Regression tracking with data relevance determination

Ioannis Patras
Department of Electronic Engineering
Queen Mary, University of London, UK
I.Patras@elec.qmul.ac.uk

Edwin R. Hancock
Department of Computer Science
The University of York, UK
erh@cs.york.ac.uk

Abstract

This paper [1] *addresses the problem of efficient visual 2D template tracking in image sequences. We adopt a discriminative approach in which the observations at each frame yield direct predictions of a parametrisation of the state (*e.g. *position/scale/rotation) of the tracked target. To this end, a Bayesian Mixture of Experts (BME) is trained on a dataset of image patches that are generated by applying artificial transformations to the template at the first frame. In contrast to other methods in the literature, we explicitly address the problem that the prediction accuracy can deteriorate drastically for observations that are not similar to the ones in the training set; such observations are common in case of partial occlusions or of fast motion. To do so, we couple the BME with a probabilistic kernel-based classifier which, when trained, can determine the probability that a new/unseen observation can accurately predict the state of the target (the 'relevance' of the observation in question). In addition, in the particle filtering framework, we derive a recursive scheme for maintaining an approximation of the posterior probability of the target's state in which the probabilistic predictions of multiple observations are moderated by their corresponding relevance. We apply the algorithm in the problem of 2D template tracking and demonstrate that the proposed scheme outperforms classical methods for discriminative tracking in case of motions large in magnitude and of partial occlusions.*

1. Introduction

Recently, a number of methods have been proposed for the visual tracking of the state x (*e.g.* position/scale/rotation/3D pose) of a target given a set of noisy image observations $Y = \{\ldots, y^-, y\}$ up to the current frame [6][4][3]. Generative methods, which rely on the evaluation of the likelihood $p(y|x)$ at certain points of the state space, require the inversion of the posterior $p(x|Y)$ and expensive schemes for searching the state space. Detection-based methods [13] that exhaustively search all image locations for the presence/absence of a visual target fall in this category. The search in the state space can facilitated by a dynamic model, but often, as in the case of irregular motion, the temporal evolution of the state may deviate significantly from it.

In contrast, discriminative tracking methods attempt to model explicitly the posterior $p(x|y)$, so that an observation y can deliver direct prediction of the hidden state x. The posterior is learned in a supervised way from training data that are usually artificially generated. In this framework, recently a number of researchers have proposed methods for estimating 3D human pose and for 2D template tracking. Agarwal and Triggs [2] use Relevance Vector Machines (RVMs) in order to learn mappings between vectors of image descriptors and the 3D poses of a human body. For 3D human pose tracking, Sminchisescu *et al*. [11] train Bayesian Mixture of Experts in order to learn a multimodal posterior $p(x|y)$. For 3D pose tracking, Agarwal and Triggs [1] use Non-negative Matrix Factorisation in order to remove parts of the observation vector that are due to noise or occlusion. For 2D tracking, Williams *et al*. [9] use RVMs in order to learn the posterior of the location/scale/orientation of a visual target (*e.g*. a human face) given an observation at a certain image location. Finally, for 2D tracking, Jurie and Dhome [8] learn in a supervised way a linear relation between the intensity differences between two templates and the corresponding motion transformation.

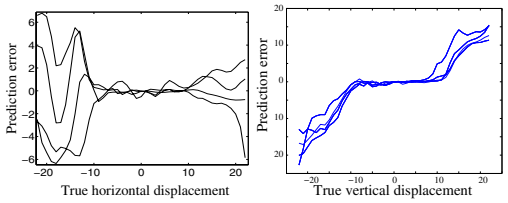
In order to deal with possibly large prediction errors most of the previous methods rely mainly on temporal filtering. Sminchisescu *et al*. [11] and Agarwal and Triggs [2] use as observations features that are extracted from a single object silhouette. They address prediction errors by adopting a multiple hypotheses tracking framework that performs temporal filtering. On the other hand, Williams *et al*. [9] couple the regression-based tracking with a detection-based scheme that is employed to validate that the target is in the predicted position/pose. In case of a validation failure a full

[1] The work has been conducted while the first author was with the CVPR group at the University of York

1-4244-1180-7/07/$25.00 ©2007 IEEE

scale detection phase is initiated. A Kalman filter is used for temporal filtering and leads to a reduction of the error of an order of magnitude.

However, none of these methods addresses explicitly the problem of assessing in advance how well the observation y can predict the state x nor do they use multiple observations in order to increase robustness. Regression-based methods are known to be sensitive to observations that do not belong to the space that is sampled by the training dataset. Therefore the accuracy of the prediction of the posterior $p(x|y)$ can deteriorate sharply for observations y that are contaminated with noise or come from areas that are uninformative of the state of the visual target (*e.g.* occluded areas). In particular in the case of the 2D tracking, when the motion magnitude is larger than in the training set the prediction error is likely to be large and the tracking is likely to fail. In Fig. 1 we illustrate this effect by plotting the prediction error as a function of the true displacement in an artificial example. Similar observations are reported in [8] and [9].



(a) (b)

Figure 1. Prediction error as function of the true displacement. (a) error as a function of the true horizontal displacement (b) error as a function of the true vertical displacement. The performance deteriorates sharply outside the training area (in this example a Bayesian mixture of Experts was trained for displacements in the interval $[-11 \dots 11]$ for a template of size 10×10)

In this paper, for 2D visual tracking, we extend the discriminative/regression tracking framework ([11]) in two ways:

- We explicitly address the problem of the determination of the relevance/reliability of an observation to the state estimation by learning in a supervised way the underlying conditional probability distribution.
- We explicitly devise a probabilistic framework that allows multiple observations $y(r)$ to contribute to the prediction of the state of the target according to their corresponding relevance/reliability.

In this way, the contribution of the predictions that come from relevant observations is higher, while observations that come from occluded areas or observations that can not give good predictions are largely suppressed. We propose an extension of the discriminative particle filtering framework that incorporates additional random variables r that are used to obtain/utilise multiple observations denoted by $y(r)$, and binary random variables z that are related to the observations' relevance. We use Relevance Vector Machines [12] in order to learn the conditional probability $p(z = 1|y(r))$ (the probability that the observation $y(r)$ is relevant/reliable). A Bayesian Mixture of Experts [15] is used for modelling $(p(x|y(r)))$ (the posterior probability of the state x given an observation $y(r)$). An outline of the proposed method is given in Fig. 2.

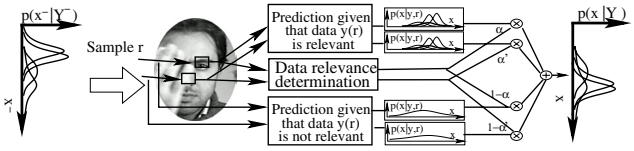


Figure 2. Overview of the proposed method

The remainder of the paper is organised as follows. In Section 2 we provide an outline of the proposed discriminative tracking framework with data relevance determination. In Section 2.1 we briefly describe the Bayesian Mixture of Experts framework and in Section 2.2 we present our method for data relevance determination. In Section 3 we present experimental results for 2D target tracking and in Section 4 we present some conclusions.

2. Regression-based tracking with relevance determination

Filtering, such as Kalman filtering or particle filtering, has been the dominant framework for recursive estimation of the conditional probability of the unknown state x given a set of observed random variables $Y = \{\dots y^-, y\}$ up to the current time instant. In the discriminative filtering framework (Fig. 3(a)) the filtered density can be derived as [11]:

$$p(x|Y) = \int dx^- p(x^-|Y^-) p(x|x^-, y). \qquad (1)$$

This derivation ignores the fact that for certain problems different parts of the observation y can give different predictions of the state of the target. For example, in [9], for 2D tracking where the evidence y is an image frame, the prediction of the state of the target (*e.g.* 2D location) is based on the data $y(r)$ in a single window, which (in the absence of a motion model) is centred around the estimated position $r = \hat{x}^-$ of the target in the previous frame [2]. This disregards the information that is available at other positions r. Similarly, for 3D tracking, in [2] [11] a single feature vector is extracted from the object silhouette. On the other hand, in the generative particle filtering framework for 2D tracking it

[2] In the case that the state x is not only a 2D displacement obtaining $y(r)$ requires warping

is common practice that several parts of the observation are examined. This is achieved by using multiple samples (particles) r and modelling the likelihood that is used to evaluate the importance of each particle as $p(y|r) = p(y|y(r))$. The particles r are sampled using the transition probability $p(x|x^-)$ and, in the simplest case, a number of measurements $y(r)$ around the positions of the particles in the previous frame are utilised.

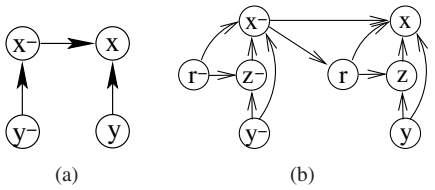


Figure 3. Graphical models for (a) classical discriminative tracking and (b) for regression tracking with relevance determination

Here, we propose a discriminative particle filtering method that utilises the fact that several parts of the observation can yield predictions of the state of the target. We do so by introducing a random variable r that determines which parts, or in general how, the observation y will be used. In order to simplify the notation we will assume here that r has the dimensionality and the physical meaning of the hidden state x. For example, when $x \in R^2$, the random variable $r \in R^2$ will determine the centres of the windows/patches at which we will extract observations $y(r)$ that will give predictions of x. In general r will be used for obtaining a set of candidate observations $y(r)$ and does not need to have the dimensionality of x. We will also condition r on x^- as we expect that the previous state can be sufficiently informative on how candidate observations can be obtained. Subsequently, we introduce a binary variable z and denote with $p(z = 1|y, r)$ the probability that the observation $y(r)$ is relevant for the prediction of the unknown state x. The dependencies of the variables are depicted in Fig. 3(b) where y is observed and the rest is hidden.

In this network the filtered density can be derived as:

$$p(x|Y) = \int dx^- p(x^-|Y^-) \left(\int dr p(r|x^-) \int dz p(x|z, x^-, y, r) p(z|y, r) \right) \quad (2)$$

In order to deal with posteriors with multiple modes and to recover from tracking failures we maintain an approximation of the $p(x|Y)$ using a mixture of M Gaussians. In Table 1 we summarise our modelling choices. We assume that the probability distributions in Table 1 are either given or learned in the training phase (as explained in Sections 2.1 and 2.2). For example, in the testing phase, given a triple (y, r, x^-), the trained BME yields a mixture of Gaussians that is our approximation of $p(x|z = 1, x^-, y, r)$.

In what follows we will describe a computational scheme that given an approximation of the posterior $p(x^-|Y^-)$ of the state at the previous frame, yields an approximation of the state posterior $p(x|Y)$ at the current frame. This is achieved by the following procedure:

1. Sample a state x^- from $p(x^-|Y^-)$.
2. Sample r from $p(r|x^-)p(x^-|Y^-)$ by sampling r from $p(r|x^-)$ for each of the state samples x^- obtained in step 1. Let us assume that R samples are obtained this way.
3. For each of the R samples r
   (a) Evaluate the relevance of the observation $y(r)$ as $\alpha = p(z = 1|y, r)$
   (b) Given (x^-, y, r) use the trained Bayesian Mixture of Experts to obtain a probabilistic prediction of the state x, given that $y(r)$ is relevant (*i.e.* $p(x|z = 1, x^-, y, r)$) as a mixture of K Gaussians. Model the probabilistic prediction given that $y(r)$ is irrelevant (*i.e.* $p(x|z = 0, x^-, y, r)$) as a single Gaussian with a large covariance matrix. This leads to a mixture of $K + 1$ Gaussians

$$\int dz p(x|z, x^-, y, r) p(z|y, r) = \alpha \sum_{i=1}^{K} g_i \mathcal{N}(\mu_i + r, S_i) + (1 - \alpha) \mathcal{N}(x^-, S_0) \quad (3)$$

4. Approximate the resulting mixture of L Gaussians ($L = R(K + 1)$) with a mixture of M Gaussians (see appendix A).

Note that in Eq. 3, the integral is approximated using $K+1$ Gaussian components. In practice, in order to reduce the number of the components, we use the approximation:

$$\int dz p(x|z, x^-, y, r) p(z|y, r) = \begin{cases} \alpha \sum_{i=1}^{K} \pi_i \mathcal{N}(\mu_i + r, S_i) & \text{if } \alpha > \theta_z \\ \mathcal{N}(x^-, S_0) & \text{otherwise} \end{cases}, \quad (4)$$

As a result we approximate the term $\int_r p(r|x^-) \int_z p(x|z, x^-, y, r) p(z|y, r)$ with an (unnormalised) mixture of L ($R \leq L \leq RK$) Gaussians. This is reduced to an M-component mixture in step 4.

2.1. Bayesian mixture of experts for regression

In what follows we will describe a method that, given an observation $y(r)$ and the target state at the previous frame

$p(r\|x^-)$	Uniform around x^-.
$p(z=1\|y,r)$	Probability that the observation $y(r)$ is relevant. Modelled using a probabilistic classifier (RVM) as $sigm\left(\sum_i w_i^{rvm}\phi(y(r),\tilde{y}_i)\right)$, where $sigm$ is the sigmoid function and $\{\tilde{y}_i\}$ is the training set of the classifier. It can only be evaluated.
$p(x\|z=1,x^-,y,r)$	Probabilistic prediction of the target state given that the observation $y(r)$ is relevant ($z=1$). Modelled as a mixture of Gaussians using Bayesian Mixture of Experts, that is as $\sum_{i=1}^{K} g_i\mathcal{N}(r+\mu_i,S_i)$
$p(x\|z=0,x^-,y,r)$	Probabilistic prediction of the target state given that the observation $y(r)$ is not relevant ($z=0$). It is modelled as a Gaussian with mean x^- and a large covariance matrix, that is as $\mathcal{N}(x^-,S_0)$.

Table 1. Modelling choices.

x^-, yields a probabilistic prediction of the state x at the current frame. For notational simplicity, let us denote here with y the couple $(y(r),x^-)$.

Our method, follows the work of Sminchisescu *et al.* [11] and uses for regression Bayesian Mixtures of Experts. The rationale behind our choice, over other regression methods (*e.g.* RVMs [12]) is that the BME can model well predictive distributions that are multimodal. Such distributions arise often in the case of 3D tracking due to for example front/back and left/right ambiguity [11][2][10]. They are also expected to arise in the case of 2D tracking due to the aperture problem [5].

The (Hierarchical) Mixtures of Experts, which were first introduced by Jordan and Jacobs [7], is a method for regression and classification that relies on soft probabilistic partitioning of the input space. This is determined by gating coefficients $g_i(y)$ (one for each expert i) that are input dependent and have a probabilistic interpretation; that is the coefficients of the siblings at each level of the hierarchy sum up to one. The prediction of each expert i is then moderated by the corresponding gating coefficient. Formally, for regression and for the simple case of a flat hierarchy:

$$p(x|y)=\sum_{i=1}^{K} g_i(y)f_i(x|y) \tag{5}$$

where $f_i(x|y)$ is a probability density function, usually a Gaussian centred around the prediction of the expert i. In the simple linear case:

$$g_i(y)=\frac{e^{\xi^T y}}{\sum_i e^{\xi^T y}}, \tag{6}$$

and

$$f_i(x|y)=\mathcal{N}(w_i^T y,S_i). \tag{7}$$

where the w_i and ξ_i are the unknowns to be estimated. Jordan and Jacobs [7] proposed a Maximum Likelihood method for the estimation of w_i and ξ_i, while in [15] Bayesian approach is used. We adopt the approach in [15] in which a set of hyperparameters model the prior distributions of w_i and ξ_i, and follow a variational approach for the estimation of their posterior distributions. As in [15] we make a Laplace approximation under which we estimate the mode and the variance of the posteriors, which (with a slight abuse of notation) denote here as (w_i,Σ_{w_i}) and (ξ_i,Σ_{ξ_i}). In the process, we also estimate the optimal value for the hyperparameters β_i that are associated with the noise (co)variance S_i of the prediction of expert i. For more details the reader is referred to [15].

In [15] a procedure is described for scalar regression. In the case that the target is a vector x with dimensionality D we may train D different Mixture of Experts. Here, we have extended the methodology to experts that have multidimensional output (*i.e.* $f_i(x|y)$ is a multidimensional Gaussian with diagonal noise covariance).

For prediction we marginalise over the parameters and hyperparameters as in [15]. For a new observation y the predictive distribution is a mixture of Gaussians given by:

$$\hat{p}(x|y)=\sum_i g_i(y)\mathcal{N}\left(w_i^T y,S_i'\right), \tag{8}$$

where the k^{th} element of the diagonal covariance matrix S_i' is given by:

$$y^T\Sigma_{w_{ik}}y+S_{ik} \tag{9}$$

For the problem of 2D visual tracking, we aim at the estimation of the transformation x (*e.g.* translation/rotation/scaling) that a visual target undergoes in an image sequence. We train the BME in a supervised way with pairs $(y(x),x)$ in which the observations $y(x)$ are produced by artificially transforming (*e.g.* translating) the visual target with the transformation x. Subsequently in the test phase, an observation will give a probabilistic prediction according to Eq. 8.

2.2. Data relevance determination

For the determination of the relevance $p(z|y,r)$ of an observation $y(r)$ we use a classification scheme with the Relevance Vector Machines (RVM). The goal is to obtain an *a priori* assessment of whether the probabilistic prediction $\hat{p}(x|y(r))$ (Eq.8) of the state of the target is expected to be good. To this end, we train an RVM classifier in a supervised way with a set of positive examples that yield good predictions and with a set of negative examples which yield

bad predictions. Let us denote with $sigm$ the sigmoid function, with $\{\tilde{y}_i\}$ the training set of the classifier and with $\phi(y_i, y_j)$ a kernel function (in our case a Gaussian).

Then, after training and when presented with a novel observation $y(r)$, the RMV yields a prediction of the relevance of the observation $y(r)$ as

$$p(z = 1|y(r)) = sigm\left(\sum_i w_i^{rvm}\phi(y(r), \tilde{y}_i)\right), \quad (10)$$

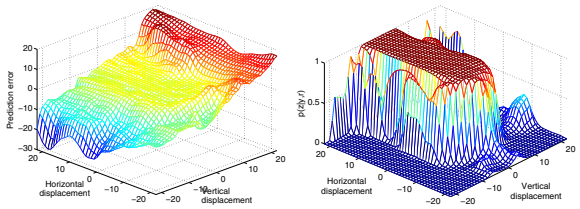
where w^{rvm} is a sparse weight vector that is learned in the training phase.

The training set $\{y(r)\}$ is constructed as follows. A candidate observation $y(r)$ is generated by artificially transforming (*e.g.* translating) the visual target with a transformation which we denote here with r. Then, for each of the candidate observations, a probabilistic prediction is made using Eq. 8. We put in the set of positive examples candidate observations for which, an appropriate norm of the difference between the true transformation r and the expected value (*i.e.* the mean) of the prediction $\hat{p}(x|y(r))$ is less than a threshold. That is,

$$\|r - E_x\left[\hat{p}(x|y(r))\right]\| < \theta_r \quad (11)$$

As $\hat{p}(x|y(r))$ is a mixture of Gaussians, the mean in the above equation can be obtained in closed form. Alternative schemes for constructing the positive training set, such as thresholding the distance between the true transformation $t(r)$ and the mode of $\hat{p}(x|y(r))$, or by thresholding the probability of the ground truth transformation $t(r)$ (*i.e.* $\hat{p}(t(r)|y(r)) > \theta_r$) are also possible. The set of the negative examples comprises the observations for which Eq. 11 is not satisfied. Other examples, such as observations from regions in the background could be also added in the negative training set. Clearly, the transformations r that generate the candidate training set need to explore larger parts of the state space than the ones that are used to construct the training set of the BME.

In Fig. 4, and for the toy example that we used in Fig. 1, we illustrate the true prediction error of an BME with 8 experts that has been trained to predict 2D displacements in the interval $[-11 \dots 11]$ and the corresponding 2D plot of $p(z|y, r)$. Note that we test with observations that result from displacements from both inside and outside the training interval. The RVM has been trained on positive examples that have been selected by thresholding the L_∞ error norm. It is clear that we can predict reasonably well which observations are associated with a low prediction error. Note, that not all observations that fall outside the training range give high prediction errors which indicates that the BME is capable of extrapolating.



(a) (b)

Figure 4. (a) Prediction error and (b) $p(z|y, r)$ as functions of the true displacement. In this example a Bayesian mixture of Experts was trained to predict displacements in the interval $[-11 \dots 11]$ for a template of size 10×10. An RVM was trained to classify observations in the interval $[-22 \dots 22]$ that can deliver accurate $(+/-2$ pixels) predictions of the true displacement

.

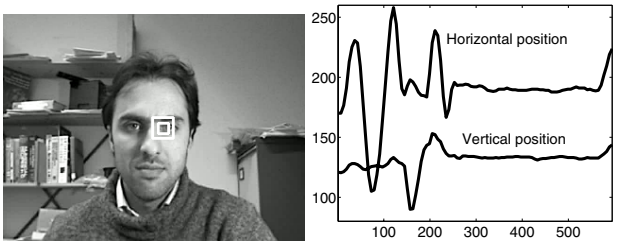
3. Experimental results

We have performed a number of experiments in order to illustrate the performance of the proposed method under different conditions, including occlusions, fast motion and moderate deformations. Here, we present quantitative and qualitative results for image sequences that are annotated by hand as well as comparative results with other methods in the literature. More specifically we compare our algorithm to discriminative tracking when a single observation is used (*e.g.* [11][9] and to the degenerate version of the proposed algorithm in which the data relevance determination mechanism is not used. We do not use any dynamic model, or temporal filtering in order to judge the performance when large deviations from the motion model are present. In addition we compared some of the easiest sequences (i.e. no occlusions) with a generative method that uses a particle filtering algorithm. In order to reduce the influence of the observation model in the comparison we used as likelihood the output of an RVM that was trained to recognise the target against templates that were extracted around it.

For all of our experiments, and for computational efficiency, we reduce the data dimensionality by applying Principal Component Analysis to the data with which the Bayesian Mixture of Experts (BME) is trained. Before being used, the training data for the BME, the training data for the RVM, and all test data are projected to the new space. In all of our experiments, unless explicitly stated otherwise, we tracked windows of 11×11 pixels. For training the BME we used pairs $(y(x), x)$ in which the observations $y(x)$ are produced by artificially transforming (*e.g.* translating) the visual target with the transformation x. We used translational transformations of up to 11 pixels, that is, transformations that generate observations $y(x)$ that had some overlap with the target. The examples that were used for training the RVM were generated using transformations with range 2-3

times the range that was used for training the BME. In order to reduce the complexity we perform a k-means clustering on the set of the candidate observations and train the RVM using the cluster centres. The label of a cluster (positive or negative) is determined by the majority of the labels of the examples that belong to it. Finally, in order to deal with changes in the illumination intensity we normalise the data by the average intensity in the window in question (before applying the PCA transform). For all the experiments we track 5 Gaussians (*i.e.* $M = 5$) and use 25 samples r (*i.e.* $R = 25$), unless stated otherwise.

We first present results for tracking a facial feature (*i.e.* an eye corner) under changes in the illumination, large head motion and deformations due to facial expressions and head rotations. In Fig. 5 we depict the windows that we used to track the eye corner and the ground truth position of the target. We have used 600 frames which are annotated every 6 frames. We have experimented by down-sampling the image-sequence spatially (by a factor $DSS = 1, 2$) and temporally ($DST = 1, 2, 3$) in order to create sequences with different motion magnitudes. In all cases we track an 11×11 window (the larger window in Fig. 5 is drawn to show the information that is used for tracking when $DSS = 2$).



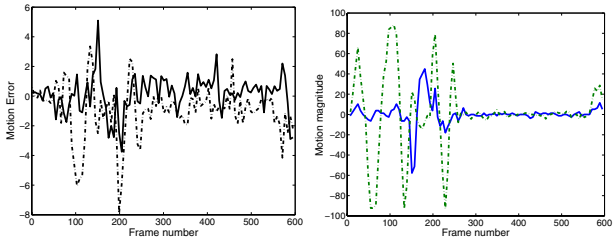
(a) (b)

Figure 5. (a) Target windows (b) Target position vs frame number for the 'Head' sequence

In Fig. 6 we present tracking results for the 'Head' sequence for some characteristic frames. The tracking is consistently good throughout the image sequence even at the presence of large motion (as percentage of the window size), occlusion and some deformations. In Fig. 7(a) we depict the horizontal and vertical components of the error in pixels and in Fig. 7(b) the motion magnitude as a percentage of the window size.

In order to illustrate the benefits of using multiple observations and the benefit of data relevance determination we present here comparatively results with two degenerate cases of our algorithm. The first (ALG1) is similar to classical regression-based tracking methods [9][2] that use a single observation. The second (ALG2) is a degenerate version of our algorithm in which the relevance determination is not used, that is, the probabilistic predictions of all

Figure 6. Tracking results for frames 75, 119, 163, 199, 357, 595 of the 'Head' sequence ($DSS = 1$ and $DST = 2$).



(a) (b)

Figure 7. (a) Motion error in pixels (RMS = 1.7) and (b) true motion horizontal and vertical component as percentage of the window size for the 'Head' sequence

candidate observation $y(r)$ are used. For both algorithms, we reinitialise the tracking at the ground truth position when the error is larger than 20 pixels (almost twice the window size). This gives a very conservative estimate of the cases that the validation scheme that is proposed by [9] will initiate a full scale detection. Re-initialisation is necessary for $ALG1$ but usually $ALG2$ can recover from tracking failures.

Alg. vs Params	Proposed	ALG1	ALG2	Cond.
DSS = 1, DST = 2	1.7	12	4.8	33
DSS = 1, DST = 3	4.8	12.1	5.8	-
DSS = 2, DST = 6	2.12	6.5	7.1	8.39
DSS = 2, DST=6 (HLF)	3.64	51.23	46.71	–

Table 2. RMS errors for the "Head" sequence

In Table 2 we summarise the RMS error for a number of different spatial and temporal sub-samplings of the original image sequence. Note the fact (Fig. 7(b)) that after frame 230 there is practically no motion (the sequence contains some facial expressions and closing/opening the eyes) which makes less acute the differences in the performance. For a baseline comparison, in the last column we present results for the modified condensation algorithm when using 150 particles. Finally, in the last row we present the results for the challenging case that both large persistent occlusions

and large motion are present. More specifically, we temporally subsample with a factor of 6 the sequence and artificially occlude half of the target. Two frames are presented in Fig. 8. While the proposed algorithm tracks the target by succesfully accessing the relevance of the information around the target, all other algorithms fail.

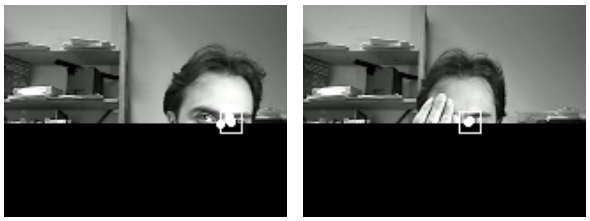


Figure 8. Tracking results for the "Head" sequence (last row Table 2

Similar differences in performance have been observed for a number of image sequences. Here we present results for sequences that we used to test the performance under large and persistent occlusions. First, we created an image sequence depicting a moving rigid object. In this sequence we manually annotated the position of the target every 6 frames and subsequently created image sequences in which up to half the visual target was occluded. In Fig. 9 we present some frames of the sequences that depict the occluded target and the estimated position of the target.

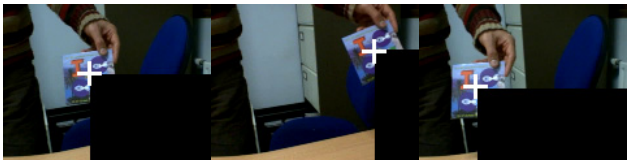


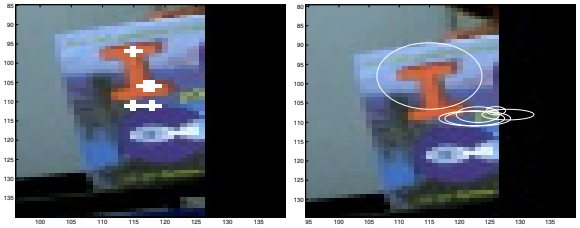
Figure 9. Tracking results for frames 1, 55 and 302 of the 'CD cover' sequence ($DSS = 4$ and $DST = 6$). A quarter (QRT) of the target is artificially occluded.

In Table 3 we summarise the results by reporting the RMS error for a number of subsamplings and for different occlusions of the target. In the first row, the target is completely visible, in the second a quarter of the target is occluded and in the last row half of the target is occluded. The target is occluded at the frames for which there is available annotation, that is every 6 frames. This means that in the experiment in the last row of Table 3 the target is completely visible in half of the frames and in the experiments in the 2nd and 3rd row it is occluded in all of the frames. A larger number of candidate observations are used here ($R = 50$). It is clear that the method is capable of tracking under partial occlussions and that it clearly outperforms the method that uses a single observation. For the latter, we used a more realistic re-initialisation scheme that is initiated when the true error is larger than 10 pixels (that is, almost equal to the template size). The results indicate that a validation scheme (as the one proposed in [9]) would fail (and therefore a full-search detection would be performed) in 38% of the frames when a quarter of the target is occluded, in 78% of the frames when half of the target is occluded and in 28% of the frames when half of the target is occluded but the target visible in half of the frames. Note, that when a large part of the target is occluded a validation scheme is more likely to fail even when the prediction is accurate. In this case the full scale detection scheme is also likely to fail.

Algorithm vs Parameters	Proposed	ALG1	ALG1 fails
DSS = 4, DST = 6	3.2	11.2	5 %
DSS = 4, DST = 6 (QRT)	4.2	17.8	38 %
DSS = 4, DST = 6 (HLF)	11.1	17.5	78 %
DSS = 4, DST = 3 (HLF)	2.9	19.2	28 %

Table 3. RMS errors for the 'CD Cover' sequence

Finally, in Fig. 10 (as in Fig. 8), we illustrate the ability of the algorithm to overcome large occlusions. In this case, observations that are located at areas neighbouring to the true target position are used to deliver reliable predictions of the target state. In Fig. 10(a) we depict the relevant observations and in Fig. 10(b) the corresponding probabilistic predictions (each ellipse represents a Gaussian). Note that our relevance determination scheme suppressed observations that were on the true target location, a result that indicates that a validation scheme using the trained RVM classifier would also fail.



(a) (b)

Figure 10. Tracking results for the last frame of the 'CD cover' sequence ($DSS = 4$ and $DST = 6$). Half of the target is artificially occluded. (a) Relevant observations (b) Probabilistic predictions.

4. Conclusions

In this paper we have presented a method for efficient and robust visual tracking. We propose a discriminative framework in which multiple observations provide predictions of the state of the target. Each prediction is moderated by the relevance of the corresponding observation, as this is determined by a probabilistic classification scheme. To

the best of our knowledge this is the first work that utilises multiple observations for discriminative tracking or uses a classification scheme to access in advance the relevance of an observation (as opposed to the *a posteriori* validation of the prediction). We have illustrated the efficiency of our approach in a number of image sequences for the problem of 2D tracking and in particular its ability to deal with large motion and with partial occlusions. For future work we intend to extend the proposed scheme for tracking 3D human pose under occlusions and background clutter.

Appendix A

In this appendix we will briefly outline a method for approximating a mixture of L Gaussians with a reduced M-component mixture. Our derivation, builds on the method of Vlassis and Verbeek [14] for learning a Gaussian Mixture from noisy data.

Let us denote with

$$f(x) = \sum_{l=1}^{L} p_l f(x|l)$$

the given L-component mixture, where $f(x|l) = \mathcal{N}(x_l, C_l)$, for $l = 1 \dots L$, is a Gaussian with known mean x_l and covariance C_l. Let us also denote with

$$p(x) = \sum_{m=1}^{M} \pi_m p(x|m)$$

the unknown M-component mixture, with $p(x|m) = \mathcal{N}(\mu_l, S_m)$, for $m = 1 \dots M$, is a Gaussian whose mean μ_m, covariance S_m and mixture coefficient π_m we seek to estimate.

As in [14] we minimise the Kullback-Leibler divergence between $p(x)$ and $f(x)$, by maximising an objective function that is a lower bound of the negative of the KL-divergence. Formally, we maximise

$$F = \sum_{l=1}^{L} \int dx\, dx\, f(x|l) \left\{ \log p(x) - KL_m \left[q_l(m) \| p(m|x) \right] \right\}, \tag{12}$$

where $q_l(m)$, for $l = 1 \dots L$, are auxiliary variational distributions that are introduced for bounding from below the negative of the KL-divergence between $p(x)$ and $f(x)$.

The update equations are identical to the ones derived in [14] in the case that $p_l = \frac{1}{L}$, and very similar to the update equations of the EM algorithm. More specifically the variational distributions q_l are updated as

$$q_l(m) \propto \pi_m p(x_l|m) \exp\left\{ -\frac{1}{2} Tr\left[S_m^{-1} C_l \right] \right\} \tag{13}$$

while the mixture components are updated as

$$\pi_m = \frac{\sum_{l=1}^{L} p_l q_l(m)}{\sum_{m=1}^{M} \sum_{l=1}^{L} p_l q_l(m)}, \quad \mu_m = \frac{\sum_{l=1}^{L} p_l q_l(m) x_l}{\sum_{l=1}^{L} p_l q_l(m)} \tag{14}$$

$$S_m = \frac{\sum_{l=1}^{L} p_l q_l(m) \left(x_l x_l^T + C_l \right)}{\sum_{l=1}^{L} p_l q_l(m)} - \mu_m \mu_m^T \tag{15}$$

References

[1] A. Agarwal and B. Triggs. A local basis representation for estimating human pose from cluttered images. In *Asian Conf. Computer Vision*, 2006.

[2] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(1), Jan. 2006.

[3] S. Avidan. Support vector tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(8):1064– 1072, Aug. 2004.

[4] J. Deutscher, A. Davison, and I. Reid. Automatic partitioning of high dimensional search spaces associated with articulated body motion capture. In *Int' Conf. Computer Vision and Pattern Recognition*, Dec. 2001. Kauai, Hawaii.

[5] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, Aug. 1981.

[6] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *Int'l Journal of Computer Vision*, 29(1):5–28, 1998.

[7] M. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, (6):181–214, 1994.

[8] F. Jurie and M. Dhome. Hyperplane approximation for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[9] O.Williams, A.Blake, and R.Cipolla. Sparse bayesian regression for efficient visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (8):1292–1304, Aug 2005.

[10] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *Int'l Conf. Computer Vision and Pattern Recognition*, 2004.

[11] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[12] M. Tipping. The relevance vector machine. In *Advances in Neural Information Processing Systems, San Mateo, CA*. Morgan Kaufmann, 2000.

[13] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.

[14] N. Vlassis and J. Verbeek. Gaussian mixture learning from noisy data. Technical report, Informatics Institute, University of Amsterdam, 2004. IAS-UVA-04-01.

[15] S. Waterhouse, D. MacKay, and T. Robinson. Bayesian methods for mixtures of experts. In *Advances in Neural Information Processing Systems*, volume 8, pages 351–357. The MIT Press, 1996.