A Robust Region-based Global Camera Estimation Method for Video Sequences

Xuesong Le
School of ICT
Griffith University, Parklands Drive, Southport QLD 4222
Australia

Ruben Gonzalez
NICTA, School of ICT
Griffith University, Parklands Drive, Southport QLD 4222
Australia

***Abstract*—Global motion estimation (GME) plays an important role in video object segmentation. This paper presents a computationally efficient three stage affine GME algorithm, using radius-based Fourier Descriptors from histogram-based image segmented regions. Then a variance-cut KD tree is used for initial matching between FDs. and an efficient two-step outlier method is applied to remove incorrect outliers. Experiments with different video sequences are used to demonstrate the performance of the proposed approach.**

Keywords—affine motion estimation; outliers removal; Fourier Descriptors

I. INTRODUCTION

Global motion estimation (GME) is an important task in video object segmentation., and can be either pixel-based, block-based, or feature based. Pixel-based approaches [1] suffer from heavy computation requirements. Block-based approaches [2], requires detection and removal of noise in the resulting motion vector fields. This paper is concerned with low complexity feature based approaches.

Most state of the art feature-based matching can be divided into two steps; finding features in an input frame, and computing local descriptors for each one; and then matching those features between the current and previous frame. At the feature detection step, compared to numerous feature point based algorithms surveyed in [3] [4], the use of regional descriptors as a step in GME for video sequences has been little explored due to the difficulty in obtaining segmentation consistency across frames. In contrast, histogram based segmentation methods provide high correlation of regions across frames in video segmentation which typically have limited differences between frames. The other advantage is that histogram based segmentation methods requires only one pass through the pixels in the image to perform segmentation. Appropriate region descriptors need to be used selected to facilitate matching. The matching itself can be very computationally intensive, so a variance-cut KD-tree and a search algorithm are proposed in this paper together with various optimisations such as pre-filtering to remove outliers. Finally RANSAC is used to select the most likely matches. Compared schemes, such as the SIFT [5], the number of required matches are far less, reducing the computational requirements because of the lower probability of incorrect matches.

II. FEATURE DETECTION

A. Region Based Detection Algorithm

Histogram-based image segmentation comprises three stages: finding the valleys of the histogram; applying thresholds to images based on the valleys; finally performing a connected component analysis on threshold images to extract regions where pixels have been assigned to the same value. For this work, we have used a divisive clustering approach [6] to search for the optimal multi-level thresholds. It starts with a single class and recursively splits the class into two classes, based on the gray level, i, in which the between-class variance, σ_b^2, is maximized. The process iterates with the class that has the largest total variance, σ_k^2, among all classes until the predetermined number of classes, M is reached, M classes: c_1 for $[0,\dots,t_1]$, c_2 for $[t_1+1,\dots,t_2]$,..., c_i for $[t_{i-1}+1,\dots,t_i]$,..., and c_M for $[t_{M-1}+1,\dots,255]$. The total variance, σ_k^2, and between-class variance, σ_b^2 for class, c_k, is defined as following:

$$\sigma_b^2 = \omega_1\omega_1(\mu_1-\mu_2)^2 \tag{1}$$

$$\omega_k = \sum_{i\in c_k} p_i \; \mu_k = \sum_{i\in c_k} ip_i/\,\omega_k \sigma_k^2 = \sum_{i\in c_k}(i-\mu_k)^2 p_i/\,\omega_{k,}(2)$$

Where p_i is probability at intensity level i and $k = 1,2$, ω_k is cumulative probabilities, and μ_k is mean, for each class, c_k.

The optimal threshold t in any class c_k for intensity level interval $[t_{i-1}+1,\dots,t_i]$ is the intensity level where between-class variance, σ_b^2 is maximized.

$$t = \mathrm{Arg\,Max}\{\sigma_b^2(t)\},\; t_{i-1}+1 \le t < t_i \tag{3}$$

After selection of optimal thresholds, every gray level, i, in histogram is labelled as one of the following:

$$L(i) = k, k = 0,1,\dots M, i \in c_k \tag{4}$$

Where L(i) represents the k^{th} label value that gray level, i, is assigned to, given i is within the class c_k . This labelling is a lookup operation based on gray level. Finally a connected component analysis is performed on both labelled images to extract regions where pixels have been assigned to the same value.

B. Regional Descriptor Extraction

Shape is an important descriptor in feature matching. Among various shape representations and matching methods reviewed in [3], Fourier descriptor (FD) based methods achieve both good representation and transformation robustness. In [7], centroid-distance shape signatures were found to outperform six other different signatures as the basis for FDs. In this paper, we propose an improved centroid-distance shape signature for initial matching. In combination with the FDs, the regional mean, and radius-area based compactness are also used.

The following definitions related to shape descriptors are used in this paper.

Computing Centroid: Given n points $\{(x_0, y_0), (x_1, y_1), \ldots, (x_{n-1}, y_{n-1})\}$ in a region, R, then the position vector of the centre of gravity, $\left(\overline{C_x}, \overline{C_y}\right)$, is given by:

$$\overline{C_x} = \sum_{i=0}^{n-1} x_i \, , \overline{C_y} = \sum_{i=0}^{n-1} y_i \qquad (5)$$

Centroid Distance Function: The centroid distance function expresses the distances of the boundary points $(x_i, y_i)$from the centroid $\left(\overline{C_x}, \overline{C_y}\right)$ of a shape. Each distance, $f(k)$, is called radius and is given by the following formula:

$$f(k) = \sqrt{(x_i - \overline{C_x})^2 + \left(y_i - \overline{C_y}\right)^2} \qquad (6)$$

The radius of the region f(k), at any angle is calculated as the distance between the shape centroid and the region boundary. This is sampled in steps of 5.625 degrees.

The Fourier coefficients F_u must be normalized to make them invariant to the rotation and scaling of shapes. The FD is intrinsically translation invariant [8]. Shape rotation is reflected in the phase information of F_u, and taking the magnitude of each Fourier coefficient, $|F_u|$ makes FDs rotation invariant. $|F_0|$ reflects the energy of the shape radii, and dividing all Fourier coefficients by $|F_0|$ could make FDs scaling-invariant. This results in a translation, rotation and scale invariant feature vector:

$$V = \left[\frac{|F_1|}{|F_0|}, \frac{|F_2|}{|F_0|}, \text{⍰}, \frac{|F_{N-1}|}{|F_0|}\right]^T \qquad (7)$$

Zhang and Lu [9] have found that 10 FD features are sufficient to represent a shape. In our experiment, 10 FD features are used. In addition we have used the mean, and compactness of regions to improve matching accuracy.

Mean of Gray Levels: The mean of gray levels in a region where $p(x_i, y_i)$ is the gray level of a region pixel.is defined as:

$$M = \sum_{i=0}^{n-1} p(x_i, y_i) \qquad (8)$$

Radius-Area Based Region Compactness: We propose a radius-area based compactness descriptor instead of perimeter-area based compactness . Perimeter-area based compactness is sensitive to small deviations in. Radius-area based compactness, on the other hand, is more robust to noise to its smoothing operation.

Proposed radius-based compactness averages 64 radii

$$\bar{f} = 1/64 \sum_{k=0}^{63} f(k) \qquad (9)$$

Then the radius based compactness of a region, RC where, A, is area of a region, defined by:$A = \sum_{i=0}^{n-1} 1$ is defined as:

$$RC = \frac{(\bar{f})^2}{A} \qquad (10)$$

The final feature vectors extracted from each segmented frame are 12-dimensional, respectively denoted by $A = \{a_0, \ldots, a_i, \ldots a_{m-1}\}$, and $B = \{b_0, \ldots, b_i, \ldots b_{n-1}\}$.

C. Initial Matching of Regional Descriptors

Given two lists of features descriptors, preliminary matches are initially established between them. We propose a variance-cut KD tree method to cluster feature vectors in the similarity space. This is a variation of Median-cut KD tree [10]. Median-cut KD tree repeatedly subdivides a data space into smaller and smaller rectangular spaces based on the median value of its dimension, in which the data exhibits the greatest variance. The median-cut approach does not behave optimally when a space contains two unequal feature vector groups as it tends to cut through the cluster with more feature vectors, so the feature vectors after the split are not located at the right places [11]. In contrast, a variance-cut method in a space attempts to minimize the total variance of sub-spaces so that the similar feature vectors are grouped as much as possible.

Construction of the Variance-cut KD Tree: Given a set of N points, $X = \{x_0, x_1, \ldots, x_{n-1}\}$ in k dimensional feature space E^k and each x_i is a k-dimensional vector, it can be represented as an n by k matrix,

$$\begin{bmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,k-1} \\ x_{1,0} & x_{1,0} & \cdots & x_{1,k-1} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n-1,0} & x_{n-1,j} & \ldots & x_{n-1,k-1} \end{bmatrix} \qquad (11)$$

a variance-cut method in a space attempts to minimize the total variance of sub-spaces so that the similar feature vectors are grouped as much as possible.

$$Var_{max} = \text{Arg Max}\left(\text{Max}\left(x_{j,i}\right) - \text{Min}\left(x_{j,i}\right)\right),$$

$$i \in [0, k-1], j \in [0, n-1] \qquad (12)$$

Similar to the (1 optimal threshold selection in equation 3, a cut is made at) the position, t, where between-space variance is maximized.

$$t = Arg\ Max\{\sigma_b^2(t)\},\ t_{i-1} + 1 \le t < t_i \qquad (13)$$

A node in k-d tree contains following elements:

- A pointer is either null or points to a space E.
- Two pointers which are either null or point to another node in the k-d tree.
- Low bounds of data at each dimension.
- High bounds of data at each dimension.

The construction of variance-cut KD tree can be described as following:

1 *Start adding space E_0 which includes all the feature vectors $E_0 = \{x_0, x_1, \ldots, x_{n-1}\}$ to list D, create node, n_0, and insert node n_0 to k-d tree as root node.*

2 *Search each space E_i in list D and find the space E_{max}, that has the greatest variance, Var_{max} at l^{th} dimension.*

3 *Exhaustively search for the optimal partition, t, in space E_{max} that maximize the between-space variance $\sigma_b^2(t)$, at l^{th} dimension as following:*

   a. *Sort all feature vectors X in space E in ascending order of value at l^{th} dimension*

   b. *Step through all possible partitions $t = 0, 1, \ldots, m-1$, and find the partition where the between-space partition, $\sigma_b^2(t)$, is maximized.*

   c. *An internal node is created to store dimension index, l, and cut point value $cut(t)$.*

4 *Split space E_0 into two subspaces E_u and E_v, by comparing each feature vector element at l^{th} dimension with cut point value cut (t) , $E_u = \{x_i | x_{i,l} \le cut(t)\}$, $E_v = \{x_i | x_{i,l} > cut(t)\}$*

   a. *Create a node n_1 including E_u, insert n_1 to left branch of k-d tree.*

   b. *Create a node n_2 including E_v, insert n_2 to right branch of k-d tree*

   c. *Calculate low bounds and upper bounds of each subspace.*

5 *Remove E_0 from the list D and add two split spaces E_u and E_v to the list D. increment number of spaces m by one.*

6 *If any of the following criteria is satisfied, stop iteration.*

$$\mathrm{Var}_{\max} < T_{\min\_var} \vee \mathrm{sizeOf}(x') < T_{\min\_sz} \quad (14)$$

where $sizeOf(x')$ is the number of feature vectors in current space and $T_{\min\_sz}$ is the minimum required number of feature vectors in any space.

7 *Else, go back to step 2.*

This creates a balanced tree with depth $d = |\log_2 N|$ at complexity $O(kN\log N)$.

Improved Search Algorithm: Search methods, such as Nearest Neighbor Search (NNS) in KD-tree [12] move down the tree recursively and the search direction at each level depends on whether the feature vector is less than or greater than the cut point stored at current node in the split dimension. Given a query feature vector, a descent down the KD tree requires $\log_2 N$ comparisons and leads to its nearest match. It then backtracks the near nodes in order of their distance to the query feature vector. The search terminates when there are no more nodes within the distance defined by the best match found so far or if the m nearest neighbors have been found.

The performance of the search relies on three things. The first is how the tree subdivides the space. Improved partitioning leads to searching fewer nodes to find good matches, such as proposed variance-cut.

The second is how to make branching decisions. Comparing a k-dimensional feature vector with key value in a single split dimension may leads to the incorrect branch due to noise. K-comparison at non-leaf node can improve the probability of success with the following criteria:

$$\left(\sum_{i=0}^{k-1} p(q_i)\right) > Thresh_p \; p(q_i) = \begin{cases} 1/k & if\ l_i \le q_i \le u_i \\ 0 & else \end{cases} \quad (15)$$

Where $p(x_i)$ is the assigned probability if query vector, q_i is within the lower range, l_i, and upper range, u_i , of a child node at i^{th} dimension. $Thresh_p$, the minimum probability of branching to the child node, is 0.5 in our experiment.

The last one is the distance metric used to sort found matches in order of their distance to the query feature vector. Since the lower FD terms determine the global shape and higher FD terms account for fine detail, the weight attached to each dimension in distance metric should be treated differently. The Weighted Euclidean Distance in our experiment is defined as:

$$d_{x,q} = \left(\sum_{i=0}^{k-1} w_i (x_i - q_i)^2\right)^{1/2} \quad (16)$$

Where both x and q are k-dimensional vector, w_i is the value of the weight attached to the i^{th} measure. w_i can be derived in the following:

$$w_i = \left(\frac{1-r^k}{1-r}\right) r^i \quad (17)$$

Where $0 < w_i < 1$, $\sum_{i=0}^{k-1} w_i = 1$ $w_i = w_{i-1} r$, r is the diminishing rate between weight w_i and w_{i-1}.

The search in variance-cut KD tree for a query feature vector, q_i, can be described as following:

1 *(Initialize) Let n_0 be the current node, n, to be inspected.*

2 *Let n_l be the left child node of n and let n_r be the right child node of n;*

3 *If n is a leaf node, then add all feature vectors in current space to a list D.*

4 *Otherwise*

- a. *If the probability of the query feature vector falls into left branch meets the criteria defined in equation 15, then go the left branch.*
- b. *If the probability of the query feature vector falls into right branch meets the criteria defined in equation 15, then go the right branch.*
- c. *Else return.*

5 *Sort all the feature vectors in list D in ascending order of their Weighted Euclidean distance to the query feature vector.*

The output of this step is a list of one-to-many matches, T= $\{(b_0, A_0) \ldots, (b_i, A_i), \ldots (b_{m-1}, A_{m-1})\}$, where each (b_i, A_i) represents a single feature vector, b_i , from the second segmented frame and its candidate matches, A_i from first segmented frames.

III. Outlier Removal

In [5], the best candidate match for each feature vector is found by identifying its nearest neighbor in its candidate set. The nearest neighbor is defined as the feature vector with minimum Euclidean distance. However, the nearest neighbor query for FD feature vector is not the best candidate because it arises from background clutter or similar shapes. Therefore, it would be useful to have a way to discard candidate feature vectors that do not have any good match to the query feature vector. Robust statistics methods are usually used to refine global feature matching results. Among these methods, RANSAC [13] is most popular. RANSAC is a non-deterministic algorithm for the estimation of a mathematical model from observed data that contains outliers. It is essentially composed of two steps that are repeated iteratively.

Hypothesize: A sample of size m among the N data points is randomly selected. The model parameters are computed from this sample. m is the smallest sufficient cardinality to determine the model parameters.

Test: The hypothesis is verified against the rest of the data by counting the points consistent with the estimated model parameterization.

These two phases are repeated until the probability of finding a better solution falls below a pre-selected threshold t.

A disadvantage of RANSAC is that there is no upper bound on the time it takes to compute these parameters. The number of necessary iterations increases very fast in proportion to the size of the feature sets to be matched, the outlier ratio and the model complexity, m. RANSAC is impractical for many real time applications. Reducing the number of features to be matched and removing as many obvious outliers before RANSAC is applied can reduce computation time.

Many well-known probability-dependent RASAC methods, that try to reduce processing time, such as MLESAC, PROSAC, or Guided Sampling for MLESAC [14], perform poorly when the percent of inliers falls much below 50%, Exhaustive Ransac algorithm tends to perform better but suffers high number of iterations. We present a computationally efficient two stage filtering algorithm to solve this problem.

A. Ransac

The proposed matching approach is based on KD-tree algorithm, targeting the 12 dimension descriptors of each FD feature. Inevitably, false matches occur. Applying Random Sample Consensus (RANSAC) with a 2D affine transformation model filters out these false matches (outliers).

2D affine transformation can be modeled as composite matrix operation, which maps a point p = (x_i, y_i) T to a point q = (u_i, v_i) T as follows:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta * s & -\sin\theta * s & t_x \\ \sin\theta * s & \cos\theta * s & t_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{18}$$

The affine matrix can be simplified as $M = \begin{bmatrix} a & -b & c \\ b & a & d \\ 0 & 0 & 1 \end{bmatrix}$ and the parameters in the simplified matrix can be solved by the pseudo-inverse solution.

Given a set of points and its correspondences in the form U = WA.

$$\begin{bmatrix} u_1 \\ \ldots \\ u_n \\ v_1 \\ \ldots \\ v_n \end{bmatrix} = \begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ & \ldots & & \\ x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ & \ldots & & \\ y_n & x_n & 0 & 1 \end{bmatrix} \tag{19}$$

The pseudo-inverse solution $A = (W^T W)^{-1} W^T U$ is computed to solve for the four affine coefficients.

U and M are then used for calculating the re-projection error of the remaining pairs of correspondence. Re-projection error is defined as the Euclidean distance between the transformed point of point p in the previous frame according to the model M and the original correspondence of q in the current frame:

$$e = \left\| \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} - M \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \right\| \tag{20}$$

The re-projection error threshold T_{thres} is determined such that among all the initial matches, only those pairs with $e < T_{thres}$ will be regarded as inliers and kept. The process is repeated for different subsets, resulting in different transformations, with the number of inliers recorded for each transformation. The transformation supported by the highest number of inliers is selected to be the winning transformation. In the final step, all inliers from the winning transformation are used to re-estimate the transformation. The resulting transformation will be considered as the final relative transformation between these two consecutive frames.

B. Prefiltering Outliers

To reduce the number of iterations during exhaustive Ransac matching, constraints may be applied to the camera transformation. Due to the high frame rates in video, it is unlikely to have large translation and scaling between frames within the 0.04 second interval between frames since the motion would have to be extremely fast resulting in motion blur if otherwise.

Taking these factors into consideration, the filtered candidate matches, A_i', for each b_i in one-many-matches (b_i, A_i) can be defined as:

$$A_i' = \left\{ a_i \in A_i \middle| \begin{array}{c} \mathrm{Diff_{tx}}(a_i, b_i) < T_{tx\_thres} \wedge \\ \mathrm{Diff_{ty}}(a_i, b_i) < T_{ty\_thres} \wedge \\ \mathrm{Diff_{A}}(a_i, b_i) < T_{A\_thres} \wedge \\ \mathrm{Diff_{I}}(a_i, b_i) < T_{i\_thres} \wedge \\ \mathrm{Diff_{c}}(a_i, b_i) < T_{c\_thres} \end{array} \right\} \quad (21)$$

Where $\mathrm{Diff_{tx}}(a_i, b_i)$, and $\mathrm{Diff_{ty}}(a_i, b_i)$ are the translation difference between two gravity centrods in both X and Y directions, $\mathrm{Diff_{A}}(a_i, b_i)$ is the difference in region size, $\mathrm{Diff_{I}}(a_i, b_i)$ is the intensity difference, and $\mathrm{Diff_{c}}(a_i, b_i)$ is difference in compactness between any match (a_i, b_i).

The output of this step is an updated list of one-to-many matches, $T' = \{(b_0, A_0') \dots, (b_i, A_i'), \dots (b_{m-1}, A_{m-1}')\}$, where each (b_i, A_i') represents a single feature vector, b_i, from the second segmented frame and its filtered candidate matches, A_i' from first segmented frames.

C. Using matching similarity to improve RANSAC

To further reduce the number of iterations in the following exhaustive Ransac, a similarity function defined in equation 16 is evaluated over a number of correspondences, and subsequently thresholded to obtain a set of tentative correspondences. Based on the assumption that points with high similarity are more likely to be inliers than points with low similarity, it may be possible to generate better hypotheses by sampling from a reduced set of points with high similarity scores.

Consider a sequence of samples of 2, drawn by exhaustive RANSAC from the set of all N correspondences, the number of iterations in exhaustive Ransac is $\binom{N}{2}$. Let U_n denote a subset of samples containing n points with the highest quality and if the samples are only selected from, U_n then the number of iterations can be reduced to $\binom{n}{2}$, saving $1 - \binom{n}{2} / \binom{N}{2}$ computation.

The selection of n is estimated by the experiment observations. After pre-filtering, we build up a list of correspondences from 1-M matches first. Then the correspondences are sorted based on the similarity function. As shown in Figure 1, at least 50% of inliers can be found at the top 30% of the sorted N correspondences.

In our experiments, the number of iterations required can be reduced by 66% compared with simple RANSAC sampling, with better performance than other sampling algorithm that are based on estimated prior probabilities.

The result list of matches between the centroids of each region in two video frames permits the final camera transformation from equation 19.

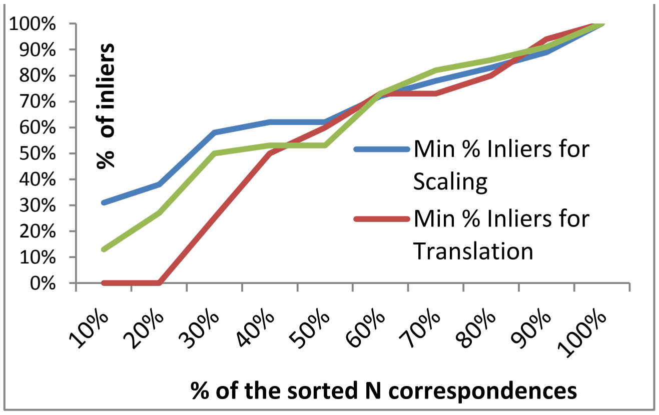


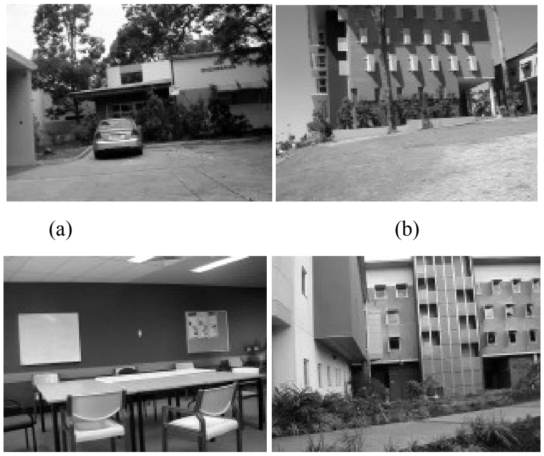
Figure 1 Percent of Inliers

IV. Experiment Results

In our experiments, four video sequences in Figure 2 are used. Each type has been further tested under varying scaling, rotation, and translation camera transformations. The algorithm is implemented in Java with the Apache Commons Mathematics Library [15].

A. Ground Truth Match Simulation

We extracted the gravity centre points of each image region from two segmented images using histogram based segmentation method. Then we used exhaustive RANSAC method to simulate the ground truth matches between region centroids. The re-projection error threshold T_{thres} defined in equation 20, is set to 2.



(a) (b)

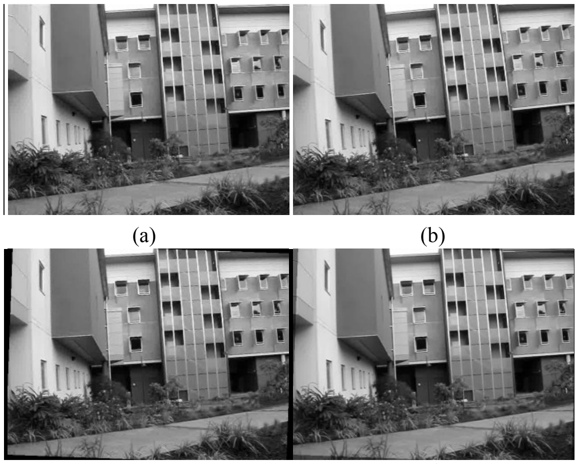
(c) (d)

Figure 2 Testing Videos: (a) car-park (b) entry (c) lounge (d) court

B. *Evaluation of Matching Performance*

To test transformation robustness over a range of images, we evaluated the performance of matching over fixed transformation intervals. We use the first frame as reference frame and evaluate matching performance between the chosen reference frame and successive frames and determine the accuracy of matches in each pair.

Testing images for rotation tests were selected given 4 degree interval. The maximum rotation between the first frame and the last frame in test sequence is 12 degree which is reasonable in video sequence. Figure 3 shows an image registration pair, resampled image, and mosaiced image for "court" sequence under rotation.

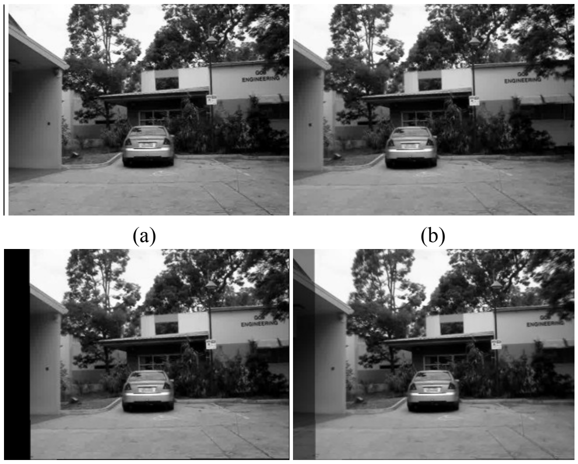


(a) (b)

(c) (d)

Figure 3 (a-b) original "court" pair under rotation, (c) resampled image given image pair (a) and (b), (d) mosaiced images given (a) and resampled image (c)

The horizontal translation in tests was measured as percentage of amount of translation over the image width. Images in translation tests were selected every 4% image width interval. The maximum translation range in the testing sequence was 15%. Figure 4 shows an image registration pair, resampled image, and mosaiced image for "carpark" sequence under translation.



(a) (b)

(c) (d)

Figure 4 (a-b) original "carpark" pair under translation, (c) resampled image given image pair (a) and (b), (d) mosaiced images given (a) and resampled image (c)

Incrementing 0.1 scaling ratio each time, we selected test images with zooming ratio up to 1.4 that covers the range of expected zooming ratios in typical video sequences. Table 1 shows that proposed method was able to recover at least 86% of ground truth pairs in scaling case, 80% of ground truth pairs in translation case, and 88% of ground truth pairs in scaling case.

Knowing the transformation matrix, each successive frame was transformed and resampled using cubic convolution interpolation. The resampled images for "entry" sequence under scaling are shown in Fig. 5(e-g). The resampled image was then mosaiced with the reference image, as shown in Fig. 5(h-j). The registration accuracy was estimated averaging all the pixels difference of gravity centre points between first image and each of its transformed successive images. As shown in Table 2, the average registration error measured in city-block distance for “entry” sequence in Figure 5 is below 0.6 pixel in both X and Y directions.

C. *Computation Time Test on Matching*

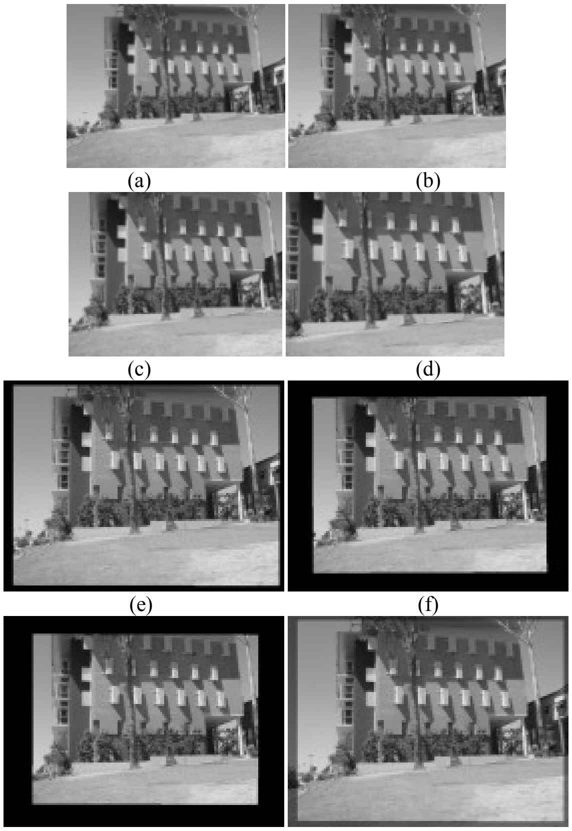
Matching using the proposed FD-based method runs much faster than SIFT on a PC with Intel(R) Core(TM)i7 CPU Q 720@1.6GHz and 8G Memory. Table 3 shows that the running time for the proposed method is 5 times faster than SIFT matching.

V. CONCLUSION

This paper has presented an original algorithm for global motion estimation in video sequences. Designed for low computation requirements, it is faster than conventional methods while providing high accuracy. Future work involves extending this method for robust real time object extracting and tracking in video sequences.

TABLE I. MATCHING RESULTS: THE LEFT COLUMN SHOWS AMOUNT OF TRANSFORMATION, AND THE VALUES ARE THE % OF RECOVERED INLIERS BETWEEN EACH IMAGE PAIR

image seqs	entry	carpark	vendor	court	hospital	lounge
Scaling	**Recovered Inliers%**					
1.1	96	93	93	95	98	94
1.2	94	95	95	98	93	96
1.3	100	100	91	91	90	100
1.4	100	100	100	86	89	91
Translation	**Recovered Inliers%**					
3%	97	95	88	93	89	95
7%	100	89	80	91	92	88
11%	94	96	90	95	81	86
15%	100	94	86	89	86	80
Rotation	**Recovered Inliers%**					
2°	100	96	96	95	95	96
6°	95	92	98	94	94	96
10°	95	88	98	93	93	94
14°	92	96	94	93	94	96
16°	88	89	93	96	96	98

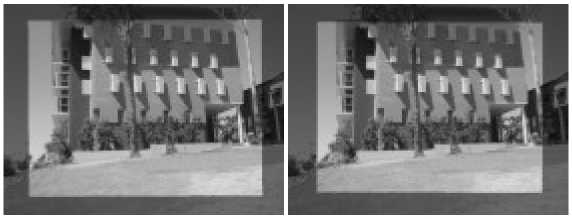


(a) (b)

(c) (d)

(e) (f)

(g) (h)



(i) (j)

Figure 5 (a-d) original "entry" sequence (e-g) resampled image given (a) and successive images (h-j) mosaiced images given (a) and resampled images

References

[1] David Fleet and Yair Weiss, "Optical Flow Estimation," in Handbook of Mathematical Models in Computer Vision, N. Paragios, Y Chen, and O. Faugeras, Eds.: Springer, 2006, ch. 15, pp. 239-258.

[2] Shan Zhu and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-Matching," IEEE TRANSACTIONS ON IMAGE PROCESSING, vol. 9, no. 2, FEBRUARY 2000.

[3] Barbara Zitová and Jan Flusser, "Image registration methods: a survey," Image and Vision Computing, vol. 21, no. 11, pp. 977–1000, October 2003.

[4] Krystian Mikolajczyk and Cordelia Schmid, "A Performance Evaluation of Local Descriptors," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 27, no. 10, pp. 1615-1630, OCTOBER 2005.

[5] David Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, November 2004.

[6] Xuesong Le and Gonzalez Ruben, "A Robust Histogram Region-based Global Camera Estimation Method for Video Sequences," in Pacific-Rim Conference on Multimedia, Sydney, 2011.

[7] Dengsheng Zhang and Guojun Lu, "A Comparative Study on Shape Retrieval Using Fourier Descriptors," in 5th Asian Conference on Computer Vision , 2002.

[8] Guojun Lu and Atul Sajjanhar, "Region-based shape representation and similarity measure suitable for content-based image retrieval," Multimedia Systems, vol. 7, no. 2, pp. 165-174, March 1999.

[9] Steven Smith, "Chapter 8: The Discrete Fourier Transform," in The Scientist and Engineer's Guide to Digital Signal Processing. San Diego: California Technical Publishing, 1999, ch. 8.

[10] Guojun Lu, "Review of shape representation anddescription techniques," Pattern Recognition, vol. 37, no. 1, pp. 1–19, January 2004.

[11] Jon Louis Bentley, "Multidimensional binary search trees used for associative searching," Communications of the ACM, vol. 18, no. 9, pp. 509-517, Sept 1975.

[12] Kuntee Viriyothai and Paul Debevec, "Variance minimization light probe sampling," in SIGGRAPH 2009, New Orleans, 2009.

[13] Jeffrey S Beis and David G Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in CVPR, 1997, pp. 1000–1006.

[14] Martin A Fischler and Robert C Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Comm. of the ACM, vol. 24, no. 6, pp. 381–395, June 1981.

[15] Apache Commons. http://commons.apache.org/proper/commons-math/.