IN-LOOP FEATURE TRACKING FOR STRUCTURE AND MOTION WITH OUT-OF-CORE OPTIMIZATION

Nicolas Herrero[1], *Jose-Luis Landabaso*[2], *Guillermo Gallego*[3], *Jose-Carlos Pujol-Alcolado*[4]

[1,2,4]Telefonica Research. Via Augusta 177. 08021, Barcelona, SPAIN
[3]School of Electrical and Computer Engineering. Georgia Institute of Technology. GA-30332, USA

ABSTRACT

In this paper, a novel and approach for obtaining 3D models from video sequences captured with hand-held cameras is addressed. We define a pipeline that robustly deals with different types of sequences and acquiring devices. Our system follows a *divide and conquer* approach: after a frame decimation that pre-conditions the input sequence, the video is split into short-length clips. This allows to parallelize the reconstruction step which translates to a reduction in the amount of computational resources required. The short length of the clips allows an intensive search for the best solution at each step of reconstruction which robustifies the system. The process of feature tracking is embedded within the reconstruction loop for each clip as a difference with other approaches. A final registration step, merges all the processed clips to the same coordinate frame.

Index Terms— Structure and Motion, 3D Reconstruction, Frame Decimation, Feature Tracking

1. INTRODUCTION

Structure and Motion (SaM) techniques have evolved from providing solutions for particular geometric problems to the definition of robust pipelines for automatic 3D reconstruction systems from both video sequences or photo collections [1, 2, 3]. Nowadays, a growing interest on reducing their computational complexity while preserving their performance and reliability has arisen. Bad scalability of Newton-like optimization represents the most restrictive bottle neck in terms of computational complexity for this discipline. Therefore several approaches are found in literature to face this problem [4, 5].

In the case of photo collections, an accepted practice consists on pre-analyzing the set of snapshots and clustering them upon an affinity criterium. Clusters are independently processed [4, 5] and *locally* optimized. The equivalent when dealing with video sequences consists on defining atomic structures (*i.e.* triplets [6, 7]) that, once reconstructed and combined [8], represent the whole scene. This type of practice allows a parallelization for both the process of reconstruction and for the optimization step, which dramatically reduces the amount of computational resources needed.

Nevertheless, other issues may be listed in SaM from video: drift propagation through frames as the sequence length grows, temporal redundancy [9] and the difficulty of providing the reconstruction loop with an accurate set of tracked features through frames.

In this work, a full pipeline for the automatic recovery of SaM from rigid scenes is presented. With the aim of tackling the aforementioned issues associated to SaM, we add to the process of reconstruction the following contributions: *(i)* a frame decimation algorithm, *(ii)* a feature tracking process interlaced within the reconstruction loop and *(iii)* a parallel approach through the use of algorithms for registering partial 3D reconstructions. Our system is proven to be robust to different qualities of the camera used for capturing the scene. Thus, results for handycams and cell-phone cameras are presented.

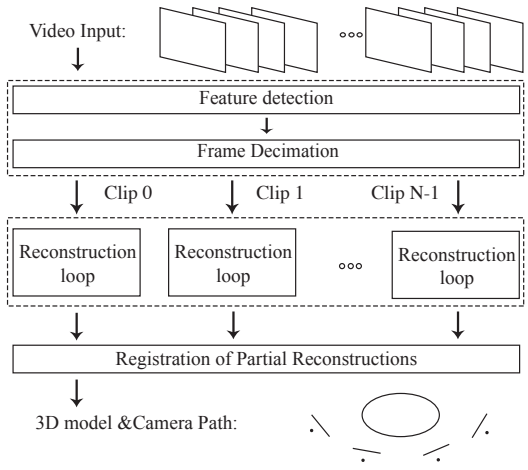


Fig. 1. Global scheme for the SaM pipeline

2. SYSTEM OVERVIEW

A global description of the proposed SaM pipeline is depicted in Fig. 1. SIFT-like features are detected for all the input frames of the video sequence. They serve as the input for an smart frame decimation algorithm in order to get rid of temporal redundancy. Next, the retrieved set of decimated frames are divided into clips that are independently processed. This frame decimation scheme provides the SaM block a set of well-conditioned Key-Frames (KF's) with a sufficient number of common features.

The presented system integrates the steps of feature matching and tracking within the reconstruction loop. This makes a difference with respect to other approaches where the tracking is carried out beforehand [1, 3, 9]. Finally, a merge step registers all the partial 3D reconstructions into a common coordinate frame. This last stage is mandatory since the retrieved partial reconstructions are referenced to an arbitrary coordinate system although they represent the same static scene.

3. ROBUST STRUCTURE AND MOTION

Along this section, the core blocks of our SaM system are presented. The notation used is the following: the $j-th$ 3D point is represented

by a 4-vector $\mathbf{X}^j$ and its projection to camera i as a 3-vector $\mathbf{x}_i^j$, both in homogeneous coordinates. 3D points are mapped to the image plane by means of the camera 3×4 projection matrix $\mathtt{P}_i = \mathtt{K}[\mathtt{R}|\mathbf{t}]$ as $\mathbf{x}_i^j = \lambda \mathtt{P}_i \mathbf{X}^j$. $\mathtt{K}$ refers to the intrinsic parameters of the camera and $\mathtt{R}$ and $\mathbf{t}$ to its relative rotation and translation to a metric coordinate frame. Parameter λ refers to an *up to scale* ambiguity.

The epipolar geometry between two frames is satisfied by the 2D relation $\mathbf{x}_i^{jT} \mathtt{F} \mathbf{x}_k^j = 0$ as a consequence of the projection of the same 3D point $\mathbf{X}^j$ to two different cameras $\mathtt{P}_i, \mathtt{P}_k$. A point $\mathbf{x}^{jT}$ is transferred from one image to another according to $\lambda \mathbf{x}_i^j = \mathtt{H} \mathbf{x}_k^j$. $\mathtt{F}$ and $\mathtt{H}$ are the Fundamental Matrix and a two-view Homography respectively [10].

3.1. Frame Decimation

In [9], some of the desired features for the retrieved key-frames are listed: KF's need to be sharp, that is not affected by motion blurring or auto-focus artifacts. Baseline and parallax between frames has to be large enough to allow a good conditioning of the epipolar geometry. Finally, a sufficient number of point correspondences $\{\mathbf{x}_i^j, \mathbf{x}_k^j\}$ between pairs of frames has to be available. In addition the frame decimator is desired to be *idempotent*.

We propose a new frame decimation algorithm inspired in [9] but adapted to work in a feature-based approach. Frames I_i from the input sequence $I[n]$ are sorted into a list F in order of increasing number of detected feature points. The set of feature points associated to frame I_i is denoted as F_i. Therefore we are assuming that *focused* or *sharp* frames are the ones with larger number of features with respect to their neighbors in a small time interval.

A frame I_i in F is removed if it is considered to be redundant in the input sequence $I[n]$. The condition to determine if I_i is redundant, consists on evaluating the image affinity between its two neighbors I_{i-1} and I_{i+1}. Let us define the image affinity measure as the Jaccard index, which represents the similarity between sample sets:

$$a_i = \frac{\#(F_{i-1} \cap F_{i+1})}{\#(F_{i-1} \cup F_{i+1})} \tag{1}$$

and the apparent motion m_i is estimated as the median of 2D motion for each feature correspondence: $m_i = \underset{j}{\text{median}} ||\mathbf{x}_{i-1}^j - \mathtt{H}\mathbf{x}_{i+1}^j||$. Operator $\#$ denotes number of elements in a set.

In our experiments, I_i is considered to be redundant if its neighbors I_{i-1} and I_{i+1} fulfill the following conditions:

$$I_i \text{ redundant iff } \begin{cases} a_i < 0.25 \\ m_i < 10\% \text{ of image diagonal} \\ \#(F_{i-1} \cap F_{i+1}) \geq 100 \end{cases}$$

Frames are processed in the same order than they are stored in F and multiple passes could be required. The algorithm stops when no frame is discarded after a pass.

With this scheme, non-sharp frames are the first to be removed since they are the first to be evaluated. The frame decimation algorithm adapts to the motion of the camera with respect to the scene and outputs a set of frames whose relative motion is more isotropic.

3.2. In-Loop Feature Tracking and SaM

3.2.1. Feature Tracking

As mentioned, one of the major issues in SaM is to feed the reconstruction loop with a good set of features tracked along time. Traditionally, feature matching and tracking has been carried out prior to the reconstruction loop and, therefore, using no available 3D information [1]. In our work, the process of assigning feature points to existing tracks (the reprojection of a single 3D point to a set of images) is interlaced within the SaM process. This practice robustifies the feature tracking process since it combines 2D and 3D information. That is only grouping into tracks those feature points triangulated.

Several candidates are considered for both selecting an initial pair and adding new views. These candidates are evaluated upon a score function computed after matching pairs of frames. Once a new camera is added to the current SaM state or the initial pair has been determined, matched points are assigned to common tracks. Hence, feature tracking is carried out along frames in the same order as their position and their associated 3D points are retrieved.

Feature matching follows the *kd-tree*-based approach for retrieving the nearest neighbor of a SIFT-like descriptor. Moreover, feature correspondences are used to feed a RANSAC algorithm that computes a fundamental matrix $\mathtt{F}$ between frames. This matrix is used to classify each putative match as outlier/inlier [10]. The threshold is set to 1 pixel of *point to epipolar line* cost.

3.2.2. Initial-pair estimation

Selecting a good initial pair for starting the process of reconstruction is a crucial choice that will determine the final overall performance. Thus, this step needs to be as robust as possible. Since the input for the SaM block is a small set of KF's selected from a short clip (20 KFs), an exhaustive search can be carried out.

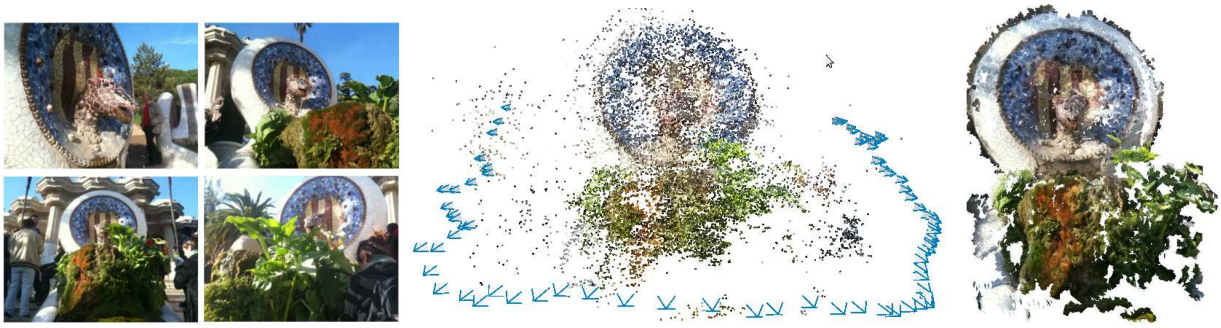
Nevertheless, there are some requirements for the initial pair that need to be fulfilled. The epipolar geometry needs to be satisfied, that is well modeled by a fundamental matrix $\mathtt{F}$. Moreover, the selected pair of frames can not configure a degenerate case and there must be enough correspondences. Therefore, a quality measure may be defined both for selecting the best pair for the initial pose estimation and for discarding bad conditioned pairs, which speeds up the full search.

For each tested pair I_i and I_k, a feature matching is performed and correspondences are used for robustly computing both $\mathtt{F}$ and $\mathtt{H}$ with RANSAC. We measure the good conditioning of the initial pair as the percentage of outliers ρ_H when modeling the pair of images with an homography $\mathtt{H}$:

$$\rho_H(i,k) = 1 - \frac{\#\left(F_i \cap F_k\right)_H}{\#(F_i \cap F_k)}, \tag{2}$$

where $\left(F_i \cap F_k\right)_H$ refers to the total inliers for the homography $\mathtt{H}$ case. The threshold for classifying a correspondence as inlier/outlier to an homography is of 1 pixel of 2D distance $m_{ik}^j = ||\mathbf{x}_i^j - \mathtt{H}\mathbf{x}_k^j||$. Pairs of views whose percentage of outliers for $\mathtt{H}$ is $\rho_H(i,k) < 0.5$ are discarded for further processing. On the other hand, the pair of cameras $\{\mathtt{P}_i, \mathtt{P}_k\}$ with the biggest score $\rho_H(i,k)$ and more than 100 valid 3D points after triangulation, is selected as the initial pair. The algorithm for estimating the relative position between the first pair of cameras consists on placing one of the cameras at the origin $\mathtt{P}_i = \mathtt{K}[I|\mathbf{0}]$ and estimating the relative rotation and translation of the other one $\mathtt{P}_k = \mathtt{K}[\mathtt{R}|\mathbf{t}]$. This is achieved by means of factorizing the Essential matrix $\mathtt{E} = \mathtt{K}^T \mathtt{F} \mathtt{K}$ as in [10].

Once the projection matrices are retrieved, the process is followed by a linear triangulation step. Obtained structure is further refined by discarding those points with large uncertainty (angle less than 1°) and reprojection error above 1 pixel. After this purge step, the process of feature tracking starts. In this case feature tracking is straightforward. Since 3D point are triangulated for the first time, each pair of matched features are assigned into a common track.



(a) Snapshots from input video sequence (b) SaM point cloud (c) Densification

Fig. 2. Visual results for *Garden* sequence

3.2.3. Updating Structure and Motion

After initial pair estimation, the system enters the loop of progressively adding new cameras and triangulating more points. Candidate projection matrices for cameras to be incorporated are estimated from 2D/3D correspondences inside a RANSAC process. Next, projection matrices are refined via a Gold Standard Resection algorithm [10].

Once again, since the number of KF's selected from each video clip is small, we may exhaustively search which the best camera is to be added at each iteration in the update loop. Candidate cameras to be resected are sorted in order of decreasing shared 2D/3D points with respect to already resected cameras. A new candidate is accepted if its reprojection error after resection is below 5 pixel.

In our system, resection of each new camera is performed with respect to an already resected camera. That implies estimating the projection matrix just by means of the common 2D/3D points shared with another already resected view. According to our experience, this relative resection robustifies SaM, since the linear algorithm is best conditioned.

Linear triangulation and recursive feature tracking to already existing cameras follows resectioning. Even for this case, the quality of new triangulated points is evaluated and non consistent 3D points are discarded. The whole pipeline ends with an optional Bundle Adjustment (BA) to improve the consistency of the SaM updating.

For this step of the SaM pipeline, several possible situations need to be checked for each pair of matched features in order to ensure a reliable feature tracking. Given two matched features $\{\mathbf{x}_i^j, \mathbf{x}_k^r\}$, j and r refer to the track they are assigned to. If their value is not assigned yet, a new track is created and then $j = r$. Otherwise, if j was assigned but not r, $\mathbf{x}_k^r$ is assigned to track j. If they belong to different tracks $j \neq r$, the possibility of merging is studied and, if it is not possible, both of them are deleted for the sake of consistency. The merging condition is evaluated by checking if there exists a resected view which contains a pair of features with the tracks j and r. If not, both tracks are considered to be same and consequently merged.

3.3. Registration of partial reconstructions

As presented in 3.2 the process of relative pose estimation for the first pair is not constrained to be performed in a concrete coordinate system. Therefore each one of the metric reconstructions of clips will differ in rotation, translation and an scale factor although they represent the same static scene.

The basic idea of registration: given two partial reconstructions $\{\mathtt{P}_i, \mathbf{X}^j\}, \{\mathtt{P}'_k, \mathbf{X}'^r\}$ in different coordinate systems estimate a similarity transformation $\mathtt{H}_s$ such that $\{\mathtt{P}_i, \mathtt{P}'_k\mathtt{H}_s^{-1}, \mathbf{X}^j, \mathtt{H}_s\mathbf{X}'_r\}$ are referenced to the same global frame. Available registration algorithms make use of structure and motion correspondences such as common 2D/3D points and/or overlapping cameras [8, 6]. In our case the registration technique used is the one from [11], where point and camera correspondences between partial reconstructions are exploited in order to derive a linear algorithm for estimating $\mathtt{H}_s$.

It is important to note that for short video sequences, the pipeline described in 3.2 may be used standalone. That is without the needing of an initial clipping process and a final registration algorithm. Nevertheless, the parallel approach followed allows dealing with longer sequences avoiding drift propagation and reducing computational cost since moving from a global to a local BA.

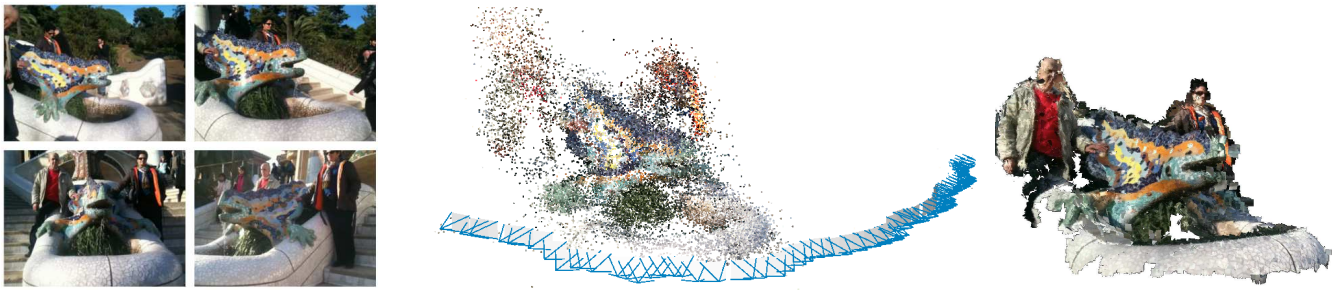
4. RESULTS

Three sequences are used for studying the performance of the proposed system[1]: *(i)* Gaudí's Dragon, *(ii)* a garden and *(iii)* a set of objects over a table. First two sequences were captured with an iPhone 3GS phone, while the second one with an HD handycam. Two configurations are studied: (*a*) OL-IL refer to Out- or In-Loop feature tracking and (*b*) US-FD to a key frame selection based on an Uniform Sampling or in a Frame Decimation algorithm respectively.

Since US and FD provide different number of KFs, one of the quality parameters listed in Table 1 is the reconstruction time. This parameter is directly related to the number of cameras obtained respect the number of KFs. As seen, IL-FD outperforms OL-UL for the first and third sequences. Since the IL-FD configuration provides a more isotropic temporal distribution of KFs and a more stable feature tracking, isolated frames are less frequent than for OL-US. Therefore the pose of a larger number of cameras is obtained.

The number of triangulated 3D points is similar for both configurations. Nevertheless, total frames processed in IL-FD is lower than for OL-US. That indicates that bounding the affinity factor in eq. (1) during FD, is allowing to introduce larger amounts of non-triangulated points at each iteration of the SaM update loop. In other words, it is acting as a *renewal factor*. Furthermore, the global reprojection error between both approaches has been studied before and after a global BA. Post-BA error is similar for any configuration or

[1]The three sequences and additional results may be checked in project's web page: http://surfing.tidprojects.com/loginICIP.php



(a) Snapshots from input video sequence (b) SaM point cloud and camera trajectory (c) Densification

Fig. 3. Visual results for *Dragon* sequence

Sequence	Method	3D Points	Cameras	KFs/Frames	Clips	Error: pre-BA / post-BA	Time: Rec./Total
Dragon	OL-US	10367	34	50 / 248	4	4.840 / 0.314	6.8s / 9.9s
	IL-FD	14759	45	45 / 248	5	0.424 / 0.261	9.9s / 9.9s
Garden	OL-US	18239	116	116 / 577	8	1.581 / 0.323	23.1s / 23.1s
	IL-FD	22537	77	77 / 577	8	0.401 / 0.256	23.1s / 23.1s
Table	OL-US	12307	66	80 / 400	6	0.397 / 0.348	13.2s / 16.0s
	IL-FD	9618	30	32 / 400	4	0.488 / 0.429	15.2s / 16.0s

Table 1. Numerical evaluation of three different registration schemes on the given datasets.

sequence. However the reprojection error before BA is lower for the IL-FD step for the first two sequences (captured with cell-phone). That leads us to think that IL-FD represents a robust configuration for any type of camera quality even without the need of an expensive global BA.

Finally, the accuracy of the obtained pose for cameras is validated by serving as the input for a densification algorithm. Although several approaches could be chosen, one based on Furukawa's patch-based densification algorithm [12] was selected for generating results from Figs.2 and 3.

5. CONCLUSIONS AND FUTURE WORK

A full pipeline for recovering the structure and the camera motion of a rigid scene from a video sequence has been presented. Several contributions have been included along this work: frame decimation, in-loop feature tracking and a parallel approach for SaM. Robustness and reliability has been proven for the system and compared to other approaches. In addition, a configuration ensuring quality and avoiding expensive global optimization has been defined.

In the future, several improvements may be done: deep analysis of computational complexity, avoiding of exhaustive searches and content-based determination of clip lengths.

6. REFERENCES

[1] N. Snavely, S.M. Seitz, and R.S. Szeliski, "Modeling the world from internet photo collections," *IJCV*, vol. 80, no. 2, November 2008.

[2] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Generic and real-time structure from motion using local bundle adjustment," *Image Vision Comput.*, vol. 27, no. 8, pp. 1178–1193, 2009.

[3] R. Gherardi A. M. Faranzaena, A. Fusiello, "Structure-and-motion pipeline on a hierarchical cluster tree," in *Proceedings of the IEEE International Workshop on 3-D Digital Imaging and Modeling*, October 2009.

[4] Kai Ni, Drew Steedly, and Frank Dellaert, "Out-of-core bundle adjustment for large-scale 3D reconstruction," *ICCV*, 2007.

[5] Xiaowei Li, Changchang Wu, Christopher Zach, Svetlana Lazebnik, and Jan-Michael Frahm, "Modeling and recognition of landmark image collections using iconic scene graphs," in *ECCV*, Berlin, Heidelberg, 2008, pp. 427–440, Springer-Verlag.

[6] Andrew W. Fitzgibbon and Andrew Zisserman, "Automatic camera recovery for closed or open image sequences," *ECCV*, pp. 311–326, 1998.

[7] David Nistér, "Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors," *ECCV*, pp. 649–663, 2000.

[8] Changchang Wu, Brian Clipp, Xiaowei Li, Jan-Michael Frahm, and Marc Pollefeys, "3d model matching with viewpoint-invariant patches (vip)," *CVPR*, vol. 0, pp. 1–8, 2008.

[9] David Nistér, "Frame decimation for structure and motion," in *SMILE '00: Revised Papers from Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, London, UK, 2001, pp. 17–34, Springer-Verlag.

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

[11] Authors, "Paper submitted to double-blind submission conference. pending of approval," .

[12] Yasutaka Furukawa and Jean Ponce, "Accurate, dense, and robust multi-view stereopsis," in *CVPR*, 2007.