Multibody Structure-from-Motion in Practice

K. E. Ozden[♢] K. Schindler[♡,♠] L. Van Gool[♢,♠]

[♢]ESAT / PSI - IBBT
K.U. Leuven, Belgium

[♡]Computer Science Dept.
TU Darmstadt, Germany

[♠]Computer Vision Lab
ETH Zürich, Switzerland

***Abstract*—Multibody structure from motion (SfM) is the extension of classical SfM to dynamic scenes with multiple rigidly moving objects. Recent research has unveiled some of the mathematical foundations of the problem, but a practical algorithm, which can handle realistic sequences, is still missing. In this paper, we discuss the requirements for such an algorithm, highlight theoretical issues and practical problems, and describe how a static structure-from-motion framework needs to be extended to handle real dynamic scenes. Theoretical issues include different situations, in which the number of independently moving scene objects changes: moving objects can enter or leave the field of view, merge into the static background (e.g. when a car is parked), or split off the background and start moving independently. Practical issues arise due to small freely moving foreground objects with few and short feature tracks. We argue that all these difficulties need to be handled online, as structure-from-motion estimation progresses, and present an exemplary solution using the framework of probabilistic model-scoring.**

***Index Terms*—Structure-from-motion, motion segmentation, scale ambiguity, model selection, affine degeneracy**

I. Introduction

Over the past two decades, structure-from-motion(SfM) has developed into a mature discipline of visual information processing. The classic case of a camera moving through a static scene is essentially solved in a coherent theory [4], [8], and several practical systems exist, which are able to robustly recover the structure and motion from unconstrained real-world sequences. One of the main limitations of real-world applications of structure-from-motion at present is the requirement for a static scene.

In recent years, SfM theory has been extended to dynamic scenes composed of rigidly moving objects, starting with work on two views [19], [25], [27], and progressing to linear motion in multiple views [6], [7], [27], arbitrary motion in affine sequences [2], [24], [28] (a benchmark can be found in [23]), and recently also perspective sequences [5], [11], [20].

There is a trade-off between the number of moving objects that are fitted to the feature points, and the fitting error: increasing the number of objects always decreases the error, even if the additional object is already an overfit. Several researchers have cast the problem in terms of statistical model selection [9], [19], [20], [22]. Recursive filters investigated with static SfM have also been applied to dynamic scenes in both parametric [3] and non-parametric [17] form.

Dynamic SfM thus far has concentrated on the mathematical properties of the problem, and proof-of-concept experiments restricted to simple, short sequences. The topic of this paper is to adapt, for the first time, dynamic SfM theory for application in a robust system suitable for real-world sequences. The step from the theoretical algorithm to a system-level solution brings up various challenges.

Theoretical issues arise due to the complex interactions of objects in the scene, particularly when the number of objects changes. This situation occurs quite regularly in the real world, often because moving objects stop their motion relative to the background, or start to move. Closer inspection reveals that the two operations, which at first glance appear inverse to each other, actually require a separate, and quite different treatment. Another issue is the handling of the two ambiguous, mirror-symmetric reconstructions, which arise when the perspective mapping of a small object degenerates to an affine one.

A challenge of a more practical engineering nature is that foreground objects are in most cases smaller than the background scene and move in a less constrained way, and this destabilizes the SfM computations. In order to cope with the weaker geometric constraints, one needs to take extra care to ensure a sufficient number of feature tracks of adequate length at any time. This leads to the requirement for *online SfM*: 3D reconstruction must run in parallel with feature tracking, rather than as a subsequent batch process, because the 3D geometry is needed to guide feature tracking.

The paper is structured as follows: in section II the necessary building blocks for a practical multibody SfM framework are identified and their specific difficulties are discussed. In section III we describe in some detail a real system which implements these ideas. Section IV presents experimental results on several scenes recorded outdoors with hand-held cameras, followed by a discussion and conclusion in Section V.

II. A SfM Framework for Dynamic Scenes

A. Requirements

The main requirements for a dynamic SfM framework are (i) to determine the number of independently moving objects at the beginning of a sequence, and whenever that number changes, (ii) to segment the feature tracks into different moving objects in each frame, (iii) to compute their 3D structure and the camera motion for the frame with the required accuracy, (iv) to resolve geometric ambiguities, (v) to be robust to short feature tracks due to (self-)occlusion, motion blur, etc., and (vi) to scale to realistic recording times (at least several hundreds of frames). This list suggests that feature tracking, segmentation into independent objects, and 3D reconstruction cannot be carried out as independent tasks, but need to be interleaved. In this way, all the 3D information extracted at a certain time-step is already known when processing the next frame. Feature tracking, in particular, benefits from known 3D motion since blind 2D feature search in a window is reduced to 1D search along an epipolar line, or even to the refinement of a known location ("0D search") if the corresponding 3D point is known. The accuracy of the motion estimate in turn depends on correct segmentation.

For static scenes, the described process of interleaved feature tracking and SfM computation has been well studied [1], [8], [16]. In the case of dynamic scenes, we additionally need to estimate the number of moving objects online. This number can change for a variety of reasons: objects can enter or leave the field of view, which are both easy to detect directly from the feature tracks. Furthermore, objects can also split and start to move independently, or merge and start to move as one object. Finally, the number of objects also has to be determined during initialization, since more than one object can be present from the beginning. Further requirements, which arise only in dynamic SfM are the resolution of reconstruction ambiguities, namely the relative scale between independently moving objects, and the mirror symmetry in case of affine degenerate objects. In the following, we will look at the listed components at a conceptual level, and specify their desired behaviour. Section III describes a practical implementation, which achieves this behaviour with model-scoring equations tailored to the different tasks.

B. Initializing the multibody structure-from-motion

During initialization, the 3D structure and camera motion as well as the segmentation into independent motions are established for the first time, after observing an initial set of images. This is exactly the multibody SfM problem in the form it has been treated so far in the literature, and one can directly resort to one of the developed algorithms. We will apply a method similar to [20], which naturally fits into our model-scoring framework – see Section III-D.

C. Splitting and merging motions

As explained above, splitting and merging of objects refer to situations, where the number of independently moving objects changes *without any object leaving the field of view*. In the case of splitting, several objects which previously moved as one rigid body start moving independently. Merging is the opposite: objects, which previously moved independently, attach to each other and start moving as one rigid structure. Those events should be detected properly for a robust and accurate scene modelling, and to resolve scale ambiguities – see Section II-D.

Despite their apparent relationship, splitting and merging are two significantly different problems, both formally and practically. At first glance, they are inverse operations (in the sense that one becomes the other when playing the sequence in reverse). This may lead one to believe that they can be treated with the same mathematical model. However, the direction of the time-line causes significant differences, both due to order of computations during SfM estimation, and due to the typical motions of real split and merge operations.

We will give a simple example to illustrate the subtle theoretical differences: consider a rigid object A, for which the structure and motion are already known. At a certain point in time, an object (a set of 3D points) B splits off from A, and starts to move independently of the remaining points A'. Since the 3D structure of B has been reconstructed before splitting (as a part of A), the only remaining problem is to find the new rigid transformation of B relative to the camera, which can be done with simple resection in a single frame. Now assume that we are processing the sequence in reverse order. Initially the objects A' and B are moving independently, and at a certain point in time they merge. This event can only be detected reliably, if we observe it long enough: otherwise, near-degenerate configurations can easily be miss-interpreted as apparent fusion: one can often fit a reasonable joint model, if the observed sequence is short, because there was no time to accumulate enough translation (baseline). Furthermore, the two objects have been reconstructed separately, so there is a scale ambiguity between them [13], which needs to be resolved (see next subsection), and this again requires enough baseline.

Another issue is that merging more than two objects can be safely accomplished by iteratively merging pairs of objects in arbitrary order. The same is not true for splitting: if an object splits into 3 parts – or more practically relevant, into 2 parts and some outliers – there is no split into 2 parts which produces valid structure and motion. Agreement can be tested in a greedy pairwise fashion, but disagreement cannot.

Practical considerations give rise to a further difference: when a 3D object splits, it is desirable to immediately have access to the new 3D models, because of two reasons: (1) it is quite often the case that one of the new motions (that of a small object splitting off the background) has a large component of self-rotation, causing rapid loss of features due to self-occlusion. Hence, the right model must be instantiated immediately so as not to lose the object completely. In the opposite case, when a smaller object merges into the background, there is no immediate danger of losing large numbers of features; and (2) guided tracking relies on the SfM. The 3D structure and relative camera motion are crucial for reliable feature tracking, and if splitting is delayed, tracking will suffer. Again, merging does not suffer from this problem, because the previous SfM is still valid.

D. Relative Object Scale

It is a well-known property of SfM that the resulting 3D structure is only determined up to scale. In dynamic scenes, this ambiguity becomes a serious problem: each reconstructed object has a *different* unknown scale, which means that objects are distorted with respect to each other. Leaving the relative scales unresolved produces unrealistic 3D reconstructions – for example a giant car floating in the sky. In general, only additional constraints, usually on the object trajectories, can resolve the ambiguity [13]. However, if splitting and merging occur, then they connect 3D objects to each other, and in most cases resolve the relative scale without further constraints.

Splitting and merging propagate the object scale in a transitive way: no matter how many splits occur, all objects which stem from the same parent will be scaled correctly with respect to each other (except for possible scale drift over time). Similarly, if an object is the result of an arbitrary number of merging events, then the merge determines the relative scales of all the previously independent components. Together, these two trivial observations are a powerful rule to resolve the scale ambiguity: for example, if object A splits into A_1 and A_2, the relative scale between them will be correct. The same is true for the object B when it splits into B_1 and B_2. When now A_1 merges with B_1, this sets the relative scale not only between A_1 and B_1, but also between A_2 and B_2. If those dependencies are represented as a graph where objects are the nodes and the edges are the dependencies, the scales can be resolved for each object relative to the other objects that are in the same *set of connected components*. If there are more than one set that are not connected, then motion constraint approaches can be used [13] to connect those sets.

E. Affine degenerate reconstruction

An often neglected problem in the sequential SfM literature is the proper handling of nearly affine scenes: if an observed object is small compared to its depth, then its projection with a pinhole camera degenerates from a projective to an affine mapping. Usually this is not a problem when observing a static scene – the 3D structure in most cases has enough depth variation to create perspective affects. In dynamic scenes, however, moving objects are often small compared to the field of view, and hence fall into the degenerate case. The most important practical consequence of this is the mirror symmetry ambiguity: a 3D object and its mirror reflection w.r.t. a plane parallel to the camera's image plane have the same affine projection (the *Necker cube reversal*), which in the multi-view setting gives rise to two different 3D reconstructions. Fig. 1 depicts the phenomenon in 2D.

In practice, perspective reconstruction of distant, relatively small objects will also suffer from the mirror ambiguity, because the perspective effects are outweighed by the measurement noise. In most cases the ambiguity can be resolved if the object is observed for a longer time: in the best case the object approaches the camera, so that the amount of projective distortion grows, but even if that is not the case, the small perspective effects will be corroborated as more views are collected, because the error is systematic.

Unfortunately, sequential SfM pipelines initialize the structure and motion once, and then keep updating that initial guess. If the incorrect, reflected geometry is recovered during initialization, the system gets stuck in a wrong reconstruction: as new frames come in, the camera path is continuously extended by resection from the wrong structure points; bundle adjustment is initialized at the incorrect reconstruction, and converges to one of the associated local minima. One way to solve this issue is a multi-hypothesis approach, where for each new object two symmetric motion models are created and propagated. The decision which of them is correct is cast as a model selection problem in line with the rest of the paper.

F. Practical Considerations

From a theoretical point of view, SfM computation is the same for any rigidly moving structure, and all reconstructed objects are on equal footing. In practice however, foreground objects significantly differ from the dominant background scene : they are much smaller and move with greater freedom, in particular show fast rotations, hence challenge the canonical SfM algorithms. The basic issues are (i) since the objects are small, only few feature points with small aperture angle can be tracked, which makes them susceptible to noise, and (ii) the objects own motion causes more frequent self-occlusion, as well as strong illumination changes (because of changing orientation to the light sources), resulting in shorter feature tracks. As a consequence, accurate feature tracking and the ability to exploit short tracks are crucial factors for successful SfM estimation of such foreground objects.

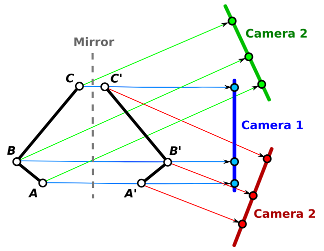

Fig. 1. The mirror symmetry ambiguity of affine projections. The object and its mirror reflection generate identical projections in *camera 1*, which in turn both generate valid orientations for *camera 2*.

This additional difficulty of multibody SfM so far has been ignored, e.g. [2], [11] assume complete tracks spanning the entire sequence. However even without considering a variable number of objects full-length tracks without outliers are all but impossible to obtain for more than a handful of frames. Some approaches alleviate this strict requirement and allow for missing data points [24] or assign them small weights [3], [17]. But these measures do not solve the questions of how to reconstruct from sets of mostly short tracks (in factorization terms, $>90\%$ of missing data), and how to compensate the progressive loss of feature tracks. To date, the only solution is to interleave feature tracking with object reconstruction. This allows the feature set to evolve over time, circumvents the missing data problem, and at every frame provides a maximum of information to guide feature tracking.

III. Implementation Details

A. Overview

In this section, we describe our prototype implementation of the presented ideas. The multibody SfM routines are based on the approach of Schindler et al. [20] due to its flexibility however it is also possible to use other methods, e.g. subspace approaches [25], [28], as long as they are consistently adapted. In the approach we adopted, an over-complete set of motion hypotheses is generated for the entire sequence, and then pruned to an optimal set using model selection. This method is computationally intractable for long sequences, because of the combinatorial explosion of potential motion models. We therefore apply model scoring only in a temporal window rather than as a global batch optimization, replacing parsimony over the entire sequence by local parsimony over the last few frames. The local approach guarantees scalability to long sequences, and intuitively corresponds better to the way humans interpret visual data over time – it is unlikely that information from the distant past influences our decisions about the number of moving objects.

The core SfM engine is based on well-established sequential SfM techniques for static scenes (with calibrated intrinsics), and is applied to each moving object independently. Splitting, merging, and newly appearing objects are handled with task-specific subroutines. Note that although motion segmentation is done locally in short time windows, the underlying global 3D model could still be used to improve the segmentation result for a given frame: the ability to project 3D points reconstructed in distant frames allows one to cover the object more densely and obtain more precise object boundaries.

B. System Details

The core of the system is a multiple model version of the standard sequential SfM pipeline [1], [16]. In order to initialize the algorithm, corner features are detected and tracked through a small number of frames. Three-view motion segmentation is applied (see III-D) to this initialization sequence, yielding an initial segmentation into rigidly moving objects. For each segment, the 3D structure and the camera motion are recovered independently via epipolar geometry decomposition using the 5-point algorithm [12]. As new frames arrive, the existing feature points are tracked and new ones are instantiated, while incrementally computing the new camera pose for each moving object with standard 3-point resection and RANSAC. As explained above, it is important to detect split events as early as possible. Therefore, if a significant number of outliers are detected during the camera resection, this is taken as indication for a possible split and a sub-routine is called which inspects the outliers for new motions, see Sec. III-E. Further sub-routines, with less critical timing, run at regular intervals (in our experiments, at every 5th frame): the *initialization* routine to detect new motions in the set of unexplained feature tracks; the *merging* routine to check the current set of motions for recent merges, see Sec. III-F; and *bundle adjustment* to polish the solution. New tracks, which have no associated 3D point yet, are assigned to the motion they fit best, assuming normally distributed residuals. If all residuals exceed a threshold $T = E\sigma$ (with σ the standard deviation of the noise, and E a positive constant, in our implementation set to 5), then no 3D point is generated, but the track is continued, in case it belongs to a new object.

The mirror symmetry ambiguity is handled as follows: for each new object the two symmetric hypotheses are created, by running camera resection twice, once with positive and once with negative z-coordinates. Both alternatives are then propagated until enough evidence has been collected to resolve the ambiguity[1]—see Sec. III-G. Algorithm 1 gives a high-level description of the system in pseudo-code.

C. The number of motions – a model selection problem

In three situations, namely initialization, splitting, and merging, we are faced with similar questions: given two (or more) competing hypothesis about the scene, decide which is the "best" one, i.e. has the higher likelihood in the light of the observed feature tracks. In principle, the decision is a probability ratio test. However, it is complicated by the fact that the models have different numbers of free parameters, and hence the likelihood of a model is *not* just its goodness-of-fit: the less constrained model has more freedom to adapt to the data, and would always win, leading to over-fitting.

Algorithm 1 Dynamic structure and motion pipeline.

```
1) Instantiate new features, and track all features;
2) If not enough parallax
   (e.g. average feature displacement $< 50$ pixels)
     • goto step 1;
   elseif sufficient parallax and no SfM yet:
     • do initial segmentation and 3D structure computation;
       create symmetric models for small objects.
     • goto step 1;
   else continue;
3) Try to estimate new 3D motion for active models;
   If too many outliers (e.g. $> 25\%$) for a motion model:
     • try to split.
   Instantiate new models if necessary;
4) If waiting period is over and the number of unexplained tracks
   surpasses a threshold (e.g. 10%):
     • try to detect new motion models;
       create symmetric models for small objects.
5) If waiting period is over:
     • try to fuse active motion models greedily;
6) Remove symmetric models, where enough evidence available
   (e.g. $\mathcal{M}_{mirrored}/\mathcal{M}_{correct} > 1.2$,$\mathcal{M}$=model score)
7) goto step 1;
```

This situation is the classical domain of statistical model selection – Occam's razor calls for the simplest possible model, while the fitting error favours more complex models. Model complexity can increase in two ways, either by increasing the number P of free parameters in the model, or by increasing the model's dimension D, and hence reducing the number of constraints per data point. A full treatment is beyond the scope of this article. Intuitively speaking the question is "for N data points, what is the a-priori probability of a model $\mathsf{M}_{D,P}$ with dimension D, and P parameters?"

Here, we will estimate these probabilities with a classical result from model selection theory [18], [21]: if the prior on the model parameters $\mathbf{p}$ is a diffuse Gaussian, then in first order approximation the log-likelihood of a model $\mathsf{M}_{D,P}$ is

$$\mathcal{L}(\mathsf{M}_{D,P}|N) \approx -P\log(D\cdot N) \quad \text{for} \quad N \gg P\ . \qquad (1)$$

The log-likelihood decreases linearly with the number of model parameters, and logarithmically with the number of equations used to estimate the parameters.

In the first-order approximation, not all free parameters of a structure and motion problem depend on all data points [20], [22]: each world point $\mathbf{x}$ depends only on observations of its projections, and each image's motion $\mathbf{c}$ depends only on observations in that image. Hence, expression (1) has to be split into two parts. Let M be the number of images, then

$$\mathcal{L}(\mathsf{M}_{D,P}) = \sum_{i=1}^{M}\mathcal{L}(\mathbf{c}|N_i) + \sum_{i=1}^{N}\mathcal{L}(\mathbf{x}|M_i)\ . \qquad (2)$$

[1] Currently split/merge operations are not allowed before the ambiguity has been resolved.

With equation (2) we can now set up decision functions for initialization, splitting, and merging. As explained above, all decisions are taken locally on short sub-sequences.

D. Initialization

Initialization of new motion models starts from all feature tracks of a predefined length, which have not yet been assigned to any motion. The minimal solution is to use only two frames [22], [25], [27]. Unfortunately, SfM for only two views is notoriously unstable. The other extreme would be a full n-view computation [11], [20]. In practice this is an overkill, since all we need is the number of motions, and coarse parameter estimates to initialize bundle adjustment. What matters is hence mainly the baseline rather than the number of frames, so we apply 3-view segmentation to the first, middle, and last frames of the initialization sequence.

An over-complete set of possible 3-view motions is generated by random sampling and inlier/outlier separation with the TSSE estimator [26], and the best subset is selected with model selection. Let the number of correspondences on a candidate model A be N_{A}, the sum of (squared) residuals E_{A}, and the standard deviation of the residuals σ. Outliers are assumed to be uniformly distributed in the tracking window of size $(w{\times}w)$, so the (log-)likelihood that all correspondences are outliers is $2\mathcal{L}_{\not\mathsf{A}} = -6N_{\mathsf{A}}\log(w^2)$. Conversely, the likelihood of model A is

$$2\mathcal{L}_{\mathsf{A}} = -6N_{\mathsf{A}}\log(2\pi\sigma^2) - \sigma^{-2}E_{\mathsf{A}} - 3N_{\mathsf{A}}\log(6) - 11\log(2N_{\mathsf{A}}) \;,$$

assuming that all points are visible in all 3 frames. The first two terms are the likelihood of the correspondences (assuming zero-mean normally distributed residuals). The third and fourth term are the likelihood (2) for N_{A} structure points each estimated from 6 observations, and $(3\times 6-7) = 11$ motion parameters each estimated from $2N_{\mathsf{A}}$ observations. The benefit of using model A is $\mathcal{D}_{\mathsf{A}} = \mathcal{L}_{\mathsf{A}} - \mathcal{L}_{\not\mathsf{A}}$. If this value is positive, then model A is a more likely explanation of the data than the all-outlier assumption. However, we are given many candidate models $\{\mathsf{A},\mathsf{B}..\mathsf{N}\}$, and it is not known how many are required to explain the data. This results in a quadratic Boolean optimization problem

$$\max_{\mathbf{m}} \left(\mathbf{m}^{\mathsf{T}} \begin{bmatrix} 2\mathcal{D}_{\mathsf{A}} & -\mathcal{D}_{\mathsf{A}\cap\mathsf{B}} & \dots & -\mathcal{D}_{\mathsf{A}\cap\mathsf{N}} \\ -\mathcal{D}_{\mathsf{A}\cap\mathsf{B}} & 2\mathcal{D}_{\mathsf{B}} & \dots & -\mathcal{D}_{\mathsf{B}\cap\mathsf{N}} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathcal{D}_{\mathsf{A}\cap\mathsf{N}} & -\mathcal{D}_{\mathsf{B}\cap\mathsf{N}} & \dots & 2\mathcal{D}_{\mathsf{N}} \end{bmatrix} \mathbf{m} \right) \;, \qquad (3)$$

where $\mathbf{m}$ is an boolean vector which indicates whether a candidate is used ($m_i = 1$) or discarded ($m_i = 0$) in a certain explanation. The off-diagonal terms $\mathcal{D}_{\mathsf{A}\cap\mathsf{B}}$ handle the interactions between candidates, making sure each correspondence is assigned to only one motion. For details, see [10], [20].

E. Splitting

Splitting events show up as a sharp increase in the outlier ratio when computing the camera parameters for the parent object. Consequently, new models are searched in those outliers. As stated before, splitting should be detected instantly. Since 3D structure is already known at this point, the decision can be made on a single frame by inspecting resection results. Considering the possibilities that an object can split into more than two children, and that some feature tracks may be true outliers, a recover-and-select approach is adopted, similar to the initialization. Multiple motion hypotheses are generated by random resection, then an optimal set is selected to explain the 3D-2D correspondences. With the same symbols as above, the likelihood of all outliers is $2\mathcal{L}_{\not\mathsf{A}} = -2N_{\mathsf{A}}\log(w^2)$, and the likelihood of a new motion A is

$$2\mathcal{L}_{\mathsf{A}} = -2N_{\mathsf{A}}\log(2\pi\sigma^2) - \frac{1}{\sigma^2}E_{\mathsf{A}} - 6\log(2N_{\mathsf{A}}) - 3\frac{N_{\mathsf{A}}}{F_{\mathsf{A}}}\log(2F_{\mathsf{A}}).$$

The first and second term are the likelihood that the points *in the new frame* were generated by 3D structure and motion, and the third term is the likelihood of the new motion. The fourth term is an estimate of how much the likelihood of the 3D structure changes due to the extra motion: obviously, the structure does not change during resection; however, once the new model has been accepted, the structure and motion will be re-estimated using also the new image, and this will slightly change the likelihood. This contribution should theoretically be computed separately for each 3D point, depending on its track length. We strike a practical compromise and count the average number of frames F_{A}, during which a feature point on the object remains visible, and equally divide the coding length between these frames.

Again, the benefits of using model A are $\mathcal{D}_{\mathsf{A}} = \mathcal{L}_{\mathsf{A}} - \mathcal{L}_{\not\mathsf{A}}$, and the competition between the candidates leads to a quadratic Boolean problem (3).

F. Merging

Merging amounts to checking whether two objects A and B still move independently, or whether they got attached to each other and now move as one rigid body C (as explained in section II-C, only two objects need to be considered, because merging can be done iteratively). Both hypotheses are evaluated, and the one with the higher likelihood is preferred. The likelihood of the two independent motions (again, for three frames, and assuming that all structure points are seen in all three images) is

$$\begin{aligned} 2\mathcal{L}_{\mathsf{A},\mathsf{B}} = & -\sigma^{-2}(E_{\mathsf{A}} + E_{\mathsf{B}}) - 11\big(\log(2N_{\mathsf{A}}) + \log(2N_{\mathsf{B}})\big) \\ & - 3(N_{\mathsf{A}} + N_{\mathsf{B}})\log(6) - 6(N_{\mathsf{A}} + N_{\mathsf{B}})\log(2\pi\sigma^2) \;, \end{aligned}$$

and the likelihood for a single joint motion is

$$2\mathcal{L}_{\mathsf{C}} = -\sigma^{-2}E_{\mathsf{C}} - 11\log(2N_{\mathsf{C}}) - 3N_{\mathsf{C}}\log(6) - 6N_{\mathsf{C}}\log(2\pi\sigma^2) \,.$$

Note, the last two terms cancel out because $N_{\mathsf{C}} = N_{\mathsf{A}} + N_{\mathsf{B}}$.

G. Mirror Symmetries

When choosing between the two hypotheses generated by the mirror-symmetry ambiguity, we are again faced with the question: given two possible solutions, which one is more likely? Again, we need to take into account not only the residuals, but also the power of the model to describe the image data – otherwise a wrong model with fewer inliers will win over a correct one, which has a larger cumulative sum of residuals for the simple reason that it has more residuals.

This time, we cannot make the simplifying assumption that the two models have the same sets of corresponding points. Since SfM estimation and feature tracking are alternated

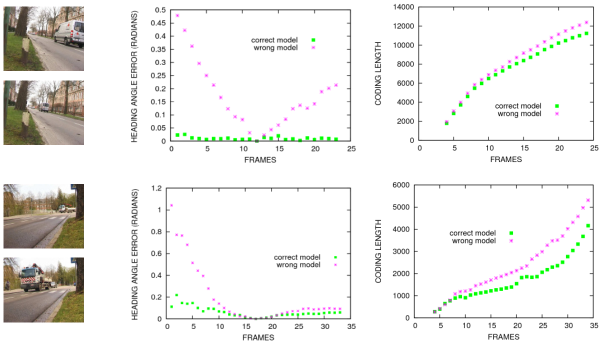

Fig. 2. *Left: The* van *(top) and* lorry *(bottom) sequences. Middle: errors of the heading angle for the two symmetric reconstructions w.r.t. the middle frame. As expected, the correct model gives lower errors. Right: Negative log-likelihoods for both reconstructions. The data-driven scores allow one to pick the correct reconstruction without trajectory constraints.*

independently for the two models, their inlier sets will differ to some degree. This can be handled by trying to explain *all* data as a combination of the respective model and a set of uniformly distributed outliers.

The likelihood of each of the two candidates after observing F frames is given by

$$2\mathcal{M} = -\sum_{i=1}^{F} N_i \log 2\pi\sigma^2 - \frac{1}{\sigma^2}\sum_{i=1}^{F}\sum_{j=1}^{N_i} r_{ij}^2 + 3\sum_{j=1}^{N}\log(2F_j)$$
$$-\,(6-7/F)\sum_{i=1}^{F}\log(2N_i) - \log(w^2)\sum_{i=1}^{F} N_i^{outlier}\ ,$$

where N is the total number of structure points explained by a model, N_i is the number of those points visible in a certain frame i, and F_j is the number of frames in which a certain point j is visible. Again, the first and second term are the likelihood of the residuals, the third and fourth terms are the likelihood of the N structure points and the $(6F-7)$ motion parameters, and the fifth term is the likelihood that the remaining image points are outliers.

For objects which suffer from the mirror-symmetry ambiguity, the log-likelihood for the two symmetric reconstructions will initially be similar. Over time the likelihood of the *wrong* model will decrease faster, since it will encounter higher reprojection errors, lower inlier numbers, or both. When the difference becomes significant, the better model is selected and the other one is discarded. The difference is checked by simply thresholding the ratio between the two model scores, however one could imagine more complex tests which take into account the temporal evolution.

IV. Experiments

A. Mirror Symmetries

In order to illustrate the mirror-symmetry ambiguity on a practical example, and to test the validity of our way to resolve it, we have conducted two basic experiments with simple scenarios: a static camera observing a distant moving object with (at least approximately) known type of motion. The knowledge of the motion characteristics is important as a measure to judge the quality of the obtained reconstructions. In our experiments, we exploit the fact that the motion of most land vehicles (e.g. cars, bicycles) is non-holonomic. In the context of multibody SfM, Ozden *et. al.* have called this the "heading constraint", and used it to resolve the unknown relative object scales [14], [15]. Formally, the constraint is given by the relation $\mathbf{D}_{\mathrm{o}}^{ij}\mathbf{v}_{\mathrm{o}}^{i} = \mathbf{v}_{\mathrm{o}}^{j}$, where $\mathbf{D}_{\mathrm{o}}^{ij}$ is the rotation of the *object* from frame i to frame j, and $\mathbf{v}_{\mathrm{o}}^{i}$ is the unit tangent vector to the object's trajectory at frame i (estimated in practice as a central difference). Geometrically, the constraint simply means that the direction of the motion vector is fixed with respect to the object, i.e. the vehicle can only be displaced in its heading direction.

In the following experiments, we will use the heading constraint as "ground truth" to judge the quality of the reconstruction *without using the constraint*. As a side note, it would of course be possible to use the heading constraint itself (or other motion constraints) to resolve the mirror ambiguity since the final reconstruction for the wrong symmetry will be inferior due to accumulated perspective affects. However this requires that the observed object undergoes a constrained motion, and that the type of constraint – and thus the object category – is known. In contrast, the presented model scoring approach is purely geometric, and thus completely general.

Figure 2(top left) shows samples from a sequence of 25 frames, in which a van moves away from the camera along a linear trajectory. The object moves over a large depth range, hence the perspective distortion over the entire sequence is large enough to resolve the ambiguity. However this is no longer the case, if only the first few frames are considered. In the experiment, *both* the best model found by 3-view motion segmentation *and* its symmetric counterpart are initialized and propagated through the sequence. Initially, the two models are almost indistinguishable (similar reprojection errors even at sub-pixel precision). However, the wrong model deteriorates over time, until the correct one is a clearly better explanation of the feature tracks. In Fig. 2(top middle) we plot for each frame the angle between the van's heading direction *estimated* from its trajectory, and its *true* heading direction according to the heading constraint (relative to frame 12). The correct model obeys the constraint, whereas the heading of the wrong model gradually drifts, which would mean that the van is skidding. The experiment proves our claim that structure-from-motion lacks the ability to recover from the wrong initialization, and if initialized with the mirrored structure, will deliver distorted and unrealistic reconstructions, even when enough perspective distortion has become available. Figure 2(top right) shows the evolution of the two models' coding length estimated with Eq. (III-G), starting at frame 4, after initialization. As more frames come in, the coding length of the wrong reconstruction grows faster, until the hypotheses are clearly distinguishable.

Figure 2(bottom left) shows sample frames the second experiment, with a total of 35 frames. In this case a distant lorry turns left towards the camera, and then approaches it in a straight line. Since the lorry turns in the first few frames, there is abundant parallax to initialize the 3D structure, still it is too far from the camera to resolve the mirror ambiguity. Again, Fig. 2(bottom middle) shows the error in the heading angle, and Fig. 2(bottom right) shows the evolution of the model coding length. As in the previous example, the two scores are initially indistinguishable, but the correct reconstruction quickly becomes obvious.

B. System-level Experiments

Garden Sequence (250 frames, see Fig. 3) starts with a person who is carrying a cardboard box, while the camera is also moving. Later another person enters from the left and leaves the scene, and finally the remaining person merges with the background. The segmentation results in Fig. 3 demonstrate that the motion detection and merging operations succeeded. The sequence illustrates another advantage of online SfM, namely recovery from failure: after few frames, the person on the right moves non-rigidly and is lost, but is immediately redetected once the motion becomes rigid.

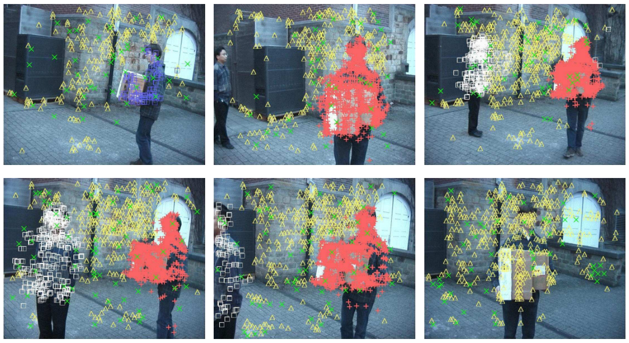

Fig. 3. Garden sequence results. Operations: Initial segmentation, new motion detection, merging. Note how the person is re-detected immediately after being lost due to non-rigid motion. Green crosses are outliers.

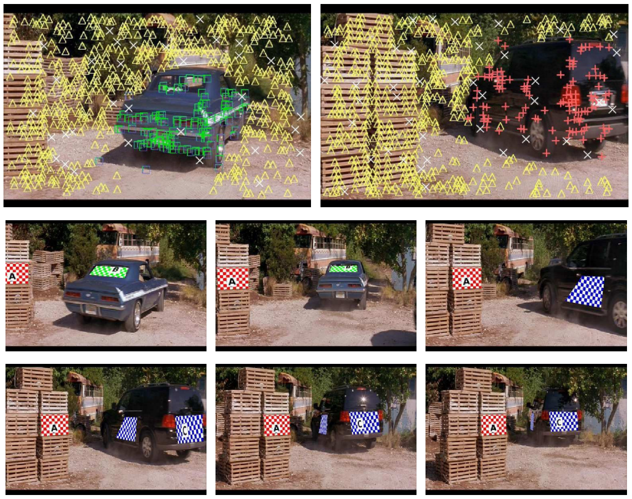

Fig. 4. Car sequence. Top: segmentation results (operations: initial segmentation, motion termination, new motion detection). Bottom: 3D augmentation.

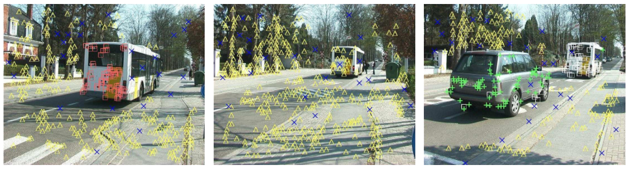

Fig. 5. Bus sequence results. Operations: initial segmentation, merging, splitting, new motion detection. Blue crosses are outliers.

Car Sequence. One of the successful application areas for SfM algorithms is movie post-production where the original image sequence is “augmented” with artificial 3D objects. One opportunity that comes with the proposed technique is the possibility of augmenting not only the background but also the moving foreground objects, while at the same time preserving depth consistency, which is usually a problem due to relative scale ambiguities between the components. We have tested our technique on a 107 frames long sequence taken from the movie “2 Fast 2 Furious” (see Fig. 4). There is no merging or splitting, the sequence serves mainly as an example for challenging feature tracks generated by freely moving shiny objects. The experimental results show the effectiveness of simultaneous segmentation and reconstruction.

Fig. 4 shows two sample frames from the segmentation results. To check the accuracy of 3D reconstruction, we have augmented the sequence with artificial objects. Their stability through the sequence demonstrates that SfM was successful.

Bus-stop Sequence (170 frames) demonstrates the full range of system capabilities (see Fig. 5). A bus enters a scene while the handheld camera moves forward. The bus stops at the bus-stop, then pulls out again, while a car appears behind the bus. Both vehicles follow nearly straight paths. Lighting conditions are challenging: strong specular reflections occur on the car windows, and tree shadows constantly wipe out features on the moving vehicles.

Fig. 5 shows sample frames from the segmentation results, where the initial detection of the moving vehicles, the merging of the bus with the background, and its split from the background are demonstrated. To check the 3D reconstruction accuracy, we have also augmented this sequence with artificial objects (see Fig. 6). Note that after merging the pattern which attached to the bus is rendered with the camera matrices computed for the background, showing that the relative scale has been resolved correctly.

Two views of the 3D point clouds for the car, the bus, and their paths are also shown. The first one is more or less a top view and the second one a side view. Note that when the bus merges with the background, their relative scales are automatically solved. The car does not merge with the background or the bus, so linear motion had to be used as additional constraint to set its scale [13]. The 3D tracks of both vehicles appears visually correct.

Increased accuracy due to merging. One benefit of correct merging is a higher accuracy of SfM estimation, since fewer parameters have to be estimated from the same data. Without ground-truth it is hard to conclusively demonstrate this property, but known geometric relationships in the scene can provide some evidence. We exploit line pairs, which are known to be orthogonal, and measure how much the reconstructed angle differs from the expected 90° to check the correctness of the estimated 3D geometry.

The proposed algorithm was run on the bus-stop sequence both with and without the merging routine, and 4 line pairs on the bus were checked. The deviations from the expected right angles were $\Delta = \{7°, 15°, 9°, 7°\}$ with merging disabled, compared to $\Delta = \{3°, 8°, 5°, 2°\}$ with merging, supporting the claim that correct merging improves reconstruction accuracy.

Runtime. Experiments are performed on a 2GHz standard CPU with an unoptimized C++/Matlab implementation. Pro-

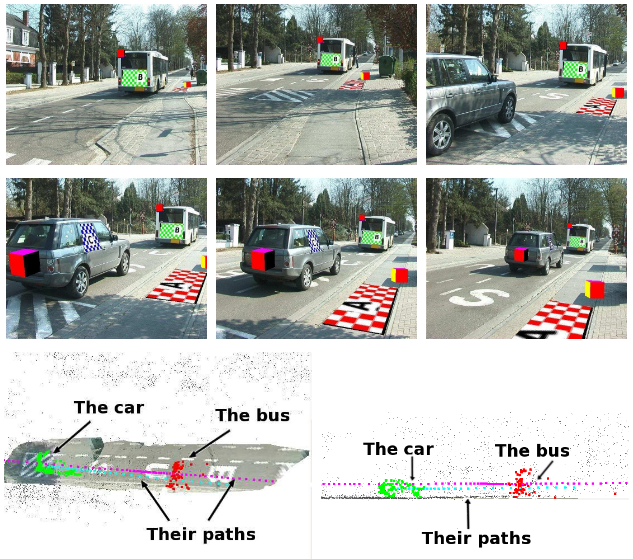

Fig. 6. Bus sequence. Top: 3D augmentation. Bottom: Top and side view of the 3D reconstruction.

cessing standard video resolution (720x576) currently takes a bit less than 1 minute per frame. The most costly single step is the initial motion segmentation, and computation times increase with growing sequence length due to periodic global bundle adjustment.

V. Conclusion

Compared to the large volume of research on practical systems for static structure from motion, 3D reconstruction of dynamic scenes has so far been investigated mainly theoretically for short, simple image sequences. We have tried to fill this gap, by identifying which components are required in a general and efficient SfM framework for dynamic scenes. We have identified and discussed several subtle issues of large-scale dynamic SfM, and have proposed a novel framework to solve the task. Model selection techniques are deployed to detect changes in the number of independently moving objects. The advantages of such an approach have been demonstrated, and successful experiments have been presented.

There are still several open issues, which need to be addressed in order to reach the maturity of static SfM systems. At the system level, a prominent bottleneck is the regular bundle adjustment of all past frames. A possible solution would be to limit the periodic bundle adjustment to a short time window, like in visual SLAM systems. This should still be accurate enough to ensure correct results, since the long-term drift does not impair the local accuracy. A final global adjustment could still be performed offline for the entire sequence. A more complex system-level question is how to automatically adapt the (currently fixed) system parameters in response to changes of the environment (e.g. lighting, camera speed, ...).

At a more conceptual level, an open research question is the combination with object detection and/or categorization. Multibody SfM estimation could be a valuable source of information for these tasks, since it provides (sparse) 3D scene structure, object motion, and motion segmentation. Conversely, SfM estimation could benefit from the semantic information extracted by an object detector, for example through more detailed object segmentation, object-specific motion constraints, or simply more reliable detection of new objects in the scene. Interleaving SfM and detection in an online system offers many interesting possibilities.

The proposed model-selection approach is not restricted to any specific camera model, hence an obvious extension of the presented ideas is to apply multibody SfM also to more powerful cameras (omnidirectional, multi-camera setups, etc.), which are widely used in the context of navigation and robotics. A more fundamental extension would be to drop the requirement for piecewise rigid scenes and also model articulated and non-rigid objects.

References

[1] P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *ECCV*, 1996.

[2] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *ICCV*, 1995.

[3] T. Darrel, A. Azerbayejani, and P. Pentland. Segmentation of rigidly moving objects using multiple Kalman filters. MIT Media Lab, 1994.

[4] O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The geometry of multiple images*. MIT Press, 2001.

[5] A. W. Fitzgibbon and A. Zisserman. Multibody structure and motion: 3D reconstruction of independently moving objects. In *ECCV*, 2000.

[6] M. Han and T. Kanade. Reconstruction of a scene with multiple linearly moving objects. In *CVPR*, 2000.

[7] M. Han and T. Kanade. Multiple motion scene reconstruction from uncalibrated views. In *ICCV*, 2003.

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

[9] K. Kanatani. Motion segmentation by subspace separation and model selection. In *ICCV*, 2001.

[10] A. Leonardis, A. Gupta, and A. Bajcsy. Segmentation of range images as the search for geometric parameteric models. *IJCV*, 14, 1995.

[11] T. Li, V. Kallem, D. Singaraju, and R. Vidal. Projective factorization of multiple rigid-body motions. In *CVPR*, 2007.

[12] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *CVPR'04*.

[13] K. Ozden, K. Cornelis, L. Van Eycken, and L. Van Gool. Reconstructing 3D trajectories of independently moving objects using generic constraints. *CVIU*, 96(3), 2004.

[14] K. E. Ozden, K. Cornelis, L. Van Eycken, and L. Van Gool. Reconstructing 3d independent motions using non-accidentalness. In *CVPR*, 2004.

[15] K. E. Ozden and L. Van Gool. Background recognition in dynamic scenes with motion constraints. In *CVPR*, 2005.

[16] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *IJCV*, 59(3), 2004.

[17] G. Qian, R. Chellappa, and Q. Zheng. Bayesian algorithms for simultaneous structure from motion estimation of multiple independently moving objects. In *IEEE TIP*, volume 15, 2005.

[18] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[19] K. Schindler and D. Suter. Two-view multibody structure-and-motion with outliers through model selection. *IEEE PAMI*, 28, 2006.

[20] K. Schindler, D. Suter, and H. Wang. A model selection framework for multibody structure-and-motion of image sequences. *IJCV*, 2008.

[21] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[22] P. H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *IJCV*, 50(1), 2002.

[23] R. Tron and R. Vidal. A benchmark for the comparison of 3D motion segmentation algorithms. In *CVPR*, 2007.

[24] R. Vidal and R. Hartley. Motion segmentation with missing data using powerfactorization and GPCA. In *CVPR*, 2004.

[25] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multibody structure from motion. *IJCV*, 68(1), 2006.

[26] H. Wang and D. Suter. Robust fitting by adaptove-scale residual consensus. In *ECCV*, 2008.

[27] L. Wolf and A. Shashua. On projection matrices $P^k \rightarrow P^2, k = 3, \dots, 6$, and their applications in computer vision. In *ICCV*, 2001.

[28] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, 2006.