Int J Comput Vis (2008) 79: 159–177
DOI 10.1007/s11263-007-0111-7

A Model-Selection Framework for Multibody Structure-and-Motion of Image Sequences

Konrad Schindler · David Suter · Hanzi Wang

Received: 17 March 2006 / Accepted: 19 November 2007 / Published online: 11 December 2007

Abstract Given an image sequence of a scene consisting of multiple rigidly moving objects, multi-body structure-and-motion (MSaM) is the task to segment the image feature tracks into the different rigid objects and compute the multiple-view geometry of each object. We present a framework for multibody structure-and-motion based on model selection. In a recover-and-select procedure, a redundant set of hypothetical scene motions is generated. Each subset of this pool of motion candidates is regarded as a possible explanation of the image feature tracks, and the most likely explanation is selected with model selection. The framework is generic and can be used with any parametric camera model, or with a combination of different models. It can deal with sets of correspondences, which change over time, and it is robust to realistic amounts of outliers. The framework is demonstrated for different camera and scene models.

Keywords Multibody structure-and-motion · 3D motion segmentation · Model selection

1 Introduction

Structure-and-motion recovery from images, using the image motion as the only source of information, has been extensively studied in the last decade. For the case of static scenes, the problem of fitting a 3D scene compatible with the images is well understood and essentially solved (Hartley and Zisserman 2000; Faugeras et al. 2001; Ma et al. 2003). Soon after the main structure-and-motion theory had been established, researchers turned to the more challenging case of *dynamic* scenes, where the segmentation into independently moving objects and the motion estimation for each object have to be solved simultaneously (see Fig. 1). Even in the case of rigidly moving scene parts, which we will call *multibody structure-and-motion* or MSaM, the geometric properties of dynamic scenes turned out to be non-trivial. The solutions available so far can be broadly classified into algebraic methods, which exploit algebraic constraints satisfied by all scene objects, even though they move relative to each other, and non-algebraic methods, which essentially combine rigid structure-and-motion with segmentation. Algebraic solutions based on matrix factorization exist for the case of two views (Wolf and Shashua 2001; Vidal et al. 2002; Vidal and Ma 2004), for multiple affine views (Costeira and Kanade 1995; Vidal and Hartley 2004), and for multiple affine views of linearly moving points (Han and Kanade 2000). Recently, an iterative algebraic solution for multiple perspective views has also been presented (Li et al. 2007). Non-algebraic approaches, which detect different motions iteratively, have been presented for 2 views (Irani and Anandan 1998; Torr 1998; Tong et al. 2004), and for multiple perspective views (Fitzgibbon and Zisserman 2000). A non-algebraic approach, which recovers all motions concurrently, has been developed for two views

Most of the presented research was carried out while all three authors were at Monash University.

K. Schindler (✉)
Computer Vision Laboratory, Eidgenössische Technische Hochschule, Sternwartstr. 7, 8092 Zürich, Switzerland
e-mail: konrads@vision.ee.ethz.ch

D. Suter
Electrical and Computer Systems Engineering,
Monash University, 3800 Clayton, Australia

H. Wang
Department of Computer Science, Johns Hopkins University,
Baltimore, 21218 MD, USA

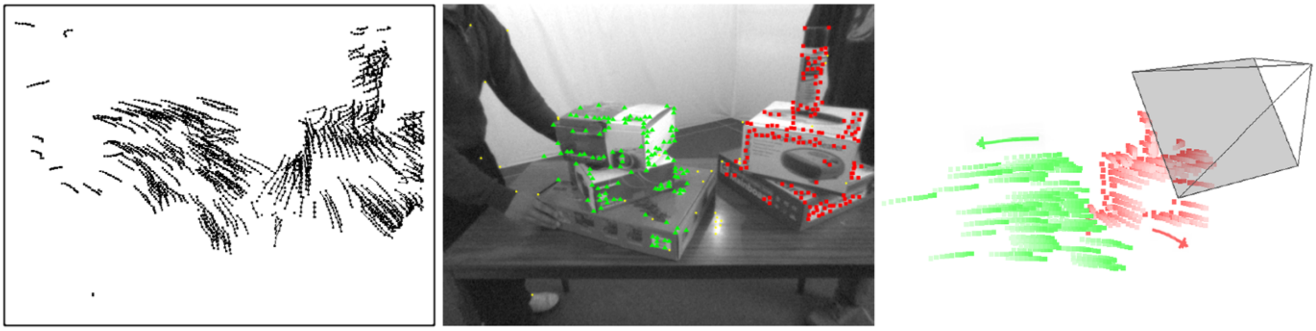


Fig. 1 The multibody structure-and-motion problem. From a number of tracked correspondences, estimate the number of moving objects, the segmentation into different objects, and the 3D motion of the objects

in Schindler and Suter (2005), and extended to multiple perspective views in Schindler et al. (2006).

Here, we present a generic model-selection framework for multibody structure-and-motion with any (parametric) camera model. The setting is the following: a scene with an unknown (small) number of rigidly moving objects is recorded with a camera (the camera can be static or moving as well, because only motion relative to the camera matters). Image correspondences are tracked through the captured sequence with a feature point tracker. Points may be lost (e.g., due to occlusion), new points may be detected to replace the lost ones, and the set of point tracks may contain outliers, which have been wrongly matched between frames. Furthermore, the number of motions present may vary throughout the sequence, e.g. when an object leaves the field of view.

The presented framework is a generic way to solve the stated problem, and contains all the MSaM problems listed earlier as special cases. Its main strength is that it can handle arbitrary parametric projection models,[1] and is robust to large amounts of outliers and missing data. The missing data problem does not arise in our method, since it is not based on factorization, while the resistance to outliers is due to the fact that model selection explicitly includes a model for unexplained tracks so as to handle them correctly. An outline of the complete process for multibody structure-and-motion recovery is given in Algorithm 1.

We first introduce the generic model-selection framework in Sect. 2, then describe a way to generate candidate motions in Sect. 3. Experimental results and a discussion are given in Sect. 4, and a conclusion sums up the paper and points out limitations and possible extensions (Sect. 5).

Algorithm 1 Outline of n-view multibody structure-and-motion method.

```
1. Tracking: track feature points through the sequence
2. Generating candidates: for each pair of consecutive frames $(j, j+1)$
   (a) Sample a set of two-view motions $\{\mathcal{Q}_i^j\}$
   (b) For each $\mathcal{Q}_i^j$, estimate inlier set and standard deviation
   (c) Cluster $\{\mathcal{Q}_i^j\}$ and re-estimate representatives $\{\overline{\mathcal{Q}}_i^j\}$ for each cluster
3. Motion linking: recursively link $\{\overline{\mathcal{Q}}_i^j\}$ through frames to obtain candidate motions $\{\mathcal{D}_k\}$
4. Model selection: pick the best subset from $\{\mathcal{D}_k\}$
   (a) build the codelength/likelihood function $\mathcal{D}(\mathbf{b})$ for the candidate motions
   (b) maximize $\mathcal{D}(\mathbf{b})$ over the index vector $\mathbf{b}$ to determine the best subset
5. Postprocessing: enforce temporal consistency to clean up segmentation
6. (optional) Triangulation: triangulate 3D coordinates of feature points
```

2 Multibody Structure-and-Motion as Model Selection

Model selection is a branch of statistics and information theory, which is concerned with finding the right parametric model for a given set of data. This is accomplished by fitting different models to the data, and devising a scoring function, which assigns a scalar to each of the models. The model with the best score is then selected as the most appropriate one. Depending on whether the interpretation is probabilistic or information-theoretic, models are scored according to their probability $\mathcal{P}$ conditioned on the data, or according to their coding length $\mathcal{L}$, which is thought of as a measure of simplicity. The two aims of maximum probability or minimum codelength are essentially equivalent, since by Shannon's theorem they are related by $\mathcal{L} \sim -\log(\mathcal{P})$ (Shannon 1948). Different approximations of the model coding length

[1] In practice, the method is not suitable for models with a large number of parameters, since candidate generation relies on sampling with the minimal solution.

and different priors have led to a host of criteria being proposed.

One of the most popular criteria is Akaike's *An Information Criterion* (AIC) (Akaike 1973), which aims to minimize the residual of yet unobserved data by minimizing the expected entropy. It differs from the rest of the criteria in that the penalty for model complexity is independent of the number of data. Unfortunately, AIC has a well-documented tendency to overfit (Leontaritis and Billings 1987; Torr 2000; Kanatani 2004).

Other popular model selection criteria are: Wallace's *Minimum Message Length* (MML) (Wallace and Boulton 1968; Wallace and Freeman 1987), probably the earliest rigorous approach to minimizing the coding length. Rissanen's very similar, but independently developed *Minimum Description Length* (MDL) (Rissanen 1978, 1984). The proponents of MDL aim to circumvent the problem of choosing a prior distribution for the candidate models. From a Bayesian standpoint, this is an impossible endeavor, and only amounts to always using the same implicit prior. Consequently, the *Bayesian Information Criterion* (BIC) of Schwarz (1978), which is a first-order approximation of the posterior probability, assumes a diffuse Gaussian prior. Despite the somewhat different intentions this leads to a criterion very similar to MML/MDL. Furthermore, there are variants of AIC called CAIC and CAICF, which do take into account the volume of data in order to counter overfitting (Bozdogan 1987), and are also similar to MDL/MML. Although proponents of the different schools continue to argue in favor of one or the other criterion, all these criteria are very similar and in practice mostly give the same results.

Assume for the moment that we already have a redundant set $\mathcal{S}$ of candidate object motions, which contains the correct motions to explain our image measurements, but also many other spurious motions (generating such a candidate set will be treated in Sect. 3). Our goal is to prune all spurious motions, so that we end up with a minimal set to explain the image feature tracks. This can be viewed as a model selection problem: from the combinatorial set of explanations given by all subsets of $\mathcal{S}$, select the most appropriate one. We will now derive a selection criterion for this problem. Given the small practical differences (Kverh and Leonardis 2004), we see no need to use the higher-order approximations of MML or CAIC. We will therefore borrow mostly from the derivations of BIC and MDL. Since for our purposes the two are equivalent, we do not need to commit to one or the other school of thought. Most of the derivation follows an information-theoretic approach based on codelengths, while we adopt Torr's extension of Schwarz' BIC approximation (Torr 2000) to estimate the coding length of the structure-and-motion.

Note that such a hypothesize-and-select approach can deal with model *overlap*, meaning the case, where a significant number of data points have low residuals in more than one potential motion. This is important in situations, where a wrong motion model finds strong support in the data: then, a decision has to be made to explain a set of points either with the wrong motion model with higher residuals (and possibly some outliers), or with two or more independent motions with lower residuals, but fewer points. In such cases, the correct decision can only be taken, if all candidate motions, and their overlap, are considered in a joint optimization. Iteratively searching for the strongest candidate will commit to the wrong motion and fail.[2]

2.1 Coding the Data

Given is a sequence of F images, through which feature points have been tracked. The tracker may lose points (e.g., due to occlusions) and replace them by detecting new ones. The search area for the tracker is usually restricted to a window of size $w \times w$ around a point's position in the previous image (for unrestricted matching, w is the image size).

Now let us assume that over a part of the sequence, a rigid object $\mathcal{M}$ has moved through the scene. The total number of tracked 3D points on the rigid object is N, of which only N_i are visible in each frame $i \in \{1, \dots, F\}$ (if the object is not visible in all frames, then $N_i = 0$ for some frames). Conversely, each 3D point $\mathbf{x}_j$ is only seen in F_j of the F frames. If we want to code these points without using the scene structure, we need to specify their coordinates within the search window for each frame. Assuming uniform density over the search area, the average coding length for one point is the negative log-likelihood of a 2D uniform distribution, and the total coding length for the image tracks is

$$\mathcal{L}_{\text{img}} = \sum_{i=1}^{F} N_i \log \frac{1}{w^2}. \tag{1}$$

On the other hand, if the scene structure-and-motion is known, then the approximate coordinates of each point can be constructed by projecting the corresponding scene point $\mathbf{x}_j$ to an image point $\tilde{\mathbf{v}}_{ij} = p_i(\mathbf{x}_j)$, and only the residual $r_{ij} = |\mathbf{v}_{ij} - \tilde{\mathbf{v}}_{ij}|$ with respect to this location has to be coded. Assuming that the residuals have zero-mean normal distribution with standard deviation $\sigma_x = \sigma_y = \sigma$, the codelength for a point is the negative log-likelihood of the 2D Gaussian, and we get

$$\mathcal{L}_{\text{err}} = \frac{1}{2\sigma^2} \sum_{i=1}^{F} \sum_{j=1}^{N_i} r_{ij}^2 + \sum_{i=1}^{F} N_i \log(2\pi\sigma^2). \tag{2}$$

[2] If there is only little overlap, iterative search will however be faster, because only the most dominant motion has to be detected in each iteration step, which is much easier than finding good candidates for *all* present motions.

2.2 Coding the Structure-and-Motion

However, if the image points shall be coded as projections of the scene structure, we also have to encode the structure-and-motion of the scene. The structure consists of N scene points, each with λ_D coordinates $\mathbf{x}_i$, and the motion consists of F frames, each with λ_C camera parameters $\mathbf{c}_i$. Both are estimated from $\lambda_V = 2\sum_{j=1}^{F} N_j$ image points $\mathbf{v}_i$, up to an ambiguity of the global coordinate frame with λ_G degrees of freedom. For convenience of notation, we collect all $\mathbf{c}_i$ and $\mathbf{x}_i$ in a vector $\mathbf{S}$, and all $\mathbf{v}_i$ in a vector $\mathbf{V}$. The structure-and-motion is estimated from the image points by minimizing an error function $E(\mathbf{S}, \mathbf{V})$:

$$\widehat{\mathbf{S}} = \arg\min E(\mathbf{S}, \mathbf{V}). \tag{3}$$

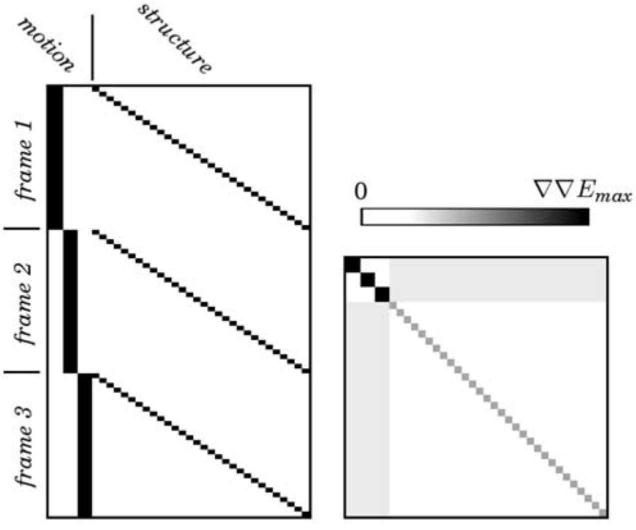
If the prior to the structure-and-motion parameters $\widehat{\mathbf{S}}$ is assumed to be a very diffuse Gaussian, then the coding length $\mathcal{L}_{\text{sam}}$ of the parameters is approximately the logarithm of the determinant of the Hessian of E at $\widehat{\mathbf{S}}$, $\mathcal{L}_{\text{sam}} \approx \frac{1}{2}\log|\ddot{E}|$. This is a classical result from model selection theory (Schwarz 1978; Ripley 1996). The coding length $\mathcal{L}_{\text{sam}}$ is estimated with an asymptotic result for large samples:

$$\log|\ddot{E}| \approx \lambda_S \log(\lambda_V) \quad \text{for } \lambda_V \gg \lambda_S, \tag{4}$$

where λ_S is the number of free parameters, and λ_V is the number of data to estimate the parameters. This last approximation is based on the assumption that all parameters are computed from all data points, and it has been shown to be an overly crude approximation for the structure-and-motion problem (Torr 1998).

For the case of two-view structure-and-motion, Torr has developed the GBIC approximation (Torr 1998, 2000), which uses the specific structure of the estimation problem to devise a better approximation for $\mathcal{L}_{\text{sam}}$. His idea can be extended to the multi-view case: the error function E for the multi-view bundle adjustment case has a near block-diagonal form. A camera is only dependent on the image points seen by that particular camera, and a structure point is only dependent on all projections of that particular structure point, while the correlations between the two are comparatively small. This leads to the following structure of the Hessian (an illustration is given in Fig. 2):

$$\ddot{E} = \begin{bmatrix} \ddot{E}_{CC} & \ddot{E}_{XC} \\ \ddot{E}_{CX} & \ddot{E}_{XX} \end{bmatrix}$$
$$= \begin{bmatrix} \ddot{E}_{c_1c_1} & 0 & \cdots & \ddot{E}_{c_1x_1} & \ddot{E}_{c_1x_2} & \cdots \\ 0 & \ddot{E}_{c_2c_2} & \cdots & \ddot{E}_{c_2x_1} & \ddot{E}_{c_2x_2} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ \ddot{E}_{c_1x_1} & \ddot{E}_{c_2x_1} & \cdots & \ddot{E}_{x_1x_1} & 0 & \cdots \\ \ddot{E}_{c_1x_2} & \ddot{E}_{c_2x_2} & \cdots & 0 & \ddot{E}_{x_2x_2} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{5}$$



(a) Equation system (b) Hessian

Fig. 2 Schematic illustration of structure-and-motion estimation. A camera is only estimated from the points it sees, and a scene point is only estimated from its projections. The (linearized) error function is sparse, and its Hessian is near block-diagonal. *White* denotes 0-elements, *darker color* means larger value

The elements of $\ddot{E}_{CX} = (\ddot{E}_{XC})^{\mathsf{T}}$ are small compared to the elements of $\ddot{E}_{CC}$. Hence, their influence can be neglected, and the determinant factors into the sub-determinants along the block-diagonal:

$$|\ddot{E}| = |\ddot{E}_{CC}|\left|\ddot{E}_{XX} - \ddot{E}_{XC}(\ddot{E}_{CC})^{-1}\ddot{E}_{CX}\right| \approx |\ddot{E}_{CC}|\,|\ddot{E}_{XX}|$$
$$= \prod_{i=1}^{F} |\ddot{E}_{c_ic_i}| \prod_{i=1}^{N} |\ddot{E}_{x_ix_i}|. \tag{6}$$

Each sub-determinant can now be evaluated independently using (4), leading to

$$\mathcal{L}_{\text{sam}} = \frac{1}{2}\log|\ddot{E}|$$
$$\approx \frac{F\lambda_C - \lambda_G}{2F}\sum_{i=1}^{F}\log(2N_i) + \frac{\lambda_D}{2}\sum_{i=1}^{N}\log(2F^i). \tag{7}$$

The complexity has two parts, one for coding the motion parameters, and one for coding the structure points. For the cameras considered in this paper, the appropriate values are given in Table 1. The uncalibrated affine camera has 8 degrees of freedom, and the global coordinate frame has a 12-parameter ambiguity. The calibrated perspective camera has 6 degrees of freedom, and the global coordinate frame has a 7-parameter ambiguity. If the scene is planar, the scene points have only 2 degrees of freedom each, but the global ambiguity is reduced to 4 because the parameters of the plane have to be computed. For simplicity, assume that the scene is positioned in the xy-plane of the global coordinate

Table 1 Complexity parameters for different camera and scene models

Camera type	Scene type	λ_C	λ_G	λ_D
Uncalibrated affine	general	8	12	3
Calibrated perspective	general	6	7	3
Calibrated perspective	planar	6	4	2

system. Then only a 2D-translation, an in-plane rotation, and a global scale are required to fix the coordinate frame, and the z-coordinate of all scene points is frozen at $z_i = 0$.

Finally, the goal is to model multiple motions, so we need an index to store which points belong to object $\mathcal{M}$ in each frame of the sequence. To this end, we need a binary index of length N to indicate the scene points on $\mathcal{M}$ (the $N^{\mathcal{M}}$ points, whose projection belongs to $\mathcal{M}$ in any of its frames). Furthermore, we have to code the index of the first frame, in which $\mathcal{M}$ becomes visible, with coding length $\log(F)$. Finally, we have to look at the following $F^{\mathcal{M}}$ frames (the frames, in which the object remains visible), and for each of the object's points code the subset of frames in which the point is visible. There are $\frac{1}{2}F^{\mathcal{M}}(F^{\mathcal{M}}-1)$ such subsets, so the total coding length is

$$\mathcal{L}_{\text{idx}} = N\log(2) + \log(F) + N^{\mathcal{M}}\log\left(\frac{F^{\mathcal{M}}(F^{\mathcal{M}}-1)}{2}\right), \tag{8}$$

where the factor $\log(2)$ accounts for the fact that the binary index requires N bits, while in the remaining derivation we have used the natural logarithm, thus choosing $\log_2(e)$ bits as the unit of codelength.

2.3 Making up the Balance

Using the structure-and-motion representation, we reduce the codelength by $\mathcal{L}_{\text{img}}$, but increase it by $(\mathcal{L}_{\text{err}} + \mathcal{L}_{\text{sam}} + \mathcal{L}_{\text{idx}})$. The total savings thus are

$$\begin{aligned}\mathcal{D}_{\mathcal{M}} = {} & \log\frac{w^2}{2\pi\sigma^2}\sum_{i=1}^{F} N_i - \frac{1}{2\sigma^2}\sum_{i=1}^{F}\sum_{j=1}^{N_i} r_{ij}^2 \\ & - \frac{\lambda_D}{2}\sum_{j=1}^{N}\log(2F_j) - \left(\frac{\lambda_C}{2} - \frac{\lambda_G}{2F}\right)\sum_{i=1}^{F}\log(2N_i) \\ & - \Bigg[N\log(2) + \log(F) \\ & + N^{\mathcal{M}}\log\left(\frac{F^{\mathcal{M}}(F^{\mathcal{M}}-1)}{2}\right)\Bigg].\end{aligned} \tag{9}$$

If this value is positive, using the structure-and-motion representation reduces the total codelength, or equivalently, it increases the probability of the model. Intuitively, (9) can be interpreted in the following way:

- The first term rewards the motion for reducing the number of outliers. The benefit per explained point is a function of the relative dispersion between the uniform distribution for an outlier and the Gaussian of standard deviation σ for an inlier.
- The second term penalizes motions with large residuals, aiming for goodness-of-fit.
- The third term penalizes the complexity of the scene model and the fourth term that of the camera model, aiming for a simple and concise explanation.
- The last term accounts for the book-keeping overhead of the motion. In particular, this term assigns a basic cost to each new motion which is introduced into the model, thus making sure that motions are not arbitrarily broken in time: due to the basic cost of a new motion, it will be cheaper to explain the same set of tracks by a single motion over, say, four frames, than by two separate ones over two frames each.

Remark Contrary to a number of model selection applications in computer vision (Torr 1998; Matsunaga and Kanatani 2000; Kanatani 2004), we use model selection not only to detect, whether the model we are fitting is degenerate and should be replaced by a more restrictive one, but also to decide, whether we should fit a parametric model at all. In other words, we are also comparing against a background model, which asserts that the data do not follow any parametric constraint. Therefore the first and last term *cannot* be dropped, other than in the case of degeneracy detection, where they are discarded, since they are the same independent of which model we fit to the data. The present case, which has the option of not choosing any model, has been treated in Leonardis et al. (1995), Maybank and Sturm (1999), Kverh and Leonardis (2004) following a more ad-hoc approach to model selection with an empirically engineered cost function. In this work, we have attempted to base the codelengths on simple underlying probability distributions (even though some approximations have to be made in the course of the derivations).

2.4 Dealing with Overlap

Now assume that we have two motions, $\mathcal{M}_1$ and $\mathcal{M}_2$. Then a point $\mathbf{u}_i$ may be an inlier to both, and it is at this stage not possible to decide, which one it shall be assigned to. To assure the minimal codelength, we therefore have to make sure that the point is only coded once in each frame. Adding the savings $(\mathcal{D}_1 + \mathcal{D}_2)$ unjustly assumes that coding these points twice could reduce the codelength further. To remedy this, we need a correction term, $(\mathcal{D}_1 + \mathcal{D}_2 - \mathcal{D}_{1\cap 2})$, where

$$\begin{aligned}\mathcal{D}_{1\cap 2} = \log\frac{w^2}{2\pi\sigma^2}\sum_{i=1}^{F} N^{|1} - \frac{1}{2\sigma^2}\sum_{i=1}^{F}\sum_{j=1}^{N^{|1}} r_{1j}^2 \\ - N^{\|1}\log\left(\frac{F^1(F^1-1)}{2}\right) \\ + \log\frac{w^2}{2\pi\sigma^2}\sum_{i=1}^{F} N^{|2} - \frac{1}{2\sigma^2}\sum_{i=1}^{F}\sum_{j=1}^{N^{|2}} r_{2j}^2 \\ - N^{\|2}\log\left(\frac{F^2(F^2-1)}{2}\right).\end{aligned} \tag{10}$$

The first term of the correction makes sure that the savings for those $N^{|1}$ points, which have a larger normalized residual in motion $\mathcal{M}_1$ (and thus a less efficient coding) are only counted once, for motion $\mathcal{M}_2$. The second term corrects for the codelength of their residual in $\mathcal{M}_1$, which is no longer required. The third term takes into account that the number of scene points, whose tracks are explained by $\mathcal{M}_1$, has been reduced by $N^{\|1}$ (which need not be the same as $N^{|1}$), and corrects the indexing cost accordingly. The remaining three terms have the same effect for those points, which have larger normalized residuals in $\mathcal{M}_2$. Note that the coding length for the structure-and-motion parameters is not influenced by the overlap. Since the points fit both motions well, they can be used to estimate both of them, even though they are later only coded using one of them.

Remark It is important to understand that correct treatment of ambiguous points is a fundamental requirement in a model selection scheme, which simultaneously recovers multiple motions. If the deduction for the overlap is neglected, any motion whose likelihood outweighs the complexity penalty will increase the total likelihood, and will be selected. As an extreme example, consider the case where $\mathcal{M}_2$ consists of the first $(F^1 - 1)$ frames of $\mathcal{M}_1$, i.e., it is a subset of $\mathcal{M}_1$ representing the case that the object has left the field of view or become occluded in the last frame. If $\mathcal{M}_2$ reduces the codelength, then in most cases so does $\mathcal{M}_1$, and both will be selected, which clearly contradicts the desire to minimize the model complexity. If, on the contrary, we take care not to "explain the same points twice" by introducing $\mathcal{D}_{1\cap 2}$, then the two will never both be selected, because if one is a subset of the other, $\mathcal{D}_{1\cap 2} > \min(\mathcal{D}_1, \mathcal{D}_2)$.

2.5 Minimizing the Codelength

To minimize the codelength one must maximize the total savings $\mathcal{D}$. The question is which motions to use, hence the variable is a boolean vector $\mathbf{b}$ of length M, which indicates the presence $(b_i = 1)$ or absence $(b_i = 0)$ of a motion in the model (Leonardis et al. 1995; Stricker and Leonardis 1995). The total savings in codelength, as a function of which motions are used, are then given by the quadratic boolean expression $\mathcal{D}(\mathbf{b}) = \frac{1}{2}\mathbf{b}^\top \mathtt{D}\mathbf{b}$, where $\mathtt{D}$ is a symmetric matrix of the following form:

$$\mathtt{D} = \begin{bmatrix} 2\mathcal{D}_1 & -\mathcal{D}_{1\cap 2} & \dots & -\mathcal{D}_{1\cap M} \\ -\mathcal{D}_{1\cap 2} & 2\mathcal{D}_2 & \dots & -\mathcal{D}_{2\cap M} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathcal{D}_{1\cap M} & -\mathcal{D}_{2\cap M} & \dots & 2\mathcal{D}_M \end{bmatrix}. \tag{11}$$

Note that no parameters have to be tuned in (9, 10). The formulation as a quadratic problem is only possible, because the contributions of different motions to the codelength have been separated, and this is achieved by the simplification of only considering the joint probabilities of up to 2 motions.[3]

Obviously, we are only interested in candidate motions, which can potentially reduce the codelength. This implies that the diagonal elements of matrix $\mathtt{D}$ are strictly positive, $\{\forall i : \mathcal{D}_i > 0\}$, and its off-diagonal elements are non-positive, $\{\forall i \neq j : \mathcal{D}_{i\cap j} \leq 0\}$. It is easy to see that a quadratic boolean function with the latter property is a submodular set function (Nemhauser et al. 1978). Maximizing $\mathcal{D}$ over $\mathbf{b}$ is known to be NP-hard. However, a few simple observations can help us to solve it for our specific case. As a starting point, we know that our solution will only contain few motions. Furthermore, the following holds:

Lemma 1 *Let $\widehat{\mathbf{b}}$ be the vector, at which $\mathcal{D}$ attains the global maximum, and let $\mathbf{b}'$ be a subset of $\widehat{\mathbf{b}}$, $\{\forall i :\ \widehat{b}_i \geq b'_i\}$. Let $\mathbf{b}''$ be obtained by switching exactly one 0-element of $\mathbf{b}'$ to 1, $|\mathbf{b}'' - \mathbf{b}'| = 1$. Then, $\mathbf{b}''$ can be a subset of $\widehat{\mathbf{b}}$ only if $\mathcal{D}(\mathbf{b}'') > \mathcal{D}(\mathbf{b}')$. If $\forall \mathbf{b}'' :\ \mathcal{D}(\mathbf{b}'') \leq \mathcal{D}(\mathbf{b}')$, then $\mathbf{b}' = \widehat{\mathbf{b}}$.*

Proof Let b'_k denote one of the 0-elements of $\mathbf{b}'$ which needs to be switched to 1 to obtain $\widehat{\mathbf{b}}$, $\{\widehat{b}_k > b'_k\}$. Let $\mathbf{b}^-$ denote the vector which is obtained by starting from $\widehat{\mathbf{b}}$ and switching off element k: $b^-_k = 0$, $|\widehat{\mathbf{b}} - \mathbf{b}^-| = 1$. Since $\widehat{\mathbf{b}}$ is the global maximum, $\mathcal{D}(\widehat{\mathbf{b}}) - \mathcal{D}(\mathbf{b}^-) > 0$. Submodularity implies $\mathcal{D}(\mathbf{b}') - \mathcal{D}(\mathbf{b}) \geq \mathcal{D}(\widehat{\mathbf{b}}) - \mathcal{D}(\mathbf{b}^-)$, and therefore $\mathcal{D}(\mathbf{b}') - \mathcal{D}(\mathbf{b}) > 0$. □

The lemma states that the path from any subset $\mathbf{b}'$ to $\widehat{\mathbf{b}}$ does not contain descent steps. Making use of the fact that the scene only contains a small number R of motions, and that the empty solution $\mathbf{b} = \mathbf{0}_{M\times 1}$ is a subset

[3] If ≥3 motions share points, their joint use is over-penalized, e.g., for 3 motions the last term of the joint savings $\mathcal{D}_1 + \mathcal{D}_2 + \mathcal{D}_3 - \mathcal{D}_{1\cap 2} - \mathcal{D}_{1\cap 3} - \mathcal{D}_{2\cap 3} + \mathcal{D}_{1\cap 2\cap 3}$ is disregarded. However the influence of this approximation is small, because the number of affected points is small, whenever the candidate motions are significantly different (which in our scheme is ensured by the clustering stage).

of $\widehat{\mathbf{b}}$, one can devise a multi-branch ascent method (see Algorithm 2). The method always leads to the global maximum, however it is still exponential in complexity. For larger sets of candidate motions, one has to resort to a heuristic version: for each level of the search, the solutions are sorted by the objective value $\mathcal{D}(\mathbf{b})$ and only the best ones for each branch are retained. The number of branches, which are retained for the next step, decreases in geometric progression with factor α from one level to the next, say $\alpha = 4$, $T = \{128, 32, 8, \ldots\}$, so that the total number of search paths is $\{128, 4096, 32768, \ldots\}$. Retaining multiple sub-branches avoids getting stuck at a weak local minimum and only gradually focuses on the most promising branches. The complexity for M candidates and R actual motions is $\mathcal{O}((\frac{1}{\alpha})^{(1+2+\cdots+R)} M^{(R+1)})$. Theoretically, the heuristic does not guarantee a global maximum anymore. In practice it produces good solutions, and in our experiments it outperforms all-purpose search methods such as Tabu-search or multi-start gradient ascent, which do not exploit the special structure of the problem to the same extent.

Algorithm 2 Multi-branch optimization for $\mathcal{D} = \frac{1}{2}\mathbf{b}^{\mathsf{T}}\mathrm{D}\mathbf{b}$.

```
1. Level 0: Start from a scene without any motions: $R = 0$, $\mathcal{D} = 0$, $\mathbf{b} = \mathbf{0}_{M\times 1}$
2. Level 1: Compute the value of $\mathcal{D}$ for all $M$ possible solutions with ($R = 1$) motion
3. Discard all solutions with $\mathcal{D} \leq 0$, since adding motions to such a solution cannot lead to the maximum (Lemma 1)
4. Level 2: Build all pairwise combinations ($R = 2$) of the remaining motions, and compute $\mathcal{D}$ for them
5. Discard those pairs, which do not attain a higher value than any of the two motions alone (again, these cannot lead to the maximum)
6. Level 3: Join the remaining pairs to triplets ($R = 3$) and compute $\mathcal{D}$ for them (Note that based on the previous steps, computing $\mathcal{D}$ for an $R$-tuple of motions only requires $R$ additions)
7. Level R: Keep discarding dead-end search paths and increasing $R$, until no $(R + 1)$-tuple exceeds the previous maximum attainable with $R$ motions
```

Remark Simple greedy gradient-ascent is not a suitable method for the problem. The theoretical error bounds are so loose that they are meaningless (Nemhauser et al. 1978), and in practice it fails more often than not. For an exemplary illustration, consider the following simple example: given is a scene with two motions, each giving rise to the same number of tracks. Furthermore, there is a candidate in the set, which explains significantly more than half of the points, albeit with somewhat higher error. Hill-climbing will commit itself to this candidate without knowing about the next step, and never will be able to recover. This behavior can actually be seen in our experiments: the strongest motion after the first selection step is often not part of the final solution.

3 Multibody Structure-and-Motion Algorithm

A unifying property of all model-selection methods is that in fact they are *scoring* methods. They provide a way of computing the scalar score of a given model, but are not concerned with a search procedure which leads to a model with a high score. To make model selection useful, it needs to be integrated into an optimization framework. In this section we will devise a Monte-Carlo type framework, which starts from randomly generated candidate motions, progressively refines and filters the set of candidates to a set of putative motions for a given sequence, and then uses model selection to pick the best combination of motions from this set as a model for the sequence. Langs et al. have proposed a different optimization strategy in the context of active shape models (Langs et al. 2005): to find the best collection of sub-models, they start from a heavily oversegmented data set and in a random fashion merge segments such that the codelength is reduced. In this way, they arrive at a set of local minima. The best of these local minima is selected with a second model scoring step. While potentially more efficient, their concept is not applicable in the presence of outliers, since outliers will cause the merging procedure to break down, yet the outliers are hard to detect on the small initial segments due to the small amount of data.

A candidate motion is a hypothetical object moving in 3D space, modeled as a number of scene points $\mathbf{x}$. The object moves through the field of view for a number of frames, and the points $\mathbf{x}$ give rise to point tracks $\mathbf{v}$ through these frames, which satisfy an appropriate n-view constraint. For two consecutive frames, points on the same rigid object have to satisfy the appropriate two-view constraint. The set of candidates, which forms the input for model selection, may be redundant, but for each moving object actually present in the scene, there has to be at least one candidate, which describes it well.

We start from a sequence of F frames recorded with some camera. With the point tracker, N points have been tracked through the sequence. Each of these points appears in at least 2 and at most F consecutive frames. At this point, it is unknown how many moving objects are visible in the sequence, and hence it is also unknown, which object a point belongs to, or whether the track for that point contains false matches.

As atomic hypotheses to start from, two-view constraints between consecutive frames are generated at random and linked to longer motions. Since brute-force random sampling and linking leads to a combinatorial explosion, some care has to be taken: very improbable motions need to be pruned from the candidate set as early as possible, and redundant motions, which are very similar, need to be avoided. As will be seen, the important notion here is that in correspondence-based structure-and-motion, a moving object is modeled as a rigidly moving *set of points*. The common trait of the steps in this section is that they focus on this *inlier set*, rather than the motion parameters, to compare and judge tentative motions.[4]

3.1 Pairwise Sampling

Let an (unknown) 3D scene point be denoted by $\mathbf{x}$, and its image in the ith frame by a homogeneous 3-vector $\mathbf{v}^i$, with $i \in \{1, \dots, F\}$. Candidate motions will be generated by randomly generating two-view constraints between neighboring frames and linking them to longer motions. Different camera models and scenes lead to different constraints. In the following, we will consider two camera models and two scene models.

In the case of an uncalibrated affine camera viewing a general scene, two views are related by the affine epipolar constraint $(\mathbf{v}^{i+1})^\mathsf{T} \mathtt{F}_\mathtt{A} \mathbf{v}^i = 0$, where $\mathtt{F}_\mathtt{A}$ denotes the affine fundamental matrix. For a calibrated perspective camera, they are related by the epipolar constraint $(\mathbf{u}^{i+1})^\mathsf{T} \mathtt{E} \mathbf{u}^i = 0$, where $\mathtt{E}$ denotes an essential matrix, and $\mathbf{u}$ denotes an image point in canonical coordinates $\mathbf{u}^i = \mathtt{K}^{-1}\mathbf{v}^i$. K is the calibration matrix containing the intrinsics of the camera. These relations only hold for the most general scene model, a full 3D scene. If the scene viewed by the two cameras is planar, then the epipolar constraint degenerates. For example, for the case of a perspective camera viewing a planar scene, we get the projectivity constraint $(\mathbf{v}^{i+1}) \times \mathtt{H}\mathbf{v}^i = 0$, where $\mathtt{H}$ denotes a 2D homography.

As atomic hypotheses to start from, two-view constraints are generated from minimal subsets of the data. To simplify the explanation, we will assume for the following explanation that the camera is known to be perspective, but the scene model is unknown. The described method works in exactly the same way for other combinations. An essential matrix for two calibrated perspective cameras can be estimated with the five-point algorithm of Nistér (2004). By examining the residuals of all correspondences with respect to the estimated essential matrix, it is possible to separate inliers, which satisfy the constraint within a small tolerance, from outlier, which do not. We use the TSSE estimator of Wang and Suter for this task (Wang and Suter 2004), but any statistical outlier rejection procedure could be used. Such procedures are sometimes called “robust estimators”, however the task really is to determine the inlier set, from which it is trivial to estimate the parameters of the constraint with a least-squares fit.

To cover the possibility of a planar scene, it may also be necessary to estimate homographies, which can be done with linear methods (Hartley and Zisserman 2000). Since the projectivity is a tighter constraint than the epipolar geometry, all points which satisfy the projectivity will also satisfy the epipolar geometry, even though the estimation of the latter is ambiguous. The robust estimation problem is therefore much easier, because only the points already identified as inliers to the epipolar geometry form the base data.

To increase the chance of finding an uncontaminated sample, we recommend using a local sampling scheme (Schindler and Suter 2005). Except for transparent objects, points belonging to the same rigid object will be clustered in the image plane, and local sampling will dramatically reduce the number of samples required to find an uncontaminated one.[5]

Note that for each moving object we only have to make sure good candidates are found in *one* of the sub-regions. If, as in most practical scenarios, the minimum image area covered by an object is known, it is easy to find such a subdivision. This means that the required sample number per frame is constant, independent of the number of motions, and the total number of samples grows linearly with the length of the sequence.

Having estimated the inlier set and standard deviation of all putative two-view motions, the candidate set can be pruned for the first time: only plausible candidates in terms of inlier count and standard deviation are retained. The thresholds can be chosen conservatively, since they only serve to discard the most improbable candidates: an upper bound for the allowable standard deviation is the localization uncertainty of the image points, which is easily obtained from the point tracker, while the minimum inlier number is set to some very low value, say 5% of all image points in a frame.

3.2 Estimating the Standard Deviation

In most cases the standard deviation (i.e., the scale of the measurement uncertainty of the image points) is unknown

[4] We are aware that making hard inlier/outlier decisions at an early point is theoretically questionable from a statistical point of view. For the sake of simplicity, we will nevertheless explain the method using hard decisions. The described algorithms can easily be extended to fuzzy membership values by replacing the binary inlier/outlier index of each point with its inlier probability. However, the practical difference is small, and in our view does not warrant the additional computational burden.

[5] For the experiments in Sect. 4, images were was subdivided into 3 overlapping rows and 3 overlapping columns, and samples were drawn from the entire image, each column, each row, and each of the 9 regions defined by a row-column intersection. This hierarchical scheme proved to be a reasonable compromise between local coherence and global extension, which works well for different images.

and needs to be estimated from the data. As pointed out by Kanatani (2004), the standard deviation should always be estimated from the residuals of the most general motion: this estimate is correct even for degenerate cases, where parameter estimation becomes ambiguous—the motion parameters cannot be estimated reliably precisely *because* the error is similar for different sets of parameters. Although computing the standard deviation from the residuals of a more restrictive model will yield a higher value, it would be unreasonable to assume that the measurement uncertainty of the same point set changes depending on what motion we later fit to the measurements.

3.3 Motion Clustering

The two-view motions recovered at this point will be highly redundant. Many of the candidates will correspond to the same object and be similar. Conversely, it is improbable that there are many clusters of similar motions among the spurious candidates, which have survived to this point. Clustering will detect and remove as much as possible of the redundancy. This will both reduce the number of candidate motions further, and allow an improved estimation of the remaining ones.

Clustering in parameter space is difficult. Even similar sets of moving points may yield motions with very different parameters. We therefore return to the definition of similarity as "explaining the same tracks", and resort to clustering based on the inlier sets, similar to Wills et al. (2003). A two-view motion is represented by a binary vector of size N, with entries 1 for its inliers, and entries 0 for its outliers. The Hamming-distance d_{H} between these vectors (the number of differing bits) is then used as a similarity measure for clustering. $d_{\mathrm{H}}=0$ means that two inlier sets are identical, while $d_{\mathrm{H}}=N$ means that the outliers of one set are exactly the inliers of the other. Our implementation uses simple average-linkage hierarchical clustering (Duda et al. 2001), however more sophisticated methods could potentially be used.

The new set of candidates is now given by the representative "mean" motions of all clusters. These "means" are obtained with a simple consensus mechanism: the inliers of the "mean" are all points, which are inliers to $>50\%$ of the cluster members. The epipolar geometry of a cluster is re-estimated from this inlier set. Optionally, one can discard very small clusters (say, with ≤ 2 members), which are likely to be spurious motions, in order to further reduce the candidate set.

Remark Clustering random samples by their inlier sets and then forming a consensus per cluster can be regarded as a simple refinement step, which seeks to polish the candidates at low computational cost. The alternative, to polish each individual sample with an iterative algorithm, and then prune candidates which have converged to identical solutions, is usually not tractable, because of the high computational cost of iteratively fitting such a large number of models.

3.4 Motion Linking

After clustering, we are left with a small number of putative two-view motions (in practice, <10 per frame), each representing the motion of a set of points from one frame of the sequence to the next. It is important to notice that we have not yet achieved an optimal set for each pair of consecutive frames. It is quite possible that some of the candidate motions only explain part of a moving object, or that they explain two objects, if their relative motion between the two frames is small. It is also quite likely that some spurious motions accidentally are strong enough to survive up to this point.

The two-view motions have to be concatenated to longer motion chains. It is not known, when each moving object has entered and left the field of view, so all chains of length ≥ 2 frames are potential candidates. Again, exhaustively linking all possible chains of length $\leq F$ leads to a combinatorial explosion, and it disregards the temporal coherence of motions. Since sequence analysis only makes sense, if the scene changes slowly compared to the frame-rate, few tracks on each moving object will be lost per frame. Linking only motions with similar inlier sets, thus enforcing the temporal coherence, greatly reduces the number of candidates. Only a loose threshold (say, 50%) should be used, so as not to eliminate motions with strong self-occlusions due to rotation.

At the linking stage, we can no longer avoid the inherent complexity of the problem. Unlike the previous steps, motion linking provokes a potentially exponential increase in the number of candidates. This is why great care has been taken to prune the candidate set as early as possible. Although the described pruning measures are extremely simple, they are efficient. Experimentally, for sequences of up to 4 motions and 15 frames, the number of candidate motions is generally <1000.

Note that generating motions by linking epipolar geometries in this way does not impose any restrictions on the motion other than rigidity, so long as the inlier sets are consistent. Within the limitations of the feature tracker, irregular and jerky motions are not penalized compared to smooth ones, since there is no temporal prediction involved.

3.5 Model Selection

In the hypothesis generation stage described so far, we have generated the required input for the model selection procedure—a redundant set of M putative object motions, each given by a changing set of image point tracks observed through some part of the sequence. From this set, the best explanation for the observed image point tracks is selected by applying the theory developed in Sect. 2. Each candidate motion is described by a number of scene points $N^{\mathcal{M}}$, its length in frames $F^{\mathcal{M}}$, a number $N_i^{\mathcal{M}}$ of image points in each of these frames (with $N_i^{\mathcal{M}} \leq N^{\mathcal{M}}$), and their corresponding residuals with respect to the estimated motion, $\{r_{ij}\}$. With this information, the entries of matrix $\mathtt{D}$ are computed using (9, 10), and the objective function maximized as described in Sect. 2.5.

3.6 Motion Segmentation

Model selection yields an optimal set of motions and their respective inlier sets. Segmentation therefore reduces to the problem of disambiguating points, which change from one motion to the next over time, or satisfy more than one motion model. An obvious solution in the presence of multiple frames is to enforce temporal consistency. So far, temporal consistency has only been used to link motions between consecutive pairs of frames, but not at the multi-frame level. Since a scene point is located on a physical object, it cannot normally pass from one motion to another, while the two objects continue to move independently. An exception is the case that the tracker drifts, i.e., it wrongly matches a point between two frames, but then locks onto the new point and tracks it correctly.

If in a part of the sequence a point switches back and forth between motions, or is an inlier to more than one motion, we form a consensus over time. Within the time window, during which both motions exist, the point is assigned such that it drifts at most once, while changing as few class memberships as possible. This will clean up any false assignments due to points accidentally satisfying the epipolar constraint (the vast majority of cases). Even in the case that a point truly drifts from one motion to the other, the heuristic will detect this behavior and try to fix it, but the transition may happen in the wrong frame. Note that by temporal consistency, we again mean consistent class membership, rather than smoothness of tracks. The effect of the consensus over time is illustrated on a practical example in Fig. 3.

The segmentation of correspondences *between motions* is almost perfect (i.e., very few points are assigned to the *wrong* motion). However, some points on moving objects are often miss-classified as outliers. This is an inherent difficulty of robust classification methods, which provide a rejection class for outliers. The parameters of each class are estimated at the same time as the class membership, therefore there exists the possibility of estimating slightly incorrect parameters based on a subset of the class, and assigning the remainder of it to the outliers. Methods without an outlier class do not encounter this problem, because all points *have* to be assigned to one of the motions.

4 Experimental Evaluation

4.1 Synthetic Data

A synthetic data set was generated with 5 views of 4 rotating planar objects. The experiment was designed to test the method as well as study the influence of correct modeling of degenerate scenes. Each object has 50 tracks, and 50 outliers were added by randomly generating tracks with a displacement between adjacent frames, which is similar to the correct tracks. The image size was set to (512 × 512) pixels, and the image point coordinates were contaminated with Gaussian noise of magnitude 0.5 pixels. Two different camera models were applied: in a first experiment, the data was segmented using a perspective camera model and a 3D scene model, while in a second experiment the algorithm should also classify each moving object as planar or

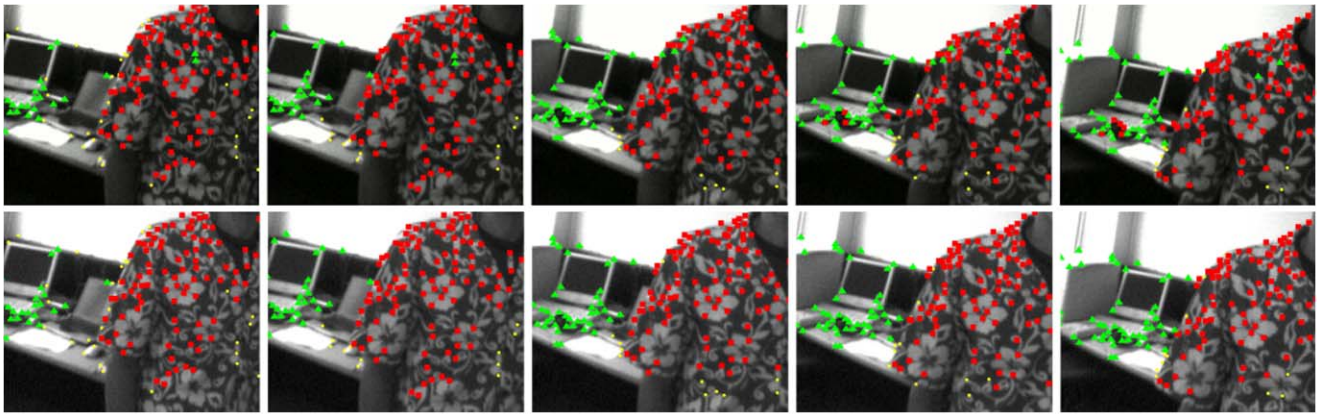


Fig. 3 Enforcing temporal consistency to improve motion segmentation. The two rows show the same region of the "flowershirt" sequence, with several points satisfying both epipolar geometries. *Top row*: segmentation based on individual residuals without enforcing coherence over time. *Bottom row*: segmentation after building a consensus over time

Table 2 Segmentation results for "spinning wheels" sequence (4 moving objects with 50 points each, 50 outliers). False positives (FP) are outliers assigned to a motion, false negatives (FN) are points from the motion classified as outliers. No points were assigned to the wrong motion

General object		Frame 1	Frame 2	Frame 3	Frame 4	Frame 5
Motion A	FP / FN	0 / 0	0 / 0	0 / 0	1 / 1	1 / 0
Motion B	FP / FN	0 / 0	2 / 0	2 / 0	1 / 0	0 / 0
Motion C	FP / FN	1 / 0	2 / 0	2 / 0	4 / 0	2 / 0
Motion D	FP / FN	0 / 1	2 / 0	2 / 0	1 / 0	0 / 3
Total error		1.0%	3.0%	3.0%	4.0%	3.0%
Planar object						
Motion A	FP / FN	0 / 0	1 / 0	2 / 0	1 / 0	0 / 0
Motion B	FP / FN	0 / 0	0 / 0	0 / 0	1 / 0	1 / 1
Motion C	FP / FN	0 / 1	0 / 0	0 / 0	0 / 0	0 / 0
Motion D	FP / FN	0 / 1	0 / 0	0 / 0	0 / 0	0 / 0
Total error		1.0%	0.5%	1.0%	1.0%	1.0%

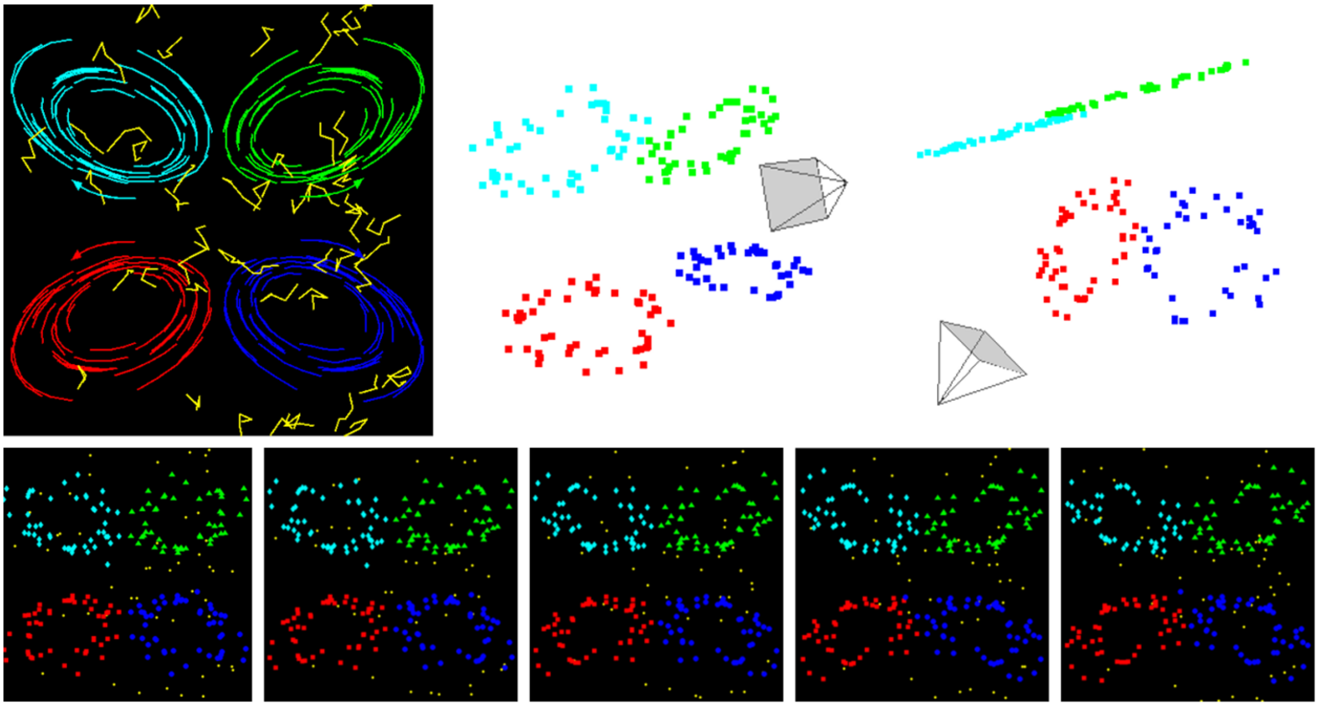


Fig. 4 Segmentation of the synthetic "spinning wheels" sequence. *Top*: feature tracks (*colors* denote the ground truth segmentation), and two views of the recovered 3D points in the first frame. Note that these results were computed using a 3D scene model, thus the coplanarity of the points visually proves the accuracy of the recovered motions. *Bottom*: recovered segmentation through the sequence. *Yellow dots* are points classified as outliers

non-planar. In the first experiment, the method correctly recovered the 4 motions. The segmentation into the 5 classes (including outliers) is 97.5% correct. In the second experiment, the method also recovered the 4 motions, and also classified all 4 objects as planar. The segmentation into the 5 classes is 99.3% correct—the tighter constraint leaves less room for miss-classification. See Table 2 and Fig. 4.

4.2 Real Data—Affine Camera

The method has been tested on various real data sets with different camera and scene models. The first set of experiments was conducted with the affine camera model, and general 3D scenes. The first image sequence has 12 frames and shows 3 objects moving on a table. A total of 173 points were tracked through the sequence. Since this se-

Table 3 Segmentation results for "three cars" sequence (3 moving objects). False positives (FP) are points wrongly assigned to a motion, false negatives (FN) are missed points on a motion

	#tracks		Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Frame 8	Frame 9	Frame 10	Frame 11	Frame 12
Motion A	44	FP / FN	0 / 9	0 / 4	0 / 4	0 / 5	0 / 5	0 / 5	0 / 4	0 / 5	0 / 5	0 / 5	0 / 6	0 / 9
Motion B	49	FP / FN	0 / 4	4 / 1	4 / 2	4 / 1	4 / 0	4 / 0	4 / 0	4 / 1	4 / 0	3 / 1	4 / 2	4 / 2
Motion C	81	FP / FN	0 / 9	0 / 1	0 / 0	0 / 2	0 / 1	0 / 2	0 / 1	0 / 3	0 / 4	0 / 5	0 / 4	0 / 15
Total segmentation error			12.7%	3.5%	3.5%	4.6%	3.5%	4.0%	2.9%	5.2%	5.2%	6.4%	6.9%	15.0%

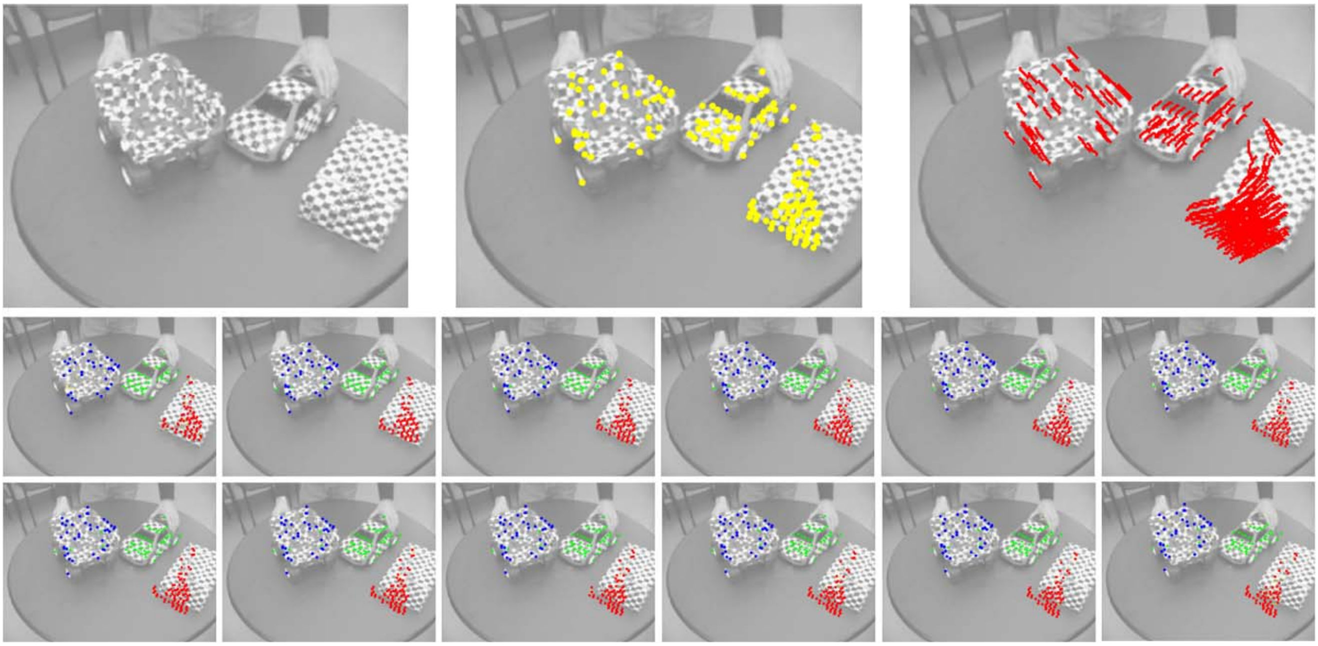


Fig. 5 Segmentation of the "three cars" sequence. *Top*: first frame, first frame with feature points superimposed, first frame with feature tracks superimposed. *Bottom*: recovered segmentation through the sequence. *Yellow dots* are points classified as outliers. All results are overlayed on the same frame, the remainder of the video was not available

quence was originally designed for a non-robust method (Vidal et al. 2002), only those features which could be tracked through all frames have been retained, and no outliers are present. This setup made it easy to assess the results quantitatively. The segmentation results are 93.9% correct—see Table 3 and Fig. 5. In Vidal and Hartley (2004), the authors achieved a 95.4% correct segmentation, using the additional constraints that the number of motions is known, there are no outliers, and each point belongs to the same motion throughout the sequence.

The second sequence for affine cameras consists of 8 frames showing a person moving diagonally towards the camera, while the camera itself moves through an office. 200 initial points were tracked with the KLT-tracker (Tomasi and Kanade 1991), lost points were immediately replaced, and points, which could not be tracked for at least one frame after their initial detection were removed. This led to a set of 263 tracks, including several outliers on apparent contours. The method was applied and correctly recovered two motions. Results are shown in Fig. 6.

4.3 Real Data—Perspective Camera

Next, the method was tested for the perspective camera model. This case, including the experiments given, has been treated earlier in Schindler et al. (2006). The first example is

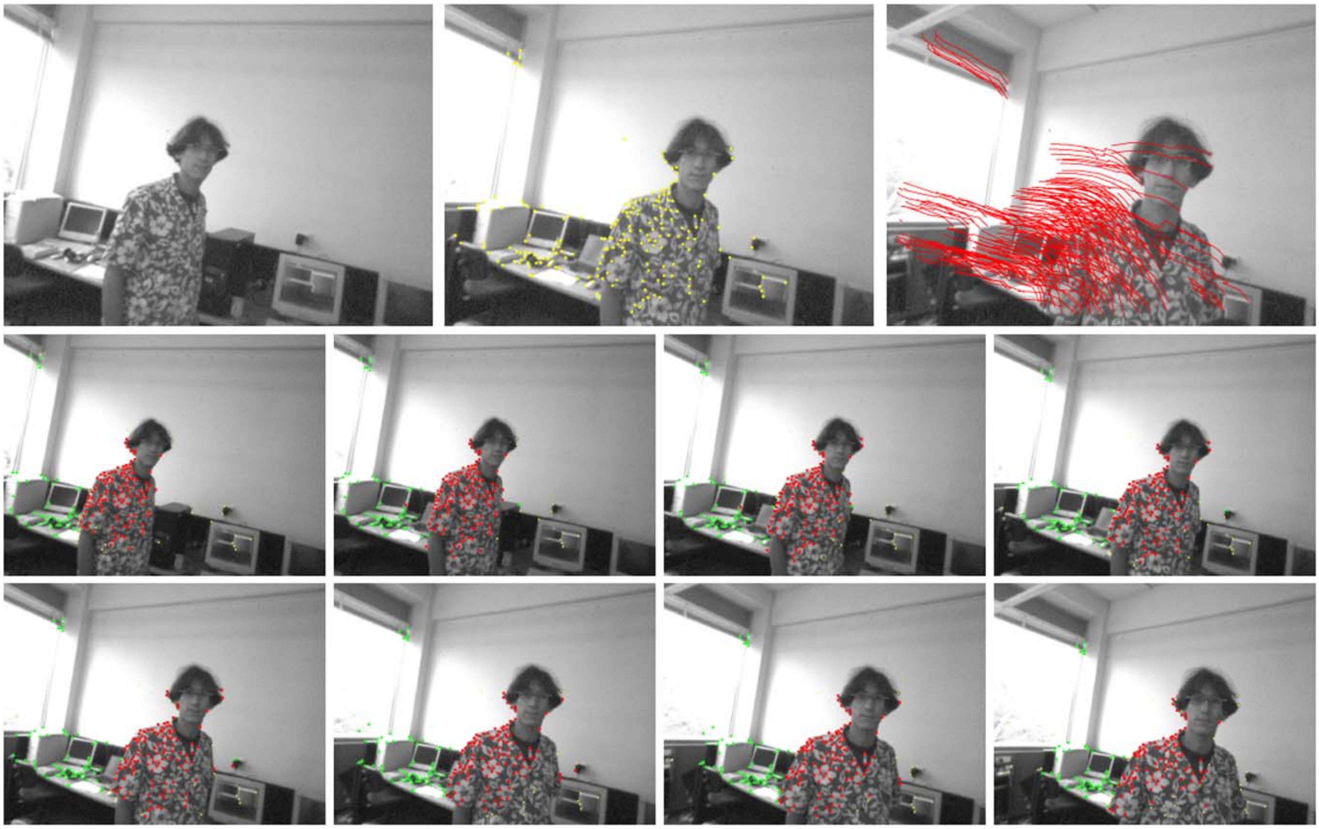


Fig. 6 Segmentation of the "flowershirt" sequence. *Top*: first frame, fifth frame with feature points superimposed, last frame with feature tracks superimposed. *Bottom*: recovered segmentation through the sequence. *Yellow dots* are points classified as outliers

a sequence of 10 frames showing two independently moving piles of boxes. 300 initial points were tracked with the KLT-tracker, lost points were immediately replaced, and points which could not be tracked for at least one frame were removed, leading to a total of 350 tracks. The set of tracks includes several outliers on apparent contours. The method was applied and correctly recovered two motions. For this scene the 3D reconstruction has actually been computed (using the knowledge that the camera did not move). Figure 7 shows the first and last frame with the point tracks superimposed, and the recovered motions both in the image plane and in a top view to show the motion of the 3D scene.

The second scene is a sequence of 10 frames showing 3 objects moving on a table (see Fig. 8). 300 feature points per frame were tracked, resulting in a total of 439 tracks. The third object is not visible in the beginning, but enters the field of view later, and a part of the box on the upper right leaves the field of view towards the end of the sequence. Furthermore, the motion is not smooth, with two of the three objects stopping at some point.

4.4 Real Data—Different Scene Models

In this section experiments are shown where the method also had to choose between a 2D or 3D scene model. Both experiments were carried out with a calibrated perspective camera. The first scene consists of 6 frames showing a (planar) magazine moving through a laboratory environment, while the camera itself moves through the environment independently. 300 feature points per frame were tracked, resulting in a total of 316 tracks, with several wrongly tracked outliers (see feature tracks in Fig. 9). The method correctly recovers the two motions and recognizes the magazine as planar. A few background points in the periphery are missed, because the background spans the entire image and is affected by remnants of radial distortion. Again, the 3D reconstruction was computed for this scene, this time keeping the background static. Figure 9 depicts the segmentation results, as well as a 3D view of the scene showing the positions of the camera and the magazine in the first and last frame with respect to the static background.

Note that the correct number of motions and a largely correct segmentation can also be obtained if only the most general scene model (a 3D object) is used. Some background points will however be wrongly assigned to the magazine motion. Using the tightest possible model reduces the chance that tracks satisfy the wrong model. Furthermore, it improves the 3D reconstruction of the scene by making sure the constraint is satisfied by the reconstructed scene.

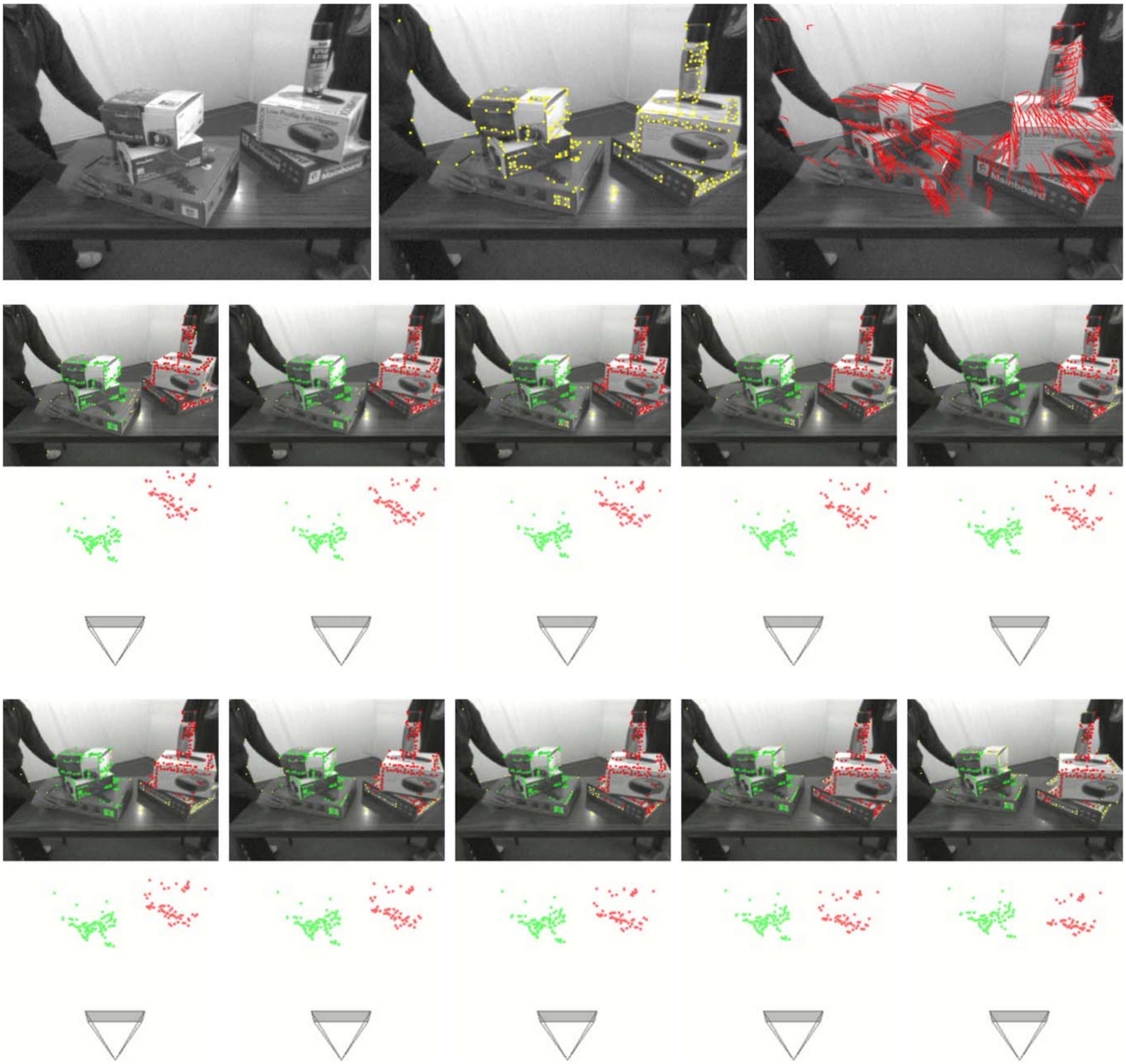


Fig. 7 Segmentation of the "boxes" sequence. *Top*: first frame, fifth frame with feature points superimposed, last frame with feature tracks superimposed. *Bottom*: recovered segmentation and top view of 3D tracks through the sequence. *Yellow dots* are points classified as outliers

The last scene consists of 11 frames from the movie "Groundhog Day". It shows a car moving diagonally towards the camera, while the camera itself pans to the right (see Fig. 10). 300 feature points per frame were tracked, resulting in a total of 524 tracks. The feature sets on both objects change a lot due to the fast motion, and there are several false matches due to strong motion blur. Again, the system was also allowed to choose between a general or planar scene. The camera motion is small and all background points are distant from the camera, therefore the most appropriate model for it is the planar scene model, while the van in the near field is best modeled as a general scene. The background motion disappears at the ninth frame because the visible background becomes almost featureless.

Again, a largely correct segmentation can also be obtained if only the most general scene model is used (as shown for this particular data set in Schindler et al. (2006)). However, in such a setting it goes unnoticed that the reconstruction of the 3D structure becomes unreliable due to the small baselines, whereas the choice between different scene models presented here explicitly uncovers the information that the background is (nearly) planar in relation to the motion.

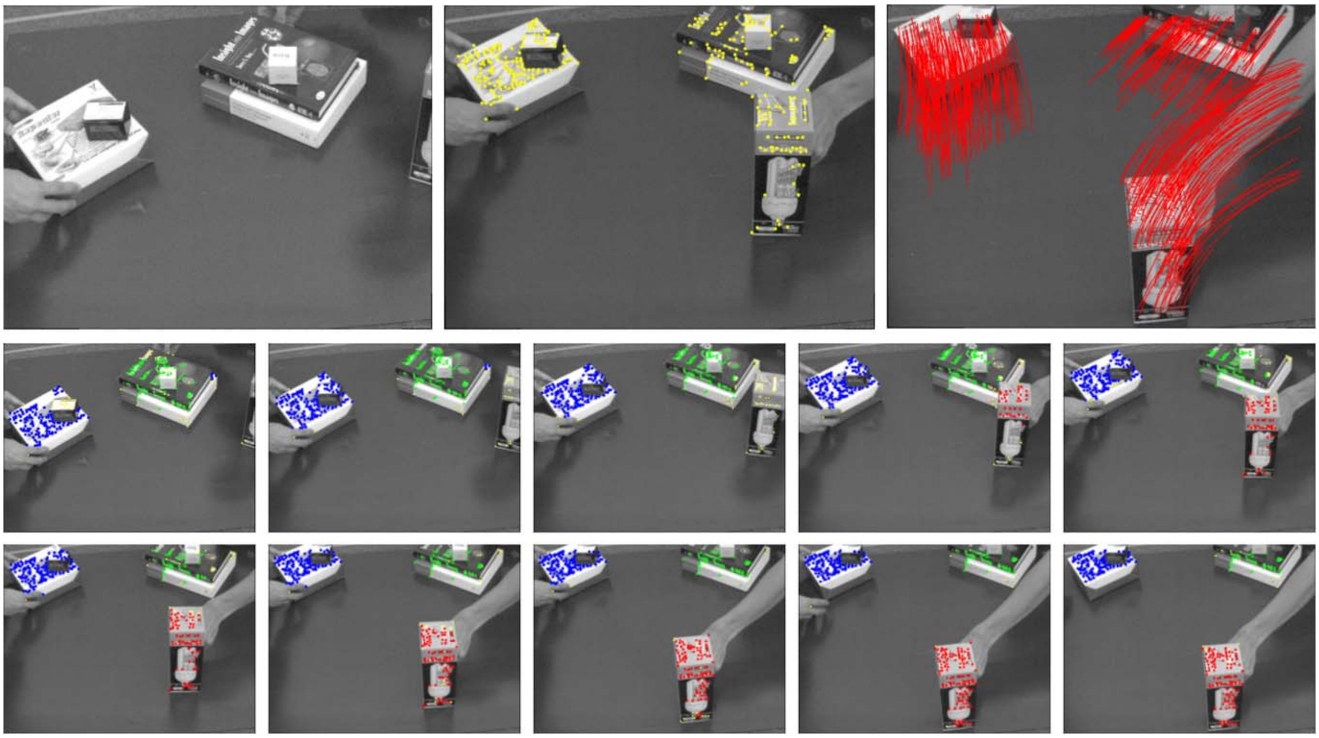


Fig. 8 Segmentation of the "lightbulb" sequence. *Top*: first frame, fifth frame with feature points superimposed, last frame with feature tracks superimposed. *Bottom*: recovered segmentation through the sequence

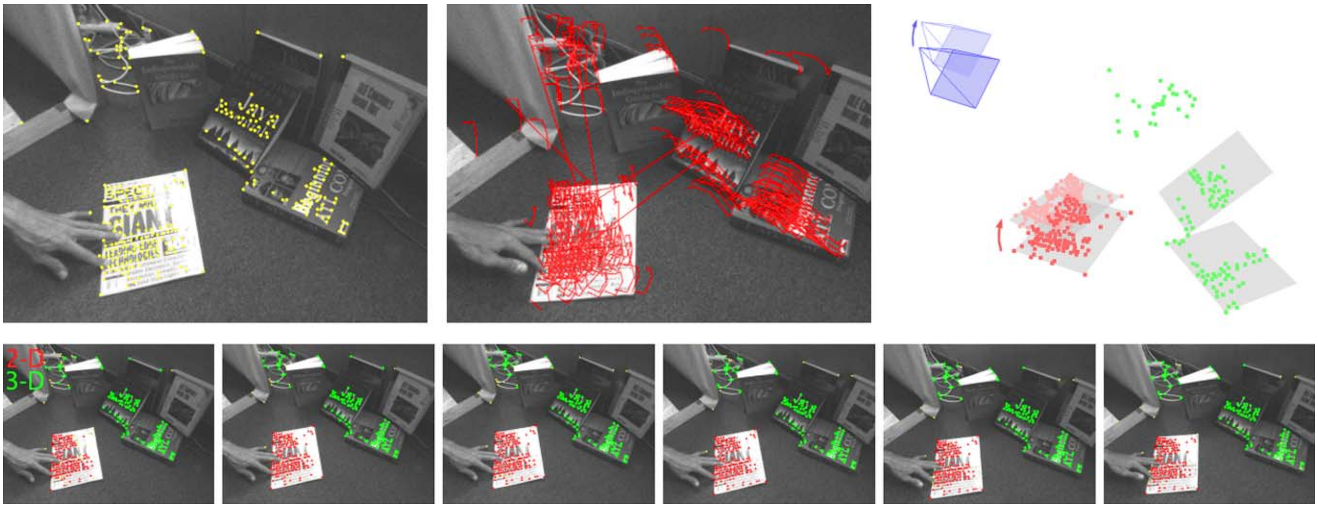


Fig. 9 Segmentation of the "magazine" sequence. *Top*: first frame with feature points superimposed, last frame with feature tracks superimposed, 3D view of reconstructed motion. The *gray rectangles* have been added manually for better visual impression. *Bottom*: recovered segmentation through the sequence. The selected type of scene model is printed into the first frame of each selected motion

4.5 Analysis of Possible Failures

The "delivery van" sequence is also used to demonstrate possible failures of the proposed method. The aim of this section is to point out potential pitfalls of model-selection based methods and give a guideline for their proper design.

Obviously, if a too restrictive model is used, the segmentation will be incorrect for those feature tracks, which do not satisfy the model. As an example, the sequence was segmented allowing only planar scenes. As can be seen in Fig. 11, the algorithm does its best to fit a planar scene, which explains as much of the van as possible, and there-

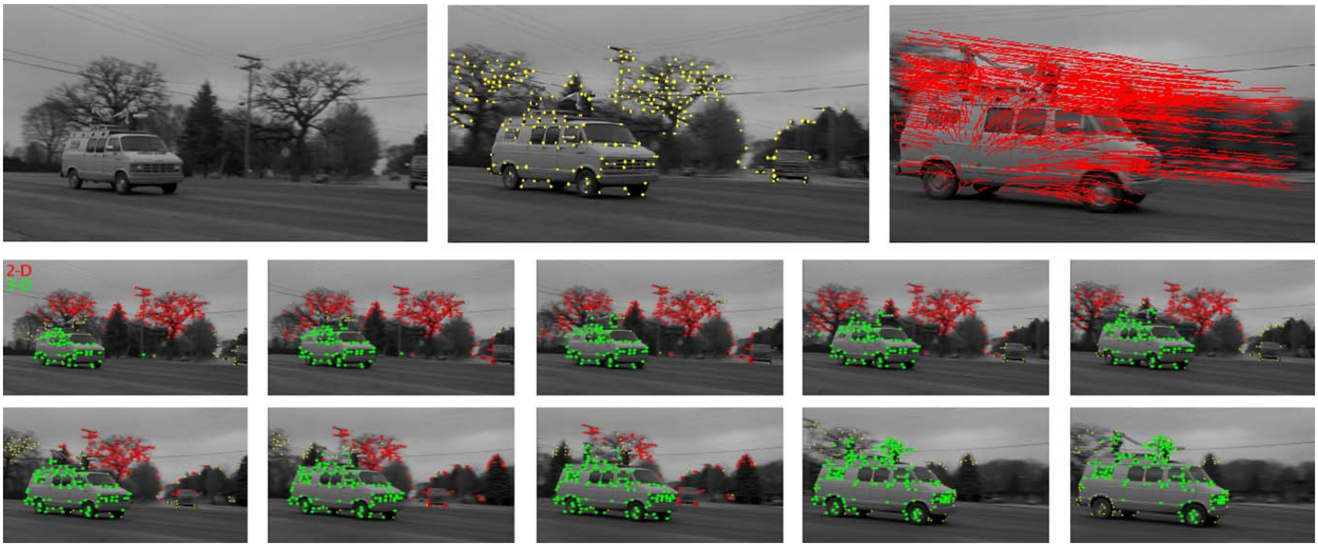


Fig. 10 Segmentation of the "delivery van" sequence. *Top*: first frame, fifth frame with feature points superimposed, last frame with feature tracks superimposed. *Bottom*: recovered segmentation through the sequence. The selected type of scene model is printed into the first frame of each selected motion

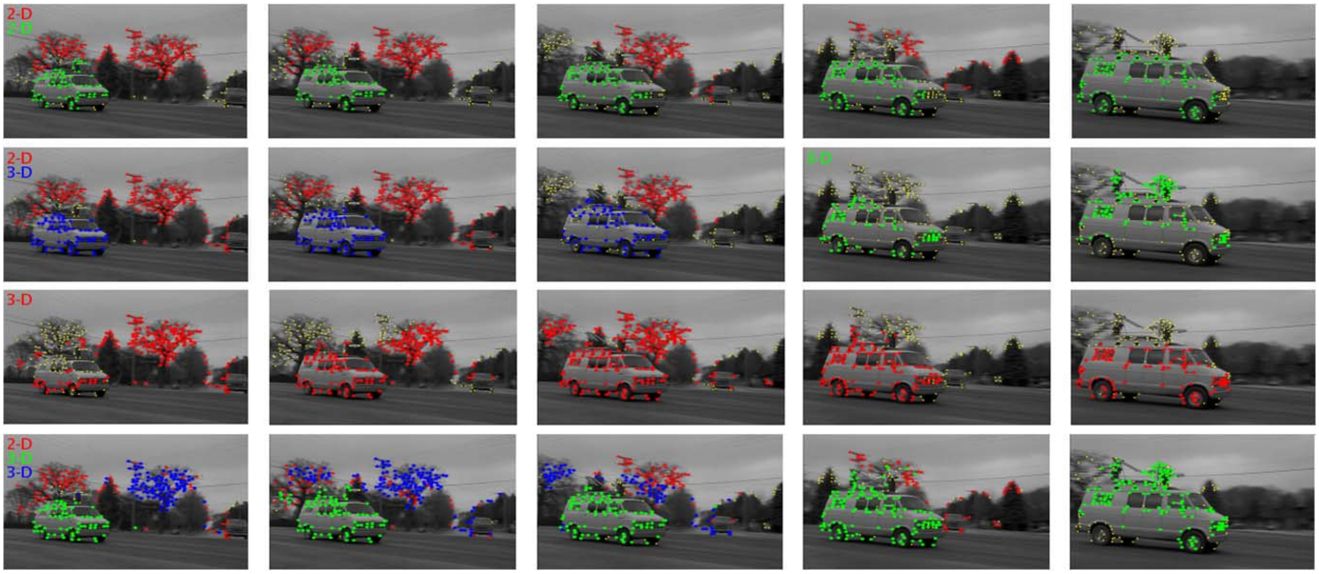


Fig. 11 Possible failures of model selection methods. The selected type of scene model is printed into the first frame of each selected motion. 1*st row*: too restrictive scene model. Only planar scenes were allowed, hence only the largest planar part of the van was detected. 2*nd row*: oversegmentation in time. The candidate set is incomplete because of overly restrictive linking, hence the van is split into two motions, and, the background model terminates one frame earlier. 3*rd row*: undersegmentation in space due to a grossly overestimated standard deviation. 4*th row*: oversegmentation in space. This case has been added for completeness, but did not occur in practice

fore assigns the points on the van, which are off its front plane, to the outliers.

Another source of failure are errors at the candidate generation stage: if model selection is not fed with the right candidates, it can only find the most reasonable solution with the motions at hand. If the candidate set does not contain an appropriate motion due to insufficient samples, then the corresponding object will usually be missed altogether, and its tracks will be classified as outliers. On the other hand, if all required two-view motions are present, a too restrictive linking procedure could still result in a candidate being broken into two, one for the first part of the sequence, and one for the second part. Model selection will typically select both parts, leading to oversegmentation in time—see Fig. 11.

Finally, in which cases does model selection itself fail, despite being supplied the correct input? The most severe

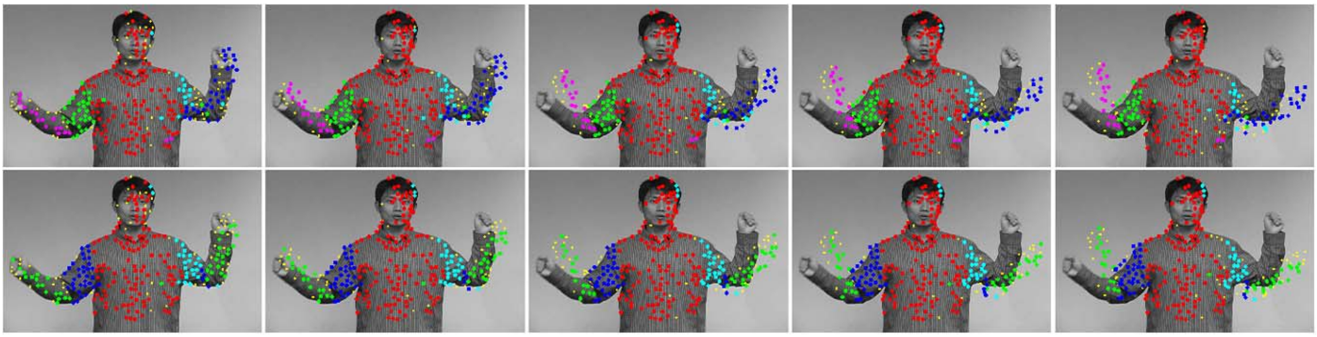


Fig. 12 Ambiguous cases. *Top/bottom*: segmentations returned by two different runs of the method for five frames taken from the "dance" sequence (affine camera model). The two forearms *together* undergo a nearly rigid motion. Depending on the randomly sampled candidate set, they are explained as one or two rigid motions. The model scores $\widehat{\mathcal{D}}$ for the two solutions differ by less than 4%. All results are overlayed on the same frame, the remainder of the video was not available

failure happens when the maximum allowable standard deviation is grossly overestimated. The system will then generate motions, which cover more than one object, at least partially. Due to the weak nature of the epipolar constraint, the penalty for the goodness-of-fit of such a model may not be large enough to outweigh the benefit of saving on model complexity, and the scene will be under-segmented. An example of this behavior is shown in Fig. 11. The allowable standard deviation was set to 3 pixels instead of 1, leading to an incorrect fit. *Over*-segmentation caused by the model selection criterion, while theoretically possible, is rarely a problem in practice. For the given sequence, we could only produce this case by unrealistically biasing the system against outliers: the size w of the matching window in (1) had to be set to 7000 pixels (10 times the image size) in order to justify an oversegmentation, which reduces the outlier count compared to the correct result (Fig. 11).

Remark One more situation should be mentioned, in which the approach may not give repeatable results. Model selection is built on the assumption that a reliable decision for one or the other model can be made, since there is no gradual transition from, say, a planar to a general scene model. However, it is quite common in practice that a scene has no clear explanation, but is on the borderline between different models. For example, the depth relative to the focal length is often such that the images are on the borderline between an affine and a perspective projection, or the depth range of the scene relative to the focal length is such that the scene is on the borderline between planar and full 3D. In these cases, the system does recover the correct number of motions and a good segmentation, but may not be able to reliably select the type of model, because the question which model is correct is ill-posed. The decision taken becomes dependent on small variations in the fitting residuals, standard deviation, and inlier numbers. Even the fluctuations caused by different random samples can change the decision. Note however that it usually does not matter, which way the decision goes, because the problem is caused precisely by the fact that there is no clearly better model to describe the scene.[6] The same considerations apply in the case where the relative motion between two objects is small, so that they move almost like one single rigid object. In that case, the two possible solutions (two independent motions, or one common motion with slightly higher reprojection errors) will obtain very similar scores, and fluctuations due to random sampling will determine the outcome. An example is given in Fig. 12.

5 Concluding Remarks

We have presented a generic scheme for multibody structure-and-motion of image sequences. The scheme is robust to outliers, can deal with unknown and varying number of moving objects, and with a set of correspondences, which changes over time. It recovers both the segmentation of the correspondences into different rigidly moving objects, and the structure of the underlying scene.

The method starts from atomic two-view motions and links them to tentative motions through the sequence, while constantly pruning redundant and overly unlikely motions to keep the size of the search space under control. In the final set of candidate motions, the best solution is found via model selection, and temporal coherence of the inlier sets is used to improve the segmentation.

[6] While the aim of an engineering application is to recover the model with the highest probability (the MAP solution), a strictly Bayesian view would be to recover the probability distribution over different models. The model scores, being essentially log-likelihoods, can be regarded as samples from that distribution, and the described situation, where no clear decision can be made, corresponds to a multi-modal distribution. A possible solution is to make the multi-modality explicit and return a set of possible solutions with their scores.

An important limitation of the method is that it is based on a set of candidate motions generated by random sampling, therefore it can handle only a small number of moving objects, because of the exponentially growing number of required samples. In this context, it should be mentioned that multibody structure-and-motion, as opposed to 2D tracking, only makes sense for a relatively small number of moving objects, because the 3D structure can only be recovered reliably for objects which subtend a sufficiently large viewing angle, and hence cover a reasonable part of the image plane.

Furthermore, the current implementation can only handle sequences of limited length (≈15 frames), because of the potentially exponential increase in candidates. Both these limitations are not caused by the model selection procedure, but by the hypothesis generator which feeds it. If candidates can be generated within tighter limits or subjected to a more rigorous pre-selection, the model selection framework can be applied to larger problems.

A more deep-rooted problem concerns objects with very few tracked points. The explicit outlier model introduces a limit for the smallest identifiable motion: there must come a point where it is cheaper to assign the small set of points to the outliers, rather than to add a more complex, hence more expensive motion model. A similar problem exists for object parts with very few points: for example, points on a small protrusion on a predominantly planar object will be assigned to the outliers, because the benefit of the simpler model outweighs the cost of a few outliers. The latter case can be remedied with recent methods developed in the context of RANSAC fitting (Chum et al. 2005; Frahm and Pollefeys 2006).

A point which needs to be addressed in more detail is to properly account for consistency, both in space and time. While we have used the assumption that tracks on the same object are clustered in space and continuous over time in an ad-hoc manner, we did not incorporate this prior belief into model selection. More research is needed to investigate how to integrate these constraints by means of a probabilistic prior.

Acknowledgements We would like to thank Rene Vidal for the "three cars" data, Josef Sivic and Andrew Zisserman for the "delivery van" data, Rene Vidal and Marc Pollefeys for the "dance" data, and James U for helping with the acquisition of the remaining sequences. This work has been carried out within the *Institute for Vision Systems Engineering*, funded by the *Faculty of Engineering, Monash University.*

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. Petrov, F. Csaki (Eds.), *Proceedings of the 2nd international symposium of information theory* (pp. 267–281).

Bozdogan, H. (1987). Model selection and Akaike's information criterion (AIC): the general theory and its analytical extensions. *Psychometrika*, *52*(3), 345–370.

Chum, O., Werner, T., & Matas, J. (2005). Two-view geometry estimation unaffected by a dominant plane. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 772–779), San Diego, CA.

Costeira, J., & Kanade, T. (1995). A multi-body factorization method for motion analysis. In *Proceedings of the 5th international conference on computer vision* (pp. 1071–1077), Boston, MA.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). New York: Wiley.

Faugeras, O., Luong, Q.-T., & Papadopoulo, T. (2001). *The geometry of multiple images*. Cambridge: MIT Press.

Fitzgibbon, A. W., & Zisserman, A. (2000). Multibody structure and motion: 3-d reconstruction of independently moving objects. In *Proceedings of the 6th European conference on computer vision* (pp. 891–905), Dublin, Ireland.

Frahm, J.-M., & Pollefeys, M. (2006). RANSAC for (quasi-) degenerate data (QDEGSAC). In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 453–460), New York.

Han, M., & Kanade, T. (2000). Reconstruction of scenes with multiple linearly moving objects. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 542–549), Hilton Head, South Carolina.

Hartley, R., & Zisserman, A. (2000). *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press.

Irani, M., & Anandan, P. (1998). A unified approach to moving object detection in 2D and 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(6), 577–589.

Kanatani, K. (2004). Uncertainty modeling and model selection for geometric inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(10), 1307–1319.

Kverh, B., & Leonardis, A. (2004). A generalisation of model selection criteria. *Pattern Analysis and Applications*, *7*(1), 51–65.

Langs, G., Peloschek, P., & Bischof, H. (2005). Optimal sub-shape models by minimum description length. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 310–315), San Diego, CA.

Leonardis, A., Gupta, A., & Bajcsy, R. (1995). Segmentation of range images as the search for geometric parametric models. *International Journal of Computer Vision*, *14*(1), 253–277.

Leontaritis, I. J., & Billings, S. A. (1987). Model selection and validation methods for non-linear systems. *International Journal of Control*, *45*(1), 311–341.

Li, T., Khallem, V., Singaraju, D., & Vidal, R. (2007). Projective factorization of multiple rigid-body motions. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 1–6), Minneapolis, MI.

Ma, Y., Kosecka, J., Soatto, S., & Sastry, S. (2003). *An invitation to 3-D vision*. Berlin: Springer.

Matsunaga, C., & Kanatani, K. (2000). Calibration of a moving camera using a planar pattern: optimal computation, reliability evaluation and stabilization by model selection. In *Proceedings of the 6th European conference on computer vision* (Vol. 2, pp. 595–609), Dublin, Ireland.

Maybank, S. J., & Sturm, P. F. (1999). MDL, collineations and the fundamental matrix. In *Proceedings of the 10th British machine vision conference*, Nottingham, UK.

Nemhauser, G., Wolsey, L., & Fisher, M. (1978). An analysis of approximations for maximizing submodular set functions I. *Mathematical Programing*, *14*, 265–294.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(6), 756–770.

Ripley, B. D. (1996). *Pattern recognition and neural networks*. Cambridge: Cambridge University Press.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 465–471.

Rissanen, J. (1984). Universal coding, information, prediction and estimation. *IEEE Transactions on Information Theory*, *30*, 629–636.

Schindler, K., & Suter, D. (2005). Two-view multibody structure-and-motion with outliers. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 676–683), San Diego, CA.

Schindler, K., U, J., & Wang, H. (2006). Perspective n-view multibody structure-and-motion through model selection. In *Proceedings of the 9th European conference on computer vision* (pp. 606–619), Graz, Austria.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 497–511.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell Systems Technical Journal*, *27*, 379–423.

Stricker, M., & Leonardis, A. (1995). ExSel++: a general framework to extract parametric models. In *Proceedings of computer analysis of images and patterns* (pp. 90–97).

Tomasi, C., & Kanade, T. (1991). Detection and tracking of point features. Technical report CMU-CS-91-132, Carnegie Mellon University.

Tong, W.-S., Tang, C.-K., & Medioni, G. (2004). Simultaneous two-view epipolar geometry estimation and motion segmentation by 4D tensor voting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(9), 1167–1184.

Torr, P. H. S. (1998). Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London A*, *356*(1740), 1321–1340.

Torr, P. H. S. (2000). Model selection for structure and motion recovery from multiple images. In A. Bab-Hadiashar & D. Suter (Eds.), *Data segmentation and model selection for computer vision*. Berlin: Springer.

Vidal, R., & Hartley, R. (2004). Motion segmentation with missing data using PowerFactorization and GPCA. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 310–316), Washington, DC.

Vidal, R., & Ma, Y. (2004). A unified algebraic approach to 2-D and 3-D motion segmentation. In *Proceedings of the 8th European conference on computer vision* (pp. 1–15), Prague, Czech Republic.

Vidal, R., Soatto, S., Ma, Y., & Sastry, S. (2002). Segmentation of dynamic scenes from the multibody fundamental matrix. In *Proceedings ECCV workshop on visual modeling of dynamic scenes*.

Wallace, C. S., & Boulton, D. M. (1968). An information measure for classification. *Computer Journal*, *11*(2), 185–194.

Wallace, C. S., & Freeman, P. R. (1987). Estimation and inference by compact coding. *Journal of the Royal Statistical Society Series B*, *49*(3), 240–265.

Wang, H., & Suter, D. (2004). Robust fitting by adaptive-scale residual consensus. In *Proceedings of the 8th European conference on computer vision* (pp. 107–118), Prague, Czech Republic.

Wills, J., Agarwal, S., & Belongie, S. (2003). What went where. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 37–44), Madison, WI.

Wolf, L., & Shashua, A. (2001). Two-body segmentation from two perspective views. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 263–270), Kauai, Hawaii.