Robust AAM Fitting by Fusion of Images and Disparity Data

Joerg Liebelt
Interactive Systems Labs
Universitaet Karlsruhe (TH)
76131 Karlsruhe, Germany
joergand@cs.cmu.edu

Jing Xiao
Epson Palo Alto Laboratory
3145 Porter Drive
Palo Alto, CA 94304
xiaoj@erd.epson.com

Jie Yang
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
yang+@cs.cmu.edu

Abstract

Active Appearance Models (AAMs) have been popularly used to represent the appearance and shape variations of human faces. Fitting an AAM to images recovers the face pose as well as its deformable shape and varying appearance. Successful fitting requires that the AAM is sufficiently generic such that it covers all possible facial appearances and shapes in the images. Such a generic AAM is often difficult to be obtained in practice, especially when the image quality is low or when occlusion occurs. To achieve robust AAM fitting under such circumstances, this paper proposes to incorporate the disparity data obtained from a stereo camera with the image fitting process. We develop an iterative multi-level algorithm that combines efficient AAM fitting to 2D images and robust 3D shape alignment to disparity data. Experiments on tracking faces in low-resolution images captured from meeting scenarios show that the proposed method achieves better performance than the original 2D AAM fitting algorithm. We also demonstrate an application of the proposed method to a facial expression recognition task.

1 Introduction

Many applications, such as facial expression analysis and lipreading, require information about not only the position of an entire face but also detailed non-rigid motions of the face. Model-based approaches, such as 2D Active Appearance Models (AAMs) [4] and 3D Morphable Models (3DMMs) [2], are commonly used for these tasks. AAMs provide a good solution since they are capable of representing shape and texture changes as well as non-rigid deformations of a human face. Over the last few years, many different algorithms and technologies have been proposed to improve performance and efficiency of AAMs, including different training methods [4], models, and efficient fitting strategies [1]. Some researchers have also explored the relationship between 2D AAMs and 3DMMs. For example, [12] introduced 2D+3D AAMs, a model with the real-time fitting speed of 2D AAMs and the 3D modeling of 3DMMs.

Most existing AAM training and fitting algorithms use 2D images for their parameter estimations. Estimation of a shape model (2D or 3D) from 2D images, however, suffers from many limitations. First, a human face is inherently a 3D object. Estimation of its shape and appearance from 2D instances requires a large amount of training data and thus a generic AAM is difficult to obtain. The AAMs are, therefore, usually user dependent, i.e., we need different AAMs for different users which is inconvenient in practice. Second, fitting AAMs from 2D images is usually performed on relatively high resolution images, which may not be available in some real world applications. Finally, 2D AAMs may not be able to accurately capture subtle movements of 3D facial features. Although 3DMMs can accurately represent a 3D face, they require error-prone manual labeling of dense 3D data for model creation and their model fitting speed is slow compared to AAMs. On the other hand, some previous work suggested that real-time range information can enhance robustness of object tracking. For example, [5] demonstrated that, in certain cases, depth information could be used to detect objects that were not discernible from intensity images alone and make model acquisition easier.

In this paper, we propose to use both 2D and 3D information to fit AAMs in order to overcome some of the limitations of the previous methods. Unlike other approaches which propose to perform face tracking directly on 3D data, such as the work of [8], we retain the 2D description of the face as well as the model-based alignment method. We describe how to combine efficient 2D AAM fitting and dense 3D mesh alignment using depth information, evaluate the proposed method by comparing it to 2D AAMs and demonstrate our approach in meeting scenarios where the system tracks participants from relatively low resolution video sequences. Furthermore, we have applied the method to an emotion classification task. We use an SVM to classify seven different emotions, yielding a recognition rate

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

of 71.3% on average. The rest of the paper is organized as follows: in section 2, we briefly review creation and efficient fitting of 2D AAMs. In section 3, we describe the 2D+3D extension for AAMs introduced by [12] and our 3D fitting algorithm. We explain how 3D data is preprocessed and we outline the integration of 3D information into the fitting process in order to improve precision of the 2D model alignment. In section 4, we present experimental results by comparing the fitting results of the proposed approach with a 2D alignment method and illustrate an application of the proposed approach to emotion classification using Support Vector Machines (SVMs) [11].

2 Background

An AAM is a statistical model of the shape and appearance of an object of interest learned from training data. The 2D AAM will serve as the basis of the proposed method and provides a coarse, yet fast initialization of the facial geometry. Matching an image to an AAM involves finding model parameters which minimize the difference between the image and a synthesized model example, projected into the image. A large number of parameters in an AAM makes the matching process a challenging problem. Much effort has been directed to developing efficient algorithms in fitting AAMs.

An AAM can be defined by a set $(\mathbf{p}, \mathbf{S}, \lambda, \mathbf{A})$, where $\mathbf{p}$ are the shape parameters, $\mathbf{S}$ a set of shape eigenvectors (called *bases* below), λ a set of texture parameters and $\mathbf{A}$ a set of texture eigenvectors (called *appearances* below). The shape parameters and bases define the warp $W(x, \mathbf{p})$ for every pixel x in an image. An AAM instance then has the form

$$I(W(x, \mathbf{p})) = A(x, \lambda), \tag{1}$$

where I is a 2D input image and A the appearance corresponding to the parameter set λ. Since we would like to track a shape that is not only internally deforming, but also undergoing pose changes, we need to account for possible similarity transformations of the shape

$$N(x, \mathbf{q}) = Rx + \mathbf{t}, \tag{2}$$

where x denotes a 2D pixel position within the shape, and R, $\mathbf{t} = (t_x, t_y)$ rotation and translation. When choosing a representation of the pose changes of

$$N(s_0, \mathbf{q}) = s_0 + \sum_{i=1}^{4} b_i^* q_i, \tag{3}$$

with a suitable set of pose bases $\mathbf{b}^*$ orthogonal to the shape bases $\mathbf{b}$, the parameters q_i instantiate the pose and can be used in the same way as the AAM shape parameters. Given an input image $I(x)$, we now wish to minimize the sum of squares difference between an instance created with an AAM and the input image:

$$\sum_{x \in s_0} [A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) - I(N(W(x, \mathbf{p}), \mathbf{q}))]^2. \tag{4}$$

The solution is obtained by simultaneously minimizing this difference with respect to the parameters $\mathbf{p}$, $\mathbf{q}$ and λ.

[1] proposed a solution to the above minimization task (4), known as *inverse compositional image alignment* (ICIA). In contrast to traditional image alignment methods, ICIA allows precomputation of the steepest descent geometry, thus increasing fitting speed significantly. Additionally, minimization is performed in a subspace of the original problem space that is independent of texture variation which is *projected out*, thus further simplifying the task.

3 Fusion of Images and Disparity Data for AAM Fitting

Given a 2D description of an AAM instance, and dense but noisy 3D disparity data obtained from the scene using a stereo camera, we would like to align the 2D AAM to optimally fit the 3D data. Unlike methods that rely exclusively on noisy 3D data and thus being limited in precision and detail resolution, we make use of the clean 2D AAM instance to improve the robustness of the alignment. While the 2D AAM only provides a rough initialization on relatively low resolution 2D images, the dense 3D disparity data then allows us to fit the 3D shape recovered by the AAM to the given scene such that the tracking will not be mis-led by the appearance changes that are not covered by the original model. In order to bridge 2D AAMs and 3D disparity data, we employ 2D+3D AAMs [12] in order to recover the 3D face shapes and poses corresponding to the 2D shapes in the input images.

3.1 Combined 2D+3D AAMs

We wish to adapt the mesh of a 2D AAM to 3D data. As a consequence, we need to derive a 3D description of the tracked object using the fitted 2D shape. While 2D AAMs can generate states that cannot be attained by a 3D face, it is possible to impose constraints such that the allowed 2D deformations match with the corresponding 3D shapes. As a result, the constrained fitting suggested by [12] recovers the 2D as well as the 3D shape $\bar{s}$ of the face.

Using the structure from motion method, a measurement matrix containing the locations of corresponding landmark points in a sequence tracked by a 2D AAM is factorized in order to reconstruct the corresponding 3D shapes and the underlying 3D shape model [13, 14]. The factorization results in a set of 3D bases $\bar{b}_i$ describing the 3D deformations

of the face shape whose image projection is represented by the 2D AAM. Given the optimally fitted 2D AAM in the input image and a set of equivalent 3D bases $\bar{b_i}$, we would like to derive the corresponding 3D shape. On the other hand, we would like to constrain the AAM fitting such that the recovered 2D shape is an image projection of a valid 3D shape, *i.e.* a linear combination of the 3D bases. The constrained 2D+3D extension described above was achieved by including an additional term into the AAM minimization [12]. This term describing the combined 2D+3D minimization has the following form

$$\sum_{x \in s_0} [A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) - I(N(W(x, \mathbf{p}), \mathbf{q}))]^2 + K \cdot \| P(\underbrace{\bar{s_0} + \sum_{i=1}^{\bar{m}} \bar{p_i} \bar{b_i}}_{=\bar{\mathbf{s}}}) + \mathbf{o} - \underbrace{N(s_0 + \sum_{i=1}^{m} p_i b_i; \mathbf{q})}_{=\mathbf{s}} \|^2, \quad (5)$$

where $\bar{\mathbf{s}}$ is an instance of the 3D shape, P is a weak projection matrix, $\mathbf{o}$ is an offset, and $\mathbf{s}$ describes the current 2D shape instance after pose transformation. K is a constant, for $K \rightarrow \infty$ the additional constraint on matching between 2D and 3D shapes becomes a *hard* constraint. Details on the 2D+3D extension are given in [12].

3.2 Obtaining and Structuring 3D Data

In order to perform alignment to 3D data, we have to choose a suitable data structure for a dense, noisy 3D point cloud computed from stereo disparity values, typically containing around 10000 elements within the face region of interest. Depending on the resolution of the model mesh and the quality of the depth data, a suitable subsampling of the input point cloud can be performed in order to speed up processing. For a query of N 3D points representing the shape vertices of the 3D-extended AAM mesh, we have to determine the nearest neighbor of each of the query points within the point cloud. We have chosen to populate one KD-Tree for each new input frame, since we do not need any other functionality, neither element manipulation nor re-balancing. In our implementation, we use the ANN library developed by [9].

Due to noises in disparity data, lighting issues, and lack of texture in many regions of a face, we have uncertainties in the depth information for each shape vertex. We need to account for those uncertainties when designing the closest-point lookup function $CP_{3D}(\bar{\mathbf{s}})$ which returns the nearest neighbor for every vertex of a given 3D shape $\bar{\mathbf{s}}$. When we assume that the initial 2D fitting is sufficiently precise such that it roughly converges towards the initial face position, we can define a threshold such that all shape vertices possessing a closest-point distance above this threshold are assigned the distance 0. This signifies the absence of alignment errors between 2D shape and 3D disparity data in this vertex and thus prevents erroneous depth values from influencing the fitted mesh. The threshold is computed dynamically as a function of the average distance of the last fitted 3D mesh using an iterative reweighting scheme for each 3D mesh vertex, similar to the work of [6]. We associate a weight with each mesh vertex and recompute these weights whenever the disparity data or the face mesh have changed. The weights are inversely proportional to the mean distance of each vertex of the last couple of optimally fitted face meshes to its corresponding closest points in the disparity data.

3.3 Fitting to 3D Disparity Data

Analogue to 2D and 2D+3D fitting, we now wish to determine how well our 3D shape representation corresponds to the depth data obtained from a stereo camera. Moreover, we would like to adapt the 3D shape and its underlying 2D shape to better fit the 3D data by removing noise and outliers. This goal can be achieved by introducing yet another constraint into the combined 2D+3D minimization step described in 3.1. Together with the other two expressions, we now wish to minimize

$$D = \| \underbrace{(\bar{s_0} + \sum_{i=1}^{\bar{m}} \bar{p_i} \bar{b_i})}_{=\bar{\mathbf{s}}} - \mathbf{c} \cdot \underset{3D}{CP}(\mathbf{c}^{-1} \underbrace{(\bar{s_0} + \sum_{i=1}^{\bar{m}} \bar{p_i} \bar{b_i})}_{=\bar{\mathbf{s}}}) \|^2, \quad (6)$$

where $\bar{\mathbf{s}}$ represents an instance of the 3D shape *after* applying the 3D pose transformation (we can easily compute the corresponding $3D \rightarrow 3D$ transformation matrix as a byproduct of the 2D+3D extension described in section 3.1), and $\mathbf{c}$ is a constant performing scale change from the camera coordinate system to the frame of reference of our 3D shape. However, $\mathbf{c}$ can assume matrix form when a more general coordinate transformation is needed.

We chose to perform minimization of expression (6) using steepest gradient descent. The steepest descent approach to iteratively minimizing the above D can be formulated as follows:

$\bar{\mathbf{p}}_{\mathbf{3D}}$ is the current 3D parameter set, and the initial value for the 3D parameter update $\triangle \bar{\mathbf{p}}_{\mathbf{stereo}}$ is the value computed on the 2D+3D extension level, $\triangle \bar{\mathbf{p}}_{\mathbf{3D}}$. This value is then modified using

$$\triangle \bar{\mathbf{p}}_{\mathbf{stereo}} = -H_{stereo}^{-1} \cdot \left(\triangle \bar{\mathbf{p}}_{\mathbf{3D}} + G \cdot \sum_{k \in \{x,y,z\}} \sum_{i=1}^{\bar{n}} (\frac{\partial D_{k,i}}{\partial \bar{\mathbf{p}}})^T D_{k,i}(\bar{\mathbf{p}}_{\mathbf{3D}}) \right), \quad (7)$$

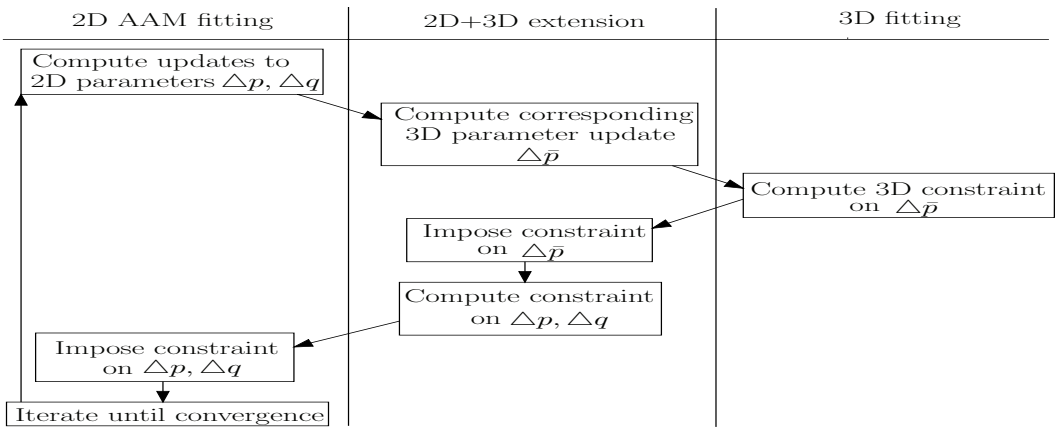

Figure 1. The interaction between different levels of model fitting: based on the 2D AAM (on the left), we create an equivalent 3D model (center), adapt it to dense 3D disparity data (on the right), and feed the resulting changes back into the 2D AAM.

where each row of the Jacobi matrix of the distance function D has the form

$$\frac{\partial D_{k,i}}{\partial \bar{\mathbf{p}}} = \bar{b}_{k,i} - \frac{\partial CP_{k,i}}{\partial \bar{\mathbf{p}}}, \tag{8}$$

where $\bar{b}_{k,i}$ denotes the components of the coordinate k in the ith 3D shape base vector $\bar{b}$. The contribution of the closest-point function CP is

$$\frac{\partial CP}{\partial \bar{\mathbf{p}}} = \frac{\partial CP}{\partial \bar{s}} \underbrace{\frac{\partial \bar{s}}{\partial \bar{\mathbf{p}}}}_{=\bar{\mathbf{b}}}, \tag{9}$$

where $\bar{\mathbf{b}}$ denotes the 3D shape base vectors. Finally, the Hessian matrix is computed from

$$H_{stereo} = H_{3D,\bar{p}} + G \cdot \sum_{k\in\{x,y,z\}} \sum_{i=1}^{\bar{n}} [\frac{\partial D_{k,i}}{\partial \bar{\mathbf{p}}}]^T [\frac{\partial D_{k,i}}{\partial \bar{\mathbf{p}}}], \tag{10}$$

where $\bar{n}$ denotes the number of 3D shape parameters and $H_{3D,\bar{p}}$ is the *submatrix* of the Hessian of the 2D+3D extension step that corresponds to the 3D shape parameters.

The constant G allows weighting the contribution of the stereo fitting to the final 3D shape parameters similar to K in the 2D+3D extension step. Increasing G results in hardening the influence of the stereo fitting constraints on the 2D model.

Since the result of the nearest neighbor search depends on the 3D shape, we have to reevaluate in each iteration the steepest descent geometries on the stereo data as well as on the 3D shape. The Jacobian of the closest-point lookup function, $\frac{\partial CP}{\partial \bar{\mathbf{p}}}$, can be interpreted as the influence of a small change $\triangle \bar{\mathbf{p}}$ to the 3D shape parameters on the computation of the nearest neighbor for each vertex of the 3D shape $\bar{\mathbf{s}}$. We use the following approximation in the implementation: we compose the shape corresponding to the current 3D parameters with a sequence of small changes, each of which has one parameter set to a small value and all other parameters set to 0. We then perform a nearest neighbor search for the resulting 3D shape and retain the change to the vertex locations. This operation has to be performed in each iteration and for each 3D parameter once on all shape vertices.

Changes to the 3D parameters that are a consequence of the alignment to the disparity data are now fed back to the 2D+3D extension computation and constrain the 3D shape. This, in turn, modifies the 2D shape to better align with the 2D intensity image and the 3D disparity data at the same time.

3.4 An Overview of the Fitting Procedure

Figure 1 illustrates which fitting steps have been performed and how different levels interact with each other.

We initially create and align a 2D AAM by computing updates to the parameters $\mathbf{p}$ and $\mathbf{q}$ (figure 1, left column) for shape and pose of the 2D model. Fitting on the 2D AAM level is fast, but might not be accurate for low resolution input images, and for faces which have not been trained or not been trained well.

The 2D+3D extension (figure 1, center) provides the link between 2D and 3D shape representation and enables feedback of changes into the 2D fitting by adapting the 3D model representation, which is controlled by the parameter set $\bar{\mathbf{p}}$, to align with the 2D model equivalently.

The 3D fitting (figure 1, right column) then updates the 3D parameter $\bar{\mathbf{p}}$ to reduce distance between the 3D model mesh vertices and the 3D disparity data by imposing the constraint described in section 3.3 on the 3D model. The 3D model now aligns properly even if the original AAM has not been trained on the appearance of the face or if its texture resolution is low. 3D fitting is relatively slow, since the steepest descent geometry on the dense 3D data has to be recomputed for each fitting step. However, since the 2D fitting already provides a good initialization, only few iterations have to be performed on the 3D fitting level.

In order to feed-back the changes to the 3D model into the 2D alignment process, we compute the updates to the 2D parameters $\mathbf{p}$ and $\mathbf{q}$ which correspond to the changes imposed on $\bar{\mathbf{p}}$, and create a new, better aligned 2D model instance. The process is iterated until the distance between the 3D model shape and the 3D disparity data is smaller than a preset threshold.

4 Experimental Results

In order to demonstrate the feasibility of the proposed method, we have conducted a sequence of evaluations in order to compare the fitting results of the proposed method to those of traditional 2D approaches using the same data. We further evaluated the method using image sequences captured from meetings. The data collections were part of multimodal meeting data collections. Five one-hour long meetings have been recorded using 8 calibrated cameras and microphones. During four meetings, we placed a stereo camera (Point Grey Bumblebee camera) on the meeting table to track different meeting participants. The data was recorded at 10 frames/second with a resolution of 320x240. While the proposed method could successfully track precise movements from low resolution video images, traditional 2D tracking methods were not able to converge in many cases. Furthermore, we applied SVMs [11] on the tracking results to classify emotions. For all experiments, we used a face mesh consisting of 68 feature points as suggested by [10].

4.1 Comparison Studies

In order to test our method under well-defined, reproducible conditions, we implemented a testing tool allowing for the generation of face instances using given 2D and 3D models. The user can specify the shape and pose parameters and a Gaussian noise level and can thus reproduce all shape and appearance modes offered by the underlying models. The created 3D face is subsampled using the resolution of a typical stereo camera. We then use the artificial face instance as input for our fitting algorithm. Initialization can be performed manually by the user. Since the exact shape and appearance parameters used for the creation of the generic face are known, we can easily compare them to the resulting parameter sets computed by our algorithm. Furthermore, we wish to compare fitting results of the 2D method and the stereo-extended 3D method. We therefore compute both fitted meshes and their corresponding residual errors with respect to the mesh based on the ground truth generic parameters. Errors are computed in pixels in city block distance between the 2D shape and the 2D-backprojected 3D shape.

Figure 2 compares the fitting results for a generic face displaying a large shape deformation, which results in local texture deformations as well as changes to the external face contour. The initial mesh is shown on the left of figure 2, the white 2D fitted mesh in the center and the 3D-enhanced fitting on the right. The black 3D-enhanced mesh is drawn on all three face images to allow for a direct comparison between both methods. The example images have been up-scaled for better visualization. For an initialization with 10% vertical and horizontal offset and 80% scale, 2D fitting converges with 2.8 pixels of average residual error per mesh vertex (in city block distance). Our 3D-enhanced fitting achieves an average residual error of 1.5 pixels per vertex after aligning to the disparity data, thus improving the alignment by 46%.

Table 1 lists a series of experiments using the above described ground truth tool and compares the residual errors.

4.2 Face Tracking in Meetings

We collected video and depth data of different participants during meetings and tracked their facial feature motions with our system. The emphasis of these recordings was to test our system under realistic conditions showing fast head movements and authentic expressions, in different environment settings and under varying camera distances and viewing angles and thereby identifying the limitations of our current system.

At a speed of 10 frames per second on a 2Ghz P4 machine, we were able to follow the faces and localize the main face features correctly for faces covering an image surface of approximately 80x80 pixels, while for faces with a resolution of 60x60 pixels or less, we could still follow the face contour, but no longer reliably identify the individual facial expressions. Our system used the same underlying 2D model for all face geometries while sustaining occlusions due to glasses, gestures and coffee mugs. Whenever the initial 2D AAM was unable to reduce the residual fitting error due to a facial geometry or texture that differed significantly from the training faces, the 3D alignment successfully adapted its parameters to represent the unseen face instance accurately. Changes in lighting intensity and direction could be sustained to a certain degree, as long as

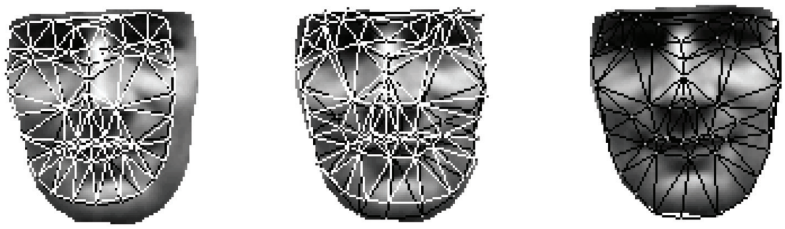

Figure 2. A ground truth example: initialization of a smiling face with initial 2D (white) and 3D (black) meshes (on the left), the aligned 2D mesh (white, center, with final 3D-enhanced black mesh for qualitative comparison), and the 3D-enhanced mesh (black, on the right).

Table 1. Residual errors for generic ground truth experiments

experiment			avg. residual error per vertex [pixels]		improvement [%]
No.	pose	shape	2D only	stereo-extended 3D	
1	neutral	neutral	1.4	1	28.6
2	neutral	smile	2.8	1.5	46.4
3	neutral	eyes	2.2	0.8	63.6
4	neutral	brows	0.9	0.8	11.1
5	neutral	mouth	2.1	1.3	38.1
6	rotation	neutral	2.2	1.6	27.3
7	rotation	smile	3.4	2.9	14.7
8	rotation	eyes	2.6	1.9	26.9
9	rotation	brows	2.6	2	23.1
10	rotation	mouth	2.7	2.1	22.2
11	turn	neutral	2.6	1.8	28.6
12	turn	smile	3.6	3.1	13.9
13	turn	eyes	2.7	1.9	29.6
14	turn	brows	2.9	2.5	13.8
15	turn	mouth	2.7	2.3	14.8
16	rotation	all 4	3.7	2.9	21.6
17	turn	all 4	3.6	3.1	13.9
18	turn+rot	all 4	4.5	3.5	22.2
...					
average over 50 exp.			2.6	2	23.1

the global color distribution within the face bore some resemblance to the training texture. Additionally, the iterative reweighting scheme was able to deal with strong locally confined texture differences, such as partial occlusion. We have experimentally determined that the model mesh still converged for occlusions of up to 30% of the face surface.

Figure 4 shows three tracking examples taken at different camera distances, figure 5 illustrates the stability of our method during a large pose change, and figure 6 shows how the iterative reweighting scheme holds the mesh in place during a strong occlusion by adapting the weights associated with the distance of each of the 3D mesh vertices to the 3D disparity data. To show the structure of the fitted shapes more clearly, we display them again on the top left corner as white meshes.

Figure 3 qualitatively compares some fitting results of a 2D AAM (white mesh) and our 3D-enhanced method (black mesh) for a head turn (on the left) and raised eyebrows (on the right). As to the head turn, the 2D AAM adapts its position on the face correctly, but fails to detect the 3D head movement, while our 3D-enhanced method forces the model into the correct pose. 2D AAMs typically have difficulties in detecting 3D movements, while our 3D alignment to disparity data is very well suited for this task. The second example in figure 3, a raising of the eyebrows, illustrates the tendency of the 2D AAM to change the scale of the model rather than to converge towards the correct shape deformation mode, thus failing to properly align with the face contour. Our 3D-enhanced method converges towards the proper shape deformation mode while at the same time taking into account the face contour, since the contours of the face result in characteristic disparity values that are easy to detect.

4.3 An Application to Emotion Classification

As an application example of our tracking system, we trained SVMs [11] using a *Radial Basis Function* kernel on differentiating between the 6 elementary emotions, i.e. surprise, joy, sorrow, fear, anger and disgust in addition to a neutral mean-shape, mean-appearance expression. For the emotions to be classified, we have tried to choose expressions already introduced in previous publications [7]. Our implementation is based on the library developed by [3]. Since we use a model-based tracking approach, the complex problem of describing facial expressions is reduced to classifying low-dimensional model parameter vectors. More-

over, since shape and pose representations are described by independent parameter sets, we are able to classify shape deformations independently of the occurring pose changes. We have chosen to classify the shape parameter vector of each video frame independently of previous frames and parameters. Although as a consequence we loose the transition context of a given emotion, we could show that due to the tracking precision of our stereo-enhanced system, we can still achieve surprisingly precise classification results.

The training was performed using tenfold cross-validation on a training sequence of 250 to 300 characteristic parameter state vectors, yielding a training precision of 87%. All parameter vectors were linearly scaled to $[-1, 1]$ in order to prevent attributes in greater numeric ranges from dominating those in smaller ranges and to avoid numerical instabilities.

Table 2 shows the classification results of a video sequence with manually labeled ground truth. Classification precision of roughly 71% is acceptable, despite the simple context-unaware approach used. Some emotions, in particular the pairs sorrow/fear and surprise/fear are inherently difficult to separate since they are based on similar muscle movements, while other emotions, such as disgust, result in characteristic texture and shape changes which make them easy to apprehend.

Table 2. Classification of emotion expressions in unseen data

emotion	detection rate
neutral	75%
surprise	68%
joy	76%
sorrow	63%
fear	61%
anger	74%
disgust	82%
avg.	71.3%

5 Conclusions

Given the results obtained during the experiments, we can conclude that incorporating 3D information, even in the presence of noise, into an existing AAM can significantly improve precision and stability. Moreover, the proposed approach relaxes the precision requirements of AAMs so that it is possible to use a generic AAM as an initial model and adapt it to different faces in real-time. By combining ease of 2D model creation and tracking reliability, the described method could serve as a means of improving usability of Active Appearance Models in real-world applications. However, more work is required to tackle the remaining issues, notably by dealing with the instability of the 2D AAM used for initialization of our system when working on low-resolution images and by increasing processing speed of the 3D alignment. Future work could focus on adding mesh points to the models on-line as well as on developing an improved context-aware unsupervised classification approach for facial expression analysis.

Acknowledgements

This research was partially supported by the European Commission within the project CHIL under contract No. 506909, and the National Science Foundation (USA) under Grants No. IIS-0205219 and IIS-0534625. The first author was also partially supported by the interACT scholarship.

References

[1] S. Baker and I. Matthews. Active appearance models revisited. *Int. Journal of Computer Vision*, pages 60:135–164, 2004.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH'99*, pages 187–194, 1999.

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: A Library for Support Vector Machines*, Apr. 2005. http://www.csie.ntu.edu.tw/cjlin/libsvm/.

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE PAMI*, pages 23:681–685, 2001.

[5] G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *Proc. CVPR'99*, pages 2:459–464, 1999.

[6] P. Holland and R. Welsch. Robust regression using iteratively reweighted least squares. *Commun. Stat.- Theor. Meth.*, pages A6:813–828, 1977.

[7] P. Michel and R. El Kalioubry. Real time facial expression recognition in video using support vector machines. In *Proc. ICMI'03*, pages 258–264, 2003.

[8] L.-P. Morency and T. Darrell. Stereo tracking using icp and normal flow constraint. In *Proc. ICPR'02*, pages 4:367–372, 2002.

[9] D. M. Mount and S. Arya. *ANN: A Library for Approximate Nearest Neighbor Searching*, May 2005. http://www.cs.umd.edu/mount/ANN/.

[10] I. S. Pandzic and R. Forchheimer. *MPEG-4 Facial Animation - the Standard, Implementation and Applications*. John Wiley and Sons, 2002.

[11] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[12] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *Proc. CVPR'04*, pages 2:535–542, 2004.

[13] J. Xiao, J.-X. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *Proc. ECCV'04*, pages 4:573–587, 2004.

[14] J. Xiao and T. Kanade. Non-rigid shape and motion recovery: Degenerate deformations. In *Proc. CVPR'04*, pages 1:668–675, 2004.

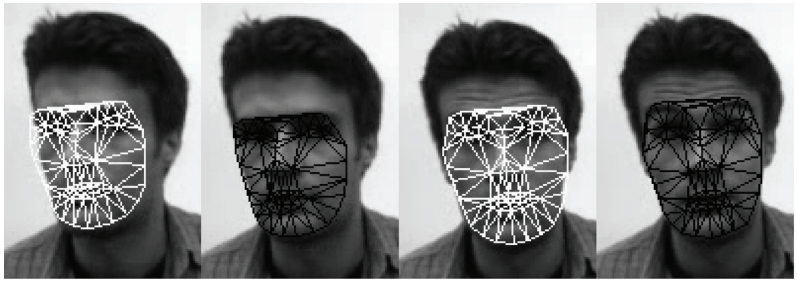

Figure 3. A qualitative comparison of 2D (white mesh) and 3D fitting results (black mesh) for a head turn (left) and a raising of the eyebrows (right).

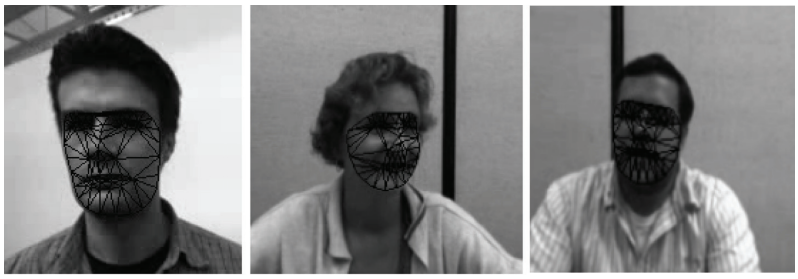

Figure 4. Staged meeting recordings: successful fitting of the same 3D-enhanced AAM on three different faces at different resolutions and under varying lighting conditions.

Figure 5. Stability during pose changes: head turn while moving quickly away from the camera.

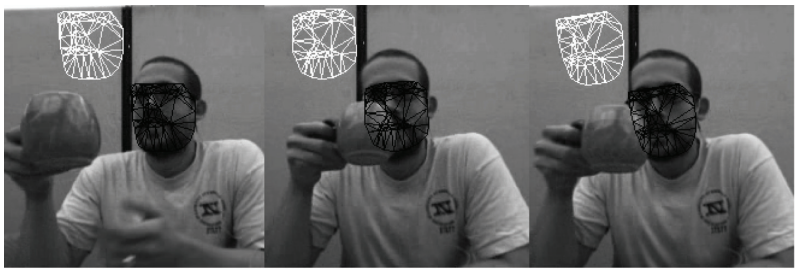

Figure 6. An illustration of partial occlusion: the 3D iterative reweighting scheme holds the mesh in place during partial occlusions of the face.