Using an Adaptive VAR Model for Motion Prediction in 3D Hand Tracking

Desmond Chik
RSISE, Australian National University
Statistical Machine Learning, NICTA
desmond.chik@rsise.anu.edu.au

Jochen Trumpf
RSISE, Australian National University
jochen.trumpf@anu.edu.au

Nicol N. Schraudolph
nic@schraudolph.org

Abstract

A robust VAR-based (vector autoregressive) model is introduced for motion prediction in 3D hand tracking. This dynamic VAR motion model is learned in an online manner. The kinematic structure of the hand is accounted for in the form of constraints when solving for the parameters of the VAR model. Also integrated into the motion prediction model are adaptive weights that are optimised according to the reliability of past predictions. Experiments on synthetic and real video sequences show a substantial improvement in tracking performance when the robust VAR motion model is used. In fact, utilising the robust VAR model allows the tracker to handle fast out-of-plane hand movement with severe self-occlusion.

1. Introduction

Recent research into human motion tracking is mainly motivated by HCI (human computer interaction) applications and markerless motion capture. Tracking the 3D pose of the hand (and in fact any articulated structure, such as the human body) is a challenging problem. The high dimensionality of the hand, frequent self-occlusion and fast finger movements make tracking an ill-posed problem.

Motion prediction is an important aspect in tracking and often improves tracking performance. For particle filter based trackers, motion prediction plays a significant role in a smart redistribution of particles for the next frame, given the past motion history [6]. Analogously, motion prediction can be used to find the initial value for the optimisation routine in the next frame for gradient based trackers.

A variety of motion models have been proposed, ranging from anatomically correct dynamic models [13] to learned motion models derived from offline training on motion capture data [14, 11, 6, 1, 8, 12]. Learned motion models are popular and are generally described in a lower dimensional space [6, 14, 1, 11, 8, 12]. The rationale is that typical human motion lies in a subspace of the high dimensional angle-parameter space. In [1], PCA is used to reduce dimensionality, and an autoregressive motion model is trained on this reduced subspace. For applications where the set of motions being tracked is restricted, such a framework has been shown to be robust. However, a motion model learned this way does not generalise well to motions not observed in the training data. This is even more so if dimensionality reduction is used, since it is entirely possible that the new motion does not lie in the subspace the motion model is trained for. For a motion model learned offline to generalise well, one needs to ensure that the motion data used for training is not biased. This often means training over a large dataset that is rich enough, which can be impractical at times.

In this paper we introduce an online adaptive vector autoregressive (VAR) model for motion prediction. This is an attractive alternative in the sense that it is efficient to evaluate and does not require offline training. The VAR prediction model is highly adaptative via trust factors and, as an online algorithm, potentially generalises better to different hand movements.

The tracking system used to test the VAR prediction model is a gradient based tracker that works in the stochastic approximation framework [5]. The tracker is simple to implement and has the additional property of theoretical local convergence. Note that the proposed VAR prediction model is independent of the tracking system used and could for example be applied to particle filter based systems.

In section 2, the base tracking system is described. The VAR model is formulated in section 3. Section 4 details the initial tracking results for a synthetic sequence when the VAR prediction model is applied. The results are compared against the no-motion and the decelerating motion prediction model. In section 5, we introduce a robust adaptive version of the VAR motion prediction model. Experimental results for the real hand sequences are presented in 6. Concluding remarks are given in section 7.

2. The Tracking System

A stereo pair of cameras is used for tracking. The cameras point towards the hand being tracked in a convergent

978-1-4244-2154-1/08/$25.00

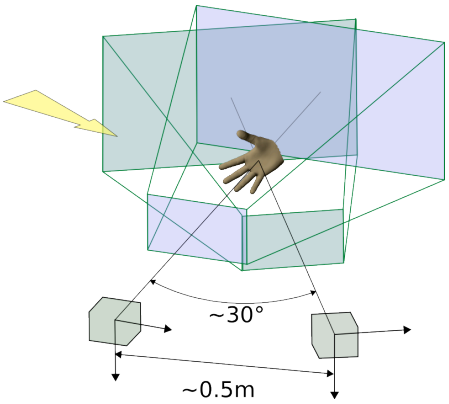

Figure 1. Diagram of the tracking system setup. A pair of cameras (square boxes) is placed in a convergent setup facing the hand. A strong light source (yellow arrow), that points towards the hand from above and behind one of the cameras, illuminates the scene.

setup as shown in figure 1.

A model-based approach is used where sample points taken from the tracker's hand model surface are projected onto the camera model image planes. At the corresponding pixel coordinates of the real images, a cost is evaluated based on the visual cues in the images. Mismatch errors are backpropagated as gradients to the model parameter space. An optimisation algorithm uses these gradients to refine the hand pose estimate. This is repeated for successive frames. A detailed deformable hand model with 26 degrees of freedom is used by the tracking system. This model is obtained from the 3D scanning of a real hand. Further details of the tracking system can be found in [5].

2.1. The Overall Cost Function

Our overall cost function is a modified version of that given in [5], in that a silhouette filling component has been added to improve tracking performance.

The filling cost function C_f penalises the tracker when the projection of the hand model does not completely fill the actual hand silhouette extracted from the camera images. A dense set of sample pixels is randomly chosen from the actual hand silhouette to evaluate C_f. Let $x \in \mathbb{R}^{26}$ be the set of hand model parameters. Also, let $\hat{s}_{i,j}$ be the pixel coordinate of the ith sample pixel chosen inside the actual hand silhouette in the jth camera view. Let $h_{i,j}(x)$ be the pixel coordinate of the point on the hand model projection that is closest to $\hat{s}_{i,j}$. Then the filling cost function over two camera views is given as

$$C_f(x) = \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{2}\frac{1}{2}||\hat{s}_{i,j} - h_{i,j}(x)||^2, \qquad (1)$$

where M is the cardinality of the dense set of sample points. Note from (1) that sample pixels which are covered by the projection of the hand model do not contribute to the filling cost since $\hat{s}_{i,j} = h_{i,j}(x)$ in such a situation.

Although beyond the scope of this paper, the modified overall cost function can be shown to inherit the theoretical local convergence property described in [5].

3. Motion Prediction Using a VAR Model

The tracker's local convergence property suggests that tracking performance depends heavily on the initial value used at each frame for the optimisation routine. Ideally, the starting set of model parameters $\hat{x}_t$ for the tth video frame should be initialised as close as possible to the optimal value x_t^*, such that the tracker starts within the basin of attraction.

In a data-driven approach, we treat the evolution of x_t^*, $(x_1^*, ..., x_T^*)$, over T video frames as a 26-dimensional multiple time series. Given this multiple time series $(x_1^*, ..., x_T^*)$ at time T, a VAR (Vector Autoregressive) model can be used to predict a suitable initial value $\hat{x}_{T+1}$ for the next frame $T+1$.

3.1. VAR Model Formulation

Consider a multivariate weakly stationary process Y_t and its realisation, a 26-dimensional multiple time series $(y_1, ..., y_T)$, up to time T. The weakly stationary property of Y_t requires the expectation $E(Y_t)$ and autocovariance $E[(Y_t - \mu_Y)(Y_t - \mu_Y)]$ to be time invariant. Let U_t be a 26-dimensional vector of white noise. The VAR(p) model [9] of this process is given as

$$Y_t = A_1 Y_{t-1} + ... + A_p Y_{t-p} + U_t, \qquad (2)$$

where p denotes the order of the VAR model and $A_1, ..., A_p$ are $\mathbb{R}^{26\times 26}$ parameter matrices.

Let X_t^* be the process that generates $(x_1^*, ..., x_T^*)$. X_t^* is not weakly stationary. If it were, it would imply that the hand pose in the video sequence is more or less stationary for all time t which is false. Hence one cannot directly apply a VAR model of X_t^* for motion prediction. Instead, differencing [4] is used to generate a weakly stationary process, Y_t^*, from X_t^*. Let Y_t^* be the process that generates model parameter 'accelerations' derived from X_t^*, *i.e.*,

$$Y_t^* = X_t^* - 2X_{t-1}^* + X_{t-2}^*. \qquad (3)$$

We assume Y_t^* to be a weakly stationary process. This is a reasonable assumption since a statistical analysis of Y_t^* for the video sequences reveals that the model parameter accelerations consistently fluctuate around zero, with $E(Y_t^*)$ being time invariant. The autocovariance of Y_t^* is also at least locally constant.

Thus, we model Y_t^* as a VAR process and use the VAR model to make a one-step ahead prediction $\hat{y}_{T+1}$ for frame

$T+1$ of the model parameter accelerations. This is done using the realisation-equivalent form of equation (2), *i.e.*

$$\hat{y}_{T+1} = A_1 y_T^* + ... + A_p y_{T-p}^* + u_T. \quad (4)$$

In practice, u_T is taken to be the expectation $E(U_t) = 0$, since it is usually unknown. $\hat{y}_{T+1}$ is then mapped back to $\hat{x}_{T+1}$ via

$$\hat{x}_{T+1} = \hat{y}_{T+1} + 2x_T^* - x_{T-1}^*. \quad (5)$$

In general, the tracking system does not know $(x_1^*, ..., x_T^*)$ as they are the ground truth values of the hand model parameters. The best that one can do is to use the corresponding tracker estimates $(x_1, ..., x_T)$ in place of $(x_1^*, ..., x_T^*)$ for motion prediction. Hence a more accurate model would be

$$\begin{aligned} Y_t^* &= A_1 Y_{t-1}^* + ... + A_p Y_{t-p}^* + U_t & (6)\\ Y_t &= Y_t^* + CV_t, & (7) \end{aligned}$$

which can be rewritten as

$$\begin{aligned} Y_t^* &= A_1 Y_{t-1} + ... + A_p Y_{t-p} + & (8)\\ &\quad (U_t - (A_1 C V_{t-1} + ... + A_p C V_{t-p})), & (9) \end{aligned}$$

where V_t is assumed to be a noise vector and C a noise-mixing matrix. However this ARMA-type model is difficult to solve as the statistics of V_t is unknown. We opt instead to use the VAR(p) model described in (2) for motion prediction. This is equivalent to treating the $U_t - (A_1 C V_{t-1} + ... + A_p C V_{t-p})$ term in (9) as white noise.

Doing this will obviously produce a less accurate prediction $\hat{y}_{T+1}$ and the corresponding prediction $\hat{x}_{T+1}$ for the next frame $T+1$. But it is important to reiterate again that we are *not* trying to find x_{T+1}^* using a VAR motion prediction model. Our prediction $\hat{x}_{T+1}$ merely provides a better initial value for the stochastic optimisation routine, which is set to find x_{T+1}^*. It is sufficient for our prediction $\hat{x}_{T+1}$ to be close enough to x_{T+1}^* such that starting the optimisation routine at $\hat{x}_{T+1}$ for frame $T+1$ is better than starting at x_T *i.e.* without motion prediction. In fact, as we will see later in the experimental results, using this simplified model is already enough to improve tracking performance.

3.2. Estimation of VAR Parameters

Given the past observations $(y_1, ..., y_T)$ up to time T, the VAR parameters $A_1, ..., A_p$ are estimated via least squares estimation [9]. Note that the least squares estimator has the additional interpretation of being a maximum likelihood estimator if one assumes U_t to be gaussian. Let

$$\begin{aligned} \mathbf{Y} &:= (y_1, ..., y_T) & (10)\\ \mathbf{B} &:= (A_1, ..., A_p) & (11)\\ W_t &:= \begin{pmatrix} y_t \\ \vdots \\ y_{t-p+1} \end{pmatrix} & (12)\\ \mathbf{W} &:= (W_0, ..., W_{T-1}) & (13) \end{aligned}$$

Then the least squares estimate, $\hat{\mathbf{B}}$, of $\mathbf{B}$ is given as

$$\hat{\mathbf{B}} = \mathbf{Y}\mathbf{W}'(\mathbf{W}\mathbf{W}')^{-1}. \quad (14)$$

In our case, we solve for $\hat{\mathbf{B}}$ online. That is, $\hat{\mathbf{B}}$ is re-evaluated with each new y_t obtained as the tracking system completes another frame in the video sequence. Therefore our prediction $\hat{y}_{T+1}$ of the accelerations of the hand model parameters for the next frame $T+1$ is

$$\hat{y}_{T+1} = \hat{\mathbf{B}}_T W_T. \quad (15)$$

An issue that is problematic in practice is that the computational cost for $\hat{\mathbf{B}}_T$ increases as new y_t's are obtained. To address this, we adopt a limited memory version of the least squares estimator by setting $Y := (y_{T-N}, ..., y_T)$ for a reasonably sized N. In other words, we drop the oldest sample y_{T-N} upon receiving a new sample y_{T+1}. Testing on video sequences shows that having a limited memory actually improves the prediction results. This is because a limited memory implementation caters better for instances of volatility clustering where the autocovariance is locally constant, but can be observed to vary over longer time periods. Using a limited memory least squares estimator essentially means we are only using a local portion of the time series, which is more likely to have a constant autocovariance, thereby more likely to satisfy the weak stationarity assumption for a VAR model.

3.3. Performance Evaluation

A measure $\tilde{R}^2$ is introduced to quantify the improvement in tracking peformance when a VAR motion prediction model has been added to the tracking system. $\tilde{R}^2$ is loosely based on the R^2 measure used in regression analysis [2].

Let $y_t^*(i)$ be the ground truth time series for the ith hand model parameter in the acceleration domain. Also, let $y_t^0(i)$ be the times series produced by the tracking system without motion prediction, *i.e.* where the motion prediction for the next time step $t+1$ is

$$\hat{x}_{t+1}^0(i) = x_t^0(i), \quad (16)$$

and thus the corresponding $\hat{y}_{t+1}^0$ is

$$\hat{y}_{t+1}^0(i) = -x_t^0 + x_{t-1}^0. \quad (17)$$

Similarly, let $y_t^{\text{VAR}}(i)$ be the corresponding time series estimated by the tracking system with motion prediction via a VAR model. Given a video sequence of length τ in total, the $\tilde{R}_y^2(\tau, i)$ measure for the ith hand model parameter in the acceleration domain is defined as

$$\tilde{R}_y^2(\tau, i) := 1 - \frac{\sum_{t=1}^{\tau}(y_t^*(i) - y_t^{\text{VAR}}(i))^2}{\sum_{t=1}^{\tau}(y_t^*(i) - y_t^0(i))^2}, \quad (18)$$

evaluated over the entire sequence, from $t = 1, ..., \tau$. $\tilde{R}^2_y(\tau, i)$ rates the improvement in tracking accuracy by comparing the squared sum of residuals between $y^{\text{VAR}}_t(i)$ and the ground truth against the residuals between the no-prediction time series $y^0_t(i)$ and the ground truth.

Measuring these residual errors in the acceleration domain of the hand model parameters is less indicative of tracking performance than measuring residual errors of the hand's joint and fingertip positions in 3D space. Hence, let G be the function that maps the hand model parameters x_t to hand joint/fingertip positions of the hand z_t. Specifically,

$$G : \mathbb{R}^{26} \rightarrow \mathbb{R}^{3\times 21} \tag{19}$$

$$z_t = G(x_t). \tag{20}$$

The columns of z_t are the joint/fingertip positions in 3D space (there are16 joints and 5 fingertips in total). Let $z_t(j)$ be the 3D position of the jth joint/fingertip at frame t. Then $\tilde{R}^2(\tau, j)$ is defined as

$$\tilde{R}^2(\tau, j) := 1 - \frac{\sum_{t=1}^{\tau} ||z^*_t(j) - z^{\text{VAR}}_t(j)||^2}{\sum_{t=1}^{\tau} ||z^*_t(j) - z^0_t(j)||^2}. \tag{21}$$

As a guideline, $\tilde{R}^2(\tau, j) > 0$ indicates that the tracking estimate with the help of motion prediction is more accurate than that without motion prediction, for the jth joint/fingertip. The converse is true if $\tilde{R}^2(\tau, j) < 0$.

4. Initial Experiments

All the different types of VAR models examined in this paper are tested on a synthetic sequence (175 frames, 640×480 pixels) generated from real hand movements. The synthetic sequence provides ground truth values for a quantitative analysis of tracking performance. It contains elements of typical hand movements such as palm rotations and finger articulation.

Each experiment is repeated over 50 trials. As mentioned previously, the VAR parameters $\hat{\mathbf{B}}_\mathbf{t}$ are evaluated online for each frame t; no offline training is involved. The tracker is manually initialised to a good starting position for the first frame of the video sequence. About 600 sample points are used to track the moving hand. Following [5], SMD (stochastic meta-descent) [10, 3] is the optimisation algorithm used, with the parameters $\mu = 0.1$, $\lambda = 0$ and the initial step sizes $p = 0.2$ for the parameters of the finger/thumb joints, $p = 0.4$ for the palm's rotation parameters, $p = 2.5$ for the palm's translation parameters. The weight on C_f, the filling cost function, is 0.4 while the weight on C_s, the silhouette cost function, is 1.7. Optimisation for each frame terminates when either the overall cost reaches below a threshold of 0.006 or a maximum of 50 iterations is reached.

On average, 3.7 seconds are required to track each frame on a P4 3.4 GHz machine, of which 2.5 seconds are spent on the optimisation routine and the remaining 1.2 seconds are spent on the image pre-processing procedures. Additional computation time for motion prediction evaluation is comparatively negligible, varying between 2-3 orders of magnitude below the time taken by the optimisation routine.

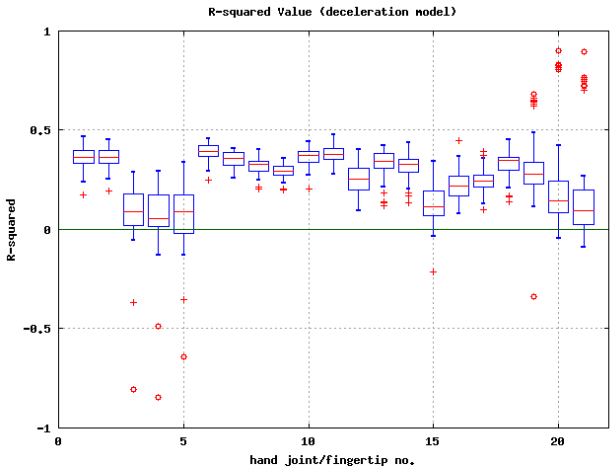

Figure 2. $\tilde{R}^2(\tau)$ of z^{D}_t for each hand joint/fingertip over the 50 trials. Outliers that lie between 1-3 times interquartile range are marked as '+' while those that lie over 3 times the interquartile range are marked as 'o'.

The results of two control experiments are used as benchmarks. The first control experiment is tracking without motion prediction. We denote the multiple time series of the resulting joint/fingertip positions from this experiment as z^0_t. The other control experiment uses the popular decelerating motion prediction model; let $z^{\text{D}}_t = G(x^{\text{D}}_t)$ be the corresponding multiple time series. The predictor $\hat{x}^{\text{D}}_{t+1}$ is defined as

$$\hat{x}^{\text{D}}_{t+1} = x^{\text{D}}_t + \rho(x^{\text{D}}_t - x^{\text{D}}_{t-1}), \tag{22}$$

where $\rho \in [0, 1]$. Thus the corresponding $\hat{y}^{\text{D}}_{t+1}$ is

$$\hat{y}^{\text{D}}_{t+1} = (\rho - 1)(x^{\text{D}}_t - x^{\text{D}}_{t-1}). \tag{23}$$

Note that $\rho = 1$ gives the constant velocity model. However, tracking with the constant velocity model is found to be unstable. Testing on a range of ρ values show that setting $\rho = 0.4$ produces the best tracking performance. This will be used in our comparison with the VAR models.

Figure 4 shows the overall mean error, taken to be the average of all joint/fingertip errors over all 50 trials (see [5] for details of the measure) for both the standalone tracker and the tracker with the deceleration predictor. There are three noticeable error peaks in the video sequence. The first peak (frame 73) represents the tracking inaccuracy due a misfit of the little finger (see figure 6). The second peak (frame 100) corresponds to the part where the hand undergoes a global rotation. The third peak (frame 150) occurs as the thumb

move across the palm, partially occluding the fingers and the palm.

One can already see the improvement with the naive prediction model, especially between frames 100 to 130. This improvement is reflected in the $\tilde{R}^2$ value of the entire sequence for z_t^{D} in figure 2. Excluding several outliers for the thumb, there is a noted improvement over tracking without motion prediction.

The following subsections describe our results for the different variations of the VAR (vector autoregressive) model tested, namely the traditional full-VAR model and a structured-VAR model where the kinematic relations of the hand are accounted for.

4.1. VAR Model Order Selection

VAR model order selection based on the AIC or BIC measure is a standard procedure [9]. This is meaningful when the training time series x_t is fixed while the various VAR models are fitted during the criterion evaluation process. A complication in our situation is that the time series is dependent on the VAR model used. Different VAR models produce different predictions of the initial value for the finite-length optimisation routine, which leads to different evolutions of x_t.

At best, the AIC and BIC criteria can only give a rough initial value for order selection. Initial AIC and BIC tests evaluated on the static ground truth sequence tend to favor VAR models of higher orders (> 5). However, the tracking performances of the tracker with these motion predictors of higher order are substantially worse. Only the results of the 1st order variants of the VAR model are shown; the results of the higher order models are omitted as they are consistently worse for each case.

4.2. Full-VAR model

Recall that the VAR(1) predictor $\hat{y}_{T+1}$ for the hand model parameters in the acceleration domain is given as

$$\hat{y}_{T+1} = \hat{A}_1 y_T, \tag{24}$$

where $\hat{A}_1$ is obtained directly by solving (14). We denote the multiple time series generated by the tracking system using this motion predictor in the hand joint/fingertip domain as z_t^{FV}. Initial tests reveal that all the VAR-based prediction models tend to give large error deviations when the change in acceleration is abrupt, leading to gross tracking inaccuracies. To mitigate this, a hard threshold has been set such that the difference between $\hat{x}_{T+1}$ and x_T cannot be more than 10 degrees.

The tracking result z_t^{FV} is in general worse than z_t^0. To understand why this is so, one should note that $\hat{A}_1$ quantifies the correlations not the causation (*i.e.* tendon forces controlling the hand) in the joint movements. The limited-memory sample set (memory size $N = 150$) used to refine $\hat{A}_1$ online is simply not rich enough and represents a biased sample of all possible hand movements. This leads to inaccurate predictions. Additional prior information, in the form of constraints, is needed for solving $\hat{A}_1$ sensibly.

4.3. Structured-VAR model

In this situation, $\hat{A}_1$ is solved under the constraints induced by the hand's kinematic structure. We observe that the movements of joints along the kinematic chain of each digit are correlated, and that their dependency on each other allows for the flexion of each digit. We then assume that the movement of each digit is independent of other digits. In addition we assume that the rotation and translation movements of the palm are independent of each other and are also independent of the movement of each digit.[1] Applying these priors constrains $\hat{A}_1$ to a block diagonal form

$$\hat{A}_1 = \begin{pmatrix} \mathbf{G_{6\times 6}} & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{R^{Y}_{5\times 5}} & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{R^{1}_{3\times 3}} & 0 & \vdots \\ \vdots & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \mathbf{R^{5}_{3\times 3}} \end{pmatrix}, \tag{25}$$

where $\mathbf{G_{6\times 6}}$ is a diagonal matrix that relates to the translation and rotation parameters of the palm. $\mathbf{R^{Y}_{5\times 5}}$ is a diagonal matrix that relates to the rotation parameter of each digit that models the abduction/adduction of each digit. Lastly, $\mathbf{R^{k}_{3\times 3}}$ relates to the rotation parameters of the kth digit responsible for the flexion of the digit.

The multiple time series in the hand joint/fingertip domain generated by the tracking system using this motion predictor shall be denoted z_t^{SV}. A limited memory size of $N = 30$ has been used for the structured-VAR model.

Enforcing kinematic constraints on $\hat{A}_i$ results in better tracking performance than when the full-VAR model is used. The first two error peaks that have been observed in both control experiments are dampened when the structured-VAR motion predictor is applied (see figure 4). The dampened errors are reflected in the $\hat{R}^2$ values for z_t^{SV}, where the tracking accuracy of the palm joint (joint/finger no. 1) and the joints on the little finger (joint/fingertip no. 18-21) has drastically improved. Errors for the middle finger have worsened, in particular the fingertip (no. 13), resulting in the enlarged third peak in figure 4.

Compared to the deceleration predictor, adding the structured-VAR predictor to the tracker increases the variance of the tracking performance. This is generally undesirable. Then again, having a larger variance does mean

[1] These assumptions are rough at best, *e.g.*, it is well known that the flexion of the little finger and the ring finger are not completely independent. [13] would be a more realistic (and sophisticated) model. For our application however, we find that our simple model already works well.

the tracker has a better chance of escaping from local minima. The extended upper tails of the $\tilde{R}^2$ distributions for the thumb's PIP joint (no. 4) and the thumb tip (no. 5) attest to this.

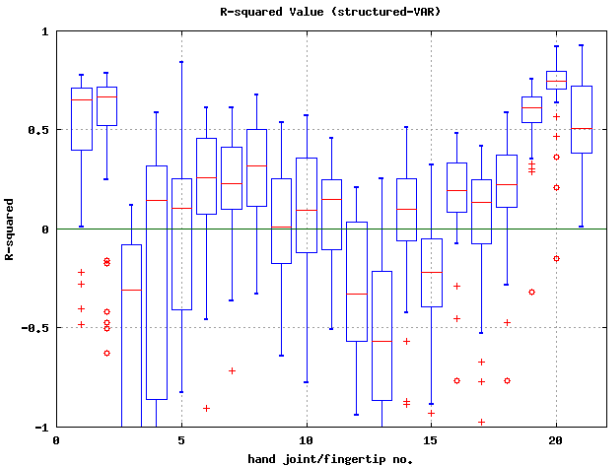

Figure 3. $\tilde{R}^2(\tau)$ of z_t^{SV} shows a large improvement for the base joint and the little finger.

5. Robust VAR

An ideal motion predictor should have the high performances of the structured-VAR model observed for certain joints, augmented with the general consistency and low variance of the deceleration predictor.

To achieve this, we combine the structured-VAR predictor and the deceleration predictor (with $\rho = 1$), tempered with adaptive trust factors. We shall refer to this adaptive scheme as the RVAR (Robust Vector Autoregressive) prediction model.

The RVAR model interpolates online a weighted combination of $\hat{x}^{\text{SV}}_{T+1}$, $\hat{x}^{\text{D}}_{T+1}$ and $\hat{x}^{0}_{T+1}$ predictions. The interpolation is based on the reliability of the past predictions for each predictor when compared against the past history $x_{T-N}...x_T$ of tracking results. The RVAR predictor for the ith hand model parameter $\hat{x}_{T+1}(i)$ is defined as

$$\hat{x}^{\text{RVAR}}_{T+1}(i) = (1-\gamma)\hat{x}^{0}_{T+1}(i) + \gamma(\alpha_i \hat{x}^{\text{SV}}_{T+1}(i) + \beta_i \hat{x}^{\text{D}}_{T+1}(i)), \tag{26}$$

where $\gamma \in [0,1]$, $\alpha_i + \beta_i \leq 1$, and $\alpha_i, \beta_i \geq 0$.[2] Using motion prediction to find a good initial value for the optimisation routine can be tricky in that there is always an inherent danger of overshooting. Hence we introduce γ, an upper bound on how much one should trust the motion predictions. γ is fixed throughout the tracking sequence. α_i and β_i are adaptive weights that minimise the following sum

$$E_i(T, \alpha, \beta, i) = \sum_{t=T-c}^{T} k_t [x_t(i) - (\alpha \hat{x}_t^{SV}(i) + \beta \hat{x}_t^{\text{D}}(i))]^2, \tag{27}$$

where c is a cut-off constant and k_t is a decaying factor defined as

$$k_t = \frac{1}{2^{T-t}}. \tag{28}$$

Minimising $E_i(T)$ gives the optimal α_i and β_i for $x_t(i)$ at frames $t \in [T-c, T]$. These weights are then used in (26) when the prediction $\hat{x}^{\text{RVAR}}_{T+1}(i)$ for the next unknown frame $T+1$ is made. $E_i(T)$ is minimised separately for each hand model parameter i to obtain separate sets of α_i and β_i weights.

The absolute interpolation of the predicted model parameters in (27) is just one of many interpolation approaches. For example, one could instead choose to interpolate the changes in predicted model parameters relative to the position in the last frame *i.e.* $\hat{x}^{\text{SV}}_t(i) - x_{t-1}(i)$. Note that the update equation (26) will change according to the interpolation approach.

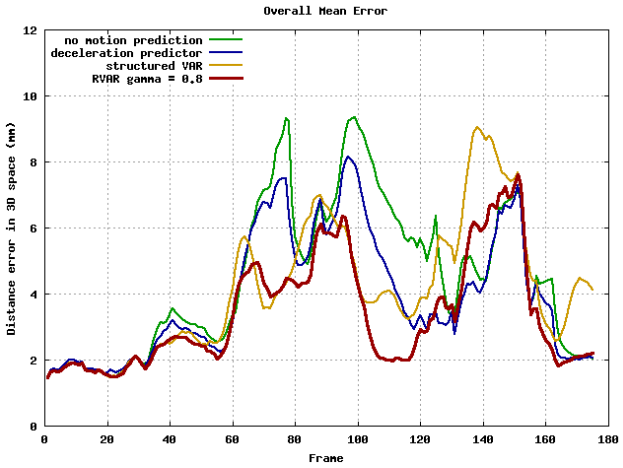

Figure 4. Overall mean error for RVAR model (red) with γ = 0.80, and cut-off constant $c = 8$. Tracking performance without motion prediction (green), with the deceleration predictor (blue) and with a structured-VAR model (yellow) is shown for comparison.

Results for the RVAR predictor (with $\gamma = 0.8$, $c = 8$) show a marked improvement over both the deceleration and the structured-VAR predictor (see figure 4). From figure 5, one can appreciate that the RVAR predictor has inherited the desired low variance property of the deceleration predictor while retaining the high performances of the structured-VAR model.

The $\tilde{R}^2$ values for most joints/fingertips reside around 0.5. With the exception of the middle finger's PIP joint (no. 12) and possibly the thumb joints/tip (no. 3-5), the

[2] Note again that the no-motion predictor $\hat{x}^{0}_{T+1}(i) = x_T(i)$.

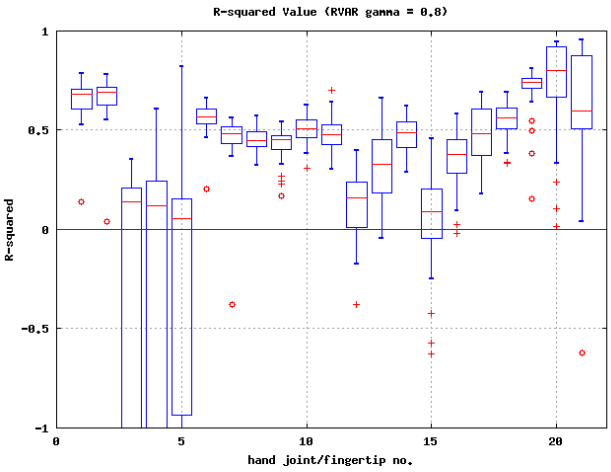

Figure 5. $\tilde{R}^2(\tau)$ for the RVAR model

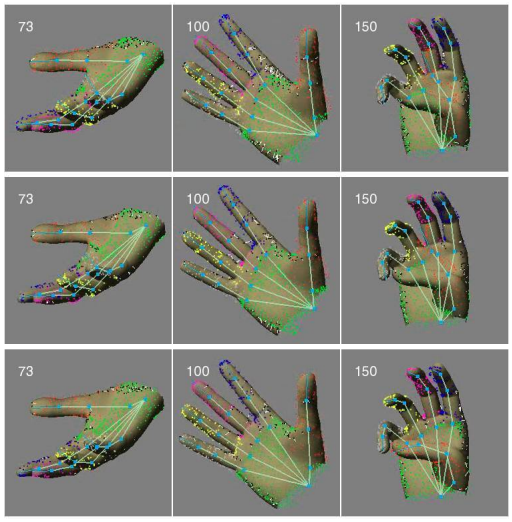

Figure 6. Selected frames where the overall mean error peaks. Top row are tracking results with no motion prediction, middle row corresponds to tracking with the deceleration predictor, and the last row corresponds to tracking with RVAR motion prediction.

RVAR model gives better performance over the deceleration model. Although the variance of the thumb joints are comparatively larger, the median of $\tilde{R}^2$ for the RVAR model is actually slightly higher. Tuning $\gamma = 0.7$ results in a variance (for the thumb joints) that is similar to the variance of the deceleration case, without noticeable change to the statistics of the other joints/fingertips. We prefer keeping the higher variance for the thumb (*i.e.* with $\gamma = 0.8$) since it gives the tracker a better chance to escape from local minima. Frame 150 in figure 6 illustrates where using a RVAR model sometimes allows the tracker to escape from a local minimum whereas this is impossible with the deceleration model.

6. Experiments on Real Sequences

A pair of firewire cameras with a resolution of 640×480 pixels is used to capture video sequences of a moving hand at 25 frames per second. The silhouette of the hand is extracted via a learned skin colour model [7]. The initial pose for the starting frame is crudely fitted by eye. Tracking performances with the $\hat{x}^{\text{RVAR}}_{T+1}$, $\hat{x}^{\text{D}}_{T+1}$ and $\hat{x}^{0}_{T+1}$ motion predictors are examined.

The first sequence is 280 frames long and starts with individual flexion of fingers followed by simultaneous finger flexions. A visual inspection of the tracking results indicates that tracking without motion prediction performs worst. Figure 7 contains excerpts of the video sequence and compares the performances of the tracker using the three different motion predictors. The tracker with the no-motion predictor temporarily loses tracking accuracy during fast motion sequences, *e.g.*, frames 86, 111, as expected. The tracking performance of RVAR (with $\gamma = 0.8$, $c = 8$) is also more accurate than the deceleration predictor (see frames 111, 203, 236 and 242).

To verify that the RVAR motion predictor is indeed better than the deceleration predictor, we use a more challenging sequence (80 frames), where the hand undergoes an out-of-plane movement as the palm face completely flips away from the camera and back (see figure 8). The tracker with the RVAR motion predictor is able to follow through with the movement whereas the deceleration predictor is unable to do this, despite repeated trials.

7. Conclusion

An online adaptive VAR prediction model has been presented. The results clearly indicate that using an online VAR motion predictor naively to find a good initial starting point for the optimisation routine of the tracking system will degrade tracking performance. Applying kinematic-based constraints when solving for the VAR parameters is required for more sensible predictions. The addition of a trust factor and adaptive weights based on the past prediction accuracy has improved results. Experimental results on real sequences support these findings and demonstrate the flexibility of the RVAR model in handling different hand motions. Future work will include a comparison of the RVAR model against motion models derived from offline training on motion capture data.

Acknowledgements

We wish to thank Manfred Deistler for the insightful discussions and the anonymous reviewers for their helpful suggestions. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

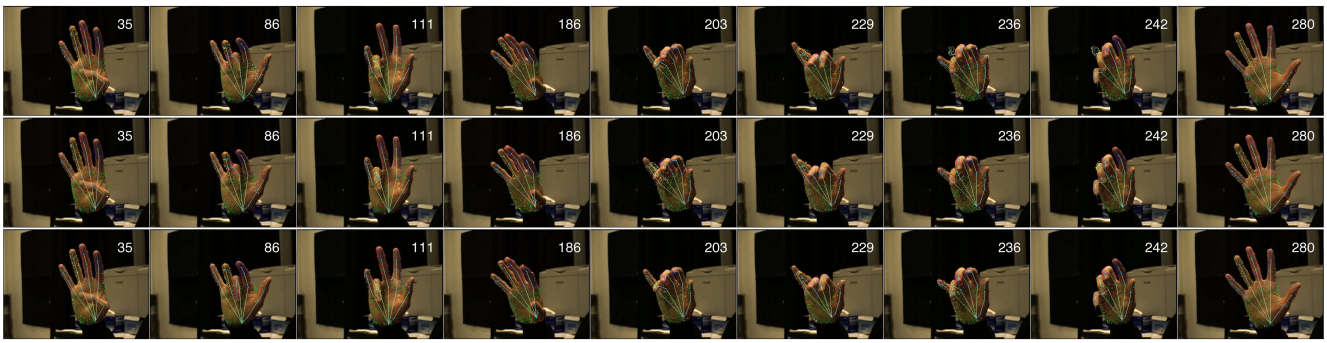

Figure 7. Tracking results for the real sequence. Top row: No-motion predictor. Middle row: Deceleration predictor. Bottom row: RVAR predictor. Frames 86 and 111 illustrate the advantage of the RVAR predictor for fast flexion of the fingers. In general, the tracking results are cleaner with the RVAR predictor.

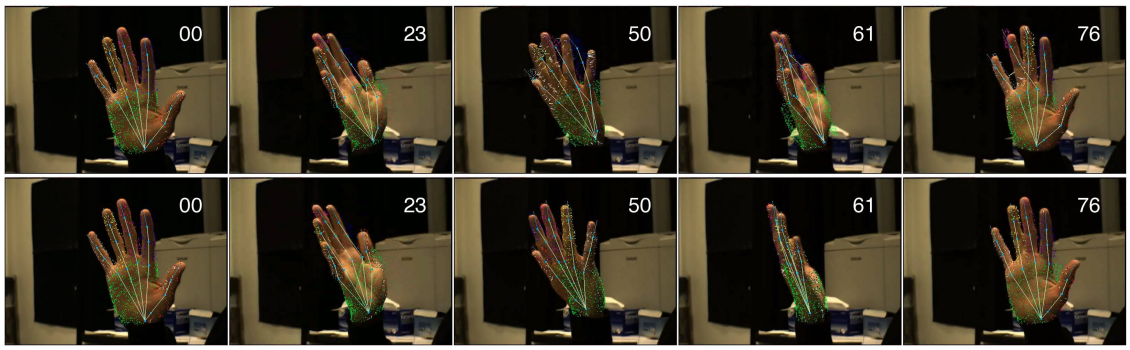

Figure 8. Difficult out of plane twist motion that involves severe self-occlusion. Top row: Tracker with the deceleration predictor consistently fails. Botton row: Tracker with the RVAR predictor is able to track through the gesture correctly.

References

[1] A. Agarwal and B. Triggs. Tracking articulated motion using a mixture of autoregressive models. In *European Conference on Computer Vision*, pages Vol III: 54–65, 2004.

[2] A. Atkinson and M. Riani. *Robust Diagnostic Regression Analysis*. Springer, 1st edition, 2000.

[3] M. Bray, E. K. Meier, N. N. Schraudolph, and L. J. V. Gool. Fast stochastic optimization for articulated structure tracking. *Image and Vision Computing*, 25(3):352–364, Mar. 2007.

[4] P. Brockwell and R. Davis. *Introduction to Time Series and Forecasting*. Springer, 2nd edition, 2002.

[5] D. Chik, J. Trumpf, and N. N. Schraudolph. 3d hand tracking in a stochastic approximation setting. In *2nd Human Motion Workshop - Understanding, Modeling, Capture and Animation*, pages 136–151, 2007.

[6] S. Hou, A. Galata, F. Caillette, N. Thacker, and P. Bromiley. Real-time body tracking using a gaussian process latent variable model. In *International Conference on Computer Vision*, 2007.

[7] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, Jan. 2002.

[8] M. Kato, Y. W. Chen, and G. Xu. Articulated hand tracking by PCA-ICA approach. In *International Conference on Automatic Face and Gesture Recognition*, pages 329–334, 2006.

[9] H. Lutkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, 1st edition, 2006.

[10] N. N. Schraudolph. Local gain adaptation in stochastic gradient descent. In *ICANN*, pages 569–574, Edinburgh, Scotland, 1999. IEE, London.

[11] H. Sidenbladh, M. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *European Conference on Computer Vision*, pages 784–800, 2002.

[12] B. D. R. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 28, pages 1372–1384, Sept. 2006.

[13] W. Tsang, K. Singh, and E. Fiume. Helping hand: an anatomically accurate inverse dynamics solution for unconstrained hand motion. In *SCA '05:ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 319–328, 2005.

[14] H. N. Zhou and T. S. Huang. Tracking articulated hand motion with eigen dynamics analysis. In *International Conference on Computer Vision*, pages 1102–1109, 2003.