Simultaneous Modeling and Tracking (SMAT) of feature sets

NDH Dowson and R Bowden
Centre for Speech Vision and Signal Processing
University of Surrey
Guildford, GU2 7XH, UK
n.dowson;r.bowden@surrey.ac.uk

Abstract

A novel method for the simultaneous modeling and tracking (SMAT) of a feature set during motion sequence is proposed. The method requires no prior information. Instead the a posteriori *distribution of appearance and shape is built up incrementally using an exemplar based approach. The resulting model is less optimal than when* a priori *data is used, but can be built in real-time. Data in any form may be used, provided a distance measure and a means to classify outliers exists. Here, a two tier implementation of SMAT is used: at the feature level, mutual information is used to track image patches; and at the object level, a structure model is built from the feature positions. As experiments demonstrate, the tracker is robust and operates in real-time without requiring prelearned data.*

1. Introduction

The goal of this work was to robustly track objects in cluttered scenes without a prelearned model of appearance or structure. Despite much research, modeling and tracking features as they vary in shape and appearance is still difficult. Appearance based online tracking is generally solved by using *a priori* learning, by tracking large information rich regions, or by tracking groups of small patches. (The position of features within the object was important, precluding contour tracking.)

The use of *a priori* data is avoided, since it is often difficult to obtain (e.g. manual ground-truthing, photographing subjects). Also, preprocessing may be required (e.g. boosting to build hockey player models by Okuma *et al* [8]), restricting one to a particular application. Tracking objects can be effective without *a priori* data, e.g. the trackers of Jepson *et al* [4] and Collins & Liu [2]. However for [2], features within the object cannot be matched, since it is assumed that the object is contained by the central region of a rectangle. Likewise for [4], which tracks large objects well, but is then too slow for real-time use.

Hence this work focusses on tracking small patches. The method presented: Simultaneous Modeling and Tracking (SMAT), overcomes the need for *a priori* data while still maintaining the desirable properties of being robust and operating in real-time. This is achieved by a generalisation and improvement of the work in [3] to a model selection framework, where the focus is on tracking single features based on appearance only. While tracking, a mixture model is fitted to the exemplar appearances and structures obtained from the sequence. A semi-optimal component for tracking is (greedily) selected online based on the confidence value of each component. Given sufficient samples the model becomes as close to optimal as available data allows.

SMAT is related to Jepson *et al*'s work [4], where objects are tracked using a three component model: a stable, wandering and lost component. This elegant blend of a 2-frame tracker and an alpha filter performs well for large objects but tends to drift for small patches. SMAT attempts to overcome drift directly and is more closely related to the work on the template update problem by Matthews *et al* [7] and Kaneko & Hori [5].

The remainder of the paper is organized as follows. After some background, the SMAT method is presented in Section 2. Section 3 describes the implementation of the proposed approach. Section 4 presents the experiments comparing SMAT to other approaches and the results obtained, followed by the conclusion in Section 5.

1.1. Background

One of the earliest attempts to locate a representative feature in an image was made by Lucas and Kanade [6]. They limited the processing by using a Newton-Raphson method to traverse the search space. Although only translations were considered, they mentioned how the method could easily be adapted to optimize other transforms such as rotation. Indeed, later research extended this method to consider rotation, scale, and affine transforms [1]. Also new minimization methods have been applied, which have faster convergence and greater likelihood of finding the global minimum, such as the Levenberg-Marquardt method.

It is natural to extend feature detection to tracking a feature over a motion sequence. This leads to a further problem: how and when to update the template. If the template is never updated, tracking will only work as long the template closely represents the current appearance of the feature. This assumption is generally safe for several frames after the one from which the feature was extracted. Eventually however, the template does not represent the feature sufficiently well and a catastrophic failure ensues: i.e. the error suddenly becomes very large. One alternative to this is to update the template after every frame. However, sub-pixel errors inherent to each match are stored in each update. These drift errors gradually accumulate and the template drifts off the original feature.

Two recent approaches to overcome the problem of drift, include those by Matthews *et al* in [7], and Kaneko and Hori in [5]. Matthews *strategic update* approach is a simple but effective extension of the naive update algorithm, where the first template from the frame is retained and used to correct location errors made by the updated template. If the size of the correction is too large, the algorithm acts conservatively by not updating the template from the current frame. Kaneko on the other hand trades off between accumulated drift error and misrepresentation error. In Kaneko's algorithm the template is updated just before a catastrophic error occurs. Each of these errors is estimated from the boundary of the maximum error for possible templates.

Although [7] was perhaps not intended in the context of matching small patches, the single drawback to Matthews approach, is that the appearance of the feature could eventually change enough that the distance between the two matches will always differ by a large amount. This will result in no further updates being made, and failure due to misrepresentation. Kaneko's approach attempts to minimize the number of updates, but eventually sub-pixel errors accumulate sufficiently for drift to occur. Both of these methods also rely on the first template being a good representation of the feature.

The SMAT approach grew out of attempts to overcome the problems of drift and misrepresentation errors, by storing as many exemplars as required to fully describe a feature. Updates required could theoretically be reduced to zero, while the feature should never be misrepresented.

2. Simultaneous Modeling and Tracking

It is useful to begin by formalizing the problem of tracking an object in an image given a template. Given a motion sequence of N images, where $S_n(\mathbf{x})$ represents the pixel intensity at position $\mathbf{x}$ in the frame S_n, a transformation or warp W is sought that minimizes some distance function d between the template T_n and S_n:

$$\arg\min_{\forall p} d\left[S_n(\mathbf{x}), T_n(W(\mathbf{x},\mathbf{p}))\right] \quad (1)$$

where $\mathbf{p}$ contains the parameters of the warp. Distance functions that increase with greater similarity may be trivially converted to the above form by a multiplication by -1.

No assumptions about the object appearance and structure are made. Instead a probabilistic model that describes variations in the object is built on the fly, in an approach inspired by [10]. Nothing is known about the object appearance before operation except for the single exemplar T_0 extracted from S_0, which on its own would be an inadequate description. Simplistic updates serve only to move the position of the model in feature space rather than expanding it to properly describe S's occupation. To overcome this, the distribution function $P(T_n|S_n \ldots S_0)$ is constructed incrementally. The model P is represented as a mixture of K components:

$$P(T_n) = \sum_{k=1}^{K} w_k \eta(\mu_k, \rho_k) \quad (2)$$

where w_k is the weight parameter of the kth component and satisfies the condition $\sum_{\forall k} w_k = 1$. In the distribution of the kth component: $\eta(\mu_k, \rho_k)$, μ_k is the median kernel and ρ is some parameter(s) of the distribution. The distance from the median is calculated by the distance function d.

Considering templates as points in some probability space, similar templates will tend to cluster together. These clusters are what each component in the mixture represents. Overly large components could result in misrepresentation of the object, so some discrimination between inliers and outliers is required. The membership m_{nk} of a particular exemplar S_n to a component η_k is determined by its proximity to the component's representative exemplar, which is also its median, i.e. $T_k = \mu_k$:

$$m_n k = \begin{cases} 1 & d(S_n, T_k) < \tau(\rho_k) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where τ is the threshold. Of course if a Gaussian distribution was assumed, τ would simply be a set number of standard deviations to explicitly choose what percentage of data was outlier data. The use of a median rather than a mean could result in an expanded component if the median and mean are far apart. However, this should not occur if the exemplars are correctly clustered, and if it does, a tighter inlier threshold will counteract this effect.

Components that are sufficiently close together may overlap, so a template may belong to more than one component. K will generally be some small number greater than one, since although more than K clusters may be required to fully describe an object over all poses, the clusters update themselves to describe the object for the poses that have recently been relevant.

To obtain the median, the standard approach of constructing a matrix of distances between each pair of templates, is used:

$$C = \begin{pmatrix} d(T_1,T_1) & d(T_1,T_2) & \dots & d(T_1,T_M) \\ \vdots & & & \vdots \\ d(T_M,T_1) & d(T_M,T_2) & \dots & d(T_M,T_M) \end{pmatrix} \quad (4)$$

The sum of each column in C is found and the exemplar corresponding to the column with the lowest sum is chosen as the median. The cluster is limited to M samples, to limit computation time. M can be large since it is cheap to compute incrementally, so each component is representative of the local cluster of templates. When M is exceeded, the most outlying exemplar is replaced, i.e. $\arg\min_k(w_k)$. The median rather than the mean of each component is used, since a mean in image space results in blurring and a loss of feature representation. Moreover a mean is less robust to outliers.

In general, newly created components are less reliable than older ones, since they have fewer samples and the variation in structure and appearance may be due to transient environmental effects. To model the effect of a particular component gaining "trust", its weight is increased each time a region in the image is located that is an inlier. The increase is at the expense of the other components:

$$w_k^{(n+1)} = \begin{cases} (w_k^{(n)} + \alpha)\frac{1}{1+\alpha} & k = K' \\ w_k^{(n)}\frac{1}{1+\alpha} & \text{otherwise} \end{cases} \quad (5)$$

where K' is the index of the successfully matched component and α is the learning rate such that $\alpha \in (0;1)$. α^{-1} defines the time constant of change. The learning parameter sets the rate at which component rankings change, how rapidly outmoded representations are removed, and the speed at which "trust" in new representations gained.

Using all K components to search for an object in each frame is time-consuming and unnecessary. In practice, a representation that is within the threshold τ_k of the current component is adequate. So, a greedy algorithm is employed that uses the components in descending order of weight w_k. If a successful match is made, no further components are checked and the new exemplar is incorporated into the successfully matching cluster.

Initially only one component exists. The first outlier T_n to this component is used to seed a new component. A basic assumption is made that components explaining the first α percent of the model (say 80%) are inliers and the remaining components are outliers. This means that any new components need to gain sufficient weight before they may be used. For this reason, when traversing the mixture model, the "untrusted" components, may still have exemplars added to them, however they are not used to update the warp parameters. After sufficient matches have been made the component is treated as inlying data and used to track the object normally.

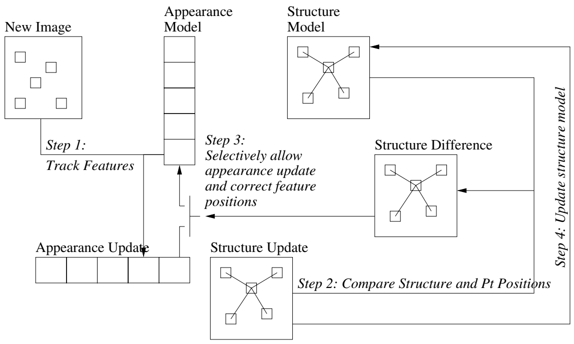
New Image
Appearance Model
Structure Model
Step 1: Track Features
Step 3: Selectively allow appearance update and correct feature positions
Structure Difference
Step 4: Update structure model
Appearance Update
Structure Update
Step 2: Compare Structure and Pt Positions

Figure 1. Schematic of 2 tier SMAT

When all K components are populated, it is still conceivable that the feature can take on an appearance outside of the existing model. When this occurs, the lowest weighted component is destroyed and a new one is created in its place, allowing obsolete representations to be aged out.

SMAT may be applied to any type of data, be it a sparse set of point positions, an image or a contour, so long as some means exists to describe the data and some distance measure exists that inliers and outliers may be classed differentiated, as is shown in the next section on the proposed implementation of an object tracker.

3. Implementation

It was decided to use SMAT in two tiers: at the low level for modeling individual features on an object based on appearance, and at a higher level for modeling the structure of the object as it deforms. The data at the appearance and structure levels was treated semi-independently. Updates to individual appearance models were sometimes prevented based their disparity with the structure model. This ties the models together sufficiently to improve robustness. A schematic of the two tier approach is presented in Figure 1.

A strong dependence between models (e.g. by concatenating the appearance and structure vectors) was avoided for two reasons. Firstly, different distance metrics were used to measure the similarity between the object feature appearance and structure: mutual information and Mahalanobis distance respectively. Secondly, when appearance and structure are dependent, many more exemplars would be required for the model to become reliable implying an increased likelihood of failure early on for dependent appearance and structure.

3.1. Modeling Appearance

The appearance model was built from image patches of the features being tracked in the sequence. Numerous methods exist for comparing image patches, such as Sum of Squared Difference (SSD), normalized SSD and Mutual Information (MI). SSD is probably the most wide spread, since it is simple to implement, fast and easy to differentiate analytically. However, Mutual Information was chosen to measure the distance between appearance models, due to its robustness to environmental lighting conditions, robustness to noise, pronounced maxima and similar computational cost to SSD.

MI quantifies information shared between two sets of data, S and T:

$$d_J(S,T) = H(S) + H(T) - H(S,T) \tag{6}$$

In this case S is a region in the image and T is the template. $H(.)$ is the entropy of the probability distribution function as defined by Shannon in [9]:

$$\begin{aligned} H(S) &= -\sum_{\forall i} p(i_S) \ln p(i_S) && (7) \\ H(S,T) &= -\sum_{\forall i,j} p(i_S, j_T) \ln p(i_S, j_T) && (8) \end{aligned}$$

where $p(i_S)$ is the probability that a random pixel in S has intensity i. In this case the the PDF's $p(.)$ are estimated from the histogram of intensity values in images S and T. In the 2D case, $H(S,T)$ is simply the joint entropy and is obtained from the 2D joint histogram of the two datasets.

The MI of an image patch and the median exemplar is limited to values between zero and some maximum value. This maximum is the entropy of the median, and is an intrinsic property of MI[9]. This complicates the selection of the threshold τ for deciding whether a point is an inlier or not. The approach taken was to make the threshold proportional to the second moment of the MI values $x_{(2)}$ in the column corresponding to the median of the correspondence matrix C given in equation (4):

$$\begin{aligned} \tau &= C_{\max} - t_J \overline{C}_{(2)} = H(T) - t_J \overline{C}_{(2)} && (9) \\ \overline{C}_{(2)} &= \int_{m=1}^{M} (C_m - \overline{C})^2 \mathrm{d}m && (10) \end{aligned}$$

where t_J may be used to change the number of inliers, but is generally set to 1.

With the above distance measure and inlier threshold a feature tracker based on appearance alone may be implemented. In testing this implementation outperformed other methods for solving the template update problem, as shown in the Section 4. When tracking groups of features, the robustness could be improved further by modeling their position dynamics as well and constraining their motion relative to each other. The next section details a proposed approach to modeling structure using the SMAT framework.

3.2. Modeling Structure

To form a model of structure, the positions of each feature are concatenated together to form a $2a \times 1$ vector (assuming a features). In this case a mixture of Gaussians was assumed. It should be emphasized that a mixture of any type of distribution could have been assumed. However, if the distribution was not representative of the underlying data in local regions, more components would be required for SMAT to track successfully.

Since a Gaussian distribution is assumed in this case it is convenient to use Mahalanobis distance to compute the distance between structure vectors:

$$d_M(S) = [(S' - \overline{T}'_k)^T \mathbf{\Sigma}^{-1} (S' - \overline{T}'_k)]^{1/2}$$

where $\mathbf{\Sigma}$ is the covariance matrix of the data in the a particular component. $S' = S - S_\mu$ is the structure extracted from the current frame with the inter-point mean S_μ removed. $\overline{T}'$ is the median of component k, with its inter-point mean also removed. In practice $\mathbf{\Sigma}$ will often be non-invertible, so singular value decomposition (SVD) is applied to obtain the eigenvectors $\mathbf{A}$ and eigenvalues $\mathbf{D}$ from Σ, and the equation below is used:

$$d_M(S) = [\mathbf{A^T}.(S' - \overline{T}')] \cdot \mathbf{D}'^{-1/2} \tag{11}$$

where $\mathbf{D}'^{-1}$ is the pseudo-inverse of $\mathbf{D}$, with values close to zero being zeroed for numerical stability.

Unlike the MI distance value, the Mahalanobis distance is already normalized to the standard deviation (or second moment) of the data. Hence the threshold may be set to some real number t_M, independent of other parameters:

$$\tau = t_M = 1 \tag{12}$$

Like equation (9)), t_M was set 1. Varying t_M corresponds directly to the number of standard deviations away at which exemplars are classed as inliers.

3.3. Linking Structure and Appearance

The structure model described above may be used to improve the robustness of individual appearance based feature trackers, by constraining their positions. The positions of all the features are concatenated to form an *unconstrained* structure vector S', which is whitened to form a *constrained* vector $S'_\_$ as follows:

$$S'_\_ = \mathbf{A}. \min(\mathbf{A}^T.(S' - \overline{T}'), \mathbf{D^{-1/2}}) + \overline{T}' \tag{13}$$

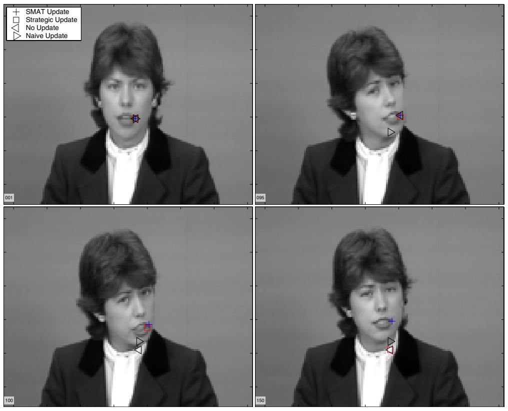

Figure 2. Positions during "Claire" sequence

For features where the disparity between S' and $S'\_$ is larger than δ pixels the update to the appearance model of that feature is blocked. A conservative approach is also taken for the formation of new components in the structure model. For a new structure component to be created less than 25% of the feature trackers must have created new components (which indicates possible failure), and all of these possible failures must be within δ pixels of their constrained position. The feature trackers are always initialized to start their feature search from their constrained position.

This conservative approach was taken, because the output from the feature trackers tends to be noisy especially when drift or misrepresentation errors start to occur. The result was a more robust tracker, that recovers from such errors, as is shown in Section 4.

4. Experiments and Results

Several experiments were undertaken to test the proposed 2 tier SMAT approach. The six sequences tested ranged in size from 60 to 500 frames and in quality from uncompressed CIF sequences to highly compressed DivX and XviD sequences with all the associated artifacts and noise. Demo's of the test sequences are obtainable from `www.ee.surrey.ac.uk/Personal/n.dowson`.

First, SMAT was implemented at the feature level only, based on appearance modeling with no additional dynamics or constraints. The results of these experiments were compared to other methods for template tracking, namely: the strategic update[7], a naive update and never updating at all. A typical sequence from these experiments is shown in Figure 2. For clarity only one feature is displayed: the left-hand corner of the newscaster's mouth. The error plots in Figure 3 details the tracking performance of each algorithm. For this and all other error plots shown the ground truth was obtained by hand labelling.

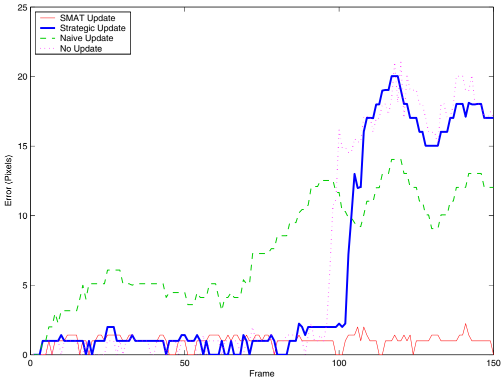

Figure 3. Error plot for "Claire" sequence

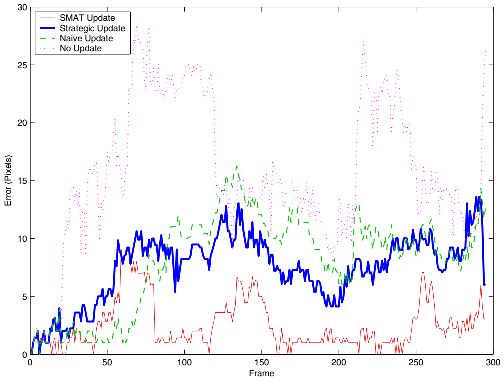

Figure 4. Error plot for "Sign" sequence

The non-updating algorithm successfully tracks the feature well, until it becomes partially occluded at Frame 95. A misrepresentation error occurs and the tracker fails . The strategic update method tracks successfully for several more frames, until it too starts to drift off at Frame 100 due to misrepresentation error. The correction between the original and new template is greater than the threshold, preventing further updates. At frame 105 failure occurs, since the feature changes its appearance too much. The naive update steadily drifts away from the original feature due to an accumulation of sub-pixel errors. Unlike its competitors, the proposed algorithm tracks the feature throughout the sequence, despite changes in the appearance of the feature and its partial occlusion during the sequence.

Figure 4 shows error plots from the "signing" (a different) sequence. (Frames from the "signing" sequence are shown in Figure 7, although the points shown do not correspond to the error plots). As shown similar results were obtained. The non-updating algorithm fails early (frame 25) due to the rapid change in feature appearance. The strategic update algorithm and the SMAT algorithm have drifted off slightly by frame 60, but by frame 90 SMAT has recovered.

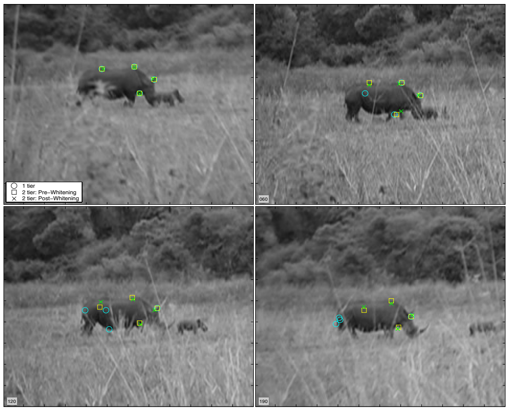

Figure 5. Positions during “Rhino” sequence.

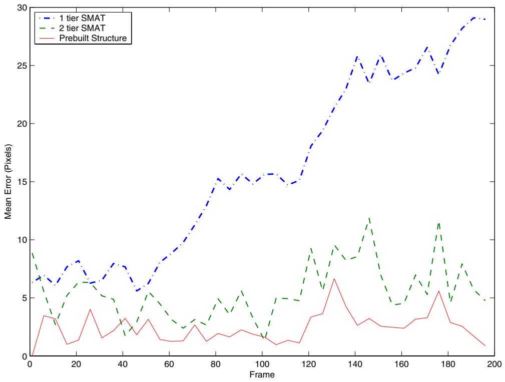

Figure 6. Error plot for “Rhino” sequence

The naive update algorithm has drifted off target by this time and continues to do so until the end of the sequence. The strategic approach partly recovers near the end due the retention of the initial appearance and because its tolerance value was increased. This increase made the strategic update approach less prone to misrepresentation failure but more prone to drift. In these and all the other sequences tested, the single tier SMAT tracker performed as well as, or better than its competitors, especially in sequences with large amounts of noise and feature deformation.

Next the two tier approach modeling both structure and appearance was tested and compared to the single tier trackers. A noisy sequence with large amounts of occlusion was chosen for this purpose, to make failure without a structure model more likely. Some excerpts from the sequence are given in Figure 5, where *pre-* and *post-whitening* indicate positions before and after whitening using the structure model and equation (13). Frame 1 shows the initial exemplar of structure and features. The single tier SMAT trackers are mostly still on target by frame 60. The exception is the tracker on the Rhino’s back, which has drifted slightly, due to the weakness of the feature. By frame 120, all the single tier trackers have failed, except for the tracker on the ear, which is a strong feature. These failures are primarily due to occlusions by the grass in the foreground. At frame 180 the 2 tier SMAT algorithm is still tracking successfully. Frames 120 and 180 also show how the structure model attempts to “pull” the features into a known shape, and how error is distributed between the features.

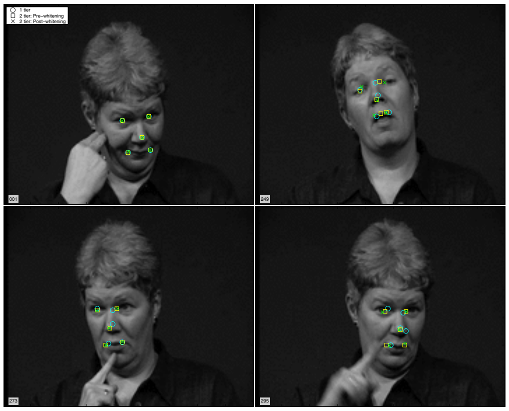

Figure 7. Positions during “Sign” sequence.

Data was ground-truthed (by hand) at five frame intervals for the “Rhino” sequence. The mean feature position error is shown in Figure 6. The performance when using a prelearned model from the ground-truth data was also plotted. The improvement for the prelearned model was substantial, since in this sequence, the structural variations were limited and the ground truth data was less noisy than the input from the appearance based feature trackers. As shown, the two tier SMAT approach had a lower and more stable mean error than single tier trackers. In all the sequences tested, using a two tier approach served to improve the robustness of the tracking algorithms. Again, this was particularly evident in the noisy sequences with large deformations.

Figure 7 shows the signing sequence, where the structure model was used to constrain feature position. Frame 1 shows the initial position. By Frame 249 some of the appearance trackers have drifted slightly, creating some bias in the structure, since the structure model distributes the error amongst the features. The feature positions are “pulled” back towards a known structure as shown in Frame 273, which allows for a better recovery by Frame 295.

4.1. Note on Parameters and Performance

In this set of experiments a learning rate of 0.05 was used, the number of exemplars per component M was limited to 60, and the number of components in each mixture model K was limited to 5. Depending on the features sizes in the sequences, the image patch size was set to either 11^2

or 13^2 pixels. To compensate, the threshold for appearance t_J was adjusted from 1.0 to 0.7 for the larger patches. Apart from these changes, to deal with the scale of features, the algorithm demonstrated consistent performance across all six sequences tested without any parameter optimization.

The proposed method operated at near real-time frame rates despite the structure model being partly implemented in Matlab. When 5 features were being tracked with a 13 × 13 kernel frames, framerates of 9fps were obtained. When no structure model was applied, framerates of 59fps were obtained on a Pentium 4 2.6GHz processor.

The computational cost of an MI based appearance tracker when using SMAT compared to the cost evaluating to the distance D_H, l times is $(k + M.X^{-1}).O(l.D_H)$, where k is a value between 1 and K (the number of templates), X is the number of pixels and M is number of exemplars in each component. With representative features k should remain somewhere between 1 and 2, so the cost is comparable to a standard template update method.

5. Conclusion

A new framework called Simultaneous Modeling and Tracking (SMAT) has been proposed for tracking objects through motion sequences. The framework is based upon a library of exemplars being built up, to which a mixture model is incrementally fitted. Any type of data may be used, whether image patches, contours or point coordinates, provided some distance measure exists as well as a method of classifying inliers and outliers. Likewise, any distribution may be used, dependent on the distance measure.

SMAT was formulated to deal with the problem of tracking an object through a motion sequence robustly, in real-time, without any *a priori* data on noisy sequences. To the authors' knowledge no other approach exists that manages that achieves all these objectives successfully.

SMAT was applied in a two tier approach to model the appearance and structure of an object during a number of motion sequences. Robust performance was obtained when tracking features individually and based on appearance alone. As such this is a novel solution to the template update problem that outperforms all other solutions of this type that were tested. Modeling structure served to further improve robustness when tracking multiple features, by making the detection of tracking failure simpler and applying some constraints to the object structure as it deforms. Different distance measures were used for each tier, namely: Mutual Information for appearance and Mahalanobis distance for the structure. The SMAT framework applied equally well at each of these levels.

Significantly, without any prior data, the proposed method performs as well as *a priori* model based tracking methods and outperforms all the existing methods with no *a priori* information. In fact, the ability of the proposed method to build up a model on the fly means that the longer it successfully tracks, the more robust it becomes. The additional computational cost of building the appearance and structure model is not much greater than that required for tracking features normally, so the proposed two tier SMAT algorithm operates at near real-time, despite being implemented partly in Matlab.

Currently features are selected manually. Whether these points are optimal for tracking (when using MI) is still an open question, which is being examined. The method is also being extended to be affine invariant, for which a multi-dimensional global optimization method will be required.

Acknowledgements

This work was made possible by funding from the CVSSP at Surrey University and an ORS scholarship provided by the Department for Education and Skills, UK. The authors gratefully acknowledge an award by the BMVA that made travel to this conference possible, and the helpful comments of the anonymous referees.

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, Mar 2004.

[2] R. Collins and Y. Liu. On-line selection of discriminative tracking features. In *Proc. of 9th IEEE Int'l Conf. on Computer Vision*, Oct 2003.

[3] N. Dowson and R. Bowden. Metric mixtures for mutual information tracking. In *Proc. 17th Int'l Conf. on Pattern Recognition*, volume 2, pages 752–756, Aug 2004.

[4] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, Oct 2003.

[5] T. Kaneko and O. Hori. Template update criterion for template matching of image sequences. In *Proc. Int'l Conf. on Pattern Recognition*, volume 2, pages 1–5, 2002.

[6] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Int'l Conf. on Artificial Intelligence*, pages 674–679, Aug 1981.

[7] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(6):810–815, Jun 2004.

[8] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: multitarget detection and tracking. In *Proc. 8th European Conf. on Computer Vision*, volume 1, pages 28–39, May 2004.

[9] C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, Jul 1948.

[10] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):747–757, Aug 2000.