A Simple and Practical Solution to the Rigid Body Motion Segmentation Problem using a RGB-D Camera

Samunda Perera[*†] and Nick Barnes[*†]
[*]CECS, The Australian National University, Canberra, ACT 0200, Australia
[†]NICTA, Canberra Research Laboratory, Canberra, ACT 2601, Australia
{Samunda.Perera, Nick.Barnes}@nicta.com.au

***Abstract*—Motion segmentation with a moving camera has many applications in computer vision. This paper presents a novel method for rigid motion segmentation using a RGB-D camera. The method estimates the lengths of edges in a Delaunay graph of the feature points and observes any variation of these lengths. Non-rigid edges in the graph are identified and pruned. The resulting graph is then examined for connected components to find different motion groups. We present results which show that the approach is able to correctly identify the number of motions and their memberships on real sequences taken from a moving RGB-D camera. The method is fast and suitable for realtime applications.**

I. Introduction

In image segmentation we are interested in segmenting different objects/parts which made up a scene. Likewise given an image sequence/video, the motion segmentation problem is to segment objects which undergo different motion patterns. In the simplest case motion segmentation can mean extracting moving objects from a stationary camera e.g. [1], [2]. In general, the camera can also move which introduces the relative motion of the static background. Motion segmentation has numerous applications in computer vision and robotics including video surveilance in security applications, sports scene analysis, road safety applications in intelligent vehicles, augmented reality, etc.

In computer vision, this problem is formally referred to as the Multibody Structure and Motion (MSaM). Given a set of feature point trajectories the task is to cluster the trajectories into different motion groups (i.e. determining motion memberships), estimate each group's motion parameters and the 3D structure of the points. In general, the number of motions is unknown. Identification of different motion groups requires the simultaneous estimation of motion parameters and motion memberships. Due to this chicken-and-egg nature, the motion segmentation problem is inherently difficult and researchers have made significant and continuing efforts to tackle the problem.

The majority of existing work on MSaM can be grouped into four main methods, namely model selection, factorization based, algebraic and statistical methods. In the model selection method [3]–[6], the first step is to generate candidate motion models using RANSAC [7] style sampling of the point set. Different types of models may include fundamental matrix, affine fundamental matrix, homography, etc. Here the success depends on the probability of finding at least one all-inlier set from each of the model. Points from one motion act as pseudo-outliers to other motions and hence with increased number of motions chances of finding an all-inlier set is low. Therefore the performance deteriorates with the increasing number of motions since the sampling time increases rapidly. To mitigate this issue, methods like local sampling (assuming spatial coherence of points belonging to the same motion) [4], [5] and guiding sampling based on information derived from residual sorting [8] can be used. However the number of generated model samples can still be large and therefore methods for clustering and pruning duplicates would be required. Once candidate motions are generated, a subset is selected as the optimal motions based on the goodness of the fit. However, as the most complex (general) model always fits over a simpler model (e.g. fundamental matrix over affine fundamental matrix) and more motions fits over a limited number of motions, special information criteria which assigns a cost depending upon the model complexity and number of motions are required.

Under an affine camera model feature point trajectories associated with each moving object lie in a low dimensional (2,3,4) linear subspace. Therefore, 3D motion segmentation is equivalent to clustering point trajectories into different motion subspaces. Factorization based methods [9], [10] aim to decompose the trajectory matrix into a motion matrix and a block diagonal structure matrix by finding an unknown permutation matrix which groups trajectories into corresponding motions. However this requires the motion subspaces to be independent which does not always hold in practice. On the other hand GPCA (Generalized Principal Component Analysis) [11] which is an algebraic method, can deal with partially dependent motions and missing data. However it also has drawbacks including sensitivity to outliers and the need for a large number of feature points for an increased number of motions (of the order $O(n^4)$ for n motions). Further, as noted these are confined to an affine camera model.

In statistical methods, the EM (Expectation Maximization) algorithm for MSaM [12] iteratively alternates between membership assignment (E-step) and motion parameter estimation (M-step) until convergence. It is sensitive to the initialization and can converge to a local minima instead of the global minima.

Similar to MSaM for a single camera, motion segmentation has been attempted with stereo cameras as well. However, in the case of stereo or RGB-D determination of the 3D structure of the feature points is straightforward[1] and therefore the motion segmentation problem only involves determination of the motion parameters and membership assignment.

The paper by Agrawal et al. [13] desribes a system that can detect independently moving objects from a mobile platform equipped with a stereo camera. Their system first randomly samples and selects the dominant motion as the ego motion of the camera. This is done by three point sampling, and three point pose estimation [14] followed by scoring in the disparity space. Next the regions that are incompatible with the ego motion are identified and grouped as independent moving objects. A pixel belonging to an independent motion must move at least 2 pixels from its projected location (as a stationary object). The paper gives results only for single independent motion.

The set of rigid body motions lie on the matrix Lie group $SE(3)$. In [15], [16], motion points on $SE(3)$ are generated by three point sampling and mean shift clustering is applied. Interestingly, this method returns the number of motions, motion parameters as well as the memberships. However, the method requires that every input motion point to be converged to a local mode separately which can be considered as a drawback.

In [17] Rabe et al. present *Dense6D* and *Variational6D* algorithms which estimate the dense 3D motion field using spatial and temporal regularization of stereo and optical flow data. The estimated motion field can be used to segment the moving objects. However, these algorithms are computationally intensive in which stereo and optical flow computations runs on a dedicated FPGA and a GPU respectively.

A related problem to MSaM is the non-rigid Structure From Motion (NRSFM). Here the task is to recover the 3D positions of observed points of a non-rigid object over time. In [18], [19] factorization based methods have been applied. In [20] Salzmann et al. solves a system of quadratic equations involving distance constraints between nearby mesh vertices on a triangular mesh surface model. Interestingly the paper by Taylor et al. [21] solves local 3-point N-view ($N \geq 4$) structure from motion problems independently for each triplet and identify 3D triangles consistent with near rigid motion. However the solution presented is for an orthographic camera model.

In this paper we make an important contribution to multi-body structure from motion estimation, a fast algorithm for finding pairs of points which are moving with the same motion. In contrast to the 3-point N-view ($N \geq 4$) affine motion problem [21], given the assumption that we have metric depth from image pairs, our algorithm evaluates a 2-point N-view ($N \geq 4$) rigidity constraint under perspective projection. We demonstrate that this method can segment points on separately moving objects in real sequences from a commercial RGB-D camera, with minimal computational cost.

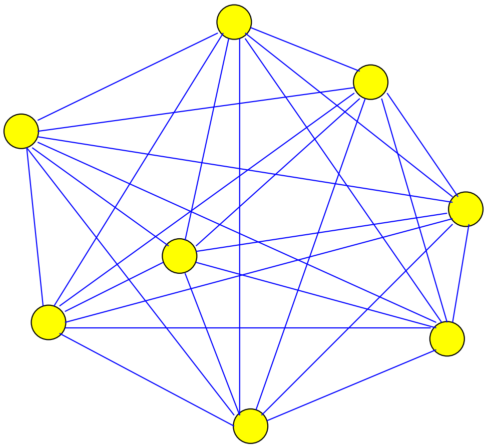


Fig. 1. Complete graph for a rigid body with n vertices and $\frac{n(n-1)}{2}$ edges (n = 8)

II. Our Approach

If accurate depth map computation is possible e.g. from structured light/active stereo as in [22], the 3D structure of a point set relative to the camera frame can be readily obtained. However to express the points w.r.t. the world frame, it is essential to have the pose of the camera in the world frame. On the other hand to recover the ego motion of the camera (e.g. as from [14]) and then to estimate its pose, point correspondances need to be identified from the static background. Unfortunately this is not directly feasible and therefore we are unable to observe the movement of the 3D points in the world frame in a straight forward manner.

Although the 3D positions of points can not be determined in the world frame, the relative position between each pair of points is the same in both camera and world frames. For a set of points on a rigid object, the relative positions and hence distances between each other are invariant over time. This can be visualized as a fully interconnected undirected graph (complete graph) with invariant edge lengths (Figure 1). Therefore for n points on a rigid object we have $\frac{n(n-1)}{2}$ rigidity constraints.

In the case of two points on two rigidly moving objects with non-zero relative motion[2], the distance between them is not invariant and changes over time. Therefore non-zero variation of the distance acts as an indication of the different motions. Note that for a point pair on different objects, the objects could have relative rotation about the points without changing their relative 3D length, hence 3D relative length preservation is a necessary but not sufficient condition to conclude that two objects have no relative motion.

[1] The 3D reconstruction can still be noisy and therefore spatial regularization may be required.

[2] In motion segmentation, if two objects have no relative motion, we can consider them as a single entity.

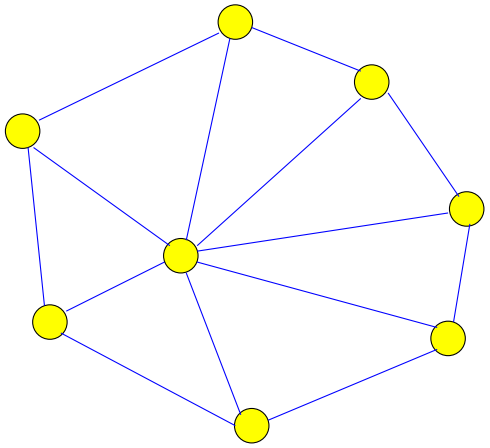


Fig. 2. Graph formed by 2D Delaunay triangulation with $2n-2-h$ triangles and $3n-3-h$ edges where n is the total number of data points and h is the number of points on the convex hull (n = 8, h = 7)

Therefore we propose to measure the standard deviation σ of the distances over time and to reason about different motions. In practice as the point matches and depth estimation are noisy, σ will have non-zero values even for two points on the same rigid object. However σ will have considerably large values for points undergoing different significant motions and therefore a suitable threshold can be used.

The rest of this section explains our motion segmentation pipeline and is organised as follows. In II-A, we describe a technique utilized to reduce the number of rigidity constraint evaluations. Subsection II-B details how changing edge lengths are found using a RGB-D camera. After non-rigid edges are identified (II-C), subsection II-D explains the recovery of motion groups.

A. Delaunay Triangulation

A 2D triangulation is a subdivision of the convex hull of a 2D point set in to triangles (2-simplices) such that any two triangular faces either have an empty intersection or share an edge or a vertex. The 2D Delaunay triangulation for a 2D point set is such a triangulation where no point in the point set is inside any triangle. In 2D the Delaunay triangulation produces a graph with $2n-2-h$ triangles and $3n-3-h$ edges for n data points with h points lying on the convex hull (Figure 2). Therefore the number of edges in the graph and so the complexity of edge evaluation stage decreases from $O(n^2)$ to $O(n)$.

B. Estimation of Edge Lengths

For a given image point $x_i = (u, v)$ and depth z from a RGB-D camera we obtain an estimate $\hat{X_i} = (\hat{X}, \hat{Y}, \hat{Z})^T$ for the 3D position of a point X_i from

$$\hat{X} = \frac{k(u-u_0)z}{f} \tag{1}$$

$$\hat{Y} = \frac{k(v-v_0)z}{f} \tag{2}$$

$$\hat{Z} = z \tag{3}$$

where (u_0, v_0) is the image plane center, f denotes the focal length and k denotes the pixel size of the camera.

As noted the Delaunay triangulation essentially gives us pairings of adjacent vertices x_i, x_j on the image plane. The corresponding edge length estimate $\hat{L_{ij}}$ is given by the 2-norm of the vector $(\hat{X}_i - \hat{X}_j)$ i.e. $\hat{L_{ij}} = |\hat{X}_i - \hat{X}_j|_2$. It is related to the true edge length $L_{ij} = |X_i - X_j|_2$ as

$$\hat{L_{ij}} = L_{ij} + \epsilon \tag{4}$$

where ϵ accounts for the errors from image position and depth estimates.

C. Thresholding of Edge Lengths

Let the standard deviation of each $\hat{L_{ij}}$ over the image sequence be $\hat{\sigma_{ij}}$. Assuming ϵ is zero mean and independent of L_{ij} we get

$$\hat{\sigma_{ij}} = \sqrt{\sigma_{ij}^2 + \sigma_\epsilon^2} \tag{5}$$

where σ_{ij}, σ_ϵ are the standard deviations of L_{ij} and ϵ respectively. As noted if two points belong to the same rigid object then $\sigma_{ij} = 0$. Therefore $\hat{\sigma_{ij}}^{rigid} = \sigma_\epsilon$.

However if two points belong to different rigid bodies undergoing relative motion then L_{ij} would change over the image sequence and therefore $\sigma_{ij} > 0$. We assume σ_ϵ will be small in comparison to any large movement of points. Hence $\hat{\sigma_{ij}}^{non-rigid} \approx \sigma_{ij}$. Therefore we compute $\hat{\sigma_{ij}}$ using $\hat{L_{ij}}$ data over the image sequence and identify non-rigid edges by using a practical threshold. Naturally small movements will be difficult to differentiate from noisy observations. The non-rigid edges thus identified are pruned from the initial Delaunay graph.

D. Connected Components

The resulting graph after thresholding is examined to find the connected components. We use the breadth-first search algorithm to find the connected components. Starting at a particular vertex the connected component i.e. set of vertices connected together by paths is extracted and the process is repeated for the vertices which are not included in previously found connected components. The final result of this stage is the number of connected components and their individual vertices. This gives us the number of motion groups and the membership assignment for each group.

III. Implementation

A commercially available active stereo RGB-D camera was interfaced with a laptop computer using the OpenNI SDK [23]. 24 bit RGB and 16 bit Depth (which provides depth in millimeters) images were captured at 640x480 resolution. The Depth image was registered with the RGB image using available API functions. Due to the difference in RGB and Depth camera viewpoints, the effective size of the registered Depth image was about 588x424 (see Figure 4). For pixels with no depth information (e.g. due to registration, occlusion, etc.), a depth value of zero was returned. The camera parameters were obtained as $f = 120$ pixels, $k = 0.20838$ mm and $(u_0, v_0) = (320, 240)$.

In an image sequence, each RGB image was converted to grayscale and SIFT [24] keypoints and descriptors were extracted. The keypoints corresponding to points with no depth information were thrown away. The descriptors were matched across the sequence and only points that survived matching in all of the images were retained.

IV. Experimental Results & Discussion

We present results using three real image sequences captured with the RGB-D camera. The sequences are denoted **3 Books**, **Box** and **Calender-Books**.

The **3 Books** image sequence consists of six images of three books moving away from each other (Figure 3). Here the object motions were predominantly translations with occasional slight rotations. The images were captured in different time instants under varying camera viewpoints. A sample depth image from this sequence is shown in Figure 4.

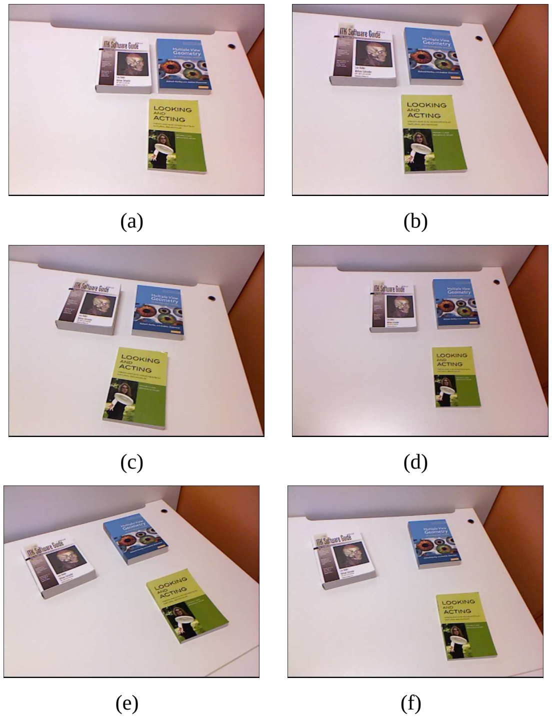

(a)
(b)
(c)
(d)
(e)
(f)

Fig. 3. **3 Books** Image Sequence

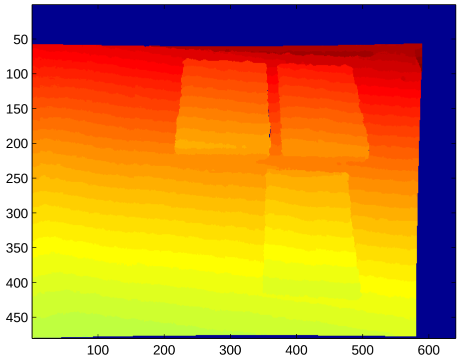


Fig. 4. The Registered Depth Image corresponding to the first image of **3 Books** Image Sequence (Figure 3 (a))

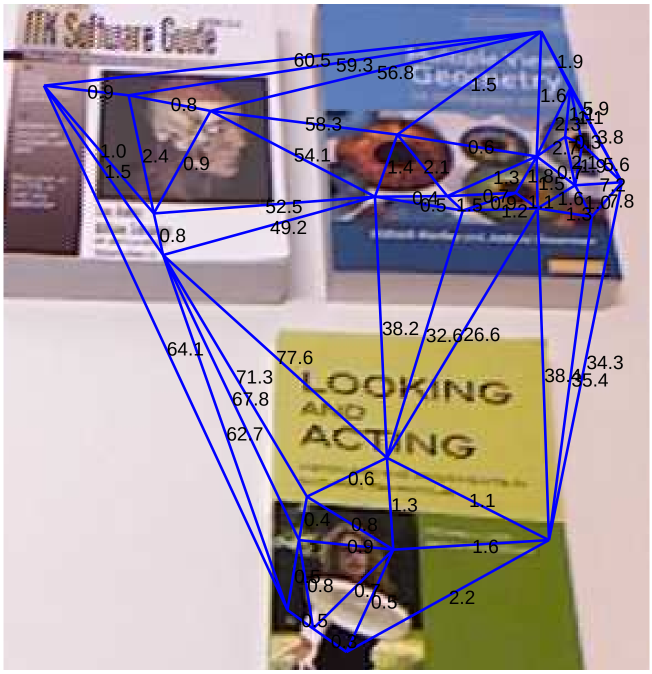
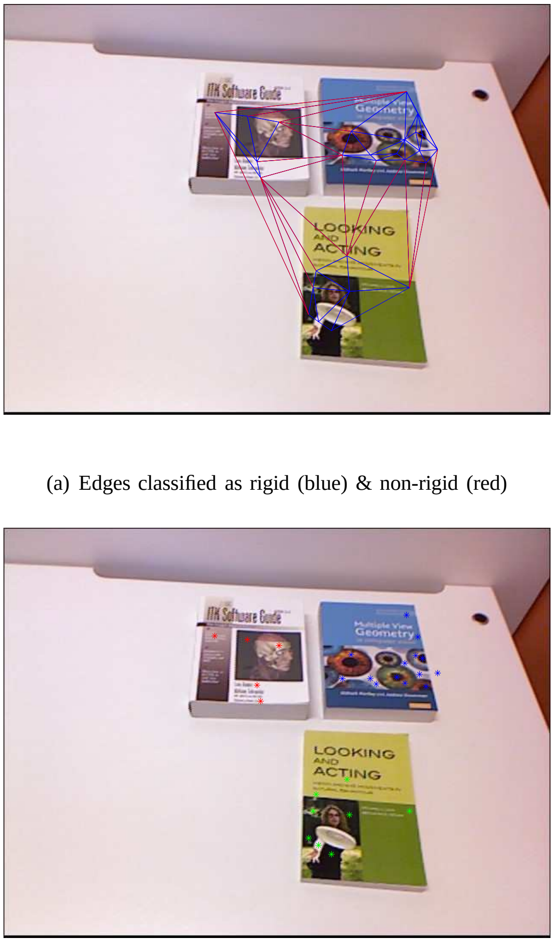


Fig. 5. Delaunay graph and edge length standard deviations for the **3 Books** Image Sequence

SIFT feature matching resulted in 27 point trajectories and the Delaunay triangulation on the initial image resulted in a graph with 71 edges (Figure 5). The figure also shows the standard deviation of edge length for each of the edges (best viewed when on-screen and zoomed in).

A practical threshold of $\sigma_{TH} = 20$ mm was used to identify the non-rigid edges. Figure 6(a) shows the edges thus identified marked in red. Clearly all the non-rigid edges which separate the three moving objects were correctly identified. Three connected components were found (Section II-D) and the resulting memberships are shown in Figure 6(b).

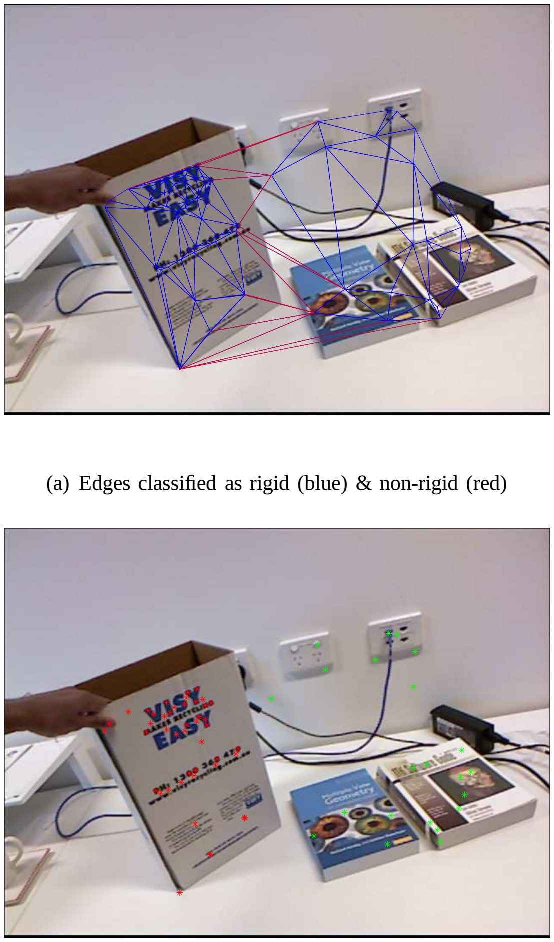
The **Box** image sequence consists of 5 images of a box moving under translation while the camera was also moved parallel. The number of SIFT features and the edges in the



(a) Edges classified as rigid (blue) & non-rigid (red)

(b) Motion memberships

Fig. 6. Results for the **3 Books** Image Sequence



(a) Edges classified as rigid (blue) & non-rigid (red)

(b) Motion memberships

Fig. 7. Results for the **Box** Image Sequence

Delaunay graph were 67 and 185 respectively. Figure 7 shows the results for this sequence ($\sigma_{TH} = 20$ mm). As expected two motions were detected with correct memberships.

The **Calender-Books** sequence consists of 4 images of a stationary calendar and three books moving away from each other (42 SIFT points). The Delaunay graph had 112 edges. The motion of the upper two books w.r.t calender was relatively small. Results of motion segmentation using a threshold of $\sigma_{TH} = 5$ mm are given in Figure 8.

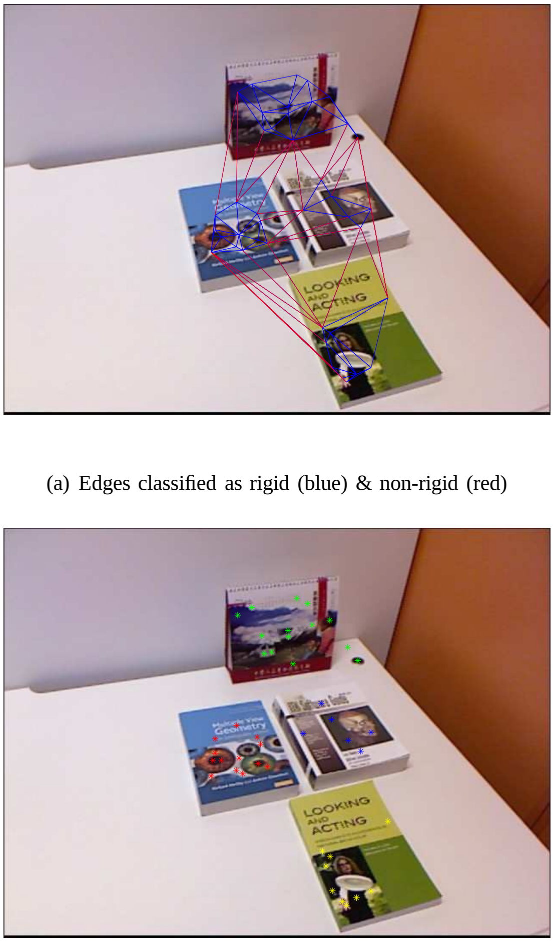
Identifying an edge as rigid/non-rigid is essentially a binary classification problem. We define the binary test as “Is $\sigma > \sigma_{TH}$?”. Then Figure 9 shows how misclassifications vary with the threshold for each of the three sequences. Here FP, FN, P, N stands for the counts of false positives (rigid edges identified as non-rigid), false negatives (unidentified non-rigid edges), non-rigid edges and rigid edges respectively.

In **3 Books** and **Box** sequences, object(s) undergo significant motion. For the **3 Books** sequence, Figure 9(a) clearly shows that there are zero misclassifications when $\sigma_{TH} \in [8, 26]$ i.e. a 18 mm margin. In the misclassification plot for the **Box** sequence (Figure 9(c)) rigid and non-rigid motions are separated by more than 50 mm margin. In contrast **Calender-Books** sequence exibits zero misclassifications only in a narrow region. However this is to be expected, since the relative motion between the calender object and the nearby two books are small (This sequence was prepared to examine the limitations of separating small motions from noise).

Typical execution times for various stages of the processing for each of the sequences are given in Table I. Here the runtimes reported are based on an unoptimised MATLAB implementation running on single thread of a Corei7 1.6 GHz laptop computer.

V. Conclusion

We have presented a method to the rigid body motion segmentation which tests for the rigidity of 2-points in N-views ($N \geq 4$). To the best of our knowledge this is the first time motion segmentation has been attempted with a RGB-D camera. The method is very fast and suitable for realtime



(a) Edges classified as rigid (blue) & non-rigid (red)

(b) Motion memberships

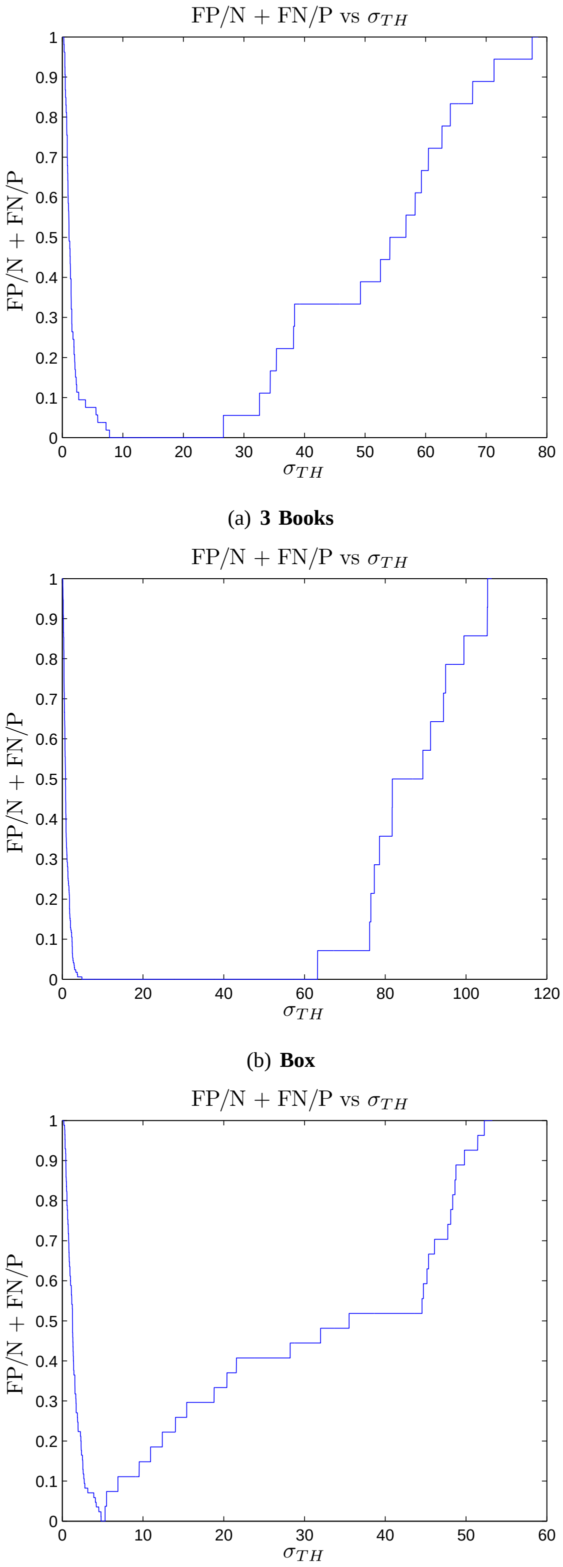
Fig. 8. Results for the **Calender-Books** Image Sequence

TABLE I
PROCESSING TIMES

	Time (ms)		
	3 Books	**Box**	**Calender-Books**
Delaunay & Edge extraction	7	7	7
Edge Length	2	4	2
Std. & Thresholding	12	21	16
Connected Component	12	12	12

applications.

One limitation of the work is the use of a global threshold for rigity/non-rigidity testing. The scale of noise was assumed to be independent of the length of an edge. However, in reality the error in edge length depends on the errors in the 3D position estimates of the vertices. Unfortunately the error in depth direction is currently unavailable for the RGB-D camera used. In future work we plan to model this error and thereby use local threshold for each of the edges. Moreover the work in progress include incorporating a real time feature tracker



(a) **3 Books**

(b) **Box**

(c) **Calender-Books**

Fig. 9. FP/N + FN/P vs σ_{TH} (which is in mm) for the three Image Sequences (Note: FP = no. of false positives, FN = no. of false negatives, P = no. of non-rigid edges, N = no. of rigid edges)

instead of the SIFT feature matching which is relatively slow. Further, we will extend our result to full MSaM.

ACKNOWLEDGMENT

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications, and the Digital Economy, and the Australian Research Council (ARC) through the ICT Centre of Excellence Program. This research was also supported in part by ARC through its Special Research Initiative (SRI) in Bionic Vision Science and Technology grant to Bionic Vision Australia (BVA).

REFERENCES

[1] R. Jain and H.-H. Nagel, "On the analysis of accumulative difference pictures from image sequences of real world scenes," no. 2, pp. 206–214, 1979.

[2] Y. Sheikh and M. Shah, "Bayesian object detection in dynamic scenes," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2005*, vol. 1, 2005, pp. 74–79.

[3] K. Kanatani, "Geometric information criterion for model selection," *International Journal of Computer Vision*, vol. 26, pp. 171–189, 1998, 10.1023/A:1007948927139. [Online]. Available: http://dx.doi.org/10.1023/A:1007948927139

[4] P. H. S. Torr, O. Faugeras, T. Kanade, N. Hollinghurst, J. Lasenby, M. Sabin, and A. Fitzgibbon, "Geometric motion segmentation and model selection [and discussion]," *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 356, no. 1740, pp. pp. 1321–1340, 1998. [Online]. Available: http://www.jstor.org/stable/54848

[5] K. Schindler and D. Suter, "Two-view multibody structure-and-motion with outliers through model selection," vol. 28, no. 6, pp. 983–995, 2006.

[6] H. Li, "Two-view motion segmentation from linear programming relaxation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR '07*, 2007, pp. 1–8.

[7] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, June 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[8] T.-J. Chin, J. Yu, and D. Suter, "Accelerated hypothesis generation for multi-structure robust fitting," in *Computer Vision ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin / Heidelberg, 2010, vol. 6315, pp. 533–546.

[9] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, pp. 159–179, 1998, 10.1023/A:1008000628999. [Online]. Available: http://dx.doi.org/10.1023/A:1008000628999

[10] K. Kanatani, "Motion segmentation by subspace separation and model selection," in *Proc. Eighth IEEE Int. Conf. Computer Vision ICCV 2001*, vol. 2, 2001, pp. 586–591.

[11] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," vol. 27, no. 12, pp. 1945–1959, 2005.

[12] A. Gruber and Y. Weiss, "Multibody factorization with uncertainty and missing data using the em algorithm," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2004*, vol. 1, 2004.

[13] M. Agrawal, K. Konolige, and L. Iocchi, "Real-time detection of independent motion using stereo," in *Proc. IEEE Workshop Motion and Video Computing WACV/MOTIONS '05 Volume 2*, vol. 2, 2005, pp. 207–214.

[14] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," vol. 13, no. 4, pp. 376–380, 1991.

[15] O. Tuzel, R. Subbarao, and P. Meer, "Simultaneous multiple 3d motion estimation via mode finding on lie groups," in *Proc. Tenth IEEE Int. Conf. Computer Vision ICCV 2005*, vol. 1, 2005, pp. 18–25.

[16] R. Subbarao and P. Meer, "Nonlinear mean shift over riemannian manifolds," *Int. J. Comput. Vision*, vol. 84, pp. 1–20, 2009.

[17] C. Rabe, T. Mller, A. Wedel, and U. Franke, "Dense, robust, and accurate motion field estimation from stereo image sequences in real-time," in *Computer Vision ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin / Heidelberg, 2010, vol. 6314, pp. 582–595. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15561-1_42

[18] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3d shape from image streams," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, 2000, pp. 690–696.

[19] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd, "Coarse-to-fine low-rank structure-from-motion," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2008*, 2008, pp. 1–8.

[20] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua, "Closed-form solution to non-rigid 3d surface registration," in *Computer Vision ECCV 2008*, ser. Lecture Notes in Computer Science, D. Forsyth, P. Torr, and A. Zisserman, Eds. Springer Berlin / Heidelberg, 2008, vol. 5305, pp. 581–594. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88693-8_43

[21] J. Taylor, A. D. Jepson, and K. N. Kutulakos, "Non-rigid structure from locally-rigid motion," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2761–2768.

[22] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, vol. 1, 2003.

[23] "Openni." [Online]. Available: http://www.openni.org

[24] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004, 10.1023/B:VISI.0000029664.99615.94. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94