

# Two-View Motion Segmentation from Linear Programming Relaxation

Hongdong LI  
RSISE, Australian National University  
VISTA, National ICT Australia

## Abstract

*This paper studies the problem of multibody motion segmentation, which is an important, but challenging problem due to its well-known chicken-and-egg-type recursive character.*

*We propose a new Mixture-of-Fundamental-matrices model to describe the multibody motions from two views. Based on the maximum likelihood estimation, in conjunction with a random sampling scheme, we show that the problem can be naturally formulated as a Linear Programming (LP) problem. Consequently, the motion segmentation problem can be solved efficiently by linear program relaxation. Experiments demonstrate that: without assuming the actual number of motions our method produces accurate segmentation result. This LP formulation has also other advantages, such as easy to handle outliers and easy to enforce prior knowledge etc.*

## 1. Introduction

Dynamic multibody scenes are very common in reality: e.g., traffic surveillance camera monitoring a busy intersection, sport camera tracking a group of soccer players, or a hand-held camera following a flying bird, etc. In the last case, the camera and the bird each contributes to an independent motion.

To enable a computer understanding a multibody dynamic scene, an efficient *multibody motion segmentation* algorithm is desirable. This problem, also known as the *multibody structure-and-motion* problem ([21][14][9][11]), consists of the following tasks: segmenting image points according to their motions, estimating individual motion's parameters, and recovering 3D structure of the points.

We propose a new algorithm in this paper, based on a principled framework of Linear Programming Relaxation. We assume that the camera model is fully projective (as oppose to linear affine camera models) so as to address close range applications where large perspective distortions appear in the images.

## 1.1. A brief review: existing approaches

Multibody motion segmentation is a challenging problem. This is mainly due to its well known chicken-and-egg-type character: in order to estimate multiple motions' parameters, one has better segment these motions first (i.e., find each point's membership); Reversely, to segment the multiple motions the information of each individual motion is much helpful.

**EM algorithm** is commonly adopted for solving such a recursive problem. It performs by alternating between parameter estimation and membership segmentation. Based on the EM algorithm, moderately successful applications have been reported [5][20]. However, the EM is only guaranteed to converge to a local minimum. Practices often show that an EM algorithm gets trapped into a local minimum thus produces erroneous segmentation.

**Subspace separation** has been suggested for motion segmentation. The most known algorithm is the multibody factorization due to Costeria and Kanade [9]. There are much other incremental work, some of them have substantially improved this technique (c.f., [3] [26][25] [5][16]). Unfortunately, without nontrivial modification these methods have to confine themselves only to linear camera model.

**Model selection** is another adopted method for the problem [19][10]. Torr [19] proposes an algorithm based on removing single motion (inliers) sequentially according to a residual analysis. Schindler et al's work [15] represents some recent development along this direction. A drawback of this method is that many parameters (e.g., threshold) need to be turned simultaneously.

**GPCA** is an interesting algebraic method, proposed by Vidal et al [22][8]. It is elegant and has broad applications. However, when used to segment multiple of general motions (say,  $m$  motions,  $m > 5$ ) it requires a large number of feature points, say, in the order of  $O(m^4)$ , which is impractical in many circumstances. Furthermore, being algebraic in nature, the method is vulnerable to outliers.

In the present paper, we propose a novel algorithm for two-view multibody motion estimation. The algorithm is based on the Linear Programming framework, thus has a

guaranteed global optimality. It needs not an initial estimate of the the number of independent motions, and can handle a large number of motions. Moreover, it deals with outliers naturally and under the same framework with little (computational) overhead.

## 2. Optimal Motion Segmentation

### 2.1. Optimal criterion

In this paper we formulate the multi-body motion segmentation problem as a global optimization. To achieve a truly *globally-optimal* motion segmentation two issues must be addressed. For one thing, the recovered motion models must fit well with the feature points; for the other, the estimated number of motions must reflect the physical fact. To this end, we need to design a proper optimal criterion (i.e., an objective function). Akaike Information Criterion (AIC) and the Maximum Likelihood principle are adopted in this paper. The choice of the AIC is only for its simplicity but not essential. The user may replace it with any other criterion, such as the BIC [19][10], when necessary.

Specifically, we are seeking a *global optimum* which minimizes the following AIC criterion:

$$J = -2 \log L + 2C \quad (1)$$

where  $L$  is a *likelihood* term (i.e. *data term*) measuring how well the motion models explain the data, and  $C$  is a *complexity* term measuring how complex the models are.

In the present work,  $C$  is proportional to the number of motions, i.e.,  $C \propto m$ . It remains to derive the likelihood term. Under a perspective camera model, a pair of matching points from two images,  $\mathbf{x}_i$  and  $\mathbf{x}'_i$ , are related by the epipolar equation [6]:  $\mathbf{x}'_i{}^T \mathbf{F} \mathbf{x}_i = 0$ , where  $\mathbf{F}$  is the fundamental matrix (FM). For any FM, we have  $\det(\mathbf{F}) = 0$ ,  $\mathbf{F} \in \mathbb{R}^{3 \times 3}$ .

As noise is unavoidable, the right-hand-side is nonzero, and can be used as a metric of the estimation—so-called *algebraic distance*. It is well known that the algebraic distance is problematic in practice. Another more popular, better metric, is the (squared) Sampson's distance [19][6], given by

$$d_s(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{F}_k) = \frac{(\mathbf{x}'_i{}^T \mathbf{F}_k \mathbf{x}_i)^2}{(\mathbf{F}_k \mathbf{x}_i)_1^2 + (\mathbf{F}_k \mathbf{x}_i)_2^2 + (\mathbf{F}_k^T \mathbf{x}'_i)_1^2 + (\mathbf{F}_k^T \mathbf{x}'_i)_2^2}. \quad (2)$$

In this formula, we explicitly denote the FM by  $\mathbf{F}_k$  to indicate that  $\{\mathbf{x}_i, \mathbf{x}'_i\}$  belong to motion- $k$ . In reality, before the motions are segmented we have no knowledge of which point belongs to which motion and how many motions are involved. To circumvent this, Vidal et al's GPCA (see also Wolf and Shashua [24]) proposes a concept of multibody-fundamental-matrix (mFM) by constructing the product of all possible motion membership assignments:

$\prod_{k=1}^m (\mathbf{x}'_i{}^T \mathbf{F}_k \mathbf{x}_i) = 0$ . This mFM equation always holds regardless of from which motion the image points actually arise. But the number of terms of the product grows dramatically as the number of matches ( $n$ ) or the number of motions ( $m$ ) increases.

### 2.2. Mixture of fundamental matrices (MoF)

In reality, we do not know how to actually write eq.(2) of  $d_s(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{F}_k)$ , because whether a matched pair  $(\mathbf{x}_i, \mathbf{x}'_i)$  does arise from the motion  $\mathbf{F}_k$  is not known *a-priori*.

To circumvent the obstacle, we propose a new model—called mixture-of-fundamental-matrix model (MoF). This model is inspired by the EM algorithm.

Define binary *membership variables*  $z_{ik}$  with  $z_{ik} = 1$  if the image point- $i$  is from motion- $k$ , and  $z_{ik}=0$  otherwise.

Under the condition of  $\sum_{k=1}^m z_{ik} = 1$ , we can replace the conventional FM equation  $\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$  with the following mixture-of-fundamental-matrices (MoF) form,

$$\mathbf{x}'_i{}^T \left( \sum_{k=1}^m z_{ik} \mathbf{F}_k \right) \mathbf{x}_i = 0. \quad (3)$$

This equation is equivalent to  $\sum_{k=1}^m (\mathbf{x}'_i{}^T z_{ik} \mathbf{F}_k \mathbf{x}_i)^2 = 0$ , and it always holds regardless of the actual image point memberships.

Analogously, we can replace the algebraic distance with the Sampson's distance, and obtain the following mixture-of-Sampson's-distance (MoS) formula,

$$d_{mos}(\mathbf{x}_i, \mathbf{x}'_i) = \sum_{k=1}^m z_{ik} d_s(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{F}_k). \quad (4)$$

Using *finite mixture model* is not a new concept in computer vision, even in the context of motion segmentation. Torr [19] and Weiss [5] had exploited this idea before. Nevertheless, the idea of mixing a set of independent fundamental matrices (each of which corresponds to a distinct motion) has not been reported previously. More remarkably, we will show later that such an MoF (or MoS) model, when incorporated into the maximum likelihood estimation framework, naturally leads to a simple Linear Program formulation.

### 2.3. Maximum Likelihood Estimation

With the aid of the MoF model, now we are allowed to formally write the two terms of the AIC criterion function, as described below.

Under the Gaussian noise assumption, a matched image pair  $(\mathbf{x}_i, \mathbf{x}'_i)$ , given motion models  $\mathbf{F}_k$  and memberships  $z_{ik}$ ,

will contribute to a *likelihood* term  $p$  as:

$$\begin{aligned} p(\mathbf{x}_i, \mathbf{x}'_i | \mathbf{F}_k, z_{ik}, k = 1, \dots, m) \\ = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_{mos}(\mathbf{x}_i, \mathbf{x}'_i)}{\sigma^2}\right) \\ = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\sum_{k=1}^m z_{ik} d_s(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{F}_k)}{\sigma^2}\right) \end{aligned} \quad (5)$$

At this stage we temporarily assume that the number of motions  $m$  is known.

Consider all  $n$  image point matches. The complete likelihood function  $L$  is thus (assuming statistical independency):

$$\begin{aligned} L(\mathbf{x}_i, \mathbf{x}'_i, i = 1..n | \mathbf{F}_k, z_{ik}, i = 1..n, k = 1..m) \\ = \prod_{i=1}^n p(\mathbf{x}_i, \mathbf{x}'_i | \mathbf{F}_k, z_{ik}, k = 1..m). \end{aligned} \quad (6)$$

Substituting this into eq.(1) and after some algebraic simplifications, we reach the following **minimization** problem:

$$\begin{aligned} \min_{\mathbf{z}, \mathbf{F}, m} J(z_{ik}, \mathbf{F}_k, m | i = 1..n, k = 1..m) \\ = \min_{\mathbf{z}, \mathbf{F}, m} \sum_{i=1}^n \sum_{k=1}^m z_{ik} d_s(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{F}_k) + \beta m, \end{aligned} \quad (7)$$

$$\begin{aligned} \text{such that,} \\ m \geq 1, \end{aligned} \quad (8)$$

$$z_{ik} \in \{0, 1\}, \sum_{k=1}^m z_{ik} = 1, \text{ for } i = 1..n, k = 1..m, \quad (9)$$

$$\det(\mathbf{F}_k) = 0, \text{ for } k = 1..m. \quad (10)$$

Recall that the last term of the AIC criterion  $\mathbf{C}$  represents the model's complexity. Here we simply let  $\log(\mathbf{C}) = \beta m$ , where  $m$  is the (unknown) number of motions and  $\beta$  a trade-off factor. This amounts to say that: more motions are considered more complex.

Now that the multibody motion segmentation problem is converted to a typical ML optimization problem:

- Find the best  $\mathbf{F}_k$ ,  $z_{ik}$  and  $m$ , so that the above cost function is minimized.

## 2.4. Convert to Linear Programming Problem

However, solving the above optimization problem is very hard. This is because: the cost function (7) itself is highly nonlinear and non-convex in the unknowns  $\mathbf{F}_k$ ; the det constraint of (10) is cubic and non-convex; the variables  $z_{ik}$  are binary integers. Besides, even the number of motions  $m$  is unknown.

The readers who are familiar with the EM algorithm might recognize that the form of (7) is similar to the *complete-data log-likelihood* in EM, hence may wonder

whether a *marginalization* (as the EM does) over the unknown  $z_{ik}$  would be of any help. We think this is a promising direction for future work.

In this work we explore a different direction based on Linear Programming Relaxation idea, which is non-conventional and very effective.

Examine eq.(7) again. Suppose that somehow we already have a list of  $M$  *candidate* motion models, denoted by  $\Phi = \{\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_M\}$ ,  $|\Phi| = M$ . We assume that the candidate list does contain the  $m$  true motions.

Define auxiliary binary *indicating variables*  $y_k$  with  $y_k=1$  if the motion- $k \in [1, \dots, M]$  is indeed one of the  $m$  true motions and  $y_k=0$  otherwise. Clearly we have  $\max_{i=1}^n \{z_{ik}\} = y_k$  and  $\sum_{k=1}^M y_k = m$ . Using these notations we re-write eq.(7) as

$$\begin{aligned} \min_{\mathbf{z}, \mathbf{F}, \mathbf{y}} J(z_{ik}, \mathbf{F}_k, y_k | i = 1..n, k = 1..M) \\ = \min_{\mathbf{z}, \mathbf{y}} \sum_{i=1}^n \sum_{k=1}^M z_{ik} d_s(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{F}_k) + \beta \sum_{k=1}^M y_k \\ = \min_{\mathbf{z}, \mathbf{y}} \sum_{i=1}^n \sum_{k=1}^M z_{ik} d_{sik} + \beta \sum_{k=1}^M y_k. \end{aligned} \quad (11)$$

In this formulation, if we assume that the  $M$  candidate motions have already been found somehow, and they do contain the  $m$  true motions, then the terms of  $d_s(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{F}_k)$  can be pre-computed and considered as known coefficients (of  $z_{ik}$ ). Now the problem becomes linear in the unknown binary variables  $z_{ik}$  and  $y_k$ ,  $k = 1, \dots, M$ . In fact, it is a standard 0-1 Integer-Linear-Programming problem.

## 3. Facility Location Problem and Relaxation

So far we have successfully reduced the problem to an integer linear programming (under the assumption that all the candidate motions  $\mathbf{F}_k$  are known beforehand). Now a question arises: how to solve the problem?

To *exactly* solve an integer linear programming is extremely hard (it is NP-hard in general). By *exactly*, we mean that all the unknowns (to be solved) are well constrained to be integers.

Before proceeding to explain how we actually solve the NP-hard problem, we make a detour and introduce the *facility location problem* (FLP)—more precisely the uncapacitated FLP—a well-known subject of Operations Research ([1][17]). The reason for such a detour will become clear shortly.

Imagine a big retailer company plans to open some local shops (i.e. *facilities*) to serve  $n$  customers. The locations of all customers are known beforehand. The locations of shops are not known but to be chosen from a set of candidate sites, denoted by  $\mathcal{F}$ . The number of shops to be opened (denoted by  $m$ ) is initially unknown. Suppose there is a

fixed cost  $\beta$  associated with the opening of a new shop, and a transportation cost  $d_{ik}$  if customer  $i$  is served by shop  $k$ .

The problem of FLP is then to decide which shop to open, and which shop serves each customer so as to minimize the sum of the opening and transportation costs.

We introduce opening indicating variables  $y_k = 1$  if shop  $k$  is chosen to open and  $y_k = 0$  otherwise, also introduce membership variables  $z_{ik} = 1$  if customer  $i$  is served by shop  $k$ . Then, the FLP problem is formally written as:

$$\min_{\mathbf{z}, \mathbf{y}} \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{F}} z_{ik} d_{ik} + \beta \sum_{k \in \mathcal{F}} y_k, \quad (12)$$

such that,

$$\forall i, k, z_{ik} \leq y_k, \quad (13)$$

$$\forall i, \sum_{k \in \mathcal{F}} z_{ik} = 1, \quad (14)$$

$$\forall i, k, z_{ik} \in \{0, 1\}, y_k \in \{0, 1\}. \quad (15)$$

The inequality (13) ensures that a customer can only be served by an opened shop.

Comparing this FLP problem with eq.(11) we find that they are essentially identical. Both are challenging (NP-hard) integer linear programming problems.

In practice, in order to tackle NP-hard problems various strategies have been proposed (e.g. [7]). Among them, the **relaxation** is one of the most promising approaches, which has been applied successfully, particularly to the FLP problem.

The key idea of linear program relaxation (LPR) is to *relax* the 0-1 integer constraints (15) into some linear inequalities of continuous variables:

$$\forall i, k, 0 \leq \hat{z}_{ik} \leq 1 \text{ and } 0 \leq \hat{y}_k \leq 1. \quad (16)$$

**Rounding; de-relaxation.** Now that the integer programming problem becomes a Linear Programming problem, one can solve it easily by any standard LP solver. It is further *hoped* that the solved real variables of  $\hat{z}$  and  $\hat{y}$  are very close to 0 or 1. Therefore, a rounding procedure needs to be applied. A natural rounding scheme is: let  $z = 1$  if  $\hat{z} > 0.5$  and  $z = 0$  otherwise. More delicate (and better) rounding schemes can be found in [7] [17].

## 4. The Proposed Segmentation Algorithm

The high-level outline of our new motion-segmentation algorithm is given as follows.

### 4.1. Algorithmic description

1. Generate a set of candidate motions by certain method. Store them in a list  $\Phi$ . The size of  $\Phi$  is chosen so that  $M \gg m$  where  $m$  is a rough estimate of the number of motions. During this step, one must ensure that the

correct  $m$  motion models are indeed contained in the list.

2. Compute the Sampson's distances  $d_{ik}$  between each image point pair  $i$  and each candidate motion  $k$ . Establish a relaxed FLP problem, and solve it with any LP solver (e.g. CPLEX or Matlab's `linprog`).
3. Round the LP solutions so that the resulting variables are binary. Decode and output the results.
4. Using the estimated memberships  $z_{ik}$  to re-compute refined fundamental matrices. Output segmentation results and fundamental matrices.

### 4.2. Remarks

- The number of motions can be estimated from  $m = \sum_{k=1}^M y_k$ . The trade-off factor  $\beta$  will affect the estimated  $m$ . However, in experiments we found  $\beta$  can be chosen from a wide range (see experiment section).
- As long as the true motions are indeed included in the list  $\Phi$ , the algorithm will converge to the global optimum. Modern LP solvers are generally considered high efficient (e.g. having only polynomial-time complexity). However, to solve a large-scale LP problem is still computationally expensive. To reduce the size of  $\Phi$  as much as possible is desired.
- If two candidate motions containing in the list  $\Phi$  are very resemble to each other, then after convergence the LP solver will automatically pick the one that gives rise to the smallest residual error. This is assured by the global optimality of LP.

## 5. Guided Random Sampling

In previous section, we make a crucial assumption that a set of candidate motions is known beforehand. This section gives an effective method to generate such a candidate set. Our method is based on random sampling, which is in essence similar to the RANSAC.

Note that the two-view motion (i.e. the FM) can be calculated quickly from minimal 7 or 8 points. Furthermore, if the camera is calibrated then 5 points (or 6 points for semi-calibrated case) are sufficient [13] [12]. All these suggests that the random sampling scheme can be performed efficiently.

### How many samples?

Similar to the probabilistic analysis of the RANSAC, there is also a theoretical guarantee that, with a confidence level of  $p$  (e.g.,  $p=0.9$ ) it is assured that one can obtain at least one motion from each of the motion groups after a sufficient number of  $N$  samplings.



Specifically, assume the smallest motion group takes up a ratio of  $\epsilon$ , then this number of least required samplings to ensure, with a probability  $p$ , that at least one sample from each of the motion groups has been captured, is given by,

$$N = \log(1 - p) / \log(1 - \epsilon^s), \quad (17)$$

where  $s$  is the size of the minimal-set required for computing a camera motion (e.g.,  $s = 7$  if the 7-point algorithm is adopted).

**Guided Sampling.** Being theoretically correct, the above result however, appears to be over-optimistic ([18][4]). In practice, due to image noise etc., the actually required samples are often much more than predicted. In particular, for the multi-body case the required number of samples often grows too large to be computationally tractable. In addition, for LP's sake we also expect a short list of the candidate motions. To accelerate the process of sampling, while still guarantee the quality of the candidates, we suggest the following mechanisms:

- *Spatial coherence.* Unless in rare cases (e.g., transparency), motion vectors belonging to a single object often spread contingently. This spatial coherency is very useful for sampling. Paper [15] introduces an image-partition technique which significantly increases the chance of obtaining good samples.
- *Prior distribution.* Conventional RANSAC samples image points uniformly with a equal probability. However, in reality, some data are born to be better than the other. Before computing the residual error, there is plenty of prior information that can be used to guide the sampling. Paper [18] has exploited such idea and has achieved significant improvements (e.g, 1 – 2 orders of magnitude faster).
- *Chirality constraints.* Projective chirality basically says that not every group of seven points produces a valid fundamental matrix. Some may correspond to the case when 3D scene points are behind the camera which is not possible in reality (using a real camera). Based on the chirality constraint Chum and Werner et al obtained a two-fold speed-up [2].
- *Motion Clustering.* It is conceivable that, among some spurious motion models there are also many duplications existing in the candidate motion list. Clustering on these candidate motions effectively removes redundancy, and reduces the computational burden. We adopt the inlier-outlier pattern method ([23]) which works well and is easy to implement.

Finally, we want to emphasize that: although we suggest the above guidelines, they are however, not essential. They

are meant to reduce the computational cost. In fact, our algorithm would work well on purely randomly sampled data set. An industrial-strength LP solver (e.g. CPLEX) is capable of solving an LP problem with millions of variables.

## 6. Dealing with outliers

Finding *fully correct* image matches is not a simple task, especially when multiple motions are present. For this reason, it is desirable that a motion-segmentation algorithm be robust to outliers.

Our algorithm can handle outliers in a uniform fashion. Little modification is needed to the original structure of the algorithm. This is another featured advantage of our method.

We simply add a virtual motion model (called the *outlier motion*) to the candidate list. This ‘motion’ can match any feature point at *nil* cost (alternatively one can introduce a fixed cost to any such match). In order to better regularize the problem, we add an upper bound  $\gamma$  to the total number of outliers, and an upper bound  $\alpha$  to the total number of true motions. In practice, guessing such an upper bound is not hard, for example, by prescribing a small portion that accounts for the outliers.

Reload some notations:  $\mathcal{F} = [1..M]$ ,  $\mathcal{C} = [1..n]$ . Define  $w_i = 1$  if point  $i$  is an outlier image point and  $w_i = 0$  otherwise. Now the **LPR with outliers** problem can be formulated as the following form, where the virtual *outlier motion* is explicitly expressed.

$$\begin{aligned} \min_{\mathbf{z}, \mathbf{y}, \mathbf{w}} \quad & \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{F}} z_{ik} d_{ik} + \beta \sum_{k \in \mathcal{F}} y_k, \quad (18) \\ \text{such that,} \quad & \\ & \forall i, k, z_{ik} \leq y_k, \\ & \forall i, \sum_{k \in \mathcal{F}} z_{ik} + w_i = 1, \\ & \sum_{i \in \mathcal{C}} w_i \leq \gamma, \\ & \sum_{k \in \mathcal{F}} y_k \leq \alpha, \\ & \forall i, k, 0 \leq z_{ik} \leq 1, 0 \leq y_k \leq 1, 0 \leq w_i \leq 1. \end{aligned}$$

Solving this LP simultaneously gives both the motion segmentation and outlier detection results. Our experiments have demonstrated the effectiveness of the method.

## 7. Experiments

### 7.1. Simulation

To validate our theory and the algorithm, we have conducted experiments on both simulated and real images.

We generate perspective images matches. The image size is about 512 by 512. Gaussian noise of std. of 0.2-1.0

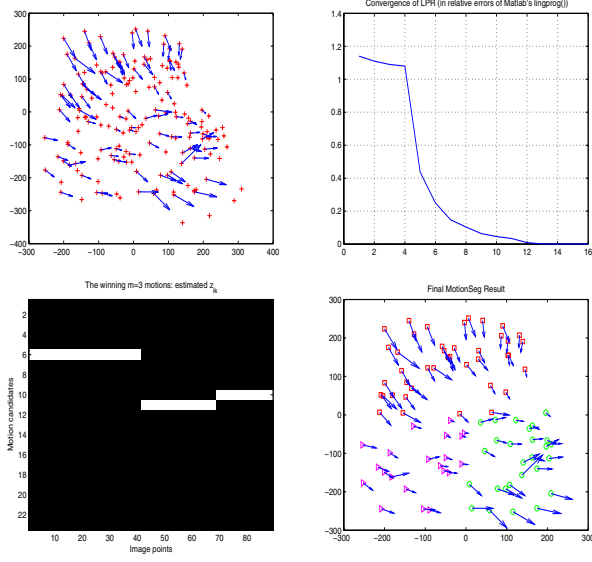


Figure 1. Motion Segmentation Experiment: 3 motions, 100 points. TL: input feature matches; TR: convergency of the linear programming; BL: obtained membership variables ; BR: motion segmentation result (up to permutations); Different groups are denoted by different symbols or colors.

pixels are added to the coordinates. Outliers are simulated by an extra uniform noise within range [1..10] pixels. The number of feature points is about 60-200. We use the 8-point algorithm as the minimal solver in order to produce a unique candidate motion (as opposed to 3 motions if the 7-point algorithm is used). We use Matlab's `linprog` as the LP solver. The number of true motions is generally within range [2-10], but this information is not known to the algorithm.

In the **first experiment** we test a 3-motion case. For an instance, fig-1 top-left gives the input feature matches. Gaussian noise was added to the point coordinates. After about 12 iterations the LP solver converges, as shown in fig-1 top-right. Fig-1 bottom-left illustrates the estimated membership variables  $z_{ik}$ , which clearly reveals exactly 3 motions, and the image points have been grouped into the three motions correctly. Note that we purposely arranged image feature points so that the memberships are in order. This makes the visual evaluation task easy.

Fig-1 bottom-right shows the motion segmentation result which is 100% correct according to the ground truth. In this experiment, we use  $n=100$  feature points,  $M=24$  candidate motions (after applying spatial partition and motion clustering). We also test cases with different levels of noise.

In the **second experiment**, we tested cases with many motions. One purpose is to evaluate how the value of  $\beta$  affects the estimate of  $m$ , i.e. number of motions. We have consistently found that within a wide range (e.g.

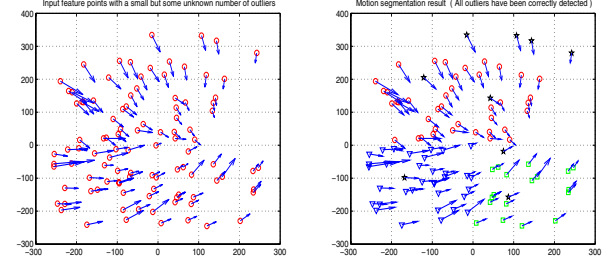


Figure 3. Motion segmentation with outliers. Left: Input feature points with some unknown number of outliers; Right: Segmentation result. All outliers (denoted by  $\star$ ) have been detected correctly according to ground-truth.

[0.00001,0.1]) of the  $\beta$  the estimates are almost always correct. There are a few failures which, in our opinion, are mainly due to the random sampling stage of our algorithm. We have found that, when  $\beta$  is over large (e.g.,  $> 0.1$ ) the algorithm tends to obtaining less motions; on the contrary, when  $\beta$  is over small (e.g.,  $< 1e-5$ ) it tends to obtaining more motions. This is not surprising, as the  $\beta$  is a trade-off parameter and itself represents a fixed cost which penalizes the inducing of a new motion (i.e., opening a new facility in the FLP version). In practice, a rule-of-thumb is let  $\beta$  equal to the median value of all Sampson's distances, which works well in most of our tests.

Fig-2 gives the results for a 5-motion case and 9-motion case. Noise level was 1.0 pixels. Our algorithm automatically recovers  $m$  and successfully segments motions. For the 9-motion case, only two errors (out of 141 points) are produced. **A Remark:** if the GPCA method were used for segmenting  $m = 9$  general motions, then at least  $N=[(m+2)(m+1)/2]^2 - 1 = 3024$  outlier-free point matches must be acquired, which is not so practical in reality.

In the **third experiment**, we add 10% outliers to the data set by perturbing point coordinates by 5-pixel uniform noise. We test our LPR-with-outlier formulation (i.e., eq.(18)). In all simulations we use an upper bound  $\gamma = (15\%) \times n$ , and an upper bound  $\alpha \approx (m + 10)$ . Good results are obtained: the outliers have been exactly detected, and the obtained segmentations are correct. One result is shown in fig-3 for instance, where we added eight outliers to the data. All the outliers (denoted by  $\star$ ) have been detected correctly. This has validated our LP-based outlier-handling method.

## 7.2. Test on real images

We test our algorithm on some real images used by vision researchers for the multi-body motion segmentation research.

We have applied the outlier-free version LPR algorithm to Vidal's 3-cars sequence (fig-4), Kanatani's Car-1 and

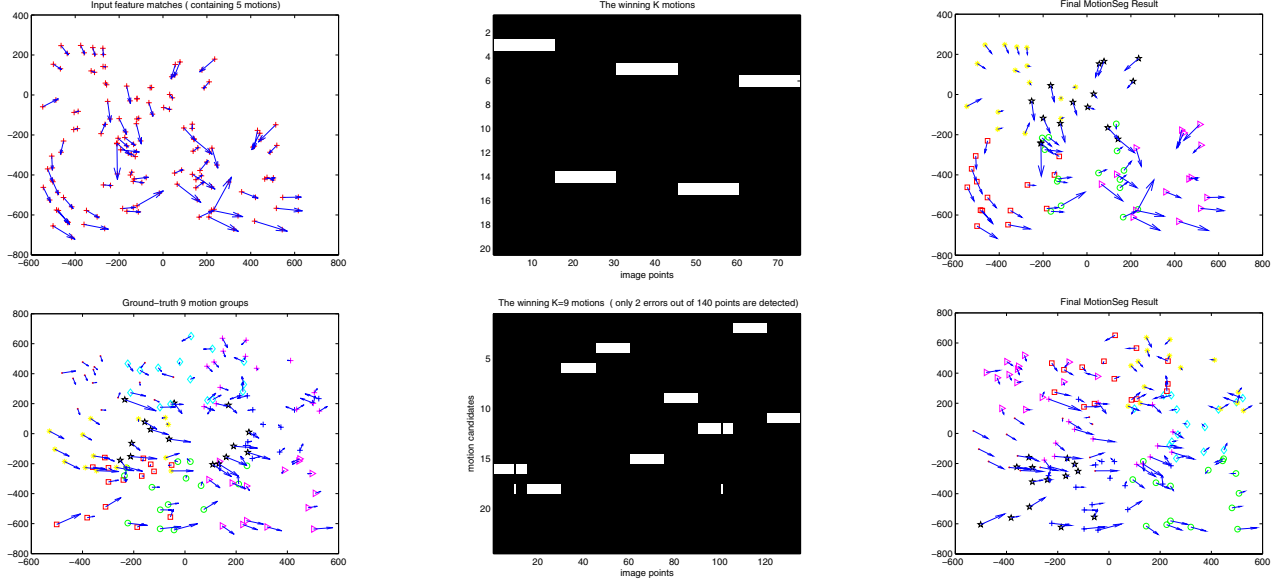


Figure 2. Motion segmentation experiments (5 motion case and 9 motion case); Left: input features ; middle: recovered memberships ; right: segmentation results (up to permutations).

Car-2 sequences (fig-5). Our results compare favorably with the state-of-the-art technique.

One of the advantages of the proposed method is that: it accomplishes both the tasks of motion number estimation and motion segmentation simultaneously, and under a unified linear programming framework. By contrast, many conventional algorithms accomplish these two tasks sequentially, i.e. first estimate the number of motion, then do the segmentation. We believe that the former fashion is advantageous over the latter, as it helps finding the truly globally optimal solution. Moreover, by the linear programming framework the user's some prior knowledge about the scene can be incorporated into the computation via a unified manner.

## 8. Discussion and Conclusion

**The roles of MoF and LP.** Without the new MoF model, we would have not reached the linear programming formulation (7). Philosophically speaking, this mixture-model allows each image point simultaneously belonging to multiple motion groups (by using fractional mixing weights). The idea of pre-computing a set of candidate motions has reduced the otherwise complicated log-likelihood function, and effectively linearized the nonlinear and non-convex problem. The role of LP is then to select from all the candidate motions the best combination that produces the least residual error.

**Incorporate prior knowledge.** Under the LPR framework the user can easily enforce other prior constraints to the

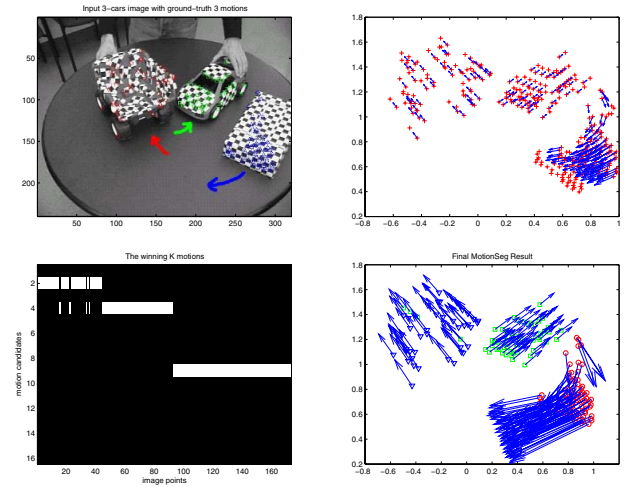


Figure 4. Experiment on the 3-car sequences. TL: one of the input images with segmentation ground truth; TR: input feature matches; BL: estimated memberships. Only 4 mis-classified points; BR: Final segmentation result.

computation. For example, information such as two points are from the same class can be easily expressed as a linear constraint, and adds little to the computational complexity of the resulting LP problem. With the aid of prior knowledge, our algorithm is likely to perform better.

**General segmentation problem.** Conceivably, our algorithm can be applied to estimating the multiple lines (i.e., linear subspaces) or multiple circles (i.e., nonlinear mani-



Figure 5. Experiment on Kanatani's two car sequences. Left: one image with detected feature points; right: final segmentation result.

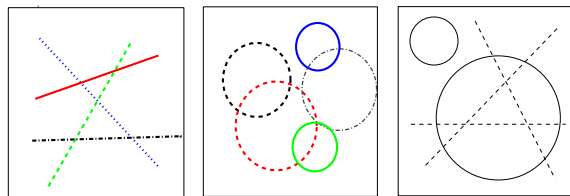


Figure 6. General multi-model segmentation problem.

folds) or even their mixture in figure-6. Formally speaking, the algorithm is applicable to the general multiple-model segmentation problem, given that each of the model's analytic form is known and preferably has a small number of parameters.

**Further applications.** We also envisage other possible applications of the algorithm. For example, in some circumstances applying the EM algorithm can be difficult, e.g, due to lack of a close-form E-step or M-step. Our suggestion is: whenever the user can build up a candidate list containing the true solutions at reasonable cost, then our LPR approach could be used as an alternative to the EM.

**Acknowledgement.** The author wishes to thank R.Hartley for many supports and inspiring discussions. Thank the anonymous reviews and the AC for invaluable comments. Thank C Shen for inspiring discussions. The real image data is courtesy of R Vidal and is greatly acknowledged.

## References

- [1] F. Chudak and D. Shmoys. Improved approximation algorithms for uncapacitated facility location problem. *SIAM J. Comput.*, pages 1–25. 3
- [2] O. Chum, T. Werner, and J. Matas. Epipolar geometry estimation via ransac benefits from the oriented epipolar constraint. *ICPR*, 2004. 5
- [3] Z. Fan, J. Zhou, and Y. Wu. Multibody motion segmentation based on simulated annealing. *CVPR*, 1:776–781, 2004. 1
- [4] L. Goshen and I. Shimshoni. Guided sampling via weak motion models and outlier sample generation for epipolar geometry estimation. *CVPR*, 1(1):1105–1112, 2005. 5
- [5] A. Gruber and Y. Weiss. Incorporating non-motion cues into 3d motion segmentation. *ECCV*, 2006. 1, 2
- [6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 2
- [7] D. Hochbaum. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, Boston, MA, 1997. 4
- [8] K. Huang, R. Vidal, and Y. Ma. Minimum effective dimension for mixtures of subspaces: A robust gpca algorithm and its applications. *CVPR*, 2004. 1
- [9] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. *Proc. of ICCV*, pages 1071–1076, 1995. 1
- [10] K. Kanatani. Geometric information criterion for model selection. *IJCV*, 26(3):171–189, 1998. 1, 2
- [11] K. Kanatani. Motion segmentation by subspace separation and model selection. *ICCV*, 2:301–306, 2001. 1
- [12] H. Li. A simple solution to the six-point focal-length problem. *ECCV*, 2006. 4
- [13] H. Li and R. Hartley. Five-point motion estimation made easy. *ICPR*, 2006. 4
- [14] K. Schindler and D. Suter. Two-view multibody structure-and-motion with outliers through model selection. *IEEE T-PAMI*, 28(6):983–995, 2006. 1
- [15] K. Schindler, J. U. and H. Wang. Perspective n-view multibody structure-and-motion through model selection. *ECCV*, 1:606–619, 2006. 1, 5
- [16] J. Shi, S. Belongie, T. Leung, and J. Malik. Image and video segmentation: The normalized cut framework. *ICIP*, 1:943–947, 1998. 1
- [17] D. Shmoys. Approximation algorithms for facility location problem. *Approximation Algorithms for Combinatorial Optimization*, 1913:27–33, 2000. 3, 4
- [18] B. Tordoff and D. W. Murray. Guided sampling and consensus for motion estimation. *ECCV*, pages 82–98, 2002. 5
- [19] P. Torr. Geometric motion segmentation and model selection. *Philosophical Trans. Royal Soc*, 356(1740):1321–1340, 1998. 1, 2
- [20] N. Vasconcelos and A. Lippman. Empirical bayesian em-based motion segmentation. *CVPR*, 00:527, 1997. 1
- [21] R. Vidal and R. Hartley. Motion segmentation with missing data using powerfactorization and gpca. *CVPR*, 2004. 1
- [22] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multibody structure from motion. *IJCV*, 68(1):7–25, 2006. 1
- [23] J. Wills, S. Agrawal, and S. Belongie. What went where. *CVPR*, pages 98–104, 2003. 5
- [24] L. Wolf and A. Shashua. Two-body segmentation from two perspective views. *CVPR*, pages 263–270, 2001. 2
- [25] J. Yan and M. Pollefeys. A factorization-based approach to articulated motion recovery. *CVPR*, pages II: 815–821, 2005. 1
- [26] L. Zelnik-Manor, M. Machline, and M. Irani. Multi-body factorization with uncertainty: Revisiting motion consistency. *IJCV*, 68(1):27–41, 2006. 1