

# On Compositional Image Alignment with an Application to Active Appearance Models Supplementary Material

Brian Amberg

Andrew Blake

Thomas Vetter

## Abstract

*Efficient and accurate fitting of Active Appearance Models (AAM) is a key requirement for many applications. The most efficient fitting algorithm today is Inverse Compositional Image Alignment (ICIA). While this algorithm is extremely fast, it is also known to have a small convergence radius. Convergence is especially bad when training and testing images differ strongly, as in multi-person AAMs. We describe a consistent theory of compositional image alignment and use it to develop two novel fitting methods. The first method Compositional Gradient Descent (CoDe) is approximately four times slower than ICIA, while having a convergence radius which is even larger than that achievable by direct minimisation with a Quasi-Newton method. The second method, LinCoDe, an approximation of CoDe, is as fast as ICIA while having an intermediate convergence radius. The success rate of the novel methods is 10 to 20 times higher than that of the original ICIA method.*

This is the supplementary Material for “On Compositional Image Alignment with an Application to Active Appearance Models”. In addition to more detailed figures and videos, it includes the implementation details for the mapping step and a closer examination of the absolute convergence rate, which had to be left out of the paper because of space limitations.

## 1. Videos

Two **videos** are included, corresponding to the two tracking experiments described in the paper. Both videos are encoded with the *msmpeg4v2* codec, for maximum portability. The codec can be downloaded from <http://www.mediacodec.org/>.

## 2. Convergence Rate

While the success rates shown in figure 3 in the original paper are significantly higher for our methods than for *ICIA*, they might appear to be relatively small in the abso-

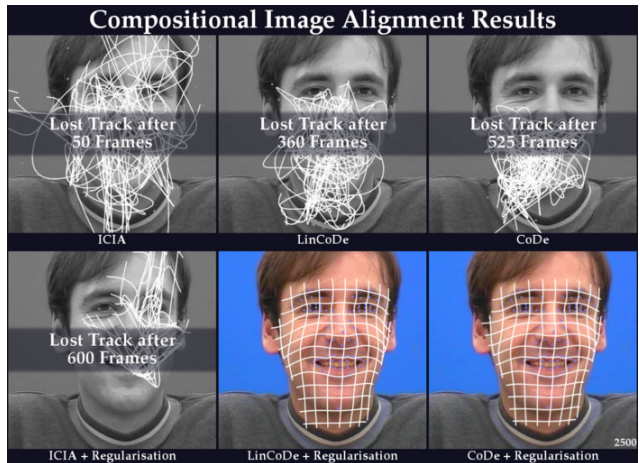


Figure 1. **CoDe can track a full 5000 frames sequence without reinitialisation, LinCoDe** and keeps track for over 2500 frames while *ICIA* – even with the orthonormalized incremental warp – loses track completely after 600 frames and does not reach the accuracy of *CoDe* and *LinCoDe*.

lute measure. Our methods achieve in these experiments between 55% and 68% success rate. This is because of the large starting displacements of up to 20% inter eye distance. For closer starting distances these success rates become even higher. While the tracking experiments show that the achieved success rate is large enough for practical applications, it can be made even higher by random initialisations. To implement an efficient random initialisation scheme it is necessary to detect divergence. We show in figure 3 that it is possible to reliably detect divergence (and try again from an offset starting position) by classifying the results based on the residual. We plot the ratio of detected coverage failures versus the number of successful runs misclassified as convergence failures for varying thresholds on the residual. This shows, that when accepting no misclassifications of converged runs as non-converged it is possible to detect 95% of the failed runs. If we are willing to invest more restarts even from converged positions, than 99% of the failed runs can be detected when misclassifying half of the converged runs.

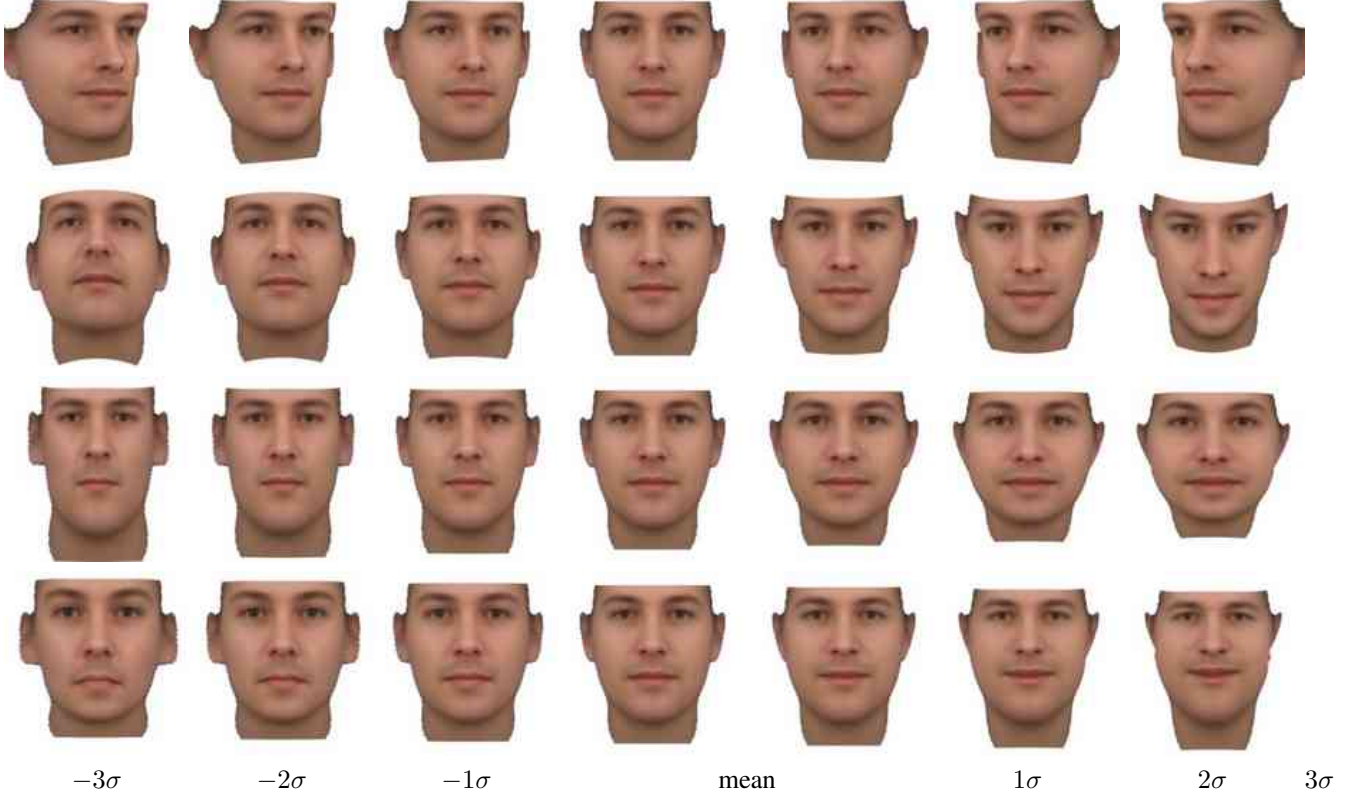


Figure 4. The first shape components

### 3. Model

The principal axis of the model can be explored interactively in the attached webpage. Note that we were able to learn quite a large number of components from the data, because of the large variability in the input data. Images of the shape and appearance variation can be accessed by opening the [index.html](#) file, and clicking on the link *An overview of the active appearance model used in the experiments*. Moving the mouse over the images allows a direct comparison of the effect of each component.

### 4. Mapping and Regularisation

In this section we will describe how to efficiently calculate the mapping  $q = C(q_0, p)$  for the model used in this paper and incorporate regularisation into this step.

Concatenating warps means solving (4.2), i.e. finding the model warp which best describes the concatenated warps. Denote the concatenated warp positions as

$$r^\dagger \triangleq [W(q_0) \circ V(p)](r) \quad (4.1)$$

Expanding the mapping Equation

$$C^\circ(q_0, p) = \arg \min_{q^*} \|W(q^*) - W(q_0) \circ V(p)\|_D^2 \quad (4.2)$$

gives

$$q = \arg \min_{\rho, \tau, \alpha} \|R_\rho(r + M(r)\alpha) + \tau - r^\dagger\|_{r \in \mathcal{D}}^2 \quad (4.3)$$

Performing a rotation and scaling of the residual vector does not change the minimum. Multiply with  $-R_\rho^{-1}$  to get

$$\rho' \triangleq \left[ \frac{1+a}{(1+\rho_1)^2 + \rho_2^2} - 1 \quad \frac{-b}{(1+\rho_1)^2 + \rho_2^2} \right]^T \quad (4.4)$$

$$\tau' \triangleq R_{\rho'} \tau$$

$$q = \arg \min_{\rho', \tau, \alpha} \frac{1}{2} \int_r \|-r - M(r)\alpha - \tau' + R_{\rho'} r^\dagger\|^2$$

which when discretized can be written as

$$q = \arg \min_{\rho', \tau, \alpha} \left\| \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} \rho' \\ \tau' \\ \alpha \end{bmatrix} - \mu \right\|^2 \quad (4.5)$$

$$C \triangleq \begin{bmatrix} r_{1x}^\dagger & r_{1y}^\dagger \\ r_{1y}^\dagger & -r_{1x}^\dagger \\ \vdots & \vdots \\ r_{Nx}^\dagger & r_{Ny}^\dagger \\ r_{Ny}^\dagger & -r_{Nx}^\dagger \end{bmatrix} \quad D \triangleq - \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \end{bmatrix} M \quad \mu \triangleq \begin{bmatrix} r_1 \\ \vdots \\ r_2 \end{bmatrix}$$

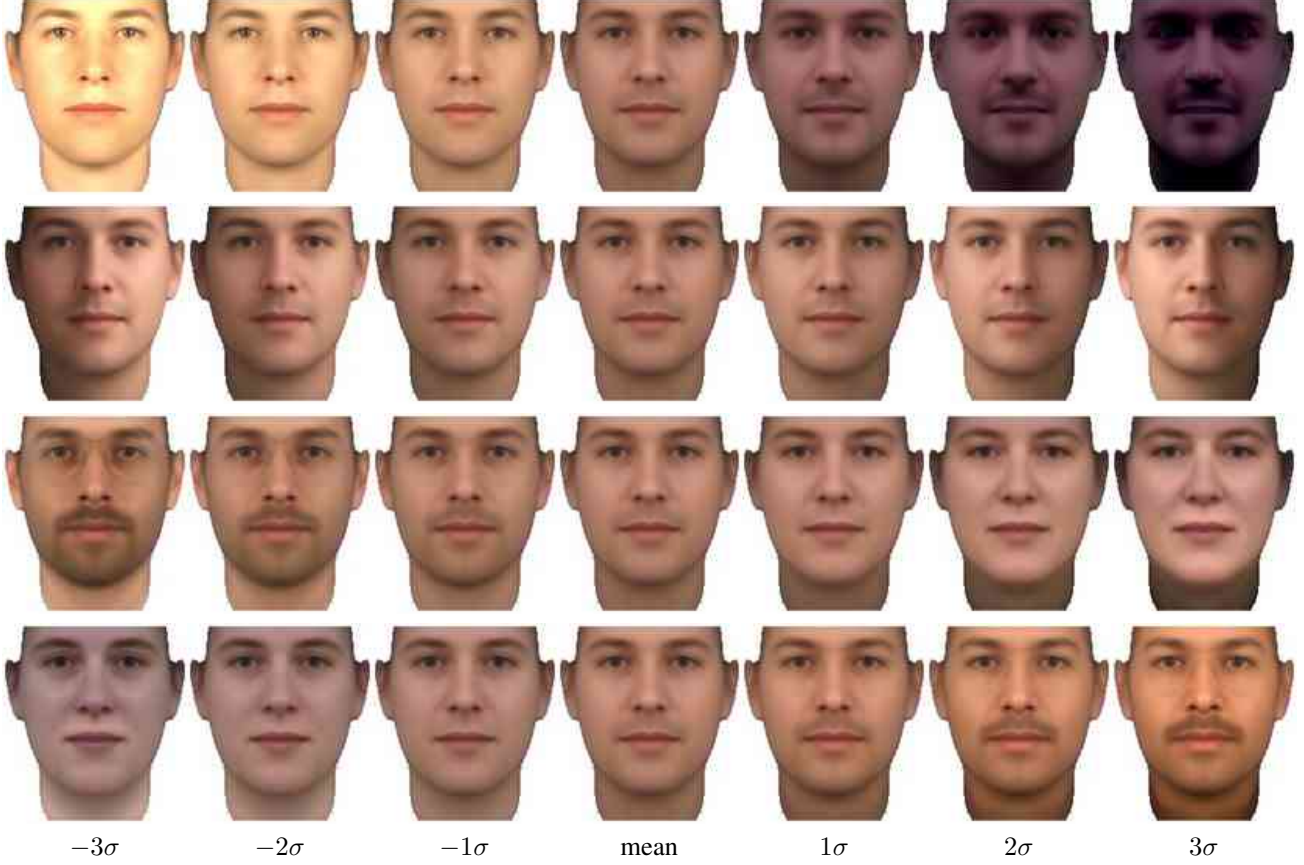


Figure 5. The first appearance components

The above minimisation problem can be solved relatively efficiently in  $O(N_{\text{Shape Parameter}}^3 + N_{\text{Shape Parameter}} N_{\text{Pixel}})$ , which for a typical AAM with  $N_{\text{Pixel}} > N_{\text{Shape Parameter}}^2$  (e.g.  $200^2$  pixel and 100 shape parameter) is dominated by  $O(N_{\text{Shape Parameter}} N_{\text{Pixel}})$ . This can be done as only  $C$  changes between iterations. The minimum is attained for

$$\begin{pmatrix} \rho' \\ \alpha \end{pmatrix} = \begin{bmatrix} C^T C & C^T D \\ D^T C & D^T D \end{bmatrix}^{-1} \begin{bmatrix} C^T \mu \\ D^T \mu \end{bmatrix} \quad (4.6)$$

It is possible to precalculate  $D^T D$  and  $D^T \mu$ , leaving the multiplications of  $C^T C$ ,  $D^T C$  and  $C^T \mu$  and solving a  $N_{\text{Shape Parameter}}^2$  linear system for each iteration.

**Regularisation** The shape coefficients are independently normal distributed, if we assume that the original distribution of face shapes is normally distributed. Knowledge of the probability density of face shape makes it possible to use a maximum likelihood estimation for the concatenated warp shapes. This corresponds to a regularisation of the shape coefficients, which can be introduced at this point without incurring additional cost. The regularisation parameter  $\lambda$

should depend on the noise characteristic of the test images. With  $\Lambda \triangleq \begin{bmatrix} 0 & \lambda I \end{bmatrix}$  we can add regularisation to (4.5) as

$$q_{t+1} \approx \arg \min_{\rho, \tau, \alpha} \left\| \begin{bmatrix} C & D \\ 0 & \Lambda \end{bmatrix} \begin{bmatrix} \rho' \\ \tau' \\ \alpha \end{bmatrix} - \begin{bmatrix} \mu \\ 0 \end{bmatrix} \right\|^2 \quad (4.7)$$

For which the minimum is attained at

$$\begin{pmatrix} \rho' \\ \alpha \end{pmatrix} = \begin{bmatrix} C^T C & C^T D \\ D^T C & D^T D + \Lambda^T \Lambda \end{bmatrix}^{-1} \begin{bmatrix} C^T \mu \\ D^T \mu \end{bmatrix} \quad (4.8)$$

## 5. Discretisation

The domain is discretized into pixel, such that the boundary is composed of segments of length one with either horizontal or vertical direction and the area is sampled at the pixel centers, where each pixel center accounts for an area of one pixel. We choose the mapping  $D$  such that it picks the values of the closest pixel for each border sample.

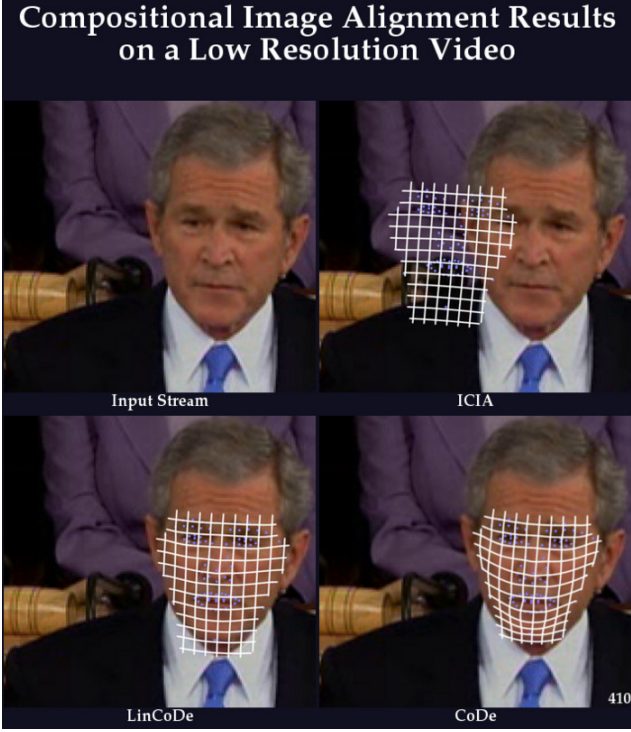


Figure 2. Tracking a low resolution video shows that *ICIA* does not generalize well, while *CoDe* tracks successfully.

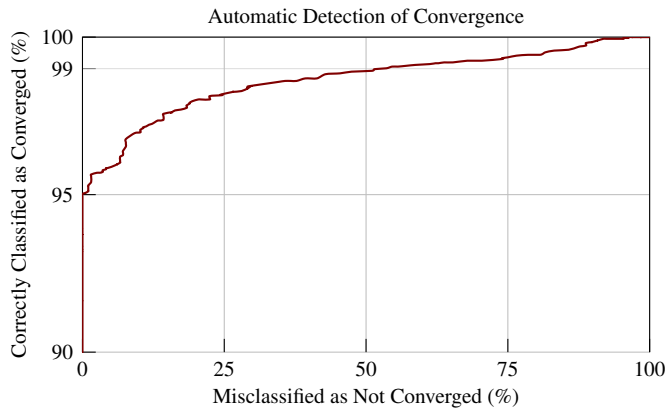


Figure 3. Convergence can be detected reliably, allowing to increase the success rate arbitrarily by random reinitializations. We plot the number of misclassifications as a function of a threshold on the residual, for the *CoDe* method with regularisation.

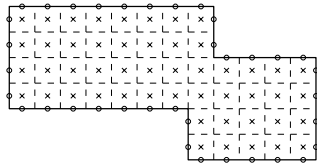


Figure 6. The parametrisation used. Squares are pixel, crosses pixel centers where the function is evaluated and circles are boundary points where the boundary is sampled. The value of  $a - I_q$  at the boundary is set to the value of the corresponding pixel.