Evaluating Error Functions for Robust Active Appearance Models

Barry-John Theobald
School of Computing Sciences, University of East Anglia, Norwich, UK, NR4 7TJ
bjt@cmp.uea.ac.uk

Iain Matthews and Simon Baker
The Robotics Institute, Carnegie Mellon University, Pittsburgh, USA, PA 15213
{*iainm,simonb*}*@cs.cmu.edu*

Abstract

Active appearance models (AAMs) are generative parametric models commonly used to track faces in video sequences. A limitation of AAMs is they are not robust to occlusion. A recent extension reformulated the search as an iteratively re-weighted least-squares problem. In this paper we focus on the choice of error function for use in a robust AAM search. We evaluate eight error functions using two performance metrics: accuracy of occlusion detection and fitting robustness. We show for any reasonable error function the performance in terms of occlusion detection is the same. However, this does not mean that fitting performance will be the same. We describe experiments for measuring fitting robustness for images containing real occlusion. The best approach assumes the residuals at each pixel are Gaussianaly distributed, then estimates the parameters of the distribution from images that do not contain occlusion. In each iteration of the search, the error image is used to sample these distributions to obtain the pixel weights.

1. Introduction

Active Appearance Models (AAMs) are generative parametric models commonly used to track faces in video [1, 2]. A major limitation of AAMs is they are not robust to occlusion and only a small amount of occlusion can cause the AAM search to diverge. A robust extension to AAMs that is an efficient formulation of earlier fitting algorithms [3, 4] was described in [5]. In this paper we consider the choice of error function for use in this robust AAM search. This is not a problem that be answered using synthetically occluded data, as was done in [5]. Choosing an error function is effectively the same as asking *what is the real distribution of outliers in images*? Two ways this could be answered are by measuring the accuracy of occlusion detection, or measuring the robustness of the search. In this paper we test eight error functions using both of these metrics. We show that for any reasonable error function (monotonic and symmetric), the occlusion detection performance is the same. However, this does not mean that fitting performance will be the same as the *type* of error is important. A search that includes a small number of borderline outlier pixels (Type I error) may converge as these pixels are down-weighted to reduced their influence. Likewise, a search that ignores a number of inlier pixels (Type II error) may also converge. In this case not all of the available information is used in the search. All evaluation in this paper is conducted on a video sequence of a deaf-signer and we show the best results are obtained when the distribution of the residual at each pixel is assumed to be Gaussian. Clean, unoccluded images are used to estimate the parameters of these distributions, which are sampled in each iteration of the search using the error image.

2. Active Appearance Models: AAMs

The *shape*, $\mathbf{s}$, of an AAM is defined by the 2D coordinates of the N vertices that form a triangulated mesh:

$$\mathbf{s} = (x_1, y_1, x_2, y_2, \ldots, x_N, y_N)^{\mathrm{T}}. \tag{1}$$

AAMs allow linear shape variation, meaning a shape can be expressed as a base shape, $\mathbf{s}_0$, plus a linear combination of n template shapes, $\mathbf{s}_i$:

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{n} p_i \mathbf{s}_i, \tag{2}$$

where the coefficients p_i are the shape parameters.

AAMs are normally computed by hand-aligning the vertices of the mesh with the corresponding features

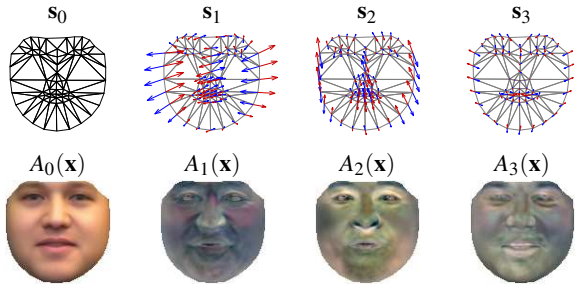

Figure 1. The linear shape model (top row) and appearance model (bottom row) of an AAM. Shown are the base shape and appearance (left column) and first three modes of variation.

in a set of training images and applying PCA [1]. The base shape is the mean shape and the template shapes are the eigenvectors corresponding to the n largest eigenvalues. An example is illustrated in the top row of Figure 1.

The *appearance* of the AAM is defined within $\mathbf{s}_0$. Let $\mathbf{s}_0$ also denote the set of pixels $\mathbf{x} = (x,y)^{\mathrm{T}}$ that lie inside $\mathbf{s}_0$, a convenient abuse of terminology. The appearance of the AAM is then an image, $A(\mathbf{x})$, defined over the pixels $\mathbf{x} \in \mathbf{s}_0$. AAMs allow linear appearance variation, meaning $A(\mathbf{x})$ can be expressed as a base appearance, $A_0(\mathbf{x})$, plus a linear combination of m appearance images $A_i(\mathbf{x})$:

$$A(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) \qquad \forall\, \mathbf{x} \in \mathbf{s}_0, \tag{3}$$

where the coefficients λ_i are the appearance parameters. As with the shape, the base appearance, $A_0(\mathbf{x})$, and appearance images, $A_i(\mathbf{x})$, are usually computed by applying PCA to the (shape normalised) training images [1]. An example is illustrated in the bottom row of Figure 1.

2.1. Robust Fitting of AAMs

The goal of the robust AAM search [5] is to minimise:

$$\sum_{\mathbf{x} \in \mathbf{s}_0} \rho\left(\left[A(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \right]^2 ; \sigma \right), \tag{4}$$

with respect to the shape and appearance parameters. $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$ is the image warped onto the base mesh, $\rho(\bullet)$ is a *robust error function* [6] and σ is a vector of *scale parameters*. Updates for λ are required that minimise:

$$\sum_{\mathbf{x}} \rho'\left(E(\mathbf{x})^2 \right) \left[E(\mathbf{x}) + \sum_{i=1}^{m} \Delta\lambda_i A_i(\mathbf{x}) \right]^2, \tag{5}$$

where $E(\mathbf{x})$ has been normalised so the component of the error image in the direction of $A_i(\mathbf{x})$ is zero [5]. The least squares minimum of this expression is:

$$\Delta\lambda \;=\; -H_{\mathbf{A}}^{-1} \sum_{\mathbf{x}} \rho'\left(E(\mathbf{x})^2 \right) \mathbf{A}^{\mathrm{T}}(\mathbf{x}) E(\mathbf{x}), \tag{6}$$

where $\mathbf{A}(\mathbf{x}) = (A_1(\mathbf{x}), \ldots, A_m(\mathbf{x}))$ and $H_{\mathbf{A}}$ is the appearance Hessian:

$$H_{\mathbf{A}} = \sum_{\mathbf{x}} \rho'\left(E(\mathbf{x})^2 \right) \mathbf{A}(\mathbf{x})^{\mathbf{T}} \mathbf{A}(\mathbf{x}). \tag{7}$$

The steepest descent parameter updates are computed using:

$$\Delta\mathbf{p} \;=\; -H_{\rho}^{-1} \sum_{\mathbf{x} \in \mathbf{s}_0} \rho'(E(\mathbf{x})^2) \left[\nabla A_0(\mathbf{x}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right] E(\mathbf{x}), \tag{8}$$

where $\nabla A_0(\mathbf{x})$ is the gradient of the base appearance and $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is the Jacobian of the warp [2]. The Hessian, H_ρ is computed using:

$$H_\rho = \sum_{i=1}^{K} \rho_i'\left(E(\mathbf{x})^2 \right) \sum_{\mathbf{x} \in T_i} \left[\nabla A_0(\mathbf{x}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} \left[\nabla A_0(\mathbf{x}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right], \tag{9}$$

where the base appearance is subdivided into K triangles, $T_1, T_2, \ldots, T_K$, allowing the search to deal with occlusion. Assume that $\rho'(E(\mathbf{x})^2)$ is constant in each triangle; i.e. assume $\rho'(E(\mathbf{x})^2) = w_i$, say, for all $\mathbf{x} \in T_i$. Pixels with a large error in $E(\mathbf{x})$ have a small weight, w_i, so have less significance in updating the parameters. In practise the assumption that w_i is constant for all $\mathbf{x} \in T_i$ holds only approximately, so w_i must be estimated from $\rho'(E(\mathbf{x})^2)$, for example by setting it to be the mean value computed over the triangle [7]. The efficiency of this search arises since the internal part of Equation 9 does not depend on the error so is constant across iterations. Denote:

$$H_\rho^i \;=\; \sum_{\mathbf{x} \in T_i} \left[\nabla A_0(\mathbf{x}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} \left[\nabla A_0(\mathbf{x}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right], \tag{10}$$

The Hessian H_ρ^i is the Hessian for triangle T_i and can be precomputed. Equation 9 then simplifies to:

$$H_\rho \;=\; \sum_{i=1}^{K} w_i \cdot H_\rho^i. \tag{11}$$

Although this Hessian does vary from iteration to iteration, the cost of computing it is minimal and the same spatial coherence approximation can be made for the appearance Hessian of Equation 7.

The following sections consider the selection of ρ, and evaluate eight possibilities using the accuracy of occlusion detection and fitting robustness as performance metrics. The evaluation is conducted on video sequences of a deaf-signer, thus we consider only *real* occlusions.

3. Error Functions for Robust AAMs

The purpose of the robust error function in Equation 4 is to down-weight pixel outliers. Desirable properties on the form of the error function include a function that is non-negative, symmetric, monotonic and piecewise differentiable. The final property is desired since it is the derivative, ψ, of the objective function that determines the influence of each pixel. The symmetry property is desired so a Gauss-Newton optimisation can be applied, rather than the less efficient Newton optimisation [8]. In this paper we consider the following eight *weighting* functions:

E1: — Huber function [6] ($c = 1.345$):

$$\psi(E(\mathbf{x});\sigma_{\mathbf{x}}) = \begin{cases} 1 & |E(\mathbf{x})| \leq c \\ \frac{c}{|E(\mathbf{x})|} & |E(\mathbf{x})| > c \end{cases}$$

E2: — Talwar function [9] ($c = 2.795$):

$$\psi(E(\mathbf{x});\sigma_{\mathbf{x}}) = \begin{cases} 1 & |E(\mathbf{x})| \leq c \\ 0 & |E(\mathbf{x})| > c \end{cases}$$

E3: — Tukey bisquare function ($c = 4.685$):

$$\psi(E(\mathbf{x});\sigma_{\mathbf{x}}) = \begin{cases} \left(1 - \left(\frac{E(\mathbf{x})}{c}\right)^2\right)^2 & |E(\mathbf{x})| \leq c \\ 0 & |E(\mathbf{x})| > c \end{cases}$$

E4: — Cauchy function ($c = 2.385$):

$$\psi(E(\mathbf{x});\sigma_{\mathbf{x}}) = \frac{1}{1 + \left(\frac{E(\mathbf{x})}{c}\right)^2}$$

E5: — Standardised distance:

$$\psi(E(\mathbf{x})) = \begin{cases} 1 & |\frac{E(\mathbf{x})}{\sigma_{\mathbf{x}}}| \leq 2\sigma_{\mathbf{x}} \\ 0 & otherwise \end{cases}$$

E6: — Pixel-wise threshold:

$$\psi(E(\mathbf{x})) = \begin{cases} 1 & |E(\mathbf{x})| \leq E_{max}(\mathbf{x}) \\ 0 & otherwise \end{cases}$$

E7: — Probability density function assuming the distribution of the residual at each pixel is Gaussian:

$$\psi(E(\mathbf{x})) = \frac{1}{\sigma_{\mathbf{x}}\sqrt{2\pi}} e^{\left(-\frac{|E(\mathbf{x})|}{2\sigma_{\mathbf{x}}^2}\right)}$$

E8: — Decaying exponential:

$$\psi(E(\mathbf{x})) = e^{\left(-\frac{|E(\mathbf{x})|}{2\sigma_{\mathbf{x}}^2}\right)}$$

E1–E4 are the *W-estimators* for the corresponding *M-estimators* [6]. The tuning constant, c, adjusts the scale, which is usually estimated from the residuals using the median of absolute deviations (MAD) [6]. In this work, we use the standard deviation of the residuals in unoccluded images as the measure of scale. We denote this as $\sigma_{\mathbf{x}}$ to reflect that each pixel is treated independently. The distribution of the residuals is modelled per-pixel, not over $E(\mathbf{x})$. Hence, the decision as to whether a pixel is occluded is not influenced by any other pixel.

4. Evaluation

Two metrics have been used to evaluate robust error functions: occlusion detection accuracy and robustness of fit. The *fitting algorithms* in [5] were tested by first labelling (occlusion-free) images using a non-robust search, then comparing the result of the robust search after adding *artificial* occlusion. This is fine since the relative performance of the fitting algorithms is not expected to depend on the data. In this work the relative performance of the *error functions* are being tested, which will depend on the data. Hence our evaluation must be performed on real data.

A short video sequence of a deaf-signer is divided into 112 frames containing occlusion and 136 frames without occlusion. Examples from the occluded set are shown in Figure 2.

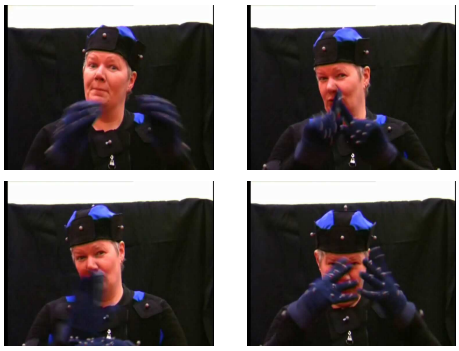

Figure 2. Example images used in the evaluation of robust AAMs. Note, the body suit forms the basis of an optical tracking system (not used in this work).

Two forms of ground-truth are required: which pixels are occluded and the location of the landmarks in each frame. Occluded pixels are identified by creating a binary mask and hand-painting over occluded regions in each image. The landmarks are slightly more tricky. It is undesirable to compare the output of the fitter with hand-labels as these are likely to be noisy. A non-robust

fit cannot be used to determine the ground-truth as the search will likely fail, see Figure 5. Also, a robust search using any *single* error function cannot be used as the results will be biased towards this function. Instead we first hand-label all 112 images in the occluded set, taking care to ensure occluded landmarks are in a reasonable position. Next, a robust AAM search using all eight error functions is performed using the hand-labels as an initial guess. Examples that diverge are ignored and the ground-truth is the mean of the converged (visible) landmarks. Example ground-truth is shown in Figure 3.

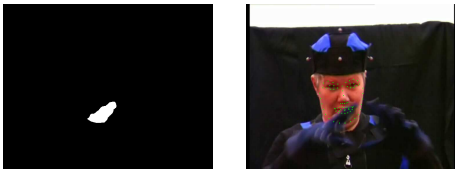

Figure 3. Example ground-truth data: The left image shows the binary mask for the image displayed on the right. White pixels in the mask denote the occluded pixels. The ground-truth landmarks are overlaid on the image on the right.

4.1. Occlusion Detection

There are two types of error when classifying pixels as inliers or outliers.

Type I Error — a pixel outlier is classified as an inlier.

Type II Error — a pixel inlier is classified as an outlier.

The following describes evaluating error functions in terms of occlusion detection accuracy.

4.1.1. Procedure. Each of the 112 (occluded) images are warped from the ground-truth landmarks onto the base shape and the residuals computed. These residuals are then input to each error function and the result compared with the hand-painted ground-truth. Since some functions make only a *soft* decision as to which pixels are occluded (i.e. **E1**, **E4**, **E7** and **E8**), a threshold, τ, is required that defines a decision boundary. This threshold is **not** used in the robust search, it is used only to make a decision in this detection experiment. Since the decision as to whether a pixel is visible or occluded is sensitive to τ, we consider the affect of varying the threshold.

4.1.2. Results. Figure 4 shows the average number of pixels correctly identified as occluded against the average Type II error for each error function.

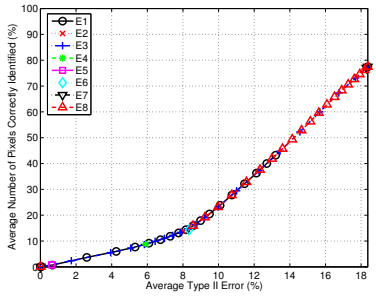

Figure 4. Curves showing the accuracy of occlusion detection against misclassified outlier rate for the error functions defined in Section 3.

The results in Figure 4 are as should be expected. The curves[1] for any symmetric, monotonically increasing error function will be the same in the following sense. Consider the two sets of pixels parameterised by the threshold τ:

$$\begin{aligned} FP(\tau) &= \{x \in \rho(|E(\mathbf{x})|) < \tau \,|\, x \notin \mathbf{y}\} \\ TP(\tau) &= \{x \in \rho(|E(\mathbf{x})|) < \tau \,|\, x \in \mathbf{y}\} \end{aligned}$$

where $\mathbf{y} \subset I(W(\mathbf{x};\mathbf{p}))$ are the occluded pixels, $TP(\tau)$ are the true positives, $FP(\tau)$ are the false positives and τ is the decision threshold. As τ varies the proportion of $|FP|$ and $|TP|$ to the total number of pixels map out the ROC curve. Thus, for any symmetric and monotonically increasing error function for which

$$\rho(E_i) > \rho(E_j) \quad \forall\, (|E_i| < |E_j|)$$

it follows that:

$$\begin{aligned} |FP(\tau)| &\leq |FP(t)| \,\forall\, (t \geq \tau) \\ |TP(\tau)| &\leq |TP(t)| \,\forall\, (t \geq \tau) \end{aligned}$$

and

$$\begin{aligned} FP(\tau) &\subset FP(t) \,\forall\, (t \geq \tau) \\ TP(\tau) &\subset TP(t) \,\forall\, (t \geq \tau) \end{aligned}$$

since ρ cannot change the ordering of the residuals. $FP(\tau)$ and $TP(\tau)$ are improper subsets of the respective supersets.

[1] For some error functions such as (**E2**, **E3**, **E5** and **E6**), the curves are degenerate and consist just of a single point.

4.2. Fitting Robustness

It is clear from Figure 4 that in terms of occlusion detection accuracy, monotonic and symmetric error functions perform the same. However this does not mean that they perform the same in terms of fitting robustness. The following describes the evaluation of the error functions from Section 3 in terms of the fitting robustness.

4.2.1. Procedure. Twenty of the images from the unoccluded frames were hand-labelled using the landmark configuration shown in Figure 3. An AAM was constructed from these labelled images and the non-robust AAM search [2] used to annotate the remaining unoccluded images. Each image was then warped onto the base shape and the error image computed. The standard deviation of the residual and maximum absolute value of the residual at each pixel was computed, providing the parameters for the error functions.

For each of the 112 occluded images, 500 starting locations for a robust search were generated by randomly perturbing the ground-truth shape and similarity transform parameters with additive white Gaussian noise. The variance of the distribution used to perturb each shape parameter was equal to a multiple of the variance captured by the corresponding mode of variation. Specifically, fifty offsets were generated for each of ten evenly spaced levels of shape perturbation ranging from 0.3 to 3 times the variance of the corresponding mode. The similarity transform parameters were generated by perturbing two points in the mesh with Gaussian noise of variance five times the shape offset and the similarity transform parameters then solved for [2]. At each iteration of the search, the image was warped onto the template and the robust error functions used to estimate and down-weight occluded pixels from resulting residuals. The triangle weights, w_i in Equation 11, were computed as the mean of the pixel weights within each triangle. This however is not the only option. For example, pixel-wise weights could be applied (an inefficient search), or the minimum pixel weight within each triangle could be used. Experiments evaluating different triangle weighting schemes are ongoing.

In all cases, the robust fitter was run for twenty iterations and the search was deemed to have converged if the RMS error between the ground-truth and fitted landmarks was below 2.0 pixels. Both the *frequency* of convergence (robustness) and the *rate* of convergence (accuracy) are used to quantify the performance of error functions.

4.2.2. Results. The set of 112 images containing occlusion were divided into two further sets: those that contain $0 < n \leq 25\%$ occlusion (80 frames) and those that contain $25 < n \leq 50\%$ occlusion (23 frames)[2]. The frequency and rate of convergence averaged over all trials and all images for each image subset are shown in Figure 5.

The performance of the error functions is similar for low degrees of occlusion ($\leq 10\%$). However, as the level of occlusion increases weighting pixels using error function **E7** appears to be the most robust technique. The average frequency of convergence is approximately 15% higher for $\leq 25\%$ occlusion than the next best error functions (**E6** and **E8**). As is expected, the frequency of convergence decreases as the amount of occlusion and shape perturbation increases. The unweighted L2 norm (non-robust) AAM search is surprisingly robust for low amounts of occlusion and performed only slightly worse than error function **E5**.

In terms of the rate of convergence, the error functions behave the same for low/moderate amounts ($\leq 25\%$) of occlusion — they are within one pixel at each iteration. Indeed it appears that, with the exception of the Tukey Bisquare function, the degree of occlusion does not influence how quickly the robust AAM will converge, only whether or not it will converge.

Figures 4 and Figure 5 suggest that robust AAMs are able to cope with a relatively large Type II error. The location on the curve for **E7** shows that, with the exception of $\tau = 0$, this error function classifies many of the unoccluded pixels as occluded. Thus, as we would expect, it is better to ignore unoccluded pixels than to include occluded pixels during the fit. In terms of the M-estimator functions (**E1**—**E4**), the best performing are the Talwar function and Cauchy function. The Talwar function was also used in [10] for robustly fitting morphable models to images.

5. Conclusions

In this paper we have reviewed the efficient robust AAM search algorithm and described a number of robust error functions that can be used in this search. We evaluated these error functions using two evaluation metrics: one to determine the accuracy of occlusion detection and another to determine the robustness of the search. We have shown that in terms of occlusion detection accuracy, all monotonic and symmetric error functions perform the same, whereas in terms of fitting robustness some perform significantly better than others. We have found that of the eight functions tested here, the best approach is to model, using a Gaussian, the distribution of the residuals at each pixel for known,

[2]The nine frames with $> 50\%$ occlusion were ignored in this experiment as the fitter always failed to converge.

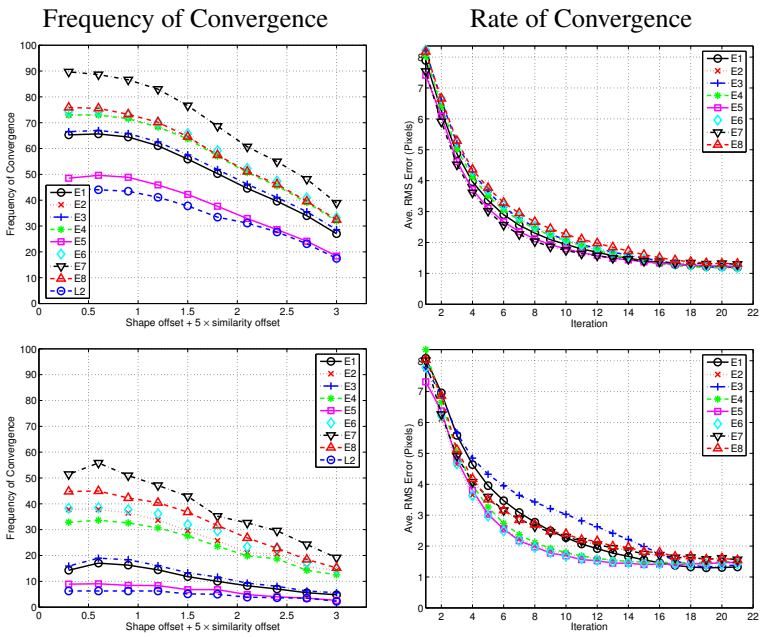

Figure 5. Frequency and rate of convergence of the robust AAM search using the error functions defined in Section 3 for Top row: $0 < N \leq 25\%$ **occlusion and Bottom row:** $25 < N \leq 50\%$ **occlusion.**

unoccluded data. The weights used during the search are then computed by sampling the respective distributions given the residuals at each pixel in each iteration. Functions **E1**–**E4** are well understood general purpose error functions used by the robust statistics community for performing an iteratively re-weighted least squares fit. It is perhaps to be expected that **E7** out-performs these as the parameters of this error function are estimated from known good data.

The error functions were tested on only a single video sequence. This was due to the difficulty in obtaining ground-truth. Every frame containing occluded pixels requires the manual placement of the landmarks and the manual marking of the occluded pixels. Further work will involve labelling more sequences and performing similar tests on more subjects. We will also compare different ways of computing the triangle weights from the pixel weights. In this work, the triangle weight is the mean of the pixel weights within the triangle. We will also contrast the robustness of this efficient algorithm with a less efficient algorithm which retains the individual pixels weights, but must recompute the Hessian in each iteration.

References

[1] Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001) 681–685

[2] Matthews, I., Baker, S.: Active appearance models revisited. International Journal of Computer Vision **60** (2004) 135–164

[3] Black, M., Jepson, A.: Eigen-Tracking: Robust matching and tracking of articulated objects using a view-based representation. International Journal of Computer Vision **36** (1998) 101–130

[4] Hager, G., Belhumeur, P.: Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 1025–1039

[5] Gross, R., Matthews, I., Baker, S.: Constructing and fitting active appearance models with occlusion. In: Proceedings of the IEEE Workshop on Face Processing in Video. (2004) 72–79

[6] Huber, P.: Robust Statistics. Wiley & Sons (1981)

[7] Baker, S., Gross, R., Ishikawa, T., Matthews, I.: Lucas-Kanade 20 years on: A unifying framework: Part 2. Technical Report CMU-RI-TR-03-01, Carnegie Mellon University Robotics Institute (2003)

[8] Baker, S., Matthews, I.: Lucas-Kanade 20 years on: A unifying framework. International Journal of Computer Vision **56** (2004) 221 – 255

[9] Hinich, M., Talwar, P.: A simple method for robust regression. Journal of the American Statistical Society **70** (1975) 113–119

[10] Romdhani, S., Vetter, T.: Efficient, robust and accurate fitting of a 3d morphable model. In: Proceedings of the IEEE International Conference on Computer Vision. (2003)