Practical Avatar Signing in British Sign Language

Report on UEA activity between 25th January 2005 and 24th April 2005

Overview

Previous work at UEA produced an effective facial tracker for images without occlusion by the hands or excessive rotation of the head. Progress at UEA focusses in two main areas that are able to proceed independently. Firstly, to develop techniques to make a tracker that will be robust in the presence of occlusion. Secondly, to integrate facial and body data in order to use the original 2D model results to drive a 3D model. The key is to detect and compensate for the pose of the face.

Introduction

25th January 2005 to 24th April 2005

Previous work at UEA produced an effective facial tracker for images without occlusion by the hands or excessive rotation of the head. Unfortunately, occlusion occurs frequently as signs involve manual elements in front of or in contact with the face. The basic tracker need modification to remain robust in the presence of occlusion. Early results are promising, as has been reported in earlier periods.

Ideally the tracker would infer the presence of occlusion and adapt. Work to achieve this objective is being done by adding artificial occlusion in a controlled fashion to sequences free from occlusion and comparing the tracking results against known results.

Should it prove unrealistic to detect occlusion automatically, it would be possible to look at other sources of information, for instance by tracking the hands or using coloured gloves to indicate the presence of occlusion.

Meanwhile, work is progressing to integrate face and body data in order to use the original 2D model results to drive a 3D model. It is essential to detect and compensate for the pose of the face so that the facial animation is correct when viewed head on. Good progress has been made, but the results are not yet sufficiently natural. This work uses images without occluded facial data but could clearly incorporate an improved tracker at a later date.

A final area is correct animation of body data. It is necessary to map HTR data to the avatar used for display, either by creating an avatar that exactly matches the geometry assumed by the HTR data or by using inverse kinematics to retarget data to a new avatar. Some work is needed to accommodate known problems with the available HTR data but that might best be seen as a stop-gap until it becomes easier to generate clean HTR data.

Manuals: Progress on mapping to correct avatar: Standard pose of avatar revealed by setting rotations to zero. Constrained movements of fingers to prevent movement out of plane. Gives animation that seems natural but may be incorrect.

Markerless Facial Feature Tracker (Barry Theobald)

Further work was undertaken on occlusion detection and fitting in the presence of occlusion. The previous work on M-estimators has been developed with statistics-based robust error functions that takes advantage of the structure of clean data when scaling the contribution of potentially occluded pixels to the error estimate.

The document “Statistics-based Robust Error Functions” containing the full report of work to date is attached.

Avatar Research Platform Toolkit (Vince Jennings)

Work on the skeleton building tool for the toolkit is now complete. Work to extend the module to provide linking from the skeleton to an avatar mesh is nearing completion, including vertex weighting

for mesh deformation. Alternative (open source) tools are also being evaluated for creation of the mesh itself. Once complete, this will remove the dependency on third party commercial software for avatar creation and provide more flexibility in skeleton construction.

With the recent release into the public domain of the FBX SDK by Alias (which has taken over Kaydara) it is now possible to write importers/exporters using the FBX file format. This is being evaluated as a possible format for the ARP Toolkit to simplify interchange of geometry, skeletons, and animations between applications. This format is already becoming an industry standard in commercial applications and this release will undoubtedly widen its popularity.

To facilitate simultaneous work on the ARP avatar and other related software by several developers, a Concurrent Versions System (CVS) has been installed and deployed. A commercial plugin has been acquired to allow access directly from relevant development environments. Some reorganisation work has been undertaken to enable the software to take advantage of development under CVS.

Motion Capture and Integration (Judy Tryggvason and Vince Jennings)

Worked has progressed to combine face data with HTR format body data for the same sequence. Since HTR assumes an avatar with geometry that does not necessarily match the ARP avatar, it will either be necessary to retarget the motion data or change the ARP geometry to match the HTR avatar. It had been assumed that Mocap would allow retargetting, but it is not clear if that approach will be possible.

Inconsistencies in the matching sets of facial and body capture data have now been resolved, although the number of complete sets remains severely limited. Further data will be needed to confirm that changes from session to session can be handled automatically.

Body animation.

With the release of a new version of Mocap by Alias, new settings have been included in the interface to allow scaling of the joint rotators that previously obscured the fingers completely. Animation of the fingers in Mocap can now be clearly observed.

The plotting of animation data from the skeleton in the HTR files to the ARP avatar skeleton depends on precise information about the default pose for the HTR data. Because there was no reference pose recorded in the HTR data sequence, a number of attempts were made at adjusting the avatar pose manually, with varying degrees of success. However, none were entirely satisfactory.

Later it became clear that setting data values to zero would yield a reasonable default pose and this has enabled acceptable matching of the default pose of the ARP skeleton to the default pose held in the HTR file. Some errors still seem to remain, however, such as hand/hand and hand/head relationships when compared to the video. Some of these are plotting errors between skeletons, but others, such as hand/head relationships, are probably due to errors in the collarbone data which were documented by the BBC when this file was supplied.

To improve the animation of the fingers the animation file exported from Mocap has been processed to apply appropriate constraints and limits to the finger joints, such as rotations around the bone's length of which there are several with ~180 degree rotations. This has improved the visibility of the remaining observed rotations of the fingers, and hand shapes can now be usefully compared with the video (see movies). The resulting avatar animation seems more realistic, but some artefacts may have been introduced in the process.

Videos have been produced that combine pose corrected face data and body data. Versions show raw body data and body data corrected using simple constraints.

Video: hands uncorrected.avi

Video: hands corrected.avi

Some other joint settings, such as the collarbones and the head, have also been modified during this processing to improve the avatar's stance.

Since a good deal of data cleaning is applied during production of the HTR files that we have used, it may be preferable to apply these and other corrections at an earlier stage. Discussions with colleagues at the BBC will take place to decide the appropriate point to apply the various corrections.

We are now engaging in a full survey of the available HTR data to determine better ways of extracting pose information from the facial data. Another approach will be to use the pose information available from the body data to drive the realignment of the facial information. This will depend on the camera being in a known position with respect to the frame of the body data.

Data File: linda_signing1_first_30s_2ndsolve.htr

Video: combined uncorrected.avi Body and face animation combined, no finger correction.

Video: combined corrected.avi Body and face animation combined, with finger correction.

Video: hands uncorrected.avi Closeup of hands showing finger errors, at 10fps, no face animation

Video: hands corrected.avi Closeup of hands with corrections, at 10fps, no face animation.

John Glauert
27 May 2005

Statistics-based Robust Error Functions

Background

The method of iteratively re-weighted least-squares using *M-estimators* is well understood. The implementation described in [BBC Report, January 2005] is different from the usual formulation. During the fit, the robust error function has only a single observation (the error image) from which it must estimate the outliers. Obtaining a single estimate of the scale by analysing the residuals of the elements across the entire error image is not ideal as the distribution of each element is likely to differ in both location and scale. Instead a scale estimate is pre-computed for each individual pixel from known "good" data and this scale estimate is used in the fit. The formulation described previously can easily be extended to account for robust errors functions that are not based on *M-estimators*. In particular, statistical-based error functions can be used, where the statistics of clean data are used to detect outliers.

Weighting Functions

The first, perhaps most naïve, weighting function simply records the largest residual at each pixel across the 300 training images, where the residual is taken as the (appearance) reconstruction error. The weighting function then returns

$$w(r) = \begin{cases} 1 & |r| <= 2R \\ 0 & |r| > 2R \end{cases} \tag{1}$$

where R is the largest error for the corresponding pixel in the template observed in training data. The second weighting function standardises the residuals using

$$D(\mathbf{x}) = \frac{r}{\sigma} \tag{2}$$

where σ is the standard deviation at each pixel estimated from the 300 training images. The weight function analyses the residuals, and any further than, say, ± 2 standard deviations is considered an outlier and assigned zero weight. The final weighting function models the distribution of each pixel (assuming a normal distribution) and assigns a weight $\propto P(I(\mathbf{W}(\mathbf{x};\mathbf{p})))$. The probability itself is not used for the weights as an image with zero error (i.e. exactly the template) will not return unit weights. Instead,

$$w(r) = e^{\left(-\frac{|r|}{2\sigma^2}\right)}, \tag{3}$$

is used, which is a Gaussian shaped function with a decay proportional to the variance of the data. (We have also run tests where the full probability is computed for each pixel).

Evaluation

To evaluate these error functions, an AAM was constructed from 30 hand-labelled images of an individual. The AAM contained approximately 10,000 pixels (the appearance images $A(\mathbf{x})$ contain approximately 30,000 elements as the model is constructed from colour images). The project-out AAM fitter [Matthews and Baker, 2004] was used to fit this model to 300 novel images and the resultant fit checked to ensure the fit was accurate. To test the robust error functions, 900 images (none of which were used to build the model or estimate the scale) were randomly selected from a sequence of a person talking (the same individual used in training the model). The test proceeded as follows:

1. Automatically label the face using the project-out AAM fitter[1].
2. Warp the image from fitted landmarks to the base shape.
3. Compute the appearance parameters, $\lambda_i = A_i(\mathbf{x})' \left[I(\mathbf{W}(\mathbf{x}, \mathbf{p})) - A_0(\mathbf{x}) \right]$.
4. Generate the template image $T(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x})$.
5. Randomly select N% of the pixels in $I(\mathbf{W}(\mathbf{x}, \mathbf{p}))$ and replace them with pixels randomly selected from an image of scenery. We place no constraints on the distance between the values of the pixels $I(\mathbf{W}(\mathbf{x}, \mathbf{p}))$ and those used to overwrite the them. If a pixel is overwritten with another of a similar colour then the pixel will unlikely be considered occluded — we overcome this by performing ten iterations at several percentage levels across 900 images and average the results. We consider a number of percentage levels, from 1% to 91% in steps of 10%.
6. Compute the error $E(\mathbf{x}) = [I(\mathbf{W}(\mathbf{x}, \mathbf{p})) - T(\mathbf{x})]$. We use $T(\mathbf{x})$ and not $A_0(\mathbf{x})$ to compute the error since the robust fitter must solve for the appearance parameters. If the model converges to the true minimum, then $T(\mathbf{x})$ during the fit will be approximately $I(\mathbf{x})$ (given the reconstruction error).
7. For each of the loss functions, detect the occluded pixels.

The results are given in Section 1.

Results

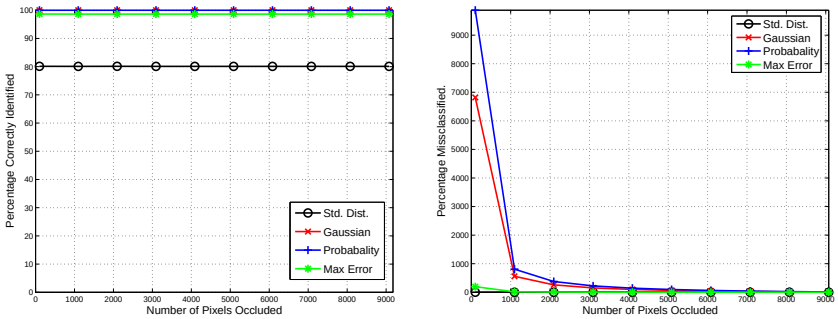
The results for the occlusion detection using the simple statistical measures are shown in Figure 1. Shown are the standardised distance, the probability of the pixel values (assuming a Gaussian distribution), the unnormalised probability (calculated by computing only the exponent), and assigning zero weight to pixels with a residual larger than the allowed maximum estimated from the training data. While the results look promising, the number of false positives for the probability-based detection scheme is enormous. In the context of robustly fitting AAMs to face images, large areas of the face would always be ignored when computing the parameter updates. These are regions that are poorly modelled using a simple Gaussian (the inner mouth for example). This could possibly be overcome, for example, by using a mixture of Gaussians to model the pixels in these areas.

References

[Matthews and Baker, 2004]] I. Matthews and S. Baker. Active Appearance Models Revisited International Journal of Computer Vision, Vol. 60, No. 2, November, 2004, pp. 135 - 164.

[BBC Report, January 2005] B.J. Theobald. Robust Error Functions. In Report to BBC, Period October 2004 to January 2005.

[1] All images were checked to ensure the fit was accurate.



(C)

Figure 1: Detection of occluded pixels randomly selected in the image. A) The percentage of correctly identified pixel plotted against the total number of pixels occluded for the *non-M-estimator* functions. B) The percentage of pixels miss-classified as occluded (false positives),for the functions in A).