Toward an Efficient and Accurate AAM Fitting on Appearance Varying Faces

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

[1]IRIT - Université Paul Sabatier
118 Route de Narbonne, F-31062 Toulouse Cedex 9, France
{mercier,dalle}@irit.fr
[2]Dipartimento di Scienze dell'Informazione
via Comelico 39/41, I-20135 Milano MI, Italia
peyras@dsi.unimi.it

Abstract

Automatic extraction of facial feature deformations (either due to identity change or expression) is a challenging task and could be the base of a facial expression interpretation system. We use Active Appearance Models and the simultaneous inverse compositional algorithm *to extract facial deformations as a starting point and propose a modified version addressing the problem of facial appearance variation in an efficient manner. To consider important variation of facial appearance is a first step toward a realistic facial feature deformation extraction system able to adapt to a new face or to track a face with changing video conditions. Moreover, in order to test fittings, we design an experiment protocol that takes human inaccuracies into account when building a ground truth.*

1. Introduction

Facial expression is one of the most powerful, natural and immediate means for human beings to communicate their emotions and intentions ([6]). Psychological and linguistic studies or applications in Human-Computer Interaction focus on the interpretation of its communicational content. An automatic, efficient and accurate facial expression extraction system would thus be a powerful tool, for these applications or studies.

Since the seminal work of Cootes, Active Appearance Models (AAM) ([3]) have been heavily used to model or track deformable objects and particularly faces ([4, 3, 9, 8]). Matthews *et al.* recently proposed ([8]) the *inverse compositional algorithm*, able to fit an AAM onto an image. Our choice to implement this algorithm was initially motivated by its mathematical correctness. Contrary to [4, 3], the optimization problem is solved using analytically derived gradients, rather than using a numerical estimate. Such properties increase the method reliability as shown in [8].

A first implementation considers shape deformations only and leaves fixed the appearance. It is very efficient and runs in real-time. Unfortunately, such a trivial situation is unlikely: a face is particularly subject to appearance variations from a still image to another, or between frames along a video sequence.

As this problem must be addressed, the authors proposed two other implementations of their algorithm. The first, called *project out algorithm*, deals with shape deformations as well as appearance variations without increasing computational costs. However it can only deal with appearance variations when they are extremely subtle, which is, one more time, very unlikely.

A second solution, called the *simultaneous algorithm* ([1]), provides far more satisfactory results ([5]). It optimizes simultaneously shape and appearance parameters of an AAM placed onto a new image. Its major drawback is its computational cost as the algorithm performs around 30 times slower than the *project out*.

In this paper we investigate the possibility to abandon the poor performing *project out* algorithm and to rely only on a modified version of the *simultaneous* algorithm, opportunely reformulated to increase its efficiency.

We compare the fitting accuracy of the *simultaneous* algorithm and our version by measuring a distance of the fitting to a ground truth. We propose here to rely on a statistically built ground truth and on a fitting error that takes the lack of localization accuracy of facial features into account.

We first introduce Active Appearance Models and the way they can be used to extract facial feature deformations in section 2. Section 3 covers the appearance variation modelling. Section 4 presents our modification to the algorithm and section 5 how it can be integrated into the *simultaneous* algorithm. The experiment protocol and results are given in section 6. Finally section 7 gives a conclusion and directions for future works.

2. Background

Faces and their variations in shape and appearance can be modelled by Active Appearance Models: a mean shape

associated with a mean appearance, where linear deformations are added to them. The two subspaces of shape and appearance variations are typically obtained by a principal component analysis on a previously hand-labelled training set.

2.1. Generative models

An Active Appearance Model describes an object of a predefined class as an instance of shape and an instance of appearance. Each object, for a given class, can be represented by its shape, described by vertex coordinates and an appearance, described by pixel intensities. It is defined by:

1. a shape $\boldsymbol{s} = \boldsymbol{s}_0 + \sum_{i=1}^{n} p_i \boldsymbol{s}_i$, where $\boldsymbol{s}_0$ is the mean shape, $\boldsymbol{s}_i$ are shape deformation vectors and p_i are coefficients that weight deformations; the basis formed by all the $\boldsymbol{s}_i$ vectors is referred to as $\boldsymbol{S}$;
2. an appearance $A(x) = A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x)$, where $A_0(x)$ is the mean appearance image, $A_i(x)$ are the appearance variation vectors and λ_i are their weighting coefficients.

In many implementations, a shape deformation is divided into a global shape deformation, corresponding to 2D similarities (rotation, translation and scale of the model) and a deformation of internal features. Global shape deformation vectors are here referred to as $\boldsymbol{s}_i^*$ forming the $\boldsymbol{S}^*$ basis. To refer to all the shape deformation parameters, we introduce $\boldsymbol{v}$ as being the stacking of a vector $\boldsymbol{q}$ (global shape deformation parameters) and a vector $\boldsymbol{p}$ (internal shape deformation parameters).

We introduce a general notation $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_0 + \sum_i \boldsymbol{\omega}_i \boldsymbol{\Gamma}_i$ where $\boldsymbol{\Gamma}_{(i)}$ stands for $A_{(i)}(x)$, $\boldsymbol{s}_{(i)}$ or $\boldsymbol{s}^*_{(i)}$ indifferently and $\boldsymbol{\omega}$ for $\boldsymbol{\lambda}$, $\boldsymbol{p}$, $\boldsymbol{q}$ or $\boldsymbol{v}$.

2.2. The inverse compositional algorithm

Algorithms based on the AAM paradigm address a solution to the problem of face fitting and facial feature deformation extraction by finding a configuration of shape and appearance on a new image, which minimizes some error measurement.

The formulation given by Baker *et al.* in [8] is based on the Lucas-Kanade image alignment algorithm ([2]), extended to piecewise affine warps. The *inverse compositional algorithm* aims at minimizing the sum of the squared errors:

$$\sum_{x \in \boldsymbol{s}_0} [E(x)]^2 = \sum_{x \in \boldsymbol{s}_0} [T(x) - Proj(I(x,s)_{\boldsymbol{s}_0})]^2 \qquad (1)$$

where $I(x,s)$ is the region of interest of the input image $I(x)$ lying into the shape $\boldsymbol{s}$ and $Proj(.)_{s_0}$ is a projection

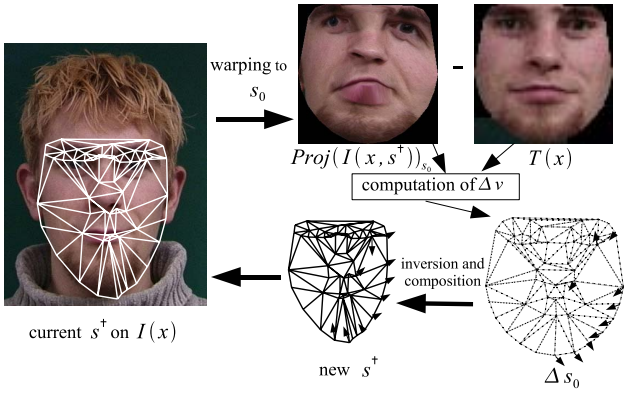


Figure 1. Illustration of the inverse compositional algorithm. Given a current estimate of the shape $\boldsymbol{s}^\dagger$, the mean shape $\boldsymbol{s}_0$ has to be modified by $\Delta \boldsymbol{s}_0$ (computed from $\Delta \boldsymbol{v}$) in order to make it, with the template appearance $T(x)$, closer to $Proj(I(x, \boldsymbol{s}^\dagger))_{\boldsymbol{s}_0}$. The shape modification $\Delta \boldsymbol{s}_0$ must be applied to $\boldsymbol{s}^\dagger$ by inversion and composition to obtain a new shape.

of this ROI to the mean shape $\boldsymbol{s}_0$ and $T(x)$ is the template appearance image.

This is a non-linear optimization problem, solved by an iterative Gauss-Newton method. Assuming initial shape parameters $\boldsymbol{v}$ forming an initial shape estimation $\boldsymbol{s}^\dagger$ are known, they are updated at each iteration by $\Delta \boldsymbol{v}$, in order to minimize the error function. Rather than updating the shape parameters in a "forward way" ($\boldsymbol{v} \leftarrow \boldsymbol{v} + \Delta \boldsymbol{v}$), the inverse compositional algorithm updates them by inversion and composition: $\Delta \boldsymbol{v}$ expresses shape deformations that have to be applied to the mean shape $\boldsymbol{s}_0$ and the next estimated shape $\boldsymbol{s}^\dagger$ is obtained by inversion and composition (see Fig. 1).

At each iteration, $\Delta \boldsymbol{v}$ is obtained from eq. 1 by:

$$\Delta \boldsymbol{v} = \boldsymbol{H}^{-1} \sum_x SD(x)^T E(x) \qquad (2)$$

where:

$$SD(x) = \nabla T \frac{\delta Proj}{\delta \boldsymbol{v}} \qquad (3)$$

$$\boldsymbol{H} = \sum_x \left[\nabla T \frac{\delta Proj}{\delta \boldsymbol{v}} \right]^T \left[\nabla T \frac{\delta Proj}{\delta \boldsymbol{v}} \right] \qquad (4)$$

and where ∇T is the *image gradient* of the template appearance $T(x)$, and $\frac{\delta Proj}{\delta \boldsymbol{v}}$ is the analytically derived Jacobian of $Proj$ (see [8] for the computation details).

Such a formulation allows the precomputation of derivatives: the matrix of first partial derivatives (Jacobian), the

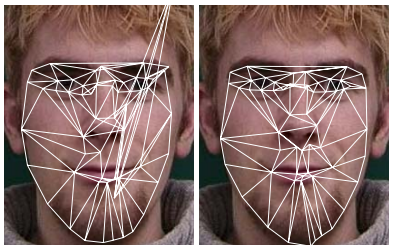


Figure 2. On the left: intermediate shape configuration obtained after some iterations *without* projection, leading to divergence. On the right: the same intermediate shape configuration when using a projection, leading to convergence.

image gradient, and the approximation of the Hessian matrix are precomputed, leading to an efficient algorithm.

According to Fig. 1, the new shape $\boldsymbol{s}^\dagger$ can be used directly in the next iteration. But, due to the approximation of the warping inversion and composition, which is not well-defined for piecewise affine warps (it can be well-defined when a dense model is used – see [9] for details), errors are accumulated during iterations and lead to shape configurations that might be subject to divergence (see Fig. 2).

In our implementation, we thus re-project the shape $\boldsymbol{s}^\dagger$ onto the learnt shape subspaces to smooth accumulated errors. Such a process could also be used to verify whether the obtained shape is a plausible one (relatively to the training set). We do not assume the similarity basis $\boldsymbol{S}^*$ and the shape basis $\boldsymbol{S}$ to be orthogonal, as it is done in [8] (*i.e.*, find $\boldsymbol{q} = \boldsymbol{S}^{*T}(\boldsymbol{s}^\dagger - \boldsymbol{s}_0)$ then find $\boldsymbol{p} = \boldsymbol{S}^T(\boldsymbol{s}^\dagger - \boldsymbol{S}^* q - \boldsymbol{s}_0)$). We prefer to use the exact formulation:

$$[\boldsymbol{p}, \boldsymbol{q}]^T = (\boldsymbol{V}^T\boldsymbol{V})^{-1}\boldsymbol{V}^T(\boldsymbol{s}^\dagger - \boldsymbol{s}_0) \quad (5)$$

where $\boldsymbol{V} = [\boldsymbol{S}, \boldsymbol{S}^*]$ are the two concatenated basis (similarity and shape). Moreover, this latter formulation involves less computation than the former, assuming $(\boldsymbol{V}^T\boldsymbol{V})^{-1}$ is precomputed.

3. Modelling appearance variation

The original inverse compositional algorithm does not treat appearance variation : the face to fit has to be a (piecewise affine) warped version of the *template appearance*. Obviously, this cannot be assumed in realistic cases.

Appearance variation is a relevant problem that occurs when nothing is known about the identity of the face to fit on a still image or when appearance changes between two video frames.

Shape deformations of a 2D model can only represent expressions that involve in-plane feature point displacements. Expressions involving out-of-plane feature point displacements (like cheek puffing for example) are represented by appearance variation within the image formation process.

The Simultaneous Inverse Compositional Algorithm
Iterate:
(1) Compute $Proj(I(x, \boldsymbol{s}^\dagger))_{\boldsymbol{s}_0}$.
(2) Compute the error image $E(x)$.
(3) Compute the gradient $SD_{sim}(x)$.
(4) Compute the Hessian matrix $\boldsymbol{H}$ and invert it.
(5) Compute $\sum_x SD^T_{sim}(x)E(x)$.
(6) Compute $[\Delta\boldsymbol{v}, \Delta\boldsymbol{\lambda}]^T = -\boldsymbol{H}^{-1}\sum_x SD^T_{sim}(x)E(x)$.
(7) Update the new shape $\boldsymbol{s}^\dagger$ with $\Delta\boldsymbol{v}$ and $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}$.

Figure 3. Essential iterative steps of the simultaneous inverse compositional algorithm.

An efficient way to address the problem of appearance variation has been proposed in ([8, 1]), called the *project-out* inverse compositional algorithm. The appearance parameters are assumed not to vary significantly. The shape deformations are considered in a subspace where the appearance does not change. The new appearance parameters can then be "projected-out" once the shape parameters converged.

The version of the algorithm addressing significantly appearance parameter variations (*Simultaneous inverse compositional* in [1]) find optimal appearance and shape parameters simultaneously by minimizing the sum of the squared errors:

$$\sum_{x\in\boldsymbol{s}_0}[E(x)]^2 = \sum_{x\in\boldsymbol{s}_0}\left[A_0(x) + \sum_{i=1}^{m}\lambda_i A_i(x) - Proj(I(x, \boldsymbol{s}^\dagger))_{\boldsymbol{s}_0}\right]^2$$

where $A_0(x) + \sum_{i=1}^m \lambda_i A_i(x)$ now plays the role of the template appearance image.

This is a very slow algorithm due to the need of computation at each iteration (see Fig. 3). The steepest descent images depend on the current appearance parameters and must be computed at each iteration (see [1] for details), as well as the Hessian matrix and its inverse which depend on the steepest descent images.

Because the Hessian computation and inversion (step 4) is the most computationally involving step, we propose here an efficient algorithm to by-pass it.

4. Coefficient regulator

In the original simultaneous algorithm, the inverse Hessian estimates the error function curvature, allowing to approximate this function by a second order polynomial function and the step is adapted consequently to optimize the descent.

4.1. Algorithm

In our modified version of the simultaneous, the basical idea consists in replacing the computation of the update at iteration t (step (6) on figure 3) by the new one :

$$[\Delta \boldsymbol{v}(t), \Delta \boldsymbol{\lambda}(\boldsymbol{t})]^T = -\boldsymbol{C}(t-1) \odot \sum_x SD^T_{sim}(x)E(x)$$

Where $\odot$ is the element-wise product and $\boldsymbol{C}$ is a $(n+m)$ long vector of weighting coefficients c_i that aims at regulating the step along the $\boldsymbol{\Gamma}_i$ directions. For each $\boldsymbol{\Gamma}_i$ direction we only consider the descent sense indicated by the sign of $\Delta\omega_i(t)(= \Delta\omega_i(t) - \omega_i(t-1))$.

As long as $\omega_i(t)$ progresses in the same sense across iterations, we foster its progression by applying $c_i(t) \leftarrow c_i(t-1)\eta_{inc}$.

When the sense of search of $\omega_i(t)$ is inverted (*i.e.*, $\Delta\omega_i(t)$ has an opposite sign with respect to $\Delta\omega_i(t-1)$), it means we passed a minimum of the error function along the $\boldsymbol{\Gamma}_i$ direction. $\omega_i(t)$ has then to pass back onto this minimum with a lower speed *i.e.*, we decrease the weighting coefficient $c_i(t)$ value with respect to its previous value, $c_i(t) \leftarrow c_i(t-1)/\eta_{dec}$.

The global routine is summarized by:

for $i = 1$ to $n+m$ **do**
  if $\Delta\omega_i(t-1)\Delta\omega_i(t) > 0$ **then**
    $c_i(t) \leftarrow c_i(t-1)\eta_{inc}$
  else
    $c_i(t) \leftarrow c_i(t-1)/\eta_{dec}$
  end if
end for

The η_{inc} and η_{dec} constants are determined empirically. One pair is used to regulate the shape coefficients (we use $\eta_{dec} = 2.3$ and $\eta_{inc} = 1.2$) and another one to regulate the appearance coefficients (we use $\eta_{dec} = 6$ and $\eta_{inc} = 2.7$).

4.2. Initialization

The starting coefficient $\boldsymbol{C}(t_0)$ is computed from the inverse Hessian matrix, which is precomputed:

$$\boldsymbol{C}(t_0) = (\frac{1}{a_1}, \ldots, \frac{1}{a_{n+m}})^T \odot (\boldsymbol{H}^{-1}\boldsymbol{a})$$

where $\boldsymbol{a} = (\sum_x SD^T_{sim}E(x))$

Without additional computational costs, the regulator can then take advantage of the first step size given by $\boldsymbol{H}^{-1}$.

5. Modified simultaneous inverse compositional algorithm

At step (4) of the algorithm, the coefficient regulator is used instead of the Hessian inverse which greatly increase computation speed as it is evaluated in $O(n+m)$ against $O((n+m)^2)$ when the Hessian is used.

5.1. Theoretical complexity

Referring to [1], the computational cost of the simultaneous inverse compositional algorithm is in $O((n+m)^2N + (n+m)^3)$ per iteration, where n is the number of shape parameters (including similarity), m is the number of appearance parameters and N is the resolution of the ($\boldsymbol{s}_0$-shaped) image.

With the modified version we propose here, the step (4) is equivalent to the evaluation of the regulator ($O(n+m)$) and the step (6) is in $O(n+m)$, where a $(n+m)$ vector is evaluated, instead of $O((n+m)^2)$, where a $(n+m)\times(n+m)$ matrix is evaluated.

The total computational cost per iteration is now $O(n^2 + (m+n)N)$ instead of $O((n+m)^2N + (n+m)^3)$.

5.2. Practical efficiency

We compare the two algorithms in terms of computational time and find our modified version to be about 6 times faster per iteration than the original one in our implementations (with $N = 9268$, $n = 24$ and $m = 30$). The computational time for the evaluation of the regulator can be neglected before the computation of the Hessian in the simultaneous algorithm. Other parts of both algorithms are common but not optimized, we can therefore hope to see further increase of the speed factor.

6. Empirical evaluation

In this section, we propose to compare performances between the original *simultaneous* algorithm and our version using the coefficient regulator.

6.1. Fitting error

Across iterations, we measure how both algorithms fit faces, by using a *fitting error* based on vertex distances to a given *ground truth shape*.

It is usual in literature to see validation tests based on biased ground-truth towards the method to be tested. In [8], for example, the ground-truth is built with the method to be tested, leading automatically to biased data.

To test the quality of an AAM fitting onto a face image, methods other than visual control are difficult to build. A common mistake consists in benchmarking the fitting result against a manual labelling: it is incorrect to claim that a manual labelling is objectively better than another one.

We introduce a statistical-based method to build the ground truth data. A fitting error is given to a labelling, either manual or automatic, taking into account the degree of accuracy of human experts to manually localize each AAM vertex.

To define the ground truth shape and the fitting error, we rely on a high number n_L of expert labels for each of the n_I face images. In this way, a probability distribution can be computed on each of the n_V vertices. For each image i, the mean $\boldsymbol{\mu}_{i,v}$ of each vertex v is computed over its n_L labels, defining the ground truth shape $(\boldsymbol{\mu}_{i,1}, \ldots, \boldsymbol{\mu}_{i,n_V})$. The covariance $\boldsymbol{\Sigma}_v$ is computed over the $n_L \times n_I$ labels, for each vertex v, considering the distance of each label $\boldsymbol{x}_{i,v,l}$ to its corresponding mean $\boldsymbol{\mu}_{i,v}$.

$$\boldsymbol{\mu}_{i,v} = \frac{1}{n_L} \sum_{l=1}^{n_L} \boldsymbol{x}_{i,v,l}$$

$$\boldsymbol{\Sigma}_v = \frac{1}{n_I n_L - 1} \sum_{i=1}^{n_I} \sum_{l=1}^{n_L} (\boldsymbol{x}_{i,v,l} - \boldsymbol{\mu}_{i,v})^T (\boldsymbol{x}_{i,v,l} - \boldsymbol{\mu}_{i,v})$$

A spread distribution (*i.e.*, having high variances σ_x and σ_y) means a vertex hard to localize or not well-defined. A concentrated distribution (*i.e.*, having low variances) means a vertex well-defined. The idea is to score a labelling (either human or automatic) with respect to the distribution of each vertex coordinates: the higher are the variances, the less is penalized a localization inaccuracy. The obtained covariances are represented by ellipses on Fig. 6.3(e).

The fitting error $e_i(\boldsymbol{s})$ of a shape $\boldsymbol{s}$ on an image i, is defined by the average of the Mahalanobis distances:

$$e_i(\boldsymbol{s}) = \frac{1}{n_V} \sum_{v=1}^{n_V} \sqrt{(\boldsymbol{s}_v - \boldsymbol{\mu}_{i,v})^T \boldsymbol{\Sigma}_v^{-1} (\boldsymbol{s}_v - \boldsymbol{\mu}_{i,v})} \quad (6)$$

where $\boldsymbol{s}_v$ is the v^{th} vertex of the shape $\boldsymbol{s}$.

6.2. Experiment protocol

We collected 40 images from the AR database ([7]) containing only frontal faces, without expression and with global fixed illumination. To build our training set, we manually labelled 11 times (to define the ground truth shape) each image with a landmark configuration describing a mesh equivalent to the one used in [8].

We distinguished two tests. On the first test, we used the 40 ground truth shapes to compute the shape deformation and appearance variation statistics. We ran both algorithms on each image of the training set. As this image contributed to the statistics computation, we will refer to this test as the "known faces test".

The second is a *leave-one-out* test, where, for each test image, the statistics is built on the remaining 39 images. We thus test the generalization ability of both algorithms. It will be referred to as the "unknown faces test".

Anytime a statistics is built, we retain enough shape and appearance eigenvectors to explain 95% of the total variance contained in the training set.

As an initial configuration given to the algorithms, we took the ground truth shape associated to the test image, we extracted the corresponding 2D similarity and shape parameters by projection onto the shape subspaces (see eq. 5) and we set to zero the shape parameters, simulating an initial configuration that could be obtained by a face and feature detector.

For each test image we launched both algorithms and record the error at each time unit. The mean error over all test images is plotted on Fig. 4.

We compare the fitting error to the one obtained by human experts. For each image, a fitting error is associated to each of the 11 manual labellings, according to the eq. 6. We retain the minimum, maximum and average of the 11 labellings for each image. Their mean over all images is superimposed on Fig. 4.

6.3. Results

For the fitting algorithms, performances can be evaluated as the ability to reach the lowest error fitting as fast as possible and to stay as close as possible to this minimum error.

On the known faces test, the Hessian-based simultaneous algorithm performs better than the regulated, as it reaches faster a lower minimum.

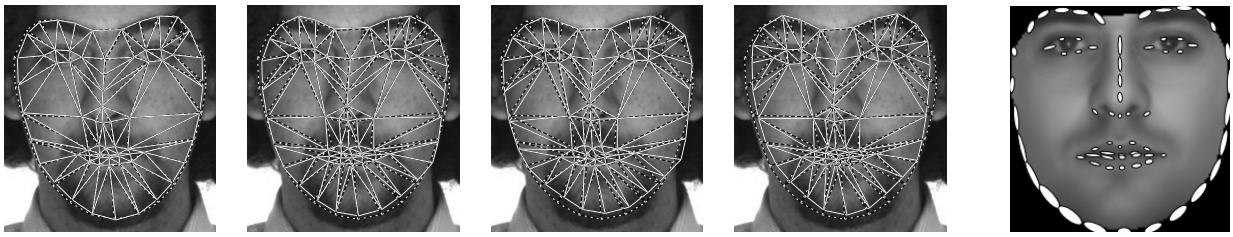
In the unknown faces test, fittings are never as accurate as those obtained in the previous test as can be expected. In both cases, minimum is reached after an equivalent processing time. The Hessian-based algorithm reaches a slightly lower minimum. However, visual fitting differences are weak as can be observed in Fig. 5.

Contrary to the previous test where the error function $E(x)$ simultaneously decreases with the fitting error (not shown due to lack of space), we often observe that minimizing the error function does not correspond to a better fitting. Consequently, reaching of a minimum error fitting does not imply the fitting to stabilize on this minimum: we observe a rise of the fitting error passed the minimum. The observed rise of the Hessian-based algorithm is greater than the rise of the regulated, showing a better stability for our regulated solution.

7. Conclusion and future works

As an issue to the relevant appearance variation problem, we proposed an algorithm[1], based on the *inverse compositional* algorithm, similar to the *simultaneous* version. In

[1] All resources and data are available on the authors' website



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

Figure 5. (a) Typical fitting obtained by the Hessian-based on known faces ($e(\boldsymbol{s}) = 1.16$). (b) and (c) are the best fittings obtained on unknown faces when the minimum error is reached for both the Hessian-based ($e(\boldsymbol{s}) = 2.45$) and the regulated ($e(\boldsymbol{s}) = 2.55$). (d) represents a fitting obtained on a rise ($e(\boldsymbol{s}) = 3.46$). The ground truth shape is plotted in dashed lines.

(e) Labelling accuracy illustration on the mean face for each vertex.

our version the computation of the most involving step is avoided. The resulting complexity is lesser than the original one. Each iteration of our algorithm is still less accurate than the simultaneous algorithm, but the resulting global computational times are equivalent to reach a minimum when the faces are not known.

We proposed a test protocol based on a fitting error that scores accuracy with a tolerance equivalent to these of human experts.

The rise of error fitting observed for the unknown faces test is probably due to the inability of algorithms to deal with non-Gaussian noise, introduced by face features which are unknown to the statistics. We will therefore investigate the use and effects of a robust error function.

Our solution only considers the descent sense of search along the shape deformation and appearance variation directions. We think it can be developed by also considering other characteristics to improve the optimization properties.

It has also to be compared to other variants of the inverse compositional algorithm, as those described in [2]: particularly the steepest descent minimization and the diagonal Hessian approximation.

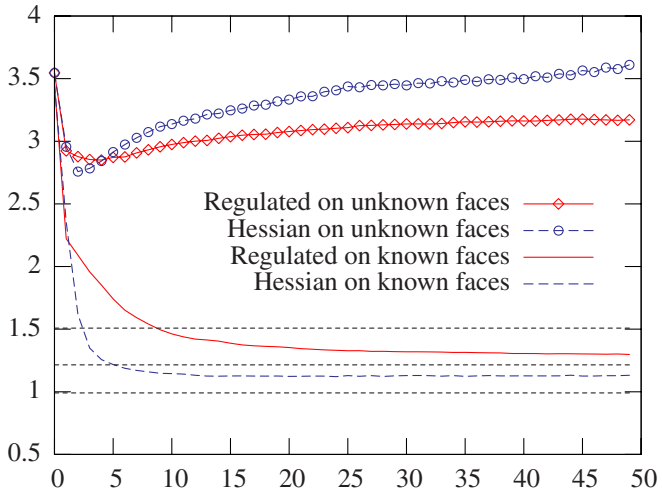


Figure 4. Fitting error evolution across time, for both the known and unknown faces tests.

References

[1] S. Baker, R. Gross, and I. Matthews. 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.

[2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221 – 255, March 2004.

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

[4] F. Dornaika and J. Ahlberg. Fast and reliable active appearance model search for 3-d face tracking. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 34(4):1838– 1853, August 2004.

[5] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(11):1080–1093, November 2005.

[6] Y. li Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001.

[7] A. Martinez and R. Benavente. The AR face database. Technical Report 24, CVC, June 1998.

[8] I. Matthews and S. Baker. Active appearance models revisited. Technical Report CMU-RI-TR-03-02, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 2003.

[9] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *IEEE International Conference on Computer Vision*, 2003.