Linear Combination Representation for Outlier Detection in Motion Tracking

Guodong Guo Charles R. Dyer
University of Wisconsin-Madison
Computer Sciences Department
Madison, WI 53706
{gdguo,dyer}@cs.wisc.edu

Zhengyou Zhang
Microsoft Research
One Microsoft Way
Redmond, WA 98052
zhang@microsoft.com

Abstract

In this paper we show that Ullman and Basri's linear combination (LC) representation, which was originally proposed for alignment-based object recognition, can be used for outlier detection in motion tracking with an affine camera. For this task LC can be realized either on image frames or feature trajectories, and therefore two methods are developed which we call linear combination of frames and linear combination of trajectories. For robust estimation of the linear combination coefficients, the support vector regression (SVR) algorithm is used and compared with the RANSAC method. SVR based on quadratic programming optimization can efficiently deal with more than 50 percent outliers and delivers more consistent results than RANSAC in our experiments. The linear combination representation can use SVR in a straightforward manner while previous factorization-based or subspace separation methods cannot. Experimental results are presented using real video sequences to demonstrate the effectiveness of our LC + SVR approaches, including a quantitative comparison of SVR and RANSAC.

1. Introduction

Robust tracking of feature points in image sequences is of great importance for tasks such as video sequence alignment [2], structure from motion [17], and motion segmentation [18]. In order to obtain good results in motion-based vision tasks, feature trajectory outliers have to be detected and removed.

For tracked features there are typically two types of errors [22]: (1) Location errors, where the location of a 2D feature is distorted and usually assumed to exhibit Gaussian behavior. (2) False matches, where there is a mismatch of 2D features, *i.e.*, two corresponding features originating from different scene points in any part of a trajectory. The first kind of error can be addressed by the rank constraint [17] as long as the location error is not too big [22], but the second kind of error usually cannot be accomodated by an estimation method. Even a small number of false matches can completely spoil the estimation process, and the final result will be useless [22]. This is also demonstrated in our experiments.

Torr [18] proposed some methods for outlier removal and motion analysis based on the affine fundamental matrix and affine trifocal tensor. Recently some other methods were developed that remove outliers in a video sequence [9] [1] [16] based on factorization methods [17] [3]. One of these approaches measured subspace distance based on a rank-4 property, but five observations were used instead of four by claiming that location errors and outliers inflate the rank [9]. Subspace separation [16] was used to deal with multiple motions by increasing the rank with the number of motions specified by the user. Another approach [1] incorporated an iterative weighting mechanism from robust statistics into a factorization method [3], which only works when the percentage of outliers is low because the M-estimator they used has a low break point. Other recent work dealt with missing features [17] [10] [6] [8]. All these approaches are based on the factorization method or subspace separation, and need to factorize a big matrix [1] or impose restrictive limitations [9] [16].

In this paper we show that for outlier detection of point feature trajectories, a different formulation can be employed that results in several improvements over existing methods. The linear combination (LC) of images representation proposed by Ullman and Basri [19] for alignment-based object recognition will be used to detect outliers. The LC relation also holds using 4 motion trajectories [21]. Based on these ideas, we develop two methods, one is frame-based and the other is trajectory-based. The approach uses linear regression without the need to factorize a large matrix [1] or impose restrictive limitations [9] [16]. Furthermore, only 4 parameters are involved for each regression. The linear combination representation can directly take advantage of support vector regression (SVR) [20], which can deal with a high percentage of outliers using slack variables.

Major contributions of this paper are: (1) showing that linear combination representations [19] [21] can be used effectively for outlier detection and removal in motion tracking, (2) designing two algorithms to eliminate outliers for image sequences, (3) using support vector regression with automatic threshold selection to cope with a high percentage of outliers and to deliver more consistent results than RANSAC, and (4) taking a simpler approach (only 4 parameters to estimate) without factorizing a large matrix as in [1], imposing restrictive limitations as in subspace separation [9], or involving 16 parameters as in the affine trifocal tensor [18].

The paper is organized as follows. In Section 2 the linear combination representations are described. SVR is presented and some key issues are discussed in Section 3. Two methods are developed in Section 4, each using either SVR or RANSAC for robust regression. In Section 5, a performance measure is proposed to evaluate outlier detection results. All proposed methods are evaluated experimentally in Section 6.

2. Linear Combination Representations

Suppose P feature points have been tracked over F frames in an image sequence, producing a sequence of image coordinates for each trajectory $\{(x_{fp}, y_{fp}) \mid f = 1, \ldots, F, p = 1, \ldots, P\}$. Let matrix W_x contain $F \times P$ x coordinates, x_{fp}, with one row per frame and one column per feature trajectory. Similarly, use W_y for the y coordinates, y_{fp}:

$$W_x = \begin{bmatrix} x_{11} & \cdots & x_{1P} \\ \vdots & & \vdots \\ x_{F1} & \cdots & x_{FP} \end{bmatrix}, \; W_y = \begin{bmatrix} y_{11} & \cdots & y_{1P} \\ \vdots & & \vdots \\ y_{F1} & \cdots & y_{FP} \end{bmatrix} \tag{1}$$

A measurement matrix, W, is built by combining W_x and W_y [17] into $W = \left[\frac{W_x}{W_y}\right]$.

Ullman and Basri [19] first showed that an image from an affine camera can be represented as a linear combination of three model images, and then used this property for alignment-based object recognition. Weinshall and Tomasi [21] observed that under arbitrary affine camera motion the image trajectories of a scene point are linear combinations of the trajectories of three reference points, and they used this property for invariant shape modeling from image sequences.

In terms of the measurement matrix, Ullman and Basri's linear combination result can be represented by

$$x_{ip} = a_x x_{jp} + b_x x_{kp} + c_x x_{lp} + d_x \tag{2}$$

$$y_{ip} = a_y y_{jp} + b_y y_{kp} + c_y y_{lp} + d_y \tag{3}$$

where x_{ip}, x_{jp}, x_{kp}, and x_{lp} are the x coordinates of any point p in four frames, i, j, k, and l, and $[a_x \;\; b_x \;\; c_x \;\; d_x]^T$ are the corresponding linear combination coefficients. When there is no translation in the sequence, $d_x = 0$. Similarly for the y coordinates in Eq. (3). Note that the coefficients for the x and y coordinates are not necessarily the same. We call this approach *linear combination of frames*.

In contrast, Weinshall and Tomasi's result can be represented with respect to the columns in the measurement matrix W:

$$x_{\cdot i} = a_{tx} x_{\cdot j} + b_{tx} x_{\cdot k} + c_{tx} x_{\cdot l} + d_{tx} \tag{4}$$

$$y_{\cdot i} = a_{ty} y_{\cdot j} + b_{ty} y_{\cdot k} + c_{ty} y_{\cdot l} + d_{ty} \tag{5}$$

where $x_{\cdot i}, x_{\cdot j}, x_{\cdot k}$, and $x_{\cdot l}$ are the x coordinates of four trajectories, i, j, k, and l, and $[a_{tx} \;\; b_{tx} \;\; c_{tx} \;\; d_{tx}]^T$ are the linear combination coefficients. Similarly for the y coordinates in Eq. (5). The coefficients for the x and y coordinates may or may not be the same, but this does not matter for outlier detection. We call this approach *linear combination of trajectories.*

Machline et al. [12] discussed the distinction between the row space and column space of the measurement matrix based on subspace analysis. They called $\left[\frac{W_x}{W_y}\right]$ the "trajectory matrix" and $[W_x | W_y]$ the "flow-field matrix". The underlying basis of the linear combination representations and subspace analysis is the same rank-4 constraint for general affine motion or rank-3 when translation is removed. But the linear combination formulation can use SVR techniques directly, while any subspace or factorization-based method cannot.

3. Support Vector Regression

Linear support vector regression (SVR) can be used to estimate the linear combination coefficients in Eqs. (2)-(5). Here we simply describe the basic theory of SVR and then show an example to illustrate some important issues.

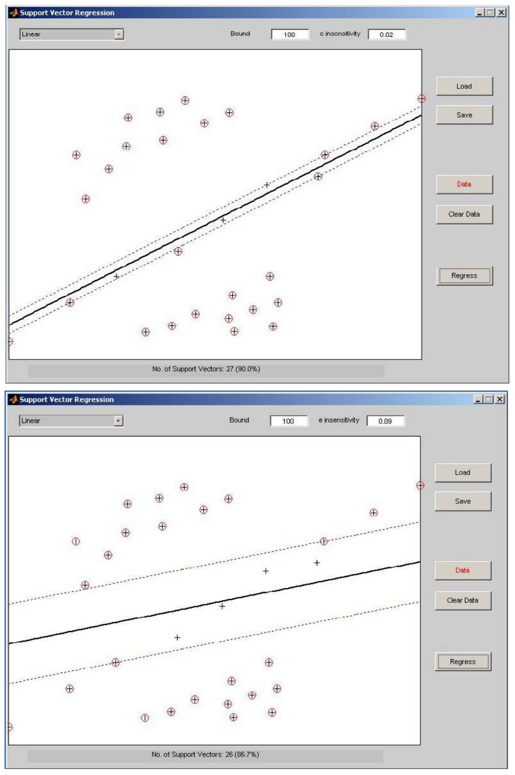
3.1. Basic Theory

Consider the problem of approximating the set of data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_l, y_l)\}, x \in R^n, y \in R$, with a linear function,

$$f(x) = \langle w, x \rangle + b. \tag{6}$$

The optimal regression function [20] is given by

$$\begin{array}{cc} \min\limits_{w,\xi} & \frac{1}{2} \parallel w \parallel^2 + C \; \sum_{i=1}^{l} (\xi_i^+ + \xi_i^-) \\ & y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i^+ \\ subject \;\; to & \langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i^- \\ & \xi_i^+, \xi_i^- \leq 0 \end{array} \tag{7}$$


Support Vector Regression

Linear

Bound 100 ε insensitivity 0.02

No. of Support Vectors: 27 (90.0%)

(a)

Support Vector Regression

Linear

Bound 100 ε insensitivity 0.09

No. of Support Vectors: 26 (86.7%)

(b)

Figure 1. SVR on real 2D data with ϵ = 0.02 in (a) and ϵ = 0.09 in (b). Note that the support vectors (marked by circles) are not the true outliers in either case.

where constant $C > 0$ determines the trade-off between the flatness of f and data deviations, and ξ_i^+, ξ_i^- are slack variables to cope with otherwise infeasible constraints on the optimization problem of (7). The ϵ-insensitive loss function is

$$L_\epsilon(x, y) = \begin{cases} 0 & if\ |f(x) - y| < \epsilon \\ |f(x) - y| - \epsilon & otherwise \end{cases} \quad (8)$$

The *primal* problem of (7) can be solved more easily in its *dual* formulation [20] resulting in the final solution given by

$$w = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) x_i \quad (9)$$

where α_i, α_i^* are Lagrange multipliers. The value of b in Eq. (6) can be determined by plugging Eq. (9) into Eq. (6) [5]. See [20] for more details.

3.2. A Toy Example

To illustrate the SVR idea, we use a toy example that contains 30 points in 2D with 10 in a line and the remaining 20 being outliers distributed on both sides of the line. Hence the data contains 67% outliers. Using the SVR algorithm implemented by Gunn [5] (which provides a user interface) and a linear kernel with ϵ = 0.02, the result is shown in Fig. 1(a). Observe that the line was correctly estimated despite the high percentage of outliers.

On the other hand, observe that SVR returns 27 support vectors (90% of the input data) and 7 of them are very close to the boundaries (two dashed lines), but there are actually 20 outliers in the original data. So we can not simply classify the support vectors (SVs) as the outliers. Increasing the ϵ value might "drag" the 7 closest support vectors inside the dashed boundaries, and then only the outliers in the data would be returned as support vectors. However, when we increase ϵ gradually up to 0.09, there are still 26 SVs returned which are still not the true outliers, as show in Fig. 1(b). And even worse, the slope of the line has changed significantly. This demonstrates that using a large ϵ is not a good idea because it may degrade the model structure.

Based on this experiment, we observe: (1) the SVR technique can potentially deal with data containing a high percentage of outliers, (2) classifying support vectors as outliers is not workable, (3) using a large value for ϵ is not a good idea for SVR, and (4) using small ϵ is preferable, especially when a large number of outliers are present.

As a result of these observations, we used a small fixed value, ϵ = 0.01, for SVR in all our experiments. We used the knee point of the residuals (see Section 3.3) to automatically choose a threshold to separate outliers from inliers.

SVR was used in [23] for two-view affine matching. They iteratively changed ϵ from big to small, and classified the SVs as outliers. As shown in the experiment in Figure 1, this approach not only misclassifies many outliers but using big ϵ may change the underlying structure. Furthermore, changing ϵ iteratively is computationally expensive because SVR optimization is executed for each ϵ.

In our linear combination representation, i.e., Eqs. (2)-(5), there is only one output variable for each regression, so standard SVR code can be used. We used SVM^{light} [11] in our experiments.

3.3. Automatic Threshold Determination

After using SVR, we measure the regression residual for each input data value and sort them in descending order. Typically, we get a curve of the sorted residuals as shown in Fig. 2, which was obtained in our experiments (see Section 6). The residual will be big for outliers and small for inliers. The knee point in the curve, which is marked by a circle in Fig. 2, is the point at which the second derivative attains a maximum. The knee point is used as a threshold only when it is bigger than the pre-set ϵ value for SVR.

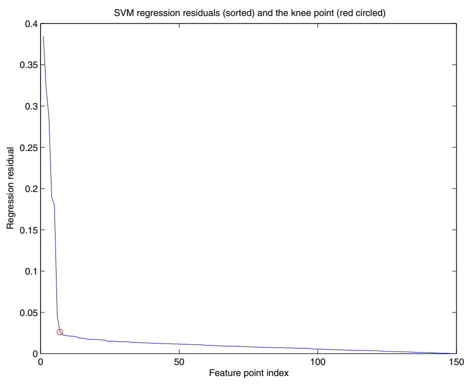


Figure 2. The SVR residual curve (sorted in descending order) and the computed knee point (marked with a circle).

3.4. SVR vs. RANSAC

The SVR method [20] employs all input points simultaneously. The ϵ-insensitive loss function proposed by Vapnik [20] is an approximation to Huber's loss function (used in M-estimators) to obtain a sparse set of support vectors [5]. It is well known that the M-estimator has a low break point in robust regression. SVR, on the other hand, can introduce slack variables to cope with otherwise infeasible constraints, i.e., dealing with outliers to some extent but not exactly. It is the using of slack variables (in quadratic programming) that gives SVR the potential to deal with a high percentage of outliers. Theoretical properties of the break point of SVR are not yet known.

As part of our evaluation of SVR for outlier detection we will compare its performance to one of the standard methods for robust regression, RANSAC, for RANdom SAmple Consensus [4], which uses a small number of randomly selected points to estimate the underlying model and uses the remaining points for verification. The process iterates many times, and the model with the largest support from the input points is selected as the model. The break point of RANSAC is about 50%.

To our knowledge, there is no previous comparison between SVR and RANSAC either conceptually or experimentally. We will compare them in our linear combination approaches experimentally.

4. Two Approaches

As shown in Section 2, the linear combination relation is satisfied for either four frames or four trajectories in a measurement matrix W without false matches. One can also factorize W for structure and motion estimation [17]. In practice, however, there are outliers in W that affect the solution [1] [9]. Our goal is to detect outliers in W based on the linear combination representation.

For image sequences captured by an affine camera, two methods for outlier detection based on two different linear combination representations are presented. The first processes image frames and the second operates on point trajectories. We call them linear combination of frames (FLC) and linear combination of trajectories (TLC), respectively. For robust regression, SVR will be compared with RANSAC.

4.1. Linear Combination of Frames

To use FLC for an image sequence requires dividing the sequence into different groups, each with four images. One way is to take four consecutive frames as a group, and sequentially use the FLC method for each consecutive group of four frames. In this case, however, there may exist little or no movement between the four images and so the coefficients $[a_x \quad b_x \quad c_x \quad d_x]^T$ and $[a_y \quad b_y \quad c_y \quad d_y]^T$ can not be well estimated. In contrast, a large temporal separation between the four frames can make the computation more robust because the linear equations for the four frames are nearly independent of each other. Based on these considerations, we divide the image sequence into four sections with no overlap, and each section has about the same length. Then, we sequentially extract one frame from each section to constitute a group until all frames are used. Assuming each section has s frames, we get s four-frame groups and therefore s FLC computations.

For each four-frame group, SVR or RANSAC (see p. 103 in [7] for algorithm details) is used to estimate the linear combination coefficients $[a_x \quad b_x \quad c_x \quad d_x]^T$ and $[a_y \quad b_y \quad c_y \quad d_y]^T$, and classify the corresponding trajectories as inliers or outliers.

Dealing with the x and y coordinates separately in Eqs. (2) and (3), we get inlier sets I_x and I_y, and outlier sets O_x and O_y. Then the true inliers are in the set $I = I_x \cap I_y$. The outliers are in the set $O = (I_x \cup O_x) - I$, that is, the points that do not satisfy the linear combination relation in either their x or y coordinate.

The FLC scheme is applied sequentially to the s groups. That is, FLC is carried out one-by-one to the sequence of four-frame groups. Only the inliers in the current group can be used in the next group. Hence the number of inliers decreases monotonically. Conversely, the number of outliers increases after each group is processed.

In summary, the FLC algorithm for outlier removal consists of the following steps: (1) Divide the given image sequence into 4 non-overlapping sections, each of about the same length. (2) Sequentially choose one frame from each section to construct a group of 4 frames. (3) Use SVR or

RANSAC to estimate the linear combination coefficients for the selected 4 frames and determine the inliers and outliers based on the 4 frames. (4) Mark the inliers and go to Step 2 using the next group of 4 frames until all frames are processed. (5) Report the inliers and outliers in the whole sequence.

4.2. Linear Combination of Trajectories

In addition to working on image frames, the linear combination relation also holds for point trajectories, i.e., Eqs. (4) and (5). Here we show how to remove outliers using a linear combination of point trajectories. Before presenting the algorithm, three properties are shown.

Property 1. *If four inlier trajectories, i, j, k, and l, satisfy the linear combination relation of Eqs. (4) and (5), then any four elements $x_{ni}, x_{nj}, x_{nk}, x_{nl}$ and $y_{ni}, y_{nj}, y_{nk}, y_{nl}$ in frame n also satisfy Eqs. (4) and (5) respectively with the same coefficients.*

Proof. Each trajectory is a column vector in W, so the relation between four vectors of the form $X_4 = a_1X_1 + a_2X_2 + a_3X_3 + a_4\mathbf{1}$, is also satisfied by each element of the four vectors corresponding to the same frame number. $\mathbf{1}$ is a vector of 1s.

Property 2. *Given four inlier trajectories, i, j, k, and l, if four corresponding elements in any single frame satisfy the linear combination relation of Eqs. (4) and (5), then these four trajectories also satisfy Eqs. (4) and (5) with the same coefficients.*

Proof. Consider each trajectory as a column vector X_h, $h = 1, 2, 3, 4$. Then the relation of the corresponding element, $x_{n4} = a_1x_{n1} + a_2x_{n2} + a_3x_{n3} + a_4$, is also satisfied by the four vectors, $X_4 = a_1X_1 + a_2X_2 + a_3X_3 + a_4\mathbf{1}$, where x_{nh} is the n^{th} element of vector X_h.

Property 3. *Given three inlier trajectories, i, j, and k, an unknown trajectory, l, and a set of linear combination coefficients $[a_t \;\; b_t \;\; c_t \;\; d_t]^T$, if any corresponding elements of the four trajectories do not satisfy the relation in Eqs. (4) and (5), then trajectory l is an outlier.*

Proof. This is a corollary of Property 1 because if trajectory l is an inlier, all elements will satisfy the linear combination relation with the same coefficients.

Properties 1 and 3 are used for RANSAC regression in order to verify whether a trajectory is an inlier or outlier, and Property 2 is used to compute the LC coefficients from four trajectories. For SVR, all data are used in four trajectories.

Since the TLC scheme characterizes only four trajectories at a time, a two-step procedure is developed to process all trajectories. The first step selects four trajectories as a basis, and the second step uses the basis to verify all the remaining trajectories:

Step 1. Select a Basis

Randomly select four trajectories and compute the linear combination coefficients. Note that the coefficients for the x and y coordinates are computed separately. The LC relation is checked for each element of the four trajectories. The residual for each element is calculated, and the maximum is recorded for the selected four trajectories.

This process is repeated many times, and the group of four trajectories with the smallest residual is selected as the basis. To improve stability, we divide all points in the first frame into 4 quadrants and randomly select one point from each quadrant.

Step 2. Verify Remaining Trajectories

After the basis is chosen, it is used to check the other trajectories. Randomly replace one trajectory in the basis with a new trajectory, and re-compute the linear combination coefficients. Using the new coefficients, re-estimate the residual for each element. If the maximum residual is small, the new trajectory is an inlier; otherwise, it is an outlier. The threshold is automatically determined by the knee point as shown in Fig. 2.

All remaining trajectories are verified in the same way in order to detect all outliers.

4.3. Characteristics of the Two Methods

In FLC, an image sequence is divided into four sections, and different four-frame groups are constructed. In this way, it is not expected to detect all outliers at once. Instead, the trajectories are analyzed in different groups and the number of inlier trajectories decreases gradually after each group computation. This is of benefit when there exist a large number of outliers.

The FLC scheme works on 4 frames, and each time there are only 4 parameters to estimate (note that Eqs. (2) and (3) deal with the x and y coordinates independently), which is one benefit of the FLC representation. On the contrary, the affine trifocal tensor [18] uses 3 frames, but there are 16 parameters to estimate together with 9 constraints. It is well-known that the more parameters there are to estimate, the more susceptible random sampling type methods [18] such as least median of squares (LMedS) [22] and RANSAC [4] are to instability. In two-view affine geometry, there are 6 parameters to estimate, which is also larger than the 4 in our case. In affine point transfer [14], correspondence between two images must be established before the corresponding point in a third image can be computed by transfer, which is different from FLC. Finally, FLC is easy to understand and very simple to implement.

In TLC, a two-step procedure was presented. The first step finds four trajectories that satisfy the LC relation well. In the case of a high percentage of outliers, this step simplifies the problem and is still capable of finding four good trajectories. This step does not find all inliers, but only four. Then we use these four trajectories to verify other trajectories based on the LC relation. This two-step approach simplifies the problem of outlier detection by incrementally testing each trajectory.

Trajectories are also considered in [9] [16], but their formulation of subspace separation can not use SVR, which can deal with a large number of outliers.

Finally, the linear combination representation is a unified framework that can be used with either image frames or point trajectories in a similar manner.

5. Quantitative Performance Evaluation

In previous work inlier feature trajectories have been displayed for visual inspection or used as input to some other process (e.g., the factorization method for structure and motion) to evaluate the results of outlier removal. Instead, we will use a quantitative performance measure.

Consider the outlier detection problem as a classification problem. With a database of feature matches labelled as inliers and outliers, we will measure the classification performance quantitatively by computing false positive and false negative rates. False positives (FP) specify when inliers are incorrectly classified as outliers. False negatives (FN) measure when outliers are incorrectly classified as inliers.

6. Experiments

We tested our linear combination methods for outlier detection on several video sequences. Because of space limits, only two sequences are reported here. For each sequence, feature points were detected and tracked using the KLT tracker [15]. The FLC and TLC algorithms were each applied to the measurement matrix W.

The first sequence contains a hotel and has frames of size 480×512. We used the first 48 frames, though any length can be used. Unlike the subspace separation approach [9], which requires a sparse sampling of images, we used all 48 consecutive images containing both in-plane and in-depth rotations. Image coordinates were normalized between 0 and 1. The first frame with tracked points (red dots, which can be seen in the electronic version) are shown in Fig. 4(a) and the trajectories are shown in 4(b). 150 features were detected and tracked over all 48 frames, containing 12 outliers (visually labelled as ground truth). The inliers and outliers determined by FLC + RANSAC are shown in Fig. 4(c) and (d), respectively. The results of TLC + RANSAC are shown in Fig. 4(e) and (f). The results of FLC + SVR are shown in Fig. 4(g) and (h), and the results of TLC + SVR are in Fig. 4(i) and (j).

The FLC + RANSAC approach detected 15 outliers and TLC + RANSAC found 14 outliers. Both FLC + SVR and TLC + SVR returned the same 12 outliers as the ground truth. The FP and FN rates are given in Table 1. Observe that each approach detected all true outliers, but SVR gave better results (0 FP and 0 FN) than RANSAC in both the frame-based and trajectory-based versions.

Table 1. Performance evaluation of the approaches. F+R stands for FLC+RANSAC, T+R for TLC+RANSAC, F+S for FLC+SVR, and T+S for TLC+SVR.

Seq.	Measure	F + R	T + R	F + S	T + S
Hotel	FP	2.2%	1.5%	0	0
	FN	0	0	0	0
Ball	FP	2.1%	0.7%	0.7%	0.7%
	FN	0	0	0	0

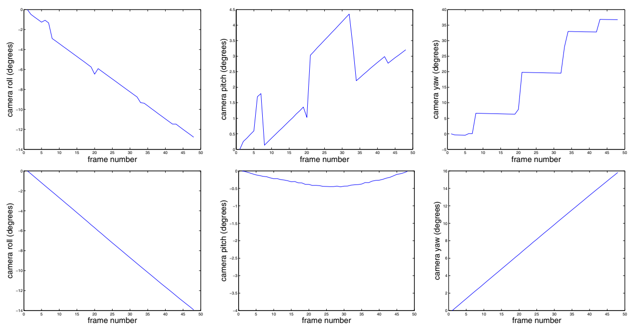


Figure 3. Camera motion (roll, pitch, and yaw in the 3 columns, respectively) estimation for the hotel sequence computed using the factorization method.

To further verify the correctness of the inliers, we used the factorization method [17] to compute the motion with and without outlier removal. The first row in Fig. 3 used all trajectories and the resulting motion is noticeably noisy. The smooth motion curves computed using the inliers detected by FLC + SVR are shown in the second row of Fig. 3. The results of TLC + SVR and LC + RANSAC were almost identical and are not shown here. Notice that the pitch values in particular are very different without outlier removal. This figure also demonstrates that outlier removal is essential for motion estimation.

The second experiment used a ping-pong ball sequence with frames of size 480×512. There is a large in-depth rotation in the sequence. 150 features were tracked in the first 12 frames, containing 8 outliers (ground truth). The first image with tracked points (red dots) is shown in Fig. 5(a) and the trajectories are shown in 5(b). The FLC + RANSAC method detected 11 outliers shown in Fig. 5(d) and inliers in (c). The TLC + RANSAC method returned 9 outliers shown in Fig. 5(f). The FLC + SVR and TLC + SVR methods detected 9 outliers shown in Fig. 5(h) and 5(j), respectively. The FP and FN rates for each method are given in Table 1. The FP of SVR is only 0.7% for either FLC or TLC, while the FN of RANSAC is 2.1% in FLC and 0.7% in TLC. In this experiment, SVR consistently gave equal or better results than RANSAC.

In both sequences, the number of outliers is small. We will investigate in the future the potential of SVR for sequences with a large number of outliers.

To use RANSAC, the smallest number of points required is 4. The threshold used to classify a point was set by trial and error. The fitting error was defined as the maximum residual of the linear combination.

Finally, the LC methods can be extended for outlier correction as in [8] [6], which we have not investigated yet.

7. Concluding Remarks

A new representation for detecting and eliminating outlier trajectories of point features in image sequences assuming an affine camera model has been described. Two methods were developed based on the linear combination idea. One is frame-based and the other is trajectory-based. For robust estimation, support vector regression was used and compared with RANSAC. Both our FLC and TLC methods detected and removed outliers completely in real video sequences and SVR gave slightly better results than RANSAC in terms of false positive rates. Our LC + SVR approach is very simple (only 4 parameters to estimate) and robust (using slack variables) without factorizing a large matrix as in [1], imposing limitations as in subspace separation [9], or involving 16 parameters as using the affine trifocal tensor [18].

References

[1] H. Aanas, R. Fisker, K. Astrom, and J. M. Carstensen, "Robust factorization," *IEEE Trans. Pattern Analysis and Machine Intell.* **24**(9), 1215-1225, 2002.

[2] Y. Caspi and M. Irani, "Alignment of non-overlapping sequences," *Proc. Int. Conf. Computer Vision*, 76-83, 2001.

[3] S. Christy and R. Horaud, "Euclidian shape and motion from multiple perspective views by affine iteration," *IEEE Trans. Pattern Analysis and Machine Intell.* **18**(11), 1098-1104, 1996.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. ACM* **24**(6), 381-395, 1981.

[5] S. R. Gunn, "Support vector machines for classification and regression," technical report, Image Speech and Intelligent Systems Research Group, University of Southampton, 1997.

[6] R. Hartley and F. Schaffalitzky, "PowerFactorization: An approach to affine reconstruction with missing and uncertain data,", *Proc. Australia-Japan Advanced Workshop on Computer Vision*, 2003.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[8] D. Q. Huynh, R. Hartly, and A. Heyden, "Outlier correction in image sequences for the affine camera," *Proc. Int. Conf. Computer Vision*, 585-590, 2003.

[9] D. Q. Huynh and A. Heyden, "Outlier detection in video sequences under affine projection," *Proc. Conf. Computer Vision and Pattern Recognition*, 695-701, 2001.

[10] D. Jacobs, "Linear fitting with missing data: Applications to structure-from-motion and its characterizing intensity images," *Proc. Conf. Computer Vision and Pattern Recognition*, 206-212, 1997.

[11] T. Joachims, "Making large-scale SVM learning practical," *Advances in Kernel Methods - Support Vector Learning*, B. Schlkopf and C. Burges and A. Smola (ed.), MIT Press, 1999.

[12] M. Machline, L.Zelnik-Manor, and M. Irani, "Multi-body segmentation: revisiting motion consistency. *Proc. Workshop Vision and Modelling of Dynamic Scenes*, 2002.

[13] J. L. Mundy and A. Zisserman, "Appendix - Projective geometry for machine vision," In *Geometric Invariance in Computer Vision*, MIT Press, Cambridge, Mass., 463-534, 1992.

[14] I. D. Reid and D. W. Murray, "Active tracking of foveated feature clusters using affine structure," *Int. J. Computer Vision*, **18**(1), 1-20, 1996.

[15] J. Shi and C. Tomasi, "Good features to track," *Proc. Conf. Computer Vision and Pattern Recognition*, 593-600, 1994.

[16] Y. Sugaya and K. Kanatani, "Outlier removal for motion tracking by subspace separation," *IEICE Trans. Inf. & Syst.* **E86-D**(6), 1095-1102, 2003.

[17] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. J. Computer Vision* **9**(2), 137-154, 1992.

[18] P. Torr, *Motion Segmentation and Outlier Detection*, PhD thesis, Department of Engineering Science, University of Oxford, 1995.

[19] S. Ullman and R. Basri, "Recognition by linear combination of models," *IEEE Trans. Pattern Analysis and Machine Intell.* **13**(10), 992-1006, 1991.

[20] V. N. Vapnik, *Statistical Learning Theory*, John Wiley, New York, 1998.

[21] D. Weinshall and C. Tomasi, "Linear and incremental acquisition of invatiant shape models from image sequences," *IEEE Trans. Pattern Analysis and Machine Intell.* **17**(5), 512-517, 1995.

[22] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence* **75**(1-2), 87-120, 1995.

[23] W. Zhu, S. Wang, R. Lin, S. Levinson, "Tracking of object with SVM regression," *Proc. Conf. Computer Vision and Pattern Recognition*, v. 2, 240-245, 2001.

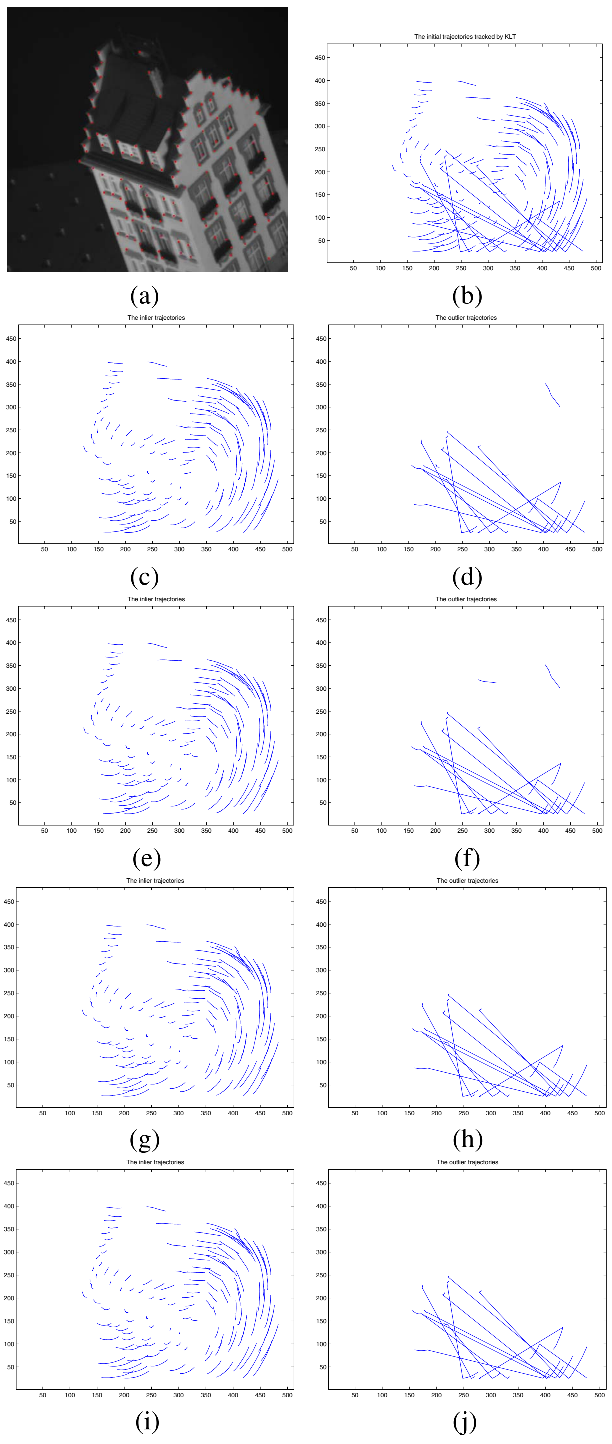


Figure 4. The first frame (a) and the KLT tracked trajectories (b) of the hotel sequence. Inliers (c) and outliers (d) computed by FLC+RANSAC. Inliers (e) and outliers (f) by TLC+RANSAC. Inliers (g) and outliers (h) by FLC+SVR. Inliers (i) and outliers (j) by TLC+SVR.

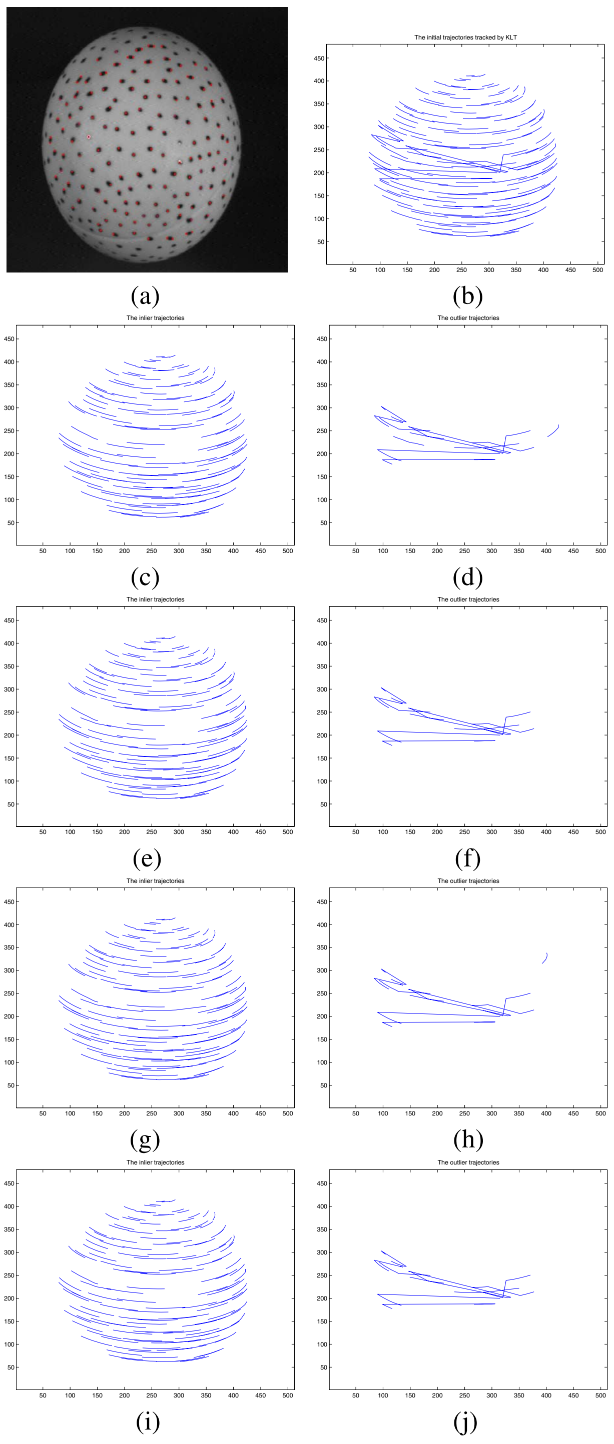


Figure 5. The first frame (a) and the KLT tracked trajectories (b) of the ping-pong ball sequence. Inliers (c) and outliers (d) computed by FLC+RANSAC. Inliers (e) and outliers (f) by TLC+RANSAC. Inliers (g) and outliers (h) by FLC+SVR. Inliers (i) and outliers (j) by TLC+SVR.