Robust SSD tracking with incremental 3D structure estimation

Adam Rachmielowski, Dana Cobzaş, Martin Jägersand
CS, University of Alberta, Canada

Abstract

While the geometric aspects of structure and motion estimation from uncalibrated images are well understood, and it has great promise in applications, it has not seen widespread use. In this paper we combine SSD tracking with incremental structure computation into a system computing both motion and structure on-line from video. We show how in combination the structure estimation and tracking benefit each other, resulting in both better structure and more robust tracking. Particularly, through the 3D structure, our method can manage visibility constraints, add new image patches to track as they come into view and remove ones that are occluded or fail. This allows tracking over larger pose variations than possible with conventional SSD tracking (e.g. going around an object or scene where new parts come into view.) Experiments demonstrate tracking and capture of a scene from a camera trajectory covering different sides without mutual visibility.

Keywords: SSD tracking, structure-from-motion (SFM), incremental estimation.

1 Introduction

Tracking and structure-from-motion (SFM) can reciprocally benefit from integration. First, tracking provides good corresponding points for SFM thus enabling autonomous real-time capturing of 3D. Second, the computed 3D structure can improve tracking stability and robustness. In this work we focus on integrating tracking with an incremental structure computation and we study how tracking stability can be improved in order to provide reliable feature correspondences in SFM. A third benefit is that the incrementally acquired structure allows the tracking of motions larger than the visible field of the camera at any one time. Our system works with uncalibrated monocular video that is known to have several advantages over the classical calibrated approach (e.g. no need to calibrate camera, robustness to variation in internal parameters during sequence).

In the past two decades the fundamental study of camera imaging in a framework of projective geometry has produced new methods and algorithms for extracting geometric structure from uncalibrated images, the well known problem of structure-from-motion (SFM) [14]. Despite the theoretical progress, there are yet few practically useful systems and hence most of the real users of 3D models such as designers and architects are still using traditional technologies (manual 3D modeling, laser scanning). This is mainly due to the difficulty of getting good automatic image correspondences, required by the SFM methods, and thus demanding significant manual intervention to add and adjust correspondences.

Traditionally feature correspondences are obtained in a two step process: feature detection followed by a robust correlation-based matching [20, 5]. When working with video the correspondence problem turns into visual tracking. Existing tracking methods take advantage of the temporal smoothness of the images and compute image points or 3D camera pose as an incremental state update. While traditional feature-based tracking requires a-priori 3D and feature models that are registered with the current image measurements (e.g. [17]), in registration-based (SSD) tracking the pose computation is based on directly aligning a (time $t = 0$) reference intensity patch with the current image [18, 11, 3, 7], thus not relying on any predefined features or special markers.

In this paper, we propose a method that aims toward automating the 3D structure and motion recovery in video-based capture systems by integrating SSD tracking and structure computation. The system tracks features and computes the structure in an incremental fashion. We take a different approach than previous systems that attempt to automate the feature detection and correspondence [5, 20] and use tracking as a way of obtaining correspondences. We concentrate on developing methods to stabilize the tracking and make use of the structure that is incrementally computed along with the tracking. Thus our method is different than Cornelis et al.'s system [8] which also uses

	correspondences	*motion*	*structure*	*performance*	applications
on-line	SSD tracking	continuous increm : small overall : large	incr. bundl. adj	real-time	mobile robotics on-line interct. modeling
off-line	robust corresp guided corresp	arbitrary increm : large overall : small	RANSAC inters.	batch	augmented video off-line modeling

Table 1. A comparision of on-line and off-line sequential SFM approaches

tracking to get initial correspondences, but still uses traditional SFM methods (RANSAC) to eliminate bad correspondences. Instead we stabilize the tracking by re-acquiring the SSD template when needed and eliminate mistracked points based on monitoring slight increases in the re-projection error. As a consequence our system is designed for on-line use and is capable of covering a larger motion than what is visible in any one view. It is therefore similar in flavor to model-based mobile robotics systems [2] but without requiring an a-priori model, as the model is also acquired along with the tracking.

In summary the contributions of our paper are:

- We designed an on-line visual tracking and 3D structure estimation system,
- We developed on-line ways of incrementally updating and optimizing the 3D structure,
- We propose two ways of stabilizing the SSD tracking using knowledge of the computed structure and motion (by re-acquiring the template after a significant motion and by eliminating mistracked points that do not agree with the reconstructed structure),
- Experimentally we verify the validity of the proposed method by successfully tracking a large motion while incrementally extending the 3D structure.

2 Related work

Structure-from-motion techniques recover the 3D features and camera positions from corresponding 2D image features in uncalibrated images [14]. It is now well known that even if the cameras are not calibrated it is possible to reconstruct a projective structure of the scene. The structure can be updated to metric (Euclidean + scale) by making some assumptions about the camera internal parameters (e.g. zero skew, known principal point), by a process known as autocalibration (e.g. [9]). To minimize errors, the structure and cameras are optimized, using bundle adjustment, a step borrowed by the computer vision community from photogrammetry [23].

The complete SFM process results in quite accurate reconstructions recovered for a set of corresponding points. One of the main drawbacks of these methods is the difficulty of getting corresponding features. Most of the initial work was done with a limited set of images and manual correspondences. Some systems like Photomodeler [19] have been designed to provide user interfaces in assisting selecting the corresponding points. But the automation of this process is nevertheless required when working with large image sequences or video. One approach is to automatically select image features (e.g. [25]) and then match them across two, three pairs of images using robust scores (cross-correlation). Once a set of corresponding features in two or three images are available, the fundamental matrix or trilinear tensor, respectively, can be computed and further used to refine correspondences (e.g. [20, 5, 10]). While this method provides good results and is successfully used even in commercial applications such as Boujou [1] or Realviz [21], it is still designed for off-line processing and does not take advantage of the motion continuity present in a video sequence.

In this paper we explore the use of tracking to provide correspondences. In registration based tracking the pose computation is based on directly aligning a reference intensity patch with the current image to match each pixel intensity as closely as possible. Often a sum-of-squared differences (e.g. L_2 norm) error is minimized, giving the technique its popular name SSD tracking [18]. The method does not require a 3D structure, as it is purely image-based and efficient real-time algorithms have been developed [11, 3] making it a good candidate for on-line SFM systems. We propose ways of improving the robustness of tracking by both using constraints from the partial structure and acquiring a better template when needed thus focusing on developing an on-line system and eliminating the need to do RANSAC at every step. Table 1 shows the main differences between off-line and on-line SFM systems showing both the characteristics on the input

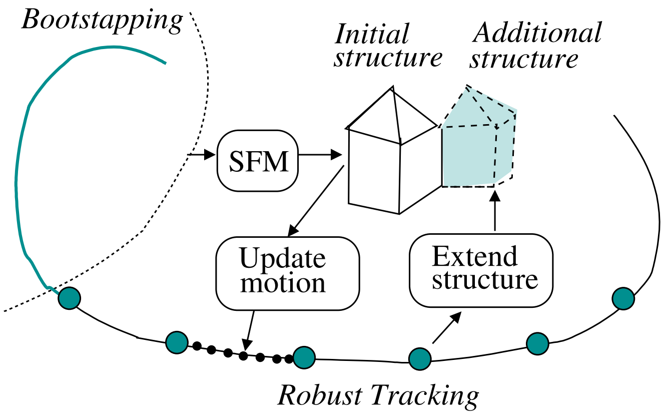


Figure 1. System overview

images, type of scene/motion and the target applications.

When working with sequences of images, the structure can also be sequentially updated [20, 5] (as opposed to classical batch updates [14]). But the final bundle adjustment is still performed at the end, thus there is no guarantee that the intermediate structure or motion is accurate. One way to deal with noise and errors is to model structure and motion update as a probabilistic process using a Kalman Filter (or EKF). Beardsley et al. [4] propose the use of an EKF for updating a quasi-Euclidean or affine structure. We are proposing a different approach to solve the error accumulation problem by performing one (or very few) bundle adjustment steps when incrementally computing the structure. Experiments show that this significantly reduces the reprojection error and it is less time consuming than a full bundle adjustment.

3 Background and system overview

3.1 SSD tracking

The goal of the SSD tracking algorithm, as originally formulated by Lucas-Kanade [18], is to find an image warp $W(\mathbf{x};\mu)$ that aligns a 2D template region $T(\mathbf{x})$ with the current image region $I(\mathbf{x})$. Under the image constancy assumption, the problem is formulated as finding the warp parameters μ that minimize the error between the template and the image warped into the space of the template. The problem is solved iteratively by finding an incremental update $\Delta\mu$ for the warp parameters from frame I_{t-1} to I_t that updates current warp.

$$\sum_x [T(\mathbf{x}) - I_t(W(\mathbf{x};\mu_{t-1} \circ \Delta\mu))]^2 \qquad (1)$$

where function composition ("$\circ$") is chosen for the warp update. Assuming small motion, the problem is linearized and the warp update is found as a least square solution of the following objective function:

$$\sum_x \left(T(\mathbf{x}) - I_t(W(\mathbf{x};\mu_{t-1})) - \nabla I_t \frac{\partial W}{\partial \mu}\Delta\mu\right)^2 \qquad (2)$$

An efficient formulation as described by Baker et al [3] inverts the role of the template and image and computes the update in the space of the template.
Advantages: SSD tracking provides an elegant formulation for pose update and results in real-time (60 Hz) algorithms [12] that perform well for dense continuous sequences.
Disadvantages: The main weakness of the algorithm is that the linear approximation is valid only under small motion (convergence 3-5 pixels), and therefore tracking easily becomes unstable. Another problem with large motions is that, due to discretization errors, the template might need re-initialization.

3.2 Structure from motion

Structure-from-motion (SFM) involves estimating structure and motion from two or more views of a static scene given a set of corresponding image features. Assuming a projective camera model, and using homogeneous coordinates, 3D points $\mathbf{X}$ are projected into image points $\mathbf{x}$ through the projection equation:

$$\mathbf{x} = P\mathbf{X} \qquad (3)$$

where P is a 3×4 projection matrix (11 DOF) that accounts for both internal and external camera parameters.

From only image measurements $\mathbf{x}$ in at least two views, both camera motion P and structure points $\mathbf{X}$ can be recovered. For example, in the two view case, the image measurements $\mathbf{x}_1, \mathbf{x}_2$ are related by the 3×3 fundamental matrix F (rank deficient, 7 DOF). After recovering F, it can be decomposed into the two canonical projection matrices $P_1 = [I|\mathbf{0}]$ and $P_2 = [[\mathbf{e}_2]_\times F|\mathbf{e}_2]$, where $\mathbf{e}_2$ denotes the epipole in the second image. Knowing P_1, P_2 the structure points are recovered through intersection from the two projection equations (Eq. 3).

Often a final step, called bundle adjustment, optimizes both structure and motion based on image reprojection error of all points in the entire sequence. Formally bundle adjustment performs a non-linear minimization of the following objective function:

$$r = \min_{P_t \mathbf{X}_i} \sum_t \sum_i (\mathbf{x}_{ti} - P_t\mathbf{X}_i)^2 \qquad (4)$$

Advantages: The now well known SFM formulation recovers both the structure and the unknown camera motion without any need for calibration.
Disadvantages: The method relies on existing corresponding image points, but the correspondence problem is still one of the most difficult vision problems. Robust estimation (using RANSAC) can be used to eliminate bad matches but it cannot be performed in real time. Similarly bundle adjustment is designed as a batch process being quite costly.

3.3 Proposed method

We are proposing here the integration of SSD tracking and SFM for an on-line acquisition system. The main challenges of the integration are:

- SFM usually assumes significant motion (e.g. for triangulation) while tracking works with small motions. Therefore, we need to ensure good tracking for significant motion and update the structure only when enough motion has been performed.
- A full bundle adjustment cannot be used in real time and therefore we designed an incremental bundle adjustment.
- RANSAC cannot be performed at every frame and therefore ways to stabilize tracking need to be found. We take advantage of the current structure knowledge to check tracking accuracy.

As depicted in Fig. 1, our system has three main components:

- In the *bootstrapping phase* (approx. 100 frames) trackers are initialized then used use to acquire an initial structure. The camera motion is relatively limited during this short time and a consistent structure is kept in view which ensures SSD tracking convergence.
- During the *robust tracking* and camera estimation phase, camera position is computed by resectioning the 2D-3D correspondences. Tracking performance is monitored in terms of reprojection error and trackers that deviate beyond a threshold are removed.
- After a sufficient amount of camera motion (approx. 15 deg. rotation with respect to major structure) the *structure is refined and extended.* To ensure that the structure is correct (as the following motion rely on a good structure) we do a bundle adjustment step and robust intersection. The motion is assumed somewhat smooth, so no large errors are introduced and a limited bundle adjustment is enough. Feature correspondences without associated 3D points are robustly matched and, if found consistent with the existing structure, are added by triangulation.

4 System Details

4.1 Acquiring initial structure

A number of choices exist for the warp function used in (Eq. 1). Six parameter affine or eight parameter homography warps provide good tracking of large planar regions, even over extended change in view, but converge poorly for small templates and are difficult to initialize automatically. Simpler warps, such as 2 parameter translation or 4 parameter translation + rotation + scale are less resilient to large changes in view, but are more stable for small templates, can be updated more quickly, and can effectively be initialized at automatically detected features. Although trackers are initialized manually in the first frame to encourage a good starting structure, additional trackers are automatically added later, so we use the efficient and stable translational warp with a template size of 11×11.

Mistracking features not only leads to outliers which later need to be removed, it also results in unnecessary computation. We use the SSD residual (Eq. 1) directly to monitor trackers in all phases. We determine a threshold and when a tracker's residual exceeds the threshold we remove it. As a safety margin we also remove the associated feature position in the previous few frames as it may affect future batch computations. The residual depends on the specific template pixels so it is not suggested to use a global threshold. We compute the threshold as follows: during tracker initialization, when the template is acquired, the initial warp parameters are perturbed. SSD residuals are computed between the template and images warped by the perturbed parameters. The maximum SSD residual is chosen as the threshold. In our implementation, the translational parameters are perturbed ± 2, corresponding to a 2 pixel motion in each direction.

Tracking continues until enough correspondence data has been collected to compute an initial structure ($\sim$100 frames). For a good initial structure, scene features need to be selected in a general configuration. Furthermore, camera motion should span the maximum possible change in position with respect to the scene while keeping all tracked features visible. We found that a spiral motion with the centre of rotation near the tracked features was effective. From the correspondences $\mathbf{x}_t$, we estimate fundamental matrices F_t between a reference view and a sparse set of other

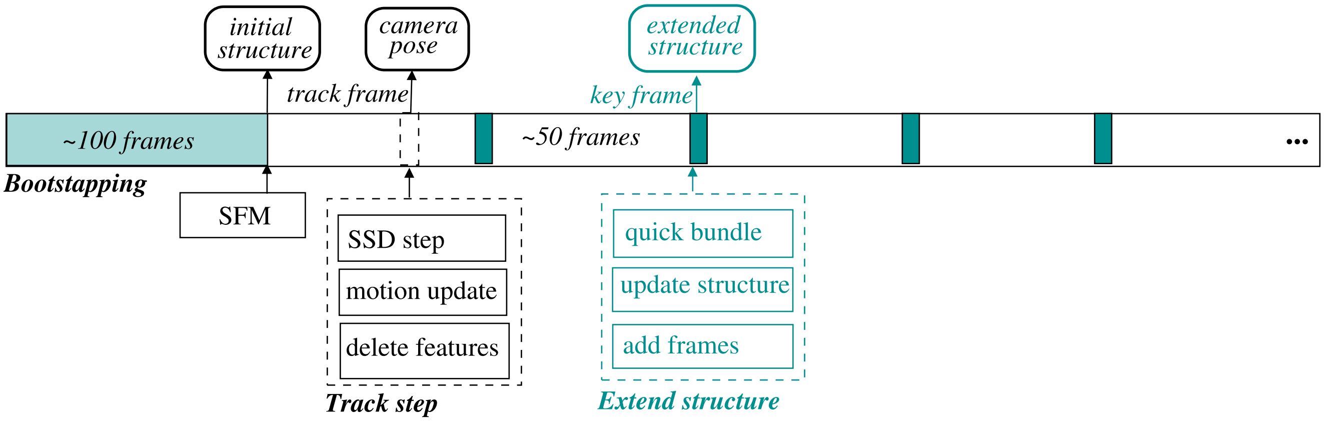


Figure 2. The three main components of the system (bootstrapping, the regular track step and the structure extension step) showed in time

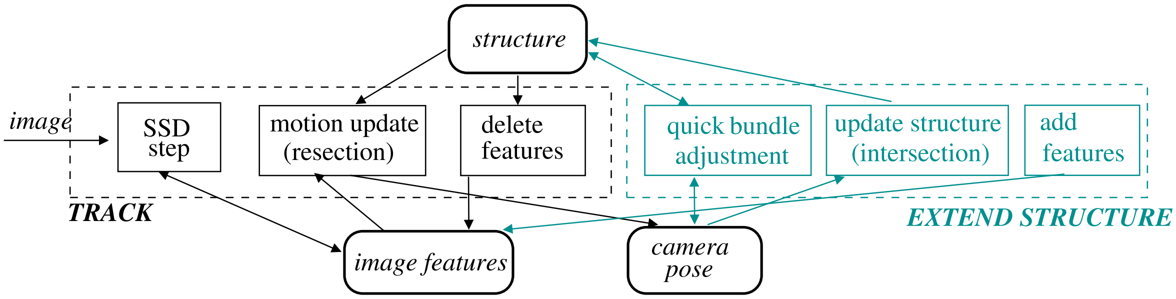


Figure 3. Processing of one image: track step and the optional structure extension step (only every approx. 50 frames).

views. From these F_t matrices, we extract projection matrices P_t in the reference frame.

While monitoring SSD error reduces the number of incorrect correspondences, in practice it is possible for some to remain even after several frames. Incorrect correspondences result in an incorrect estimate of F_t and P_t. To avoid this, we use RANSAC in estimating the fundamental matrix between any two views (only in this initial phase). To further minimize computational cost, we compute initial structure with a subset of the available views.

From the estimated cameras, P_t, and feature correspondences, $\mathbf{x}_t$, we triangulate 3D feature points, $\mathbf{X}$. These points define the initial structure of the scene, satisfy the projection equation (Eq. 3) and are in the same projective space as the camera matrices. It is possible to upgrade the structure to a Euclidean reconstruction, where scale is the only remaining ambiguity. However, we avoid the additional cost of calibrating the structure by using the projective reconstruction. (Upgrading to a Euclidean structure can be done as a final offline step if desired.) To improve the initial structure we apply bundle adjustment, which is otherwise typically the last step in SFM computation. Once the initial structure is acquired the camera motion can proceed along an arbitrary trajectory.

4.2 Robust tracking and camera motion estimation

At the beginning of each tracking cycle, we start with a set of trackers, their associated image features, the estimated structure and estimated camera pose for the previous frames. We initialize new trackers at all existing feature point positions, which have shown to track well. To prevent an accumulation of several trackers at each position we flag those features to not generate more trackers. Next, we detect new features. Harris corners [13] provide an efficient way to get features which can be tracked well, however selecting corners with the highest response in the entire image may result in densely clustered trackers in regions of high contrast. To ensure a more even distribution of new trackers, the image is tiled (3 by 4) and a maximum of 5 features is selected in each tile.

In each tracking step we take the current image, I_t, and perform an SSD step, a motion update step, and a step to remove trackers. In the SSD step, we solve (Eq. 2) for each tracker to update warp parameters and determine feature positions, $\mathbf{x}_t$, in I_t. The described SSD error threshold is used to remove potential mistracked features. Next, the current camera pose is estimated in a motion update step. We resection $\mathbf{x}_t$, $\mathbf{X}$ to estimate P_t satisfying (Eq. 3). The camera estimate is biased by mistracking but is determined mostly by the well tracked features. For each feature, $\mathbf{x}_{ti}$, reprojection error is computed:

$$\|\mathbf{x}_{ti} - P_t\mathbf{X}_i\| \tag{5}$$

and if the error exceeds a defined threshold (3 pixels in our experiments) the feature and associated tracker is removed. Under the assumption that there are few mistracked features, and mistracking is on the order of a few pixels, this formulation works well as an alternative to robustly estimating P_t. When using SSD tracking with video this assumption holds.

Robust tracking and camera motion estimation continues long enough to have a good baseline for triangulating new points (approx. 50 frames).

4.3 Refining and extending structure

From feature correspondences, $\mathbf{x}_t$, and camera matrices P_t in two or more views, 3D position of the features, $\mathbf{X}$, is triangulated. Good triangulation depends on having accurately estimated cameras and a sufficient baseline between the cameras. To improve our structure and camera motion estimates and to remove error contributed by trackers that have been removed, we perform a quick incremental bundle adjustment.

In iteratively minimizing (Eq. 4) in a bundle adjustment, each of the n camera matrices has 11 degrees of freedom and each of the m points has 3 degrees of freedom. The total number of parameters in the minimization is $11n + 3m$, and one iteration of the algorithm is cubic in this number. Efficient algorithms exist that take advantage of the lack of interaction between most parameters [16]. Nevertheless, for a real-time implementation it is important to limit the size of the problem.

Performing bundle adjustment with only a subset of cameras effectively reduces the number of parameters in the problem, but the resulting structure is biased towards the selected cameras. Since camera motion is continuous, eliminating nearby views is a good choice. We use every sixth new view and an even sparser set of older views when refining and extending structure. We perform one or two iterations of bundle adjustment on this set of views to achieve a balance between computational cost and accuracy. This typically reduces reprojection error to below 1 pixel (Fig. 4).

In addition to the good existing structure and camera motion estimates implied by a low reprojetion error, it is important to ensure that candidate points are consistent with the existing structure before adding them. To achieve this consistency we robustly compute F for the camera positions in the last cycle. From the set of all new features and any old features appearing in the recent views we find the set of inliers consistent with the existing structure. Each of these inliers is triangulated to get a new $\mathbf{X}$.

The two cameras with the widest baseline are selected. From these the point is triangulated using Hartley and Sturm's optimal method [15]. Since the 2D correspondence is not perfect, the back projected rays may not intersect. Their non-iterative method finds the closest rays that do intersect. For the correspondence $\mathbf{x}_1, \mathbf{x}_2$ and cameras P_1, P_2, it computes the new correspondence $\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2$, closest to $\mathbf{x}_1, \mathbf{x}_2$ while obeying the epipolar constraint implied by P_1, P_2. $\mathbf{X}$ is triangulated from $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$, P_1, P_2.

Once points have been added the system returns to the robust tracking and camera estimation phase.

5 Experimental Results

We show the results of the system on a 580 frame 320×240 video sequence. 40 features are initially selected by hand. They are tracked for 150 frames, while the camera is moved in a small spiral motion. Initial structure is acquired from 10 of the first 150 frames. Additional trackers are automatically initialized and tracking continues. Every 54 frames the structure is refined and extended using every sixth view (9 frames). An average of 45 additional trackers are added every cycle and an average of 25 trackers are used for camera estimation each frame.

The reprojection error computed during each tracking step is shown in Fig. 4. Since trackers may not account for the true image variation of the scene features they represent, over time they can slip from their intended position leading to an accumulation of reprojection error. This is visible in the increasing portions of Fig. 4. The vertical bars indicate the structure extension steps. A drop in reprojection error follows the bundle adjustment in these steps. Peaks in mid cycle

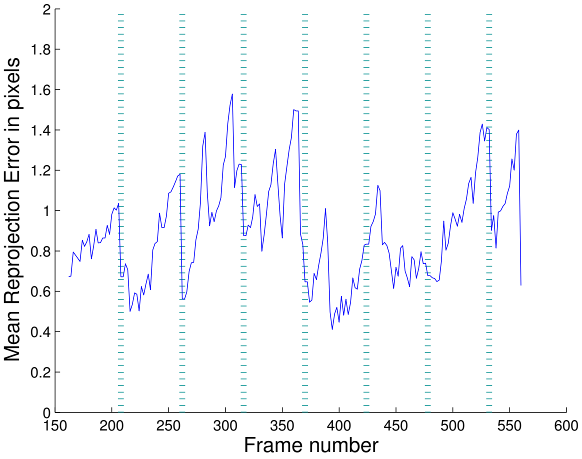


Figure 4. Mean reprojection error increases over time as tracking error accumulates. Vertical bars indicate structure extension and refinement.

correspond to trackers mistracking and increasing the reprojection error; then being removed and restoring the error to its previous value.

Images with overlayed projective structure are shown in the top row of Fig. 5. The bottom row shows a euclidean structure and camera motion. From left to right we show the initial structure and camera positions and the progress as camera motion is estimated and structure is extended. The far right column shows the resulting structure and motion after a final bundle adjustment, at which point the mean global reprojection error for all features over the entire sequence is 0.6 pixels.

6 Conclusion

We have presented an on-line system that integrates SSD tracking with uncalibrated structure computation. Two main benefits of the integration are: (1) Tracking automatically provides corresponding points used in incrementally computing the structure while partial structure provides constraints on the tracked points and therefore improves tracking accuracy by integrating large numbers of tracked image patches, and robustness by detecting image motion inconsistent with the 3D motion. (2) The structure is extended with new views of the scene or object when these appear. This allows tracking without requiring all features to be visible, and hence enables tracking through much larger motions than previously possible using only basic SSD tracking. Finally we show that both tracking and structure can be computed with very small image residuals even over large motions (A few pixels before bundle adjustment, and sub-pixel precision after bundle adjustment.)

7 Future work

In the future we plan to further stabilize the tracking by imposing strong structure constraints (either directly in the SSD tracking [7] or using a Kalman filter [6, 4]). One approach that might stabilize the tracking is to keep multiple templates (or an appearance model) for a tracker and index these by view direction. This would also allow integration (merging) of points that represent the same physical feature.

Regarding structure computation, a possible improvement would be to use model selection for choosing the frames on which bundle adjustment and structure update is performed. Also, an automatic system should detect degenerate configurations in either structure or motion [24]. Finally, combining different features types (lines, planes) could improve the quality of resulting models and provide further ways to integrate tracking with structure estimation.

Acknowledgments: Funding for this project was provided by PIMS, NSERC, IRIS, ASRA and CFI.

References

[1] Boujou 2d3. http://www.2d3.com/jsp/index.jsp.

[2] N. Ayache and O. D. Faugeras. Maintaining representation of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6):804–819, 1989.

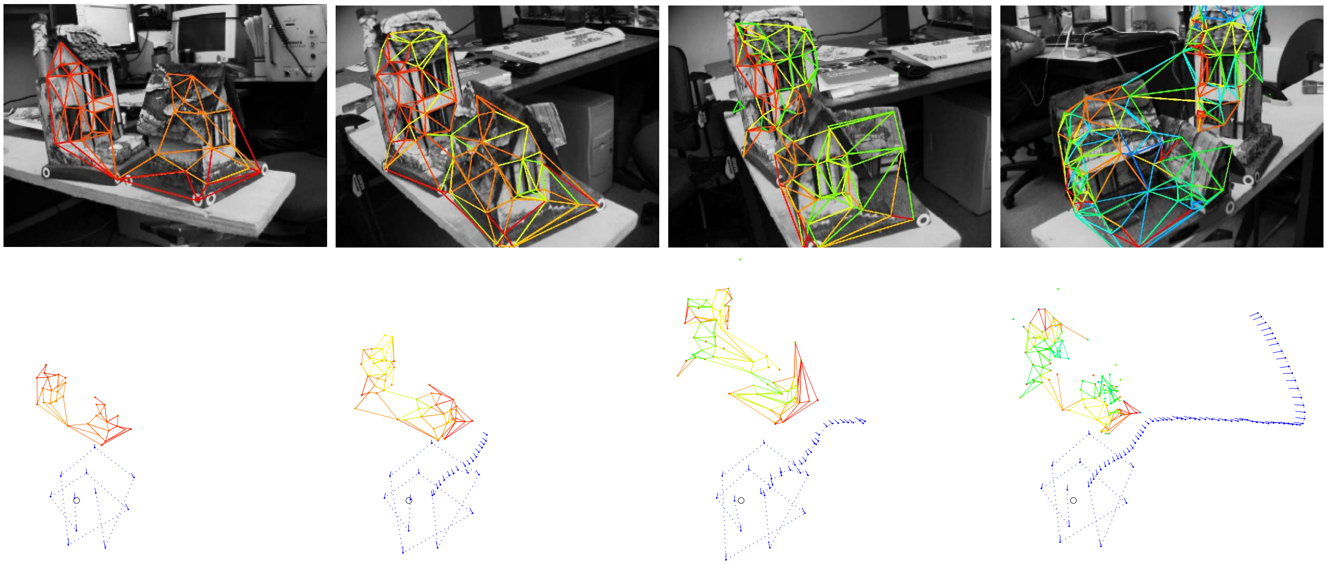
[3] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(3):221–255, 2004.

[4] P. A. Beardsley, A. Zisserman, and D. W. Murray. Sequential updating of projective and affine structure from motion. *IJCV*, 23(3):235–259, 1997.

[5] Paul A. Beardsley, Philip H. S. Torr, and Andrew Zisserman. 3d model acquisition from extended image sequences. In *ECCV '96: Vol. II*, pages 683–695, 1996.

[6] Alessandro Chiuso, Paolo Favaro, Hailin Jin, and Stefano Soatto. 3-d motion and structure from 2-d motion causally integrated over time: Implementation. In *ECCV '00: Part II*, pages 734–750, 2000.

[7] D. Cobzas and M. Jagersand. 3d ssd tracking from uncalibrated video. In *ECCV 2004 Workshop on Spatial Coherence for Visual Motion Analysis (SCVMA)*, 2004.



Frame: 150 Frame: 258 Frame: 366 Frame: 580

Figure 5. Top: Images with overlayed reprojected feature points. Bottom: Euclidean reconstruction for visualization. From left to right: After initial structure, after 2 structure extensions, after 3 structure extensions, after final bundle adjustment.

[8] Kurt Cornelis, Marc Pollefeys, and Luc Van Gool. Tracking based structure and motion recovery for augmented video productions. In *VRST '01: Proceedings of the ACM*, pages 17–24, 2001.

[9] O. Faugeras. Camera self-calibration: theory and experiments. In *ECCV*, pages 321–334, 1992.

[10] Projective Vision Toolkit G. Roth. http://www.cv.iit.nrc.ca/g̃erhard/pvt/index.html.

[11] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumin. *PAMI*, 20(10):1025–1039, 1998.

[12] G.D. Hager and K. Toyama. X vision: A portable substrate for real-time vision applications. *CVIU*, 69(1):23–37, 1998.

[13] C.G. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Fourth Alvey Vision Conf.*, pages 147–151, 1988.

[14] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[15] Richard I Hartley and Peter Sturm. Triangulation. *Comput Vision and Image Understanding*, 68(2):146–157, 1997.

[16] M.I.A. Lourakis and A.A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report 340, FORTH-ICS, 2004.

[17] D.G. Lowe. Fitting parameterized three-dimensional models to images. *PAMI*, 13(5):441–450, 1991.

[18] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Int. Joint Conf. on Artificial Intelligence*, 1981.

[19] Photomodeler. http://www.photomodeler.com/.

[20] M. Pollyfeys. *Tutorial on 3D Modeling from Images.* Lecture Nores, Dublin, Ireland (in conjunction with ECCV 2000), 2000.

[21] Realviz. http://www.realviz.com/.

[22] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings of the IEEE ICRA*, pages 2051–2058, 2001.

[23] C. Slama. *Manual of Photogrammetry*. American Society of Photogrammetry, Falls Church, VA, USA, 4th edition, 1980.

[24] Peter Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In *CVPR*, 1997.

[25] C. Tomasi and J. Shi. Good features to track. In *CVPR94*, pages 593–600, 1994.