Occluded Facial Expression Tracking

Hugo Mercier[1], Julien Peyras[2], and Patrice Dalle[1]

[1] Institut de Recherche en Informatique de Toulouse
118, route de Narbonne, F-31062 Toulouse Cedex 9
[2] Dipartimento di Scienze dell'Informazione
via Comelico 39/41, I-20135 Milano
{mercier,peyras,dalle}@irit.fr

Abstract. The work presented here takes place in the field of computer aided analysis of facial expressions displayed in sign language videos. We use Active Appearance Models to model a face and its variations of shape and texture caused by expressions. The *inverse compositional* algorithm is used to accurately fit an AAM to the face seen on each video frame. In the context of sign language communication, the signer's face is frequently occluded, mainly by hands. A facial expression tracker has then to be robust to occlusions. We propose to rely on a robust variant of the AAM fitting algorithm to explicitly model the noise introduced by occlusions. Our main contribution is the automatic detection of hand occlusions. The idea is to model the behavior of the fitting algorithm on unoccluded faces, by means of residual image statistics, and to detect occlusions as being what is not explained by this model. We use residual parameters with respect to the fitting iteration *i.e.*, the AAM distance to the solution, which greatly improves occlusion detection compared to the use of fixed parameters. We also propose a robust tracking strategy used when occlusions are too important on a video frame, to ensure a good initialization for the next frame.

Key words: Active Appearance Model; occlusion; facial expression; tracking; inverse compositional

1 Introduction

We use a formalism called Active Appearance Models (AAM – [1, 2]) to model a face and its variations caused by expressions, in term of deformations of a set of vertex points of a shape model. These points can be tracked with a good accuracy along a video when the face is not occluded and when it has been learned beforehand.

We focus here on the analysis of sign language videos. In sign language, facial expressions play an important role and numerous signs are displayed near the signer's face. Furthermore, the signer's skull frequently performs out-of-plane rotations.This implies, from the interlocutor's point of view (here replaced by the video acquiring system) that face might often be viewed only partially.

Past works mainly focused on robust variants of AAM fitting algorithms ([3], [4]) able to consider outlier data. We follow here the approach developed in [5],

where parametric models of residual image are used in order to automatically detect the *localization* of occlusions. The main idea is here to learn various parameters computed from various fitting contexts and to select one in particular at each iteration, which greatly improves occlusion detection compared to the use of only one fixed parameter in earlier work,

In section 2 are presented Active Appearance Models and the way they are used to extract facial deformations of a face with an accurate optimization algorithm that can take occlusions into account by means of a pixel confidence map. In section 3 we show, through experiments, how to optimally compute the pixel confidence map to detect occlusions. Section 4 describes a robust tracking strategy that we use to track facial deformations along a video sequence.

2 Active Appearance Models

An Active Appearance Model (AAM) describes an object of a predefined class as a shape and a texture. Each object, for a given class, can be represented by its shape, namely a set of 2D coordinates of a fixed number of interest points, and a texture, namely the set of pixels lying in the convex hull of the shape.

The shape can be described by:

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

where $\mathbf{s}_0$ is the mean shape, $\mathbf{s}_i$ are deformation vectors and $\mathbf{p}_i^s$ are weighting coefficients of these deformations. It can be written in matrix notation by $\mathbf{s} = \mathbf{s}_0 + \mathbf{S}\mathbf{p}^s$.

The texture is described by:

$$\mathbf{t} = \mathbf{t}_0 + \sum_{i=1}^{m} \mathbf{p}_i^t \mathbf{t}_i \tag{2}$$

or, in matrix notation $\mathbf{t} = \mathbf{t}_0 + \mathbf{T}\mathbf{p}^t$

The model is built upon a training set of faces, where a shape *i.e.*, 2D coordinates of a fixed set of interest points, is associated to each image. All the shapes are extracted from the training set and global geometric deformations are differentiated from facial deformations by a Procrustes analysis. It results a mean shape $\mathbf{s}_0$ and shapes that differ from the mean only by internal deformations.

Pixels that lie inside the shape of each face is then extracted and piecewise-affine-warped to the mean shape $\mathbf{s}_0$ to build the (shape-free) texture associated to a face.

Principal Component Analysis is applied both to aligned shapes and aligned textures and the eigen-vectors form the matrices $\mathbf{S}$ and $\mathbf{T}$. In our case, we retain enough eigen-vectors to explain 95% of the shape and texture variance (corresponding to 12 shape deformation vectors and 15 texture variation vectors).

A face close to the training set can then be represented by a vector of shape parameters $\mathbf{p}^s$ and a vector of texture parameters $\mathbf{p}^t$.

2.1 Weighted Inverse Compositional Algorithm

The goal of the AAM fitting algorithm is to find $\mathbf{p}^s$ and $\mathbf{p}^t$ that best describes the face seen on an input image. The shape and texture parameters are optimized by means of a residual image that represents differences between a current face estimation and the face seen on the input image I:

$$E(\mathbf{x}) = \mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{p}_i^t \mathbf{t}_i(\mathbf{x}) - I(W(\mathbf{x}; \mathbf{p}^s)), \forall \mathbf{x} \in \mathbf{s}_0 \tag{3}$$

$I(W(\mathbf{x}, \mathbf{p}^s))$ is the projection of the input image onto the mean shape $\mathbf{s}_0$, obtained by a piecewise affine warp. Instead of the Euclidean norm classically used in optimization, we can use a weighted distance:

$$\sum_{\mathbf{x}} Q(\mathbf{x}) E(\mathbf{x})^2$$

where $Q(\mathbf{x})$ weights the influence of the pixel $\mathbf{x}$.

We use the optimization scheme presented in [2], called the *inverse compositional* algorithm, which is efficient and accurate. Its main advantage is the fact that the jacobian matrix can be analytically derived, rather than learned by numerical differentiation (like in [1]).

Among all the variants proposed by the authors, we choose the *simultaneous inverse compositional* algorithm with a weighted distance. The *simultaneous* is a variant that can optimize both shape and texture parameters in an accurate manner. This is not the most efficient variant of the *inverse compositional* algorithms that can deal with texture variations (see for instance the *project-out* algorithm in [2]), but the most accurate.

Iterative update is given by (computation details can be found in [3] and [6]):

$$[\Delta \mathbf{p}^s, \Delta \mathbf{p}^t] = -H_Q^{-1} \sum_{\mathbf{x}} Q(\mathbf{x}) \left[G_s(\mathbf{x}), G_t(\mathbf{x})\right] E(\mathbf{x}) \tag{4}$$

with

$$\begin{aligned} G_s(\mathbf{x}) &= \left[(\nabla \mathbf{t}_0(\mathbf{x}) + \textstyle\sum_{i=1}^{m} \mathbf{p}_i^t \nabla \mathbf{t}_i(\mathbf{x})) \frac{\partial W}{\partial \mathbf{p}_1^s}, \ldots, (\nabla \mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{p}_i^t \nabla \mathbf{t}_i(\mathbf{x})) \frac{\partial W}{\partial \mathbf{p}_n^s}\right] \\ G_t(\mathbf{x}) &= [\mathbf{t}_1(\mathbf{x}), \ldots, \mathbf{t}_m(\mathbf{x})] \\ H_Q &= \textstyle\sum_{\mathbf{x}} Q(\mathbf{x}) \left[G_s(\mathbf{x}), G_t(\mathbf{x})\right]^T \left[G_s(\mathbf{x}), G_t(\mathbf{x})\right] \end{aligned}$$

Shape parameters are then updated by inversion and composition:

$$W(\mathbf{x}; \mathbf{p}^s) \leftarrow W(\mathbf{x}; \mathbf{p}^s) \circ W(\mathbf{x}; \Delta \mathbf{p}^s)^{-1}$$

And texture parameters are updated in an additive way by $\mathbf{p}^t \leftarrow \mathbf{p}^t + \Delta \mathbf{p}^t$. Essential steps of the algorithm are summarized on Fig. 1.

This algorithm performs accurately. For our experiments, we use what is called a person-specific AAM, meaning that the training set is composed by expressions of only one person. A more generic AAM would be less accurate and hard to control.

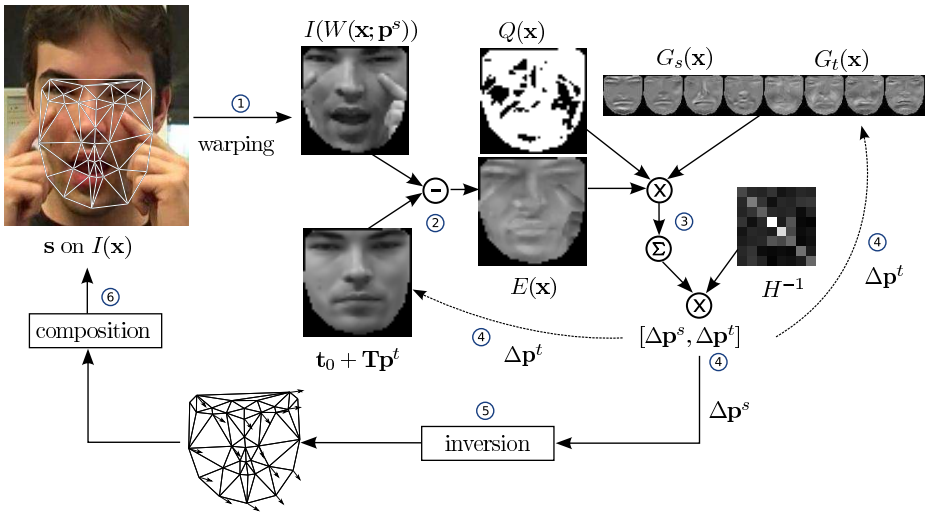

Fig. 1. Essential steps of the weighted simultaneous inverse compositional algorithm. Numbers give chronology of the steps for one iteration.

3 Occlusion Detection

The confidence map $Q(\mathbf{x})$ used in the weighted variant of the AAM fitting algorithm has to be as close as possible to the real occlusion map.

Our problem is to compute the best confidence map without knowledge on the localization of real occlusions. We propose here to model the behavior of the residual image in the unoccluded case and to detect occlusions as being what is not explained by the model, following the approach presented in [5].

3.1 Parametric Models of Residuals

We rely on parametric models of the residual image. We propose to test different confidence map computation functions:

$$Q_1(\mathbf{x}) = \begin{cases} 1 \text{ if } min(\mathbf{x}) \leq E(\mathbf{x}) \leq max(\mathbf{x}) \\ 0 \text{ else} \end{cases}$$

$$Q_2(\mathbf{x}) = \frac{1}{\sigma(\mathbf{x})\sqrt{2\pi}} e^{\left(-\frac{E(\mathbf{x})^2}{2\sigma(\mathbf{x})^2}\right)}$$

$$Q_3(\mathbf{x}) = \begin{cases} 1 \text{ if } |E(\mathbf{x})| \leq 3\sigma(\mathbf{x}) \\ 0 \text{ else} \end{cases}$$

$$Q_4(\mathbf{x}) = \begin{cases} 1 \text{ if } |E(\mathbf{x})| \leq 4\sigma(\mathbf{x}) \\ 0 \text{ else} \end{cases}$$

$$Q_5(\mathbf{x}) = e^{\left(-\frac{E(\mathbf{x})^2}{2\sigma(\mathbf{x})^2}\right)}$$

where $min(\mathbf{x})$ is the minimum value of the pixel $\mathbf{x}$ over all the residual images, $max(\mathbf{x})$ is the maximum value and $\sigma(\mathbf{x})$ is the standard deviation. One of each parameter (min, max and σ) are computed for each pixel $\mathbf{x}$ of the residual image.

The parameters of the Q_i functions could be learned from a random amount of residuals generated when the AAM fitting algorithm is run on unoccluded images. However, a residual image generated when the shape model is far from the solution is totally different from a residual image generated when the model is close to the solution.

That is why the parameters used in the computation of the Q_i functions have to depend on the distance to the solution: they have to be high (resulting in a permissive toward errors Q_i function) when the model is far from the solution and low when it gets closer (resulting in a strict function).

3.2 Partitioned Training Sets

To explicit the link between the model parameters and the distance to the solution, we conducted the following experiment.

A set of residual images are generated: the (non-weighted) AAM fitting algorithm is launched from perturbed ground truth shapes 15 iterations until convergence. To initialize the AAM, each vertex coordinate is perturbed by a Gaussian noise with 10 different variances (between 5 and 30), and the $\mathbf{p}^s$ parameters are obtained by projecting the perturbed shape model onto the shape basis $\mathbf{S}$. It is launched 4 times on 25 images that belong to the AAM training set. The distance to the solution, computed by the average Euclidean distance of the shape model vertices to the optimal ground truth shape vertices, and the residual image are stored at each iteration.

Instead of computing the model parameters ($min(\mathbf{x})$, $max(\mathbf{x})$ and $\sigma(\mathbf{x})$) on all the residual images, we form 15 partitions by regrouping residual images according to their corresponding distance to the solution. Each partition P_i contains 210 residual images and can be characterized by its minimum d_i^- and maximum distance d_i^+ to the solution. The model parameters are then learned, for each pixel $\mathbf{x}$, on all the residuals of each partition.

On figure 2 are represented standard deviations $\sigma(\mathbf{x})$ learned on each partition. For visualization purpose, only the average standard deviation σ, computed over all the pixels $\mathbf{x}$ is plotted.

3.3 Model Parameter Approximation

When the fitting algorithm is run on test face images, that might be occluded, the model distance to the solution is difficult to estimate. In the unoccluded case, a rough estimate of the distance to the solution can be extracted from the residual image. Such an information is not reliable anymore in the occluded case,

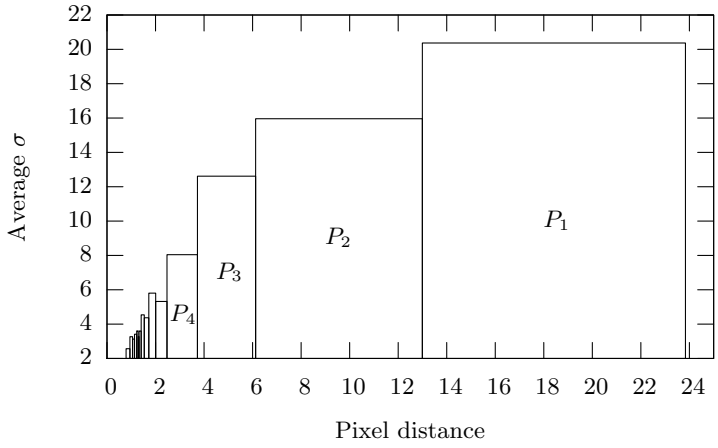

Fig. 2. Average standard deviation learned for each partition.

because the residual image reflects either errors caused by a misplacement of the model or errors caused by occlusions.

However, we assume that we can rely on the iteration number of the fitting algorithm to select the appropriate partition, especially if the model distance to the solution in the occluded case is lower than the maximum distance used to collect residuals in the first partition.

To validate this assumption, we proceed to the test that follows. Using variances computed for each of the 15 partitions, we test the weighted fitting algorithm launched for 20 iterations from Gaussian perturbed optimal positions (with a variance of 20) on occluded (known) images (25% of the input image is covered with 8×8 blocks of pixels of random intensity). Note that the amount of shape perturbations is less important than the amount used in the partition construction. Among all the $Q_i(\mathbf{x})$ functions, we use only $Q_3(\mathbf{x})$ to compute the confidence map at each iteration, for we are only interested in how to select its parameter, not how it performs. Different ways of getting the variance at each iteration are tested:

- S_{real}: selection from P_i where the real distance to the solution d_{model} is bounded by the distance range of P_i: $[d_i^-, d_i^+]$; for comparison purpose;
- S_{it}: selection from P_i where i is the current iteration (and $i = 15$ for iterations 15 to 20);
- S_f: selection from P_1;
- S_m: selection from P_7;
- S_l: selection from P_{15}.

The results on Fig. 3 show clearly that the best choice for the residual model parameter computation is S_{real}. It is not usable in practice (the ground truth shape is not *a priori* known), but we can rely on the S_{it} approximation. As a comparison, results are given for the unoccluded case and for fixed variances (S_f, S_m and S_l).

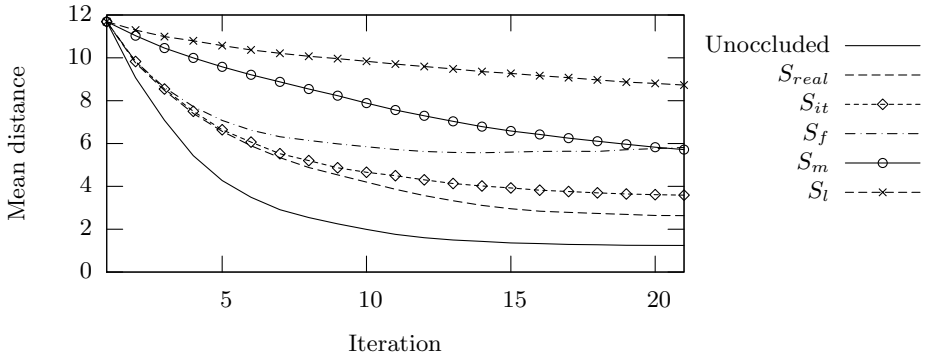

Fig. 3. Average behavior of the fitting algorithm for the reference unoccluded case, and for the occluded case with different computations of the variance.

In [5], variances are fixed and computed on residual images obtained from the converged AAM, which corresponds here to the S_l selection strategy. When observing the mean distance obtained after 20 iterations, the proposed S_{it} variance selection strategy results in a distance divided by about 2 compared to S_l.

3.4 Choice of the Parametric Model

With the previous result we can then test what is the best way to compute the confidence map at each iteration.

For that purpose, we proceed to the following experiment: the weighted AAM fitting algorithm is launched on images of known faces, covered with a varying amount of occlusions, from a Gaussian perturbed shape (we use a variance of 20 for each vertex coordinate). We test each of the Q_i confidence map computation functions with a parameter choosed using S_{it}.

The convergence frequency is determined by computing the number of fittings that result in a shape with a distance to the ground truth lower than 2 pixels.

Results are summarized on Fig. 4. The Q_4 function clearly shows the best results. All the other functions perform worse, except for the function Q_1 that seems to be a good detector in the case of low occlusion rate and a very bad one in the case of high occlusion rate. Q_1 relies on computation of minimum and maximum value, which are very robust measures compared to variance, that is why the behavior of Q_1 is not always reliable.

4 Robust Tracking Strategy

The goal of the tracking algorithm is to take occlusion into consideration as much as possible. However, on some video frame, occlusions are too important to expect good fitting results, because too many pixels are considered unreliable.

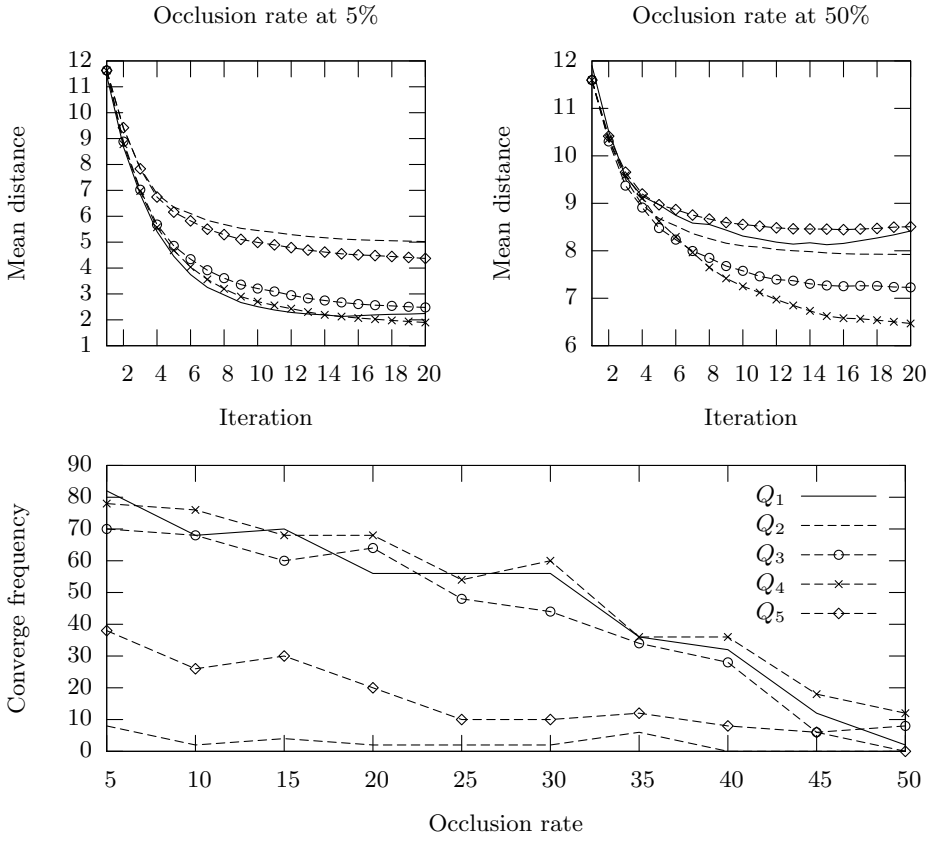

Fig. 4. Characterization of the confidence map computations. Average distance to the solution across iterations for 5% and 50% of occlusions (top curves) and convergence frequency (bottom curve).

In such a case, the fitting algorithm is prone to divergence and the resulting shape configuration could be a bad initialization if used directly in the following frame.

That is why we propose to rely on a measure of divergence and on a rigid AAM to initialize the model.

The goal is to avoid bad configurations of the shape model in order not to perturb the fitting process on subsequent frames. We detect such bad configurations by detecting shapes that are not statistically well explained. For that purpose, we compare the shape parameters $\mathbf{p}^s$ to their standard deviations σ_i, previously learned from the shape training set. The divergence is decided, if:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{|\mathbf{p}_i^s|}{\sigma_i} > \rho_1 \text{ or } \max_{i=1,\ldots,n}\left\{\frac{|\mathbf{p}_i^s|}{\sigma_i}\right\} > \rho_2$$

The thresholds ρ_1 and ρ_2 are determined empirically and can be high (here we choose $\rho_1 = 2.5$ and $\rho_2 = 7.0$). The divergence is only tested after ten iterations, for the model deformations that occur during the first iterations can lead to convergence.

On a frame, if convergence is detected, the final shape configuration is stored and serves as an initialization for the next frame.

If divergence is detected, we rely for the following frame on a very robust tracker: an AAM build by retaining only the geometric deformation vectors. It is represented by the mean shape that can only vary in scale, in-plane rotation and position but not in facial deformations. Such a model gives a rough estimate of the face configuration that can be used as an initialization for the non-rigid AAM. It avoids the non-rigid shape model to being attracted by local minima. The rigid AAM fitting algorithm uses also a confidence map to take occlusions in consideration. However, the confidence maps computed for the non-rigid AAM are too strict for the rigid AAM, we thus use a coarse occlusion detector (for example, the confidence map computed over the second partition for the non-rigid AAM).

The rigid AAM fitting is launched for 5 iterations from the last configuration that converged. The non-rigid AAM fitting algorithm is then launched from the resulting position.

We test this tracking algorithm on a video sequence of about 500 frames where signs frequently occlude the signer's face.

We show some typical results on selected frames (see figure 5). Blocks of white pixels represent areas of occlusions detected by our method. Compared to a naive tracker, the AAM always converges to an accurate configuration on unoccluded frames that occur after an occluded one.

5 Conclusion

We have presented a way to track facial deformations that occur on a video, taking into account hand occlusions by means of an Active Appearance Model of a face, a robust optimization scheme that down-weights pixel contributions in the presence of occlusions, an optimal way to compute the pixel confidence

(a) (b) (c) (d)

Fig. 5. Video tracking results. (a) Example of a good occlusion detection. (b) Example of a divergence. Divergence on a frame (c) and convergence on the next frame (d).

map and a robust tracking strategy based on a measure of divergence and a rigid AAM.

The pixel confidence map is computed based on a model of residual images. We use one model per iteration of the fitting algorithm, rather than one fixed model. This is clearly a better choice that improves occlusion detection, compared to earlier work.

Concerning the tracking test, experiments on convergence frequency of the algorithm with respect to the occlusion rate have still to be conducted.

The video sequence used to test the tracking algorithm contains only weak out-of-plane rotations. This is why the rigid 2D AAM can give a good initialization configuration for the non-rigid AAM fitting algorithm. On realistic sign language videos however, out-of-plane rotations may be important and we would have to rely on a rigid AAM that can take 3D pose into consideration.

We use the most accurate and most time-consuming robust variant of the *inverse compositional* algorithm. We have to investigate if approximations presented in [3], [6] or [5] could be applied to obtain an accurate and efficient facial deformation tracker.

References

1. Cootes, T.F., Edwards, G.J., Taylor, C.J.: 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**(2) (November 2004) 135 – 164
3. Baker, S., Gross, R., Matthews, I., Ishikawa, T.: Lucas-Kanade 20 years on: A unifying framework: Part 2. Technical Report CMU-RI-TR-03-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (February 2003)
4. Gross, R., Matthews, I., Baker, S.: Active appearance models with occlusion. Image and Vision Computing **24**(6) (2006) 593–604
5. Theobald, B.J., Matthews, I., Baker, S.: Evaluating error functions for robust active appearance models. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition. (April 2006) 149 – 154
6. Baker, S., Gross, R., Matthews, I.: Lucas-Kanade 20 years on: A unifying framework: Part 3. Technical Report CMU-RI-TR-03-35, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (November 2003)