Reconstruction from Uncalibrated Sequences with a Hierarchy of Trifocal Tensors

David Nistér

Visual Technology, Ericsson Research
Ericsson Radio Systems
SE-164 80 Stockholm SWEDEN
david.nister@era.ericsson.se

Abstract. This paper considers projective reconstruction with a hierarchical computational structure of trifocal tensors that integrates feature tracking and geometrical validation of the feature tracks. The algorithm was embedded into a system aimed at completely automatic Euclidean reconstruction from uncalibrated handheld amateur video sequences. The algorithm was tested as part of this system on a number of sequences grabbed directly from a low-end video camera without editing. The proposed approach can be considered a generalisation of a scheme of [Fitzgibbon and Zisserman, ECCV '98]. The proposed scheme tries to adapt itself to the motion and frame rate in the sequence by finding good triplets of views from which accurate and unique trifocal tensors can be calculated. This is in contrast to the assumption that three consecutive views in the video sequence are a good choice. Using trifocal tensors with a wider span suppresses error accumulation and makes the scheme less reliant on bundle adjustment. The proposed computational structure may also be used with fundamental matrices as the basic building block.

1 Introduction

Recovery of the shape of objects observed in several views is a branch of computer vision that has traditionally been called Structure from Motion (SfM). This is currently a very active research area [1-32]. Applications include synthesis of novel views, camera calibration, navigation, recognition, virtual reality, augmented reality and more. Recently, much interest has been devoted to approaches that do not assume any a priori knowledge of the camera motion nor calibration [1-4,6,9,13,20,31]. Thus, both the cameras and the structure are recovered. It is therefore relevant to speak of Structure and Motion (SaM). These approaches are very promising, especially as part of a potential system that extracts graphical models completely automatically from video sequences.

A very brief outline of one of many possible such systems is as follows. Features are extracted in all views independently [33,34]. Features are then matched by correlation into pairs and triplets from which multiple view entities such as fundamental matrices [6,11,29,32,35,36] or trifocal tensors [21,23,25,26] are

D. Vernon (Ed.): ECCV 2000, LNCS 1842, pp. 649-663, 2000.
© Springer-Verlag Berlin Heidelberg 2000

calculated. Camera matrices are then instantiated in a projective frame according to the calculated multiple view entities. The obtained pairs or triplets of camera matrices are transformed into a coherent projective frame [1] and optimised via bundle adjustment [9]. This yields a projective reconstruction that can be specialised to Euclidean by the use of autocalibration [10,16,20,30]. The views are now calibrated and a dense graphical model suitable for graphical rendering can be produced with any scheme developed for calibrated cameras. Examples of such schemes are space carving [15] and rectification [20] followed by conventional stereo algorithms.

This paper will concentrate on the stage where multiple view entities are calculated and registered into a coherent projective frame. Factorisation approaches are available that avoid the registration problems by obtaining all the camera matrices at once [19,27,28]. There are, however, compelling reasons for using an iterative approach and to build up the camera trajectory in steps. Direct estimation typically expects all features to be observed in all views, an assumption that in practice is heavily violated for long sequences. With an iterative approach, it is easier to deal with mismatches, another key to success in practice. Last but not least, an iterative approach makes it possible to use bundle adjustment in several steps and thereby build the reconstruction gradually with reprojection error as the criterion.

The contribution of this paper is a hierarchical computational structure that integrates feature tracking and geometrical validation of the feature tracks into an iterative process. The algorithm was tested as part of a system that automatically goes from frames of a video sequence to a sparse Euclidean reconstruction consisting of camera views, points and lines. The rest of the paper is organised as follows. Section 2 gives a motivation for the approach and the overall idea. The approach relies heavily on an algorithm to derive trifocal tensors and feature triples geometrically consistent with them. This estimation essentially follows [3,25], but is briefly sketched in Section 3 due to its importance. The proposed computational structure is given in more detail in Section 4. Section 5 describes how the trifocal tensors of the structure are carried on to a projective reconstruction. Results and conclusions are given in Sections 6 and 7.

2 Motivation and Method

The task of matching e.g. corners or lines between widely separated views by correlation is notoriously difficult. Rotation, scaling, differing background, lighting or other factors typically distort the texture of the region around a feature. Feature tracking makes it possible to establish matches with larger disparities. However, feature tracks are eventually lost and features move out of view in favour of new ones as the sequence goes on. Therefore, the number of feature correspondences between widely separated views is not always sufficient to allow a reliable estimation of a fundamental matrix or trifocal tensor.

On the other hand, a certain amount of parallax is necessary for a unique determination of the camera matrices (up to an overall homography) and the disparity between views is the clue to the actual depth of the features. A wider baseline is preferable, since it typically increases the amount of parallax and disparity.

To summarise the discussion, the loss of feature correspondences over time and the necessity of a reasonable baseline create a trade off. This means that in practice there is a sweet spot in terms of view separation, where calculation of the multiple view entities is best performed.

Many test sequences used in the vision community are well conditioned in the sense that consecutive views are reasonably separated. A typical sequence taken by a non-professional photographer using an ordinary handheld camera does not possess this quality. The number of views needed to build up a reasonable separation will typically vary over the sequence and depends on the frame rate and the speed and type of camera motion.

A too low frame rate in relation to the motion is of course difficult to remedy. It is therefore reasonable to use a high frame rate when acquiring the sequence and to develop algorithms that can adapt to the sequence and find the sweet spot in separation automatically. It is proposed here to achieve this by a two-step scheme. The first step is a preprocessor that can cope with a potentially excessive number of frames. The preprocessor is based on a rough global motion estimation between views and discards redundant views based on correlation after the motion estimation. The details are described in [18]. The second step, which is the topic of this paper, is a hierarchical computational structure of trifocal tensors. This structure integrates feature tracking and geometrical validation of feature tracks into an iterative process. It can be considered a generalisation of [1]. There, projective frames with triplets of camera views and corresponding features are hierarchically registered together into longer subsequences. The three views of one such triplet are always adjacent in the sequence, unless a preprocessor performs tracking.

In the scheme proposed here, the trifocal tensor algorithm is first used on triplets of consecutive views. This produces trifocal tensors together with a number of feature triplets consistent with every tensor. The consistent feature triplets from adjacent tensors are connected to form longer feature tracks. The longer tracks are then used together with new feature triplets, provided by raw matching, as input to new trifocal tensor calculations. The new tensors are between views twice as far apart in the sequence. This is repeated in an iterative fashion to produce a tree of trifocal tensors. From the tree of trifocal tensors, a number of tensors are now chosen that together span the sequence. The choice is based on a quality measure for trifocal tensors and associated feature triples. The goal of the quality measure is to indicate the sweet spot in view separation discussed earlier. The measure should therefore favour many consistent features and large amounts of parallax. The resulting tensors will be referred to as wide tensors and can stretch over anything from three up to hundreds of views, depending entirely on the sequence.

In this manner, frame instantiation and triangulation are postponed until disparity and parallax have been built up. The registration of intermediate views can then proceed to provide a denser sequence. In some cases it is desirable to include all views, while no intermediate views at all are required in others. The interpolative type of registration can sometimes be more accurate than its extrapolative counterpart. By taking long steps in the extrapolative registration, error accumulation is suppressed. This in turn makes the algorithm less reliant on the bundle adjustment process.

The success of the algorithm relies on two properties. The first one is that the algorithm for the trifocal tensor performs reasonably well at determining the consistent triples also when the baseline is insufficient to accurately determine the depth of the features or to give a unique solution for the camera matrices. The second one is that when presented with too widely separated views, the trifocal tensor algorithm yields a result with very few matches and as a consequence, unreliable tensors can be detected.

It is also in place to point out that an almost identical approach as the one described here may be taken with view pairs and fundamental matrices. The advantages of using three views as the basic building block are that lines can also be used and that the more complete geometric constraint removes almost all the false matches. These advantages come at a cost of speed.

3 Tensor Algorithm

The way to obtain new features essentially follows [2,3] and is sketched in Figure 1. Harris corners [34] are matched by correlation to give pairs upon which a fundamental matrix estimation is performed. The estimation is done by RANSAC [37], followed by optimisation of a robust support function.

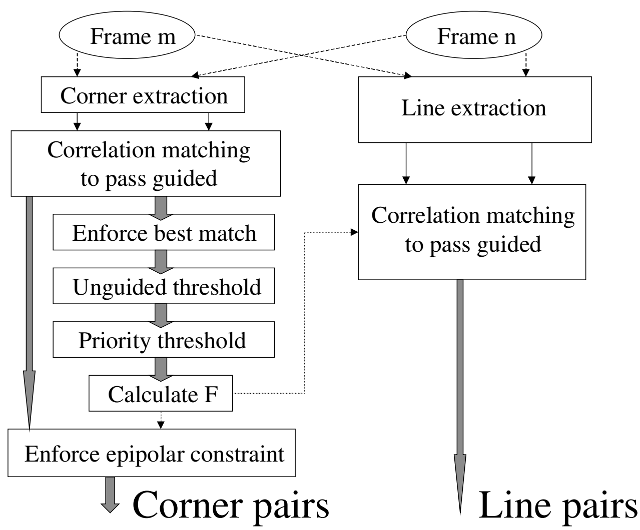

Figure 1. A sketch of how feature pairs are derived

The result is a fundamental matrix and a number of geometrically consistent corner pairs. Lines derived from Canny edges [33] are matched by correlation, guided by the extracted fundamental matrix, as described in [38]. The feature pairs are then connected into correspondences over three views. Given a set of feature triples, another RANSAC process [25] is conducted to estimate the trifocal tensor. Minimal sets consisting of six point triples are used to find tentative solutions for T using the method of [21]. The support for the tentative solutions is measured using both points and lines. Following [25], a robust support function based on maximum likelihood error of points and lines is used. The best tentative solution from the RANSAC process is then optimised. For best accuracy, the parameterization used in the optimisation should be consistent meaning that all possible choices of values for the parameters should yield a theoretically possible tensor.

4 Computational Structure

The proposed computational structure is applicable to any number of views, but as it simplifies this description, it is assumed that the sequence length is $N = 2^n + 1$ for some $n \in \mathbf{N}$. First, basic tensors $\{T(1,2,3), T(3,4,5), \ldots, T(N-2, N-1, N)\}$ are calculated for triplets of adjacent views. Then the next layer $\{T(1,3,5), T(5,7,9), \ldots, T(N-4, N-2, N)\}$ of tensors with double baseline (in an abstract sense) is found.

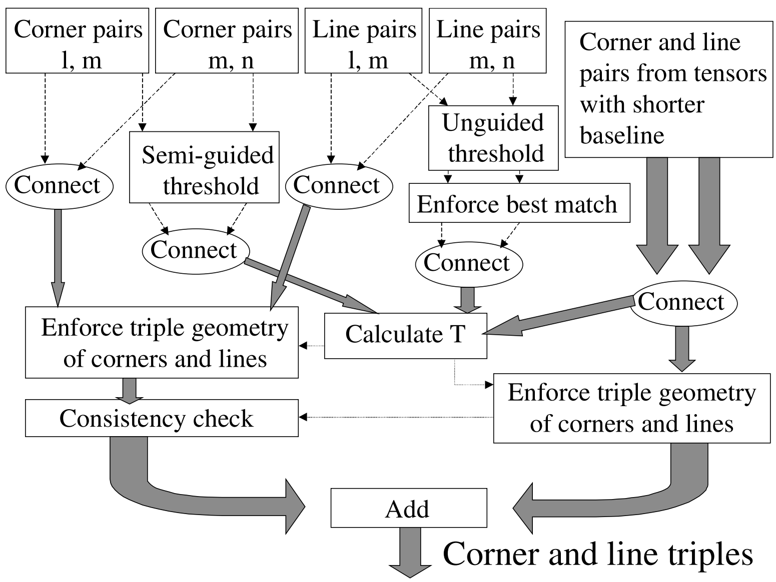

Figure 2. One iterative step of the trifocal tensor computation

The result of the first layer is passed on to this new layer. More specifically the calculation of the trifocal tensor $T(i, i+2^j, i+2^{j+1})$, where $i, j \in \mathbf{N}$ is fed with corner and line triples obtained from tensors $T(i, i+2^{j-1}, i+2^j)$ and $T(i+2^j, i+2^j+2^{j-1}, i+2^{j+1})$. The calculation of the narrower tensors provides a number of consistent corner and line triples. These triples are then connected at frame $i+2^j$ to provide longer triples by simply dropping the nodes at frames $i+2^{j-1}$ and $i+2^j+2^{j-1}$. The longer triples are fed into the wider tensor calculation together with new triples extracted by the basic algorithm sketched above in Section 3. One iterative step as just described is illustrated in Figure 2. The recursion then proceeds until a complete tree of tensors has been built up (see Figure 3). In this way the total number of tensor calculations becomes $2^n - 1 = N - 2$. Intertwining the connection of feature triples with the geometric pruning provided by the tensor calculation means that geometric constraints and tracking are integrated.

Although straightforward connection of feature triples was used in the current implementation, more sophisticated schemes could be used, for example by taking smoothness of motion into account.

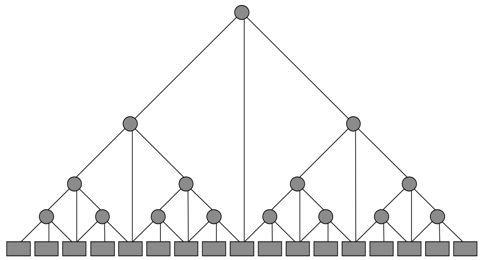

Figure 3. The hierarchical structure of trifocal tensors. The frames of the sequence are shown as rectangles at the bottom of the figure

For a long sequence, the number of consistent features has generally dropped under an unacceptable level before the top of the tree with a wide tensor spanning the whole sequence can be reached. As mentioned above, there is a sweet spot to be found in the width of the tensors. To determine the best width, a quality measure is defined for the tensors. A tensor is ruled unacceptable and the quality measure set to zero if the number of consistent features is below a threshold. A rather conservative threshold of 50 was used in the experiments. Otherwise the quality measure is defined as

$$Q = b \quad p\,. \tag{1}$$

Here b is an abstract baseline. It is defined as the distance between the first and the last frame of the tensor, but simply measured in frame numbers. The parameter is a constant determining how greedy the algorithm is for wider tensors. A reasonable choice was found to be $0.5 \qquad 1$. The parameter p indicates whether there is support in the data for a 3-dimensional geometric relationship rather than a 2-dimensional. With the use of this parameter, the algorithm tries to avoid degeneracies. The parameter p is related to the more sophisticated criterion described in [24]. If there is no translation between two camera views, or if all features seen by the views are located in a common plane, corresponding points x and x in the first and second view are related by a homography H as $x \quad Hx$ where H is represented by a 3x3 matrix defined up to scale. Likewise $l \quad lH^{\;1}$ for two corresponding lines l and l. Two homographies are fitted to the feature triples consistent with the trifocal tensor (again using RANSAC). The parameter p is then defined as the number of features triples that are consistent with the trifocal tensor but inconsistent with any of the homographies.

The consideration of tensors begins at the top of the tensor tree. If a tensor is ruled unacceptable or its children of narrower tensors both have a higher quality, the choice is recursively dropped to the next level of the tree. If the basic tensors at the bottom of the tree are also ruled unacceptable, the sequence is split into subsequences at the views in question. The result is now subsequences of wide tensors. A small example can be seen in Figure 4.

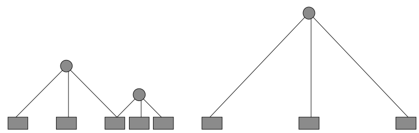

Figure 4. A small example of a choice of wide tensors

5 Building the Projective Reconstruction

The wide tensors are registered into a common projective frame following [1]. Before, or after this is done, the wide tensors can (optionally) be complemented with some of the intermediate views. In the current implementation of the algorithm, this is done first. A wide tensor typically has two child tensors in the tensor tree. Each of these children provides an extra view and also some additional features. The extra views are inserted recursively, deeper and deeper in the tensor tree. One way to accomplish the insertion is to register the child tensors into the frame of the parent. However, all features and views spanned by a wide tensor can in fact be instantiated directly in a common frame. This avoids inconsistency problems and is done in a similar fashion as for sequential approaches [2,9]. The differences are that the process is now interpolative rather than extrapolative plus that reliable correspondences are already extracted.

A tensor that provides the extra view to be filled in consists of the extra view plus two views that are already instantiated. The additional view is positioned between the old ones. Furthermore, all the new features that are provided have been found consistent with the trifocal tensor. Therefore all the new features have two observations in the already instantiated views. Thus they can be triangulated directly [12]. Once this is done, the new view can be determined from the features seen in it through another RANSAC process followed by optimisation.

It should be remarked that once trifocal tensors spanning the video sequence are known, the camera matrices are in theory determined without further consideration of the features. However, the consistent use of reprojection error has been found crucial in practice. Furthermore, error accumulation is suppressed by bundle adjustment after every merge.

In the process of intermediate view insertion, a quality assurance mechanism can be included that rejects the introduction of a new view and stops the recursion at the current depth if there are indications of failure. Depending on the application, it can also be desirable to limit the insertion to a certain number of steps. A wide tensor spanning many views typically indicates that there is little motion between the views of the original sequence. Thus, when it is desirable to homogenise the amount of motion between views, this can be accomplished by limiting the depth of insertion.

Finally, the wide tensors with associated features are registered into a common projective frame. Also here, it is beneficial to use some kind of heuristic as to whether a merge should be accepted or not. In case of failure, it is better to discard the merge and produce several independent reconstructions, or to prompt for a manual merge. To build reliable automatic reconstruction systems, work on quality monitoring of this kind is essential.

6 Results

Experiments were performed on approximately 50 sequences. The projective reconstructions are specialised to Euclidean with the use of autocalibration. A fixed focal length, fixed principal point and zero skew are forced upon the model. The result is then bundle adjusted in the Euclidean setting. Some results are shown in Table 1.

Table 1. Results from five sequences. The table is explained below

Sequence	Frames	Views	Points	Lines	P_error	L_error	Figure
Nissan Micra	1-28	17	679	93	0.73	0.007	-
	28-235	89	4931	170	1.11 (0.67)	0.039 (0.017)	-
	235-340	59	3022	226	0.79	0.024	6
Flower Pot	1-115	61	3445	23	0.68	0.029	8
	122-180	43	3853	1	0.65	0.000	-
	180-267	83	10068	0	0.67	-	9
Swedish Breakfast	1-64	41	3071	273	0.74	0.024	11
	64-249	125	8841	739	0.75	0.019	-
	249-297	29	749	240	0.68	0.025	-
Bikes	1-161	103	8055	369	0.73	0.018	13
David	1-19	11	688	38	0.52	0.020	15
Shoe & Co	19-39	21	942	55	0.67	0.032	17

All sequences in the table have the resolution 352 x 288 pixels. The column ‘Frames’ shows the frame span in the original sequence. Due to preprocessing all frames are not used. The number of views, points and lines in the reconstruction are displayed in the columns ‘Views’, ’Points’ and ‘Lines’. ‘P_error’ is the root mean square point reprojection error in number of pixels. ‘L_error’ is the root mean square line reprojection error. The line reprojection error is measured as the length of the vector $l \quad l$, where l and l are the observed and reprojected line, represented as homogenous line vectors normalised to hit the unit cube. The column ‘Figure’ indicates in which figure the reconstruction is displayed graphically. Observe that the reprojection error of the second sub-sequence of ‘Nissan Micra’ is higher than the other results. This is due to a failure of the autocalibration. The camera trajectory is torn into two separate parts. The reprojection error in the projective frame is therefore shown in parenthesis.

The sequence ‘David’ was taken by the person in the sequence, stretching out an arm with the camera and moving it in an arc. The sequence ‘Shoe & Co’ is a ‘home made’ turntable sequence. It was taken in front of a refrigerator with the camera in a kitchen cupboard. The turntable is a revolving chair that was turned by hand with the camera

Figure 5. Some frames from the sequence Nissan Micra

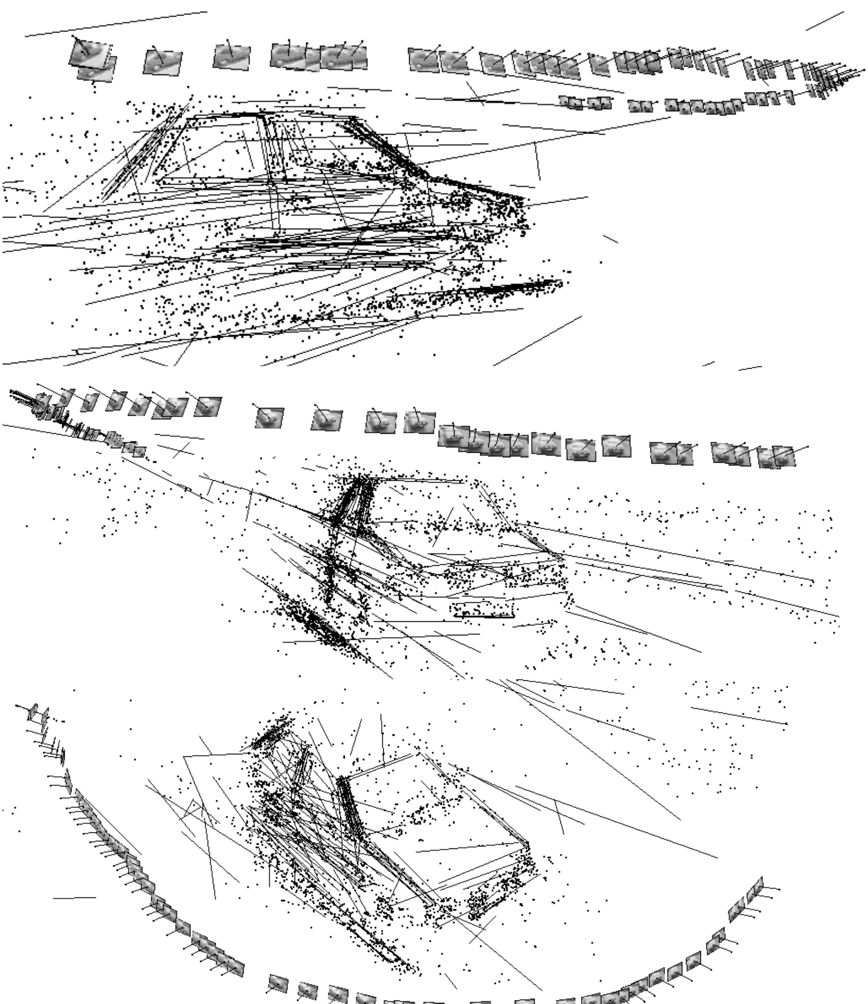

Figure 6. Three views of a reconstruction from Nissan Micra

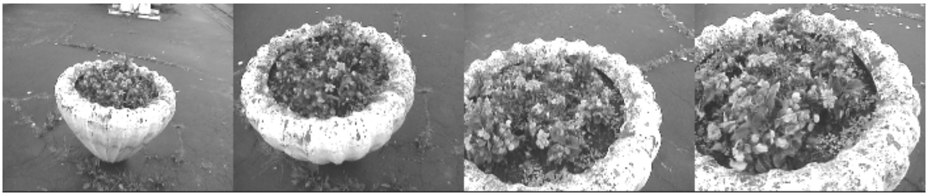

Figure 7. Some frames from the sequence Flower Pot

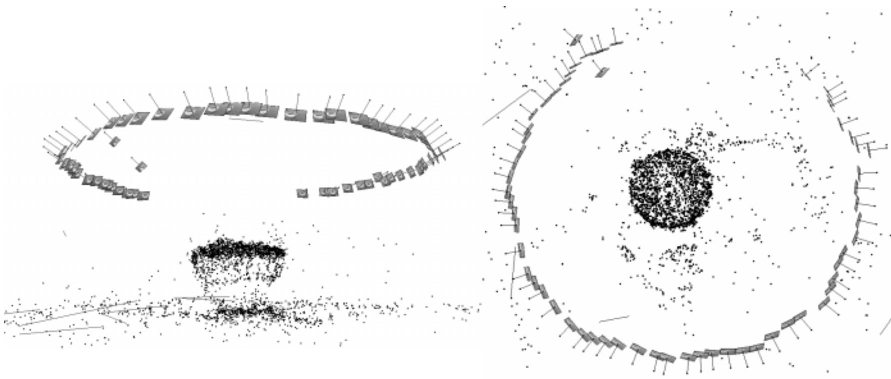

Figure 8. Two views of a reconstruction from the sequence Flower Pot

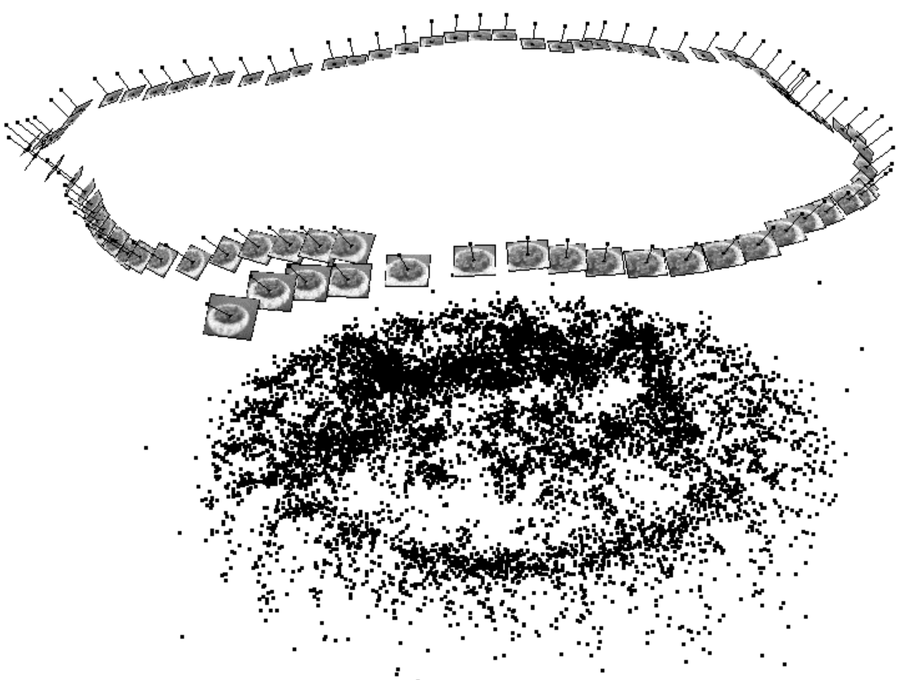

Figure 9. View of a reconstruction from the sequence Flower Pot

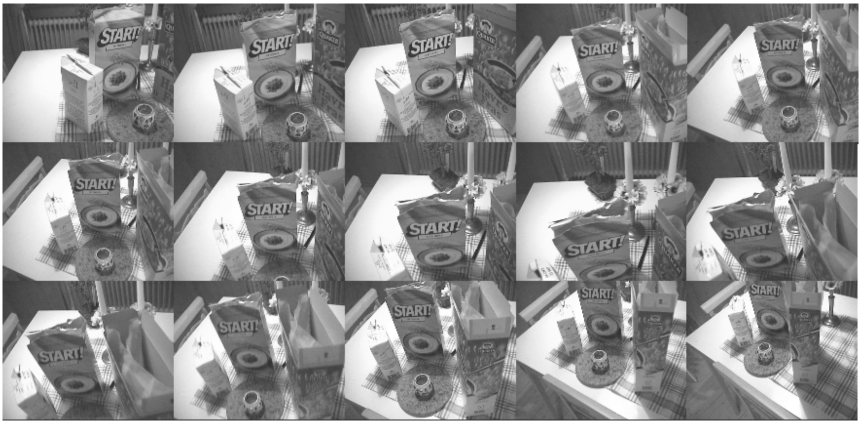

Figure 10. Some frames from the sequence Swedish Breakfast

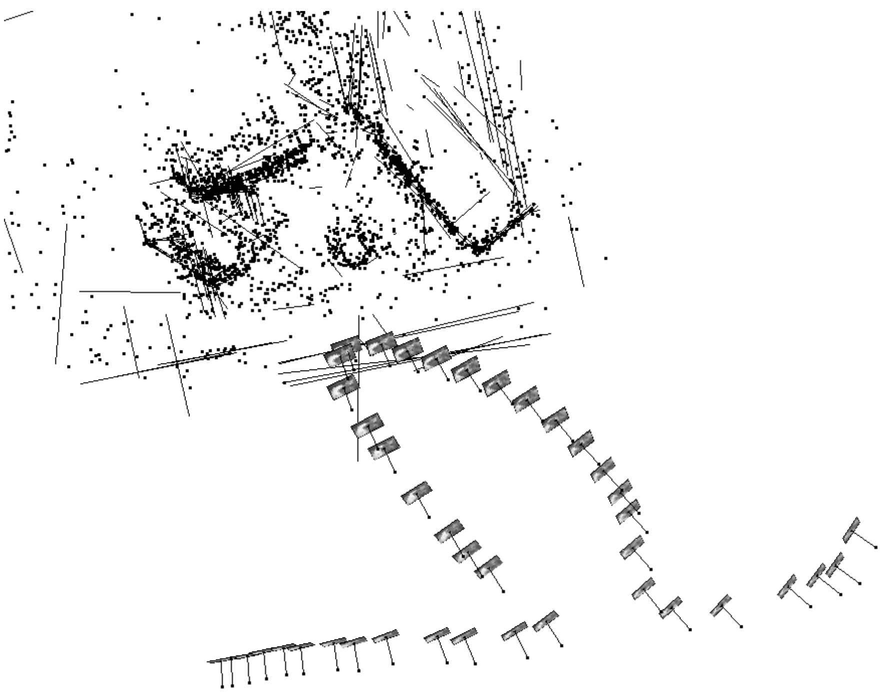

Figure 11. View of a reconstruction from Swedish Breakfast

running. The remaining sequences are handheld. Figures 5-17 display results graphically. In some of the figures it is difficult to interpret the structure. What can be seen in all of them however, is that the extracted camera trajectory is good. This is the most important result. The algorithm is mainly intended for intrinsic and extrinsic camera calibration. In a complete system, it should be used together with a dense reconstruction algorithm that uses the calibration. It is not clear though, that the dense

Figure 12. Some frames from the sequence Bikes

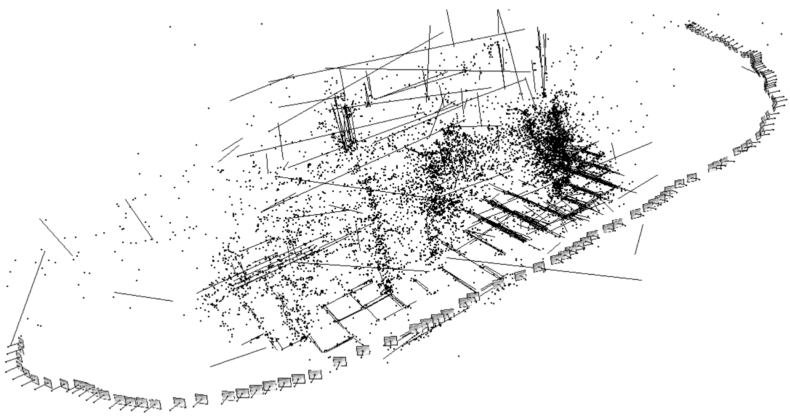

Figure 13. View of a reconstruction from the sequence Bikes

Figure 14. Some frames from the sequence David

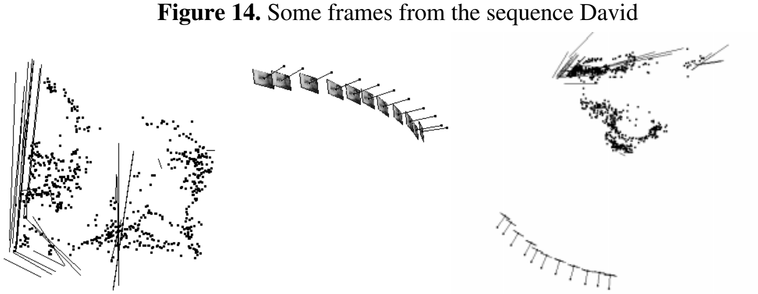

Figure 15. Two views of a reconstruction from the sequence David

Figure 16. Some frames from the sequence Shoe & Co

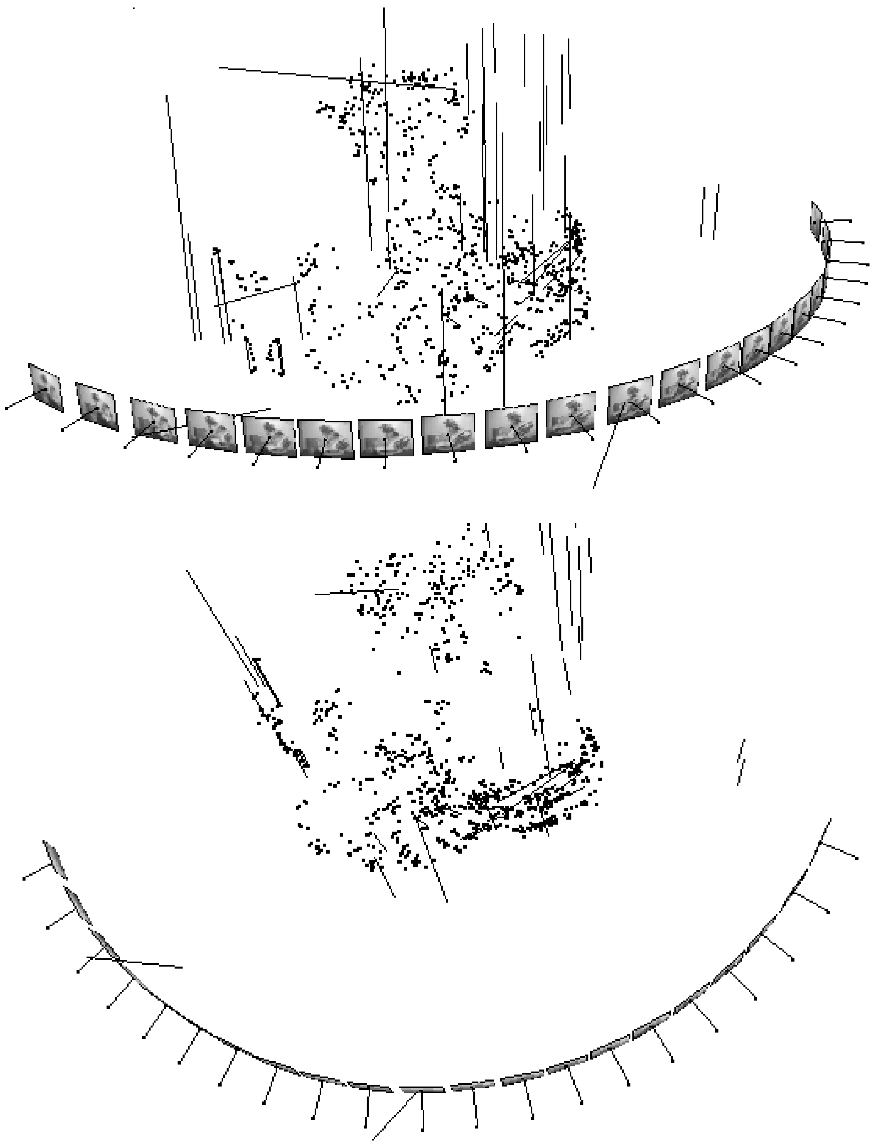

Figure 17. Two views of a reconstruction from the sequence Shoe & Co

reconstruction algorithm should necessarily use the structure. Dense reconstruction is part of the future work and out of the scope of this paper.

7 Conclusions

A hierarchical computational structure of trifocal tensors has been described. It is used to solve for structure and motion in uncalibrated video sequences acquired with a handheld amateur camera. In agreement with this purpose, the algorithm was tested mainly on video material with different amounts of motion per frame, frames out of focus and relatively low resolution. With the presented computational structure, the instantiation of camera matrices and feature triangulation are held until disparity and parallax have been built up. The structure also integrates feature tracking and geometrical constraints into an iterative process. Experimental results have been shown in terms of sparse Euclidean reconstructions consisting of views, points and lines. The presented results are taken from experiments on approximately 50 sequences. Future work includes more sophisticated ways to monitor the quality of the results, work on the autocalibration stage of the reconstruction and also the use of a method to derive a dense reconstruction.

References

[1] A. Fitzgibbon, A. Zisserman, Automatic camera recovery for closed or open image sequences, *Proc. ECCV 98*, pp. 311-326.

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

[3] P. Beardsley, P. Torr, A. Zisserman, 3D model acquisition from extended image sequences, *Proc. ECCV 96*, pp. 683-695.

[4] R. Cipolla, E. Boyer, 3D model acquisition from uncalibrated images, *Proc. IAPR Workshop on Machine Vision Applications*, Chiba Japan, pp. 559-568, Nov 1998.

[5] P. Debevec, C. Taylor, J. Malik, Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach, *SIGGRAPH 96*, pp. 11-20.

[6] O. Faugeras, What can be seen in three dimensions with an uncalibrated stereo rig?, *Proc. ECCV 92*, pp. 563-578.

[7] P. Fua, Reconstructing complex surfaces from multiple stereo views, *Proc. ICCV 95*, pp. 1078-1085.

[8] K. Hanna, N. Okamoto, Combining stereo and motion for direct estimation of scene structure, *Proc. ICCV 93*, pp. 357-365.

[9] R. Hartley, Euclidean reconstruction from uncalibrated views, *Applications of Invariance in Computer Vision*, LNCS 825, pp. 237-256, Springer-Verlag, 1994.

[10] R. Hartley, E. Hayman, L. de Agapito, I. Reid, Camera calibration and the search for infinity, *Proc. ICCV 99*, pp. 510-517.

[11] R. Hartley, Estimation of relative camera positions for uncalibrated cameras, *Proc. ECCV 92*, pp. 579-587.

[12] R. Hartley, P. Sturm, Triangulation, *American Image Understanding Workshop*, pp. 957-966, 1994.

[13] A. Heyden, K. Åström, Euclidean reconstruction from image sequences with varying and unknown focal length and principal point, *Proc. CVPR 97*, pp. 438-443.

[14] M. Irani, P. Anandan, M. Cohen, Direct recovery of planar-parallax from multiple frames, *Proc. Vision Algorithms Workshop (ICCV 99)*, Corfu, pp. 1-8, September 1999.
[15] K. Kutulakos, S. Seitz, A theory of shape by space carving, *Proc ICCV 99*, pp. 307-314.
[16] S. Maybank, O. Faugeras, A theory of self-calibration of a moving camera, International Journal of Computer Vision, 8(2), pp. 123-151, 1992.
[17] P. McLauchlan, D. Murray, A unifying framework for structure from motion recovery from image sequences, *Proc. ICCV 95*, pp. 314-320.
[18] D. Nistér, Frame decimation for structure and motion, *Submitted to SMILE 2000*.
[19] C. Poelman, T. Kanade, A paraperspective factorization method for shape and motion recovery, *Proc. ECCV 94*, pp. 97-108.
[20] M. Pollefeys, R. Koch, L. Van Gool, Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters, *IJCV*, 32(1), pp. 7-26, Aug, 1999.
[21] L. Quan, Invariants of 6 points from 3 uncalibrated images, *Proc. ECCV 94, LNCS 800/801*, Stockholm, pp. 459-469. Springer Verlag, 1994.
[22] L. Robert, O. Faugeras, Relative 3D positioning and 3D convex hull computation from a weakly calibrated stereo pair, *Proc. ICCV 93*, pp. 540-544.
[23] A. Shashua, Trilinear tensor: the fundamental construct of multiple-view geometry and its applications", *Proc. AFPAC 97*, Kiel Germany, pp. 190-206, Sep 8-9, 1997.
[24] P. Torr, An assessment of information criteria for motion model selection, *Proc. CVPR 97*, pp. 47-53.
[25] P. Torr, A. Zisserman, Robust parametrization and computation of the trifocal tensor, *Image and Vision Computing*, Vol. 15, p. 591-605, 1997.
[26] M. Spetsakis, J. Aloimonos, Structure from motion using line correspondences, *IJCV*, pp. 171-183, 1990.
[27] P. Sturm, W. Triggs, A factorization based algorithm for multi-image projective structure and motion, *Proc. ECCV 96*, pp. 709-720.
[28] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography: a factorization approach, *IJCV*, 9(2), pp. 137-154. November 1992.
[29] P. Torr, D. Murray, The development and comparison of robust methods for estimating the fundamental matrix, *IJCV*, pp. 1-33, 1997.
[30] B. Triggs, Autocalibration and the absolute quadric, *Proc. CVPR 97*, pp. 609-614.
[31] L. Van Gool, A. Zisserman, Automatic 3D model building from video sequences, *Proc. ECMAST 96*, pp. 563-582.
[32] Z. Zhang, Determining the epipolar geometry and its uncertainty: a review, *IJCV*, 27(2), pp. 161-195, 1997.
[33] J. Canny, A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, Nov 1986.
[34] C. Harris, M. Stephens, A combined corner and edge detector, *Proc. 4th Alvey Vision Conference*, pp. 147-151, 1988.
[35] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*, MIT Press, 1993.
[36] H. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, *Nature*, vol. 293, pp. 133-135, 1981.
[37] M. Fischler, R. Bolles, Random sample consencus: a paradigm for model fitting with application to image analysis and automated cartography, *Commun. Assoc. Comp. Mach.*, vol 24, pp. 381-395, 1981.
[38] C. Schmid and A. Zisserman, Automatic line matching across views, *Proc. CVPR 97*, pp. 666-671.