Robotica (1999) volume 17, pp. 649–660. Printed in the United Kingdom © 1999 Cambridge University Press

Robust segmentation of visual data using ranked unbiased scale estimate

Alireza Bab-Hadiashar and David Suter

Intelligent Robotics Research Centre, Department of Electrical & Computer Systems Engineering, Monash University, Clayton Vic. 3168 (Australia)
e-mail: {ali.bab-hadiashar, d.suter)@eng.monash.edu.au

(Received in Final Form: June 1, 1999)

SUMMARY
A method of data segmentation, based upon robust least K-th order statistical model fitting (LKS), is proposed and applied to image motion and range data segmentation. The estimation method differs from other approaches using versions of LKS in a number of important ways. Firstly, the value of K is not determined by a complex optimization routine. Secondly, having chosen a fit, the estimation of scale of the noise is not based upon the K-th order statistic of the residuals. Other aspects of the full segmentation scheme include the use of segment contiguity to: (a) reduce the number of random sample fits used in the LKS stage, and (b) to "fill-in" holes caused by isolated miss-classified data.

KEYWORDS: Robust segmentation; Visual data; Scale estimate; LKS method; Robust statistic

1. INTRODUCTION

The majority of visual tasks (surveillance, inspection, recognition, obstacle avoidance, etc.) either rely on, or could be enhanced by accurate data segmentation.

Any data segmentation algorithm is faced with the dilemma of combining two contradictory elements: separation and continuity. A successful segmentation algorithm is a right combination of forceful division at boundaries of each cluster of data while preserving the continuity of the cluster. The approach proposed here, is a form of parametric segmentation based on least K-th squares (LKS) robust statistics. The concept of parametric data segmentation has a wide range of application. In particular, the segmentation task is a vital part of various computer vision problems such as recovering structure from motion, range, etc.

It will be shown that the new segmentation tool can be adapted for the motion segmentation problem. Solution to this problem is very important for both passive and active vision based navigation systems. Moreover, new developments in video coding (e.g. MPEG4) are based on having reliable (and automated) motion segmentation tools.[1]

Although the data segmentation problem can be formulated in many different ways, we choose to approach this problem as a multi-structural parametric fitting problem. In this context, the segmentation problem is similar to robust statistical regression. The main difference is that the regression in robust statistics usually involves statistics for data having *one* target distribution and corrupted with random outliers. Accordingly, most of the robust statistic tools are prepared by assuming that there is always a cluster, with absolute majority, in the data which has only one target distribution.[2,3]

Here, we extend this notion to problems where there are a number of populations with distinct distributions. In this case, every population of data will be an outlier group, or "pseudo-outlier",[4] to every other population. Moreover, there won't necessarily exist a population having absolute majority.

To provide some background, the history of robust statistics and some recent developments in the computer vision arena are briefly reviewed first in (section 1.1). An overview of motion segmentation literature is also presented (section 1.3). In section 2, a novel way of segmenting multi-structrual data similar to other LKS based approaches is introduced and the advantages of our approach over similar methods are highlighted. The experimental performance of this method is also investigated in this section. We introduce our approach to range segmentation in section 3. Section 4 contains the results of our experiments for segmenting synthetic and real image sequences. A brief conclusion is provided in section 5.

1.1 Robust estimator

The problem of fitting an a priori known model to a set of noisy data (with random outliers) was studied in the statistical community for a number of decades without much success. The break-through came in 1984, when the Least Median of Squares (LMS) was introduced.[5] LMS has found many applications in a wide variety of engineering problems during the last decade. Computer vision is no exception to this. On the other hand, different groups of researchers in computer vision have discovered limitations, which are associated with applying LMS to computer vision problems.[4,6]

The main limitation, and one that has motivated the development of other robust methods, is the fact that the break down point of the LMS is 50%. This means that the LMS technique needs the population recovered to have at least a majority of 50% (plus 1) of all the data, to ensure the return of the true fit. Since the LMS always returns an answer, it will "hallucinate" a fit to some part of data that

https://doi.org/10.1017/S0263574799001812 Published online by Cambridge University Press

minimizes the median of square residuals even for multi-structural data (e.g. "bridging fit"). This fit can be far from any underlying structure embedded in the data (as minority groups). To overcome the above problem, which is frequently encountered in computer vision tasks, a number of other estimators have been introduced by assuming some characteristics for outliers, for inliers, or for both. Lee et al.[7] proposed a robust estimator named the Adaptive Least K-th Order Squares (ALKS). In this technique, the estimator searches for the model, which minimizes the K-th order statistics of the squared residuals where the so-called optimum value for the K is determined from the data. The value of K is determined as the K which minimizes the ratio of square unbiased scale estimate (using the preliminary "inliers" found by LKS) divided by the square K-th order scale estimate based on rank ordering statistics of normal distribution.

Similarly, Miller and Stewart[8] proposed a robust estimator named the Minimum Unbiased Scale Estimator (MUSE). In this method, the value of K is determined as the K, which minimizes an unbiased scale estimate of the ordered residuals.

In both approaches, first the proposed estimator tries to find the size of the biggest minority population (K) within the whole set of data through minimization of some scaled measure of variance of residuals found by preliminary LKS fit. Then, the least K-th order statistics technique is used to find the embedded structure. To solve the LKS minimization problem, both approaches rely on random sampling of the data based on the assumption that a high probability of choosing a good sample (a sample from the majority) of data can be achieved with a low number of samples.

There is a problem with both approaches, which will be described here. The minimization process in both approaches for finding the true value of K, relies on knowing the right value of K (thus, it is a "chicken and egg" problem). The estimation of scale using order statistics assumes there is only one underlying distribution for the data. In case of multi-structural data, the scale cannot be truly estimated from rank ordering statistics as the overall ranks (percentile) of residuals from data drawn from multiple distributions will be different from their ranks in their own distribution.

To overcome the computational complexity of the above problem, we proposed a novel method[9] named Selective Statistical Estimator (SSE) whereby one does not seek to optimize K. This can be seen as relaxing the requirement of demanding an estimator to find the best (with any measure of goodness) structure embedded in the data first. Put simply, SSE is proposed as a variation of the Least K-th Order Statistics data regression where the user proposes the K as the lower limit of the size of populations one is interested in. After deciding what would be the minimum acceptable population for every structure (K), then one follows a similar procedure to that of the Least Median of Squares.[3] The only exception is that, instead of the sub-sample that has the smallest median of square residuals, the sub-sample that has the smallest K-th order least squares is found. This will arbitrarily return one of the structures, which has a population of at least K points. By scaling all the residuals with respect to the returned structure, using the same recipe as described by Rousseeuw and Leroy,[3] all the points that belong to this structure will be found. The only difference is that the scale for SSE is found using K-th order statistics of the residuals (as opposed to the median of the residuals) with different correction factor for different K.

Although this approach finds a fit based on an assumed value of K and therefore can guarantee the return of a good fit, it had the same limitation as MUSE and ALKS: it used the rank ordering statistic for estimating the scale of residuals. Since the exact size of every population is not known a priori, the rank ordering based estimation of scale will not be accurate.

Our segmentation algorithm is based on SSE and follows the same principal for finding the first fit. We then find the scale from the unbiased scale *estimate without using the rank ordering statistics.* The details of our method are provided in section 2.

1.2 Range segmentation

Unlike motion based segmentation (reviewed in the next section), range data segmentation is a relatively well-known problem in computer vision arena. Thus, a survey of different range segmentation methods is not included in this paper and we refer the interested reader to the survey of Hoover et al.[10] for theoretical and experimental details of different range segmentation algorithms. It should however be noted that, although LKS based approaches to range data segmentation has been considered before,[7,8,11] the presented method differs from its predecessors in the use of a different robust estimator of scale (section 2).

1.3 Motion segmentation

Segmenting a sequence of images based on the coherent motion of different objects in a dynamic scene is commonly known as motion (or spatio-temporal) segmentation. Following Dufaux and Moscheni,[12] we categorize the existing segmentation methods into the following groups:

- Optic flow based segmentation
- Simultaneous or sequential recovery of motion and segmentation
- Fusion of motion recovery and static segmentation.

These approaches are briefly explained here.

In the first approach, a dense optic flow field is recovered first and the segmentation is performed by fitting a model (often affine) to the computed flow field.[13–19] Since reliable computation of optic flow by itself is a daunting task (and often requires expensive computations), these methods will likely be limited to off-line applications for a number of years to come. Moreover, separating the two processes causes the error associated with estimating the motion to propagate into the segmentation stage.

To resolve the problems associated with the first type of approach, a number of methods for simultaneously[20–22] or sequentially[23,24] recovering the motion and performing segmentation have been proposed. In these techniques, the segmentation is often formulated using a Markov Random Field (MRF) based relaxation problem. Since these method

https://doi.org/10.1017/S0263574799001812 Published online by Cambridge University Press

rely on recovering a dominant motion, which in turn can be extremely unreliable due to camera motion (there may not be a distinct dominant motion), the performance of these methods may vary based on scene structure.

The third approach to motion segmentation is to combine the static (based on the intensities of a single image) and dynamic segmentation. Thompson[25] was the first to use a similarity constraint on brightness and motion to achieve motion segmentation using a region-merging logarithm. Black,[26] Schroeter and Ayer[27] and Dufaux et al.[28] have proposed methods in which their algorithms are initiated by a static segmentation of the scene. This static segmentation is then further refined in regions where the motion estimation process has failed. Gelgon and Bouthemy[29] used a region-level graph labeling approach to combine the static and dynamic segmentations. Since the support area for estimating the motion is chosen based on the static segmentation, biases in the motion estimation are likely to mislead the segmentation algorithm.

In concluding this section, it should be noted that there exist motion based segmentation techniques which do not use the dense motion estimation approaches just outlined. For instance, Torr and Murray[30] proposed using the fundamental matrix for motion segmentation purposes. Such approaches generally only segment a small number of tracked features.

2. ESTIMATION OF SCALE

As mentioned in section 1.1, our estimation algorithm is based on SSE and it is named the Modified Selective Statistical Estimator (MSSE). The estimation process starts by choosing the smallest number of points (K′) that can be considered as a structure. Then the process finds the fit, which has the least K′-th order square residual using random sampling.

The original version of SSE computes the scale for residuals by using the K-th order statistics of normal distributions. As mentioned before, in multi-structural data, different segments of data have different target distributions and thus the K-th order statistics of the combined data would not be the same as the K-th order statistics drawn from single data segment alone. Therefore, using the K-th order statistics of the residual population is not justified.

To estimate the scale, after finding a fit, we sort the square residuals in an ascending order and calculate the unbiased scale estimate using the first n-th residuals as:

$$\sigma_n^2 = \frac{\sum_{i=1}^{n} r_i^2}{n-p} \tag{2.1}$$

where p is the dimension of model (for a large data set the size of p is not important for calculating the scale). As we increase n, the value of the n-th residual jumps when it comes from data having different target distribution. The first *outlier* residual should be classified as such by:

$$r_{n+1}^2 > T^2 \sigma_n^2 \tag{2.2}$$

where T is a constant factor and is often chosen as 2.5 to identify the 93 percentile of a normal distribution as inliers (the scheme is not sensitive to this value, as the jump is often much larger than 2.5 times of the scale). Combining 2.1 and 2.2, we have:

$$\frac{\sigma_{n+1}^2}{\sigma_n^2} > 1 + \frac{T^2 - 1}{n - p + 1} \tag{2.3}$$

Since the validity of the fit relies on the first K-th points being part of the recovered segment, we start at n = K + 1, and by checking the validity of the above inequality for the ordered square residuals, the transition point and the residual scale are found simultaneously.

This method of computing scale does not rely on the K-th order statistics (it does rely indirectly on the K-th order statistics for one fixed K in so far as minimizing the statistic defines the preliminary fit. However, this only requires that a good fit minimizing this statistic – not that the statistic is a good estimate of scale) and therefore is not biased for the case of multi-structural data from different target distributions. Moreover, we have empirical evidence that this method is capable of distinguishing between a true fit and a bridging fit because a bridging fit does not produce any spike in the scale ratio (LHS of inequality 2.3). This is shown in the following example. In this example, two linear structures are considered (corrupted by Gaussian noise with standard deviation of 0.15) and the MSSE is applied. Figure 1 shows the first linear structure (solid line) and all the associated points to this structure are marked with circles. Figure 2 shows both sides of inequality (2.3). As can be seen

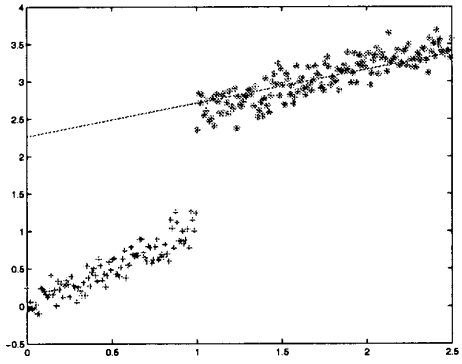


Fig. 1. Two linear structures. The solid line is the result of applying MSSE. All the points associated with the first fit are marked with circles – the segmentation is correct.

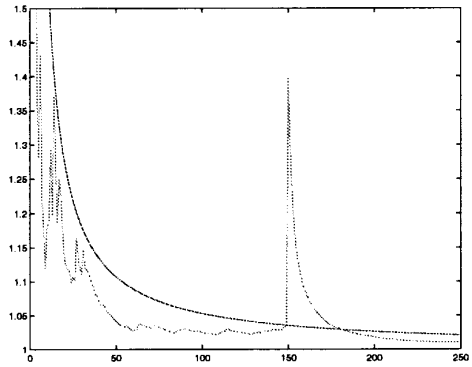


Fig. 2. Both sides of the inequality 2.3 are plotted. The intersection of these two graphs provides the unbiased scale estimate.

https://doi.org/10.1017/S0263574799001812 Published online by Cambridge University Press

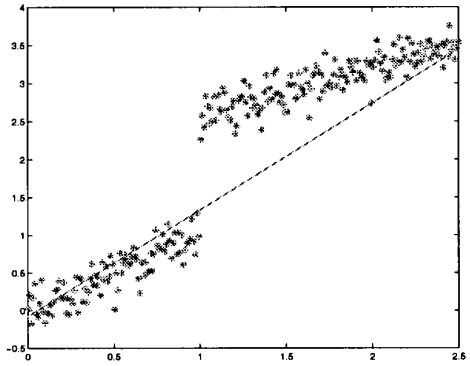


Fig. 3. Data samples (same as in figure 1) but the bridging fit is returned.

in this figure, the graph of scale ratio has a significant spike at the same place as the discontinuity in the data and the calculated scale is 0.1519 which corresponds very favorably with the true scale.

The following figures display the results of the same experiments performed without using enough samples to find a sample entirely in one population. Thus, a bridging fit is returned (solid line in figure 3). As can be seen in figure 4, the bridging fit does not create a spike and can easily be recognized. For this recognition process to work reliably, the data has to have at least a few outliers (something that is almost always guaranteed with real data).

3. RANGE SEGMENTATION

In this section, it will be shown that the proposed estimator can be used to robustly segment range data. Range data are usually clean and accurate and although they are made of multi-structures (at least in our examples), they do not have impulsive outliers. Thus, the range data segmentation task probably does not challenge our methods as greatly as the motion segmentation task. However, clearly there is still some utility in employing robust segmentation methods on range data.[8] We will exploit the full potential of our method when we consider the motion segmentation problem. Although the proposed method does not depend on the choice of model representing the data, for the sake of simplicity, we consider man-made environments where objects are made of either planar or quadratic surfaces. Thus, every segment of data is described by a 3-D plane as:

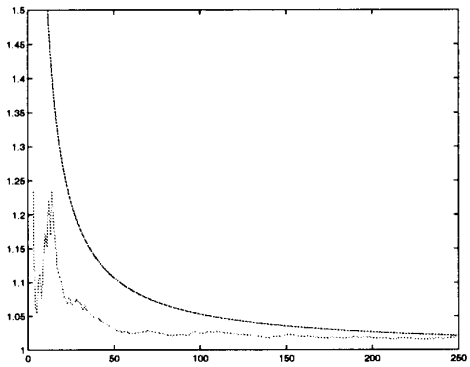


Fig. 4. Same as figure 2. As can be seen, these two graphs do not intersect in the case of bridging fit.

$$f(x, y, z) = ax + by + cz = 1 \tag{3.1}$$

or

$$f(x, y, z) = ax^2 + by^2 + cz^2 + dx + ey + fz = 1 \tag{3.2}$$

The experiments are repeated for five cases. In the first two experiments, the range data representing a dodecahedron and icosahedron were segmented. In the third experiments an object which is made of parallel surfaces is placed on a table. All the objects in these three experiments are made of planar surfaces and the linear model shown in equation 3.1 is used. Figures 5, 6, and 7 show the results of these experiments. Figures marked with “a” show the intensity image while the figures marked with “b” and “c” depict the original and segmented range images, respectively.

In the last two experiments for range segmentation, the objects of experiments are made of either quadratic or combination of planar and quadratic surfaces. For these experiments, the simplified quadratic model which is shown in equation 3.2 is used. Figures 8 and 9 show the results of these experiments. Similar to previous sets of results, figures marked with “a” show the intensity while the figures marked with “b” and “c” depict the original and segmented range images, respectively.

4. MOTION SEGMENTATION

Having developed a robust estimator capable of segmenting multi-structural data, it is now possible to introduce our motion segmentation algorithm. It should be noted here that the other robust estimators (ALKS, MUSE, etc.) have only been tested for range data segmentation and have not been applied to such a complex problem. Therefore, this section goes well beyond presenting just another application of a robust estimator to a computer vision problem. Before outlining the algorithm, there are two important issues which need to be explained.

First of all, in any motion segmentation scheme, the main interest is to find the boundaries of different objects with distinct motions. This implies that the object occupies some contiguous space on the image plane. In other words, we implicitly assume that every object of interest has at least non-vanishing spatial extent on the image plane.

To enforce this assumption into our previous estimator, we add a median filter to the MSSE algorithm. The size of this filter depends on how small any object of interest can be in the considered application. In our implementation, we always assume that every object is bigger than half of a 9 pixels wide square.

The second important issue is the fact that the number of samples required to ensure a high probability of choosing at least one preliminary fit that belongs to one of the structures, grows exponentially with the percentage of contamination and the dimension of the parameter space.[3] Therefore, it is computationally not feasible to apply the MSSE (or any other known robust estimator) directly to motion segmentation in real world applications (where the models used typically have 8 or more parameters). To overcome this hurdle, an efficient way of finding at least one good sample using a few random samples will be introduced which considerably reduces the computational complexity

https://doi.org/10.1017/S0263574799001812 Published online by Cambridge University Press

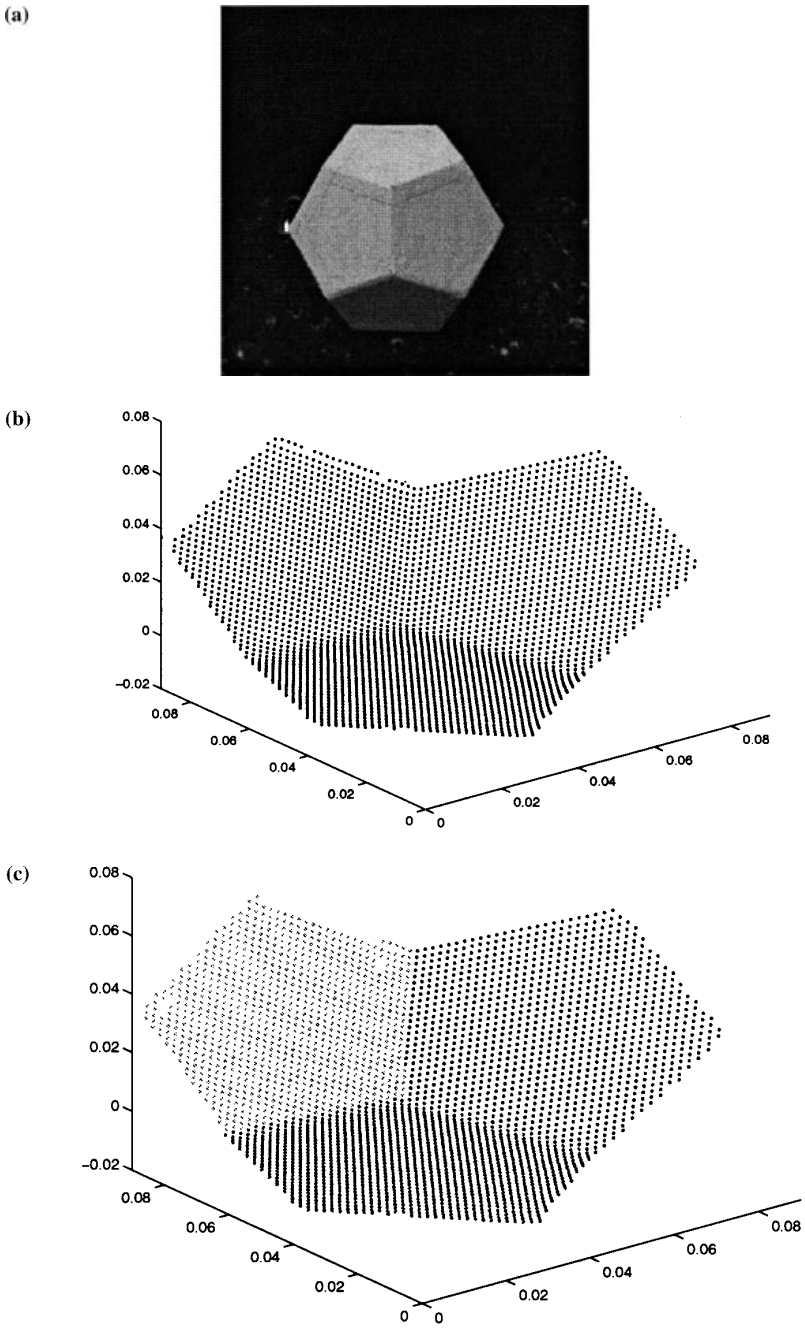


Fig. 5. (a) The scene of the first experiment. (b) The original range image of the first experiment. (c) The segmented range image of the first experiment.

of the proposed segmentation algorithm (see step 5 and 6 below).

4.1 Motion segmentation algorithm

For the sake of clarity, we write the algorithm in a sequential order:

(i) Estimate the spatio-temporal derivatives of the image brightness function. The precise form of estimate, whilst important for accuracy, is not essential to our proposal. We choose to use, for our experiments, convolutions with derivatives of Gaussian functions with equal spatial and temporal standard deviation. In our experiments, both deviations are always set to 1.5 pixels for real data and 1.0 for synthetic data.

(ii) Since the perception of motion requires objects in the scene to have some texture, the first group in our segmentation represents the collection of points with very small spatial derivatives. The second group represents the stationary points where the temporal derivative is very small. From a motion point of view, there would be no distinction between these two groups (both groups would be perceived as stationary). However, in some applications like autonomous navigation, the distinction is quite useful. In our experiments, the points where the sum of square spatial or the sum of square temporal derivatives is below 3% or 1.5% of the average value for the entire image, are grouped together. The scheme is insensitive to these values and little changes were observed when these thresholds were multiplied by 5.

https://doi.org/10.1017/S0263574799001812 Published online by Cambridge University Press

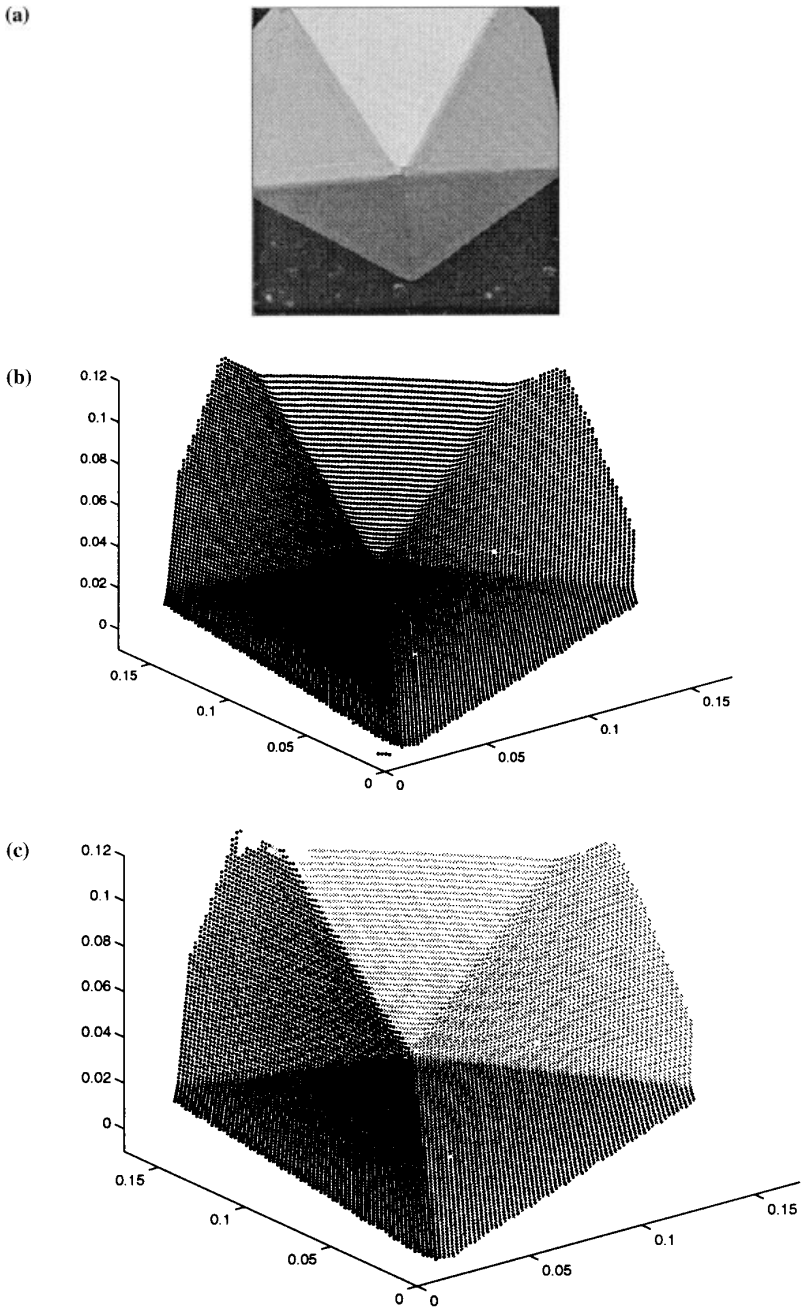


Fig. 6. (a) The scene of the second experiment. (b) The original range image of the second experiment. (c) The segmented range image of the second experiment.

(iii) Employ a motion model, which represents the transformation of the 3-D object motion onto the image plane. For our experiments, we use the projection of a planar surface with rigid motion. The velocity model for this case can be written as:[31]

$$u = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy \quad (4.1)$$

$$v = a_6 + a_7 x + a_8 y + a_5 y^2 + a_4 xy \quad (4.2)$$

where u and v represents the horizontal and vertical velocities and (x, y) are image coordinates.

(iv) Create an over-determined set of linear equations by combining the well-known Optic Flow Constraint $I_x + I_y v + I_t = 0$[32] and the above motion model for all the pixels that remain after step (ii). This set of equations contains one linear equation (in terms of eight unknowns, a_1 to a_8) for every pixel in the image.

(v) Randomly choose a point from the remaining pixels (after step ii) and compute the optic flow using the Optic Flow Constraint and the *LMS fit assuming constant flow* for a 15 pixels wide square centered at the chosen pixel (see reference 19 for details). This step determines all the pixels in the square, which are associated with the computed flow, i.e. inliers.

(vi) Solve the subset of above system of linear equations for planar rigid motion (step iv) associated with the pixels

https://doi.org/10.1017/S0263574799001812 Published online by Cambridge University Press

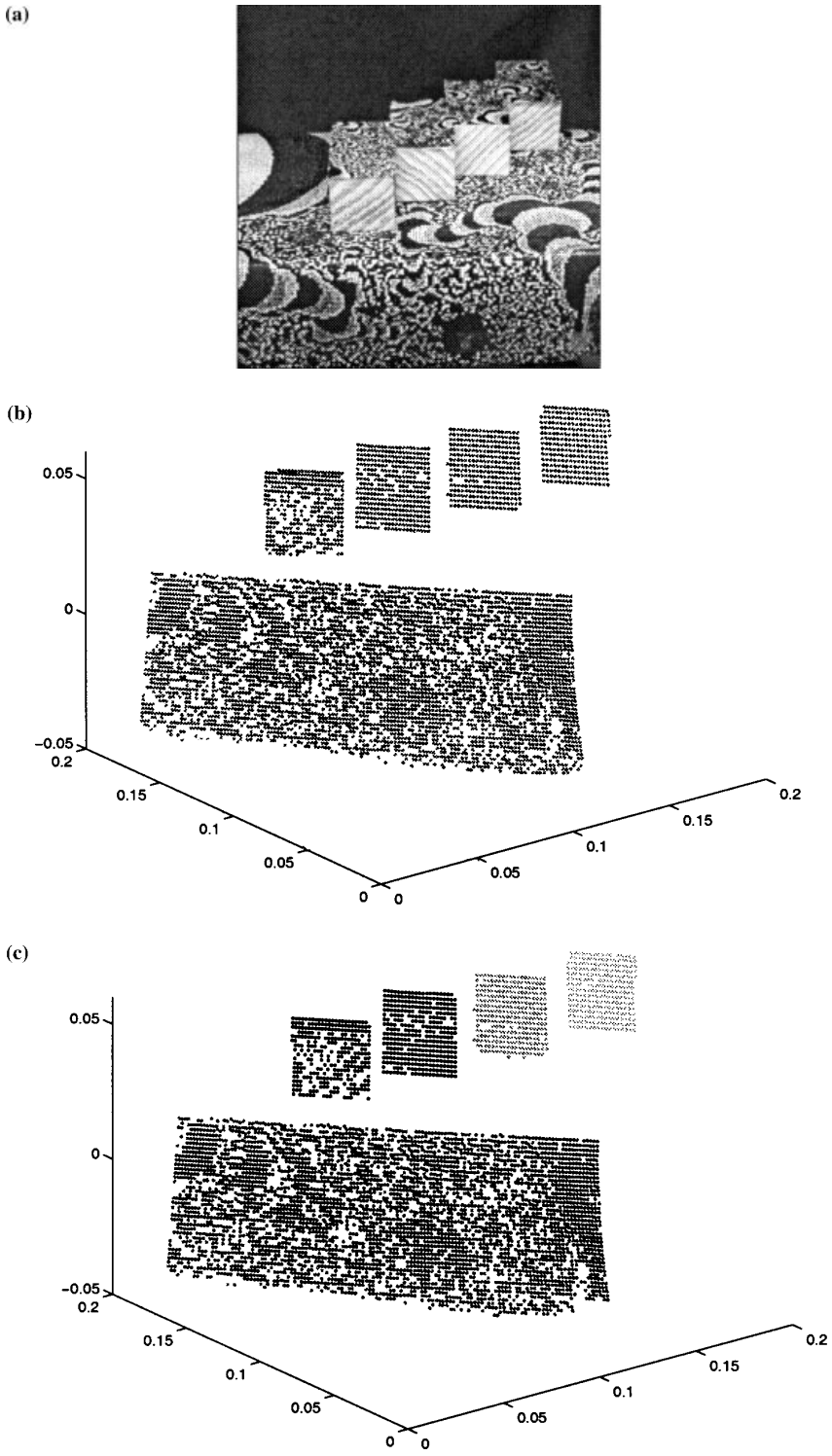


Fig. 7. (a) The scene of the third experiments. (b) The original range image of the third experiments. (c) The segmented range image of the third experiments.

considered as inliers in previous step using least squares. Since the inliers from step 5 are determined robustly, this process has a high probability of providing a good fit for some part of the image. Nevertheless, as step five uses LMS and can produce erroneous flow (under specific circumstances, see reference 33) these two steps must be repeated a few times to ensure the return of a good fit. Instead of repeating the above steps, one can apply the reliability check of Bab-Hadiashar and Suter[33] and make sure that the step five result is reliable.

(vii) Set a value for K′ based on the smallest size of data that could be considered as a structure and perform MSSE for finding the scale of the square residuals.

(viii) Having obtained the scale, one proceeds by calculating the square residual for *every pixel* in the image and

https://doi.org/10.1017/S0263574799001812 Published online by Cambridge University Press

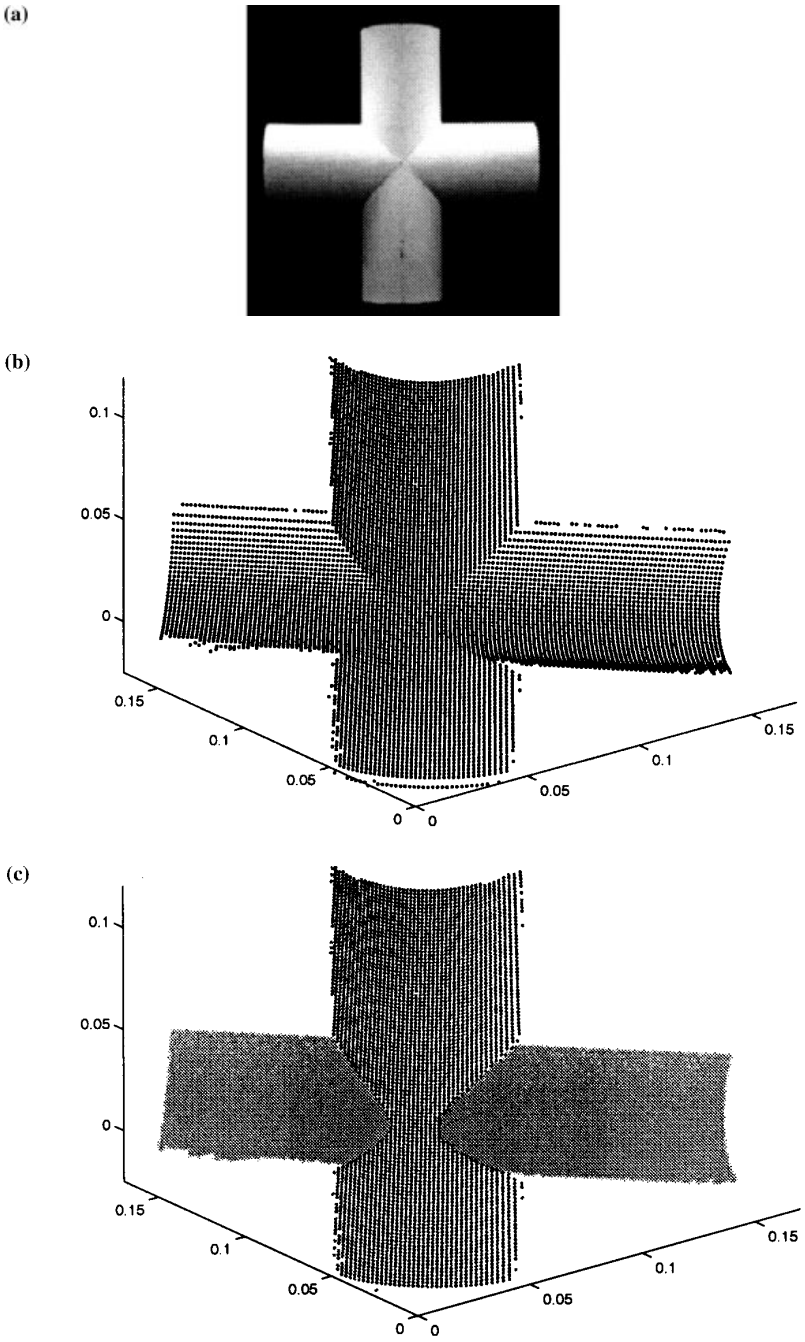


Fig. 8. (a) The scene of the fourth experiment. (b) The original range image of the fourth experiment. (c) The segmented range image of the fourth experiment.

comparing it with the scaled residual. At this step, the spatial continuity is enforced by using a median filter with the support of 9 pixels (depends on the size of object considered) wide square window centered at the considered pixel. *Thus, instead of comparing the pixel square residual with the scaled square residual, the median of square residuals over the window is compared to the scaled square residual.* If the median of residuals is smaller than 2.5 times of the scaled residual, the central pixel in the window is set to be part of a new group. The algorithm proceeds by considering all the previously unassigned pixels. Steps v–viii will be repeated sequentially until the population of remaining pixels goes below one K-th of the original population.

5. EXPERIMENTAL RESULTS

As explained before, the model of motion is based on the motion of a planar surface having rigid motion. Therefore, the algorithm segments moving objects into planar patches. Depending on the physical geometry of the objects, this may or may not be a good representation of their 3-D shapes. However, as can be seen from the results, this simple representation yields an impressive spatial segmentation of their 2-D appearance on the image plane particularly

https://doi.org/10.1017/S0263574799001812 Published online by Cambridge University Press

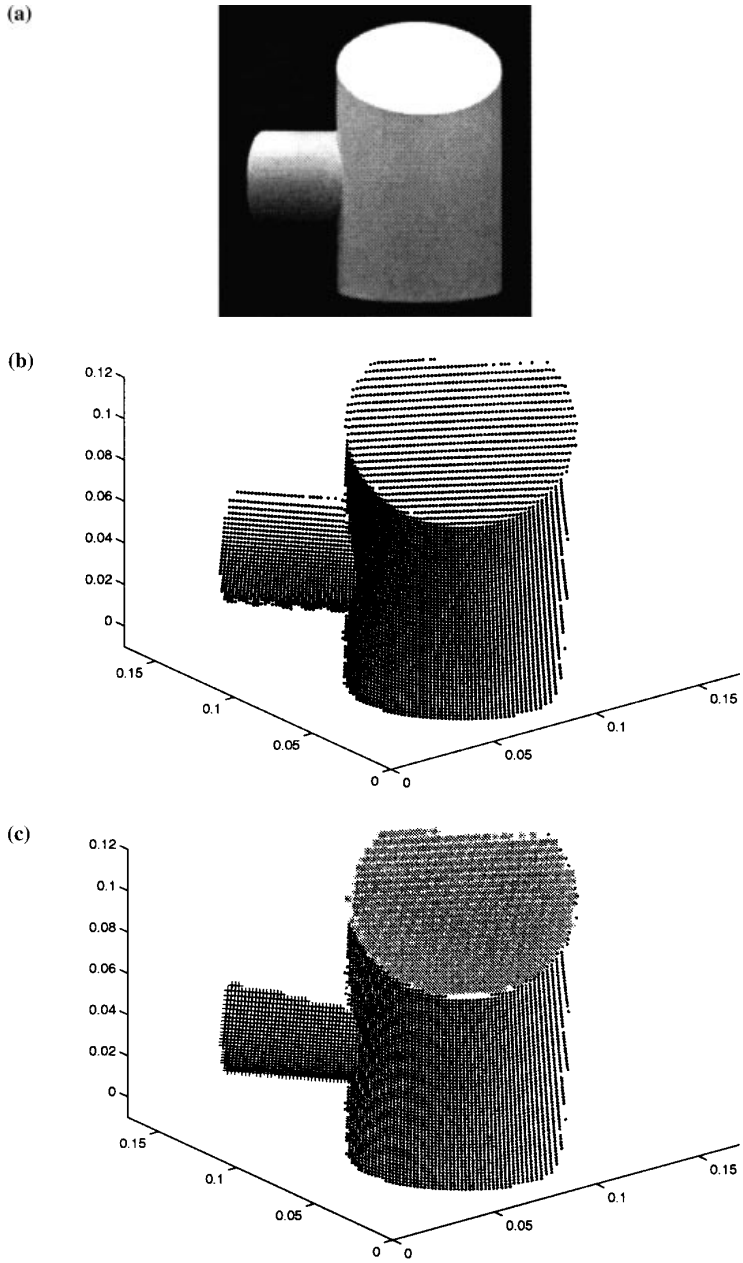


Fig. 9. (a) The scene of the fifth experiment. (b) The original range image of the fifth experiment. (c) The segmented range image of the fifth experiment.

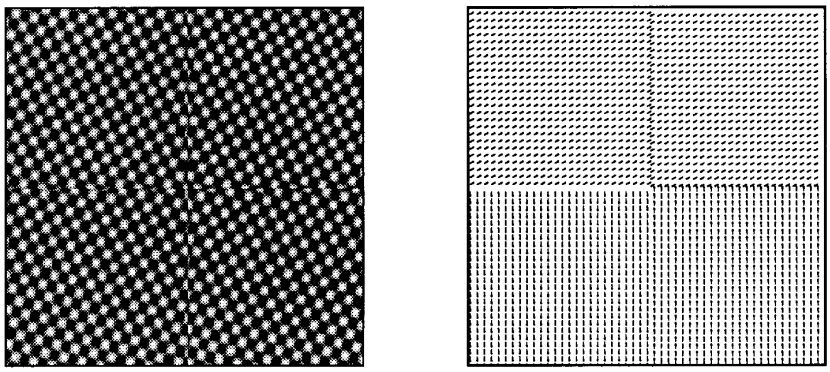


Fig. 10. A snap shot of mosaic sinusoid sequence (left) and its optic flow field (right).

https://doi.org/10.1017/S0263574799001812 Published online by Cambridge University Press

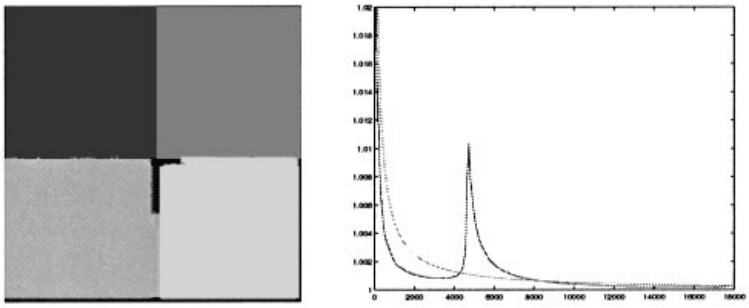


Fig. 11. Segmentation results (left) and plot of quantities contained in inequality 2.3 for explicitly calculated derivatives (corrupted with Gaussian noise).

in man-made environments. To visualize the segmentation results, a gray-scale image with the same size as of the original image is created. In these images, the black regions represent the non-segmented section. White represents the area with little spatial information and the next darker shade represents the stationary area. Every other gray-scale shade represents a planar patch with distinct motion.

5.1 Synthetic image sequence

To demonstrate the performance of the proposed algorithm, a mosaic image is created using the same texture as the "Sinusoid1" image sequence[34] while it has different motion in every quarter of the images. Figures 10 shows a snap shot of the sequence and its optic flow field.

The results for this sequence are presented for two cases. First, since the function describing the texture is known, the derivatives were exactly calculated (i.e. *symbolically*) and the values of the temporal derivatives were corrupted with Gaussian noise (with different but known standard deviations for every quarter). The final segmentation result and the scale ratio plot for the first segment are shown in figure 11. Table I compares the actual and computed scale for every segment of this sequence.

Table I. Comparison between true and computed standard deviation of noise.

	True	Computed
Quarter 1 (NSW)	1.0	1.086
Quarter 2 (NE)	0.75	0.822
Quarter 3 (SW)	0.5	0.555
Quarter 4 (SE)	0.25	0.306

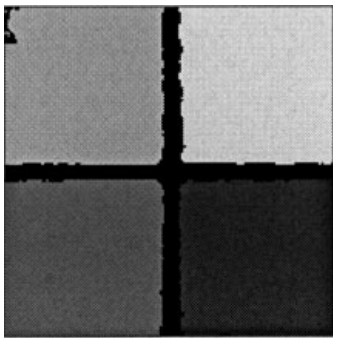


Fig. 12. Segmentation results for the mosaic sinusoidal image sequence (numerically estimated derivatives).

For the second case, the motion segmentation is applied to the sinusoidal sequence directly. This means that the derivatives are calculated from the texture pattern following the step (i) of section 4 (i.e. *numerically*). As can be seen in figure 12, the derivatives along the motion boundaries cannot be reliably estimated and thus appear as outliers to the main moving sections.

5.2 Real image sequences

To demonstrate the performance of the proposed algorithm for real world data, the calendar and Rubik cube sequences are chosen.

In the calendar sequence, a toy train is moving toward the left pushing a ball in front along its tracks. A calendar on the wall is also moving upward while the camera translates toward the left. Figure 13 shows a snap shot of this sequence and its segmented results.

As can be seen, the scene is segmented into three main areas based on their distinct motions. Finally, we present the results of our approach applied to the Rubik cube sequence (created by Richard Szeliski). The *Rubik* cube sequence is a particularly good example mainly because the projected motion of its planar surfaces is well described by the

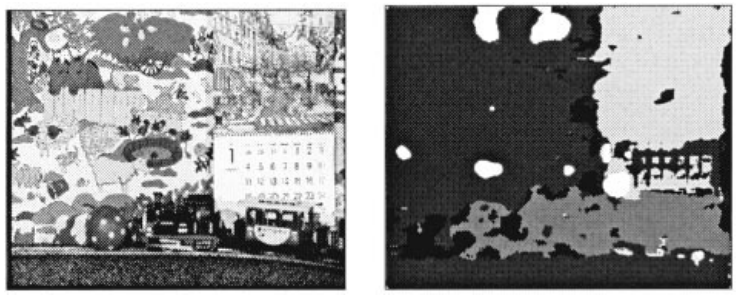


Fig. 13. A sample image and the segmentation results for Calendar image sequence (brownish hue represents stationary segments).

https://doi.org/10.1017/S0263574799001812 Published online by Cambridge University Press

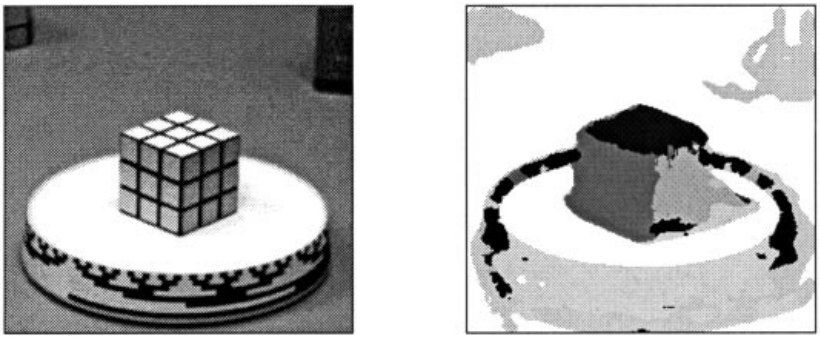


Fig. 14. A sample image and the segmentation results for Rubik image sequence (brownish hue represents stationary segments).

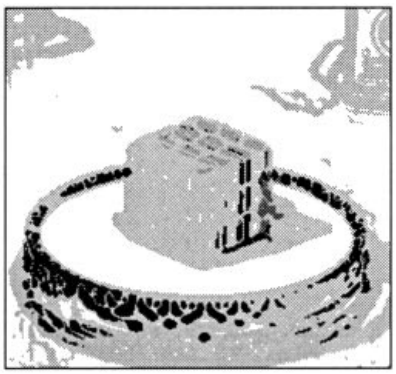


Fig. 15. The segmentation results for Rubik image sequence without enforcing the spatial continuity.

proposed model (Eqs 4.1 and 4.2). In this image sequence, a *Rubik* cube is placed on a turntable and this turntable is rotating in front of a stationary background. The velocities on the turntable are around 1.2 to 1.4 pixels/frame and on the cube itself are around 0.2 to 0.5 pixels/frame.

A snap shot and the segmentation results for this sequence are shown in figure 14. It can be seen from these pictures that the segmentation has been able to correctly identify different moving planes on the cube. However, in case of the turning table face, which is circular, the fit represents a break-down in the validity of the motion model (see section 4 step iii).

To demonstrate the effect of enforcing spatial continuity, a set of results for the above sequence is also created without using the median filter of step viii. This set of results is shown in figure 15. As can be seen, the *median filter* plays an important role in preserving the fidelity of the segmentation results.

6. CONCLUSION

A new algorithm built upon MSSE for data segmentation is proposed in this paper. The segmentation algorithm uses an LKS fitting approach with a novel method for calculating the scale of residuals without using the rank ordering statistics. As explained in the text, the rank order statistic taken from a population containing multiple distributions cannot be used to estimate scale as if it were the same rank order statistic of a single distribution. We thus avoid some of the computational and theoretical complexities of similar approaches (MUSE and ALKS).

The proposed method is also used to develop a robust range and motion segmentation algorithms. In adapting MSSE to these real world problems, several modifications to improve efficiency were made. The segmentation algorithms are tested on synthetic and real data and has shown to be effective.

Acknowledgements

The authors wish to acknowledge the assistance of the 3D shape measurement group at Monash University and in particular A/Prof. Kim Ng and Mr. John Minack in obtaining the range images used for experiments in this work.

References

1. L. Torres and M. Kunt, *Video Coding (The Second Generation Approach)* Kluwer Academic Publishers, The Netherlands, 1996.
2. F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw and W.A. Stahel, *Robot Statistics: The Approach Based On Influence Functions* (Wiley, New York, 1986).
3. P.J. Rousseeuw and A.M. Leroy, *Robust Regression and Outlier Detection* Wiley, New York, 1987).
4. C. Stewart, "Bias in Robust Estimation Caused by Discontinuities and Multiple Structure," *IEEE TPAMI* **19**, 818–883 (1997).
5. P.J. Rousseeuw, "Least median of squares" *J. Amer. Statistical Association* **85**, 115–119 (1984).
6. P. Meer, D. Mintz, A. Rosenfeld and D.Y. Kim, "Robust regression methods for computer vision: A review," *Int. J. Computer Vision* **6**, 59–70 (1991).
7. K.-M. Lee, P. Meer and R.-H. Park, "Robust Adaptive Segmentation of Range Images," *IEEE Trans. Pattern Anal. Machine Intell.* 200–205 (1998).
8. J.V. Miller and C.V. Stewart, "Muse: Robust surface fitting using unbiased scale estimates," *Proceeding of CVPR '96* (1996) pp. 300–305.
9. A. Bab-Hadiashar and D. Suter, "Robust Motion Segmentation Using Rank Ordering Estimators," *Proceedings ACCV '98*, Hong Kong (1998) pp. 599–606.
10. A. Hoover, G. Jean-Baptiste, X. Jiang, P. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon and R. Fisher, "An Experimental Comparison of Range Image Segmentation Algorithms," *IEEE TPAMI* **18**, 673–689 (1996).
11. X. Yu, T.D. Bui and A. Krzyzak, "Robust Estimation for Range Image Segmentation and Reconstruction," *TPAMI* **19**, 139–143 (1994).
12. F. Dufaux and F. Moscheni, "Segmentation-Based Motion Estimation for Second Generation Video Coding Techniques," In: *Video Coding (The Second Generation Approach* (L. Torres and K. Kunt, Eds.) (The Netherlands, Kluwer Academic Press, The Netherlands, 1996) Chapter 6.
13. G. Adiv, "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving

https://doi.org/10.1017/S0263574799001812 Published online by Cambridge University Press

Objects," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **7**, 384–401 (1985).
14. J. Wang and E. Adelson, "Repreenting Moving Images with Layers," *IEEE TPAMI* **3**, 625–638 (1994).
15. H. Kollnig and H.H. Nagel, "Matching Object Models to Segments from an Optical Flow Field," *Proc. ECCV'96* (1996) pp. 388–399.
16. Y. Huang, K. Palaniappan, X. Zhuang and J.E. Cavanaugh, "Optic Flow Field Segmentation and Motion Estimation Using a Robust Genetic Partitioning Algorithm," *IEEE TPAMI* **17**, 1177–1190 (1995).
17. J. Weber and J. Malik, "Rigid Body Segmentation and Shape Description from Optical Flow Under Weak Perspective," *IEEE TPAMI* **19** 130–143 (1997).
18. A. Bab-Hadiashar and D. Suter, "Motion Segmentation Using Robust Motion Estimation," *Proc. APRS Image Segmentation Workshop*, Sydney, Australia (1996) pp. 7–11.
19. A. Bab-Hadiashar and D. Suter, "Motion Based Segmentation Using Robust Statistics," *Proc. IAIF'97*, Adelaide, Australia (1997) pp. 271–280.
20. D.W. Murray and B. Buxton, "Scene Segmentation from Visual Motion Using Global Optimization," *IEEE TPAMI* **9**, 220–228 (1987).
21. P. Bouthemy and E. Francois, "Motion Segmentation and Qualitative Dynamic Scene Analysis from an Image Sequence," *Int. J. Computer Vision* **10**, 157–182 (1993).
22. J.M. Odobez and P. Bouthemy, "Robust Multiresolution Estimation of Parametric Motion Models," *J. Visual Communication and Image Representations* **6**, 348–365 (1995).
23. M. Hötter and R. Thoma, "Image Segmentation Based on Object Oriented Mapping Parameter Estimation," *Signal Processing* **15**, 315–334 (1988).
24. S. Peleg and H. Rom, "Motion based segmentation," *Proc. IEEE International Conference on Pattern Recognition*, Atlantic City, NJ (1990) pp. 109–113.
25. W.B. Thompson, "Combining Motion and Contrast for Segmentation," *IEEE TPAMI* **2**, 543–549 (1980).
26. M. Black, "Combining Intensity and Motion for Incremental Segmentation and Tracking over Long Image Sequences," *Proc. Second European Conference on Computer Vision* (1992) pp. 485–493.
27. P. Schroeter and S. Ayer, "Multi-frame Based Segmentation of Moving Objects Combining Luminance and Motion," *Proc. VII European Signal Processing Conference EUSIPCO'94*, Edinburgh, UK (1994) pp. 22–25.
28. F. Duvaux, F. Moscheni and A. Lippman, "Spatio-temporal Segmentation Based on Motion and Static Segmentation," *Proc. IEEE International Conference on Image Processing ICIP'95*, Washington DC (1995) pp. 306–309.
29. M. Gelgon and P. Bouthemy, "A Region Level Graph Labeling Approach to Motion Based Segmentation," *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR'97*, Puerto Rico (1997). pp. 514–519.
30. P.H.S. Torr and D.W. Murray, "Stochastic Motion Clustering," *Proc. Third European Conference on Computer Vision* (1994) pp. 328–337.
31. H.C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proc. R. Soc. Lond. B.* **208**, 385–397 (1980).
32. B.K. Horn and B.G. Schunck, "Determining Optic Flow," *Artificial Intelligence* **17**, 185–203 (1981).
33. A. Bab-Hadiashar and D. Suter, "Robust Optic Flow Comptuation," *Int. J. Computer Vision* **29**, 59–77 (1998).
34. J.L. Barron, D.J. Fleet and S.S. Beauchemin, "Systems and experimental performance of optical flow techniques," *Int. J. Computer Vision* **12**, 43–77 (1994).

https://doi.org/10.1017/S0263574799001812 Published online by Cambridge University Press