Multibody Structure-and-Motion Segmentation by Branch-and-Bound Model Selection

Ninad Thakoor, *Student Member, IEEE*, Jean Gao, *Member, IEEE*, and Venkat Devarajan, *Senior Member, IEEE*

***Abstract*—An efficient and robust framework is proposed for two-view multiple structure-and-motion segmentation of unknown number of rigid objects. The segmentation problem has three unknowns, namely the object memberships, the corresponding fundamental matrices, and the number of objects. To handle this otherwise recursive problem, hypotheses for fundamental matrices are generated through local sampling. Once the hypotheses are available, a combinatorial selection problem is formulated to optimize a model selection cost which takes into account the hypotheses likelihoods and the model complexity. An explicit model for outliers is also added for robust segmentation. The model selection cost is minimized through the branch-and-bound technique of combinatorial optimization. The proposed branch-and-bound approach efficiently searches the solution space and guaranties optimality over the current set of hypotheses. The efficiency and the guarantee of optimality of the method is due to its ability to reject solutions without explicitly evaluating them. The proposed approach was validated with synthetic data, and segmentation results are presented for real images.**

***Index Terms*—Branch-and-bound, combinatorial optimization, model selection, structure-and-motion segmentation.**

I. Introduction

SEGMENTATION of structure-and-motion is a vital step towards interpretation of a dynamic scene. The structure of a typical dynamic scene includes multiple independently moving objects, and these objects are captured by a moving camera. Conventional approaches based on the frame difference [1], [2] or the 2-D flow based methods [3], [4] are restricted in segmenting such a scene. The frame difference based approaches are limited due to the need for camera motion compensation. On the other hand, the 2-D flow based approaches are limited by the camera model used which is typically affine.

To address the segmentation problem in a better way, a comprehensive theory of structure-and-motion (SaM) estimation from perspective images has been developed by computer vision researchers over the years [5]. Analysis of dynamic scenes based on this theory, also known as multibody structure-and-motion (MSaM), is now being extensively explored. The two-view MSaM problem can be interpreted as a geometric problem [6]. However, direct application of the geometric interpretation to the real world problems is limited as it lacks an outlier model. Clustering is an interesting alternative to solve the MSaM problem. Two-view MSaM clustering turns out to be a chicken-and-egg problem. To segment a scene, motion models for all the objects in the scene are needed and to estimate the motion models of individual objects, the objects have to be segmented first. To solve this recursive problem, iterative technique such as expectation maximization (EM) can be used [7]. As the results of EM are only locally optimal, the quality of the final segmentation depends on the initial clustering. An alternative to the iterative method is a sequential extraction strategy where the dominant motions are segmented, and separated one by one, until the entire scene is explained [8]. A limitation of such methods is that the objects with similar motions are often incorrectly segmented. The object encountered earlier in the search is assigned some fraction of other objects which have similar motion.

To get out of the chicken-and-egg dilemma, some researchers have applied random sampling to generate multiple hypotheses for the motions in a scene [9], [10]. Prior knowledge that the segmentation is spatially coherent, helps in the selection of reliable hypotheses by local sampling. The local sampling can be carried out as random sample consensus (RANSAC) [5] applied to local spatial neighborhood. Once hypotheses are available through sampling, a suitable cost function can be optimized to achieve MSaM segmentation. Another important aspect of clustering is the *number* of clusters. While most of the clustering techniques assume that the number of clusters is known, such assumption is not valid for the segmentation of a dynamic scene. Typically, clustering is carried out by varying the number of clusters, and the best fitting clustering under a certain criterion is selected. In this work, the problem of selecting optimal number of clusters is formulated as a combinatorial optimization problem under a sampling based framework.

The paper gives a general combinatorial framework to optimize a model selection cost function. The cost function integrates maximum likelihood of hypotheses, a clustering cost and uniform distribution of outliers. Initially, hypotheses for motion are generated by local sampling of correspondences between two views. A null hypothesis is also introduced, which suggests that a correspondence is an outlier, with uniform likelihood. Next, a model selection criterion that penalizes the likelihood of the clustering with increasing number of clusters is added to the framework. The model selection criterion is optimized through a branch-and-bound process to obtain the final

Manuscript received July 08, 2009; revised January 13, 2010. First published March 10, 2010; current version published May 14, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Rick P. Millane.

N. Thakoor and V. Devarajan are with the Electrical Engineering Department, University of Texas at Arlington, Arlington, TX 76010 USA (e-mail: ninad.thakoor@mavs.uta.edu; venkat@uta.edu).

J. Gao is with the Computer Science and Engineering Department, University of Texas at Arlington, Arlington, TX 76019 USA (e-mail: gao@uta.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2042647

MSaM segmentation. Our preliminary work based on this idea appeared in [11].

The paper is organized as follows. Section II formulates the MSaM segmentation as a combinatorial optimization problem. A branch-and-bound solution to the problem is formulated in Section III. The experimental results are presented in Section IV, and Section V enlists the concluding remarks.

II. Formulation

Consider a set of M image correspondences $\mathbf{X} = \{(\mathbf{x}_1, \mathbf{x}_1'), (\mathbf{x}_2, \mathbf{x}_2'), \ldots, (\mathbf{x}_M, \mathbf{x}_M')\}$, where $\mathbf{x}_i$ and $\mathbf{x}_i'$ are image coordinates of the ith correspondence. The relationship among various object structures and motions in a scene can be expressed as

$$\mathbf{x}_i'^T \left(\sum_{j=1}^{K} \mathcal{L}_j(i) \mathtt{F}_j \right) \mathbf{x}_i = 0. \tag{1}$$

Here, $\mathtt{F}_j$ is the fundamental matrix [5] for the jth rigid body in the scene, and $\Theta = \{\mathtt{F}_1, \mathtt{F}_2, \ldots, \mathtt{F}_K\}$ is the set of fundamental matrices of K rigid bodies in the scene. The indicator function $\mathcal{L}_j(i)$ is one when the ith correspondence belongs to the jth rigid body and is zero otherwise. A label field $L = [l_1, l_2, \ldots, l_M]$ is associated with the indicator function $\mathcal{L}_j(i)$ such that, if $\mathcal{L}_j(i) = 1$, $l_i = j$. The goal of MSaM segmentation is to estimate the label field L. Once the label field is known, the least squares estimate of the fundamental matrix $\mathtt{F}_j$ can be computed as

$$\mathtt{F}_j = \arg\min_{\mathtt{F}} \sum_{\forall i, l_i = j} d(\mathbf{x}_i, \mathbf{x}_i', \mathtt{F})^2. \tag{2}$$

Here, d is a distance measure such as symmetric transfer error, reprojection error or Sampson approximation [5].

On the other hand, if $\mathtt{F}_j$s are known, the maximum likelihood estimate for the label of the ith match is given by

$$\hat{l}_i = \arg\min_{l} d(\mathbf{x}_i, \mathbf{x}_i', \mathtt{F}_l)^2. \tag{3}$$

Equations (2) and (3) represent parameter estimation and label estimation or segmentation steps respectively. Since these steps are interdependent, the MSaM problem can be solved iteratively to maximize the likelihood of the correspondences. Assuming that the uncertainties in the matches are normally distributed, with zero mean and standard deviation σ, the log likelihood of the fundamental matrices is given by

$$\log\{\mathrm{Lik}(\Theta)\} = -\frac{1}{2}\left(\frac{\mathrm{SSD}(\Theta)}{\sigma^2}\right) + \mathrm{Constant} \tag{4}$$

where

$$\mathrm{SSD}(\Theta) = \sum_{i=1}^{M} \min_{j=1}^{K} d(\mathbf{x}_i, \mathbf{x}_i', \mathtt{F}_j). \tag{5}$$

This optimization procedure also assumes that the number of objects K is known *a priori*. This assumption is unrealistic in most of the scenes. Since the likelihood of the correspondences increases as K is increased, the likelihood alone cannot be applied to select an optimal value of K. A model selection criterion such as the Bayesian information criterion (BIC) or the Akaike information criterion (AIC) can be utilized to select the optimal K [12]. These criteria penalize the likelihood in proportion of K. A generalized cost function to incorporate this idea can be defined as

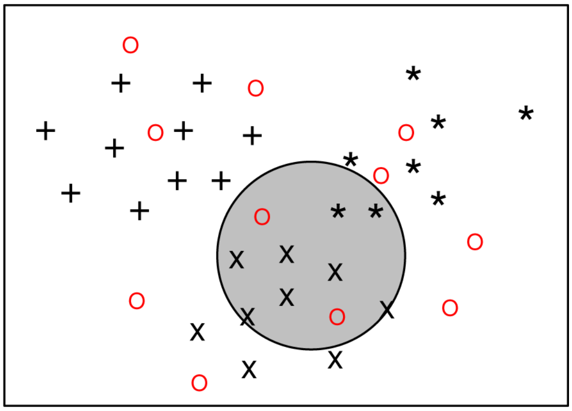


Fig. 1. Spatially coherent sampling.

$$\mathcal{C} = -2\log\{\mathrm{Lik}(\Theta)\} + \alpha \cdot K \tag{6}$$

where α is a positive constant. For BIC, $\alpha = N\log(M)$ and for AIC, $\alpha = 2$ NM. N is the number of free parameters per cluster. The first term of (6) gives the negative log likelihood of the model, which decreases with increase in K. The second term of (6) is the penalty term, which increases with increase in K. Thus, the minimum cost $\mathcal{C}$ compromises between the likelihood and the number of clusters to select an optimal value for K. The cost function in (6) can be minimized by varying K and iteratively optimizing the likelihood in (4) for that value of K.

Alternative to this approach is a simultaneous model selection and segmentation approach. In this approach, multiple hypotheses for fundamental matrices $\mathtt{F}_j$ where $j = 1, 2, \ldots, N_h$ are generated by local sampling of the correspondences. Fig. 1 shows three different motions marked with $+$, $*$ and X and the outliers marked by O. RANSAC is applied to correspondences in the circular spatial neighborhood of a correspondence to estimate $\mathtt{F}_j$. Use of the spatial neighborhood ensures that RANSAC can correctly and quickly estimate hypotheses for fundamental matrices. Once these hypotheses are known, the MSaM segmentation problem is reduced to a combinatorial optimization problem to select K hypotheses out of total N_h hypotheses. Note that there are 2^{N_h} possible solutions for this problem. Thus, even for a moderate value of N_h, an exhaustive search becomes intractable. However, the nature of the problem allows us to use a branch-and-bound approach to obtain an optimal solution in a reasonable time for practical problems.

III. Branch-and-Bound

The branch-and-bound approach [13] to global optimization splits an optimization problem into smaller subproblems. For these subproblems, upper and/or lower bounds on the cost function are estimated. These bounds are used to eliminate the subproblems that would not lead to an optimal solution. For the subproblems that survive, splitting and bound calculation is continued till all the subproblems are explored. The branch-and-bound procedure is applied in diverse areas such as optimal feature subset selection [13]–[15], image registration [16], rate-distortion based coding [17], job scheduling [18], [19], and clustering where number of clusters are known [13], [14], [20]. The rest of this section constructs a branch-and-bound algorithm for the optimization of the cost function in (6).

A branch-and-bound algorithm requires formulation of various components such as branching, bounding, pruning and retracting [13]. Apart from bounding, all the other components can be represented as a rooted tree. In the following subsection, the tree representation of the MSaM segmentation problem is formulated.

A. Solution Tree

An ordered set $H = \{\mathtt{F}_1, \mathtt{F}_2, \ldots, \mathtt{F}_{N_h}\}$ gives N_h hypotheses for the fundamental matrices $\mathtt{F}_j$ and K of them have to be chosen to minimize the criterion in (6). All possible solutions of this optimization problem can be represented as a rooted tree. Each node of the tree represents a solution. A node is also a partial solution for its descendent nodes. It is important that every solution is listed only once to avoid unnecessary computations. This can be ensured by creating child nodes that are different than:

- left siblings;
- ancestors;
- left siblings of ancestors.

A simple way of generating such a solution tree for $N_h = 5$ is shown in Fig. 2. The figure includes an additional null hypothesis $\mathtt{F}_0$ which is shown as $z_0 = 0$. According to the null hypothesis, none of the N_h hypotheses is valid for a given image correspondence. This means that the match is an outlier. The null hypothesis is introduced in detail later in the section. Note that, in a solution tree of height n

- $z_0 < z_1 < z_2 < z_3 \ldots < z_n$;
- left sibling < right sibling.

These two conditions ensure that the rule stated above to generate the child nodes is followed. Note that this gives rise to a binomial tree of degree N_h [21]. This tree has the following properties.

- The tree has 2^{N_h} nodes, and each node corresponds to a solution.
- The height of the tree is N_h which is equal to the largest possible value of K.
- At any given depth z_n, the tree has $(z_n!)/(N_h!(N_h - k)!)$ nodes.

The solution tree can be explored by search algorithms such as breadth first search and depth first search. The depth first search was chosen to take advantage of the recursive relationships of various computations which will be clear in the discussion later. In general, the depth first search avoids the exponential space complexity as well. A branch-and-bound search algorithm can be applied as a series of *branch forward*, *branch right*, and *retraction* operations.

To understand the various tree operations and their physical interpretation, it is assumed that the circled node in Fig. 2 indicates the current search location. The current location can be represented by the nodes traversed to reach it, i.e., $\langle 0, 1, 3\rangle$. This means that the null hypothesis, hypotheses $\mathtt{F}_1$ and $\mathtt{F}_3$ are included in the current solution. Additionally, a binary representation for the node can be defined. In an N_h bit wide binary representation, the nodes traversed to reach the current node are represented by "1" and the hypotheses that are not traversed are indicated by "0".

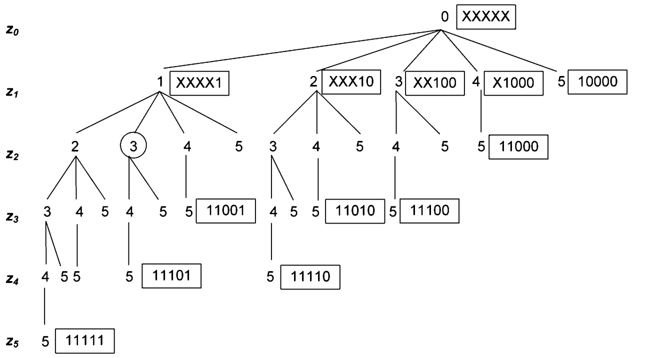


Fig. 2. Solution tree for $N_h = 5$ and a null hypothesis, number in the rectangle indicates extended representation for the node.

With a slight abuse of notation, this binary notation is extended to include a representation for the partial solutions. The hypotheses that can be traversed in the future are denoted by X (don't care) in this extended representation. By replacing Xs by 0s for a node, its *extended* representation as a partial solution can be turned into a *binary* representation as a solution. Thus, the circled node in Fig. 2 is represented by XX101. The extended representation indicates that the current solution is 00101, and through Xs it indicates that it is a partial solution for the solutions 01101, i.e., $\langle 0, 1, 3, 4\rangle$, 10101, i.e., $\langle 0, 1, 3, 5\rangle$, and 11101, i.e., $\langle 0, 1, 3, 4, 5\rangle$.

A branch forward operation moves deeper in the tree by one level. After a branch forward operation, the partial solution $\langle 0, 1, 3\rangle$ would lead to $\langle 0, 1, 3, 4\rangle$. In terms of the extended representation, the trailing X is replaced by "1." A branch forward operation adds one more hypothesis to the solution.

A branch right operation moves to the sibling branch towards right. In the extended representation, the leading "1" is replaced by "0" and the trailing X is replaced by "1." The solution $\langle 0, 1, 3\rangle$ would branch right to give the solution $\langle 0, 1, 4\rangle$. A branch right operation replaces the last included hypothesis with the next hypothesis. Thus, the number of hypotheses after the branch right operation remains the same.

A retraction moves the solution one level up the tree. A retraction is carried out when no forward or right branching is possible. Solution $\langle 0, 1\rangle$ is the result of the retraction at the circled node. For the extended representation, the operation first replaces leading 1 with X and then all the leading "0s" with Xs. Note that a retraction is generally followed by a branch right step. The branch-and-bound algorithm is terminated when a retraction leads to the root node.

B. Monotonicity of Partial Costs

The solution representing a node at depth n be given by $Z(n) = \{\langle z_0, z_1, z_2, \ldots, z_{n-1}, z_n\rangle, \mathcal{D}(n)\}$. The hypotheses $z_0, z_1, z_2, \ldots, z_{n-1}, z_n$ correspond to fundamental matrices $\mathtt{F}_{z_0}, \mathtt{F}_{z_1}, \mathtt{F}_{z_2}, \ldots, \mathtt{F}_{z_{n-1}}, \mathtt{F}_{z_n}$ respectively. The minimum distances at depth n are given by

$\mathcal{D}(n) = [D(1,n), D(2,n), \ldots, D(N,n)]$ and $D(i,n)$ corresponds to the minimum distance for the ith match among the current set of hypotheses

$$D(i,n) = \min_{k=0}^{n} d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_k}\right). \tag{7}$$

The cost function for the solution $Z(n)$ can be written as

$$\mathcal{C}(Z(n)) = \underbrace{\frac{1}{\sigma^2}\sum_{i=1}^{N} D(i,n)}_{\text{Negative log likelihood}, \tilde{L}} + \underbrace{\alpha \cdot n}_{\text{Penalty}}. \tag{8}$$

The cost function is made up of two terms, one corresponding to the negative log likelihood $\tilde{L}$ and the other corresponding to the penalty.

Equation (7) can be rewritten as

$$\begin{aligned} D(i,n) &= \min\left\{\min_{k=0}^{n-1} d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_k}\right), d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_n}\right)\right\} \\ &= \min\{D(i,n-1), d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_n}\right)\}. \end{aligned} \tag{9}$$

Thus, from $D(i,n-1)$, the newly formed $D(i,n)$ can be calculated incrementally with (9). When a new hypothesis z_n is added to the existing partial solution, the matches which are better explained by the new hypothesis are reassigned to the new hypothesis while others remain unchanged. Additionally, it is clear from (9) that

$$\forall i, D(i,n) \leq D(i,n-1).$$

This suggests that

$$\sum_{i=1}^{N} D(i,n) \leq \sum_{i=1}^{N} D(i,n-1) \tag{10}$$

$$\tilde{L}\{Z(n)\} \leq \tilde{L}\{Z(n-1)\} \tag{11}$$

when a new hypothesis is added, the penalty term of the cost function increases by α while the negative log likelihood term decreases or remains the same. These monotonicity properties are used in the following subsection to establish a lower bound on the cost function.

Leading from the monotonic decrease of the negative log likelihood and the linear increase of the penalty term, a monotonicity requirement can be imposed on the optimal solution. A *likelihood value* for a hypothesis $z_m, G_n(z_m)$ can be defined as increase in the negative log likelihood of the solution $Z(n)$ if z_m is removed from the solution to form a new solution $Z'(n-1)$ (Note that $Z'(n-1)$ and $Z(n-1)$ are different if $m \neq n$). Similar to the likelihood value, the *per pixel value* of hypothesis z_m at depth n for pixel $i, v_n(i, z_m)$, can be defined as

$$v_n(i, z_m) = D'(i, n-1) - D(i,n) \tag{12}$$

where $D'(i,n-1)$ is the minimum distance for the ith match with an updated set of hypotheses $(z_0, z_1, z_2, \ldots, z_{n-1}, z_n) \setminus z_m$. From the definition, the likelihood value can be written in terms of the per pixel value of a hypothesis as

$$G_n(z_m) = \frac{1}{\sigma^2}\sum_{i=1}^{N} v_n(i, z_m). \tag{13}$$

Using these definitions, the proof of the monotonicity of the cost function can be constructed.

Theorem 3.1: The per pixel value of a hypothesis z_m for the pixel i is maximum when it is first added, i.e., for any $n > m, v_n(i, z_m) \leq v_m(i, z_m)$.

Proof: A hypothesis z_m is first added at depth m. For depth, $n < m, z_m$ is not part of the solution and has zero per pixel value. From (7) and (12), the per pixel value of the hypothesis z_m for the pixel i when $n \geq m$ is

$$\begin{aligned} & v_n(i, z_m) \\ &= \min\left\{\min_{k'=0}^{m-1} d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_{k'}}\right)\right. \\ &\left.\min_{k^*=m+1}^{n} d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_{k^*}}\right)\right\} \\ &\quad - \min_{k=0}^{n} d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_k}\right) \end{aligned} \tag{14}$$

$$\begin{aligned} &= \left(\min\left\{\min_{k'=0}^{m-1} d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_{k'}}\right)\right.\right. \\ &\left.\min_{k^*=m+1}^{n} d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_{k^*}}\right)\right\} \\ &\quad \left. - d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_m}\right)\right)_+ \end{aligned} \tag{15}$$

where

$$(f(\cdot))_+ = \begin{cases} f(\cdot), & \text{if } f(\cdot) > 0 \\ 0, & \text{otherwise.} \end{cases}$$

is a function which maps negative values to zero while keeping positive values unchanged. When $n = m$

$$v_m(i, z_m) = \left(\min_{k'=0}^{m-1} d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_{k'}}\right) - d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_m}\right)\right)_+. \tag{16}$$

Comparing (15) and (16), for any $n > m$

$$v_n(i, z_m) \leq v_m(i, z_m). \tag{17}$$

■

Theorem 3.2: For optimality of a solution $Z(n)$, it is necessary that $\alpha \leq G_n(z_m)$ for all $m \leq n$.

Proof: If the solution $Z(n)$ is optimal, then for any $m \leq n$

$$\begin{aligned} \mathcal{C}\{Z(n)\} &\leq \mathcal{C}\{Z'(n-1)\} \\ \tilde{L}\{Z(n)\} + \alpha \cdot n &\leq \tilde{L}\{Z'(n-1)\} + \alpha \cdot (n-1) \\ \alpha &\leq \tilde{L}\{Z'(n-1)\} - \tilde{L}\{Z(n)\} \\ \alpha &\leq G_n(z_m). \end{aligned} \tag{18}$$

■

Theorem 3.3: If the initial likelihood value of a hypothesis $z_m, G_m(z_m) < \alpha$, any solution leading from the current partial solution cannot be optimal.

Proof: From (13) and (17)

$$G_n(z_m) \leq G_m(z_m). \tag{19}$$

If $G_m(z_m) < \alpha$ then

$$G_n(z_m) < \alpha. \tag{20}$$

Then according to Theorem 3.2, any solution which includes z_m cannot be optimal. ■

If $\mathcal{C}\{Z(n)\} > \mathcal{C}\{Z(n-1)\}$ then $G_m(z_m) < \alpha$. Thus, for Theorem 3.3 to hold, the cost function must be monotonically decreasing.

C. Lower Bound on Cost

To establish a lower bound on the cost, a complementary variable $D^*(i, z_n)$ can be defined as

$$D^*(i, z_n) = \min_{k=z_n+1}^{N_h} d(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_k).$$

The variable $D^*(i, z_n)$ gives the minimum of distance measures from all the hypotheses which can be included in the solution in the future. In case of the variable $D^*(i, z_n)$, its value solely depends on the last node z_n. As there are only N_h possibilities for the value of $z_n, D^*(i, z_n)$ can be precomputed to speed up the branch-and-bound process. Similar to $D(i, n), D^*(i, n)$ can also be calculated incrementally as

$$D^*(i, z_n) = \min\left\{D^*(i, z_n + 1), d\left(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{z_n+1}\right)\right\}.$$

Consider a possible partial solution $Z(4) = \langle 0, 1, 3, 4, 7\rangle$ for $N_h = 10$. The variable D at level $n = 4$ can be computed as

$$\begin{aligned} D(i, 4) = \min\{&d(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_0), d(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_1) \\ &d(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_3), d(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_4), d(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_7)\}.\end{aligned}$$

Now for the same example, the complementary variable D^* is given by

$$D^*(i, \mathtt{F}_7) = \min\{d(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_8), d(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_9), d(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_{10})\}.$$

With help of the complementary variable, the lower bound on the solutions leading from $Z(n)$ is

$$\mathcal{C}_{\text{Lower}}(Z(n)) = \frac{1}{\sigma^2}\sum_{i=1}^{N} \min\{D(i, n), D^*(i, z_n)\} + \alpha \cdot (n + 1).$$

If $\mathcal{C}_{\text{Lower}}(Z(n)) > \mathcal{C}^*$, then the current partial solution can be safely abandoned as it would not lead to a better solution than the current optimal solution $\mathcal{C}^*$.

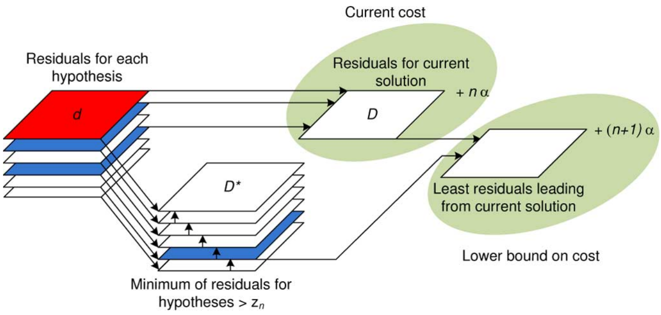

Current cost
Residuals for each hypothesis
Residuals for current solution
d
D
+ n α
+ (n+1) α
D*
Least residuals leading from current solution
Lower bound on cost
Minimum of residuals for hypotheses > z_n

Fig. 3. Computation of the lower bound on the cost for $N_h = 5$ and the node $\langle 0, 1, 3\rangle$.

Fig. 3 depicts the computation of the bound. Each stack of parallelograms indicates various quantities involved in bound computation and arrow-heads leading to a parallelogram indicate a minimum taken over the parallelograms attached to the arrow-tails.

D. Null Hypothesis Likelihood

Matching errors are common in the MSaM segmentation problems. These outliers can severely deteriorate the quality of the solutions achieved for the MSaM segmentation. The outliers can be assumed to be uniformly distributed throughout the image with likelihood d_0. For ease of notation, it is assumed that

$$d(\mathbf{x}_i, \mathbf{x}'_i, \mathtt{F}_0) = d_0.$$

With introduction of this outlier likelihood as the null hypothesis, the proposed MSaM segmentation scheme acts as a simple redescending M-estimator [22].

E. Branch-and-Bound Algorithm

Based on the monotonicity requirement and the lower bound, the branch-and-bound segmentation algorithm is listed below.

1) Initialization: Set the tree level $n = 1$, the current node $z_0 = 0$ and the current optimal cost $\mathcal{C}^* = \mathcal{C}(Z(0))$.
2) Generate child nodes: Initialize $\text{LIST}(n)$

$$\text{List}(n) = \{z_{n-1} + 1, z_{n-1} + 2, \ldots, N_h\}.$$

3) Select a new node: If $\text{List}(i)$ is empty, go to step (5). Otherwise, set $z_n = k$ where $k \in List(i)$. Set the current solution $Z(n) = \{z_0, z_1, \ldots, z_n\}$. Delete k from $\text{List}(i)$.
4) Check bounds:
   - Compute $\mathcal{C}(Z(n))$ and $\mathcal{C}_{\text{Lower}}(Z(n))$.
   - If $\mathcal{C}(Z(n)) < \mathcal{C}^*$, set $\mathcal{C}^* = \mathcal{C}(Z(n))$ and $Z^* = Z(n)$.
   - If $\mathcal{C}(Z(n-1)) < \mathcal{C}(Z(n))$ or $\mathcal{C}_{\text{Lower}}(Z(n)) > \mathcal{C}^*$, go to step (3).
   - If $\mathcal{C}(Z(n-1)) > \mathcal{C}(Z(n))$ and $\mathcal{C}_{\text{Lower}}(Z(n)) < \mathcal{C}^*$, set $i = i + 1$ and go to step (2).
5) Backtrack to the lower level: Set $n = n - 1$. if $n > 0$ go to step (3), otherwise terminate the algorithm.

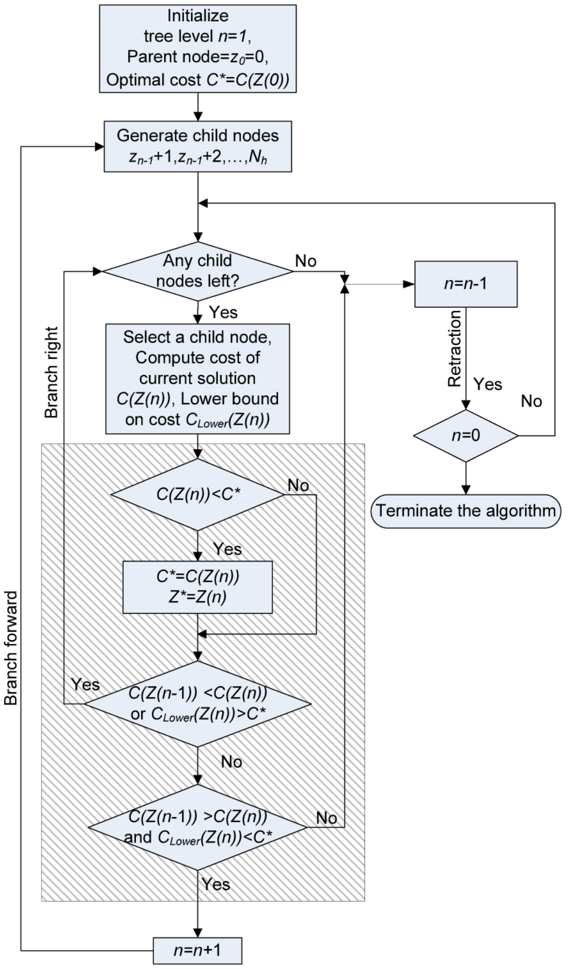

Initialize
tree level $n=1$,
Parent node=z_0=0,
Optimal cost $C^*=C(Z(0))$
Generate child nodes
$z_{n-1}+1, z_{n-1}+2, \ldots, N_h$
Any child nodes left?
No
Yes
$n=n-1$
Retraction
Yes
No
$n=0$
Terminate the algorithm
Select a child node, Compute cost of current solution $C(Z(n))$, Lower bound on cost $C_{Lower}(Z(n))$
Branch right
$C(Z(n))<C^*$
No
Yes
$C^*=C(Z(n))$
$Z^*=Z(n)$
Branch forward
Yes
$C(Z(n\text{-}1)) <C(Z(n))$ or $C_{Lower}(Z(n))>C^*$
No
$C(Z(n\text{-}1)) >C(Z(n))$ and $C_{Lower}(Z(n))<C^*$
No
Yes
$n=n+1$

Fig. 4. Flowchart of the proposed algorithm, highlighted portion of the chart checks for various bounds.

The flowchart of the algorithm is shown in Fig. 4. In the following section, the branch-and-bound hypothesis selection was implemented and the achieved results are presented.

IV. Experimental Results

The proposed MSaM segmentation approach was implemented and tested with synthetic data and publicly available data sets. The segmentation algorithm was implemented in MATLAB and executed on a Core 2 Duo processor operating at 2.33 GHz as a single thread.

To generate the motion hypotheses, for each image correspondence, the fundamental matrix was computed from circular neighborhood of the correspondence. The fundamental matrices were computed using "Structure-and-Motion Toolkit" from [23]. Similar to RANSAC, outliers and inliers were selected for each fundamental matrix with d_0 as the threshold. To avoid similar repeated hypotheses, a hypothesis with smaller support was suppressed when it had substantial $(>80\%)$ inliers overlapping with a larger hypothesis. Finally, the surviving hypotheses were arranged in a decreasing order of the number of inliers. The Bayesian information criterion (BIC) was optimized for these hypotheses to select the optimal hypotheses combination.

A. Synthetic Data

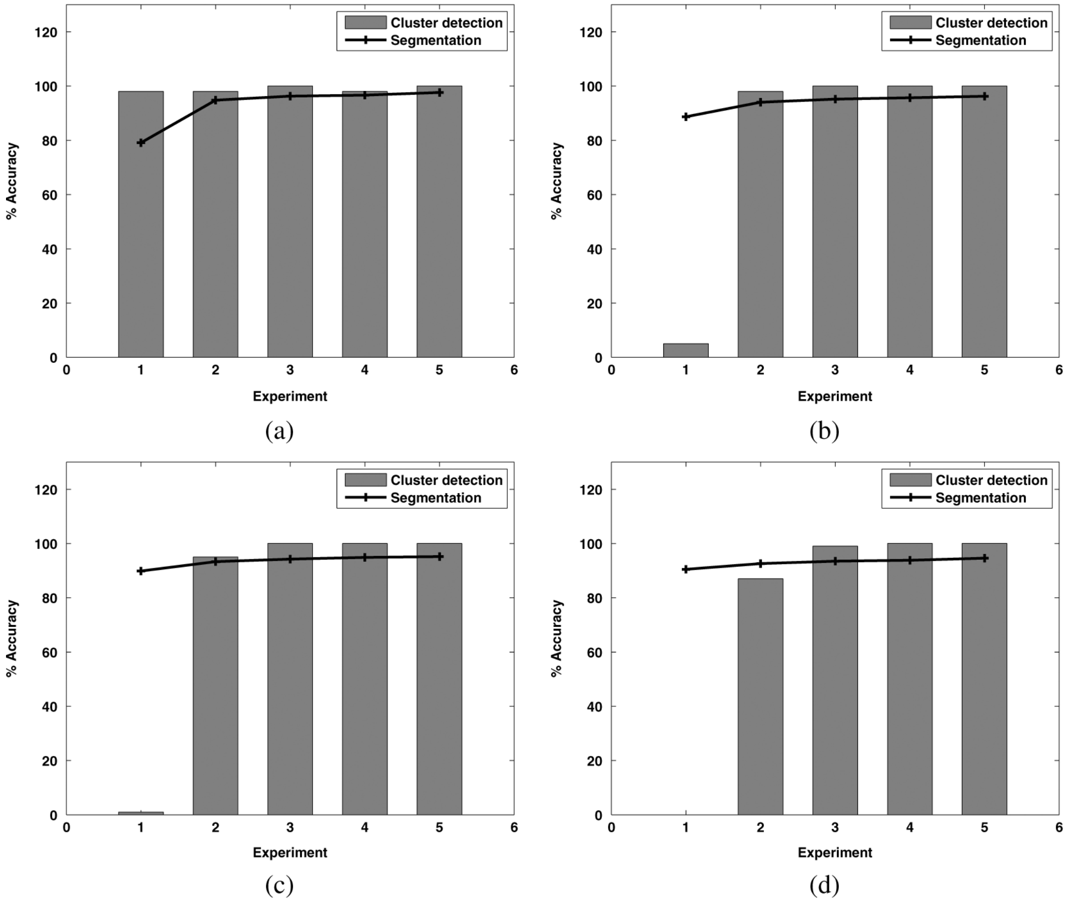
The proposed MSaM segmentation approach was first tested with synthetic data. For the experiments, 100 random 3-D motions were generated with [23].[1] These motions were combined together to form various experimental data sets. The goal of these experiments was to test effectiveness of the approach for detecting clusters of varying size and for identifying varying number of clusters. The results for the experiments are shown in Fig. 5 and are discusses in the rest of this subsection. Four different sets of experiments were carried out. The experimental results present cluster detection accuracies and segmentation accuracies for each of the set.

1) Set 1—50 Outliers $+1$ *Cluster of Varying Size 10 to 50:* For the first experiment in this set, 10 correspondences from one of the 100 motions were randomly selected. One sample each from 50 other motions was selected to form a set of outliers. The MSaM segmentation was carried out to estimate the number of clusters and their memberships. The process was repeated 100 times. The experiment was repeated by changing the size of the cluster to 20 (experiment 2), 30 (experiment 3), 40 (experiment 4) and 50 (experiment 5). For this set of experiments, the expected number of clusters was 2, one for the outliers and one for the motion with varying number of samples. Since the framework detects at least one motion cluster and one outlier set, as seen in Fig. 5(a), 2 clusters are always detected irrespective of the varying inlier cluster size. This leads to a 100% cluster detection accuracy for all the experiments. Thus, for this experiment, the cluster detection accuracy is invalid. However, it should be noted that the number of clusters is rarely overestimated.

The outliers were also included in estimating the segmentation accuracy, i.e., to reach 100% accuracy all the inliers must be labeled as one cluster while all the outliers should be labeled as the other cluster. It can be seen that for the inlier cluster size of 10, the segmentation accuracy is 79.1% which indicates that majority of 83.33% outliers are correctly identified. As the varying cluster size rises to 20 and beyond, the segmentation accuracy is more than 94%.

2) Set 2—50 Outliers $+1$ *Cluster of Size* $50+1$ *Cluster of Varying Size 10 to 50:* The second set of experiments was carried out by adding a randomly selected motion with 50 correspondences to the data in the first experiment. Thus, the expected number of clusters was 3 in this experiment; one for the outliers, one for the motion of size 50 and one for the inlier motion with varying cluster size. When varying cluster size is 10 (experiment 1), the proposed method fails to detect that cluster 95% of times [Fig. 5(b)]. This happens as it is difficult to obtain a clean sample to detect the correct motion hypothesis due to the large number of outliers compared to the inliers. Additionally, for less number of samples it might be "cheaper" to explain them as outliers rather than assigning them to a new cluster. In this scenario, the expected segmentation accuracy is $90.9\%((50+50)/(50+50+10))$ if all the outliers and the inlier cluster of size 50 is correctly identified. The experimental segmentation accuracy is 88.67% which is close to the expected accuracy. When the varying cluster size becomes 20 (experiment 2), the cluster detection accuracy is 98%. As the size of the cluster goes beyond 30 (experiments 3, 4, and 5) the cluster

[1] Function 'torr_gen_2view_matches' with the default parameters was used.



(a) (b) (c) (d)

Fig. 5. Synthetic data cluster detection and segmentation accuracy. (a) Set 1—50 Outliers + 1 cluster of varying size 10 to 50. (b) Set 2—50 Outliers + 1 cluster of size 50 + 1 cluster of varying size 10 to 50. (c) Set 3—50 Outliers + 2 clusters of size 50 each + 1 cluster of varying size 10 to 50. (d) Set 4—50 Outliers + 3 clusters of size 50 each + 1 cluster of varying size 10 to 50.

detection accuracy reaches almost 100% and the segmentation accuracy percentage reaches about 95.

3) Set 3—50 Outliers + 2 Clusters of Size 50 Each + 1 Cluster of Varying Size 10 to 50: The third set of experiments was carried out by adding a randomly selected motion with sample size 50 to the data in the second experiment. The expected number of clusters is 4 in these experiments. Failure to detect the motions which have 10 samples continues in this set of experiments. However, the cluster detection accuracy drops to 95% for the cluster size of 20. This happens because as the number of correspondences becomes larger, adding a cluster is more "expensive".

4) Set 4—50 Outliers +3 Clusters of Size 50 Each +1 Cluster of Varying Size 10 to 50: The fourth set of experiments was carried out by adding a randomly selected motion with sample size 50 to the data in the third experiment. The cluster detection accuracy as well as segmentation accuracy in this case is slightly lower compared to previous experiments. However, this is expected due to increase in clustering penalty.

In the other synthetic data experiment, "Spinning wheels" test data from [24] was used. This sequence contains four rotating objects, with 50 tracked points each, with 50 outliers. Frames 1 and 3 of the sequence were used in the experiment. After sampling and non maximal suppression, 22 hypotheses were selected. As seen in Fig. 6, the proposed approach detects 4 clusters along with outliers. The total number of solutions explored by the branch-and-bound process was 2043.

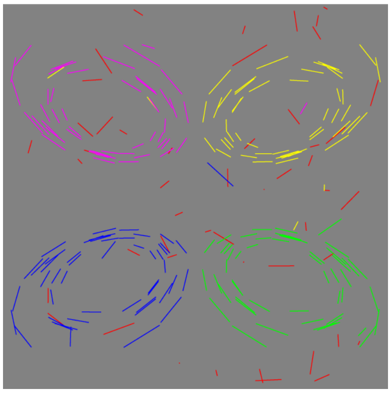


Fig. 6. Spinning wheels: Correspondences between two frames, each cluster is denoted by different color, matches marked by red are outliers.

B. Real Data

For all the real data used in the experiments, sparsely matched features were provided in the data set. For the first experiment with real data, "Box-book-mag" and "Desk" image pairs from [10] were used. "Box-book-mag" pair has three independently

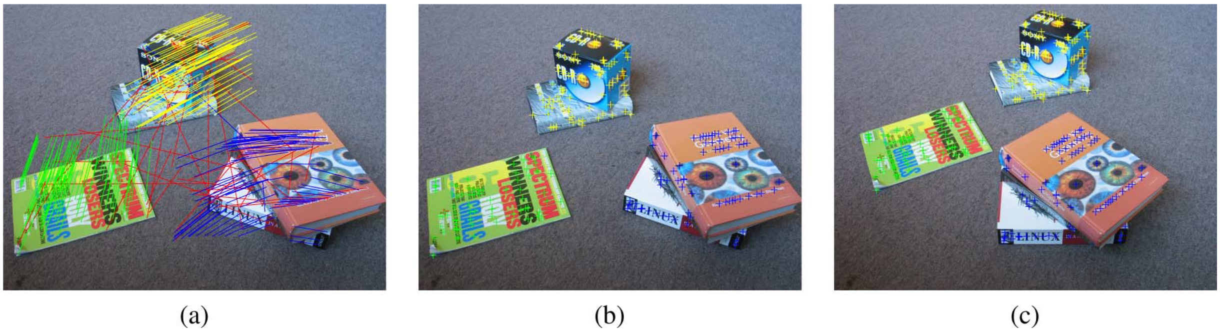


Fig. 7. Box-book-mag: (a) Correspondences between two views, each cluster is denoted by different color, matches marked by red are outliers; (b) segmentation result for the first view; (c) segmentation result for the second view.

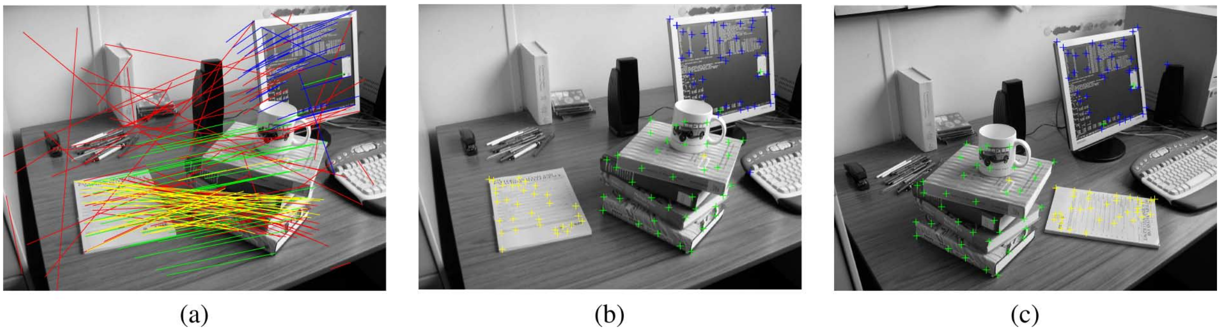


Fig. 8. Desk: (a) Correspondences between two views, each cluster is denoted by different color, matches marked by red are outliers; (b) segmentation result for the first view; (c) segmentation result for the second view.

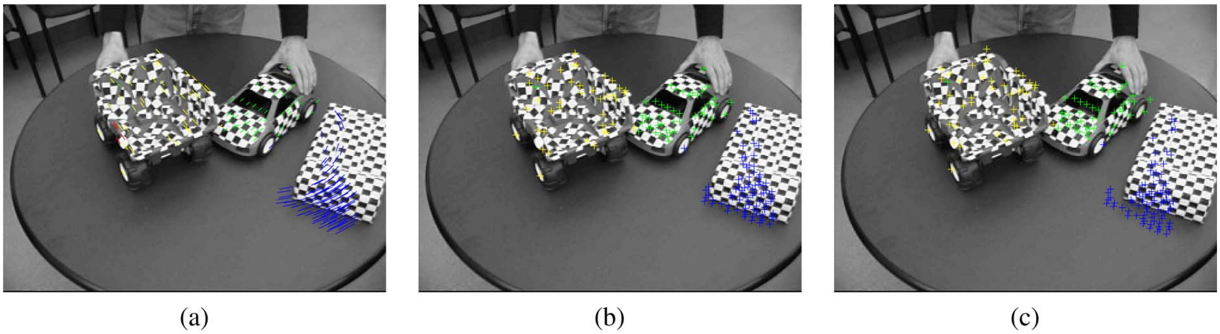


Fig. 9. Car-truck-box: (a) Correspondences between two views, each cluster is denoted by different color, matches marked by red are outliers; (b) segmentation result for frame 1; (c) segmentation result for frame 8.

moving objects while the camera is stationary. Fig. 7(a) shows correspondences between the image pair with each object indicated by a different color. The red colored matches are the detected outliers. For the "Desk" image pair shown in Fig. 8, there are three moving objects namely the pile of books, the computer screen and the journal. Although the camera has also moved, there are no matches available for the background. Thus, the background motion is not detected. The result of segmentation can be seen in Figs. 8(b) and (c).

In the next experiment, the proposed method was applied to the "car-truck-box" sequence used by Vidal *et al.* [25], [26]. The motion between frame 1 and frame 8 of the sequence was analyzed. In this sequence, there are three different motions. The box lies on a rotating desk, while the car and the truck are moved away from each other with hands. As seen in Fig. 9, three moving objects are correctly identified; however, some of the correspondences are incorrectly assigned. This is due to the sampling scheme used, rather than the cost function being optimized. If the optimal motions are subset of the hypotheses being constructed, then the segmentation results are guaranteed to be optimal with respect to the cost function.

In the next sequence, taken from Sugaya and Kanatani [27], has a single moving object, i.e., the car. However, the camera is also moving for this sequence. Frames 10 and 15 are used for segmentation in the experiment. The egomotion of the camera and the motion of the car are correctly segmented, and are shown in Fig. 10(c).

Finally, the proposed approach was tested with JHU155 database sequences [28], which include various checkerboard and traffic sequences, with two or three motion groups. The "cars3" sequence shown in Fig. 11(a) has two moving cars captured by a moving camera. Fig. 11(b) gives segmentation results for

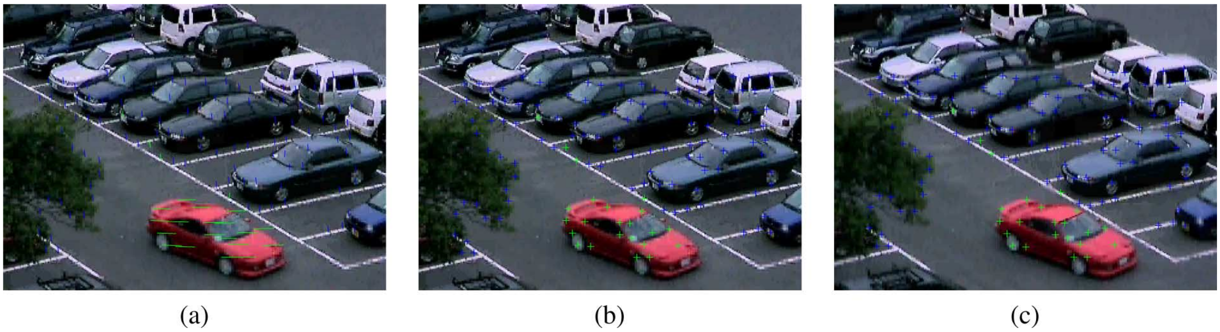


Fig. 10. Kanatani: (a) Correspondences between two views, each cluster is denoted by different color; (b) segmentation result for frame 10; (c) segmentation result for frame 15.

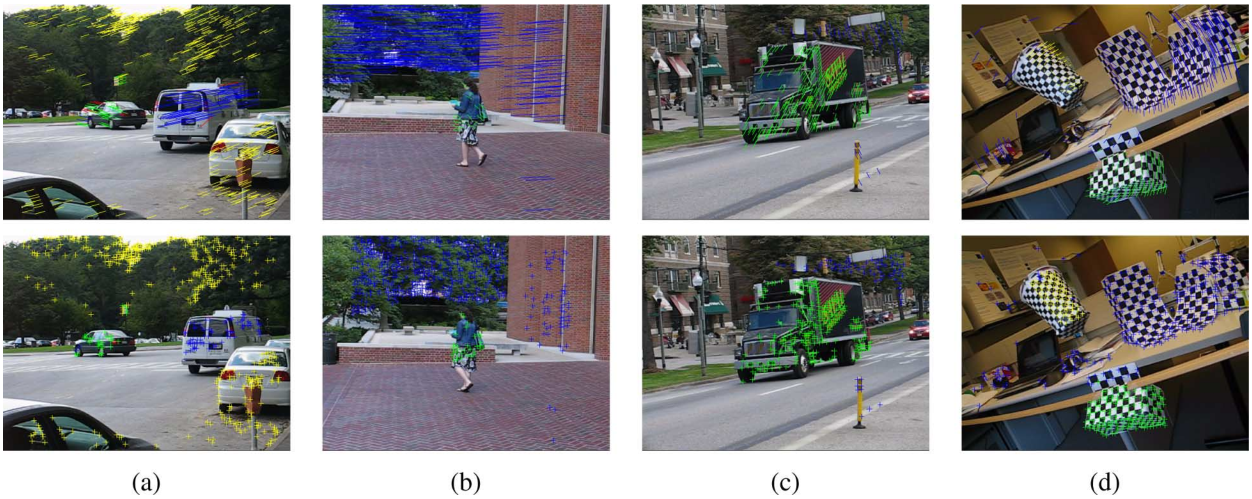


Fig. 11. Sequences from JHU155 database: **Top**: Segmentation result with correspondences for the first view **Bottom**: Segmentation result for the second view (a) “cars3” sequence; (b) “people1” sequence; (c) “truck2” sequence; (d) “1R2TCR” sequence.

TABLE I
EXECUTION SUMMARY FOR THE EXPERIMENTS

Sequence	N_h	Solutions explored	Fraction explored	Time (Seconds)
Spinning wheels	22	2043	4.87e-4	0.17
Three car-Vidal	35	30542	8.89e-7	2.54
Book-box-mag	16	511	7.80e-3	0.06
Desk	25	1898	5.66e-5	0.16
Kanatani	14	354	2.16e-2	0.06
cars3-JHU	53	20166	2.24e-12	2.44
people1-JHU	95	7149	1.80e-25	0.94
truck2-JHU	34	622	3.62e-8	0.07
1R2TCR-JHU	58	21550	7.48e-14	2.40

the “people1” sequence which depicts a pedestrian captured by a moving camera. The “truck2” sequence is segmented in the moving vehicle and the background in Fig. 11(c). The checkerboard sequence in Fig. 11(d), with one rotating object and one translating object captured by a rotating camera, was also successfully segmented by the proposed approach.

Table I shows a summary of the execution of the proposed method for all the experiments. Fraction of solutions explored shown in the table is calculated as

$$\text{Fraction explored} = \frac{\text{Solutions explored}}{2^{N_h}}.$$

As seen from the table, in all the cases, the fraction of the solutions explored is very small. This is also reflected in the execution speed. Note that the execution times for search alone are listed, and they do not include sampling and precomputing involved. The speedups achieved increase with increase in N_h, since more solutions are generally rejected implicitly by rejecting a partial solution. In our more recent work [29], the computational complexity of the algorithm is treated in detail.

V. CONCLUSION AND FUTURE WORK

A versatile multiple structure-and-motion segmentation scheme was proposed and its effectiveness was demonstrated through experiments. The branch-and-bound scheme can easily be scaled for parallel processing by solving one branch of the problem on a processor. Scheduling of these branches can be also an interesting direction of research. Although the method is proposed for a multibody SaM segmentation, it can be also applied to various other computer vision problems involving clustering such as segment based stereo [30], [31] and dense motion segmentation [3]. Since the outcome of the method heavily depends on the initial hypotheses chosen, various available guided sampling approaches have to be evaluated as to how well they explore and represent the solution space. The current approach can also be extended to an iterative approach. After each iteration of segmentation, fundamental matrices can be recalculated based on membership of the matches and these can added as additional hypothesis to repeat the segmentation.

REFERENCES

[1] F. Moscheni, S. Bhattacharjee, and M. Kunt, "Spatio-temporal segmentation based on region merging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 9, pp. 897–915, Sep. 1998.

[2] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[3] J. Wang and E. Adelson, "Representing moving images with layers," *IEEE Trans. Imag. Process.*, vol. 3, no. 5, pp. 625–638, Sep. 1994.

[4] H. Nguyen, M. Worring, and A. Dev, "Detection of moving objects in video using a robust motion similarity measure," *IEEE Trans. Imag. Process.*, vol. 9, no. 1, pp. 137–141, Jan. 2000.

[5] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004, ISBN: 0521540518.

[6] R. Vidal and Y. Ma, "A unified algebraic approach to 2-D and 3-D motion segmentation," in *Proc. Eur. Conf. Computer Vision*, 2004, pp. 1–15.

[7] A. Gruber and Y. Weiss, "Incorporating non-motion cues into 3-D motion segmentation," in *Proc. Eur. Conf. Computer Vision*, 2006, pp. 84–97.

[8] M. Irani and P. Anandan, "A unified approach to moving object detection in 2-D and 3-D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 6, pp. 577–589, Jun. 1998.

[9] H. Li, "Two-view motion segmentation from linear programming relaxation," in *Proc. Computer Vision and Pattern Recognition*, Jun. 2007, pp. 1–8.

[10] K. Schindler and D. Suter, "Two-view multibody structure-and-motion with outliers through model selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 983–995, Jun. 2006.

[11] N. Thakoor and J. Gao, "Branch-and-bound hypothesis selection for two-view multiple structure and motion segmentation," in *Proc. Computer Vision and Pattern Recognition*, Jun. 2008, pp. 1–6.

[12] A. D. R. McQuarrie and C.-L. Tsai, *Regression and Time Series Model Selection*. Singapore: World Scientific, 1998.

[13] M. Brusco and S. Stahl, *Branch-and-Bound Applications in Combinatorial Data Analysis*. New York: Springer, 2005.

[14] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. San Diego, CA: Academic, 1990.

[15] P. Somol, P. Pudil, and J. Kittler, "Fast branch & bound algorithms for optimal feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 7, pp. 900–912, Jul. 2004.

[16] L.-K. Shark, A. A. Kurekin, and B. J. Matuszewski, "Development and evaluation of fast branch-and-bound algorithm for feature matching based on line segments," *Pattern Recognit.*, vol. 40, no. 5, pp. 1432–1450, 2007.

[17] M. Roder, J. Cardinal, and R. Hamzaoui, "Branch and bound algorithms for rate-distortion optimized media streaming," *IEEE Trans. Multimedia*, vol. 8, no. 1, pp. 170–178, Feb. 2006.

[18] J. Jonsson and K. Shin, "A parametrized branch-and-bound strategy for scheduling precedence-constrained tasks on a multiprocessor system," in *Proc. Int. Conf. Parallel Processing*, 1997, pp. 158–165.

[19] S. Fujita, M. Masukawa, and S. Tagashira, "A fast branch-and-bound algorithm with an improved lower bound for solving the multiprocessor scheduling problem," in *Proc. Int. Conf. Parallel and Distributed Systems*, 2002, pp. 611–616.

[20] W. L. G. Koontz, P. M. Narendra, and K. Fukunaga, "A branch and bound clustering algorithm," *IEEE Trans. Comput.*, vol. 24, no. 9, pp. 908–915, Sep. 1975.

[21] T. T. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.

[22] P. J. Huber, *Robust Statistics*. Hoboken, NJ: Wiley, 1981.

[23] P. Torr, A Structure and Motion Toolkit in Matlab [Online]. Available: http://cms.brookes.ac.uk/staff/PhilipTorr/Beta/torrsam.zip

[24] K. Schindler, J. U. , and H. Wang, "Perspective n-view multibody structure-and-motion through model selection," in *Proc. Eur. Conf. Computer Vision*, 2006, pp. I: 606–619.

[25] R. Vidal and S. Sastry, "Optimal segmentation of dynamic scenes from two perspective views," in *Proc. Computer Vision and Pattern Recognition*, Jun. 2003, vol. 2, pp. 281–286.

[26] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, "Segmentation of dynamic scenes from the multibody fundamental matrix," presented at the Eur. Conf. Computer Vision Workshop on Visual Modeling of Dynamic Scenes, 2002.

[27] Y. Sugaya and K. Kanatani, "Multi-stage optimization for multi-body motion segmentation," *IEICE Trans. Inf. Syst.*, vol. E87-D, no. 7, pp. 1935–1942, Jul. 2004.

[28] The Hopkins 155 Dataset [Online]. Available: http://www.vision.jhu.edu/data/hopkins155/

[29] N. Thakoor, V. Devarajan, and J. Gao, "Computation complexity of branch-and-bound model selection," presented at the Int. Conf. Computer Vision, 2009.

[30] M. Lin and C. Tomasi, "Surfaces with occlusions from layered stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1073–1078, Aug. 2004.

[31] L. Hong and G. Chen, "Segment-based stereo matching using graph cuts," in *Proc. Computer Vision and Pattern Recognition*, 2004, vol. 1, pp. 74–81.

Ninad Thakoor (S'04) received the B.E. degree in electronics and telecommunication engineering from University of Mumbai, Mumbai, India, in 2001, and the M.S. and Ph.D. in electrical engineering from the University of Texas at Arlington in 2004 and 2009, respectively.

His research interests include visual object recognition, stereo disparity segmentation, and structure-and-motion segmentation.

Jean Gao (S'96–M'93) received the B.S. degree in biomedical engineering from the Shanghai Medical University, Shanghai, China, in 1990, the M.S. degree in biomedical engineering from the the Rose-Hulman Institute of Technology, Terre Haute, IN, in 1996, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 2002.

She is currently an Associate Professor with the Computer Science and Engineering Department and the Director of Biocomputing and Vision Lab, University of Texas at Arlington. Her research interests include object tracking, motion estimation, stereo object segmentation, pattern recognition, medical imaging, and applications in computational biology, and clinical medical informatics.

Dr. Gao is the recipient of the prestigious CAREER Award from the National Science Foundation and the Outstanding Young Faculty Award from University of Texas at Arlington.

Venkat Devarjan (SM'02) studied electrical engineering at Indian Institute of Technology Madras, Chennai, India, where he received the B.S.E.E. and M.S.E.E. degrees in 1973 and 1975, respectively. He performed his doctoral research in the area of color television image compression and received the Ph.D. degree in 1980 from the University of Texas at Arlington.

He worked in the aerospace industry for over a decade. He helped develop photo-based visual systems technology, which has since found wide acceptance in the flight simulation community. He was the chief architect of TOPSCENE, the mission rehearsal system used by the U.S. Navy, Air Force, and Army pilots to train their air-to-ground missions. His present research interests include visual systems, image processing and virtual reality applications to biomedical, aerospace, and manufacturing.

Dr. Devarajan is an associate fellow of AIAA.