Linear Predictors for Fast Simultaneous Modeling and Tracking

Liam Ellis[1], Nicholas Dowson[1], Jiri Matas[2], Richard Bowden[1]
[1]:Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, UK
[2]:Center for Machine Perception Czech Technical University, Prague, Czech Republic
{L.Ellis N.Dowson R.Bowden}@Surrey.ac.uk, matas@cmp.felk.cvut.cz

Abstract

An approach for fast tracking of arbitrary image features with no prior model and no offline learning stage is presented. Fast tracking is achieved using banks of linear displacement predictors learnt online. A multi-modal appearance model is also learnt on-the-fly that facilitates the selection of subsets of predictors suitable for prediction in the next frame. The approach is demonstrated in real-time on a number of challenging video sequences and experimentally compared to other simultaneous modeling and tracking approaches with favourable results.

1. Introduction

This work seeks to develop an approach to visual tracking that removes the need for hard coding and offline learning of either the target appearance variations or motion models by learning all models on-the-fly. For a visual tracking approach to be useful it must operate at high frame rates, track fast moving objects and be adaptable to variations in appearance brought about by occlusions or changes in pose and lighting. This is achieved by employing a novel, flexible and adaptive object representation for efficient tracking comprised of sets of spatially localised linear displacement predictors bound to various modes of a multi component template based appearance model.

The main contributions of this work are: first the use of fast and efficient displacement predictors within a simultaneous modeling and tracking framework and, second a novel, flexible and adaptive object representation for efficient tracking. Furthermore, by continually evaluating and adapting the set of linear predictors based on their online performance, poorly performing linear predictors can quickly be replaced thus removing the need for complex or costly predictor placement and/or learning strategies. The tracker is shown to compare favorably to state of the art simultaneous modeling and tracking approaches in two challenging video sequences.

2. Background

Object tracking is an expansive area of research with numerous approaches proposed, each able to cope with different contexts and user requirements. For a recent review of state-of-the-art tracking methods the reader is referred to [13]. Real time simultaneous modeling and tracking is achieved here by online learning of fast displacement predictors tied to an adaptable multi-modal appearance model. Relevant background to appearance models and displacement predictors in tracking is now presented.

2.1. Appearance models for tracking

Tracking approaches typically employ appearance models in order to optimise warp parameters (e.g. translation or affine) according to some criterion function. Linear predictor trackers typically rely upon hard coded models of object geometry [10, 9]. This requires significant effort in hand crafting the models and like simple template models [8, 1, 11], are susceptible to drift and fail if the target appearance changes sufficiently. Systems that use a priori data to build the model [2] or train the tracker offline [12] can be more robust to appearance changes but still suffer when confronted with appearance changes not represented in the training data. Incremental appearance models built online such as the WSL tracker of Jepson et al. [5] have shown increased robustness by adapting the model to variations encountered during tracking, but the overhead of maintaining and updating the model can prevent real-time operation.

Two recent approaches that achieve real-time tracking and have adopted an entirely online learning paradigm are the discriminative tracker [4] that uses an online boosting algorithm to learn a discriminative appearance model on the fly and Dowson's SMAT algorithm.

Dowson et al. have shown the benefits of online learning of a multiple component appearance model when employing alignment-based tracking for Simultaneous Modeling And Tracking, SMAT [3]. The appearance model presented here is similar to the SMAT multi component model in that it partitions the appearance space into components

each represented by a median exemplar. Building a multi-modal exemplar based appearance model in this way naturally segments the appearance space and hence facilitates the assignment of efficient displacement predictors to specific aspects of the target object.

2.2. Displacement prediction

Alignment based tracking approaches obtain the warp parameters by optimising the registration between the appearance model and a region of the input image according to some similarity function (e.g. L_2 norm, normalised correlation, Mutual Information). Optimisation is often achieved using a gradient decent or Newton method and hence assume the presence of a locally convex similarity function surface with a minima at the optimal warp position. A limiting factor for such methods is the range or size of the basin of convergence. Trackers with low range require low inter-frame displacements to operate effectively and hence must either operate at high frame rates (with high computational cost) or only track slow moving objects.

Cootes et al. comment that a similar optimisation task is solved at each frame when optimising Active Appearance Model (AAM) parameters [2] by minimising the magnitude of an image intensity difference vector. Therefore a method for pre-learning a mapping between the image intensity difference vector and the error (or required correction) in AAM model parameters is proposed. They propose simple linear mappings that allow the prediction of motion parameters as a linear function of image intensities. Jurie et al. employed similar *linear predictor* (LP) functions to tack rigid objects [6].

Williams et al. presented a sparse probabilistic tracker for real-time tracking that uses an RVM to classify motion directly from a vectorised image patch. The RVM forms a regression between erroneous images and the errors that generated them. The recent work of Matas et al. [10], uses simpler linear regression for displacement prediction, similar to the linear predictor functions in [6] and [2].

A key issue for LP trackers is the selection of its reference point, i.e. its location in the image. In the work of Marchand et al. predictors are placed at regions of high intensity gradient [9] but Matas et al. have shown that a low predictor error does not necessarily coincide with high image intensity gradients [10]. In order to increase efficiency of the predictors a subset of pixels from the template can be selected as *support pixels* used for prediction. Matas et al. present a comparison of various methods for learning predictor support, including randomised sampling and normalised reprojection, is presented and it is found that randomised sampling is optimal [10]. The approach presented here avoids the need for costly reference point and support selection strategies by evaluating the real performance of a predictor over time and allowing poor performers to be replaced as opposed to minimising a learning error offline. Unlike the approach presented here, each of the displacement prediction trackers detailed in [10, 12, 9] require either an offline learning stage or the construction of a hand coded model or both.

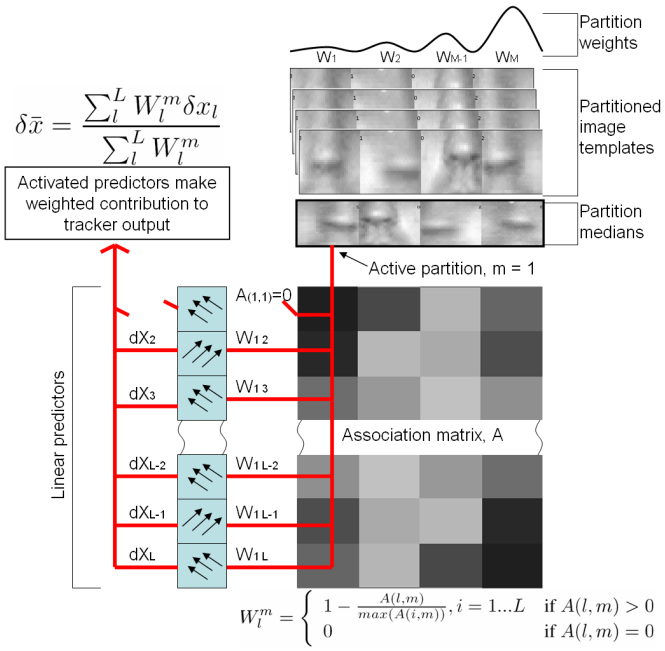

Figure 1. Activation of linear predictors associated to active appearance model component through association matrix.

3. Methodology

The proposed approach tracks a target object by online learning of constellations of spatially localised linear displacement predictors and associating them to aspect specific components of a multi-modal template based appearance model. Figure 1 illustrates the approach when applied to tracking a human nose. The approach requires no offline learning stage or hand coded models and only requires that the location of the target be given in the first frame. The appearance model, initially just the image patch at the specified location in the first frame, is learnt on-the-fly during tracking. In order to capture as much information about target appearance as possible, an image patch or template is drawn from the tracked target position in every frame. These templates are clustered online into multiple components that represent different views or aspects of the target. The appearance model is illustrated in figure 1 by the partitioned image templates, medians and component weights. Also learnt online is a set of linear mappings that predict motion parameters from image intensity difference vectors. These predictors are each associated to one or more of the appearance model components through the association matrix as illustrated in figure 1. The component of the appear-

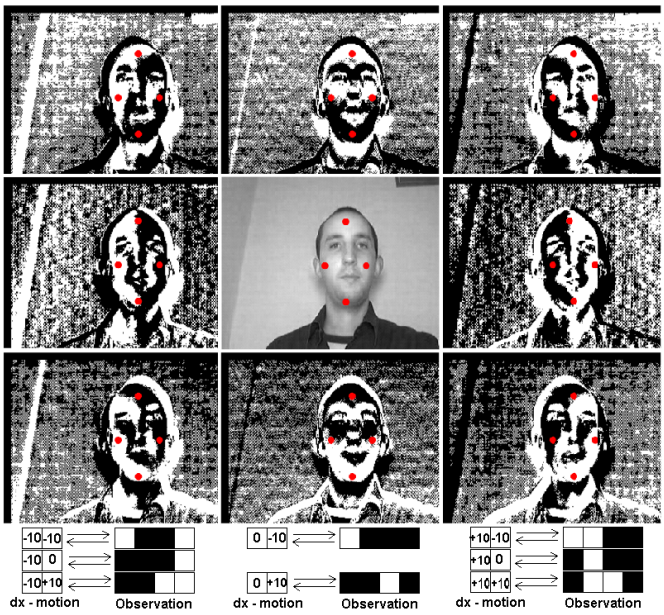

Figure 2. Intensity difference images for eight translations. Four support pixel locations illustrate the predictive potential of the difference image. The input image is in the center. All images to the left/right of the input have been translated left/right by 10 pixels. Those images above/below the input have been translated by 10 pixels up/down. Under the images, the motion and support set vectors are illustrated.

ance model that best represents the targets current appearance is found in each frame and thus activates the appropriate subset of predictors for tracking in the next frame. The performance of each predictor is continually evaluated over time and new predictors are learnt every frame to replace the worst performers. Furthermore, the contribution of a predictor to tracker output is weighted according to its online performance.

The following three subsections detail the prediction functions, appearance modeling and overall tracking method employed.

3.1. Linear predictor tracker

The linear predictor consists of three simple data structures; a $(2 \times k)$ matrix, **S**, of randomly selected translations from the linear predictor reference point that denotes the relative position of the linear predictor support pixels; a k-vector, **q**, of image intensities at the support pixel locations in the training image; and a $(2 \times k)$ matrix, **H**, that forms a linear mapping $\Re^k \rightarrow \Re^2$ from image intensity differences, **d**, to changes in warp parameters, δ**x**. A prediction can be made by first computing a k-vector, **d**, of image intensity difference between **q** and the support pixel intensities from the new input image, **P**, using the state vector, **x**, from the last frame, Eq. 1. The state vector, **x**, is the 2D position of the predictor, allowing for translation predictions. Translation is sufficient as the multi-modal appearance model copes with affine deformations of the image templates [3].

$$d_i = P_{(x^{t-1}+S_i)} - q_i, i = 1...k \tag{1}$$

Vector **d** is then multiplied by matrix **H** to obtain a motion prediction δ**x**, see Eq. 2. This efficient prediction only requires k subtractions and a single matrix multiplication, the cost of which is proportional to k.

$$\delta \mathbf{x} = \mathbf{H}\mathbf{d} \tag{2}$$

In order to learn the linear mapping, **H**, training examples of $\{\delta \mathbf{x}_i, \mathbf{d}_i\}$ pairs, (i ϵ [1,N]) are required. These are obtained from a single training image by applying synthetic warps to the training image and subtracting the deformed image from the original. For efficiency the warp and difference computation is only performed at the support pixel locations but the result of performing this operation on the entire image is illustrated in figure 2 for eight different translation warps. Also marked on the figure are four possible locations for a support pixel. It can be seen from this illustrative result that the intensities at the support pixel locations provide a good indication of translation displacements.

Linear predictor reference points are selected at random from within a predefined range R of the object center and support pixel locations are randomly selected from within a range r of the predictors reference point. The next step in learning the linear mapping **H** is to collect the training data, $\{\delta \mathbf{x}_i, \mathbf{d}_i\}$ into matrices **X**, $(2 \times N)$, and **D** $(N \times X\ k)$ where N is the number of training examples. The least squares solution, see Eq. 3, is then the linear mapping matrix **H**.

$$H = XD^+ = XD^T(DD^T)^{-1} \tag{3}$$

The parameter R determines the range around the target center that predictors are placed, it is set according to the size of the initial template. The parameter, r, defines the range from the reference point within which support pixels are selected as well as the range of synthetic displacements used for learning the predictor. Large r increases the maximum inter frame displacement at the expense of alignment accuracy. Range r is set to 30 to allow maximum of 30 pixel interframe displacement. The predictor complexity, k, models the trade off between speed of prediction and accuracy. N does not effect prediction speeds but instead parameterises a trade off between predictor learning speeds and accuracy. In all the experiments N=150 and k=100 give sufficient accuracy whilst not jeopardising the goal of real-time tracking.

3.2. Online multi-modal appearance modeling

The proposed appearance model is similar to that used in SMAT [3]. If a single template appearance of an object is

Figure 3. Four modes of the learnt appearance model for head tracking represented by medians. Also shown is the state of the tracker. Top: The head has rotated and translated and various modes of rotation have been separated out by the model allowing view specific predictors to be learnt for each aspect of the head. Bottom: Occlusion introduces a new appearance cluster but after occlusion is over an existing mode of the appearance model with all its associated predictors is reactivated. The black mark indicates which component to be used next. The left most images show the tracker output and the linear predictor placements, white predictors are currently active and black ones are inactive.

considered as one point on the appearance-space manifold, the manifold can be represented by storing all templates $\{G^0...G^t\}$ drawn from all frames $\{F^0...F^t\}$. A probabilistic modal of appearance, $P(G^t|F^t...F^0)$ is constructed incrementally by partitioning templates into components. The model is represented as a weighted mixture of M components as in Eq. 4 where η^m represents a component distribution modeled by the median template μ^m and the distance threshold τ^m.

$$P(G^t) = \sum_{m=1}^{M} w^m \eta^m(\mu^m, \tau^m) \tag{4}$$

This partitioning of the appearance space identifies different views or aspects of the target and facilitates the use of view specific displacement predictors as described in section 3.3. Two examples of the resulting appearance model when applied to head tracking are shown in figure 3.

Each of the M partitions of the appearance manifold are represented by: a group of templates, the median template μ^m, a threshold τ^m, and a weighting w^m. Use of the median rather than the mean avoids pixel blurring caused by the averaging of multiple intensity values. Weight w^m represents the estimated a priori likelihood that the m^{th} partition best resembles the current appearance of the target. During tracking, a template is drawn from the new frame at the location determined by the linear predictors. To identify the best matching partition to the new template, a greedy search is performed starting with the partition with the highest weight and terminating when a partition is found whose L_2 norm distance to the image patch is less than the threshold τ. The input image patch is then added to partition m and the median, threshold, τ^m, and weight, w^m, are updated. See Eq. 5 for the component weight update strategy. If no match is made, a new component is created with the new template and the template from the previous frame. The learning rate, α, sets the rate at which component rankings change and is set to α=0.2 which was found through experimentation.

$$w^m = \begin{cases} \frac{w_m+\alpha}{1+\alpha} & \text{if } m = m_{match}; \\ \frac{w_m}{1+\alpha} & \text{if } m \neq m_{match}. \end{cases} \tag{5}$$

To facilitate the efficient update of an appearance model component, a matrix $\mathbf{T}^m$ maintains the L_2 norm distances between each template in component m. Adding a new template to the component then requires only the computation of a single row of $\mathbf{T}^m$ i.e. the distances between the new template and each other template. The median template index can then be calculated using Eq. 6 and the component threshold τ^m can be computed using Eq. 7 which assumes a Gaussian distribution of distances and sets the threshold to three standard deviations of the distribution. The dimensions of $\mathbf{T}^m$ depend on the number, n, of templates in the model but can be limited to bound memory requirements and computational complexity. In practice, new templates replace the worst template from the component. It is also possible to limit the number of components, M. When creating a new component, if M has reached its maximum, the new component replaces the worst existing component identified by the lowest weight w^m. For all the experiments presented here a maximum of n=60 templates are maintained in each of a maximum of M=4 components of the model. This is found to be sufficient to model a stable distribution whilst preventing computational costs becoming too high for real-time tracking.

$$j^* = \arg\min_j \sum_{i=0}^{n} T_{ij}^m, j = 1....n \tag{6}$$

$$\tau^m = 3 * \sqrt{\sum_{i=0}^{n} (T_{ij^*}^m)^2} \tag{7}$$

The appearance model facilitates the activation of displacement predictors for the next frame by identifying which aspect or view is currently visible. Unlike the SMAT approach, the component medians are not used to identify displacement as no alignment process is performed. However, experiments have shown that, at significant cost to computational efficiency, some improvement in tracking accuracy may be achieved by including an additional optimisation procedure to align the images after the linear predictor stage. The results of these experiments are omitted here as the additional overhead of performing a full gradient decent optimisation was prohibitive to achieving the goal of real-time simultaneous modeling and tracking.

Figure 4. Prototypical results from tracker running at between 20 and 25 frames per second. Tracking through large pose variations and over 100 frames of occlusion

The following section describes how efficient and robust simultaneous modeling and tracking is obtained by associating view specific constellations of the linear predictors learnt online to various modes of such a multi-modal appearance model.

3.3. Robust tracking through adaptive LP subset selection

Each appearance model component, representing a partition of the target appearance manifold, can be viewed as a particular aspect of the target object. By learning predictors specific to a particular view of the target object the approach is able to continue to track through significant appearance changes. This association is achieved by an association matrix, **A**, as illustrated in figure 1. Given a bank of L linear predictors and M appearance model components, the association matrix **A** has dimension $(L \times M)$. A zero value at $\mathbf{A}_{lm}$ indicates that predictor l is not associated to component m and therefore is deactivated when component m is active. Each component is associated to L/M predictors. For all the experiments, M = 4 and L=160 meaning 40 LPs are associated to each component and hence that 40 linear predictions are computed each frame.

An error function is used to continually evaluate predictor performance over time. This allows poorly performing predictors to be replaced by predictors learnt online as well as providing a means for weighting each predictors contribution to overall tracker output, $\delta\bar{\mathbf{x}}$, defined in Eq. 10. Rather than assigning a single error value to predictor l, error values are instead assigned to the association between each of the L predictors and each of the M appearance model components. The error values are stored in the association matrix **A** and can also be interpreted as a measure of the strength of association between a predictor and an appearance model component. The performance value used is a running average of prediction error with exponential forgetting; meaning that high values indicate poor performance. The error function used is the L_2 norm distance between predictor output $\delta\mathbf{x}_l$ and the overall tracker output $\delta\bar{\mathbf{x}}$, $\|\delta\mathbf{x}_l - \delta\bar{\mathbf{x}}\|$. Equation 8 details how the association matrix is updated with these error values. The rate of forgetting is determined by parameter β=0.1, set experimentally.

$$A_{lm}^{t+1} = ((1-\beta) * A_{lm}^{t}) + (\beta * \|\delta\mathbf{x}_l - \delta\bar{\mathbf{x}}\|) \qquad (8)$$

Note that when a new component of the appearance model is created all the predictors from the previously used component are assigned to the new component by copying a column of **A**. A step by step description of the tracking procedure follows.

1 Initialise. Given the position and size of the target in the first frame, F^0, the first template, G^0, is extracted and used to create the first component of the appearance model. The range R is set to equal the largest dimension of the initial template and L/M linear predictor reference points are randomly selected from within a radius R of the target center. Linear mappings are learnt at all reference points by generating N artificial displacements and N intensity difference vectors, $\{\delta\mathbf{x}_i, \mathbf{d}_i\}$, (i ϵ [1,N]) and obtaining the least square solution using Eq. 3. Clearly the single appearance model component and this initial set of predictors is activated for prediction for the second frame F^1.

2 Compute the support pixel intensity difference vector d. Subtract the predictor's template intensity vector, **q**, from the intensity of the input image at all k support pixel locations, see Eq. 1, for each active predictor.

3 Make predictions Multiply intensity difference vectors **d** by mapping functions **H** to obtain motion predictions $\delta\mathbf{x}$, Eq. 2.

4 Compute weighted mean prediction $\delta\bar{\mathbf{x}}$ The error values stored in **A** are used to weight the contribution of each active predictor to the tracker output. The weight for predictor l when the active component is m is computed using Eq. 9 and is used to compute a weighted average of all active predictor outputs as in Eq. 10. Using the weighted mean as tracker output reduces the effect of prediction outliers caused by suboptimal predictor performance whilst minimising the computational overhead.

$$W_l^m = \begin{cases} \max(1 - \frac{\mathbf{A}_{lm}}{\mathbf{A}_{im}}), i = 1...L & \text{if } \mathbf{A}_{lm} > 0; \\ 0 & \text{if } \mathbf{A}_{lm} = 0. \end{cases} \qquad (9)$$

$$\delta\bar{\mathbf{x}} = \frac{\sum_l^L W_l^m \delta\mathbf{x}_l}{\sum_l^L W_l^m} \qquad (10)$$

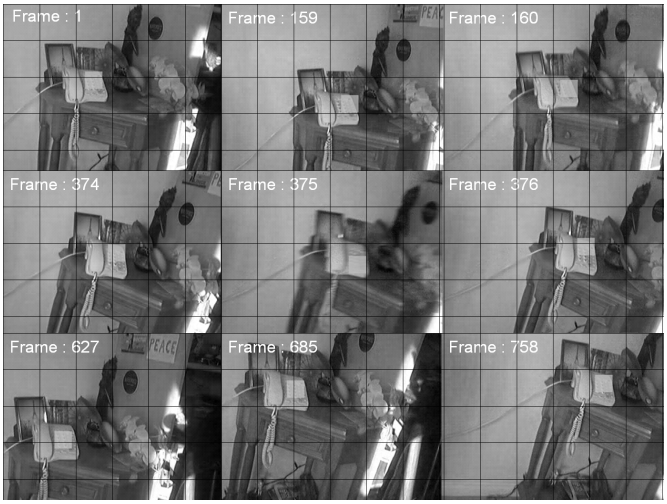

Figure 5. Prototypical results from tracker running at between 25 and 33 frames per second. Large inter frame displacement is handled as well as predicting from very blurred images.

5 Evaluate predictor performance Update association matrix $\mathbf{A}$ using Eq. 8 and identify the worst predictor, ϕ, from the current active component m using Eq. 11.

$$\phi = \arg\min_{l} A_{lm}, l = 1...L. \tag{11}$$

6 Update appearance model Using the new warp parameters, $\delta\bar{\mathbf{x}}$, a new template is extracted and compared to the median of each appearance model component. Comparison is performed in order of component weight until a match is found as described in section 3.2. The new template is then added to the appearance model and the partition is updated using Eq. 6 and 7.

7 Activate predictors for next frame Activate LPs associated to appearance model component matched and updated in step 6.

8 Learn new linear predictor A new predictor is learnt from every frame, as detailed in Eq. 3. It is learnt from synthetic displacements of the previous frame and used to make a prediction on the current frame. If the prediction error is less than the worst predictors error, $\|\delta\mathbf{x}_{new} - \delta\bar{\mathbf{x}}\| < \|\delta\mathbf{x}_{worst} - \delta\bar{\mathbf{x}}\|$, then the new predictor replaces the worst predictor (only in the current active component). This process serves both to introduce view-specific predictors as well as prevent outliers from contributing to the tracker output. Note that a predictor can be used by multiple components and is only completely destroyed if it has zero values for all components.

Repeat steps 2 through 8 for each new frame.

4. Evaluation

The system is demonstrated on two challenging and varied video sequences that illustrate the systems ability to simultaneously model and track objects through large inter frame displacements with robustness to changes in target appearance brought about by changes to pose and occlusion. Performance comparisons on both sequences are made with two alternative tracking approaches that require no offline learning or model building, namely the inverse compositional algorithm for the Lucas Kenade tracker [8] and SMAT [3]. Clearly the Lucas Kenade tracker, that has no template update strategy, is not a fair comparison and provides only a baseline performance. However, SMAT has been shown, [3], to outperform both naive and strategic update approaches [11, 7] and therefore provides a useful comparison. The sequences are both captured from a low cost web cam. The first sequence is 2500 frames long and features a human head and torso with the head undergoing large pose variations and at one point becoming occluded by a cup for over 100 frames. The second sequence is of a static scene and a moving camera. The camera undergoes considerable shaking causing large inter frame displacements as well as translation, rotation and tilting. The target patch is identified by hand only in the first frame. Ground truth for every frame was achieved by hand labeling and was used to generate the comparative error plots in figures 6 and 7.

For sequence 1 the target patch is 100*90 pixels and the tracker achieves between 20 and 25 frames per second without any tracking failures. Prototypical results are shown in figure 3.3. Figure 3 illustrates how the multi component appearance model adapts to the target appearance and helps the tracker overcome occlusions and large changes in pose. Figure 6 presents the comparative results using the ground truth data to compute the tracking error (euclidean distance to ground truth). It can be seen that the LK tracker (using L_2 norm and Levenberg-Marquardt optimisation) manages to track for around 150 frames after which it drifts completely off target. Both the presented tracker and the SMAT tracker achieve similar error plots on this sequence but due to the relatively large image patch the SMAT tracker operates at between 8 and 12 frames per second compared with 20 to 25. The worst results (from peaks in the error plot) for this sequence are also shown in figure 6.

For sequence 2 the target patch is 25*25 pixels and in this case the tracker operates at between 25 and 33 frames per second. Figure 4 shows some typical results with a grid placed over images so that interframe displacement can easily be seen. The middle row of figure 4 shows three consecutive frames. The displacement predicted from frame 374 to 375 is 37 pixels (16 vertical and 33 horizontal) and despite the significant blurring in frame 375, the tracker still succeeds in making a low error prediction to frame 376. Due to the constant online learning of predictors some are

Figure 6. Error plots for each tracker are shown along with the tracker output at peaks in the error surface i.e. the worst results.

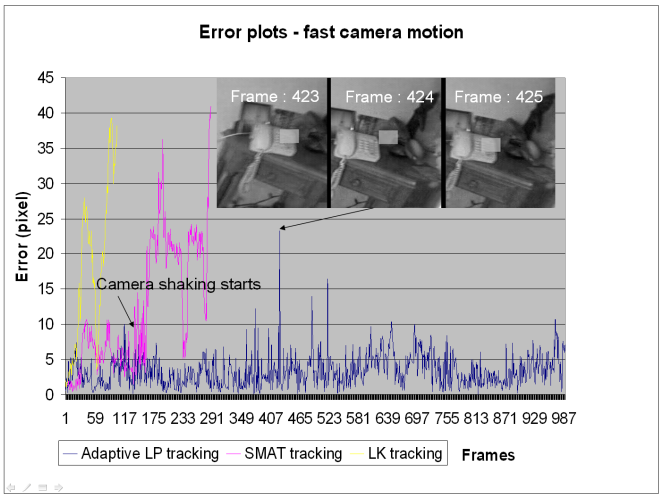

Figure 7. Error plots for each tracker are shown plus a three frame sequence illustrating the only tracking failure in the whole sequence

learnt from blurred images allowing for prediction during this blur. Figure 7 shows the error plots generated by the three trackers on this sequence. Due to the limited basin of convergence both the alignment based trackers fail to deal with the large inter frame displacements and SMAT loses track as soon as the camera starts to shake. There is one tracking failure in the sequence illustrated in figure 7 but due to the large prediction range of the predictors, recovery is achieved in the next frame.

Unique to this approach is the continual online learning and evaluation of efficient linear predictors. A fraction of the computational cost of the approach is attributed to this learning and adaptation rather than to tracking. For a sequence running at 25 frames per second, i.e. 40ms per frame, around 15ms per frame is spent learning new predictors (35% computational cost). The computational cost attributed to learning and maintaining the appearance model is around 50% of the overall cost (average 20ms per frame when running at 25 frames per second). Each LP takes less than 0.01ms per prediction so with 40 predictions per frame a total of 0.4ms per frame is spent making predictions i.e. about 1% of overall computational cost. The remaining 10 to 15% of computational cost is attributed to the computation of $\delta\bar{\mathbf{x}}$ as in Eq. 10 and maintaining and updating the association matrix.

5. Conclusion, discussion and future work

A novel approach to tracking visual features that requires no offline learning or hard coded models is presented and demonstrated. It is shown that the approach can handle large inter frame displacements and adapt to significant changes in the target appearance whilst running at up to 33 frames per second. Furthermore, the approach is shown to outperform two alternative approaches that also require no offline learning or hard coding.

The advantages of such a simultaneous modeling and tracking approach are clear when considering how much hand crafting, offline learning and parameter tuning must be done in order to employ many existing object tracking approaches. Many applications require tracking that operates at high frame rates and can handle high object velocities as well as be robust to significant appearance changes and occlusion. This is achieved here by utilising the computationally efficient technique of least squares prediction and modeling the target appearance by greedy partitioning of templates drawn from each frame.

6. Acknowledgement

Part of the work presented here was supported by the the European Union, grant COSPAL (FP6-IST-004176). This work is also partly funded by EPSRC through grants EP/P500524/1 and LILiR (EP/E027946) to the University of Surrey and the Czech Ministry of Education project grant (1M0567) to the Czech Technical University.

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework: Part 1, 2002. 1

[2] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *ECCV (2)*, pages 484–498, 1998. 1, 2

[3] N. Dowson and R. Bowden. N-tier simultaneous modelling and tracking for arbitrary warps. In M. Chantler, R. Fisher, and M. Trucco, editors, *Proc. of the 17th British Machine Vision Conference*. British Machine Vision Association, 2006. 1, 3, 6

[4] G.-M. B. H. Grabner, H. Real-time tracking via on-line boosting. In M. Chantler, R. Fisher, and M. Trucco, editors, *Proc. of the 17th British Machine Vision Conference*, pages 47–56. British Machine Vision Association, 2006. 1

[5] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. In *CVPR (1)*, pages 415–422, 2001. 1

[6] F. Jurie and M. Dhome. Real time robust template matching. In *BMVC*, 2002. 2

[7] T. Kaneko and O. Hori. Template update criterion for template matching of image sequences. In *ICPR (2)*, pages 1–5, 2002. 6

[8] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981. 1, 6

[9] É. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2d-3d model-based approach. In *ICCV*, pages 262–268, 1999. 1, 2

[10] J. Matas, K. Zimmermann, T. Svoboda, and A. Hilton. Learning efficient linear predictors for motion estimation. In *ICVGIP*, pages 445–456, 2006. 1, 2

[11] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):810–815, 2004. 1, 6

[12] O. M. C. Williams, A. Blake, and R. Cipolla. A sparse probabilistic learning algorithm for real-time tracking. In *ICCV*, pages 353–361, 2003. 1, 2

[13] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006. 1