Scalable Structure from Motion for Densely Sampled Videos

B. Resch[1,2] H. P. A. Lensch[2,3] O. Wang[1] M. Pollefeys[3] A. Sorkine-Hornung[1]
[1]Disney Research Zurich [2]Tübingen University [3]ETH Zurich

Abstract

Videos consisting of thousands of high resolution frames are challenging for existing structure from motion (SfM) and simultaneous-localization and mapping (SLAM) techniques. We present a new approach for simultaneously computing extrinsic camera poses and 3D scene structure that is capable of handling such large volumes of image data. The key insight behind this paper is to effectively exploit coherence in densely sampled video input. Our technical contributions include robust tracking and selection of confident video frames, a novel window bundle adjustment, frame-to-structure verification for globally consistent reconstructions with multi-loop closing, and utilizing efficient global linear camera pose estimation in order to link both consecutive and distant bundle adjustment windows. To our knowledge we describe the first system that is capable of handling high resolution, high frame-rate video data with close to real-time performance. In addition, our approach can robustly integrate data from different video sequences, allowing multiple video streams to be simultaneously calibrated in an efficient and globally optimal way. We demonstrate high quality alignment on large scale challenging datasets, e.g., 2-20 megapixel resolution at frame rates of 25-120 Hz with thousands of frames.

1. Introduction

Structure from Motion (SfM), i.e., reconstructing camera parameters and sparse scene structure from images, has a long history in computer vision. Early approaches concentrated on reconstruction from videos based on feature tracking techniques (see, e.g., [12, 27] for an overview). With the advent of robust feature descriptors like SIFT, SURF or ORB (see, e.g., the survey [29]), larger view differences could be matched, and SfM techniques were successfully extended to very large scale, unstructured sets of images [5,26]. Coupled with the availability of online photo collections, these approaches have become very popular, enabling a wide range of novel applications such as 3D reconstruction from thousands of photographs [6, 34].

However, for individual users, and in personalized application domains, it is often much simpler and more practical to capture video instead of photographs to ensure a sufficiently broad as well as dense coverage of a scene. At the same time, video resolution and frame rate are constantly increasing. Mobile cameras such as the iPhone can record more than 120 frames per second, yielding thousands of images in just a few seconds of capture. Older approaches based on feature tracking as well as modern sparse feature point and SLAM-based approaches do not scale well to such densely-sampled data, both due to numerical inaccuracies arising from small baselines, and computational tractability associated with the sheer quantity of pixels.

But while densely sampled, high quality video data presents many challenges, it also provides opportunities. For instance, recent approaches have demonstrated how spatiotemporal coherence can be exploited in the context of 3D reconstruction [15, 33]. The key insight in this paper is that such data also enables novel, more efficient strategies for achieving globally consistent geometric calibrations.

Contributions. The main technical contributions that we propose are: a modification to KLT that allows for drift free tracking over thousands of frames, a robust selection of confident frames, a novel *interleaved* window bundle adjustment (BA) that makes optimizing large windows more efficient, uniform image coverage based point subsampling, robust frame-to-structure verification to obtain global, wide baseline anchors between camera poses, and the utilization of an efficient linear camera post estimation (LCPE) method that integrates information from both BA windows and global anchors in a unified way.

As opposed to prior work, our approach does not rely on analytical, fixed input size n-Point methods, which we observed to be not good enough due the fact that they use less data than BA, and therefore yield less precise results for the small baselines of densely sampled video input. Moreover, we combine results from piecewise camera track reconstruction, loop closing, and linking between different camera tracks into a single nonlinear optimization procedure. This allows different camera tracks to help each other to get a good initialization for global optimization.

When all of the contributions are combined, our method is able to obtain a globally consistent extrinsic camera cali-

bration in substantially less time then previous approaches, and on datasets with 1000s of high resolution frames. Furthermore, our approach generalizes to an arbitrary number of input video sequences, allowing for rapid, globally consistent calibration and scene reconstruction across multiple capture devices. In this paper we describe all of these contributions, and present detailed psuedocode for reimplementation in the supplemental material [1].

In our system we leverage established existing approaches, such as the commonly used KLT tracker [7], SIFT histogram based frame similarity cost matrices [30], a global linear solver that integrates relative camera pose constraints [14], and a robust depth-based point parameterization [33].

2. Related Work

Depending on underlying methodology and applications, a multitude of different terminologies exists for geometric camera auto-calibration, the most common being variants of structure from motion (SfM) and simultaneous localization and mapping (SLAM). Here we classify prior works into two categories according to their preferred input data; unstructured and sparse vs. coherent and dense sampling.

Unstructured, sparsely sampled input. A key challenge for methods focusing on sparsely sampled input such as photo collections is that the data is generally unstructured and heterogeneous, with significant appearance changes between images. The current state-of-the-art is therefore generally based on iterating (see [12]): (i) robust detection and matching of feature points, (ii) n-point algorithms to establish initial geometric relationships between views, and (iii) global BA. This approach has been successfully extended to massive, very large scale datasets [6, 11, 26], with various publicly available implementations [2, 4].

A central problem for such techniques is bootstrapping, i.e., finding a good global initialization for BA that includes all images, without having to run many iterations of BA on parts of the reconstruction. Martinec and Pajdla [20] present a robust solution for finding global camera poses, concentrating on the camera orientation. Wilson and Snavely [32] and Jiang et al. [14] show how to find global camera positions given known orientations. For massive datasets like Internet photo collections, a second problem is the sheer amount of images, often in the order of millions. To this end, techniques such as skeletal graphs [25] have been proposed, which remove unnecessary data by focusing on stable subsets of cameras. Agarwal et al. [5] showed that it is necessary to reconsider well established strategies in order to tackle large datasets consisting of 10s of thousands of images. A further alternative is to perform an incremental, piecewise reconstruction of a scene [23], and later assemble individual fragments based, e.g., on extracted scene point descriptors.

All these methods are tuned towards heterogeneous, unstructured data, and as a consequence have difficulty when applied to densely sampled, coherent image sequences. This is due to per-frame feature point detection and pose estimation using n-point algorithms causing unstable reconstructions, as well as computational inefficiency. In our experiments, we show that by explicitly considering coherence in the data, it is possible to achieve high quality reconstructions at significantly faster convergence and computation times.

Coherent, densely sampled input. In contrast to above, most techniques for densely sampled input such as video sequences are based on continuously tracking feature points throughout image sequences and iterative pose optimization techniques [12, 24, 27]. These original methods were designed for short, low resolution video sequences and did not consider multi-loop closing.

Particularly related are SLAM approaches and their variants, as their aim is to compute accurate camera poses of a dynamically moving camera from a video stream. Often, however, such techniques are limited with respect to the supported scene size [16] or require additional sensor modalities [17]. Real-time methods based on feature points [9] or dense, per-pixel tracking [21] are generally designed to provide as good as possible results with a small input lag, rather than a final, fully consistent and high quality reconstruction that globally optimizes the poses of all input frames. CoSLAM [35] combines data from cooperatively acquired videos as long as some of the cameras see the same content at the same time. Other approaches achieve real time performance, but only on preconstructed scenes [18], i.e., with known geometry.

Recently, a direct SLAM method (LSD-SLAM) was proposed [10], which does not require detection and tracking of feature points, but instead recovers sparse depth maps based directly on epipolar line scanning. However, such an approach does not scale well to high image resolutions, as it requires depth estimates for many pixels. In addition, depth recovery is very sensitive to accurate intrinsic calibration. Our approach instead focuses on a subset of reliable features tracks, which is more efficient and less sensitive to image distortions, especially for high resolution input.

Specific light field calibration techniques have been proposed for dense spatio-temporal-angular sampling using camera arrays [13, 15, 31] and plenoptic cameras [22]. However, these methods generally focus on the static geometric calibration of a light field, rather than computing both structure and motion, and hence cannot be applied to the acquisition scenarios we discuss in this paper. The work on unstructured light field acquisition [8] explores this to some extent, but only supports small scale scenes and focuses on an interactive interface for guiding the user during the acquisition process.

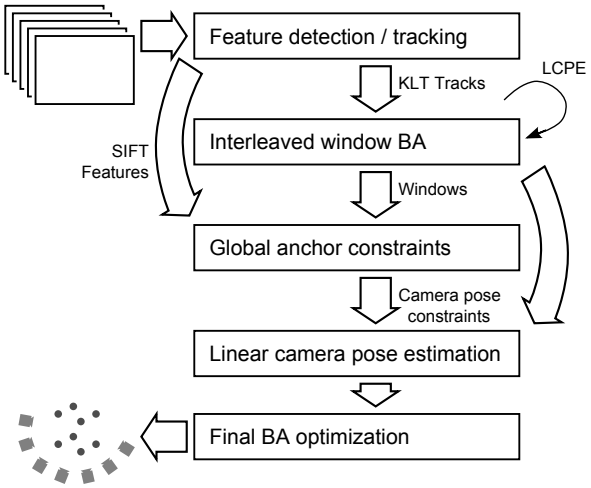


Figure 1. Algorithm overview. The feature detection stage computes KLT tracks for window BA and SIFT features for wide baseline handling. The window BA sweeps through the input, selects confident frames and computes camera pose constraints between interleaving sets of the selected frames. The global anchor stage uses the SIFT features to establish global links between different sections of the sequence. A linear camera pose estimation produces an initial arrangement of cameras which is further refined by bundle adjustment steps.

Our method focuses on densely sampled image sequences and overcomes several of the previously mentioned limitations. This results in a SfM approach that is stable and globally consistent over long, high resolution sequences, while still being able to robustly handle wide baseline matches.

3. Method

The input to our method is one or more image sequences. We focus on extrinsic calibration and assume the intrinsics to be fixed and known (in practice they can be computed from a few frames of the image sequences by using Bundler [2]).

On a high level our strategy is as follows. First, we perform a modified 2D tracking of feature points utilizing data coherence to reduce drift. Next, we apply a window BA strategy on a set of *confident* frames only. These are frames that are well connected via continuous tracks. To incorporate loop closing, we further establish global anchor links between carefully selected frame pairs of different parts of the video or even different video streams altogether. In addition to these global constraints, relative camera pose constraints from the window BA are integrated with an efficient linear camera pose estimation [14]. We then perform global BA, and finally add all the less confident images by interpolation and BA of their poses. During this step, we keep the scene structure fixed as determined by the confident images. The final result is a globally consistent calibration of all input frames from all input sequences.

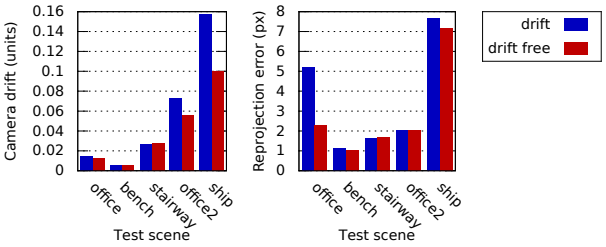


Figure 2. Influence of KLT drift on the reconstruction result. Note how drift free KLT tracks reduce the drift of the camera positions as well as the average reprojection error.

Figure 1 shows an overview of the key steps of our algorithm. The following sections discuss each step in detail.

3.1. Drift reduced feature tracking

Detectors optimized for wide baseline matching such as SIFT [19] compute incoherent feature point sets even between neighboring video frames. For continuous sequences, feature point tracking produces more reliable and efficient results. We build on the standard KLT tracker implemented in OpenCV [3], which is also the basis for earlier video-centered SfM techniques [27]. There are, however, two limitations of standard continuous KLT that have to be addressed in our application setting.

Firstly, we observed that for densely sampled video sequences, feature tracks that are visible for hundreds of frames exhibit noticeable drift. Note that for high frame rate cameras, this often corresponds to just a *second* of video. We therefore modify the basic tracking to perform a simple drift correction: when adding a frame, we track each feature from the previous frame to the new one, and then refine the feature position in the new frame using the *original* frame where the feature was detected. In our experiments this simple modification led to considerably reduced drift and higher reconstruction quality (see Figure 2).

Secondly, simply tracking points over an image sequence cannot guarantee any form of global consistency of the reconstructed cameras and scene. For example, when the camera revisits the same scene elements multiple times over a longer image sequence with intermediate occlusions, a single scene point will be represented by multiple, individually tracked and reconstructed points. This problem is known as the so called loop-closing problem in SLAM. For each feature track we therefore extract SIFT descriptors [19] in confident frames after the window BA, which is later used to re-identify points and for the generation of global anchor constraints.

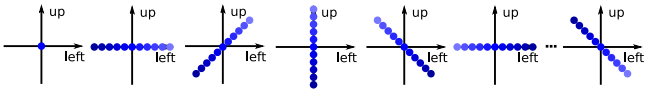


Figure 3. Illustration of different camera initializations. Cameras are arranged uniformly along a line orthogonal to the viewing direction in 45° steps.

3.2. Interleaved window bundle adjustment

Given feature tracks and descriptors, the goal of our window BA is to efficiently reconstruct camera poses for image sub-sequences without considering global consistency or jointly processing multiple individual sequences, both of which will be addressed later.

3.2.1 Window initialization

Initializing camera geometry is usually accomplished using n-point algorithms [12], which are (in contrast to BA) limited in the number of constraints and therefore generally not sufficiently accurate given the very small camera baselines encountered in high framerate video sequences. Our method is inspired by the approach of Yu and Gallup [33] designed for accidental small baseline camera motion.

We initialize a window by picking the first N consecutive images from an image sequence and immediately perform a BA step using the parameterization proposed in [33], where points are represented by inverse depth values projected from a reference frame (we use the center image in the window). We found, however, that identical initialization of all cameras [33] may cause BA to get stuck in local minima. According to our observations, this can reliably be avoided by starting from different linearly displaced configurations (see Figure 3) and optimizing first for the camera orientation and then for all extrinsics. Finally we pick the best result in terms of reprojection error. Moreover, we observed more robust results when initializing scene points with uniform instead of random depth [33]. The original method of Yu and Gallup requires a comparably large number of images for robust convergence. With our above modifications we observed stable convergence already with $N = 11$. For high frame rate handheld video, spacing between frames (e.g., 3 in our experiments) for slightly increased baselines led to improved convergence.

The next step is to grow this window. To this end, we first describe a subsampling scheme of the scene points that allows us to reduce the BA computational cost significantly at similar reconstruction quality.

3.2.2 Scene point subsampling

Following the observation that BA requires a certain minimum number of scene points but does not improve significantly with many more points, we employ a subsampling scheme on the available scene points in all BA steps. We found that sampling points randomly can lead to unstable or underconstrained optimization. We therefore choose point samples according to three rules explained below, achieving similar reconstruction quality at a fraction of the optimization time.

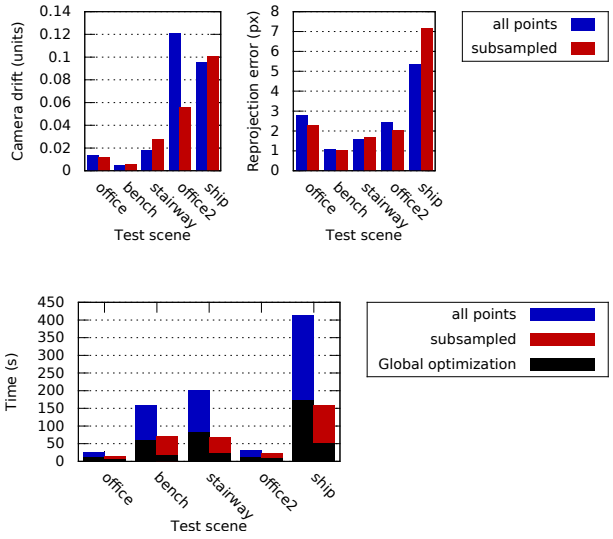


Figure 4. Comparison of reconstruction with and without point subsampling. Our subsampling strategy leads to comparable reconstruction results while reducing the computation time by roughly a factor of 3. This factor is assumed to increase on larger image resolutions since more points can be excluded there.

First, a minimum number τ of points should be visible from each camera. Second, the reprojected 2D positions of the points should be uniformly distributed in all camera images. Finally, the points should be visible in "sufficiently many" images, since in general a point provides more reliable constraints when seen by more cameras. At the same time, however, we observed that points visible in a very large number of frames, i.e. points with very long 2D tracks, are more likely to be affected by tracking errors along object silhouettes, corrupting the result. Each point is therefore picked with a probability proportional to the length of the track, but capped to no longer then 10 frames. This subsampling strategy resulted in a 3-fold speedup without sacrificing result quality (see Figure 4). For all experiments, we use a value of $\tau = 100$.

3.2.3 Confidence criteria

For efficiency reasons, and to improve result quality, we compute scene structure and camera poses initially only for a sparse set of confident frames. In order to find this set, we test cameras with a linearly increasing step size and add the furthest possible frame fulfilling a set of confidence criteria.

We define the following three confidence criteria ξ_1, ξ_2, ξ_3 for measuring whether a tested camera c is suitable for window BA.

The first term ξ_1 measures the number of features of the camera that can be matched to the points p_i of the window with a low reprojection error. This ensures that there are sufficiently many constraints for BA.

ξ_2 represents how far the camera moved around the scene points. We use the the median of all points' angular differences $\widetilde{\phi}(\overrightarrow{pc_p}, \overrightarrow{pc_n})$ between the vectors to the new c_n and the previous camera c_p. This term makes sure that two cameras are not too far apart from each other and ensures that the visual appearance of the feature points doesn't change to much so that the next confident frame has mostly the same feature tracks.

The last term ξ_3 is set to the median reprojection error $\widetilde{e}$ of the tested camera c_t and its visible points p_t: $\xi_3 = \widetilde{e}(c_t, p_t)$. This ensures that no cameras are added to the optimization which are too inconsistent with the content of the window.

We label a camera as sufficiently confident when the following criteria are fulfilled: $\xi_1 \geq 30$, $\xi_2 \leq 5°$, $\xi_3 \leq 5$px, i.e., the camera must be linked to at least 30 points, must not rotate more than five degrees around at least half of these points, and at least half of the points must have less than five pixels reprojection error. Similarly, a camera is labeled as candidate for removal from the current window as soon as it does not satisfy the following confidence constraints anymore: $\xi_1 \geq 70$, $\xi_2 \leq 10°$, i.e., at least 70 points and less than ten degrees of camera rotation around at least 50% of the points.

3.2.4 Window processing

Given the confidence criteria for addition and removal, in each iteration of the algorithm, we first remove images labeled for removal from the current window, keeping a minimum of 5 cameras in the window at all times. After this step, the current window contains usually about five to ten confident cameras.

However, for some camera (sub-)trajectories, stable windows can be much larger. To retain the efficiency of BA while keeping as much information as possible, we select a subset of the cameras in the window on which to perform the actual BA. We pick cameras with increased spacing for older images (see Figure 5). This subset is then optimized using standard BA.

After BA, all cameras in the current window are made consistent with a linear camera pose estimation technique [14], using the relative camera pose constraints of former windows. This solver works on the camera poses only, producing faster results than BA in comparable quality as long as the input is consistent.

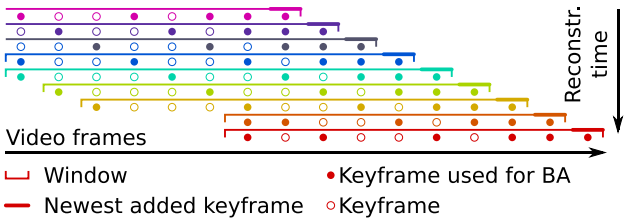


Figure 5. Selecting keyframes for interleaved window BA. Offsets between the keyframes selected for BA are linearly increasing towards older frames. To determine a consistent pose for a camera which was not part of the BA, we use the relative pose constraints that were generated in previous windows where the camera was part of the BA.

We experimented with various offsetting strategies besides the growth strategy described above. The linear increase provided the best results in terms of algorithm stability, camera sampling, and computation time. The output of this stage are camera pose constraints from each window, which we will later use for initializing the global scene optimization. We also keep all the windows for finding global anchor constraints as described in the following section.

3.3. Global anchor constraints

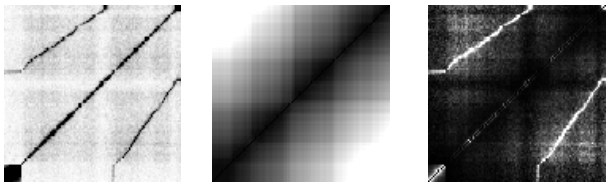
The goal of these constraints is to establish global links between different parts (possibly different subsequences) of a video that have shared scene content. These links can later be used in the linear camera pose estimation stage to obtain a good global initialization.

We establish these constraints by importance sampling frame pairs from the set of confident frames and by joining them based on SIFT features and the previously reconstructed window scene structure. To do this, we extract SIFT descriptors for all KLT features in the confident frames, and for each pair, try to integrate the camera of one frame using the scene structure associated with another frame using BA. The optimized camera pose is rated based on a confidence measure. Stable matching pairs among all possible confident frame pairs in the video sequence(s) are used as relative camera pose constraints for the global linear pose estimation stage (see Figure 1).

3.3.1 Camera stability

The stability of cameras for being used as global anchor constraints is based on the following measures ζ:

- The number of remaining points attached to a camera: $\zeta_1 = n$. This makes sure that there are enough constraints for optimization.
- The distribution of the point projections ρ in the image: $\zeta_2 = \min(\mathrm{Std}(\rho_x), \mathrm{Std}(\rho_y))$. This avoids unstable configurations with very localized feature positions.



$\mathbf{C}$ $\mathbf{V}$ $\mathbf{S} = (\mathbf{1} - \mathbf{C}) \circ \mathbf{V}$

Figure 6. Cost, variance and sampling matrices for wide baseline candidate picking. The camera circled an object twice. Dark parts of $\mathbf{C}$ indicate regions where good global anchor constraints are likely to be found. $\mathbf{S}$ shows where we sample for global anchor constraints.

- The ratio between the smallest and the largest principal component $PCA(p)_{\min}$, $PCA(p)_{\max}$ of the scene point positions p: $\zeta_3 = \frac{PCA(p)_{\min}}{PCA(p)_{\max}}$. This avoids using two-dimensional scenes which tend to be ambiguous (e.g. camera in plane) or unstable (e.g. frontoparallel plane).

3.3.2 Anchor selection

To find good anchor candidates for wide baseline links we use two measures: The cost for matching two frames, and the uncertainty of relative camera poses computed during the window BA.

Cost estimation. For robust estimation of the basic linking cost, we compute frame similarity based on histograms of SIFT features [30]. The output is a cost matrix $\mathbf{C}$ representing the cost for matching two frames of a video sequence (see Figure 6).

Uncertainty estimation. For uncertainty estimation, we approximate the variance of the camera poses for every pair of confident cameras in the following three steps:

1. Estimate the variance of each camera's pose c_i relative to each window's structure, i.e., the 3D points computed from cameras in window w_j:

$$Var(c_i, w_j) \propto 1/(\min(25, \zeta_1) \cdot \zeta_2 \cdot \zeta_3)^2 \quad (1)$$

   We assume that 25 reprojections are sufficiently many constraints.

2. Use this information to estimate the variance between windows by averaging the summed variances to common camera poses:

$$Var(w_{j_1}, w_{j_2}) = \sum_{i=1}^{n} \frac{Var(c_i, w_{j_1}) + Var(c_i, w_{j_2})}{n^2} \quad (2)$$

3. Find the camera→window→...→window→camera path with the lowest summed variance for each camera pair. While step 2 only considers variances for windows that share a camera, this step propagates the variance information to arbitrary indirectly connected camera pairs.

This results in a variance matrix $\mathbf{V}$ (see Figure 6). We can now estimate a matrix $\mathbf{S}$ representing potential anchor frames to be used as global links:

$$\mathbf{S} = (\mathbf{1} - \mathbf{C}) \circ \mathbf{V} \quad (3)$$

Note that $\mathbf{C}_{ij} \in (0, 1)$ and $\circ$ is the element-wise product of matrices. We importance sample $\mathbf{S}$ to get frame pairs (f_1, f_2) that represent useful anchor constraints.

Geometrical verification. To ensure that a global anchor constraint is truly useful, we perform a geometrical verification. We pick the window with the most available scene points that contains f_1 and BA for the pose of f_2's camera based on those points, utilizing SIFT matches for linking f_2's features to f_1's points.

In our experiments we observed that up to 40% of the matches were outliers when matching SIFT features extracted from KLT keypoints. Therefore, we exploit the already known scene geometry to gain robustness in this process. We apply four passes of BA for the camera pose parameters while removing all the points with reprojection errors worse than the average between the passes. Since BA tends to prefer consistent constraints, inconsistent reprojections are removed by this procedure. If there is not enough consistent data, BA diverges which leads to a violation of our stability constraints. We consider the geometric verification successful if it passes the following stability thresholds: $\zeta_1 \geq 25$, $\zeta_2 \geq 0.075 \cdot \mathit{ImageSize}$ and $\zeta_3 \geq 0.1$, which worked well in all our experiments. When a pair of frames representing a global anchor constraint fulfils these thresholds, we add the respective relative camera pose constraints to the existing set of constraints. In all our experiments, these thresholds reliably removed all outliers.

Figure 7 illustrates the effect of using the anchor constraints, based on sampling costs $\mathbf{C}$ and $\mathbf{S}$, which additionally takes into account variance matrix $\mathbf{V}$. Using anchor constraints considerably reduces camera drift. By concerning $\mathbf{V}$ in addition to the basic matching cost, drift can be reduced by another 40%.

3.4. Final optimization

The window BA and the global anchors now provide a large set of pose constraints. Using all these constraints we again apply global linear optimization [14] in order to compute a globally consistent 3D scene and camera calibration for all input frames.

We then apply a series of nonlinear least squares optimization passes based on the following three strategies:

A No Field of View (FoV) optimization, no bad point removal.

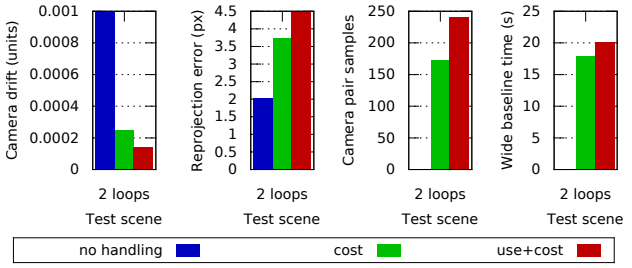


Figure 7. Comparison of loop closing strategies. Beside camera drift and reprojection error, we also show the number of samples needed to get 20 verified wide baseline links and the computation time for wide baseline handling. Without loop closing, the camera drift is quite high (out of scale: 0.08). Cost based frame selection for wide baseline handling reduces the drift drastically (use). Choosing frames also based on their value for wide baseline handling (encoded in $\mathbf{V}$) reduces the drift by another 40% for a fair amount of extra samples/runtime (use+cost). Note that the reprojection error increases because of the extra constraints that have to be fulfilled for closing the loop.

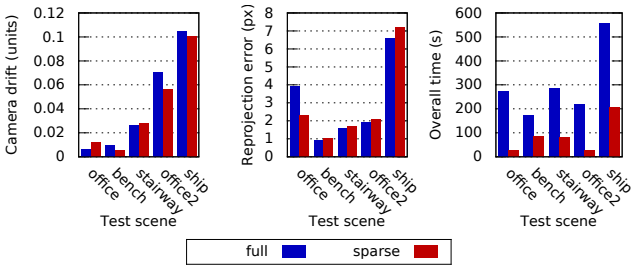


Figure 8. Keyframe selection evaluation. We compare our implementation optimizing only keyframes together with the points (sparse) to an implementation that does a full global BA (full). Using a sparse camera set yields results comparable to the full optimization but is 2-10x faster. This factor is assumed to increase with higher framerates since less keyframes are selected for the sparse set.

B No FoV optimization, bad point removal.

C FoV optimization, bad point removal.

We run the following sequence: `ABABABCCC`. Skipping bad point removal (`A`) at the beginning avoids the removal of reliable points because of a bad initialization, thus loosing valuable information for optimization. FoV optimization is added at the very end only (`C`), because it tends to converge to singularities in small or badly initialized scenes.

When all confident cameras are calibrated and corresponding stable scene points are reconstructed, we initialize the poses of all remaining, unused cameras by linear interpolation followed by a BA step constrained by the stable scene points.

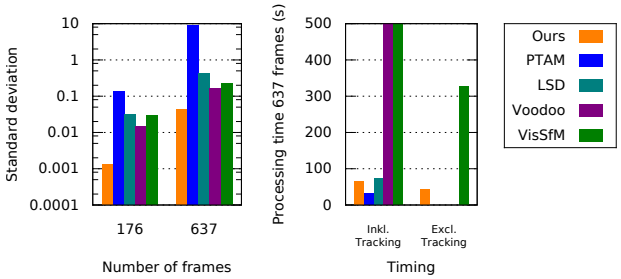


Figure 9. Evaluation based on synthetic ground truth. We give the standard deviation of the reconstructions fitted to the ground truth with an affine transformation plus timings. Our approach runs orders of magnitude faster than other SfM systems while producing results which are an order of magnitude more accurate than SLAM systems. PTAM failed after 176 frames due to too slow map update.

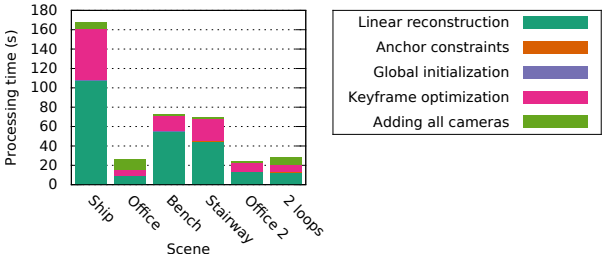


Figure 10. Breakdown of reconstruction timings to the individual pipeline parts.

4. Evaluation

In the addition to the quantitative evaluations shown throughout the paper, we provide additional results on ground truth data and timings. Most test scenes (office 1333 frames, bench 1055 frames, stairway 867 frames, office2 1180 frames) were recorded with a GoPro Hero 3 in Wide Angle 1080p 60fps mode. The ship scene (4411 frames) was recorded with a DSLR mounted on a slowly moving crane to simulate high frame rate footage. Our datasets and additional supplemental materials are provided on the accompanying project webpage [1].

Confident frame selection. In order to demonstrate the robustness of our confident frame selection process we compare the results of just using those frames in the final BA optimization to a full BA reconstruction using all frames. Figure 8 shows that there is only a small quality improvement at the cost of considerably increased compute time with the full BA.

Ground truth comparison. We have constructed a 2MP, 60 fps synthetic ground truth sequence from the Open Movie Project "Sintel" [28] containing rich scenery, motion blur, glare and camera shakes. Our method is more of a SfM approach than SLAM, as it features global BA steps

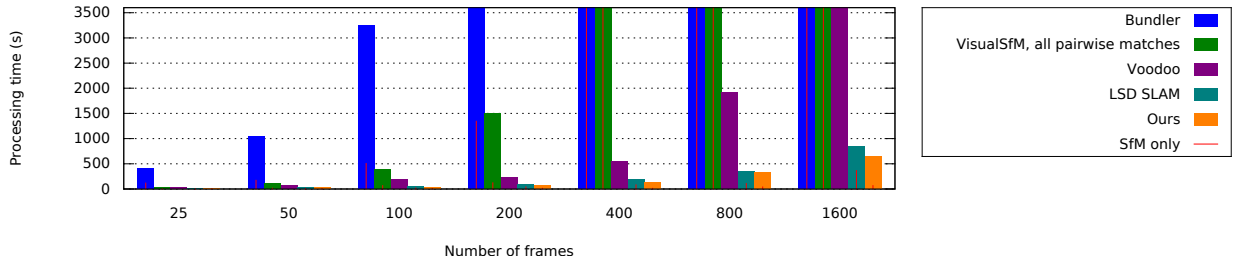


Figure 11. Computation time comparison for a FullHD (1080p) image sequence. Our technique exhibits the lowest computation time of all tested approaches. Note that if only the SfM part without feature processing is considered, we are about 6x faster than the nearest competitor. For 400, 800 and 1600 frames, we did not obtain reconstruction timings from all methods. Separate timings for IO/features and SfM reconstruction could not be obtained with Voodoo.

typical to SfM. However, Figure 9 shows that it runs orders of magnitude faster than SfM systems, at comparable speed to current SLAM systems, while producing results which are an order of magnitude more accurate than SLAM systems.

Timings. Figure 10 breaks down the reconstruction timings for the scenes used in this paper to the individual parts of our pipeline. Reconstruction timings of our approach are further compared with several other techniques in Figure 11 analyzing timings for varying numbers of frames of the FullHD outdoor video sequence. We compare to two approaches designed for handling images (Bundler [2] and VisualSfM [4] (GPU accelerated, parallelized)) as well as two approaches for video sequences (Voodoo Camera Tracker and the recent LSD-SLAM [10]).

Methods intended for sparse, unstructured data suffer from n^2 runtime for searching corresponding images. The Voodoo Camera Tracker performs well for small tracks but becomes much slower when BA has to correct accumulated drift in the end. Even in comparison to recent efficient SLAM approaches such as LSD SLAM our method is faster. We also observed that increased image resolution can lead to significant drops in performance for the tested methods, whereas our method scales well due to the proposed subsampling. We expect further significant speed gains by improved preprocessing such as feature extraction and tracking, as the majority of computing time is spent on these steps, and not our core optimization procedure (Figure 11).

Please also refer to the supplemental material on the project webpage [1] for reconstruction results on several other scenes, including very high resolution 5k video, multiple video sequence reconstructions and reconstructions from the Stanford Light Field datasets.

Limitations and future work. Our method currently computes only extrinsic camera parameters. As future work it would be interesting to support uncalibrated cameras with changing intrinsics. Moreover, the algorithm is limited by some of the components used. For instance, replacing the current OpenCV KLT tracking by a GPU based implementation and improving our point subsampling strategy, e.g., using stratified sampling, could lead to improved reconstruction quality and speed.

5. Conclusion

We introduced a novel pipeline that enables efficient computation of extrinsic camera poses and scene structure on high spatiotemporal resolution, densely sampled video sequences. One of the key insights in this work is that the coherence of such data enables the use of modified tracking, subsampling, and global optimization schemes, which in combination allow for considerably faster and more robust computation, similar to observations made in previous works [15,33] in the context of 3D reconstruction. In particular we found that common choices in SfM such as n-point algorithms for initialization are problematic in this context and can be entirely replaced by BA-based approaches.

Given the constant increase of camera resolution and frame rate, and the advent of light field sensors by companies such as Lytro or Pelican Imaging, we believe that algorithms specifically designed for densely sampled input represent a great opportunity for future research in this area.

References

[1] http://www.disneyresearch.com/project/scalablesfm. 2, 7, 8

[2] Bundler Structure from Motion Toolkit. https://github.com/snavely/bundler_sfm. [Online; accessed 09-Nov-2014]. 2, 3, 8

[3] Open Source Computer Vision Library. http://opencv.org/. [Online; accessed 09-Nov-2014]. 3

[4] VisualSFM : A Visual Structure from Motion System. http://ccwu.me/vsfm/. [Online; accessed 09-Nov-2014]. 2, 8

[5] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski. Bundle adjustment in the large. In *ECCV*, 2010. 1, 2

[6] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *ICCV*, 2009. 1, 2

[7] G. Bradski. *Dr. Dobb's Journal of Software Tools*, 2000. 2

[8] A. Davis, M. Levoy, and F. Durand. Unstructured light fields. *Comp. Graph. Forum*, 31(2pt1):305–314, May 2012. 2

[9] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera SLAM. *IEEE TPAMI*, 29(6):1052–1067, 2007. 2

[10] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: large-scale direct monocular SLAM. In *ECCV*, 2014. 2, 8

[11] J. Frahm, P. F. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. Jen, E. Dunn, B. Clipp, and S. Lazebnik. Building rome on a cloudless day. In *ECCV*, 2010. 2

[12] A. Harltey and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2006. 1, 2, 4

[13] M. B. Hullin, J. Hanika, B. Ajdin, H.-P. Seidel, J. Kautz, and H. P. A. Lensch. Acquisition and analysis of bispectral bidirectional reflectance and reradiation distribution functions. *ACM Trans. Graph.*, 29(4):97:1–97:7, July 2010. 2

[14] N. Jiang, Z. Cui, and P. Tan. A global linear method for camera pose registration. In *ICCV*, pages 481–488, 2013. 2, 3, 5, 6

[15] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):73:1–73:12, 2013. 1, 2, 8

[16] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *ISMAR*, 2007. 2

[17] M. Li and A. I. Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *Int. J. Robotics Research*, 32(6):690–711, 2013. 2

[18] H. Lim, S. N. Sinha, M. F. Cohen, M. Uyttendaele, and H. J. Kim. Real-time monocular image-based 6-dof localization. *Int. J. Robotics Research*, 2014. 2

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 3

[20] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *CVPR*, 2007. 2

[21] R. A. Newcombe, S. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011. 2

[22] R. Ng. *Digital Light Field Photography*. PhD thesis, 2006. 2

[23] R. Parys and A. Schilling. Incremental large scale 3d reconstruction. *3DIMPVT*, 2012. 2

[24] M. Pollefeys, L. J. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *IJCV*, 59(3):207–232, 2004. 2

[25] N. Snavely, S. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *CVPR*, 2008. 2

[26] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *IJCV*, 80(2):189–210, 2008. 1, 2

[27] R. Szeliski and S. B. Kang. Recovering 3d shape and motion from image streams using nonlinear least squares. In *CVPR*, 1993. 1, 2, 3

[28] T. Roosendaal (Producer). Sintel. Blender Foundation, Durian Open Movie Project. http://www.sintel.org/, 2010. 7

[29] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. In *Foundations and Trends in Computer Graphics and Vision*, pages 177–280, 2007. 1

[30] O. Wang, C. Schroers, H. Zimmer, M. H. Gross, and A. Sorkine-Hornung. Videosnapping: interactive synchronization of multiple videos. *ACM Trans. Graph.*, 33(4):77, 2014. 2, 6

[31] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph.*, 24(3):765–776, 2005. 2

[32] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *ECCV*, 2014. 2

[33] F. Yu and D. Gallup. 3d reconstruction from accidental motion. In *CVPR*, 2014. 1, 2, 4, 8

[34] C. Zhang, J. Gao, O. Wang, P. Georgel, R. Yang, J. Davis, J. Frahm, and M. Pollefeys. Personal photograph enhancement using internet photo collections. *IEEE Trans. Vis. Comput. Graph.*, 20(2):262–275, 2014. 1

[35] D. Zou and P. Tan. Coslam: Collaborative visual slam in dynamic environments. *IEEE TPAMI*, 35(2):354–366, 2013. 2