Online Learning and Partitioning of Linear Displacement Predictors for Tracking

Liam Ellis[1], Jiri Matas[2], Richard Bowden[1]
[1]:CVSSP, University of Surrey, UK
[2]:CMP, Czech Technical University, Czech Republic
L.Ellis R.Bowden@Surrey.ac.uk, matas@cmp.felk.cvut.cz

Abstract

A novel approach to learning and tracking arbitrary image features is presented. Tracking is tackled by learning the mapping from image intensity differences to displacements. Linear regression is used, resulting in low computational cost. An appearance model of the target is built on-the-fly by clustering sub-sampled image templates. The medoidshift algorithm is used to cluster the templates thus identifying various modes or aspects of the target appearance, each mode is associated to the most suitable set of linear predictors allowing piecewise linear regression from image intensity differences to warp updates. Despite no hard-coding or offline learning, excellent results are shown on three publicly available video sequences and comparisons with related approaches made.

1 Introduction

The objective of this paper is to track without the need for hard coding and offline learning of either the variation in target appearance or motion models and is therefore applicable to a wide range of applications and scenarios. Yilmaz *et. al* introduced a taxonomy of tracking methods [15]. Within this taxonomy, our work falls into the class of methods identified as multi-view kernel methods. However unlike other methods reviewed, our approach learns the views of the target online.

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 carried out using gradient decent or Newton methods and hence assumes the presence of a locally convex similarity function with a minimum at the true 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.

For a visual tracking approach to be useful it should 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 here by employing a novel, flexible

BMVC 2008 doi:10.5244/C.22.4

and adaptive object representation for efficient tracking comprised of sets of spatially localised linear displacement predictors adaptively associated to aspects (clusters) of a template based appearance model. Linear predictors can have a wider basin of convergence than registration methods and due to their simplicity are computationally efficient.

Previous linear predictor trackers have tended to rely on hard coded models of object geometry [11, 10]. This requires significant effort in hand crafting the models and like simple template models [9, 1, 12], 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 [14] 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. [7] have shown increased robustness by adapting the model to variations encountered during tracking, but with high computational overhead.

Recent approaches that achieve real-time tracking and have adopted an entirely online learning paradigm are; the discriminative tracker [6] that uses an online boosting algorithm to learn a discriminative appearance model on the fly, the SMAT algorithm[3] and variants such as [4, 5]. In the SMAT approach, a greedy clustering algorithm is used to learn a multi-modal appearance model of the object. The prototype of each cluster is then used in an LK framework [1]. Our approach maps appearance to variable banks of linear predictors which are constantly updated. Appearance is modeled using the recently introduced medoidshift algorithm to incrementally cluster templates.

1.1 Displacement prediction

Cootes et al. proposed a method for pre-learning a linear mapping between the image intensity difference vector and the error (or required correction) in AAM model parameters [2]. Jurie et al. employed similar *linear predictor* (LP) functions to track rigid objects [8]. 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. [11], uses simpler linear regression for displacement prediction, similar to the linear predictor functions in [8] 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 [10] but Matas et al. have shown that a low predictor error does not necessarily coincide with high image intensity gradients [11]. 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, and found that randomised sampling is efficient and sufficient [11]. The approach presented here avoids the need for costly reference point and support selection strategies by evaluating the 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 [11, 14, 10] require either an offline learning stage or the construction of a hand coded model or both.

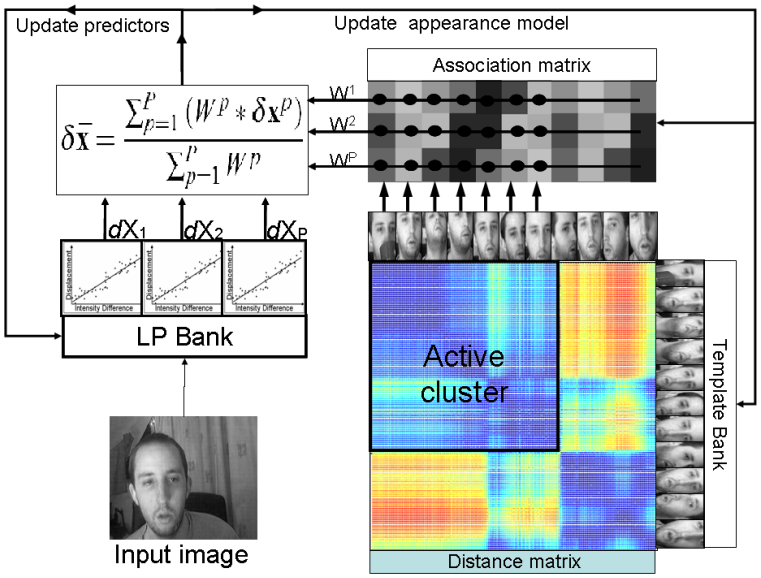

Figure 1: Overview of approach: LPs each make a prediction, the contribution to the overall output is based on the LPs association to appearance clusters

2 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 face tracking. The approach requires no offline learning stage or hand coded models and only requires that the initial location of the target be given.

The appearance model is learnt on-the-fly during tracking by clustering sub-sampled image patches or templates drawn from the tracked target position in every frame (see section 2.2). These templates are clustered online yielding modes that represent different views or aspects of the target. The appearance model is illustrated in figure 1 by the bank of templates and the clustered distance matrix. Also learnt online is a set of linear regression functions that predict motion parameters from image intensity difference vectors (see section 2.1). Tracker output is computed as a weighted mean from these predicted displacements. Weightings reflect each predictors association to the current target appearance as predicted by the appearance model.

The performance of each predictor is continually evaluated over time and is used to update the weighting matrix and hence its association to the various aspects of the target. Furthermore, new predictors are learnt every frame to replace the worst performers.

2.1 Linear predictor tracker

The linear displacement predictors compute motion at a reference point from a set of pixels sub-sampled from its neighbourhood called the support set $\mathbf{S} = \{\mathbf{s}_1, ..., \mathbf{s}_k\}$. The intensities observed at the support set $\mathbf{S}$ are collected in the observation vector $\mathbf{I}(\mathbf{S})$. The

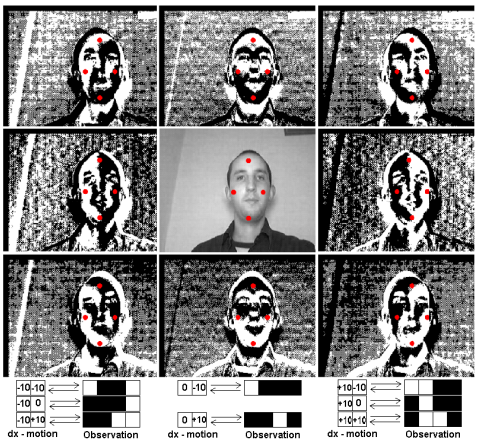

Figure 2: Intensity difference images for eight translations. Four support pixel locations illustrate the predictive potential of the difference image. The input image is in the center. All images to the left/right of the input have been translated left/right by 10 pixels. Those images above/below the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

$\mathbf{l}_0(\mathbf{S})$ vector contains the intensities observed in the initial image. Here the motion is a 2D translation $\mathbf{t}$, we use $(\mathbf{S} \circ \mathbf{t}) = \{(\mathbf{s}_1 + \mathbf{t}), ..., (\mathbf{s}_k + \mathbf{t})\}$ to denote the support set transformed by $\mathbf{t}$. Translation is sufficient as the multi-modal appearance model copes with affine deformations of the image templates, also shown in [4].

Predictions are computed as in Eq. (1) where $\mathbf{H}$ is a $(2 \times k)$ matrix that forms a linear mapping $\Re^k \rightarrow \Re^2$ from image intensity differences, $\mathbf{d} = \mathbf{l}_0(\mathbf{S}) - \mathbf{l}(\mathbf{S} \circ \mathbf{x})$, to changes in warp parameters, $\delta\mathbf{x}$. The state vector, $\mathbf{x}$, is the 2D position of the predictor after prediction in the last frame. This efficient prediction only requires k subtractions and a single matrix multiplication, the cost of which is proportional to k.

$$\delta\mathbf{x} = \mathbf{H}\mathbf{d} = \mathbf{H}(\mathbf{l}_0(\mathbf{S}) - \mathbf{l}(\mathbf{S} \circ \mathbf{x})) \tag{1}$$

In order to learn the linear regressor, $\mathbf{H}$, training examples of $\{\delta\mathbf{x}_i, \mathbf{d}_i\}$ pairs, $(i \in [1, N])$ are required. These are obtained from a single training image by applying synthetic warps and subtracting the deformed image from the original. For efficiency the warp and difference computation is only performed at the support pixel locations but, for illustration, the result of performing this operation on the entire image is shown in figure 2 for eight different translation warps. Also marked on the figure are four possible locations for support pixels and the unique observation patterns they produce.

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

$$\mathbf{H} = \mathbf{X}\mathbf{D}^+ = \mathbf{X}\mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1} \tag{2}$$

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 a maximum of 30 pixel interframe displacement. The predictor complexity, k, models the trade off between speed of prediction and accuracy. N does not affect 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 hindering the goal of real-time tracking.

2.2 Medoidshift clustering for online appearance modeling

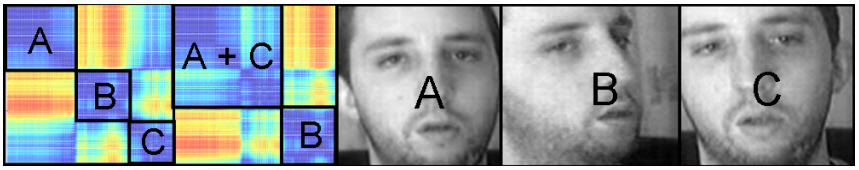

Figure 3: The distance matrix pre and post clustering is shown with three subsets of exemplars **A**, **B** and **C**. Sets **A** and **C** are temporally separated but have the same appearance. Templates from each subset are also shown.

The appearance model presented here is constructed online by incrementally clustering sub-sampled image patches to identify various modes of the target appearance manifold. If a single template appearance of an object is considered as one point on the appearance-space manifold, the manifold can be represented by storing all templates, $\mathbf{T} = \{\mathbf{G}^0 ... \mathbf{G}^t\}$ drawn from all frames $\{\mathbf{F}^0 ... \mathbf{F}^t\}$.

Clustering the set of appearance templates, $\mathbf{T}$, identifies different views or aspects of the target and facilitates the use of view specific displacement predictors as described in section 2.3. The clustering is performed by the medoidshift algorithm introduced by Sheikh et. al [13] using the SSD between subsampled image templates. Medoidshift is a nonparametric clustering approach that performs mode-seeking by computing shifts toward areas of greater data density using local weighted medoids. As Sheikh et. al show, the procedure can be performed incrementally, meaning the clustering can be updated at the inclusion of new data samples and the removal of some existing data samples.

During each of the first 10 frames of tracking, the sub-sampled image templates are collected into vectors $\{\mathbf{G}^0 ... \mathbf{G}^{10}\}$ and a distance matrix is populated with the SDD distances. On frame 11 the medoidshift algorithm partitions the distance matrix to obtain an initial template clustering and then for each subsequent frame the clustering is incrementally updated given a new $\mathbf{G}$ vector and hence (by computing SSD values) a new row/column of the distance matrix. In order to constrain the memory requirements and computational complexity of maintaining the appearance model, the number of templates retained, and hence the number of data points clustered, is limited. Once the limit has been reached the oldest template is removed and replaced with the new template. Now the cluster update must accommodate both the introduction and removal of data points.

The incremental update is achieved in a computationally efficient manner exactly as described in [13].

The effect of this clustering, illustrated in figure 3, shows the distance matrix at frame 275 of a head tracking sequence (see section 3 for details of video) before and after matrix indices are sorted according to the clustering.

2.3 Aspect specific predictor tracking

Each cluster of appearance templates can be viewed as a particular aspect of the target object. Furthermore, a cluster may represent the appearance of the target during occlusion, lighting changes or motion blurring. By learning cluster specific predictor weightings, each predictor can be associated to a greater or lesser extent to each aspect or appearance mode. This, combined with the continual learning of new predictors enables this approach to continue to track through significant appearance changes.

The weighting mechanism is achieved by an association matrix, $\mathbf{A}$, as illustrated in figure 1. Given a bank of P linear predictors and a set, $\mathbf{T}$, of M appearance templates, $\mathbf{T} = \{\mathbf{G}^0 ... \mathbf{G}^M\}$, the association matrix $\mathbf{A}$ has dimension $(P \times M)$. The value at $\mathbf{A}_{pm}$ indicates the strength (or weakness) of association between predictor p and template (exemplar) m. The values of $\mathbf{A}$ are set and updated using Eq. (3) and (4). Equation (3) shows how the prediction error is computed and used to initialise the association values between each predictor and the new exemplar. The error is the L_2 norm distance between the expected and observed pixel intensities at the predictors support pixel locations. Eq. (4) is used to update the association values for all the other exemplars in the active cluster, $\mathbf{T}_a \subset \mathbf{T}$. This has the effect of smoothing the performance measures within a cluster. The values are a running average prediction error with exponential forgetting; meaning that low values of $\mathbf{A}_{pm}$ indicate greater association between predictor p and clusters containing exemplar m. The rate of forgetting is determined by parameter β=0.1, set experimentally. In all the experiments P=80 and M=500, also set experimentally.

$$\mathbf{A}_{pM} = \|\mathbf{l}_0^p - \mathbf{l}_M^p\|, p = 1...P \tag{3}$$

$$\mathbf{A}_{pm} = \begin{cases} ((1-\beta) * \mathbf{A}_{pm}) + (\beta * \|\mathbf{l}_0^p - \mathbf{l}_M^p\|), p = 1...P & \text{if } \mathbf{G}^m \in \mathbf{T}_a \\ \mathbf{A}_{pm} & \text{if } \mathbf{G}^m \notin \mathbf{T}_a \end{cases} \tag{4}$$

This error function and update strategy are used to continually evaluate predictor performance over time. This provides a means for appearance dependent weighting of each predictors contribution to overall tracker output, $\delta\bar{\mathbf{x}}$, as defined in Eq. (5) and Eq. (6).

$$Wp = 1 - \frac{\sum_{m=1}^{M^*} \mathbf{A}_{pm}}{\max \sum_{m=1}^{M*} \mathbf{A}_{pm}}, M^* = |\mathbf{T}_a| \tag{5}$$

$$\delta\bar{\mathbf{x}} = \frac{\sum_{p=1}^{P} (W^p * \delta\mathbf{x}^p)}{\sum_{p=1}^{P} W^p} \tag{6}$$

The continuous evaluation of predictor performance also allows poorly performing predictors to be replaced by predictors learnt online. The worst predictor, p^*, is identified as in Eq. (7). The entries in $\mathbf{A}$ relating to the replaced predictor are updated as in Eq. (8).

$$p^* = \arg \max_{\{p=1,\ldots,P\}} (\min_{\{m=1,\ldots,M\}} \mathbf{A}_{pm}) \tag{7}$$

$$\mathbf{A}_{p^*m} = \frac{\sum_{p=1}^{P} \mathbf{A}_{pm}}{P}, m = 1 \ldots M \tag{8}$$

The complete tracking algorithm is summarised in Algorithm 1.

Algorithm 1 Complete tracking procedure

$\mathbf{F}^0 \leftarrow$first image
Initialise target position $\bar{\mathbf{x}}^0$ and size R from user input
for $p = 0$ to P **do**
  $\mathbf{x}^p = rand(-R/2 : R/2)$ {Randomly select reference point}
  Generate $\{\delta\mathbf{x}_i, \mathbf{d}_i\}$ {Training data}
  Compute $\mathbf{H}^p$ as is Eq. (2)
  $W^p \leftarrow 1$ {Set all initial predictor weights to 1}
end for
while $\mathbf{F}^t \neq NULL$ **do**
  Compute $\delta\mathbf{x}^p$ as in Eq. (1) for p = {0 ... P}
  Compute $\delta\bar{\mathbf{x}}$ as in Eq. (6)
  Update predictor states $\mathbf{x}^p = \mathbf{x}^p + \delta\bar{\mathbf{x}}$
  Extract new appearance template $\mathbf{G}^t$
  Compute new row and column of distance matrix, SSD $\mathbf{G}^t$ and $\{\mathbf{G}^0 \ldots \mathbf{G}^{t-1}\}$
  Obtain $\mathbf{T}_a \subset \{\mathbf{G}^0 \ldots \mathbf{G}^{t-1}\}$ {Obtained by clustering $\mathbf{T} = \{\mathbf{G}^0 \ldots \mathbf{G}^{t-1}\}$}
  Update association matrix, **A**, as in Eq. (3) and Eq. (4)
  Identify worst predictor as in Eq. (7)
  Learn new predictor as in Eq. (2)
  if new predictor performance $\geq$ old predictor performance **then**
    Replace worst predictor p^*
    Update association matrix, **A**, as in Eq. (8)
  end if
  Compute predictor weightings for next frame as in Eq. (5)
  $t \leftarrow t + 1$
end while

3 Evaluation

The system is demonstrated on three publicly available [1] challenging and varied video sequences, that illustrate the systems ability to track objects through large inter frame displacements with robustness to changes in target appearance brought about by changes to pose and occlusion, see figure 4. For comparison, four alternative trackers were run on the athletics and camera motion sequences, figure 5, namely the inverse compositional algorithm for the Lucas Kanade (LK) tracker [9], SMAT [4], SMAT using a bank of

[1] http://info.ee.surrey.ac.uk/Personal/L.Ellis/research.html

Figure 4: Tracking results obtained from running the tracker on three varied video sequences.

LPs per mode rather than gradient descent and finally a 'flock' of LPs with no learnt appearance model and no relearning.

Datasets: The *athletics* sequence is 430 frames long, all the trackers fail at around frame 400 but up to that point only the trackers with learnt appearance models (SMAT, SMAT with LPs and the proposed medoidshift with LPs) track the target successfully, figure 5. The target changes scale considerably during the sequence and, as the tracker is not scale invariant, new scales are treated as new appearance modes. The use of LPs with the SMAT appearance model performs similarly to the proposed medoidshift with LPs (LP MED), as both approaches have the benefit of a wide basin of convergence and multi-modal appearance. However, SMAT requires optimal parameter selection for clustering e.g. Number of modes, learning rate.

The second sequence, captured from a low cost web cam, 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. Figure 6 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 online learning of predictors, some are learnt from blurred images allowing for prediction during this blur. Figure 5 shows the error plots generated by the five 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. The final sequence is a head tracking sequence lasting 2500 frames with the head undergoing large pose variations and at one point becoming occluded by a cup for over 100 frames.

For each sequence the target patch is identified by hand only in the first frame, all algorithm parameters are unchanged between sequences. Ground truth for every frame of the athletics and camera motion sequences was achieved by hand labeling and was used to generate the error plots in figure 5.

The tracker runs at 15-20 fps even with the clustering procedure carried out in Matlab with high parameter passing overheads. This could be improved by implementing in C++.

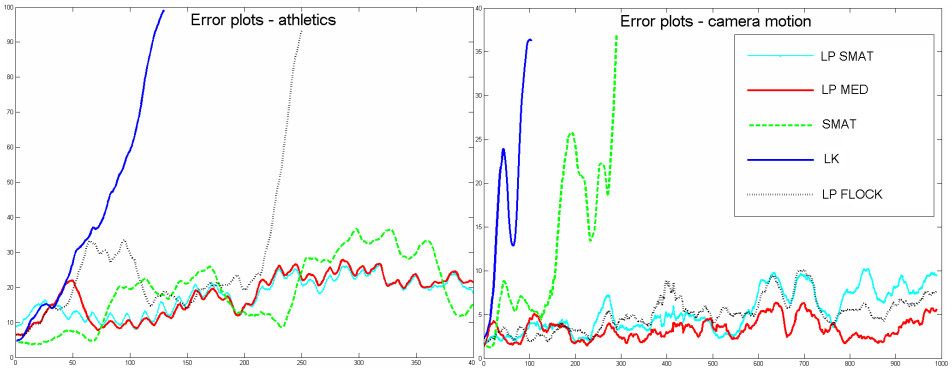

Figure 5: Error plots comparing the performance of five trackers on the athletics and camera motion sequence. This tracker is denoted *LP MED*. The other trackers: *SMAT*, *LP SMAT* (a tracker using the SMAT appearance model with LPs), *LK* (Lucas Kanade) and *LP FLOCK*(one constellation of LPs with no appearance modeling and no relearning).

Figure 6: Prototypical results show large inter frame displacement are handled as well as predicting from very blurred images

4 Conclusion

This approach to tracking visual features requires no offline learning or hard coded models and reduces the need for tuning parameters. It is shown that the approach can handle large inter frame displacements and adapt to significant changes in the target appearance with low computational cost.

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 online target appearance modeling.

5 Acknowledgments

This work was supported by the EU FP7 project DIPLECS and the EPSRC project LILiR.

References

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

[2] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. In *ECCV (2)*, pages 484–498, 1998.

[3] N.D.H. Dowson and R. Bowden. N-tier simultaneous modelling and tracking for arbitrary warps. In *Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 99–105. Computer Vision and Pattern Recognition, 2005.

[4] N.D.H. 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.

[5] L Ellis and R Bowden. Linear predictors for fast simultaneous modeling and tracking. In *submitted to Workshop on Non-rigid Registration and Tracking through Learning*, Eleventh IEEE Intl. Conf. Computer Vision, Rio de Janeiro, Brazil, 2007.

[6] Grabner-M. Bischof 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.

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

[8] Frédéric Jurie and Michel Dhome. Real time robust template matching. In *BMVC*, 2002.

[9] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.

[10] Éric Marchand, Patrick Bouthemy, François Chaumette, and Valérie Moreau. Robust real-time visual tracking using a 2d-3d model-based approach. In *ICCV*, pages 262–268, 1999.

[11] Jiri Matas, Karel Zimmermann, Tomás Svoboda, and Adrian Hilton. Learning efficient linear predictors for motion estimation. In *ICVGIP*, pages 445–456, 2006.

[12] Iain Matthews, Takahiro Ishikawa, and Simon Baker. The template update problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):810–815, 2004.

[13] Yaser Ajmal Sheikh, E.Khan, and Takeo Kanade. Mode-seeking by medoidshifts. In *Eleventh IEEE International Conference on Computer Vision (ICCV 2007)*, number 141, October 2007.

[14] Oliver M. C. Williams, Andrew Blake, and Roberto Cipolla. A sparse probabilistic learning algorithm for real-time tracking. In *ICCV*, pages 353–361, 2003.

[15] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006.