ON VISUAL OBJECT TRACKING USING ACTIVE APPEARANCE MODELS

M.R. Hoffmann, B.M. Herbst, and K.M. Hunter

Applied Mathematics, University of Stellenbosch, Private Bag X1, Matieland, 7602, South Africa

Abstract: Active appearance models provide an elegant framework for tracking objects. Using them in a deterministic algorithm to perform tracking is not robust enough since no history is used of the object's movement and position. We discuss two approaches to rectify this situation. Both techniques are based on the particle filter. The first technique initialises the active appearance model search algorithm with a shape estimate obtained from an active contour tracker. A combination of a particle filter and an active appearance model forms the foundation for the second technique. Experimental results indicate the effectiveness of these techniques.

Key Words: active appearance models, particle filter, tracking

1. INTRODUCTION

Active appearance models (AAMs) [1] provide a neat model-based framework for tracking objects. They incorporate both shape and texture into their formulation, hence they enable us to track simultaneously the outline of an object as well as its appearance. It is therefore easy to use the parameters provided by an AAM tracker in other applications.

Stegmann [2] demonstrated that AAMs can be successfully applied to perform object tracking. In his deterministic approach, the AAM search algorithm is applied successively to each frame. However, this technique is not robust since the optimisation techniques employed in the AAM search algorithm only explores a small, local region of interest. No history of the object's movement and position is used to improve the optimisation. Therefore the tracker fails when the object moves fast for example. The reason for this failure is that the sudden jumps caused by fast movements lead to a bad initialisation for the AAM optimisation routines.

In this paper, two techniques are discussed which can be used to make the deterministic AAM tracker more robust. The first technique is based on active contours. It initialises the AAM using the shape estimate obtained from active contours. In this way, the AAM search algorithm is narrowed down to a local region of interests around the estimate from active contours. Since the active contours algorithm is able to track an object robustly, it is a better initialisation for the AAM. The second technique uses a combination of a particle filter and an AAM to provide more robustness. Here temporal filtering predicts the parameters of the AAM so that the history of the object's movement and position enhances the AAM searches.

Central to both techniques is a particle filter. Particle filters have become an important tool to track objects. They have been used in conjunction with edge measurements (active contours) [3], colour histograms [4, 5] and even a combination of the aforementioned [6]. Particle filters [7, 8] can deal with non-linear systems and non-Gaussian models, and can therefore be viewed as a generalisation of the Kalman filter.

The rest of the paper is organised as follows: we briefly summarise active appearance models in Section 2, followed by a summary of particle filters in Section 3. The first technique to enhance the standard AAM tracker is detailed in Section 4, and the second technique in Section 5. Experimental results are shown in Section 6. We conclude in Section 7.

2. ACTIVE APPEARANCE MODELS (AAMS)

Active appearance models [1, 9] are template based and employ both shape and texture. Using them consists of an initial training phase to learn the parameters for an object (e.g. $\boldsymbol{\Phi}_s$, $\overline{\boldsymbol{s}}$ in Equation 1 below) and a search phase to extract the object in a new image.

In the training phase of AAM, the shape of the modelled object is defined by a vector of feature points on the outline of the object, $\boldsymbol{s}_i = [x_{i1}, \cdots, x_{in}, y_{i1}, \cdots, y_{in}]^T$, $i = 1, \ldots, l$ where l is the number of training images and n the number of feature points. The shapes are normalised with respect to translation, scale and rotation and this is done by setting up a common coordinate reference. Then all the shapes are aligned to this reference giving rise to a parameter known as the pose $\boldsymbol{p}$. Using the pose and the common coordinate reference, an aligned shape (now in relative coordinates) can be translated, scaled and rotated to its original version (in absolute coordinates). The texture of the object in question is described by a vector, $\boldsymbol{g}_i = [g_{i1}, g_{i2}, \ldots, g_{im}]^T$, $i = 1, \ldots, l$ with m the number of texture points.

Typically a piece-wise affine warp based on the Delaunay triangulation is used to collect the texture points. Principal component analysis (PCA) is performed on the aligned shapes and textures and this yields

$$\boldsymbol{s} = \overline{\boldsymbol{s}} + \boldsymbol{\Phi}_s \boldsymbol{b}_s \quad (1)$$

$$\boldsymbol{g} = \overline{\boldsymbol{g}} + \boldsymbol{\Phi}_g \boldsymbol{b}_g \quad (2)$$

where $\boldsymbol{\Phi}_s$ and $\boldsymbol{\Phi}_g$ are the eigenvector matrices of the shape and texture covariance matrices respectively and $\boldsymbol{b}_s$ and $\boldsymbol{b}_g$ are the PCA projection coefficients.

A combined model parameter $\boldsymbol{c}$ is obtained by combining the PCA scores into $\boldsymbol{b} = [\boldsymbol{\Psi}_s \boldsymbol{b}_s,\ \boldsymbol{b}_g]^T$ and performing a third PCA

$$\boldsymbol{b} = \boldsymbol{\Phi}_c \boldsymbol{c}, \quad (3)$$

where $\boldsymbol{\Psi}_s$ is a weighting matrix between pixel intensities and pixel distances and $\boldsymbol{\Phi}_c$ is the basis for the combined model parameter space. Writing $\boldsymbol{\Phi}_c = \left[\boldsymbol{\Phi}_{c,s}, \boldsymbol{\Phi}_{c,g}\right]^T$, it is now possible to generate new shapes and texture instances by

$$\boldsymbol{s} = \overline{\boldsymbol{s}} + \boldsymbol{\Phi}_s \boldsymbol{\Psi}_s^{-1} \boldsymbol{\Phi}_{c,s} \boldsymbol{c} \quad (4)$$

$$\boldsymbol{g} = \overline{\boldsymbol{g}} + \boldsymbol{\Phi}_g \boldsymbol{\Phi}_{c,g} \boldsymbol{c}. \quad (5)$$

In the search phase of AAMs, the model parameter $\boldsymbol{c}$ and the pose $\boldsymbol{p}$ are sought that best represent an object in a new image not contained in the original training set. In practise changing $\boldsymbol{c}$ varies both the texture and the shape of an object. Suppose we need to find an object in an image. The idea is to vary $\boldsymbol{c}$ (optimise over $\boldsymbol{c}$) so that the shape and texture generated by Equation 4 and Equation 5 fits the object in the image as well as possible. The objective function that is minimised is the difference

$$E = ||\boldsymbol{g}_{model} - \boldsymbol{g}_{image}||^2 = ||\delta \boldsymbol{g}||^2 \quad (6)$$

between the texture values generated by $\boldsymbol{c}$ and Equation 5, denoted as $\boldsymbol{g}_{model}$, and the texture values in the image, $\boldsymbol{g}_{image}$. Note that the image texture values $\boldsymbol{g}_{image}$ for a specific value of $\boldsymbol{c}$ are the values sampled from the shape generated by Equation 4 and then translated, scaled and rotated using the pose $\boldsymbol{p}$. This Euclidean transformation is necessary since the shape generated by Equation 4 is in the normalised coordinate system and we need to sample the texture values in the image coordinate system. In summary, the optimisation over $\boldsymbol{c}$ minimises Equation 6, i.e. produces the best fit of texture values.

For the implementation AAMs assumes that there exists a linear relationship between the differences in texture values $\delta \boldsymbol{g} = \boldsymbol{g}_{model} - \boldsymbol{g}_{image}$ and the model parameters' update, hence

$$\delta \boldsymbol{p} = \boldsymbol{R}_p \delta \boldsymbol{g} \quad (7)$$

$$\delta \boldsymbol{c} = \boldsymbol{R}_c \delta \boldsymbol{g}, \quad (8)$$

where $\boldsymbol{R}_p$ and $\boldsymbol{R}_c$ are found by conducting a set of experiments and using the results to perform multivariate linear regression. The parameters are fine-tuned by gradient-descend optimisation strategies.

The optimisation strategy described above, requires a good initialisation for the following reasons:

- The shape generated by $\boldsymbol{c}$ and Equation 4 is translated, scaled and rotated using $\boldsymbol{p}$—a large space to search.
- The assumption that there exists a linear relationship between the differences in texture values and the model parameters' updates is only reasonable for small updates.

From the discussion, we see that AAMs provide a general framework to track or segment different types of objects. Furthermore, no parameters need to be specified by an expert to use them. On the downside, AAM requires objects to have distinct features/outlines and there is a training phase involved. Also, a good initialisation is required for the search algorithm.

We used the open-source AAM-API [10] in our implementation. For further detail on AAMs, the reader may refer to [1, 9].

3. THE PARTICLE FILTER

Particle filters (also known as bootstrap filtering and the condensation algorithm) is a Monte Carlo-type technique to approximate probabilistic density functions (pdfs).

In the spirit of [3, 8], the state vector $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$ describes the object to be tracked at time step t, while the measurements are given by $\boldsymbol{z}_t \in \mathbb{R}^{n_z}$. We denote all the measurements up until the tth time step by $\boldsymbol{Z}_t \triangleq \{\boldsymbol{z}_i, i = 1, \ldots, t\}$. In the Bayesian framework, the goal is to find an estimate of $\boldsymbol{x}_t$ based on all the observations $\boldsymbol{Z}_t$. Thus the conceptual Bayes solution recursively updates the posterior pdf

$$p(\boldsymbol{x}_t|\boldsymbol{Z}_t) = \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{Z}_{t-1})\, p(\boldsymbol{x}_t|\boldsymbol{Z}_{t-1})}{p(\boldsymbol{z}_t|\boldsymbol{Z}_{t-1})} \quad (9)$$

as the measurements become available. In the particle filter, the posterior pdf is approximated by a weighted random sample set $\{\boldsymbol{x}_t^i, \pi_t^i\}_{i=1}^N$, $\sum_i \pi_t^i = 1$ where $\{\boldsymbol{x}_t^i, i = 1 \ldots N\}$ is a random measure of support points associated with the weights $\{\pi_t^i, i = 1 \ldots N\}$.

The particle filter algorithm consists of three phases that are repeated at each time step and are illustrated in Figure 1. The *selection* phase chooses particles according to their relative probabilities, i.e. particles with larger weights will be chosen several times, and those with smaller weights will be selected fewer times or discarded. During the *prediction* phase the selected particles are put through the process model to generate a prior pdf $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. To account for process

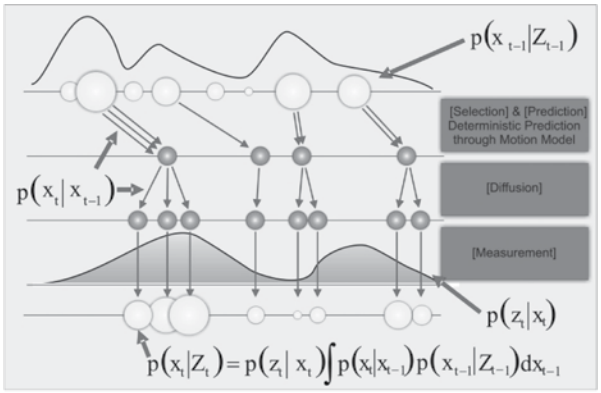


Figure 1: A graphical illustration of one iteration of the particle filter. (Image courtesy of S.Fleck [11].)

noise, the particles are *diffused* by adding noise to them. The *measurement* phase updates the weights in the light of the new measurement $\boldsymbol{z}_t$

$$\pi_t^i = \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t = \boldsymbol{x}_t^i)}{\sum_{n=1}^{N} p(\boldsymbol{z}_t|\boldsymbol{x}_t = x_t^n)}. \tag{10}$$

4. ACTIVE CONTOURS AND ACTIVE APPEARANCE MODELS

The combined active contour and active appearance model (AC-AAM) finds a shape to initialise the AAM algorithm using active contours and was proposed by Sung & Kim [12]. This method rectifies the situation in which AAM only works locally well. A summary of this technique is presented below.

4.1. *Contours and shape space*

B-splines, see [7, 13] for example, are used to model the outline of the tracked object. Given a set of coordinates of control points $(x_1, y_1), \ldots, (x_n, y_n)$, a B-spline is the curve $(\boldsymbol{r}(s) = (x(s), y(s))$ formed by a parametrisation with parameter s on the real line,

$$\boldsymbol{r}(s) = \begin{bmatrix} \boldsymbol{B}(s)^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}(s)^T \end{bmatrix} \begin{bmatrix} \boldsymbol{Q}^x \\ \boldsymbol{Q}^y \end{bmatrix} \tag{11}$$

where $\boldsymbol{B}(s)$ is the $n \times 1$ vector of B-spline basis functions, and $\boldsymbol{Q}^x$, $\boldsymbol{Q}^y$ are the vectors of control points determined by listing all the x-coordinates and y-coordinates respectively. We refer to a curve as defined by Equation 11 as a *contour*.

The dimension of the vector space spanned by $\boldsymbol{r}(s)$ is $N_Q = 2N_B$, where N_B is the number of control points of the B-spline. This implies N_Q degrees of freedom and it means that the object can deform in N_Q different ways if we track it over different frames. This amount of allowable deformation leads to many tracking errors. To rectify this situation, the allowable deformation is restricted to a lower dimensional space, known as the shape space.

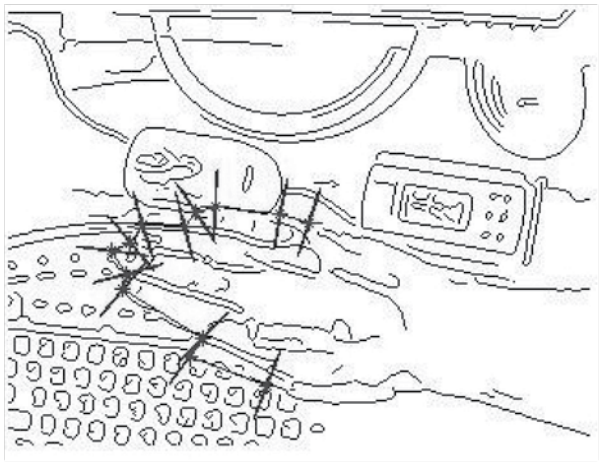


Figure 2: An example output from the Canny edge detector with a contour and its normal lines.

The shape space is defined as a linear mapping from a shape vector $\boldsymbol{X} \in R^{N_x}$ to a spline vector $\boldsymbol{Q} = \left[\boldsymbol{Q}^x,\ \boldsymbol{Q}^y\right]^T \in \mathbb{R}^{N_Q}$ and the mapping is given by

$$\boldsymbol{Q} = W\boldsymbol{X} + \boldsymbol{Q}_0 \tag{12}$$

where W is a matrix with rank $N_X \ll N_Q$, describing the allowable transformations. Variations are compared against the template curve $\boldsymbol{Q}_0$. By restricting $\boldsymbol{X}$, we can clearly restrict the transformations away from $\boldsymbol{Q}_0$.

4.2. *Active contours*

Following Blake & Isard [3], we summarise the contour based particle filter. Adapting a particle filter for a particular implementation, requires the specification of the state vector, process model and measurement model.

The state vector: The state vector at each time step t is given by the shape space vector. Thus $\boldsymbol{x}_t = \boldsymbol{X}_t$. This allow us to generate a vector of B-spline control points $\boldsymbol{Q}$ for each particle using Equation 12.

The process model: States evolve according to a simple random walk given by

$$\boldsymbol{x}_t^i = \boldsymbol{x}_{t-1}^i + \boldsymbol{S}_t^i \boldsymbol{u}_t^i \tag{13}$$

where $\boldsymbol{S}_t^i$ is the process noise covariance and $\boldsymbol{u}_t^i$ is a vector of normal distributed random variables. One can use more sophisticated process models and the reader is referred to [7] for a detailed discussion.

The measurement model: The binary edge map for the current frame is given to the algorithm to estimate the weight associated with each particle. An example of such an edge map, with a contour, its control points and normal lines superimposed on the edge map, are illustrated in Figure 2. To calculate the weight for the ith particle, the following procedure is followed:

- Using Equation 12 and the state vector, calculate the vector of control points $\boldsymbol{Q}^i$.
- Calculate the normal lines for each control point.
- For each control point, search along the normal line until an edge is found. Let the distance from the control points to the edge be $d_j, j = 1, \ldots, n$ where n is the number of control points. If an edge is not found, set the distance equal to the length of the normal line. Calculate the total length $d^i = \sum_{j=1}^{n} d_j$.
- The weight for the ith particle is then given by

$$\pi_i = \exp\left(-\frac{(d^i)^2}{\sigma^2}\right). \tag{14}$$

The weights are normalised afterwards to sum to unity. From equation Equation 14, we see that a small value of d^i will result in a larger value of π_i and vice versa. Furthermore, the variance σ determines how much preference we give to particles with a lower distance, d^i, or not.

4.3. *Initialisation of AAM with AC*

The AC-AAM tracker consists of two parts. In particular, at time step t, the first part performs standard active contour tracking and let the estimate from the tracker be $\boldsymbol{x}_t^e$. In the second part, Equation 12 is used together with $\boldsymbol{x}_t^e$ to generate a shape estimate

$$\boldsymbol{Q}_t^e = W\boldsymbol{x}_t^e + \boldsymbol{Q}_0. \tag{15}$$

Notice that $\boldsymbol{Q}_t^e$ and the AAM shape representation, $\boldsymbol{s}$, (not normalised with respect to the pose) are equivalent. This shape, $\boldsymbol{Q}_t^e$, is then used to initialise standard AAM optimisation and the result is output as the best fitted AAM. Sung & Kim [12] use the result of the AAM to initialise the active contour tracker again, but we find the aforementioned scheme adequate.

This technique uses the particle filter indirectly. The particle filter is an integral part of the active contour tracker, but the AAM part of the AC-AAM tracker does not utilise the particle filter.

5. AN ACTIVE APPEARANCE MODEL BASED PARTICLE FILTER

This section elucidates the second approach in order to increase the robustness of the AAM tracker. It is based on a direct combination of an AAM with a particle filter and was introduced by Hamlaoui & Davoine [14] with some additions made by Fleck et al [15]. It differs from the previous technique in the sense that the AAM is not initialised by a secondary technique, but instead temporal filtering predicts the parameters of the AAM. This way, history of the object improves the overall robustness of the AAM.

As in the case of the active contour tracker, the adaption of the particle filter to work in conjunction with an AAM, requires the specification of the state vector, the process model and the measurement model.

5.1. *The state vector*

The state vector is a combination of the model parameters $\boldsymbol{c}$ and the pose $\boldsymbol{p}$ and at time step t it is given by

$$\boldsymbol{x}_t = \begin{bmatrix}\boldsymbol{p}\\ \boldsymbol{c}\end{bmatrix}. \tag{16}$$

From this it is clear, that one can synthesise a shape and texture for a particular image given the model parameters.

5.2. *The process model*

The states evolves according to

$$\boldsymbol{x}_t^i = \hat{\boldsymbol{x}}_{t-1} + \begin{bmatrix}\boldsymbol{R}_p\\ \boldsymbol{R}_c\end{bmatrix}\delta\boldsymbol{g}_{t-1} + \boldsymbol{S}_t^{(i)}\boldsymbol{u}^{(i)} \tag{17}$$

where $\hat{\boldsymbol{x}}_{t-1}$ is the estimate of the state vector at time step $t-1$, $\boldsymbol{S}_t^{(i)}$ is the process noise covariance and $\boldsymbol{u}^{(i)}$ is a vector of normally distributed random variables.

5.3. *The measurement model*

Since the purpose of the measurement model is to classify how good a particular particle fits the underlying image, the optimisation criterion (equation Equation 6) will be used. Hence, for each particle the aforementioned norm is calculated and the pre-normalised weight is then given by

$$\pi_t^i = \exp\left(-\frac{E_i^2}{\sigma^2}\right) \tag{18}$$

where σ plays the same role as in the case with the active contour particle filter.

6. EXPERIMENTAL RESULTS

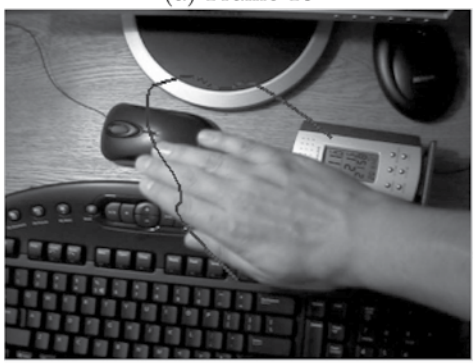
We implemented the AAM-based particle filter tracker in C++. The active contour implementation was done in MATLAB. To illustrate the effectiveness of the trackers, they were used to track a hand moving against a cluttered background and the results are available for download from the project's website [16]. The model of the hand consists of 13 feature points and the AAM was trained using 4 images.

Tracking results of performing deterministic AAM tracking is shown in Figure 3. It is clear that this approach cannot handle fast movements well as explained in Section 4.

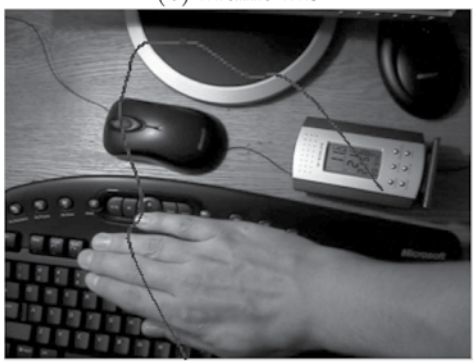
In Figure 4 the results obtained with the AC-AAM tracker are shown, while the corresponding results obtained with the AAM-based particle filter are illustrated in Figure 5. Both trackers are able to track the complete movement of the hand accurately.

When the AC-AAM approach is used, a simple dynamic model for the active contour tracker suffices. This can be seen in Figure 4 where the output of the

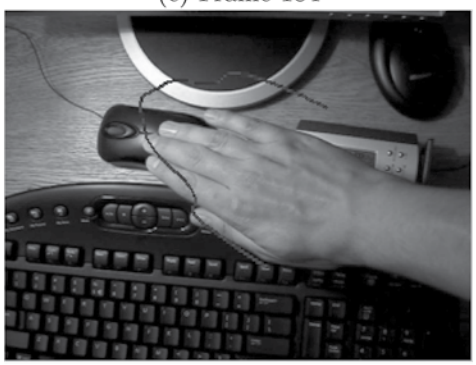
(a) Frame 10



(b) Frame 149



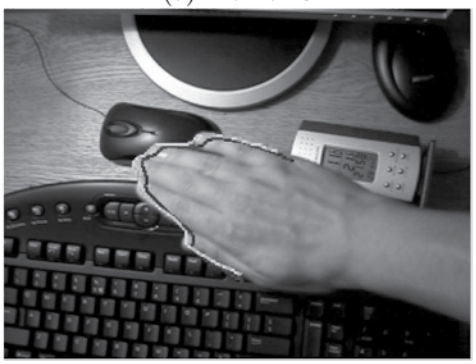
(c) Frame 184



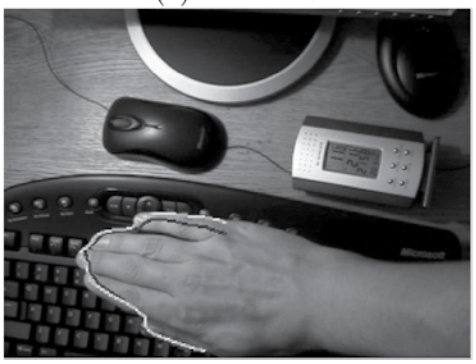
(d) Frame 310

Figure 3: A random selection of frames to indicate the performance of the deterministic AAM tracker. In (b), (c) and (d) the tracker loses its target due to fast movements.

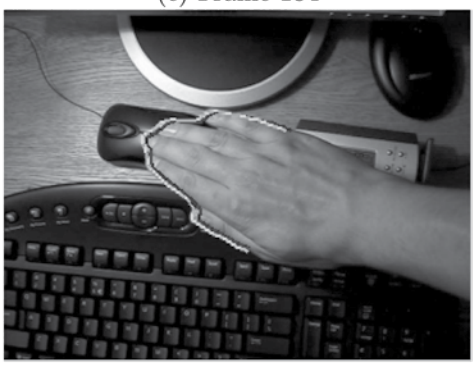
(a) Frame 10



(b) Frame 149



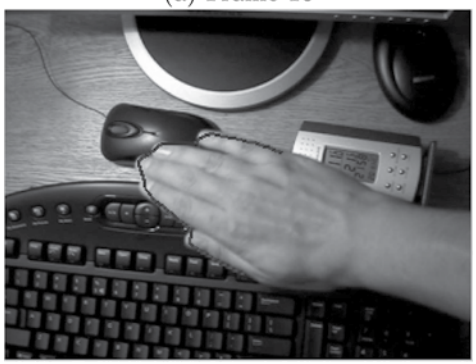
(c) Frame 184



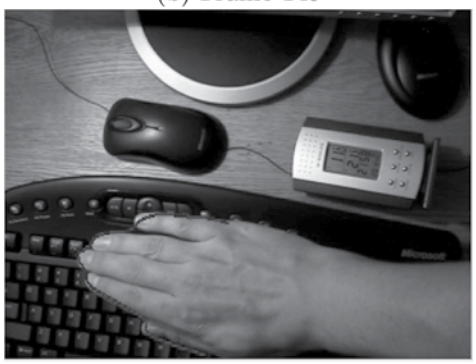
(d) Frame 310

Figure 4: A selection of frames to indicate the results of the AC-AAM tracker. The blue shape is the active contour's estimate and the green is the result after applying the AAM search algorithm, initialised with the blue shape.

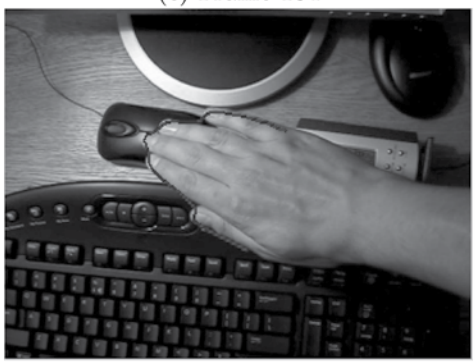
(a) Frame 10



(b) Frame 149



(c) Frame 184



(d) Frame 310

Figure 5: Selected frames to indicate the performance of the AAM-based particle filter tracker.

active contour tracker is not as accurate, but the resulting AAM fit is. This illustrates the principal idea of the AC-AAM approach: the robustness of the active contour tracker is used to initialise the AAM which in turn, adjusts well to the underlying image.

7. CONCLUSION

We discussed two techniques to enhance the deterministic AAM tracker. Central to both techniques is the particle filter. The effectiveness of the trackers was illustrated by presenting the tracking results of an object moving against a cluttered background.

Using the AAM-based particle filter has the advantage that extreme deformations can be learned in a single step. If this is require for the AC-AAM tracker (we can only handle transformations up to an affinity), the active contour as well as the AAM must be trained.

We expect that the runtime of the AC-AAM will be better compared to the AAM-based particle filter and we are currently working on creating a common framework to verify this claim. Future work includes the investigation of techniques to deal with occlusions in the AC-AAM tracker.

Acknowledgements: The financial assistance of the South African National Research Foundation (all authors) and Schlumberger (M. Hoffmann) towards this research is gratefully acknowledged. Opinions expressed and conclusions arrived at, are those of the authors and are not necessarily to be attributed to the NRF and Schlumberger.

8. REFERENCES

[1] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," in *European Conference on Computer Vision*, vol. 2, pp. 484–498, 1998.

[2] M. B. Stegmann, "Object tracking using Active Appearance Models," in *Proc. 10th Danish Conference on Pattern Recognition and Image Analysis* (S. I. Olsen, ed.), vol. 1, (Copenhagen, Denmark), pp. 54–60, DIKU, Jul 2001.

[3] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[4] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, (London, UK), pp. 661–675, Springer-Verlag, 2002.

[5] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "A color-based particle filter," in *First International Workshop on Generative-Model-Based Vision GMBV'02, in conjuction with ECCV'02*, pp. 53–60, 2002.

[6] M. Spengler and B. Schiele, "Towards robust multi-cue integration for visual tracking," in *ICVS '01: Proceedings of the Second International Workshop on Computer Vision Systems*, (London, UK), pp. 93–106, Springer-Verlag, 2001.

[7] A. Blake and M. Isard, *Active Contours*. Springer-Verlag, London, 1998.

[8] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter—Particle Filters for Tracking Applications*. Artech House, 1st ed., 2004.

[9] M. B. Stegmann, "Active Appearance Models: Theory, extensions and cases," Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, Aug 2000.

[10] M. B. Stegmann, B. K. Ersbøll, and R. Larsen, "FAME - a flexible appearance modelling environment," *IEEE Transactions on Medical Imaging*, vol. 22, no. 10, pp. 1319–1331, 2003.

[11] S. Fleck, S. Lanwer, and W. Straßer, "A smart camera approach to real-time tracking," in *13th European Signal Processing Conference (EUSIPCO 2005)*, September 4-8 2005.

[12] J. Sung and D. Kim, "A background robust Active Appearance Model using active contour technique," *The journal of the Pattern Recognition Society*, 2006. In press.

[13] B. Herbst and B. Fornberg, "Modelling in Applied Mathematics." [Online]. Available: `http://dip.sun.ac.za/courses/TW424/`, 2003.

[14] S. Hamlaoui and F. Davoine, "Facial action tracking using particle filters and Active Appearance Models," in *The Smart Objects and Ambient Intelligence conference*, October 2005.

[15] S. Fleck, M. Hoffmann, K. Hunter, and A. Schilling, "PFAAM - An Active Appearance Model based Particle Filter for both Robust and Precise Tracking," *The Fourth Canadian Conference on Computer and Robot Vision (CRV 2007) Montreal, Canada,*, May 2007.

[16] M. Hoffmann, "Project's website." `http://dip.sun.ac.za/~mcelory/vision-research/aam-tracking/`, 2006.