On Compositional Image Alignment, with an Application to Active Appearance Models

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* ICIA *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 "forward" compositional image alignment in a consistent framework which also incorporates methods previously termed "inverse" compositional, 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 Quasi-Newton descent. An intermediate convergence range with the same speed as* ICIA *is achieved by* LinCoDe*, the second new method. The success rate of the novel methods is 10 to 20 times higher than that of the original* ICIA *method.*

Active Appearance Models [6, 9] (AAM) are generative 2D models, which describe linearly the variation of shape and appearance of a set of textured objects. Shape is expressed by warps of a reference shape (the "model warp"), possibly accompanied by a shape prior. Texture is expressed as a linear model in the reference frame. AAMs are very popular and successful models because of their expressive power, coupled with the availability of efficient fitting algorithms. The challenge in developing fitting algorithms is to find a good tradeoff between runtime and performance, where performance may be characterised in terms of "capture range" — the range of starting points, relative to the optimal fit, from which the algorithm converges.

Fitting algorithms for AAMs are of two kinds: analysis-by-synthesis and regression-based. In analysis-by-synthesis, a generative model is fitted to data by iteratively minimizing the residual between the synthesized and the observed image [2, 9, 3]. Regression-based methods use a learned mapping from the residual to parameter updates. The regression may be linear, as in the seminal work of Cootes et al. [6], or nonlinear [13, 8, 16, 15].

In this paper, we revisit the so-called *Compositional* approach to analysis-by-synthesis. Compositional image alignment, which originates from [7], seeks to reduce the residual by successively applying (composing) *incremental warps*. The use of incremental warps lends itself well to efficient approximations of the gradient and Hessian of the objective function. In some cases the Hessian is approximated as constant [7, 4, 14], so that it can be precomputed, which in turn leads to particularly efficient fitting algorithms. The state of the art is often considered to be the Inverse Compositional Image Alignment (*ICIA*) method of Matthews and Baker [9, 1, 2].

First we develop a unified framework for compositional fitting algorithms, and classify them in terms of four factors: the choice of optimization algorithm; the representation of incremental warp; the approximation of the gradient; and the approximation of the Hessian, where required by the algorithm. For example, we show that *ICIA* can be expressed as a Gauss-Newton algorithm in which the space of incremental warps is the same as the model warp space, the gradient is approximate, and the Hessian is approximated as constant. The performance of such an algorithm has limits, imposed by the validity of the approximations of the gradient and Hessian. We show that for *ICIA*, the approximations are valid only when model matches the data closely. That limits the attainable capture range. This is especially the case when test data is not closely matched by training data as, for instance, in multi-person face-fitting. Therefore this paper develops alternative methods, in the framework, that improve the runtime-performance tradeoff.

Our two new algorithms, *CoDe* (Compositional Descent) and *LinCoDe* (Linearised Compositional descent), are both 1$^{\text{st}}$ order algorithms — i.e. gradient descent algorithms. It might appear surprising at first sight that gradient descent algorithms should outperform Gauss-Newton. There are several reasons for this. First Gauss-Newton is powerful, in principle, because of the availability of additional information in the Hessian, but in practice only a severely approximated Hessian can be computed. Secondly, In *ICIA* only an approximate gradient is used, whereas in one of the new algorithms, *CoDe*, the exact gradient is computed. Lastly, the choice of incremental warp for *CoDe* and *LinCoDe* is crucial. We express the incremental warps in an orthonormal basis which makes for well-conditioned optimization and removes the need for 2nd order methods.

The performance of the new algorithms, relative to *ICIA* as our benchmark, is tested thoroughly over a substantial multi-person face database, with further testing of tracked facial motion on over 5000 frames of movie data. The results on these challenging datasets confirms the predictions suggested by our alignment framework. Indeed *ICIA* has a very limited capture range. This is alleviated if *ICIA* is altered to use orthonormal incremental warps, and more so if regularisation of warps in model space is also applied. The new algorithms, *CoDe* and *LinCoDe*, used with regularisation, provide the best runtime-performance tradeoff, with *CoDe* able to sustain continued tracking for around 100 seconds of a movie with low resolution and demanding head-motions, and 5000 frames for a higher quality dataset.

1. Active Appearance Models

AAMs are generative models consisting of separate shape and appearance models. They are fitted to images

$$I(\boldsymbol{r}') \in \mathbb{R}, \qquad \boldsymbol{r}' \in \mathcal{I} \subset \mathbb{R}^2 \quad , \tag{1.1}$$

treated as continuous functions of the image domain $\mathcal{I}$. In this paper we use only linear appearance models

$$\mathit{\Lambda}(\boldsymbol{r};\boldsymbol{\beta}) \triangleq a(\boldsymbol{r}) + A(\boldsymbol{r})\boldsymbol{\beta}, \qquad \boldsymbol{r} \in \mathcal{D} \subset \mathbb{R}^2 \tag{1.2}$$

parametrized by the coefficient vector $\boldsymbol{\beta}$ and defined over the texture domain $\mathcal{D}$. It is warped into the image by a model-warp

$$W(\boldsymbol{r};\boldsymbol{q}) = \boldsymbol{r}' \quad , \tag{1.3}$$

which is parametrized by the shape parameter vector $\boldsymbol{q}$. The model warp used in this paper is a linear shape model $\boldsymbol{r} + M(\boldsymbol{r})\boldsymbol{\alpha}$ concatenated with a similarity transform. The warp parameters are $\boldsymbol{q} = (\boldsymbol{\rho}, \boldsymbol{\tau}, \boldsymbol{\alpha})$, with a global rotation $\boldsymbol{\rho}$, a translation $\boldsymbol{\tau}$ and local deformation coefficients $\boldsymbol{\alpha}$.

$$\begin{aligned} W(\boldsymbol{r};\boldsymbol{q}) &\triangleq \boldsymbol{R}_{\boldsymbol{\rho}}(\boldsymbol{r} + M(\boldsymbol{r})\boldsymbol{\alpha}) + \boldsymbol{\tau} \\ M(\boldsymbol{r}) &\in \mathbb{R}^{2\times N_{\text{Shape Parameter}}} \\ \boldsymbol{R}_{\boldsymbol{\rho}} &\triangleq \begin{bmatrix} 1+\boldsymbol{\rho}_1 & \boldsymbol{\rho}_2 \\ -\boldsymbol{\rho}_2 & 1+\boldsymbol{\rho}_1 \end{bmatrix} \end{aligned} \tag{1.4}$$

Note that the compositional image aligment method presented in this paper can be used equally with other model warps and with higher dimensional images, such as space or space-time volumes.

2. Objective Function

The objective function used typically [6, 9, 3] for AAM fitting is the squared difference between the target image I warped back into texture space and the model appearance $\mathit{\Lambda}(\boldsymbol{\beta})$, given as

$$\begin{aligned} F(\boldsymbol{q},\boldsymbol{\beta}) &\triangleq \|f(\boldsymbol{q},\boldsymbol{\beta})\|^2_{\mathcal{D}}\,, \\ \text{with } f(\boldsymbol{q},\boldsymbol{\beta}) &\triangleq \mathit{\Lambda}(\boldsymbol{\beta}) - I \circ W(\boldsymbol{q}) \end{aligned} \tag{2.1}$$

Where $\|\cdot\|^2_{\mathcal{D}} = \langle\cdot,\cdot\rangle_{\mathcal{D}}$ denotes the integral over the squared residual in $\mathcal{D}$ and $[I \circ W(\boldsymbol{q})](\boldsymbol{r}) = I(W(\boldsymbol{r};\boldsymbol{q}))$ is the function composition operator. We call f the residual function.

Appearance variation can be handled [9] by evaluating the objective function always at the optimal appearance parameters $\hat{\boldsymbol{\beta}}(\boldsymbol{q})$. For models with linear appearance $\mathit{\Lambda}(\boldsymbol{r};\boldsymbol{\beta}) = a(\boldsymbol{r}) + A(\boldsymbol{r})\boldsymbol{\beta}$, with an orthonormal basis A, the optimal parameters are

$$\hat{\boldsymbol{\beta}}(\boldsymbol{q}) = -A^T(a - I \circ W(\boldsymbol{q})) \quad . \tag{2.2}$$

Then, using the fact that A^TA is the identity, gives:

$$\begin{aligned} f(\boldsymbol{q},\hat{\boldsymbol{\beta}}(\boldsymbol{q})) &= P(a - I \circ W(\boldsymbol{q})) \\ \text{with } P &\triangleq E - AA^T \quad , \end{aligned} \tag{2.3}$$

where E is the identity. Now the appearance coefficients can be left out, and the cost becomes

$$F(\boldsymbol{q}) \triangleq F(\boldsymbol{q},\hat{\boldsymbol{\beta}}(\boldsymbol{q})) \text{ and } f(\boldsymbol{q}) \triangleq f(\boldsymbol{q},\hat{\boldsymbol{\beta}}(\boldsymbol{q})) \quad . \tag{2.4}$$

— known as the *project out norm* [9].

3. Image Alignment

This section introduces compositional optimisation, which is a class of iterative optimisation methods suitable for objective functions of the form of (2.1). Typical iterative optimisation methods use first or second order Taylor expansions to approximate the objective function. Compositional optimisation methods substitute this with a two step approximation. The objective function is first approximated by the introduction of an *incremental warp* followed by Taylor expansion. By the end of this section we will have defined five alternative optimisation methods (listed in figure 1) including ICIA and the two new methods.

Iterative optimisation methods approximate a (real valued) objective function $F(\boldsymbol{q})$ around a current estimate $\boldsymbol{q}_0$ with a simpler objective function $\tilde{F}(\boldsymbol{q}_0, \mathit{\Delta}\boldsymbol{q})$. Note that $\boldsymbol{q}$ and $\mathit{\Delta}\boldsymbol{q}$ are not necessarily from the same space. Then $\mathit{\Delta}\boldsymbol{q}$ is determined to decrease the approximate cost and is used to update $\boldsymbol{q}_0$, and the process is iterated. To update the hypothesis $\boldsymbol{q}_0$, a mapping

$$\boldsymbol{q} = C(\boldsymbol{q}_0, \mathit{\Delta}\boldsymbol{q}) \tag{3.1}$$

is needed from the parameter space of the approximate function $\tilde{F}$ to the parameter space of F. As the expansion of F happens at $\boldsymbol{q}_0$, the mapping function needs to fulfill

$$C(\boldsymbol{q}_0, \boldsymbol{0}) = \boldsymbol{q}_0 \quad . \tag{3.2}$$

An objective function F is well approximated by $\tilde{F}$ if

$$F(C(\boldsymbol{q}_0, \mathit{\Delta}\boldsymbol{q})) \approx \tilde{F}(\boldsymbol{q}_0, \mathit{\Delta}\boldsymbol{q}) \quad . \tag{3.3}$$

3.1. Additive Descent

Before defining compositional optimisation, we first define the classic gradient descent and Gauss-Newton optimisation methods for quadratic objective functions like (2.4). Both methods use the Jacobian of the residual function, the matrix of partial derivatives:

$$J_{\boldsymbol{q}_0} \triangleq \nabla_{\boldsymbol{q}} f(\boldsymbol{q}_0) \quad . \tag{3.4}$$

Gradient descent approximates the objective function linearly:

$$F(C^+(\boldsymbol{q}_0, \Delta\boldsymbol{q})) \approx F(\boldsymbol{q}_0) + \nabla_{\boldsymbol{q}} F(\boldsymbol{q}_0)\Delta\boldsymbol{q} \quad , \tag{3.5}$$

and uses the simple additive mapping function

$$C^+(\boldsymbol{q}_0, \Delta\boldsymbol{q}) = \boldsymbol{q}_0 + \Delta\boldsymbol{q} \quad , \tag{3.6}$$

hence the term "additive descent". Descending the gradient $\nabla_{\boldsymbol{q}} F(\boldsymbol{q}_0) = J_{\boldsymbol{q}_0} f(\boldsymbol{q}_0)$, the current estimate is updated as

$$\boldsymbol{q}_0 \leftarrow C^+(\boldsymbol{q}_0, -\kappa J_{\boldsymbol{q}_0} f(\boldsymbol{q}_0)) = \boldsymbol{q}_0 - \kappa J_{\boldsymbol{q}_0} f(\boldsymbol{q}_0) \quad , \tag{3.7}$$

using a stepsize parameter κ.

Gauss-Newton approximates the square error function $F(\boldsymbol{q}) = \|f(\boldsymbol{q})\|^2_{\mathcal{D}}$ by linear expansion of the residual:

$$F(C^+(\boldsymbol{q}_0, \Delta\boldsymbol{q})) \approx \left\| f(\boldsymbol{q}_0) + J_{\boldsymbol{q}_0}\Delta\boldsymbol{q} \right\|^2_{\mathcal{D}}$$

which has a minimum $\Delta\boldsymbol{q}^*$ at

$$\Delta\boldsymbol{q}^* = -(J^T_{\boldsymbol{q}_0} J_{\boldsymbol{q}_0})^{-1} J^T_{\boldsymbol{q}_0} f(\boldsymbol{q}_0) \quad , \tag{3.8}$$

so that estimates are updated as

$$\boldsymbol{q}_0 \leftarrow C^+(\boldsymbol{q}_0, \Delta\boldsymbol{q}^*) = \boldsymbol{q}_0 - (J^T_{\boldsymbol{q}_0} J_{\boldsymbol{q}_0})^{-1} J^T_{\boldsymbol{q}_0} f(\boldsymbol{q}_0) \quad . \tag{3.9}$$

3.2. Compositional Descent

Additive descent, in the form of gradient descent or Gauss-Newton, approximate the objective function with a linear or a quadratic function respectively. Alternatively, the special form of the project-out cost (2.4) allows a nonlinear approximation in which the warp $W(\boldsymbol{q})$ is *composed* with an *incremental warp* $V(\boldsymbol{p})$ [9, 12]. The resulting nonlinear "compositional" approximation, in combination with further approximations (see below), is known to give efficient fitting [9]. Compositional approximation can be used either with gradient descent or with Gauss Newton. We investigate which optimisation method, together with which compositional scheme, gives the best performance with AAMs.

The compositional mapping function for the compositional approximation is defined as

$$C^\circ(\boldsymbol{q}_0, \boldsymbol{p}) = \arg\min_{\boldsymbol{q}^*} \|W(\boldsymbol{q}^*) - W(\boldsymbol{q}_0) \circ V(\boldsymbol{p})\|^2_{\mathcal{D}} \quad . \tag{3.10}$$

The calculation of C° is more involved than C^+, and is explained in detail in section B. The compositional approximation is a *two step* approximation of the objective function (2.1). First, as in *ICIA* [1], the objective function is approximated compositionally as

$$F(C^\circ(\boldsymbol{q}_0, \boldsymbol{p})) \approx \tilde{F}(\boldsymbol{q}_0, \boldsymbol{p}) \triangleq \left\| \tilde{f}(\boldsymbol{q}_0, \boldsymbol{p}) \right\|^2_{\mathcal{D}} \tag{3.11}$$

$$\text{with } \tilde{f}(\boldsymbol{q}_0, \boldsymbol{p}) \triangleq P(a - I \circ W(\boldsymbol{q}_0) \circ V(\boldsymbol{p})) \quad .$$

and then $\tilde{F}(\boldsymbol{q}_0, \boldsymbol{p})$ is approximated by Taylor expansion with respect to the incremental warp parameters $\boldsymbol{p}$. This requires the following Jacobian matrix *with respect to* $\boldsymbol{p}$:

$$\begin{aligned}\tilde{J}_{\boldsymbol{q}_0} &\triangleq \nabla_{\boldsymbol{p}} \tilde{f}(\boldsymbol{q}_0, \mathbf{0}) \\ &= -P\nabla(I \circ W(\boldsymbol{q}_0))\nabla_{\boldsymbol{p}} V(\mathbf{0}) \quad .\end{aligned} \tag{3.12}$$

Compositional Gradient Descent

Linear expansion of (3.11) with respect to the incremental warp parameters at $\boldsymbol{p} = \mathbf{0}$ gives:

$$F(C^\circ(\boldsymbol{q}_0, C^+(\mathbf{0}, \Delta\boldsymbol{p}))) \approx \tilde{F}(\boldsymbol{q}_0, \mathbf{0}) + \nabla_{\boldsymbol{p}} \tilde{F}(\boldsymbol{q}_0, \mathbf{0})\Delta\boldsymbol{p} \quad . \tag{3.13}$$

and using $\tilde{F}(\boldsymbol{q}_0, \mathbf{0}) = F(\boldsymbol{q}_0)$ and $\tilde{f}(\boldsymbol{q}_0, \mathbf{0}) = f(\boldsymbol{q}_0)$ gives:

$$F(C^\circ(\boldsymbol{q}_0, \Delta\boldsymbol{p})) \approx F(\boldsymbol{q}_0) + f(\boldsymbol{q}_0)^T \tilde{J}_{\boldsymbol{q}_0}\Delta\boldsymbol{p} \quad . \tag{3.14}$$

This results in the update step

$$\boldsymbol{q}_0 \leftarrow C^\circ(\boldsymbol{q}_0, -\kappa\tilde{J}^T_{\boldsymbol{q}_0} f(\boldsymbol{q}_0)) \quad . \tag{3.15}$$

Compositional Gauss-Newton

In analogy to additive Gauss-Newton we calculate the quadratic approximation of the approximated cost $\tilde{F}$

$$F(C^\circ(\boldsymbol{q}_0, C^+(\mathbf{0}, \Delta\boldsymbol{p}))) \approx \left\| f(\boldsymbol{q}_0) + \tilde{J}_{\boldsymbol{q}_0}\Delta\boldsymbol{p} \right\|^2_{\mathcal{D}} \tag{3.16}$$

with the composed hypothesis update

$$\boldsymbol{q}_0 \leftarrow C^\circ(\boldsymbol{q}_0, -(\tilde{J}^T_{\boldsymbol{q}_0}\tilde{J}_{\boldsymbol{q}_0})^{-1}\tilde{J}^T_{\boldsymbol{q}_0} f(\boldsymbol{q}_0)) \quad . \tag{3.17}$$

The Jacobian $\tilde{J}_{\boldsymbol{q}_0}$ can be evaluated cheaply by calculating $\nabla(I \circ W(\boldsymbol{q}))$ with finite differences, using the backwarped image as in (3.12).

It would be even faster, if the Jacobian were approximated as a constant [7, 9]. Once we have the constant approximation, all five algorithms listed in figure 1 can be explained.

3.3. Constant Jacobian and constant Hessian Approximation

The Jacobian from (3.12) is not constant, as it depends on the backwarped image $I \circ W(\boldsymbol{q})$. Instead of evaluating the correct Jacobian, a constant approximation [7] can be used. To approximate the Jacobian by a constant, the approximate value of the Jacobian *at the optimum* of the objective function is used. Of course, the optimum is unknown but assuming the residual at the optimum is small then, using the objective function (2.1),

$$a + A\hat{\boldsymbol{\beta}}(\boldsymbol{q}) \approx I \circ W(\boldsymbol{q}) \quad . \tag{3.18}$$

and, from (3.12), the resulting Jacobian is

$$\tilde{J}_{\boldsymbol{q}_0} \approx P\nabla(a + A\hat{\boldsymbol{\beta}}(\boldsymbol{q}_0))\nabla_{\boldsymbol{p}} V(\boldsymbol{0}) \quad . \tag{3.19}$$

The Jacobian still depends on $\boldsymbol{q}_0$ through the appearance parameter $\hat{\boldsymbol{\beta}}(\boldsymbol{q}_0)$. With the same argument used in (2.4) to make the objective function independent of appearance, we can now write the approximated constant Jacobian as

$$\bar{J} = P\nabla a \nabla_{\boldsymbol{p}} V(\boldsymbol{0}) \quad . \tag{3.20}$$

The approximate Jacobian can be used for two purposes: first, to approximate the Hessian as the constant $\bar{J}^T\bar{J}$; and second to approximate the gradient as the linear function $\bar{J}^T e(\boldsymbol{q}_0)$.

3.4. The Five Compositional Algorithms

All combinations of compositional gradient descent and compositional Gauss-Newton with the gradient and Hessian approximations give five optimisation strategies:

1. Compositional gradient descent with
   (a) the true gradient $\tilde{J}^T_{\boldsymbol{q}_0} f(\boldsymbol{q}_0)$ which we call *CoDe*
   (b) the approximate gradient $\bar{J}^T f(\boldsymbol{q}_0)$, which we call *LinCoDe*
2. Compositional Gauss-Newton with
   (a) the true Gauss-Newton Hessian $\tilde{J}^T_{\boldsymbol{q}_0}\tilde{J}_{\boldsymbol{q}_0}$ and true gradient $\tilde{J}^T_{\boldsymbol{q}_0} f(\boldsymbol{q}_0)$, which we call *CoNe* [2].
   (b) approximate Hessian $\bar{J}^T\bar{J}$ and true gradient $\tilde{J}^T_{\boldsymbol{q}_0} f(\boldsymbol{q}_0)$, which is identical to the method of [4], and which we call *CoLiNe*
   (c) approximate Hessian $\bar{J}^T\bar{J}$ and approximate gradient $\bar{J}^T f(\boldsymbol{q}_0)$, which is identical to *ICIA* [9].

Our derivation of *ICIA* gives the same Hessian and Jacobian as in the original formulation [9], starting from the same cost function (2.1). Both derivations arrive at the same update formula and are therefore equivalent — differing only cosmetically, but actually representing the identical algorithm. (In the original, there is an *inverse* incremental warp introduced by a change of variables, which does not appear in our version.) We prove in the appendix that both derivations are equivalent. *ICIA* is exactly the approximate compositional image alignment method which approximates the gradient and the Hessian by exchanging the backwarped image with the model appearance. In [9] the Jacobian is treated as constant, but we have shown that this is only a good approximation when the residual is small, – i.e. when the model matches the data well and is initalized close to the solution. This explains why *ICIA* should have a relatively small capture range for convergence. This will be especially true for multi-identity AAMs, where the model may not match the data closely, and whenever the camera and light setup of the test set differ from that of the training set. So we expect *ICIA* to *generalize* badly to unseen examples.

We have now introduced the five compositional image alignment methods. They would be expected to differ in speed and capture range.

Speed The speed of computation depends on the calculation of the derivatives. It is expensive to calculate the correct Hessian, and it is moderately expensive to calculate the correct Jacobian. *ICIA* and *LinCoDe* compute neither of these quantities, so they are the fastest algorithms. Nearly as fast are *CoLiNe* and *CoDe*, which calculate the correct Jacobian for the gradient, but do not calculate the Hessian. The slowest method is *CoNe*, which has to calculate the true Hessian.

Capture range When using the incremental model warp as proposed in [7, 9], second order methods are necessary as the parameter space is badly scaled, and so has to be normalized using the Hessian. In that case *CoNe* (exact Hessian and gradient) is expected to be the best method, followed by *CoLiNe* and lastly *ICIA* (approximate Hessian and gradient). In the next section we introduce a well conditioned incremental warp, leading to a gradient descent method that performs as well as the slow, exact, 2$^{\text{nd}}$ order *CoNe* method, while being as fast as *ICIA*.

4. Incremental Warp

The incremental warp $V(\boldsymbol{p})$ should have the following two properties. First, it must be flexible enough to reach the (unknown) global optimum, it must be possible to go from any current hypothesis $\boldsymbol{q}_0$ to any other hypothesis $\boldsymbol{q}$ via a sequence of incremental warps and projections into the model:

$$\boldsymbol{q} = C(\ldots C(C(\boldsymbol{q}_0, \boldsymbol{p}_0), \boldsymbol{p}_1)\ldots, \boldsymbol{p}_k) \quad . \tag{4.1}$$

Second, it should not be so flexible that the composite warp $W(\boldsymbol{q}_0) \circ V(\boldsymbol{p})$ can stray far outside the model warp space, as this would lead to a bad approximation.

Approximating the incremental warp We have introduced two approximations of the original objective function: (1) approximation by the introduction of an incremental warp (3.11), and (2) approximation by Taylor expansion (3.13 and 3.16). To make the first approximation exact, it is

Method	Hessian		Gradient		Speed	Capture Range
CoDe (this paper)	Not used		True:	$\tilde{J}^T_{\boldsymbol{q}_0} f(\boldsymbol{q}_0)$	Fast	Large
LinCoDe (this paper)	Not used		Linear Approx:	$\bar{J}^T f(\boldsymbol{q}_0)$	Very Fast	Medium
CoLiNe [4]	Constant Approx.:	$\bar{J}^T \bar{J}$	True:	$\tilde{J}^T_{\boldsymbol{q}_0} f(\boldsymbol{q}_0)$	Fast	Medium
ICIA [9]	Constant Approx.:	$\bar{J}^T \bar{J}$	Linear Approx:	$\bar{J}^T f(\boldsymbol{q}_0)$	Very Fast	Small
CoNe [2]	Gauss-Newton Approx.:	$\tilde{J}^T_{\boldsymbol{q}_0} \tilde{J}_{\boldsymbol{q}_0}$	True:	$\tilde{J}^T_{\boldsymbol{q}_0} f(\boldsymbol{q}_0)$	Slow	Large

Figure 1. **We introduce two novel objective functions which have a larger capture range than previous methods** The newly introduced compositional optimisation methods are Hessian-Free gradient descent methods.

necessary to have $W(\boldsymbol{q}_0) \circ V(\boldsymbol{p}) = W(\boldsymbol{q})$ for a suitable $\boldsymbol{q}$. This is fulfilled by

$$V(\boldsymbol{p}) = W^{-1}(\boldsymbol{q}_0) \circ W(\boldsymbol{q}) \quad , \tag{4.2}$$

with $\boldsymbol{p} = (\boldsymbol{q}_0, \boldsymbol{q})$. This incremental warp can be used efficiently if the model warps forms a group, because then V and W share the same space. If the model warps do not form a group then it is necessary to use an approximation, as (4.2) is not efficiently tractable. This was solved in [9] by assuming that (4.2) can be approximated by $V(\boldsymbol{p}) = W^{-1}(\boldsymbol{p})$. And, as the incremental warp is evaluated at the identity and under the assumption of small $\boldsymbol{p}$, this is further approximated by

$$V^{\text{Model}}(\boldsymbol{p}) = W(-\boldsymbol{p}) \quad . \tag{4.3}$$

An alternative, more costly, solution [12] is to precalculate (4.2) over a tesselation in $\boldsymbol{q}_0$-space, and choose the closest warp from that precalculated set.

Linearized warp approximation An alternative approach to warp approximation derives from Taylor expansion of the compositional objective function. To derive the update rule the residual is linearized, and the Jacobian from (3.12) is used. The linearization applies the image derivative to a linearization of the incremental warp, at the identity warp. Accordingly, to have a good approximation of the original objective function by the linearized compositional objective function it is necessary that *the incremental warp is well approximated by its linearization*. This is not the case for the (nonlinear) incremental warps of [9, 12], but is – as our results demonstrate – essential for a good alignment algorithm.

We propose to use a linearization of $W(\boldsymbol{r}, \boldsymbol{q})$ at $\boldsymbol{q} = 0$:

$$V^{\text{Ortho}}(\boldsymbol{r}; \boldsymbol{p}) = \boldsymbol{r} + \nabla_{\boldsymbol{q}} W(\boldsymbol{r}; \boldsymbol{0})\boldsymbol{p} \tag{4.4}$$

which for the shape model used in this paper is

$$V^{\text{Ortho}}(\boldsymbol{r}; \boldsymbol{p}) = \boldsymbol{r} + L(\boldsymbol{r})\boldsymbol{p} \tag{4.5}$$

$$L(\boldsymbol{r}) \triangleq \begin{bmatrix} \boldsymbol{r}_x & \boldsymbol{r}_y & 1 & 0 & \\ -\boldsymbol{r}_y & \boldsymbol{r}_x & 0 & 1 & \end{bmatrix} M(\boldsymbol{r})$$

Linear incremental warps have another important advantage apart from providing a better approximation to the objective function. The warp basis L can be orthogonalized and normalized, such that the parameters of the incremental warp are independent and equally scaled. This gives the *Orthonormal* warp denoted V^{Ortho}. This improves the accuracy and robustness of all methods and makes it possible to use the compositional gradient descent methods (*CoDe* and *LinCoDe*), as the Hessian is no longer needed to correct the parameter scaling. Orthonormal incremental warps result in the largest capture range and highest speed, and are used in all experiments in this paper. In addition we test ICIA also with the original ICIA warp, the model warp which we denoted V^{Model}.

```
for Blur and associated regularisation values do
    Set $\boldsymbol{q}, \boldsymbol{q}_{\text{best}}$ to the initial guess and initialize $\kappa$
    repeat
        Calculate $\nabla_{\boldsymbol{p}} \tilde{F}(\boldsymbol{q}, \boldsymbol{0})$, $F(\boldsymbol{q})$
        if $F(\boldsymbol{q}) < F(\boldsymbol{q}_{best})$ then
            $\boldsymbol{q}_{\text{best}} \leftarrow \boldsymbol{q}$
            Increase $\kappa$
        else
            if stepsize smaller than threshold then
                return
            else
                decrease $\kappa$
        Calculate $V$ from $\nabla_{\boldsymbol{p}} \tilde{F}(\boldsymbol{q}_{best}, \boldsymbol{p})$ and $\kappa$
        Update $W$ to the concatenation of $W$ and $V$
    until converged
```

Figure 2. **Compositional Image Alignment**

5. Implementation

The algorithms compared in this paper all have the structure detailed in Figure 2. They differ in three respects. First, different *update rules* are used i.e. gradient descent or Gauss-Newton and the various Jacobian and Hessian approximations. Secondly different incremental warps can be used. Thirdly, we compare optimisation performance with and without regularisation. To this end we propose to regularize in the mapping step, in which the concatenated warps are projected back into model space. Regularisation is possible, as a prior on the deformation space is learned in the training phase, such that a maximum likelihood estimation

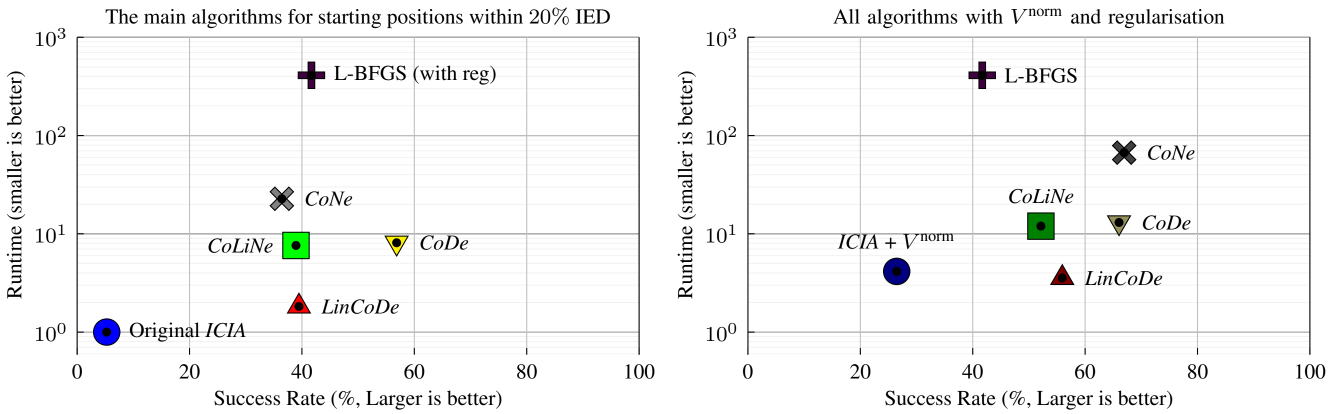

Figure 3. **Best speed–performance tradeoffs come from the two new algorithms *CoDe* and *LinCoDe*.** Left: Without regularisation. Right: with regularisation and V^{Ortho}. Note that *ICIA* is practically useless on this difficult multi-person dataset with a success rate near zero (left). It can be improved (right) by using the orthonormal incremental warp and regularisation. The *CoDe* algorithm with regularisation (right) is as accurate as the slow, approximation-free, compositional Gauss-Newton *CoNe* method but is seven times more efficient.

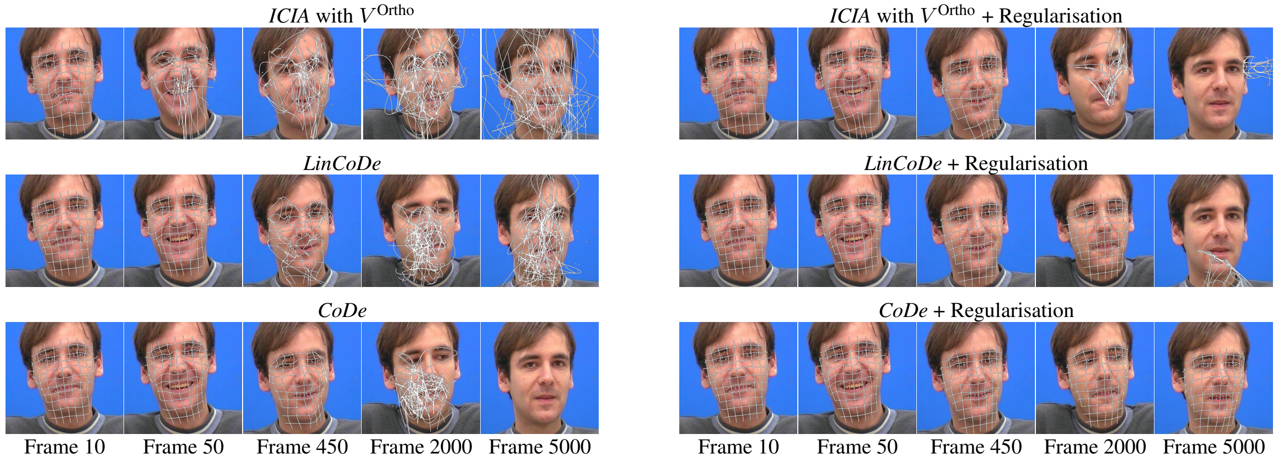

Figure 4. **Our algorithm makes fast and robust tracking possible.** We compare face tracking under natural motion, using *ICIA*, *LinCoDe* and *CoDe*. The original *ICIA* fails immediately with this large model and new face data. Substituting the orthonormal incremental warp for the original *ICIA* warp, the algorithm still loses track very early, whereas *LinCoDe* and *CoDe* can track much further. Finally, adding regularisation to all algorithms, *ICIA* still loses track completely after approximately 500 frames and does not recover the local deformations accurately. In contrast *CoDe* now tracks the full 5000 frame sequence without reinitialization, and *LinCoDe* tracks for 2500 frames.

of the shape can be determined. The details of the mapping step are described in Appendix B.

6. Experiments

We have described five compositional image alignment algorithms, in a common framework, including ICIA and two new algorithms, *CoDe* and *LinCoDe*. The algorithms are evaluated on a relatively large multi-person AAM, on images of unseen identities. This is the most difficult, but also the most typical, situation for face analysis.

We will show now, that the proposed *CoDe* algorithm has the largest capture range achieveable by any of the iterative optimisation methods, while being only eight times slower than *ICIA*. The differences in capture range are most pronounced when fitting a test face outside the training set. Then *ICIA* fails to converge for most starting positions, while *CoDe* converges much more reliably. We will also show that the approximate gradient descent algorithm *LinCoDe*, which requires similar runtime to *ICIA*, converges as reliable as a slower approximate second order method with an exact gradient (*CoLiNe*).

Model We trained AAMs[1] from publicly available images with manually selected landmarks. The models are learned from 456 images from the datasets XM2VTS [10] and IMM [11], which where marked up with 120 landmarks. The data contains 62 identities, multiple expressions, light variation and up to 30 degree out of plane ro-

[1] The AAM containing the complete training set is available on our website: `www.cs.unibas.ch/personen/amberg_brian/aam/`

tation. We used 31 identities with 248 images from the XM2VTS dataset, and 39 identities with 208 images from the IMM dataset. To increase the variability of the model, we added the mirrored version of each image to the training set. The correspondence between the models was established with thin plate splines from the manually selected landmarks. The model was calculated at a resolution of approximately 20000 color pixels. We kept 60 shape and 60 appearance basis vectors. The large variability in the training set facilitates good generalization to novel images.

Multi-Identity Fitting We trained models from subsets of all marked images in a cross-validation framework, using all images *not* from a chosen identity to build an AAM which was tested on the images of that identity. Fitting was started from randomly chosen offsets in the image plane. All fits were initialized with zero shape and rotation parameters, and the approximate size of the face in the image as the scaling factor. All variants of the algorithm were started from the same starting positions. We report the success rate, defined as the ratio of runs that converge within a distance of 5% IED, averaged over the feature points, and the runtime relative to *ICIA*. The starting positions had up to 20% IED misalignment. The speed-performance tradeoff is summarized in figure 3 On the left we show the main competing algorithms. We use *ICIA* with the original incremental warp, *LinCoDe* and *CoDe* with orthonormal warp and *CoNe* and *CoLiNe* applied, for the first time, to AAM fitting with orthonormal incremental warp. For comparison, we also added direct optimisation of the objective function 2.1 with a quasi-Newton method, and the expensive full Gauss-Newton optimisation *CoNe* of the compositional cost. *ICIA* fails to converge reliably on this difficult but realistic dataset. The success rate of *CoDe* is higher than that achievable with direct optimisation using a quasi-Newton method (L-BFGS), while being dramatically faster. *LinCoDe* is as fast as *ICIA* but converges 8 times as often, though not quite as often as *CoDe*. Using the Hessian approximation and the correct gradient (*CoLiNe*) performs no better than approximate gradient descent (*LinCoDe*) but at seven times greater cost. On the right in figure 3, we show that adding regularisation considerably improves the capture range for all methods but *CoDe* and *LinCoDe* continue to give the best speed-performance tradeoffs.

Tracking We applied the algorithms to tracking video sequences. A 5000 frame sequence (Figure 4) of a talking face [5] with a subject which was not in the training set and captured with a different camera and lighting was tracked with the compositional alignment algorithms. All tracks were initialized from a perfect fit to the first frame, and used the orthonormal incremental warp. *ICIA* immediately loses track, even though the inter-frame displacements are rather small, and this is so even when the original ICIA incremental warp is replaced with the new orthonormal warp scheme. However *CoDe*, admittedly running 8 times slower than *ICIA*, and *LinCoDe*, which runs in a time similar to *ICIA*, fail only after approximately 500 frames. When regularisation is added it is possible to track 2500 frames, accurately, with *LinCoDe* and *CoDe* even stays stable for the full 5000 frames sequence, whereas *ICIA* now fails after 500 frames and, even while tracking, delivers clearly inaccurate warps. Full results are in the online material.

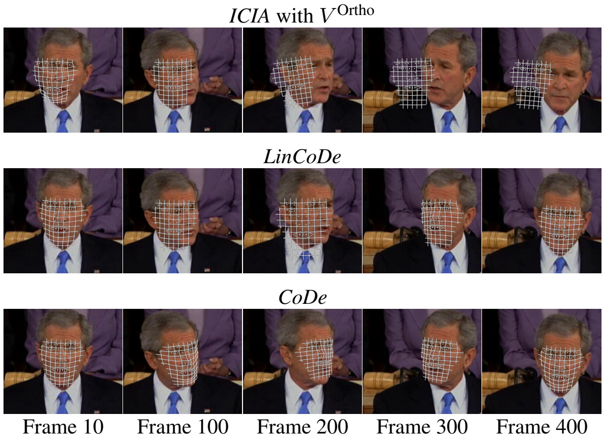

Figure 5. **Tracking a low resolution video with large head motions succeeds with *CoDe*, where *ICIA* fails.** All methods used the orthonormal incremental warp, and relatively strong regularisation. *ICIA* starts to drift in the early frames, while *CoDe* tracks the full sequence. The approximate gradient method *LinCoDe* also suceeds, but looses track of the details for about 100 frames.

To test the behaviour on a more difficult video we used a speech with large head motions and expressive gesture, acquired under uncontrolled lighting and with relatively low resolution of approximately 18 pixels IED. Again *CoDe* tracks the full sequence (figure 5), while *ICIA* fails, never to recover; *LinCoDe* temporarily loses track during a large out of plane rotation, and is on this more difficult dataset, not as accurate as *CoDe*, showing that the fast approximation of the gradient does not come without a cost. Full tracked sequences are shown in the supplementary material.

7. Conclusion

We have derived two new alignment algorithms, *CoDe* and *LinCoDe* which outperform the state of the art image alignment *ICIA*, at a modest cost in terms of runtime. In *ICIA*, as we have shown in the paper, approximations for gradient and Hessian are used that are valid only when model and data match well, and this has the effect of limiting the capture range for convergence. The most powerful of the new algorithms *CoDe* applies compositional descent on an *exact* gradient, and avoids the need for a Hessian by using an orthonormal incremental warp. It achieves a large capture range at the cost of being eight times slower than

ICIA. Similarly *LinCoDe* in which the computed gradient in *CoDe* is replaced by an approximation, requires a similar runtime to *ICIA* but has a considerably greater capture range, though not as great as *CoDe*.

A. Equivalence with *ICIA*

Our derivation of *ICIA* and the original derivation of *ICIA* lead to the same update formula, both algorithms are the same. We highlight in this section the difference between the derivation in [1] and our derivation, to show where the approximations have been made. Equation (24) from [1] is equivalent to our Equation 3.11, if we use V^{Model}. Note that $W \circ W$ in [1] is ill defined, as $W : \mathcal{D} \rightarrow \mathcal{I}$. For simplification we assume constant appearance, though varying appearance introduces another approximation step in [1]. A change of variables on (3.11) leads to

$$\tilde{F}(\boldsymbol{q}, \boldsymbol{p}) = \int_{V(\mathcal{D};\boldsymbol{p})} (a(V^{-1}(\boldsymbol{r};\boldsymbol{p})) - I(W(\boldsymbol{r};\boldsymbol{q})))^2 \left|\nabla V^{-1}(\boldsymbol{r};\boldsymbol{p})\right| \tag{A.1}$$

The derivative of (A.1) at the identity warp is

$$\begin{aligned}\nabla_{\boldsymbol{p}} \tilde{F}(\boldsymbol{q}, \boldsymbol{0}) = 2 \int_{\mathcal{D}} & f(\boldsymbol{r};\boldsymbol{q}) \nabla a(\boldsymbol{r}) \nabla_{\boldsymbol{p}} V^{-1}(\boldsymbol{r};\boldsymbol{0}) \\ &+ \int_{\mathcal{D}} f(\boldsymbol{r};\boldsymbol{q})^2 \nabla_{\boldsymbol{p}} \left|\nabla V^{-1}(\boldsymbol{r};\boldsymbol{0})\right| \\ &+ \oint_{\mathrm{d}\mathcal{D}} f(\boldsymbol{r};\boldsymbol{q})^2 \boldsymbol{n}_{\boldsymbol{0}}(\boldsymbol{r}) \nabla_{\boldsymbol{p}} V(\boldsymbol{r};\boldsymbol{0})\end{aligned} \tag{A.2}$$

where $\boldsymbol{n}_{\boldsymbol{p}}(\boldsymbol{r})$ is the normal of the boundary of $V(\mathcal{D};\boldsymbol{p})$ at $\boldsymbol{r}$. This makes the image derivative disappear in the derivative of the cost function Assuming that the current residual is small, the derivative in (A.2) is approximated as

$$\nabla_{\boldsymbol{p}} \tilde{F}(\boldsymbol{q}, \boldsymbol{0}) = 2 \int_{\mathcal{D}} f(\boldsymbol{r};\boldsymbol{q}) \nabla a(\boldsymbol{r}) \nabla_{\boldsymbol{p}} V^{-1}(\boldsymbol{r};\boldsymbol{0}) \tag{A.3}$$

As we are free to choose the incremental warp, we rename $V \leftarrow V^{-1}$ which makes the Jacobian of (A.3) equal to $\bar{J}$ from (3.20). This shows that the derivation of *ICIA* in [1] is equivalent to taking the Jacobian from the hypothetical mimimum of the objective function.

B. Mapping and Regularisation

In this section we describe how to efficiently calculate the mapping $\boldsymbol{q} = C(\boldsymbol{q}_0, \boldsymbol{p})$, i.e. solve (3.10) and incorporate regularisation into this step. Regularisation is achieved by determining the maximum likelihood (ML) value for the mapped model parameters. The ML mapping of (3.10) is, under the assumption of independently normal distributed shape coefficients $\boldsymbol{\alpha}$, given by

$$\begin{aligned}\boldsymbol{q} = \arg\min_{\boldsymbol{\rho},\boldsymbol{\tau},\boldsymbol{\alpha}} & \left\|\boldsymbol{R}_{\boldsymbol{\rho}}(\boldsymbol{r} + M(\boldsymbol{r})\boldsymbol{\alpha}) + \boldsymbol{\tau} - W(V(\boldsymbol{r};\boldsymbol{p});\boldsymbol{q})\right\|^2_{\boldsymbol{r}\in\mathcal{D}} \\ &+ \lambda \left\|\boldsymbol{\alpha}\right\|\end{aligned} \tag{B.1}$$

with a regularisation parameter λ depending on the noise characteristic of the image. By multiplication of (B.1) with $-\boldsymbol{R}_{\boldsymbol{\rho}}^{-1}$ one arrives at a quadratic problem in terms of the inverse rotation, with a mainly constant matrix. This can be solved efficiently in $O(N^3_{\text{Shape Parameters}})$ operations by precomputing most of the pseudo-inverse of the system.

References

[1] S. Baker and I. Matthews. Equivalence and Efficiency of Image Alignment Algorithms. In *CVPR '01*, volume 1, pages I–1090–I–1097 vol.1, 2001.

[2] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *IJCV*, 56(3):221–255, February 2004.

[3] V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH '99*, pages 187–194. ACM Press, 1999.

[4] H. Burkhardt and N. Diehl. Simultaneous Estimation of Rotation and Translation in Image Sequences. In *Proc. of the European Signal Processing Conference, EUSIPCO-86*, Den Haag, 1986.

[5] T. F. Cootes. Talking face video, October 2008. `www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking\_face.html`.

[6] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active Appearance Models. *PAMI*, 23(6):681–685, 2001.

[7] N. Diehl. *Methoden zur allgemeinen Bewegungsschätzung in Bildfolgen*. PhD thesis, TU Hamburg-Harburg, 1988. Published as Fortschrittsbericht (Reihe 10, Nr. 92) VDI-Zeitschriften, VDI-Verlag.

[8] X. Liu. Generic Face Alignment using Boosted Appearance Model. In *CVPR '07*, pages 1–8, 2007.

[9] I. Matthews and S. Baker. Active Appearance Models Revisited. *IJCV*, 60(2):135–164, November 2004.

[10] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: The Extended M2VTS Database. In *2nd Int. Conf. on Audio and Video-based Biometric Person Authentication*, March 1999.

[11] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann. The IMM face database - an annotated dataset of 240 face images. Technical report, IMM, TU Denmark DTU, may 2004.

[12] S. Romdhani and T. Vetter. Efficient, Robust and Accurate Fitting of a 3D Morphable Model. In *ICCV '03*, Washington, DC, USA, 2003. IEEE Computer Society.

[13] J. Saragih and R. Goecke. A Nonlinear Discriminative Approach to AAM Fitting. In *ICCV '07*, pages 1–8, 2007.

[14] B. C. Vemuri, S. Huang, S. Sahni, C. M. Leonard, C. Mohr, R. Gilmore, and J. Fitzsimmons. An efficient motion estimator with application to medical image registration. *Medical Image Analysis*, pages 79–98, March 1998.

[15] M. Wimmer, F. Stulp, S. Pietzsch, and B. Radig. Learning local objective functions for robust face model fitting. *PAMI*, 30(8):1357–1370, 2008.

[16] H. Wu, X. Liu, and G. Doretto. Face Alignment via Boosted Ranking Model. In *CVPR '08*, pages 1–8, 2008.