusing non-vision sensors, such as Inertial Measurement Unit (IMU)[3, 4], LiDAR[16] or GPS[10]. There are also systems which make motion assumption, such as constant motion model, to propagate estimation when there is no visual information available. As a result, good motion measurements from non-vision sensors or motion assumptions can be obtained easily at these "visual breaks" locations. However the bundle adjustment is still not able to use the motion estimates from these techniques directly due to a missing approach to incorporate them inside the cost function during optimization.

In this paper, in order to overcome the above issue, we propose a "Virtual Insertion" scheme to construct elastic virtual links on these "visual breaks" positions to fill visual landmark link outage with the measurements provided by other sensors or motion assumptions, so that all camera positions can be linked in the long video by the real or virtual scene landmarks before bundle adjustment. This way enables the traditional bundle adjustment algorithms to achieve robust large-area structure from motion over long video sequences. Specifically, with the measurements from non-vision sensors at the "visual break" positions, we actually convert them into a set of virtual landmark links that will serve as 3D-2D projection constraints in the cost function of bundle adjustment optimization. As a result, measurements from other sensors can be integrated into existing bundle adjustment framework. Experiments on real-world long video sequences show that the virtual insertion scheme can significantly enhance both robustness and global accuracy of bundle adjustment over long video sequences in challenging real-world environments.

1.1 Related Work

Many proposed SLAM algorithms can handle the "visual breaks" reasonably well by incorporating the motion estimations from non-vision sensors and motion assumptions[2, 9, 11, 13, 14]. Oskiper et al. [14] integrated visual odometry system with a IMU using the extended Kalman filter framework. However camera poses are locally optimized for real-time performance. Chiu et al.[2] proposed using Sliding-Window Factor Graphs [9] as short-term smoother fusing estimations from multiple sensors, together with a long-term smoother incorporating loop closure constraints to achieve improvements locally and globally. [13] introduced an approximate Maximum A Posteriori estimator-based keyframe approach incorporating constraints generated from visual landmarks and IMU from marginalized frames, achieving computation efficiency and elevated accuracy globally. Instead of minimizing a weighted sum of image and GPS errors, which is much harder to do, Lhuillier [11] proposed an incremental Bundle Adjustment framework by enforcing an upper bound for the image reprojection error with the GPS data.

The proposed virtual insertions scheme is handling "visual breaks" while maintaining

globally optimized accuracy in a different but innovative fashion. When visual correspondences are non available, motion estimates from non-vision sensors are used to generate 3D-2D constraints that can be incorporated by cost function of traditional bundle adjustment. However, while many of these SLAM algorithms directly incorporate error models specifically built for different types of sensors in the cost function of bundle adjustment, our proposed algorithm can adapt to existing bundle adjustment frameworks seamlessly, which is one of our primary motivation. Moreover, in these SLAM algorithms, error models for sensors usually require elaborate calibrations, and serious error can be induced throughout the whole sequence if they are poorly calibrated. With the propose approach, the non-vision sensors are free from calibration since motion estimates from these sensors are used only at the locations with "visual breaks". That way, the motions estimates from non-vision sensor can be considered reliable within short intervals even without calibration, and the results for other locations will not be degraded after bundle adjustment.

Hence, compared to the existing SLAM algorithms, our goal is to propose a simple alternative way handling "visual breaks" problem for global accuracy, and make bundle adjustment more widely applicable, convenient and robust over long video sequences.

2 Motivation and Proposed Solution

2.1 "Visual Breaks" and Bundle Adjustment

A "visual break" is critical especially for sequential structure from motion, where usually a camera position has feature correspondences only with neighboring positions. With the help of IMU and Kalman filter[16], a visual odometry system is able to output reasonable and continuous poses using measurements from IMU especially at the "visual breaks". However the measurements from IMU cannot be integrated into the framework of bundle adjustment directly, resulting large jumps and drifts, which is demonstrated below.

The goal of bundle adjustment is to find 3D landmark positions and camera parameters that minimize the 2D re-projection error of 3D landmarks on the images, that is, let $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \cdots, f_N(\mathbf{x})]$ be the vector of re-projection errors over N camera positions, we wish to solve

$$\mathbf{x}' = \arg\min_{\mathbf{x}} \sum_{i=1}^{N} \|f_i(\mathbf{x})\|^2.$$

Here we take Levenberg-Marquart[13] (LM) algorithm as an example, in each iteration, LM solves the non-linear least squares problem in the form of

$$\Delta\mathbf{x}' = \arg\min_{\Delta x} \|\mathbf{J}\Delta\mathbf{x} + \mathbf{f}(\mathbf{x})\|^2 + \lambda\|\mathbf{D}\Delta\mathbf{x}\|$$

and updates $\mathbf{x}' = \mathbf{x} + \Delta\mathbf{x}'$. λ is a regularization scalar and $\mathbf{D}$ is the square root of the diagonal of $\mathbf{J}^\top\mathbf{J}$. For example, Figure 1(a) illustrates a typical indoor navigation sequence of a user traveling from $\mathbf{P}_1$ to $\mathbf{P}_8$, where $\mathbf{P}_i$ are camera positions and $\mathbf{q}_i$ are visual scene landmarks. Two "visual breaks" are caused by door closing and opening. Since $\{\mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5\}$ are only within the FOV of $\{\mathbf{P}_3, \mathbf{P}_4, \mathbf{P}_5\}$ which isolates themselves from other camera locations, the Jacobian $\mathbf{J}$ can be re-arranged as

$$\mathbf{J}' = \begin{bmatrix} \mathbf{J}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_B \end{bmatrix}$$

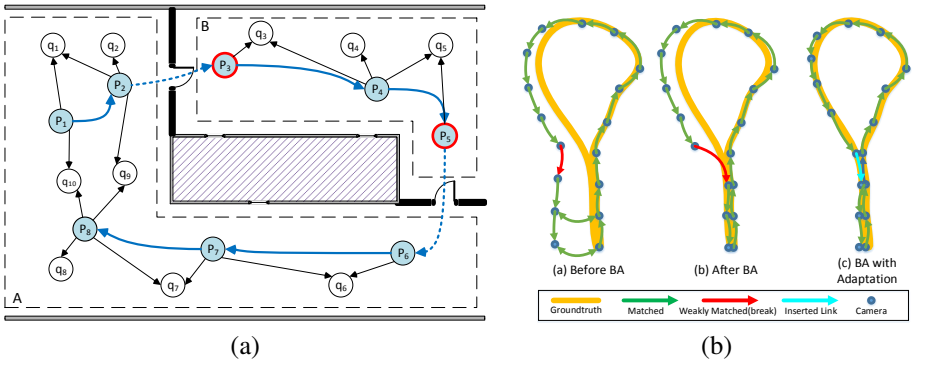


Figure 1: (a) A typical indoor navigation sequence. Black arrows indicate matched correspondences and blue ones denote motions estimated by visual odometry system. (b) Another example of sequence with "visual break" and the idea of linking the "visual break".

where $\mathbf{J}_B$ and $\mathbf{J}_B$ correspond to two nonoverlapping sets of locations. Similarly decomposition can be done to $\mathbf{D}$. This is equivalent to two independent bundle adjustment problems. In other words, the global-minima of the whole sequence becomes the combination of local-minimas in each of the two segments of the sequence because the transition between locations set A and B is unconstrained. This is the reason why large jumps can be found in the output trajectory from bundle adjustment when "visual breaks" exist in the sequence.

2.2 Linking the "Visual Breaks"

Figure 1(b) shows the illustration of a typical motion estimation over a video sequence with a "visual break" annotated with a red link arrow. Due to drift in the initial poses estimation, the loop does not close although the person travels back to the origin. It can be seen from Figure 1 (b) that the initial estimated trajectory from is continuous and smooth. After feature matching, frames at the end are matched with frames at the beginning, and for the other locations, frames are only matched with their neighboring frames. During bundle adjustment, with the constraints provided the loop closure, the drifts at the end can be reduced. However a large jump can be observed at the "visual break" location, as showed in Figure 1(b), since the constraints cannot be propagated to the other locations because of the "visual break". It is straightforward to consider this "visual break" as a "broken joint".

It is natural for us to think about adapting the "broken joints" with artificial links. From initial estimation of the camera poses fused with IMU, we can set up artificial links on the "visual breaks". Although drift will accumulate over long period in general, within a small period of time, the estimation fused with IMU can be considered as reliable and trustworthy. As shown in Figure 1(b), a virtual link estimated by IMU motion estimation can be inserted to the break location so that the constraints from loop closure can be propagated to the whole sequence. Hence, as it can be seen that the whole trajectory can reach global optima with drifts reduced on every location. In other words, this method is transferring the motion measurements from non-vision sensors into 3D-2D visual projection constraints, which are integrated into the cost function of bundle adjustment for a joint global optimization. This forms the base of proposed virtual insertion techniques.

3 Virtual Points Insertion

In this section, the detection of "visual breaks" and the method of virtual points insertion are described. Figure 2 shows an illustration of virtual points insertion.

3.1 "Visual Break" Detection

Let $\mathbf{P}_i$ be the 3D pose of camera at location i output by the visual odometry system, and $c_{i,j}$ be the number of matched feature correspondences between i and j. Assuming we have in total N key frames in the sequence, we consider location i as a "visual break" of the trajectory if it satisfies $\sum_{j=i+1}^{N} c_{i,j} = 0$. However in practice, more than one feature correspondence is needed to ensure the accuracy and strength of constraint for each position (e.g. at least 3 correspondences are needed in order to effectively remove outliers). So a "visual break" i should be defined as

$$\sum_{j=i+1}^{N} \delta\left(c_{i,j} > t_n\right) = 0 \tag{1}$$

in which $\delta(\cdot)$ is delta function which equals to 1 when the statement inside is true and 0 otherwise, t_n is a threshold as the number of feature correspondences needed to set up a link between two frames. $t_n = 8$ is used in our experiments.

3.2 Virtual Points Insertion

After a "visual break" location i is detected by the algorithm, n virtual points $\mathbf{X}^l = \{\mathbf{x}_1^l \dots \mathbf{x}_n^l\}$ should be generated in the local 3D coordinates system of frame i. These virtual points can be any structured (e.g. cubes) or unstructured points (i.e. random points) within the FOV of camera. A set of cube-structured virtual points is used in our experiments, as shown in Figure 2(a).

We want to insert virtual points to the location i and its neighboring frames so as to build a virtual link to adapt the "visual break". Here a set of neighboring frames ϕ_i are defined as

$$\phi_i = \{\mathbf{I}_j | \quad |j-i| \leq r_d\} \tag{2}$$

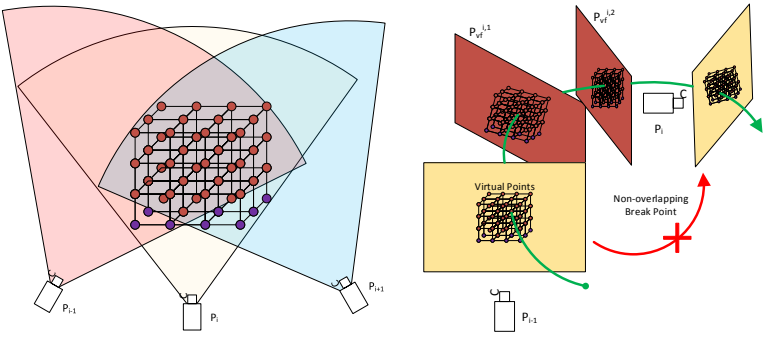
in which $\mathbf{I}_j$ is the frame index and r_d is the neighboring frame radius usually set to 1 or 2. These virtual points are all within the FOV of camera at position i but not necessarily within the neighboring frames of position i. Hence we need to select a subset of virtual points $\mathbf{X}^s \subseteq \mathbf{X}^l$ by selecting virtual points within the FOV of all neighboring frames. First, we calculate the 3D coordinates $\mathbf{x}_k^g$ of the virtual point k in global coordinates system

$$\mathbf{x}_k^g = \mathbf{P}_i^{-1} \mathbf{x}_k^l \tag{3}$$

Then we can calculate the image location $\mathbf{x}_k^{2D}$ of the virtual point k on neighbor frame j with pose $\mathbf{P}_j$. A virtual point $\mathbf{x}_k^g$ is selected if it projected at each neighboring frame within the radius of r_d. Each inserted virtual point forms an independent landmark point track with all occurrences in the broken frame and its neighboring frames.

4 Virtual Frame Insertion

In real-world scenario, a common cause for a "visual break" is that two consecutive frames do not share overlapping FOV due to rotating camera too fast i.e. angular velocities are



(a) (b)

Figure 2: (a) Illustration of virtual points insertion. The red virtual points in the overlapping field of view are selected. (b) Illustration of virtual frame insertion.

too high. In this case, no feature correspondences can be extracted from the two consecutive frames. Hence, virtual frames need to be inserted to fill the gap before the insertion of virtual points. Figure 2(b) shows the illustration of virtual frame insertion technique. We refer this kind of "visual break" as non-overlapped "visual breaks" (NVB).

4.1 Non-overlapping "Visual Break" Detection

The way of detecting non-overlapping "visual breaks" is to find motions with large changes in rotation angles. Let $\Delta\mathbf{P}_{i-1\to i}$ be the relative camera pose transformation from time $(i-1)$ to time i, which can be estimated by $\Delta\mathbf{P}_{i-1\to i} = \mathbf{P}_i \cdot \mathbf{P}_{i-1}^{-1}$. With $\Delta\mathbf{P}_{i-1\to i}$ obtained, rotation matrix $\Delta\mathbf{R}_{i-1\to i}$ and translation vector $\Delta\mathbf{T}_{i-1\to i}$ can be extracted:

$$\begin{bmatrix} \Delta\mathbf{R}_{i-1\to i} & \Delta\mathbf{T}_{i-1\to i} \\ \mathbf{0} & 1 \end{bmatrix} = \Delta\mathbf{P}_{i-1\to i} \tag{4}$$

It is easier to calculate rotation angle with rotation vector $\Delta\mathbf{v}_{i-1\to i}$ which can be obtained by Rodrigues' rotation transform: $\Delta\mathbf{v}_{i-1\to i} = \text{Rodrigues}(\Delta\mathbf{R}_{i-1\to i})$. Let $\Delta\theta_{i-1\to i} = \|\Delta\mathbf{v}_{i-1\to i}\|$, it is straightforward to consider location i to be a non-overlapped break frame if $\Delta\theta_{i-1\to i} > \theta_t$, where θ_t is a threshold depending on the FOV of the camera.

4.2 Virtual Frames Insertion

When $\Delta\theta_{i-1\to i}$ is large, i.e. the rotation angle between two consecutive frames is large, one virtual frame may not be enough to fill the gap. So first we need to determine the number of virtual frames to be inserted to location i, which we define as

$$N_i^{vf} = \frac{\Delta\theta_{i-1\to i}}{\alpha\theta_t} \tag{5}$$

in which $\alpha \in (0,1]$ is a scale factor used to adjust the density of inserted virtual frames ($\alpha = 1$ is used in our experiments). Then we can obtain the incremental rotation matrix $\delta\mathbf{R}_i$

and translation vector $\delta\mathbf{T}_i$ by

$$\delta\mathbf{R}_i = \text{Rodrigues}\left(\Delta\mathbf{v}_{i-1\to i}(N_i^{vf})^{-1}\right) \tag{6}$$

$$\delta\mathbf{T}_i = \delta\mathbf{R}_i^{-1}\Delta\mathbf{T}_{i-1\to i} \tag{7}$$

The pose for the k^{th} inserted virtual frame can be obtained by:

$$\mathbf{P}_{i,k}^{vf} = \begin{bmatrix} \delta\mathbf{R}_i & \delta\mathbf{T}_i \\ \mathbf{0} & 1 \end{bmatrix}^k \cdot \mathbf{P}_{i-1} \tag{8}$$

After inserting the virtual frames, we can insert virtual points on every virtual frame created and its neighboring frames. The algorithms of virtual points insertion and virtual frames insertion are shown in Algorithm 1 and Algorithm 2 respectively.

Algorithm 1: Virtual Points Insertion

Input: N: number of frames
$\{\mathbf{P}_1, ..., \mathbf{P}_i, ..., \mathbf{P}_N\}$: pose at each frame
Output: New virtual point tracks
for $i = 1$ **to** $N-1$ **do**
  Set visual break flag b_i =TRUE;
  for $j = i+1$ **to** N **do**
    Match frame i with j;
    if $c_{i,j} > t_n$ **then**
      b_i =FALSE;
      Break;
    end
  end
end
for *every break location* i **do**
  Generate n virtual points $\mathbf{X}^l$;
  $\mathbf{X}^s$=$\mathbf{X}^l$;
  for $k = 1$ **to** n **do**
    for $j = i - r_d$ **to** $i + r_d$ **do**
      Calculate image location $\mathbf{x}_k^{2D}$;
      if $\mathbf{x}_k^{2D}$ *is out of the image frames* **then**
        Remove $\mathbf{x}_k^l$ from $\mathbf{X}^s$;
      end
    end
  end
  if $|\mathbf{X}^s| = 0$ **then**
    Remove $\mathbf{x}_k^l$ from $\mathbf{X}^s$;
  end
end

Algorithm 2: Virtual Frame Insertion

Input: N: Number of frames
$\{\mathbf{P}_1, ..., \mathbf{P}_i, ..., \mathbf{P}_N\}$: Pose at each frame
Output: Updated poses and point tracks
for $i = 2$ **to** N **do**
  Calculate $\Delta\mathbf{P}_{i-1\to i}$;
  Extract $\Delta\mathbf{R}_{i-1\to i}$ and find $\Delta\theta_{i-1\to i}$;
  if $\Delta\theta_{i-1\to i} > \theta_n$ **then**
    mark i as a non-overlapping "visual break";
    Find N_i^{vf};
    for $k = 1$ **to** N_i^{vf} **do**
      Calculate $\mathbf{P}_{i,k}^{vf}$;
      Insert virtual points to the virtual frames and their neighboring frames;
    end
  end
end

4.3 Iterative Optimization

Since serious drifts exist in the initial estimation of poses, errors may be induced by virtual insertion since they are estimated based on initial poses. An iterative optimization mechanism should be adopted to achieve better accuracy and robustness. After each iteration, average 2D projection error of all landmarks is calculated. Iterations continue until projection error is less than the pre-set threshold. Each iteration starts with updated poses output from previous iteration, and all the virtual frames and virtual points will be re-calculated.

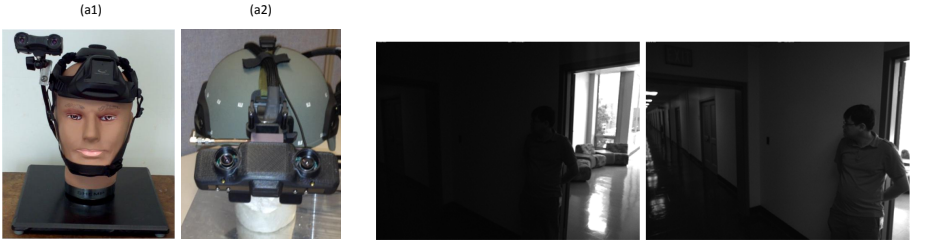


Figure 3: (a) Two different helmet-based sensor rigs used in the experiment. (b) A typical example of "visual break" in sequence 1. Only few feature correspondences can be linked for two neighboring images due to sudden illumination change and quick head movement.

5 Experiments

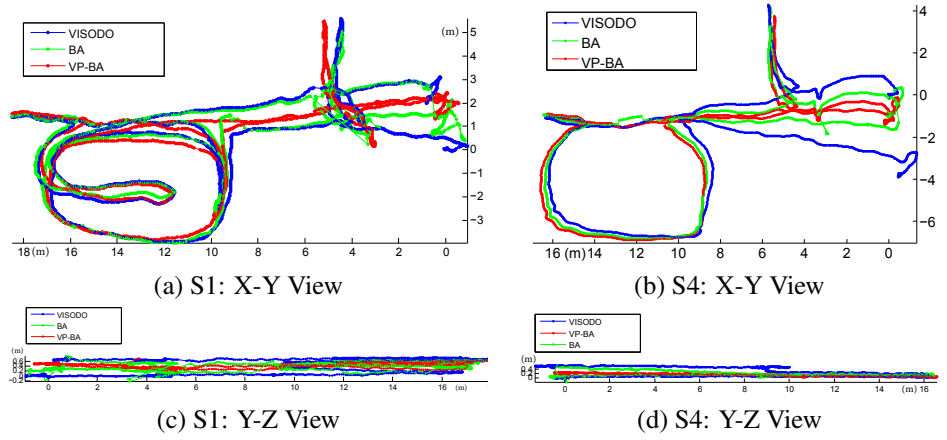
The proposed algorithm is evaluated on several datasets collected by two different multi-sensor rig as shown in Figure 3(a). Both of them consist of a pair of stereo cameras and an IMU. During the experiments, a person wearing the helmet walked and ran around inside a large building with hallways and different rooms. Four typical real-world indoor navigation sequences were collected, and each sequence has numbers of "visual breaks" and non-overlapping "visual breaks". A typical example of a "visual break" is shown in Figure 3(b), in which few correspondences can be linked from the images caused by sudden illumination change. Table 1 summarizes the number of "visual breaks" for each sequence.

To demonstrate the effectiveness on drift removal of bundle adjustment, all the sequences are collected purposely to contain loop closures at the end. The performance of the proposed algorithm is evaluated by two criteria. First, 3D reconstruction consistency over the whole trajectory is evaluated by 3D reconstructions of a set of selected landmarks in the scene observed at difference locations along the trajectory. The coordinates of each landmark observed at different locations should be very similar. Therefore the difference of the 3D coordinates of the same landmark observed at various locations with different time stamps reflects the consistency of 3D reconstruction accuracy. For each sequence, around 100 scene landmarks along the trajectory are selected for the evaluation. The other criteria is loop closure distance at the end, which is used to evaluate the drift removal performance of bundle adjustment.

A multi-core bundle adjustment implementation [15] is adopted in all the experiments. In addition, the basic Visual Odometry algorithm is implemented with the extended Kalman filtering fusing technique [14] that will fuse both camera poses and the IMU measurements for optimal camera pose estimation.

For each sequence, the performance of visual odometry system (VISODO), bundle adjustment (BA) and bundle adjustment with the proposed visual insertion technique (VP-BA) is demonstrated. Table 1 summarizes quantitative results of output from all three methods. Note that the output of VISODO is the output of Kalman filter fusing the motion estimates of IMU and Visual Odometry, so it is always continuous even if "visual breaks" exist. For each location containing "visual breaks", VP-BA generates 60 virtual point candidates.

Sequence 1 is the most challenging sequence. There are 15 "visual breaks" and 5 non-overlapping "visual breaks", and the VISODO output drifted 2.4m at the end. Due to the



(a) S1: X-Y View (b) S4: X-Y View

(c) S1: Y-Z View (d) S4: Y-Z View

Figure 4: Estimated 3D trajectories of (a)(c) sequence 1, and (b)(d) sequence 4.

Table 1: Summary of number of "visual breaks" for experiment sequences and summarized quantitative results of the consistency of 3D reconstruction accuracy and loop closure drift (LCD). (NVB denotes non-overlapped "visual breaks").

Seq #	Key Frames	Visual Breaks	NVB	Method	LCD (m)	Consistency Error (m)		
						Max	Min	Mean
1	1629	15	5	VISODO	2.3722	2.2345	0.9811	1.5284
				BA	0.5615	1.3235	0.0787	0.7985
				VP-BA	0.0512	0.1271	0.0334	0.0772
2	833	5	0	VISODO	1.7873	0.9491	0.3495	0.6181
				BA	1.1461	0.0984	0.0336	0.0649
				VP-BA	1.1227	0.0602	0.0104	0.0396
3	1388	8	1	VISODO	0.8679	0.5935	0.2102	0.4272
				BA	0.6317	0.4505	0.0174	0.1942
				VP-BA	0.5283	0.1367	0.0061	0.0828
4	1392	12	2	VISODO	3.8265	3.9768	0.9070	2.6742
				BA	0.3915	1.5386	0.5809	0.9729
				VP-BA	0.0370	0.1426	0.0284	0.0849

existence of "visual breaks", the BA failed in general. Large jumps up to 3m and barely corrected drifts observed from the BA estimated trajectory as shown in Figure 4. However the VP-BA obtained a good performance as shown in Figure 4(a)(c), and jumps have been eliminated completely with proposed virtual insertion technique. Table 1 shows significant improvements of VP-BA over BA. Mean 3D reconstruction consistency error has been reduced from 0.7985m to 0.0772m, and loop closure drift has been greatly reduced from 0.5615m to 0.0512m without any jumps. Both mean 3D reconstruction consistency error and loop closure drift correction of VP-BA have ten fold improvement compared to BA.

The sequence 2 and sequence 3 are relatively easier, and the drift at the end of VISODO output is around 1m. The estimated 3D trajectories by three methods are shown in Figure 5(a) and Figure 5(b-d) respectively. Similarly we can observe that VP-BA eliminated all jumps and improved bundle adjustment performance. Table 1 summarizes that the mean error of 3D reconstruction consistency of the whole trajectory has been reduced from 0.0649m to 0.0396m for sequence 2, and reduced from 0.1942m to 0.0828m for sequence 3.

Sequence 4 is another challenging one. The VISODO output drifted 3.8m at the end as shown in Figure 4(b)(d). Besides large jumps in the segments containing "visual breaks",

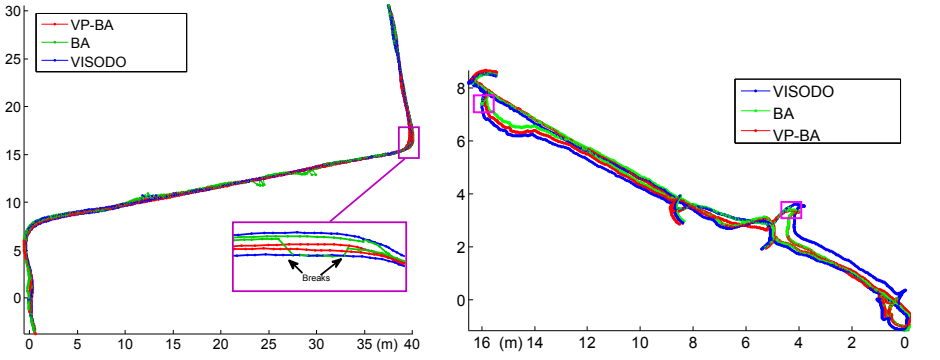


Figure 5: Estimated 3D trajectories of (a) sequence 2, and (b) sequence 3.

the BA even failed to close the loop at the end of trajectory. However, as shown in Figure 4(b)(d), VP-BA eliminated all the large jumps and closed the loop. Table 1 shows the mean consistency error of 3D reconstruction accuracy over the whole trajectory estimated by VP-BA is 0.0849m compared to 0.9729m for BA, and VP-BA also has ten fold improvement on loop closure drift reduction compared to BA, demonstrating that bundle adjustment with visual insertion technique is able to correct the drift and achieve consistent global accuracy over the sequence containing "visual breaks" and large drifts.

Note that in our experiments, adding virtual insertion does not increase computation time noticeably in bundle adjustment since only a few locations with "visual break" needs insertion. Hence the proposed scheme is inexpensive compared to other sensor fusion based techniques and suitable for real-time SfM applications.

In summary, we can see that, as expected, the performance of VP-BA, in terms of the accuracy and the global consistency, is improved dramatically over the traditional multi-sensor-based Visual Odometry system and the traditional Bundle Adjustment over these challenging real-life navigation datasets.

6 Conclusions

This paper addresses the issue of visual scene landmark outage caused by missing correspondences ("visual breaks") in bundle adjustment over long video sequences. We propose a "Virtual Insertion" scheme to address this issue by filling the visual link outage in structure-from-motion applications with the use of other non-vision sensors or motion assumptions. It provides a novel approach to incorporate information from non-vision sensors or motion assumptions, by constructing virtual points and virtual frames. These virtual points and frames are used to bridge the gap due to "visual breaks" for bundle adjustment. As a result, it enables the traditional bundle adjustment algorithms to achieve robust large-area structure from motion over long video sequences. Future work includes analyzing and improving the virtual linking strength and flexibility formed by virtual points and virtual frames, specifically the optimal number and location of virtual points to be inserted to a location with "visual break", in order to connect the "break" properly without affecting the accuracy of the neighboring locations in bundle adjustment.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building Rome in a day. In *CVPR*, 2009.

[2] Sameer Agarwal, Noah Snavely, Steven M. Seitz, and Richard Szeliski. Bundle Adjustment in the large. In *ECCV*, 2010.

[3] Chris Beall, Frank Dellaert, Ian Mahon, and Stefan B. Williams. Bundle adjustment in large-scale 3D reconstructions based on underwater robotic surveys. In *OCEANS*, 2011.

[4] H Chiu, S Williams, F Dellaert, S Samarasekera, and R Kumar. Robust vision-aided navigation using Sliding-Window Factor Graphs. In *ICRA*, 2013.

[5] Zhijun Dai, Fengjun Zhang, and Hongan Wang. Robust Maximum Likelihood estimation by sparse bundle adjustment using the L1 norm. In *CVPR*, June 2012.

[6] Nicola Fioraio and Luigi Di Stefano. Joint detection, tracking and mapping by Semantic Bundle Adjustment. In *CVPR*, June 2013.

[7] J. Hedborg, P-E Forssen, M. Felsberg, and E. Ringaby. Rolling shutter bundle adjustment. In *CVPR*, 2012.

[8] Yekeun Jeong, David Nistér, Drew Steedly, Richard Szeliski, and In-So Kweon. Pushing the envelope of modern methods for bundle adjustment. *IEEE PAMI*, 34(8):1605–17, August 2012.

[9] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental Smoothing and Mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, December 2008.

[10] Kurt Konolige. Sparse Sparse Bundle Adjustment. In *BMVC*, 2010.

[11] Maxime Lhuillier. Incremental fusion of Structure-from-Motion and GPS using constrained bundle adjustments. *IEEE PAMI*, 34(12):2489–95, December 2012.

[12] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real Time Localization and 3D Reconstruction. In *CVPR*, 2006.

[13] E.D. Nerurkar, K.J. Wu, and S.I. Roumeliotis. C-KLAM: Constrained keyframe-based localization and mapping. In *Workshop: Multi-View Geometry in Robotics, Robotics: Science and Systems*, 2013.

[14] Taragay Oskiper, Zhiwei Zhu, Supun Samarasekera, and Rakesh Kumar. Visual odometry system using multiple stereo cameras and Inertial Measurement Unit. *CVPR*, June 2007.

[15] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M. Seitz. Multicore bundle adjustment. In *CVPR*, June 2011.

[16] Zhiwei Zhu, HanPang Chiu, Taragay Oskiper, Saad Ali, Raia Hadsell, Supun Samarasekera, and Rakesh Kumar. High-precision localization using visual landmarks fused with range data. In *CVPR*, June 2011.