

A Modular Scheme for 2D/3D Conversion of TV Broadcast

Sebastian Knorr, Evren İmre[†], Burak Özkalaycı[†], A. Aydın Alatan[†], and Thomas Sikora

Communication Systems Group
Technische Universität Berlin
Einsteinufer 17, Berlin, Germany
E-mail: {knorr, sikora}@nue.tu-berlin.de

[†] EEE Department
Middle East Technical University
Balgat, 06531 Ankara, Turkey
E-mail: eimre@metu.edu.tr

Abstract

The 3D reconstruction from 2D broadcast video is a challenging problem with many potential applications, such as 3DTV, free-viewpoint video or augmented reality. In this paper, a modular system capable of efficiently reconstructing 3D scenes from broadcast video is proposed. The system consists of four constitutive modules: tracking and segmentation, self-calibration, sparse reconstruction and, finally, dense reconstruction. This paper also introduces some novel approaches for moving object segmentation and sparse and dense reconstruction problems. According to the simulations for both synthetic and real data, the system achieves a promising performance for typical TV content, indicating that it is a significant step towards the 3D reconstruction of scenes from broadcast video.

1. Introduction

3DTV technology is currently being investigated in many research labs worldwide [1]. In this context, conversion of existing 2D video material to 3D is of much interest. Many fundamental algorithms have been developed to reconstruct 3D scenes from an uncalibrated video sequence [2][3]. However, most of these approaches deal with the reconstruction of static scenes [2]. When the scene is dynamic, i.e., contains independently moving objects (IMO), they usually fail, since the triangulation techniques used for the reconstruction can only deal with one single relative motion.

In this paper, a complete system for the 3D reconstruction of a scene from broadcast video is proposed. The input to the system is an uncalibrated 2D video sequence captured from typical TV broadcast, and the output is a dense 3D reconstruction of the scene observed in the sequence. In addition to a complete 2D/3D conversion system, we are presenting two key innovations: a novel

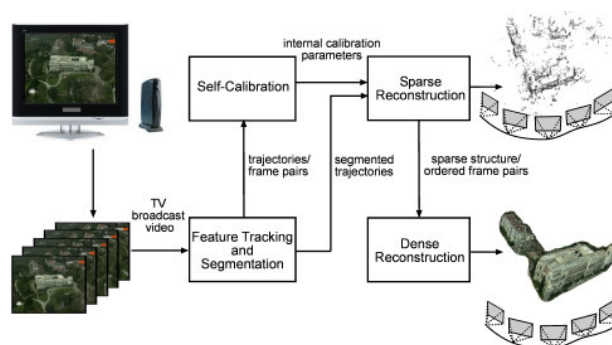


Figure 1.1: Block diagram of the proposed system

geometric segmentation approach for dynamic scenes and a prioritized sequential algorithm for sparse 3D reconstruction and camera path estimation. An overview of the complete 2D/3D conversion scheme is illustrated in Figure 1.1.

The proposed system is composed of four modules: Feature tracking and segmentation, self-calibration, camera pose estimation and sparse 3D reconstruction, and finally, dense reconstruction. Each module deals with a specific task and is the subject of a distinct field of research, thus, deserves an individual overview of the relevant literature. However, since the self-calibration is not the focus of this study, it is enough to point out that the implemented algorithm is based on the approach of Mendonca [4].

1.1. Feature Tracking and Segmentation

The typical features for 3D reconstruction problems are corners. As long as the input is a sequence of consecutive frames, which is guaranteed in broadcast video, the Kanade-Lucas-Tomasi tracker (KLT) successfully tracks features throughout the sequence [5].

In the case of a dynamic scene, an additional step is to segment the feature set into partitions conforming to the individual motions, to improve the quality and the reliability of the feature correspondence set for further processing stages. The solution approaches for the feature segmentation

problem can be classified into four categories. Optical flow-based methods assume that the scene is composed of planes at various depths, and utilize a simple clustering to achieve the desired segmentation [6]. Another set of solutions utilize the eigen decomposition of the *affinity matrix*, a structure which contains the similarity information among the features [7]. Geometric methods exploit the constraints imposed by the epipolar geometry and the rigid body motion assumption. The most common constraint is the fundamental matrix [8]. However, more general model selection-based methods are also available [9]. Finally, statistical methods also hold a niche in this field [10].

1.2. Sparse Reconstruction

In order to achieve any sparse reconstruction of the scene from video frames, the multi-frame structure from motion (MFSfM) problem should be solved. The basic solution approaches are the batch and the sequential methods. The best-known example of the former is the *factorization method* [3]. In the sequential methods, the problem is either cast into the framework of state estimation in dynamic systems [11], or the framework of inverse-MSE filtering to estimate an unknown constant vector (structure) [12].

While uncommon, there also exist techniques to solve the MFSfM problem simultaneously for all bodies involved, employing the multi-body extension of the factorization method [13], or particle filter algorithms [10].

1.3. Dense Reconstruction

The dense reconstruction problem is formulated as the depth map estimation of a single reference view using multiple views. In the literature, the proposed solution methods cover a wide range of techniques. Although their focus is primarily on the dense disparity estimation for stereo vision, it is possible to consider the multiple-view case as a generalization of the two-view case. Nearly a complete taxonomy of the dense disparity estimation from stereo vision can be found in [14]. Regarding this taxonomy, the approaches on dense depth estimation can be broadly classified into two groups. The methods in the first group utilize local optimization techniques, such as window-based approaches [15], while the second group consists of the methods employing global optimization approaches. Global methods yield more reliable results, as they easily incorporate the regularization tools to achieve a better solution. The graph-cut [16], Markov random field (MRF) based [17], and partial differential equation (PDE) based methods [18] can be listed as typical examples of global approaches.

1.4. Outline of the paper

The organization of the paper is as follows: In the next section, the feature tracking and segmentation module is

discussed in detail. The sparse and dense reconstruction modules are explained in Sections 3 and 4, respectively. The simulation results for the whole system are presented in Section 5. Finally, Section 6 concludes this paper with a discussion and future work.

2. Feature Tracking and Segmentation

2.1. Feature Tracking

Since a large baseline length is usually preferred for both the segmentation and the reconstruction processes, a slightly modified version of the well-known pyramidal KLT tracker is used to track features throughout the video sequence. The first modification is the replacement of the lost features by adding new corners extracted from the current frame. Another improvement is related to the key-frame selection to handle the baseline problem for the segmentation and the reconstruction part. In the initial segmentation step, at least a single frame pair is needed that yields a reliable F-matrix estimate for the classification of the feature trajectories, depending on whether they belong to the background or to an independently moving object (IMO). Since the baseline between consecutive frames is small or the camera rotates about its center, a 2D motion model, \mathbf{H} (*homography*), can be used to transfer features from one frame to their corresponding positions in the second frame [19]. If the baseline increases during the tracking process and if the features belong to a 3D scene structure, the projection error increases as well, i.e., the 2D motion model must be upgraded to a 3D motion model, \mathbf{F} (epipolar geometry). Hence, the current frame can be selected as a key-frame for a reliable F-matrix estimation.

The Geometric Robust Information Criterion (GRIC) [9] is a robust model selection criterion to extract key-frames. An exhaustive study of this criterion is given in [20].

2.2. Segmentation

Once the feature trajectories are constructed and the key-frames are selected, trajectory segmentation is handled by geometric means. For each independent motion in the sequence, there exists a corresponding F-matrix, \mathbf{F}_i , which satisfies the epipolar constraint

$$\mathbf{x}_1^T \mathbf{F}_i \mathbf{x}_2 = 0, \quad (2.1)$$

where \mathbf{x}_1 and \mathbf{x}_2 are corresponding points in two views. A RANSAC (RANdom Sample Consensus)-based F-matrix estimation algorithm [21] identifies the feature pairs belonging to the dominant motion and labels the rest of feature pairs as outliers. If the same procedure is repeated with the outliers (*re-RANSAC*), some of them should satisfy the epipolar constraint with respect to a new F-matrix, which

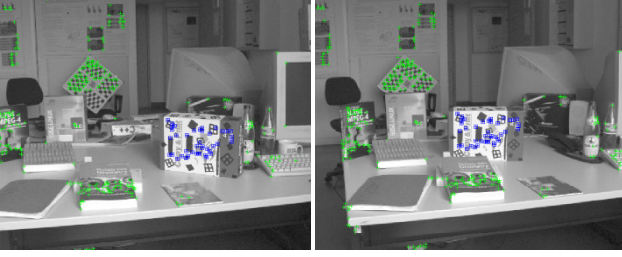


Figure 2.1: Trajectory segmentation of the “Desk”-sequence

should correspond to the motion of an IMO. This procedure is repeated until no more reliable F-matrices can be found, which means that all significant IMOs are detected. Hence, upon successive iteration of the procedure for all key-frames, the feature trajectories can be classified, either as background or IMOs.

For the cases, where one of the IMOs has a similar motion as the camera, some features may be classified as IMO features, even if they belong to the background or another IMO. This problem is handled by considering the distance of each of the segmented features to their centroid. A feature is rejected, if its distance is higher than a predefined threshold dependent on the standard deviation of the distances:

$$dist_i > D(x_i, c) + v \cdot \sigma, \quad (2.2)$$

where c is the centroid of the data set, v is a weighting factor and σ is the standard deviation.

The trajectories, which are labeled as outliers after RANSAC, re-RANSAC and the distance check are discarded from the trajectory set. Finally, guided matching is employed [19] along the epipolar lines in the key-frames to increase the number of IMO features. The segmentation algorithm can be summarized as follows:

Algorithm 1: Trajectory segmentation algorithm

1. Compute the F-matrix corresponding to the first and the second key-frame by using a RANSAC-based procedure and label the inliers as background trajectories.
2. Compute the F-matrix with the outliers of Step 1 by using again RANSAC and label the inliers as IMO trajectories.
3. Compute the centroid of the inliers of Step 2 and check their distances. Reject the features whose distances are higher than a threshold.
4. Increase the number of features on the IMO in consecutive key-frames with guided-matching.
5. Repeat Step 2 to 4, as long as the F-matrix estimation is reliable and most of the remaining features are spatially close.
6. Proceed to the next key-frame. Estimate the F-matrix between the last and the current key-frame for each

motion using the labeled trajectories and classify new trajectories using Step 1 to 5.

7. Repeat Step 6 for all key-frames.

Figure 2.1 gives an example of the background and IMO trajectory segmentation (frame 1 and 13 of the “Desk”-sequence captured in an office). The green crosses indicate the background trajectories and the blue squares belong to the IMO.

3. Sparse Reconstruction

The sparse reconstruction module handles the reconstruction of each element in the scene, by applying a novel sequential MFSfM approach, which is described below, on each segmented feature partition. The information extracted in this module is incorporated into the dense estimation procedure, either in the form of required parameters (camera pose estimates and frame set to be used in the reconstruction), or initial estimates (point cloud).

3.1 Prioritization

A typical video sequence contains a substantial amount of information. For example, a 200-frame sequence has 19900 frame pairs that can be used for triangulation. However, it is neither feasible, nor desirable to process all these pairs. Hence, the most significant design criterion for the sparse reconstruction module is to achieve a reliable structure and pose estimate, while maintaining a reasonable computational efficiency.

The batch methods are known to be efficient in handling such vast amount of data. However, they lack a significant facility that exists in sequential methods: The intermediate results, which are obtained from the already processed frames, can be incorporated into the processing of the remaining ones, to improve the final result. Obviously, utilization of this capability renders the result dependent on the processing order of the frames, which brings up the issue of how to determine a preferable order. Regarding to this issue, a novel solution is proposed in the following paragraphs.

Another motivation to study this question is the fact that consecutive frames in a video sequence have very narrow-baseline. Hence, it is not possible to process them in their default (temporal) order, since a wide-baseline is often essential for the success of the structure estimation. A common practice is to employ frame skipping, however, a reliable frame pair is not guaranteed, unless some properties of the motion is known beforehand.

For properly ordering (prioritizing) the frame pairs, a priority metric should be determined. The following two criteria should be considered for such a metric:

- **Fast convergence to a reliable estimate:** Since the quality of the subsequent reconstructions depend on the current (intermediate) structure estimate, the errors in the first few pairs might cause the entire estimation procedure to collapse.

- **Fast recovery of the whole structure:** The number of reconstructed 3D points should be maximized, while processing minimum number of frame pairs.

In this study, a novel metric is considered, in terms of the weighted sum of the baseline distance and a nonlinear function of the number of matches. For the computation of the relative poses of all frame pairs, it is enough to pick a frame pair and estimate the locations of the rest of the cameras in the sequence relative to the structure estimated from this pair. These pose estimates are later used in the sequential 3D reconstruction step. The number of matches between the frames is obtained from the trajectories.

The priority metric, p , utilized in the algorithm to evaluate the feasibility of a frame pair for reconstruction is defined as

$$p = d + \frac{a}{1 + \exp(b(n - c))}, \quad (3.1)$$

where d is the baseline distance between the cameras, n is the number of feature matches, a , b and c are the design parameters of the sigmoid function appearing in the second term. The nonlinear (sigmoidal) weighting keeps the contribution of the second term within a bound, when there is a relatively small or large number of matching features. As for the design parameters, a should be chosen such that, the contributions of the two terms of the metric are balanced. b and c determine the sensitivity to the number of points and the cut-off points. They should be chosen such that the sigmoid term is responsive to a reasonably large range of number of features.

3.2 Prioritized Sequential Reconstruction

For the sake of clarity of the following discussion, two definitions are necessary.

Definition 1: A *sub-estimate* is a structure estimate obtained by the triangulation of the matching features in a single frame pair¹.

Definition 2: A *sub-reconstruction* is an intermediate structure estimate obtained from a collection of *sub-*

¹ It should be stressed that, while in this study a 2-view reconstruction approach is preferred due to the availability of relatively simple, mature and reliable techniques, a sub-estimate can be constructed by any of the existing methods.

Ordered pair list: 1:6, 2:4, 1:5, 3:4, 3:6, 7:9, 3:8, 7:8

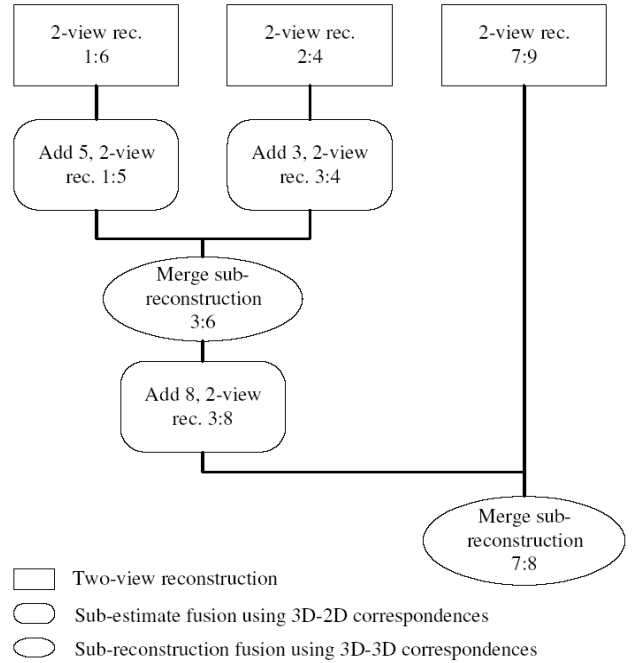


Figure 3.1: Sequential reconstruction scheme example

estimates belonging to a subset of frames of the video sequence.

Two distinct sub-reconstructions cannot have any common frames. The global motion and structure estimate is computed by merging the sub-reconstructions.

The core of the reconstruction algorithm is based on [12]. The basic idea is to start with an initial reconstruction by triangulation, and add new frames by first estimating their pose by 3D-2D matches, and then to compute the sub-estimate corresponding to the last and the current frame, again via triangulation. This sub-estimate is used to add new points and refine the reconstruction for the existing ones. However, this algorithm is designed to process the frame pairs in a certain order (e.g, $F_1-F_2, F_2-F_3, F_3-F_4, \dots$), and changing the order of frame pairs requires some modifications.

Consider the pairs F_m-F_n and F_p-F_q , which are assumed in priority order with respect to the proposed metric in (3.1). If the pairs have one common frame, then they can be processed by using the original algorithm in [12], (i.e. $n=q$, then F_m-F_n, F_n-F_p) to obtain a single sub-reconstruction. If they have no common frames, two separate sub-reconstructions (each including a single sub-estimate) can be computed for each frame pair. Assume the latter occurs and let the sub-reconstructions be T_1 and T_2 . Next, consider a third pair F_r-F_s . The cases that it has no common frames with neither of the sub-reconstructions, or has one with

either of them are already handled. A new possible case is, one member belongs to T_1 and the other to T_2 (i.e. $r=m$ and $s=q$).

The fusion of T_1 and T_2 requires the estimation of a similarity transformation defining a mapping between the points of the sub-reconstructions. The fundamentals of the estimation procedure are described in [19]. The basic idea is first to determine 3D-3D matches, then to use RANSAC to find a projective transformation that maps as many matches as possible, then to refine the estimate by using all available pairs and finally to further refine the estimate by a nonlinear minimization.

One possible final case is when both frames in the pair are already included in a single sub-reconstruction. In this case, one may skip the pair, or process it to obtain additional points. A typical reconstruction procedure is depicted in Figure 3.1. The complete reconstruction algorithm is summarized below:

Algorithm 2: Prioritized sequential 3D reconstruction

Given the internal calibration parameters and the correspondence information for all frames as trajectories:

1. Compute the initial 3D reconstruction
2. Estimate the pose of each frame with respect to the first frame in the initial reconstruction by using the 3D-2D correspondences
3. Compute the *priority metric* and order the pairs
4. While the priority metric is above the threshold or all pairs are not processed
 - a. If no member of the pair belongs to any of the existing sub-reconstructions, initialize a new sub-reconstruction
 - b. If one member of the pair belongs to an existing sub-reconstruction, add the other frame to this sub-reconstruction (Algorithm in [12])
 - c. If two members of the pair belong to the same sub-reconstruction, process using the algorithm in [12].
 - d. If two members of the pair belong to different sub-reconstructions, merge the sub-reconstructions
5. If the number of remaining sub-reconstructions is larger than 1, then merge all of them into a global estimate.

One last remaining issue is the choice of the pair for the initial reconstruction in Step 1. When choosing an initial frame pair, one should consider the following:

- The estimated structure should be reliable, as 3D-2D matches are used to locate the other cameras.
- The estimated structure should have as many matches as possible with the rest of the frames in the sequence, as the reliability of pose estimates depends on the number of matches.

The key-frames detected in the tracking module satisfy the first criterion. As for the second criterion, intuitively, it is better to choose a pair that is close to other camera views. The locations of the key-frames can be utilized to obtain a

(rather crude) estimate of the camera path, and this path estimate can be used as a guideline to choose the pair. Note that this is a computationally inexpensive procedure, as the required F-matrices are already computed in the segmentation module.

In a sequence with many covered and uncovered regions, it may not be possible to find an initial structure estimate satisfying the second criterion for the entire sequence. In that case, it is better to partition the sequence into subsequences, run separate reconstruction processes with different initial frame pairs, and finally, merge the computed structures using 3D-3D matches. Notice that this case can be detected and handled automatically by utilizing the camera path defined by the key-frames.

4. Dense Depth Estimation

The 3D structure of a scene can be represented as point samples, which are obtained with respect to a regular grid of the image plane of a camera view. If the projection parameters of the camera are known, the depth of each pixel uniquely defines the 3D position of the sampled point projected to the corresponding pixel, allowing the recovery of a dense 3D structure of the scene. Obviously, the use of multiple views improves the accuracy of the estimate. Hence, in this study, some stereo matching approaches [14], are adapted to the multiple-view case, in order to estimate the dense depth map of a camera view.

4.1. Searching the Depth Space

In dense disparity estimation for the stereo case, pixel-to-pixel matching on epipolar lines reduces the search space. However, for the multiple-view case, it is more practical to restrict the depth search space into discrete depth planes instead [22]. The depth planes are positioned parallel to the image plane of the reference camera view. Since a pixel in the reference plane lies on the corresponding epipolar lines in the other views, the depth planes determine the possible locations on these lines. In order to avoid any irregular sampling on the epipolar lines, which can cause loss of intensity information in matching, the depth planes are placed so that, on the epipolar line of the nearest camera, the distance between them is 1 pixel.

4.2. Dense Depth Estimation on Markov Random Fields

Finding a global solution for the dense depth estimation is an NP-hard problem [14], which needs some regularization tools to be utilized. Hence, in order to be able to take both the color consistency and the smooth variations of the depth (except across the object boundaries) into account, the dense depth field is modeled via a Markov Random Field (MRF).

The MRF formulation transforms the dense depth estimation problem into finding the most probable configuration of the depth field in a Bayesian framework. The Bayesian structure of MRF provides a conditioning property across the neighboring pixels of the depth field, which can be used to enforce the smoothness constraint. The probability of depth assignment for each pixel is calculated by some cost function, which favors the color consistency among the matches and the smoothness in a certain neighborhood:

$$U(X) = \sum_{x \in X} f(I_R(x_d), I_1(x_d), \dots, I_{N-1}(x_d)) + \lambda \cdot \sum_{x, y \in N} S(x_d, y_d) \quad (4.1)$$

where f measures the deviation from the color consistency, and S from the smoothness constraint, taking the depth discontinuities into account. λ is a parameter to adjust the smoothing effect of the global MRF solution. In order to solve this minimization problem, many different algorithms are proposed such as iterated conditional modes (ICM), simulated annealing, Gibbs sampler [23]. In the following section the application of belief propagation (BP) [24] on dense depth estimation will be summarized.

4.2.1. MRF solution with Belief Propagation

In the belief propagation (BP) approach [24], the depth field is modeled as a Bayesian network. The desired solution is searched by iterative inferences of nodes (pixels) in this field. The resulting iterative inference is achieved via some messages sent by the nodes from their neighboring nodes. The messages are denoted as vectors with the same dimension of the number of probable depth values. The entries vary proportionally, depending on how a message sending node assumes that the receiver has a corresponding depth value. The application of BP on the stereo disparity problem can be found in [25].

The superiority of the BP method is its ability to consider all possible configurations by calculating the probability densities of all pixels. The BP algorithm solves the MAP estimate exactly on loopless networks [26]. However, BP is also known to be a good approximation for the loopy networks [17]. Although, the BP method has drawbacks in terms of computational cost and memory requirements, the results are much more promising and less sensitive to the initialization of the estimation algorithm.

5. Simulation Results

For evaluating the performance of the overall system, experiments are conducted on various video content. The results from “Palace”, a 200-frame sequence, and “Cliff”, a 108-frame sequence, both captured from a typical TV

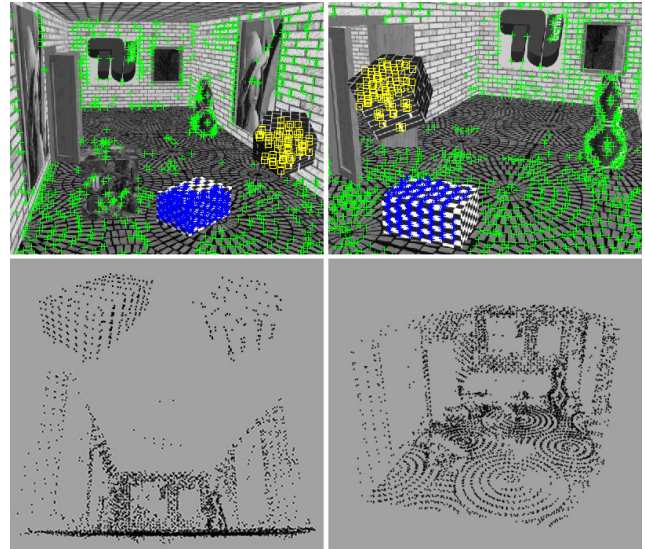


Figure 5.1: Segmentation results of “TUB-Room” (top) and sparse reconstruction of the background and the IMOs (bottom)

broadcast and “TUB-Room”, a 240-frame synthetic sequence are presented for the individual modules.

Figure 5.1 shows the segmentation (top) and sparse reconstruction results (bottom) of the “TUB-Room”-sequence. Since the sequence contains IMOs, the segmentation module labels the trajectories according to their 3D motion (green crosses indicate the background, blue triangles and yellow squares indicate the IMOs, respectively).

A sample reconstruction of the “Cliff” and the “Palace”-sequences are depicted in Figure 5.2 and Figure 5.3, respectively. The performance of the sparse reconstruction module is illustrated in Table 5.1. The algorithm successfully recovers most of the structure points using only a fraction of the all available frame pairs.

Table 5.1: Performance of the sparse reconstruction module

	# frame pairs (used/total)	# 3D points (recovered/total)	Average reproj.
Cliff	45 / 5778	5890 / 8212	0.95
Palace	25 / 19900	2771 / 3546	0.90
TUB-Room	17 / 28680	4716 / 6095	0.27

The dense depth estimation is performed for both stereo and multiple cases in order to see the improvement by increasing the frame numbers (see Figure 5.4). It is clearly observed that the matching errors and the occluded regions decrease for the multiple-view case. In order to demonstrate the dense reconstruction results visually, an arbitrary view is generated by using a mesh rendering in Figure 5.5.

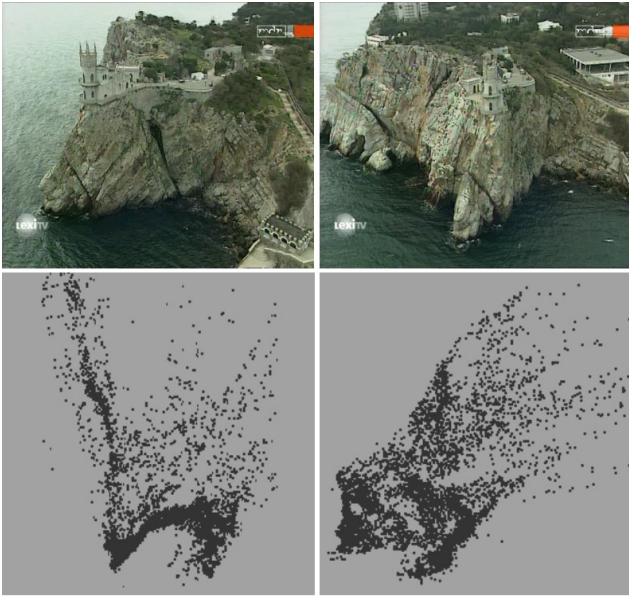


Figure 5.2: *Top row:* First and last frames of “Cliff”. *Bottom row:* Sparse reconstruction. (Top and top-right views)

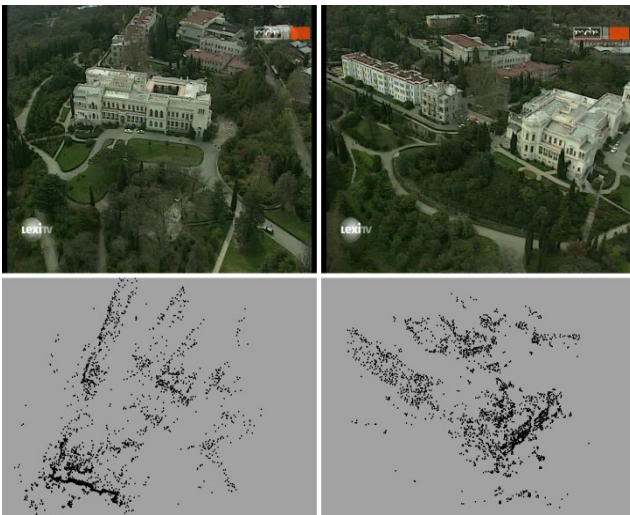


Figure 5.3: *Top row:* First and last frames of “Palace”. *Bottom row:* Sparse reconstruction (top and top-left views)

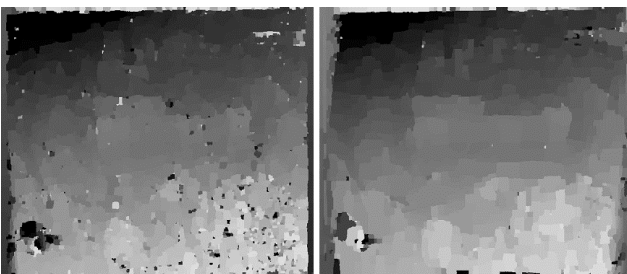


Figure 5.4: Dense depth maps of a reference view in “Palace” for the stereo (left) and multiple (right) cases using BP.

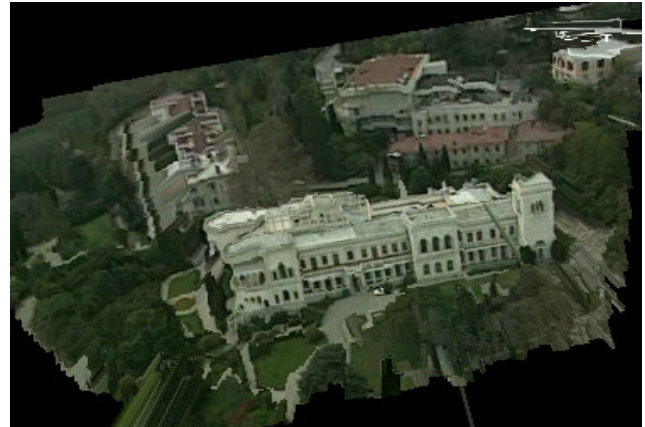


Figure 5.5: Arbitrary view generated with respect to the estimated dense field of “Palace”.

6. Conclusion

In this paper, a complete system for 3D reconstruction of dynamic scenes from broadcast video data is presented. A four stage procedure is designed to realize this task. The system utilizes well-known and reliable algorithms, such as KLT and E-matrix based self-calibration methods, as well as some novel approaches in feature segmentation, sparse and dense reconstruction. The system is fully automatic, in the sense that it only requires a video sequence, as an input, and gives a dense depth map of the observed scene, as its output.

The experiments show that the segmentation and sparse reconstruction performance is quite remarkable. As for the dense reconstruction, the utilization of multiple frames for the dense depth estimation is shown to be an effective way to handle occlusions and improve the reconstruction quality.

While the experiments indicate that the system delivers promising results in terms of efficiency and performance, it is still susceptible to degenerate sequences for self-calibration, and the lack, or degenerate configuration of features, especially in some indoor sequences and in the presence of small IMOs. Future works will focus to eliminate such problems by utilizing higher level geometric entities such as lines and planes.

7. Acknowledgment

The authors would like to thank Ugur Topay for sharing his expertise and software for self-calibration.

8. References

- [1] O. Schreer, P. Kauff, T. Sikora (Eds.), “3D Videocommunication: Algorithms, Concepts and Real-time Systems in Human Centered Communication”, John Wiley & Sons Ltd, Chichester, England, 2005

- [2] M. Pollefeys, R. Koch, M. Vergauwen, L. Van Gool, "Automated reconstruction of 3D Scenes from Sequences of Images", *ISPRS Journal of Photogrammetry and Remote Sensing* (55) 4, pp. 251-267, 2000
- [3] C. Tomasi, T. Kanade, "Shape and Motion from Image Streams: A Factorization Method", *Journal of Computer Vision* 9(2), pp. 137-154, 1992
- [4] P. R. S. Mendonca and R. Cipolla, "A Simple Technique for Self-Calibration", *CVPR*, 1999
- [5] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features", Technical Report CMU-CS-91-132, Carnegie Mellon University Technical, 1991
- [6] M. Irani, P. Anandan, "A Unified Approach to Moving Object Detection in 2D and 3D Scenes", *IEEE Trans. On PAMI* Vol. 20, Issue 6, pp. 577-589, June 1998
- [7] Y. Weiss, "Segmentation Using Eigenvectors: A Unifying View", *Proceedings of ICCV*, pp. 975-982, 1999
- [8] W. Fitzgibbon, A. Zisserman, "Multibody structure and motion: 3D reconstruction of independently moving objects", *ECCV*, 2000
- [9] P.H. S. Torr, "Geometric motion segmentation and model selection", *Phil. Trans. Royal Society of London*, pp. 1321-1340, 1998
- [10] G. Qian, R. Chellappa, Q. Zheng, "Bayesian Algorithms for Simultaneous Structure from Motion Estimation of Multiple Independently Moving Objects", *IEEE Trans. on Image Processing*, Vol. 14, No. 1, January 2005
- [11] S. Soatto, P. Perona, "Reducing 'Structure from Motion': a General Framework for Dynamic Vision Part 1: Modeling", *Pattern Analysis and Machine Intelligence*, 20 (9), September 1998
- [12] M. Pollefeys, "Tutorial on 3D Modeling from Images", *ECCV*, 2000.
- [13] J. P. Costeira, T. Kanade, "A Multibody Factorization Method for Independently Moving-Objects", *IJCV*, (29) No. 3, pp. 159-179, September 1998
- [14] D. Scharstein, R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", *International Journal of Computer Vision*, Springer, 2002
- [15] T. Kanade, M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment", *IEEE Trans. on PAMI*, 16(9), pp. 920-932, 1994
- [16] Kolmogorov V., R. Zabih, "Computing visual correspondence with occlusions using graph cuts", *ICCV*, vol. II, pp. 508-515, 2001
- [17] Y. Weiss, "Belief Propagation and Revision in Networks with Loops", MIT Technical Report AIM-1616, 1997
- [18] C. Strecha, L. Van Gool, "PDE-based Multi-view Depth Estimation", 1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT), pp. 416-425, 2002
- [19] R. Hartley, A. Zisserman, "Multiple view geometry", Cambridge University Press, UK, 2003
- [20] P.H.S. Torr, A.W. Fitzgibbon and A. Zisserman, "The Problem of Degeneracy in Structure and Motion Recovery from Uncalibrated Image Sequences", *International Journal of Computer Vision*, 32(1):27-44, August 1999
- [21] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM*, pages 381-385, 1981
- [22] R. Szeliski, P. Golland, "Stereo Matching with Transparency and Matting", Kluwer Academic Publishers, *International Journal of Computer Vision*, 32 (1), 1999.
- [23] A. M. Tekalp, "Digital Video Processing", Prentice Hall, 1995
- [24] T. Weiss, "Interpreting Images by Propagating Bayesian Beliefs", *Advances in Neural Information Processing Systems* 9, 1996
- [25] J. Sun, H.-Y. Shum, N.-N. Yeng, "Stereo Matching Using Belief Propagation", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Volume 25, No. 7, July 2003
- [26] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference", Morgan Kaufmann Publishers, 1988