N-tier Simultaneous Modelling and Tracking for Arbitrary Warps

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

Abstract

This paper presents an approach to object tracking which, given a single example of a target, learns a hierarchical constellation model of appearance and structure on the fly. The model becomes more robust over time as evidence of the variability of the object is acquired and added to the model. Tracking is performed in an optimised Lucas-Kanade type framework, using Mutual Information as a similarity metric. Several novelties are presented: an improved template update strategy using Bayes theorem, a multi-tier model topology, and a semi-automatic testing method. A critical comparison with other methods is made using exhaustive testing. In all 11 challenging test sequences were used with a mean length of 568 frames.

1 Introduction

The aim of this work is to track non-rigid objects using appearance in real-time without a pre-learned model. The tracker should be robust to noise, changes in lighting conditions, occlusions, background clutter and changes in the appearance of the object due to its non-rigidity and pose variation. Moreover, recovery from tracking failure should be possible, as should failure detection.

Extensive research has been dedicated to tracking non-rigid objects. Even within controlled environments with pre-learned models and without real-time constraints, this is a difficult problem. Some examples of successful appearance based approaches with less binding constraints, include that of Okuma *et al.* [9], who use a pre-learned model to detect and track ice-hockey players despite their small size in certain views. The WSL tracker of Jepson *et al.* [6] tracks objects without a pre-learned model in spite of occlusion by maintaining a 3 component model for each pixel in the template and shifting between the components using Expectation Maximisation (EM). However, the use of EM combined with a large model means it is too expensive for real-time use. The discriminative tracker of Collins *et al.* [2] also builds a model on the fly. The use of different combinations of RGB channels to form multiple features results in a robust tracker. However, the method assumes the foreground to be a rectangle centred in a rectangular background region, which makes tracking irregularly shaped objects that vary in pose difficult. Also, the use of many features (125) makes real-time tracking of large objects difficult.

The simpler approach of using Lucas-Kanade (LK) type tracking [7] has often proven to be effective despite using only a very simple model (a single template) to track. This is particularly so in its modern manifestation such as the strategic update method of Matthews *et al.* [8]. Moreover, modern formulations of registration algorithms use quasi-Newton methods and many of the more-expensive operations are pre-computed, thereby allowing above real-time performance. Notable examples are the Inverse Compositional approach of Baker and Matthews [1] and the parametric models of Hager and Belhumeur [5]. However these methods tend to suffer from drift.

Clearly a multi-faceted approach is required to achieve the aims of the problem statement. Our approach has its distant roots in the LK tracking method but uses multiple component models. Pixel level models have been avoided for the sake of speed. Instead, multi-tier structure-appearance models have been used, which are somewhat richer than single scale appearance templates.

In many tracking applications, a template is aligned to the current image by minimising some appearance similarity metric. In §2 the choice of similarity metric is discussed, as are several methods for modelling the appearance of a feature. The approach here uses a Bayesian method for clustering appearance exemplars together. Next a multi-tier extension of the SMAT algorithm [3] is presented in §3 where structure and appearance models are sandwiched together. Two hierarchical topologies are considered. Exhaustive tests were performed using a semi-automated testing method discussed in §4. The results are given in §5 before the paper concludes in 6.

2 Template Tracking

To begin, the problem of tracking is formalised as an optimisation problem. Given a motion sequence of N_m frames, let $S_m(\mathbf{x})$ represent the intensity at position $\mathbf{x}$ in frame S_m. A warp $\mathbf{w}$, with parameters $\mathbf{v}$, is sought that minimises some distance function d between a template T_n and S_m:

$$\mathbf{v}_{opt} = \arg_{n,\mathbf{v}} \min \quad d[T_n(\mathbf{x}), S_m(\mathbf{w}(\mathbf{x},\mathbf{v}))] \tag{1}$$

with d measuring the similarity between T_n and the region in S_m it overlaps. Several templates indexed by n may exist, hence the minimisation over n as well in (1). This general entails multiple minimisations over $\mathbf{v}$ for several values of n.

2.1 Distance functions and warps

Several constraints affect the choice of d. The evaluation of d must be fast, it should be relatively robust to outlier pixels and the basin of convergence should have sufficient gradient to allow rapid convergence without giving ambiguous results. The ability to obtain a Hessian rapidly is also a consideration. Mutual Information (MI) suits these specifications admirably, since an analytic derivative for it has recently become available [4]. Hence MI was the first choice for this work. The widely used Sum of Square Differences (SSD) also has good characteristics and is presented here as well. Distance functions that increase with greater similarity may be trivially converted for a minimisation framework by multiplying by -1.

SSD entails summing the square of the differences between each pixel in T and the corresponding pixel in S_m:

$$d_{ssd}(\mathbf{v}) = N_{\mathbf{x}}^{-1} \sum_{\forall \mathbf{x} \in T} [T(\mathbf{x}) - S_m(\mathbf{w}(\mathbf{x},\mathbf{v}))]^2 \tag{2}$$

where $N_{\mathbf{x}}$ is the number of pixels in T. SSD can be prone to outliers and assumes that the intensities in T and S_m are linearly related.

In contrast, MI treats each image as a random sample of intensities, measuring the information shared between the two sets of samples [10]. This is estimated from the joint-histogram of intensities h_{st}:

$$d_{mi} = -N_{st}^{-1} \sum_{\forall s,t} h_{st}(s,t) \log(\frac{h_{st}(s,t)}{h_s(s)h_t(t)}) \tag{3}$$

where the intensities of the two images S_m and T are $s \in [1;N_s]$ and $t \in [1;N_t]$ respectively. $N_{st} = N_s N_t$ is the number of bins in the joint-histogram. The single histograms are obtained from the joint using row and column sums: $h_s = \sum_t h_{st}$ and $h_t = \sum_s h_{st}$.

MI is only slightly more expensive than SSD to obtain: $O(N_{\mathbf{x}} + N_{st})$ and $O(N_{\mathbf{x}})$ respectively. The Hessians also have comparable costs, and we obtained speeds similar to those reported by Matthews *et al.* for SSD [1] for both metrics. Since a Hessian was cheaply available, the Levenberg-Marquardt method with a sparse pre-search step was used for minimisation of the similarity metrics.

Four types of warps were considered in this work: translation, Euclidean, similarity and affine warps. These respectively have 2,3,4 and 6 degrees of freedom (DoF). Using higher order warps does not necessarily add robustness, since for small image patches the problem becomes under-constrained.

2.2 One and two template models

Initially, only one template from the first frame exists, *i.e.* $n \in \{1\}$. After optimisation for $\mathbf{v}$ in each successive frame the matching region, X_m, may be extracted as a new exemplar for possible use as a template. The collection of templates forms an *appearance model.*

Perhaps the simplest model is one that is never updated, consisting only of the single template extracted from the first frame, *i.e.* $T_1 = X_0$, as shown in Fig. 1a. We call this the *no update* model, and will only work as long as the template closely resembles the feature being tracked. Typically the resemblance is fleeting, and tracking fails due to a mismatch.

One alternative is the *naive update* model, where the template is updated after every frame, *i.e.*: $T_1 = X_m = S_m(\mathbf{w}(\mathbf{x}, \mathbf{v}_{opt}))$, as shown in Fig. 1b. Sub-pixel errors inherent to each match are stored in each update and these errors gradually accumulate resulting in the template drifting off the feature.

A recent *strategic update* approach that trades off mis-match error and drift has recently been proposed by Matthews *et al.* [8], which is a simple and effective extension of the naive update. The updated template is used for an initial alignment, but the template from the first frame is then used in an error correction phase after alignment using the updating template. If the size of the correction is too large, the algorithm acts conservatively by preventing the updating template from being updated from the current frame n. The strategic update model is illustrated in Fig. 1c.

2.3 Simultaneous Modelling and Tracking

The inclusion of a second template in the strategic update model improves tracking results substantially. Simultaneous Modelling and Tracking (SMAT) extends this by storing all

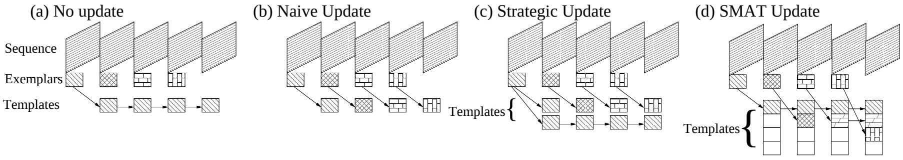

Figure 1: Building models using various update strategies

the exemplars extracted from each frame, selecting templates from amongst the exemplars to solve (1) and locate the feature, as illustrated in Fig. 1d for four templates. To limit the computational cost, the collection of exemplars is clustered on the fly, with each cluster, C_n, being represented by its median, μ_n. Only the cluster medians are ever used as templates, limiting n in (1) to $n \in [1;N_n]$, where N_n is the number of clusters. The median rather than the mean is used to avoid the pixel blurring inherent to the averaging of multiple intensity values.

A weighting, w_n, is also associated with each cluster, which represents the estimated *a priori* likelihood of the cluster resembling the current appearance of the feature being tracked. So the propability of the model matching the current feature appearance may be treated as a sum of likelihoods: $P = \sum_{n=1}^{N_n} w_n \frac{p(fg|d(X_m,\mu_n))}{p(bg|d(X_m,\mu_n))}$. The weights satisfy the constraint, $\sum_n w_n = 1$.

Two examples of such models are shown in Fig. 2, with the exemplars represented as dots in 2D analog of appearance space, where the distance between two appearances is represented by d_{MI}. Using Fig. 2, the construction of the model, P, is now described.

In each new frame, S_{m+1}, the tracked feature will change in appearance as conditions such as pose and lighting vary, with a corresponding shift in the feature's position in appearance space. In the examples of Fig. 2a and b, the new positions are indicated by stars. Updates to single template models serve only to move the position of the model in appereance space, rather than properly describe the occupation of X. Ideally each new exemplar is added to the nearest cluster, otherwise clusters become overly inflated and unspecific to the feature. Moreover, some exemplars can result from tracking failure and not be representive of the true feature. Either way mis-representation error can ensue, so a method to determine membership to a particular cluster is required.

Membership is determined by the ratio of the distance between the new exemplar and a cluster median indicating a foreground or background appearance:

$$\frac{p(fg|d(X_m,\mu_n),\nu_{fg})}{p(bg|d(X_m,\mu_n),\nu_{bg})} \tag{4}$$

The normal distribution of foreground distances, ν_{fg} is obtained from the distances between the median and each cluster member. The normal distribution of background distances, ν_{bg} is obtained by calculating the variance of d values between the median and all exemplars offset from v_{opt} by one pixel. These values represent positions possibly within the basin of convergence but not at the minimum and are hence most likely to cause confusion. This is an improvement on [3], which required a pre-selected bound factor. The bounds of the cluster produced by this ratio are indicated by the ellipses in Fig. 2.

In general newly created clusters are less reliable than previously established ones, since they have fewer samples and may be the result of an earlier tracking failure. To

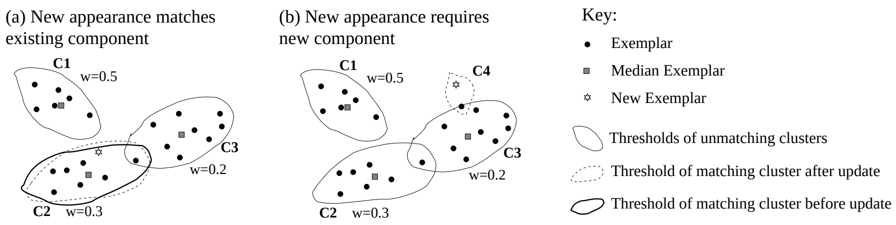
(a) New appearance matches existing component
C1
w=0.5
C3
w=0.2
C2
w=0.3
(b) New appearance requires new component
C1
w=0.5
C4
C3
w=0.2
C2
w=0.3
Key:
Exemplar
Median Exemplar
New Exemplar
Thresholds of unmatching clusters
Threshold of matching cluster after update
Threshold of matching cluster before update

Figure 2: Diagram of Simultaneous Modelling and Tracking, showing 3 clusters of exemplars (dots) in appearance space. The tracked feature has a new appearance (star), which in (a) is within the bounds of Cluster 2 after minimisation, which is appropriately updated. In (b) no cluster achieves a successful match, so a new cluster is created.

model the effects of increasing relevance and reliability, the weights of each cluster are updated after selecting the matching cluster as follows:

$$w_n \leftarrow \begin{cases} \frac{w_n+\alpha}{1+\alpha} & n = n_{match} \\ \frac{w_n}{1+\alpha} & n \neq n_{match} \end{cases} \tag{5}$$

where α is a learning parameter. This allows frequently successful clusters to dominate in the model, but allows obsolete clusters to be gradually removed.

To reduce computational expense further, a greedy approach is used to explore the clusters. Alignment using (1) is attempted starting with the highest weighted cluster. If a match within the membership boundary (4) is obtained, as in Fig. 2a, the new examplar of appearance is added to the current cluster with appropriate updates to the weightings, median, v_{fg} and v_{bg}. Otherwise alignment is attempted using the next highest weighted cluster and so on.

If none of the existing clusters achieves a successful match, as in Fig 2b, the most recently extracted exemplar is used as a template: as this is the most likely to resemble the feature in its current form. A new cluster is then formed using the template, X_{m-1}, and the newly extracted region X_m, with an initial weight of zero. The number of clusters is limited. If this limit is exceeded the lowest weighted existing cluster is replaced.

3 N-tier hierarchical models

In the case of large non-rigid objects, using the single appearance model of §2 requires large numbers of clusters to represent the variation in object appearance. Not only is this computationally expensive, but it is poorly representative as each cluster will have few exemplars due to the highly variable appearance. Higher order warp parameterisations can reduce the number of clusters required, but these are still somewhat limited in their descriptiveness. An alternative is a bag-of-features approach: where the object consists of several smaller features that are independently tracked and modelled. However, small features are more prone to failure, as they consist of fewer pixels and complete occlusions are more likely. Another alternative is to use a hierarchical approach, to track a larger region and use this to pre-align smaller child features, but it is difficult to tell if this induces tracking failure.

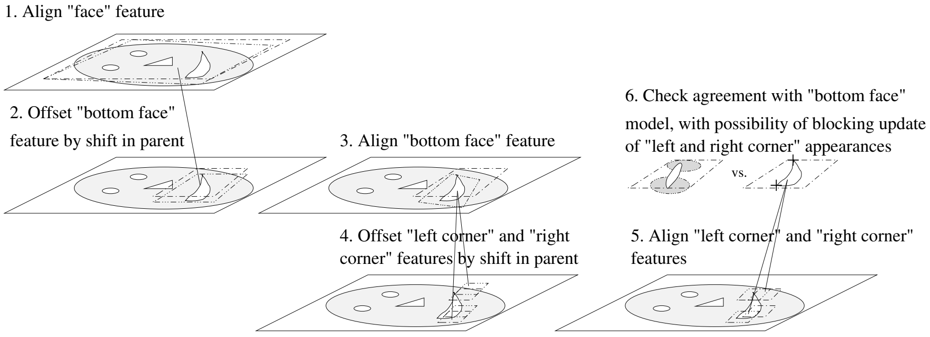
1. Align "face" feature
2. Offset "bottom face" feature by shift in parent
3. Align "bottom face" feature
6. Check agreement with "bottom face" model, with possibility of blocking update of "left and right corner" appearances
vs.
4. Offset "left corner" and "right corner" features by shift in parent
5. Align "left corner" and "right corner" features

Figure 3: Illustration of the structure of a 3 tier shape model and how the shape model is used to recursively align and update the appearance models at each tier.

N-tier SMAT overcomes this by combining all these ideas and using the relative positions of parents and children in a tree structure to build up a model of *shape*. An example of a 3 tier structure used to model a face is shown in Fig. 3. Using the shape model, a likelihood may be attached to the position of the child features, and hence to the probability of tracking failure. This allows tracking failures to be corrected and possible erroneous feature updates to be blocked. N-tier SMAT thereby combines the advantages of tracking large features (robust to noise, occlusions and large motions) with those of tracking multiple small features (rich object description and accurate modelling of articulated structures). The shape model is also treated as a collection of weighted clusters: $p_{shp} = \sum_n w_n \nu(\mu_n, \sigma_n))$.

A recursive alignment is applied starting at the root-parent feature, in a manner illustrated in Fig. 3 for a human face. First the largest "head" feature is aligned using the SMAT process discussed in §2.3. Second, the children features of the head (the "forehead" and "mouth region") are offset from their positions in the previous frame by the interframe shift of their parent. Third, a SMAT alignment is applied to each child as per §2.3, *except* the update is held in abeyance. Steps two and three are repeated for the "mouth region" children: "left mouth corner" and "right mouth corner", again with their respective SMAT updates held in abeyance.

In step six the positions of the "left mouth corner" and "right mouth corner" are concatenated into a vector describing the "mouth region" shape, which is fitted to each cluster of the shape model. If the current shape fits an existing cluster, the shape model is appropriately updated and the appearance updates for the child features that were held in abeyance are applied. If the current shape does not fit the existing model, a new cluster is generated, but the appearance updates of the children are blocked. In addition, the current shape is corrected to fit the nearest cluster, and the corrected positions are applied to the children features. Step six is applied again at the level of the parent feature.

The above shape update process is very similar to the SMAT appearance modelling process of §2.3, with some slight differences. Firstly, a PCA model is fitted to each cluster of shape exemplars, and Mahalanobis distance is use to measure the similarity of structures. Secondly, "background" and "foreground" shape scores are not available as they are for appearance, so a hard bound is required to establish cluster membership. We used a bound of 2σ, which worked well for all the sequences tested.

The shape model consists of children's positions *relative to their parent*, despite the use of different warps for each object, so a central feature or average position is not required for a coordinate system base as it was in [3]. This is an important advantage, since the use of a central feature makes tracking dependent on that feature's reliability. Similarly, use of a central position means that one outlying point in the shape can deform the entire shape and induce tracking failure in the remaining features.

The above structure model is referred to as a *thick* structure model. A similar hierarchical alignment is possible that does not explicitly model the structure of the object, but the relative shift of parent objects is applied to the initial positions of child objects. This still implicitly models the dependence of child objects on their parents. We refer to this as the *thin* structure model. Both methods yielded significant improvements over independently tracked features. However, the use of a thick structure model allows more frequent recovery from tracking failures in small child features.

4 Testing Methodology

Testing tracking applications typically involves the researcher hand selecting the positions of features in each frame. Such an operation is time consuming, so testing using the large numbers of long sequences necessary for comprehensive evaluation is impractical. Moreover, hand selected positions are not necessarily a true reflection of tracking performance, due to a human's use of context, which is unavailable to trackers.

The *zig test* can overcome these difficulties. This uses the idea that the only position known with absolute certainty from the tracker's point of view is that of the selected feature in the first frame. Any comparison against position in a frame is absolute as well. The zig test uses this property by tracking a frame through a sequence from frame 0 to frame N_m and then by reversing the sequence back to frame 0. We refer to the second instance of frame 0 as frame $2N_m$. The overlap between the X_{2N_m} and X_0 gives an indication of how far the tracker has drifted from its original target.

Tracking error generally increases with sequence length, however re-acquistion after failure can occur in the reverse direction, giving a false "success". Hence hand selected ground truth positions are used at $\frac{1}{10N_m}$ intervals to flag failures automatically when the tracker drifts too far from the feature. Although not fully automatic, this approach still makes testing less laborious. Tracking performance was gauged from the area of overlap between X_0 and X_{2N_m}: $\frac{A(X_0 \cap X_{2N_m})}{\max(A(X_0), A(X_{2N_m}))}$.

Eleven long (568 frames on average) and challenging sequences of various subjects were obtained. Eight sequences were used to track single features and three sequences were used to track 3-tier structure models of features. Objects tracked included humans in sporting events, human faces with changing expressions, gesturing human hands, pedestrians, and wild animals in Kruger National Park. Backgrounds were in some cases highly variable and cluttered, including ripples in a swimming pool and moving bushes. Likewise some sequences had lighting conditions that varied drastically. The first frame of each test sequence is shown in Fig. 4.

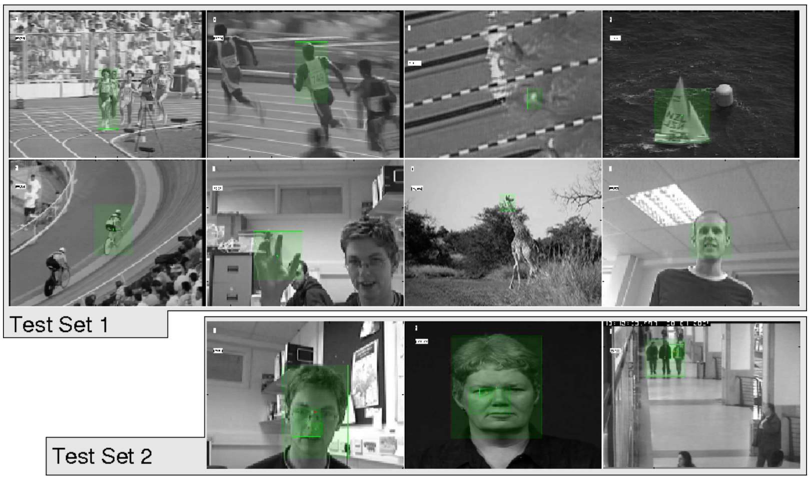

Figure 4: Test sequences used. Set 1 was used to test single feature tracks. Set 2 was used to track 3-tiers of object.

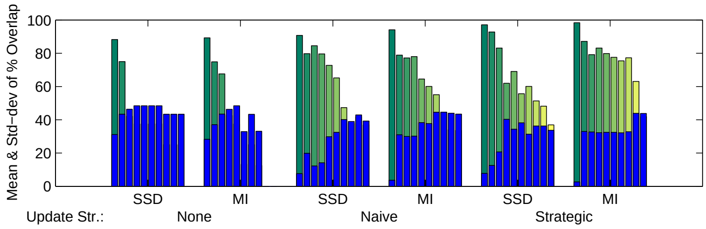

Figure 5: Comparison of SSD and MI metrics for the three one/two template models: No update, Naive update and Strategic update. Warp used: translation. Bar groups show increasing sequence length.

5 Results

The test framework of §4 was used to compare the methods in this paper to existing approaches. In particular the tests made specific comparisons between: MI and SSD, appearance models of different complexities, warps with various Degrees of Freedom (DoF) and structure models of various kinds. Ten zig tests were performed for each test sequence, ranging in length from 10% to 100% of the sequence length. The mean percentage overlap between the rectangles at the beginning and end the sequences is shown in Fig. 5 to 7 (green/light bars). The standard deviation is also shown (blue/dark bars) to express variability of the results, because the mean was taken across the test sequences. Bar groups show performance as sequence length is increased in 10% intervals from 10% to 100%.

Test-set 1 was used to compare SSD & MI, using the three “simple” template models, namely: no update, naive update and strategic update. The 2 DoF translation warp was used for these tests. Fig. 5 unsurprisingly shows that the strategic update model outperforms naive update which outperforms no update. The strategic update represents

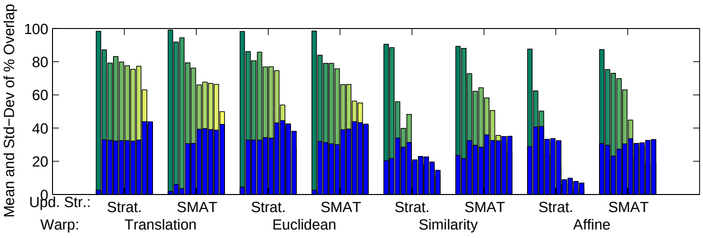

Figure 6: Comparison of performance using four warp types: Translation (2DoF), Euclidean (3DoF), Similarity (4DoF) and Affine (6DoF) for Strategic and SMAT update strategies. Bar groups are for increasing sequence length. Metric: MI.

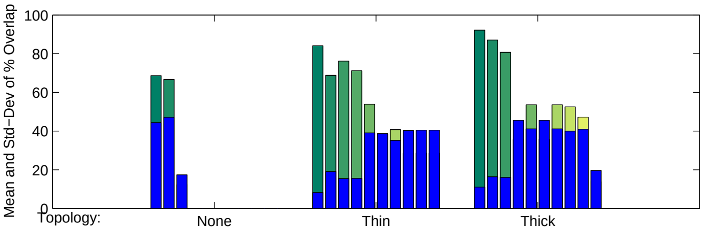

Figure 7: Comparison of performance for four structure models: none (*i.e.* independent features), thin, thick (SMAT). Bar groups are for increasing sequence length. Appearance Model: SMAT. Metric: MI. Warp: Similarity + Translation + Translation

the variability of the data better and hence performance degrades more gracefully as well. MI outperforms SSD in most cases, but appears to be more prone to mis-match when the appearance model is not updated.

Test-set 1 was used to examine whether the use of higher order warps improves performance. The results are shown in Fig. 6. In most cases SMAT outperforms the strategic update model by some margin, particularly for higher order warps and towards the end of sequences. Surprisingly, performance degrades as the DoF increase, however this could be for a number of reasons: the method of parameterisation is fairly sensitive, since (respectively) for the affine parameters, similarity, and Euclidean warps, the rotation matrix, scale and angle are explicitly encoded. The greater number of DoF allows more opportunities for drift and local minima. Drift probably accounts for the increasingly graceful degradation as DoF increases.

Test-set 2 was used to examine the effects of modelling structure models in multiple tiers. In each test three tiers with one feature per tier were used. The similarity transform was used for the top tier, and translation for the second and third tiers to avoid over parameterisation. The results in Fig. 7 show the performance for the lowest tier *i.e.* the feature most likely to fail. As shown, even using a thin structure (*i.e.* treating the problem as a multi-scale registration) improves performance significantly and explicitly modelling the structure improves performance further. In our experience thick model structures should be used with care, since they can also drag features off their correct position as well. This appears to occur in Group 3 of Fig. 7, but re-acquisition occurs later. The re-acquisition is probably due to the structure model as well.

6 Conclusion

An N-tier Simultaneous Modelling and Tracking algorithm has been presented, which learns a hierarchical model of appearance and structure on the fly. Image alignment over multiple warps was performed by minimising (negative) Mutual Information using the Levenberg-Marquardt algorithm. Multiple layers of appearance models were sandwiched together using structure models. Two types of structure model were proposed: the thin (implicit shape) model and the thick (explicit shape) model. Testing was performed using the semi-automated zig-zag test. This allowed exhaustive testing on 11 challenging sequences with a mean length of 568 frames. The use of structure models gave superior results to independent tracking, however explicitly modelled structure did not always improve upon implicitly modelled structure. Comparisons between SSD and MI similarity metrics were made and MI generally slightly outperformed SSD. Of the different appearance models, SMAT performed the best due to its use of multiple clusters. Translation, Euclidean, Similarity and Affine warps were also compared, and in general higher order warps failed sooner due to over-parameterisation. The matlab code and test harness is publicly available on the Internet at: `www.ee.surrey.ac.uk/personal/n.dowson`. In future features and articulated model structures will automatically be detected.

References

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

[2] R. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Trans. PAttern Analysis and Machine Intelligence*, 27(10):1631–1643, October 2005.

[3] N. Dowson and R. Bowden. Simultaneous modelling and tracking (smat) of feature sets. volume 2, pages 99–105, San Diego, CA, USA, June 2005.

[4] N. Dowson and R. Bowden. A unifying framework for mutual information methods for use in non-linear optimisation. In A. Leonardis, H. Bischof, and A. Prinz, editors, *Proc. 9th European Conf. on Computer Vision*, volume 1, pages 365–378, Graz, Austria, May 2006.

[5] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, October 1998.

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

[7] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. 7th Int'l Joint Conf. on Artificial Intelligence*, pages 674–679, Vancouver, Canada, August 1981.

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

[9] 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, Prague, May 2004.

[10] C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.