GEA Optimization for Live Structureless Motion Estimation

A.L. Rodríguez
DITEC
Universidad de Murcia
alrl1@um.es

P.E. López-de-Teruel
DITEC
Universidad de Murcia
pedroe@um.es

A. Ruiz
DIS
Universidad de Murcia
aruiz@um.es

Abstract

In this paper we describe a highly efficient and scalable real-time camera motion estimation system. Instead of Bundle Adjustment, this system uses Global Epipolar Adjustment (GEA) [11] to correct bootstrapping and loop-closing errors during the camera tracking process. We propose a modification of the GEA algorithm to obtain a significant speed-up in the optimization, without sacrificing loop-closing error correction performance.

As a result, our motion estimation system features increased performance and scalability. In addition, it can perform long-term live motion estimation without explicitly maintaining the whole 3D structure. The system stores only the 3D location for the most recently detected features, to resect the camera pose for new input frames.

1. Introduction

Real-time SfM techniques obtain a sparse reconstruction and the camera poses corresponding to certain key-frames from an input video sequence. This reconstruction is usually the starting point of live dense reconstruction algorithms in systems such as [10].

During the reconstruction process Bundle Adjustment (BA) [12] is used to correct bootstrapping and loop-closing errors, improving the estimated camera poses and 3D points. This is generally required in SfM systems to prevent long-term camera tracking failures [5]. More important, it increases the reconstruction accuracy, so that dense algorithms can obtain better reconstructions.

GEA [11] is a multiple-view camera pose optimization algorithm which minimizes the algebraic errors defined between pairs of views in the reconstruction. Gross errors are corrected for the estimated camera poses with a significantly smaller computational cost than BA.

Experimental evidence shows that taking certain precautions, the optimization of the algebraic error generally provides camera poses remarkably close to the optimal geometrical error. In this paper we describe a real-time SfM system which uses GEA as a stand-alone error correction technique to perform an accurate live camera tracking.

By omitting the BA refinement an interesting theoretical result is obtained: there is no need to store and update the whole 3D structure during the reconstruction process to correct the initialization and drift errors in the camera tracking. Our SfM system uses a small fraction of the 3D points only to resect the camera poses for new input frames.

The whole 3D structure can be estimated with a simple linear triangulation when it is required; for example to bootstrap a dense reconstruction process. As demonstrated in [11] this linear structure estimation is very accurate when the camera poses are sufficiently close to the optimal configuration.

To satisfy real-time constraints, common SfM systems must apply BA either limited to the most recent key-frames [5], or performed in a thread separated from the live camera tracking process [8]. The first approach cannot correctly eliminate loop-closing errors due to dead reckoning, because the correction should involve a large number (if not all) of the camera poses.

The second approach has practical limitations regarding the number of views. Bundle Adjustment can correct only a few hundred key-frames on commodity hardware, while keeping up with the real-time camera tracking process. As the map grows, BA will not be able to correct the reconstruction before new camera poses are resected, so the probabilities of tracking failure increase significantly.

A possible solution is to use a GPU implementation of BA [13] to speed up the map optimization. This can be inconvenient if the GPU is required for another task, such as the dense reconstruction. In this case the BA optimization and the dense reconstruction algorithm must share the high-performance hardware, and the overall performance speed of the application can decrease significantly.

Our SfM system implements a mixed approach to real-time map optimization. On one hand, GEA corrects the camera poses of the n key-frames most recently added to the reconstruction. On the other hand, a modified version of GEA optimizes the whole set of camera poses in the reconstruction when new loop-closing information is detected, to correct possible camera drift errors.

In this paper we introduce and describe this GEA modification, which can accelerate the loop-closing correction without sacrificing significant error correction performance.

This technique is suitable for real-time SfM and does not discards views in the optimization. Our initial experiments show that it improves the scalability of the loop closing correction, in a way that the running time grows almost linear with the number of views.

The proposed tracking system is more scalable and efficient than other alternatives based on real-time SfM techniques. It provides globally optimized camera poses, accurate enough for tasks such as augmented reality, or live dense reconstruction.

2. Global Epipolar Adjustment

Like Bundle Adjustment, the Global Epipolar Adjustment is a technique for map error correction. Instead of optimizing the geometrical reprojection error as BA, GEA optimizes the algebraic residuals corresponding to the fundamental relationships defined between pairs of views in the reconstruction. Assuming a calibrated scenario, the expression for the GEA error function is the following:

$$C_{ge} = \sum_i \sum_{j \neq i} \sum_{k \in \mathcal{P}_{ij}} \left(\mathbf{x}_{ik} \; E_{ij} \; \mathbf{x}_{jk}\right)^2 \qquad (1)$$

where: $\mathcal{P}_{ij}$ contains the indexes for set of points which are visible at both views i and j; $\mathbf{x}_{ik}$ and $\mathbf{x}_{jk}$ represent the measured projections of the 3D point k at images i and j; and E_{ij} is the essential matrix that models the epipolar geometry defined between the pair of views i and j. This matrix can be evaluated for views i and j from their camera pose orientations R_i, R_j and centers T_i, T_j:

$$E_{ij} = R_i \left[\frac{T_j - T_i}{||T_j - T_i||} \right]_\times R_j^T \qquad (2)$$

The algebraic error in expression 1 can be conveniently rewritten to:

$$C_e = \sum_i \sum_{j \neq i} \|A_{ij} \mathbf{e}_{ij}\|^2 \qquad (3)$$

where: $\mathbf{e}_{ij}$ is a vectorized version of the matrix E_{ij}; and A_{ij} is the *measurement matrix* corresponding to the point matchings detected between the views i and j (equation 8 in [14]).

Each measurement matrix A_{ij} in expression 3 can be replaced by a reduced measurement matrix of size 9×9 [7]. The GEA cost error obtained is mathematically equivalent to the original, and its optimization is significantly faster.

Using a convenient coordinate normalization as described in [6], and assuming the essential matrix expression 2, the camera poses obtained by optimizing the algebraic error are very similar to those corresponding to the optimal geometric error for the two-views case [7].

As for BA, Levenberg-Marquardt is adequate to optimize the cost error function for GEA. Most of the performance improvements obtained during years of research for the LM optimization in BA can be used as well to optimize the GEA cost error. Amongst them, exploiting the sparsity of the Jacobian of the cost error [5, 9], or using preconditioned conjugate gradient to solve the second level system [1].

3. GEA-based motion estimation

Our proposed motion estimation system combines a live camera tracking process with a long-term drift error corrector. The live process estimates the camera pose for new input frames from the video sequence using an image point tracker such as KLT [2]. It also updates the reconstruction with new key-frames, corrects initialization errors using an incremental version of GEA, and performs loop-closing detection using a fast image descriptor matching.

Once a new loop-closing matching is found, the long-term error corrector is launched in a different thread. This process corrects drift errors using a version of the GEA optimization specially accelerated for this kind of live reconstruction problems. When this optimization finishes, the camera tracking at the live process is updated, and the long-term error corrector thread is destroyed.

Both processes, short and long term tracking are structureless versions of the tracking and mapping processes from the PTAM system [8] respectively. In this case they are more decoupled, in the sense that success of the live process does not depend on the long-term drift error corrector to eliminate camera resection errors. Camera resection errors are corrected on the main process without having a global reconstruction correction running constantly in the background.

The following two subsections describe the camera tracking and long-term error correction processes in detail.

3.1. Short-term camera tracking

The problem of short-term camera tracking is well studied, and to some extent can be considered solved [4]. We use an incremental approach such as [5] to estimate the camera pose for every frame of the video sequence.

In incremental SfM, certain key-frames are registered in the reconstruction during the tracking. These key-frames should have a minimal base-line distance between them, so the reconstruction does not contain redundant views, and the structure is well constrained. Traditional SfM systems also keep and update the 3D location of the image features detected at the scene. For large tracking sequences, this structure can contain from thousands to hundred thousands of 3D points.

To resect the camera pose for new input frames during the tracking, our system stores only a reduced set of 3D points (about 100-200 points) corresponding to image features detected at the most recently registered key-frames.

The actual frame is added as a key-frame in the reconstruction when a sufficiently large baseline is detected between the actual camera pose and the most recently registered key-frame. Then, an incremental GEA optimizes the last n_c key-frames in the reconstruction. With these few optimized camera poses a new set of 3D points is estimated to resect camera poses for new input frames.

This prevents future camera resection errors, hence it improves the success rate of the tracker [5]. It is typically sufficient to optimize a number of $n_c = 10$ key-frames to avoid most tracking failures, and this optimization can be performed in real-time.

Each time a new frame is added in the reconstruction, the tracker compares this new key-frame with older key-frames, to find loop-closing correspondences. The simple image matching technique described at [3] obtains a single image descriptor for each key-frame, and compares them to find image matchings.

Once a possible matching is found between two images, a RANSAC search looks for image point matchings. If a sufficient amount of point matchings was detected, the tracker evaluates the reduced matrix from the point matchings detected, and includes it in the GEA cost error.

3.2. Lightweight GEA for drift error correction

When new loop-closings are detected the tracker uses GEA to correct the camera poses for the key-frames in the reconstruction.

The GEA optimization can be accelerated by discarding terms in the GEA cost error function C_e (from equation 3). Our tests demonstrate that most of these terms are redundant, and have little influence in the loop-closing error reduction obtained by GEA.

We discard terms corresponding to reduced matrices obtained from a small number of point correspondences. To ensure full connectivity in the view graph, we do not discard terms corresponding to key-frames which are too close in the video sequence. This way the reconstruction will not be partitioned into different groups which would be optimized separately by GEA.

This selection involves two parameters. The parameter s indicates the maximal distance between the two key-frames involved in a term to be included the GEA cost error. The parameter k indicates the maximum number of terms to include that involve views sufficiently separated in the video sequence, and involve a large number of point correspondences.

Given the set of terms from the GEA cost error $\{\|A_{ij}\mathbf{e}_{ij}\|^2\}_{i,j=1..n}$, the selection will preserve those such that $|i - j| <= s$. For each view v, the selection will also preserve the k remaining terms contained in $\{\|A_{ij}\mathbf{e}_{vj}\|^2\}_{j=1..n}$ that involve the largest amount of point correspondences.

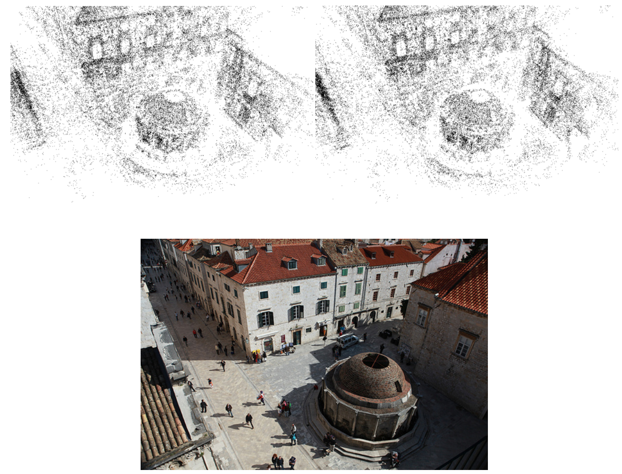

Figure 1. Visual comparison of the reconstruction obtained with sSBA and GEA. SfM reconstruction of the fountain optimized with sSBA, and optimized with GEA using linear estimation of the 3D points (top row left and right image respectively). Top view of the Onofrio fountain at Dubrovnik (bottom image).

4. Results

In this section we compare the performance of our GEA implementation against sSBA [9] (which is the fastest and more accurate state-of-the-art open implementation of Bundle Adjustment available) in real-time SfM problems. We also evaluate the suggested simplification for the GEA error described in section 3.2.

In our tests we used the data set *Dubrovnik*[1], obtained with SfM techniques from unordered sets of images from the Internet; and the data-sets *boxes-8* and *boxes-xl*[2] obtained with a real-time SfM reconstruction system which uses incremental BA to correct short-term camera tracking errors, and performs loop-closing detection using the technique described at section 3.1.

Figure 1 shows a visual comparison of the optimal reconstructions obtained with sSBA and GEA for the data-set *Dubrovnik*. Both reconstructions represent quite accurately the scene from the fountain square in Dubrovnik.

Figure 2 shows a comparison of loop-closing error correction performed by sSBA and the original GEA optimization.

Figure 3 shows a comparison of the error reduction speed obtained with the proposed improvement for GEA, and sSBA on the data-set *boxes-xl*.

[1] Referenced in [1]

[2] These two data-sets are available at `http://perception.inf.um.es/gea`

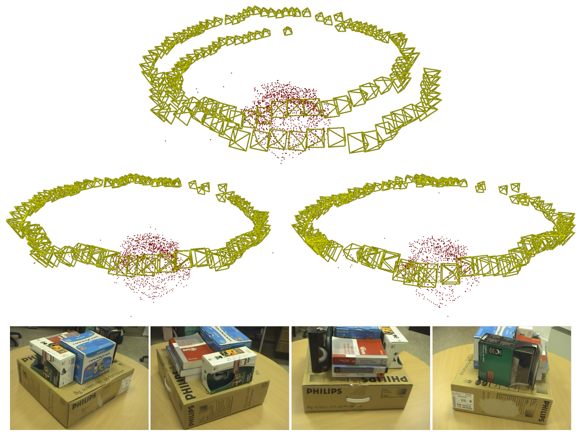

Figure 2. Reconstruction visualization for the data-set *boxes-8* without loop-closing correction (top figure); optimal reconstruction obtained with Bundle Adjustment and GEA including loop-closing information (left and right figures respectively, middle row). Some frames from the video sequence *boxes-8* (bottom row).

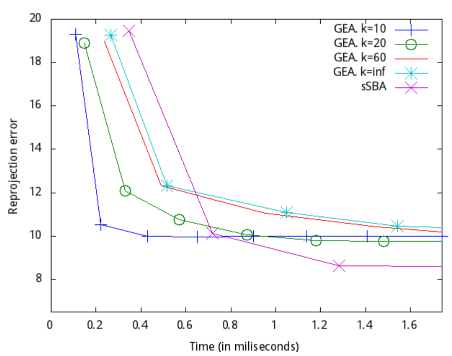

Figure 3. Performance comparison between simplified GEA and sSBA for the case study data-set. Data-set *boxes-xl*. Marks in each curve indicate the reprojection error achieved with a given number of iterations.

The λ parameter for both sSBA and GEA is fixed to 10^{-3}, which shows the best error correction speed for the sSBA algorithm. The measured performance times for GEA do not include the evaluation of the reduced matrices, or the posterior 3D structure updating. It is assumed that the reduced matrices were already obtained during the camera tracking or the loop closing detection. The value for the s parameter, corresponding to the improved GEA algorithm is set to 10 for these tests.

5. Conclusions

We present a real-time camera tracking system which proves to be more scalable and efficient than traditional SfM systems on commodity hardware (without using high-performance hardware such as GPU).

GEA is used instead of Bundle Adjustment to optimize the reconstruction during the tracking process. The accuracy obtained for the camera poses is sub-optimal, but quite close to the best reprojection error.

Drift errors are eliminated with a simple and fast loop-closing detection and a posterior global epipolar adjustment. This paper describes a modification of the GEA error specific for this kind of loop-closing error correction problem, that obtains significant performance speed-ups in the optimization.

The reconstructions obtained with this system are accurate enough for tasks such as augmented reality, or live dense reconstruction.

References

[1] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski. Bundle adjustment in the large. In *Proc. of ECCV*, 2010.

[2] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *Int. J. Comput. Vision*, 56:221–255, February 2004.

[3] R. O. Castle, G. Klein, and D. W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *Proc 12th IEEE Int Symp on Wearable Computers, Pittsburgh PA, Sept 28 - Oct 1, 2008*, pages 15–22, 2008.

[4] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: real-time single camera SLAM. *IEEE Trans. on PAMI*, 29(6):1052–1067, 2007.

[5] C. Engels, H. Stewnius, and D. Nister. Bundle adjustment rules. In *In Photogrammetric Comp. Vision*, 2006.

[6] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19:580–593, June 1997.

[7] R. I. Hartley. Minimizing algebraic error in geometric estimation problems. In *ICCV*, pages 469–476, 1998.

[8] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc.* 6^{th} *IEEE and ACM Int. Symp. on Mixed and Augmented Reality (ISMAR)*, 2007.

[9] K. Konolige. Sparse sparse bundle adjustment. In *Proc. of BMVC*, 2010.

[10] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference*, number 13-18 June 2010, pages 1498–1505, 2010.

[11] A. Rodriguez, P. L. de Teruel, and A. Ruiz. Reduced epipolar cost for accelerated incremental SfM. *Proc. of CVPR*, 2011.

[12] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 298–372, London, UK, 2000. Springer-Verlag.

[13] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. *In proceedings of CVPR*, 2011.

[14] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *Int. J. Comput. Vision*, 27:161–195, April 1998.