Efficient image alignment using linear appearance models

Jose Gonzalez-Mora, Nicolas Guil, Emilio L. Zapata
Department of Computer Architecture
University of Malaga
{jgmora,nico,ezapata}@ac.uma.es

Fernando de la Torre
Robotics Institute
Carnegie Mellon University
ftorre@cs.cmu.edu

Abstract

Visual tracking is a key component in many computer vision applications. Linear subspace techniques (e.g. eigentracking) are one of the most popular approaches to align templates with appearance variations (e.g. illumination, iconic changes). A number of well known tracking algorithms have been proposed in the last years to accurately fit these models to images. Computational efficiency is an important limitation in object tracking algorithms and different efficient techniques, such as the "projected-out" optimization, have been proposed. They reduce the computational cost using an efficient formulation in which many of the involved operations can be precomputed. On the other hand, alternative "simultaneous" algorithms jointly optimize pose and appearance parameters, providing better performance but increasing the computational cost.

In this paper, we propose an algorithm for efficient linear appearance model fitting based on the inverse compositional simultaneous optimization of pose and appearance. We introduce a novel formulation which reduces the required computational time while maintaining similar convergence properties of previous "simultaneous" approaches. Experimental results illustrate the capabilities of this algorithm in face tracking.

1. Introduction

Linear appearance models are an effective method for representing visual changes in an object class. The popularity of these models comes from their simplicity and computational efficiency. They are based on the assumption that the visual appearance of the considered target lie in a low dimensional manifold and approximate the relationship between the input image and the manifold by means of a linear mapping (such a PCA). This approach has proven to be effective in many applications. In particular, linear subspace models haven been extensively used for representing human faces, approximating facial images viewed under different conditions (e.g. illumination, expression or identities).

In many applications, the performance of the linear appearance model fitting algorithm must be balanced with practical speed requirements. Whether the task is offline, or real-time, constraints on processing time are typically an important issue in many applications. Several authors have proposed efficient techniques which reduce the execution time [7, 9] by projecting out the appearance variations of the linear basis in the fitting process. However, as other authors have pointed out from empirical testing, the performance of these algorithms is worse than the obtained by the simultaneous optimization of pose and appearance parameters [5, 3].

In this paper, we propose an efficient formulation for the inverse compositional simultaneous algorithm that reduces the computational cost while maintaining the convergence properties. The rest of the paper in organized as follows. Section 2 reviews previous works on linear models fitting algorithms. Section 3 presents a novel formulation for linear appearance model fitting algorithm. Section 5 experimental results are provided for the tracking of face images. Finally, in section 5 some conclusions and future work are presented.

2. Previous work

Linear appearance models [13, 12, 2, 3, 4] have been extensively studied within the pattern recognition community. These models can be used in conjunction with pose parameterizations to analyze the visual appearance of moving targets in images by means of image alignment techniques. Image alignment using linear appearance variation has been considered by a number of previous works, most notably by Black and Jepson for general appearance variation [2], by Hager and Belhumeur for illumination changes [7], by Buenaposada et al. [3] for face expression and illumination changes and by Cootes and Taylor for non-rigid face modeling [4].

The image alignment algorithms using linear appearance models optimize the pose and appearance parameters to minimize the difference between the model and an input image. Gradient descent is the most popular approach to im-

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

age alignment and different model fitting methods based on the Lucas-Kanade algorithm [8] have been developed in the last years. An efficient technique, the "projected out" optimization, deals with pose changes as well as appearance variations without increasing computational costs. It performs the alignment process in two steps: first, it projects out the appearance variation in the target model and solves for the warp parameters and, then, it computes the appearance parameters in a second step. Hager and Belhumeur [7] initially proposed an inverse additive formulation of this "projected out" algorithm. Baker, Gross and Matthews [1, 9, 5] introduced a different formulation using the inverse compositional framework.

Alternatively, a second solution called the "simultaneous" algorithm [1, 9, 5, 3], optimizes simultaneously pose and appearance parameters of linear appearance models placed onto a new image. It provides better convergence properties but at the cost of performing slower than the "projected out" algorithm. In order to reduce the required computational cost, different alternatives have been proposed. Mercier et al. [10] proposed a modified version of the "simultaneous" algorithm based on an approximated solution that regularizes the gradient descent update direction using a set of weighting coefficients instead of the Hessian matrix. Buenaposada et al. [3] proposed an alternative formulation of the simultaneous algorithm based on the inverse additive approach that reduces the computational cost by introducing an efficient Jacobian matrix factorization.

In the following section, we introduce a novel inverse compositional formulation of the simultaneous algorithm to jointly optimize pose and appearance parameters. The proposed technique has similar convergence properties than the original "simultaneous" algorithm but it reduces the computational cost by effectively reorganizing the involved operations into an efficient algorithm.

3. Linear appearance model fitting formulation

In this section, we formulate different methods for image alignment using linear appearance models. First, the "projected out" and the "simultaneous" optimization algorithms are presented for comparison purposes. Then, an efficient alternative formulation of the "simultaneous" algorithm is presented. We follow an inverse compositional approach because warping the linear model template instead of the target image accelerates the image fitting process by precomputing different required quantities.

The target appearance is represented using a reference template $t(\mathbf{x})$ and a linear basis of images $\{a_k(\mathbf{x})\}, k = 1, ..., m$ modeling the changes which can occur in the object appearance at each pixel $\mathbf{x} = [x, y]^T$. The tracking process can be expressed as the following minimization process:

$$\min_{\mathbf{p},\lambda} \sum_{\mathbf{x}} \left[t(\mathbf{x}) + \sum_k \lambda_k a_k(\mathbf{x}) - i(\mathbf{w}(\mathbf{x}, \mathbf{p})) \right]^2 \quad (1)$$

where $\mathbf{p} = (p_1, p_2, \cdots, p_n)^T$ and $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_m)^T$ are the pose and appearance parameter vectors respectively.

3.1. Projected out optimization

Hager and Belhumeur [7] proposed an efficient "projected out" algorithm that reduces the computational cost by marginalizing appearance changes contained in the linear basis in the fitting process. It was later reformulated by Baker, Gross and Matthews [1, 9, 5] using the inverse compositional scheme.

The error function in equation 1 can be expressed as:

$$\epsilon(\mathbf{p}, \lambda) = \sum_{\mathbf{x}} \left[t(\mathbf{x}) + \sum_k \lambda_k a_k(\mathbf{x}) - i(\mathbf{w}(\mathbf{x}, \mathbf{p})) \right]^2_{span(\mathbf{a}_k)} + \sum_{\mathbf{x}} \left[t(\mathbf{x}) + \sum_k \lambda_k a_k(\mathbf{x}) - i(\mathbf{w}(\mathbf{x}, \mathbf{p})) \right]^2_{span(\mathbf{a}_k)^\perp} \quad (2)$$

with $span(\mathbf{a}_k)$ representing the linear subspace spanned by the vectors $\mathbf{a}_k = [a_k(\mathbf{x}_1), a_k(\mathbf{x}_2), \cdots, a_k(\mathbf{x}_d)]^T$ and $span(\mathbf{a}_k)^\perp$ its orthogonal complement.

This equation can be simplified considering that the norm in the second term only operates on the error component in the orthonormal complement of $span(\mathbf{a}_k)$. Then, component in $span(\mathbf{a}_k)$ can be dropped, resulting the following expression:

$$\epsilon(\mathbf{p}, \lambda) = \sum_{\mathbf{x}} \left[t(\mathbf{x}) + \sum_k \lambda_k a_k(\mathbf{x}) - i(\mathbf{w}(\mathbf{x}, \mathbf{p})) \right]^2_{span(\mathbf{a}_k)} + \sum_{\mathbf{x}} \left[t(\mathbf{x}) - i(\mathbf{w}(\mathbf{x}, \mathbf{p})) \right]^2_{span(\mathbf{a}_k)^\perp} \quad (3)$$

The second of these two terms does not depend on λ. For any $\mathbf{p}$, the minimum value of the first term is always 0 because the term $\sum_k \lambda_k a_k(\mathbf{x})$ can represent any vector in $span(\mathbf{a}_k)$. As a result, the simultaneous minimum over both $\mathbf{p}$ and λ can be found sequentially by first minimizing the second term with respect to $\mathbf{p}$ alone, and then treating the optimal value of $\mathbf{p}$ as a constant to minimize the first term with respect to λ.

Using a inverse compositional approach for the optimization, the error function can be referred to incremental

parameters $\Delta\mathbf{p}$:

$$\epsilon(\Delta\mathbf{p}) = \Bigg[t(\mathbf{w}(\mathbf{x},\Delta\mathbf{p}) + \sum_k \lambda_k a_k(\mathbf{w}(\mathbf{x},\Delta\mathbf{p})) - i(\mathbf{w}(\mathbf{x},\mathbf{p}))\Big]^2_{span(\mathbf{a}_k)} + \Big[t(\mathbf{w}(\mathbf{x},\Delta\mathbf{p})) - i(\mathbf{w}(\mathbf{x},\mathbf{p}))\Big]^2_{span(\mathbf{a}_k)^\perp} \tag{4}$$

The optimal pose obtained from the second term can be expressed as:

$$\Delta\mathbf{p} = -\mathbf{H}_{po}^{-1} \sum_{\mathbf{x}} \mathbf{j}_{po}(\mathbf{x})^T e_{po}(\mathbf{x}) \tag{5}$$

with:

$$e_{po}(\mathbf{x}) = t(\mathbf{x}) - i(\mathbf{w}(\mathbf{x},\mathbf{p})) \tag{6}$$

$$\mathbf{j}_{po}(\mathbf{x}) = \nabla t \frac{\partial \mathbf{w}}{\partial \mathbf{p}}(\mathbf{x},\mathbf{0}) - \sum_k \left[\sum_y a_k(\mathbf{y}) \nabla t \frac{\partial \mathbf{w}}{\partial \mathbf{p}}(\mathbf{y},\mathbf{0}) \right] a_k(\mathbf{x}) \tag{7}$$

$$\mathbf{H}_{po} = \sum_{\mathbf{x}} \mathbf{j}_{po}(\mathbf{x})^T \mathbf{j}_{po}(\mathbf{x}) \tag{8}$$

Computation of $\mathbf{j}_{po}(\mathbf{x})$ is carried out by projecting $\nabla t \frac{\partial \mathbf{w}}{\partial \mathbf{p}}(\mathbf{x},\mathbf{0})$ vectors into $span(\mathbf{a}_k)^\perp$, by removing the component in the direction of $\mathbf{a}_k$, for $k = 1, ..., m$ in turns.

The optimal appearance parameters are given from the first term as:

$$\lambda = \left(\sum_{\mathbf{x}} \mathbf{a}(\mathbf{x})^T \mathbf{a}(\mathbf{x}) \right)^{-1} \sum_{\mathbf{x}} \mathbf{a}(\mathbf{x})^T e_{po}(\mathbf{x}) \tag{9}$$

with $\mathbf{a}(\mathbf{x}) = [a_1(\mathbf{x}), a_2(\mathbf{x}), \cdots, a_m(\mathbf{x})]$

It is worth pointing out that these expressions are based on the following implicit approximation in equation 4:

$$\sum_k \lambda_k a_k(\mathbf{w}(\mathbf{x},\Delta\mathbf{p})) \approx \sum_k \lambda_k a_k(\mathbf{x}) \tag{10}$$

3.2. Simultaneous pose and appearance optimization

The error function that has to be minimized is given by:

$$\epsilon(\Delta\lambda,\Delta\mathbf{p}) = \sum_{\mathbf{x}} \Big[t(\mathbf{w}(\mathbf{x},\Delta\mathbf{p})) + \sum_k (\lambda_k + \Delta\lambda_k) a_k(\mathbf{w}(\mathbf{x},\Delta\mathbf{p})) - i(\mathbf{w}(\mathbf{x},\mathbf{p})) \Big]^2 \tag{11}$$

The first order expansion of eq. 11 , neglecting second order terms, can be expressed as:

$$\epsilon(\Delta\lambda,\Delta\mathbf{p}) = \sum_{\mathbf{x}} \Bigg[t(\mathbf{x}) + \sum_k \lambda_k a_k(\mathbf{x}) - i(\mathbf{w}(\mathbf{x},\mathbf{p})) + \left(\nabla t + \sum_k \lambda_k \nabla a_k(\mathbf{x}) \right) \frac{\partial \mathbf{w}}{\partial \mathbf{p}} \Delta\mathbf{p} + \sum_k \Delta\lambda_k a_k(\mathbf{x}) \Bigg]^2 \tag{12}$$

Considering the combined parameters vector $\mathbf{q} = \left(\mathbf{p}^T, \lambda^T\right)^T$ and $\Delta\mathbf{q} = \left(\Delta\mathbf{p}^T, \Delta\lambda^T\right)^T$, the joint optimal parameters can be obtained by:

$$\Delta\mathbf{q} = -\mathbf{H}_{sic}^{-1} \sum_x \mathbf{j}_{sic}(\mathbf{x})^T e_{sic}(\mathbf{x}) \tag{13}$$

with:

$$e_{sic}(\mathbf{x}) = t(\mathbf{x}) + \sum_k \lambda_k a_k(\mathbf{x}) - i(\mathbf{w}(\mathbf{x},\mathbf{p})) \tag{14}$$

$$\mathbf{j}_{sic}(\mathbf{x}) = \Bigg[\left(\nabla t + \sum_k \lambda_k \nabla a_k \right) \frac{\partial \mathbf{w}}{\partial p_1}, ..., \left(\nabla t + \sum_k \lambda_k \nabla a_k \right) \frac{\partial \mathbf{w}}{\partial p_n}, a_1(\mathbf{x}), \cdots, a_m(\mathbf{x}) \Bigg] \tag{15}$$

$$\mathbf{H}_{sic} = \sum_{\mathbf{x}} \mathbf{j}_{sic}^T(\mathbf{x}) \mathbf{j}_{sic}(\mathbf{x}) \tag{16}$$

The steepest descent images computed in equation 15 depend on the current appearance parameters λ and must be recomputed at every iteration. In order to reduce the computational cost of the image alignment process, in the following section we derive an alternative formulation for this algorithm.

3.3. Efficient formulation of the simultaneous pose and appearance algorithm

The elements in the Steepest Descent Matrix in equation 12 can be rearranged as follows:

$$\epsilon(\Delta\lambda,\Delta\mathbf{p}) = \sum_{\mathbf{x}} \Bigg[t(\mathbf{x}) + \sum_k \lambda_k a_k(\mathbf{x}) - i(\mathbf{w}(\mathbf{x},\mathbf{p})) + \left[\begin{bmatrix} \nabla_x t \\ \nabla_x a_1(\mathbf{x}) \\ \vdots \\ \nabla_x a_m(\mathbf{x}) \end{bmatrix}^T, \begin{bmatrix} \nabla_y t \\ \nabla_y a_1(\mathbf{x}) \\ \vdots \\ \nabla_y a_m(\mathbf{x}) \end{bmatrix}^T \right] \mathbf{C}(\lambda) \frac{\partial \mathbf{w}}{\partial \mathbf{p}} \Delta\mathbf{p} + \sum_{\mathbf{x}} \Delta\lambda_k a_k(\mathbf{x}) \Bigg]^2 \tag{17}$$

with:

$$\mathbf{C}(\lambda) = \begin{bmatrix} [1,\lambda]^T & \mathbf{0} \\ \mathbf{0} & [1,\lambda]^T \end{bmatrix} \tag{18}$$

Grouping together all the elements that can be recomputed, equation 17 can be rewritten as:

$$\epsilon(\Delta\lambda, \Delta\mathbf{p}) = \sum_{\mathbf{x}} \Bigg[t(\mathbf{x}) + \sum_k \lambda_k a_k(\mathbf{x}) - i(\mathbf{w}(\mathbf{x},\mathbf{p})) + \left[\begin{bmatrix} \nabla_x t \\ \nabla_x a_1(\mathbf{x}) \\ \vdots \\ \nabla_x a_m(\mathbf{x}) \end{bmatrix}^T, \begin{bmatrix} \nabla_y t \\ \nabla_y a_1(\mathbf{x}) \\ \vdots \\ \nabla_y a_m(\mathbf{x}) \end{bmatrix}^T \right] \frac{\partial \mathbf{w}}{\partial \mathbf{p}}^* \left[\mathbf{I}_{\mathbf{l}\times\mathbf{l}} \otimes \mathbf{C}(\lambda)\right] \Delta\mathbf{p} + \sum_k \Delta\lambda_k a_k(\mathbf{x}) \Bigg]^2 \tag{19}$$

with $\frac{\partial \mathbf{w}}{\partial \mathbf{p}}^*$ being a rearrangement of the Jacobian matrix evaluated at $\mathbf{p} = \mathbf{0}$ and $I_{l\times l}$ the identity matrix of dimension l (where l depends on the considered warping function). See appendix A for a description of different motion parameterizations.

It results in:

$$\epsilon(\Delta\lambda, \Delta\mathbf{p}) = \sum_{\mathbf{x}} \left[e(\mathbf{x}) + [\mathbf{j_p}(\mathbf{x})\mathbf{C}^*(\lambda), \mathbf{j}_\lambda(\mathbf{x})] \begin{bmatrix} \Delta\mathbf{p} \\ \Delta\lambda \end{bmatrix} \right]^2 \tag{20}$$

with:

$$e(\mathbf{x}) = t(\mathbf{x}) + \sum_k \lambda_k a_k(\mathbf{x}) - i(\mathbf{w}(\mathbf{x},\mathbf{p})) \tag{21}$$

$$\mathbf{C}^*(\lambda) = \mathbf{I}_{l\times l} \otimes \mathbf{C}(\lambda) \tag{22}$$

$$\mathbf{j_p}(\mathbf{x}) = \left[\begin{bmatrix} \nabla_x t \\ \nabla_x a_1(\mathbf{x}) \\ \vdots \\ \nabla_x a_m(\mathbf{x}) \end{bmatrix}^T, \begin{bmatrix} \nabla_y t \\ \nabla_y a_1(\mathbf{x}) \\ \vdots \\ \nabla_y a_m(\mathbf{x}) \end{bmatrix}^T \right] \frac{\partial \mathbf{w}}{\partial \mathbf{p}}^* \tag{23}$$

$$\mathbf{j}_\lambda(\mathbf{x}) = [a_1(\mathbf{x}), a_2(\mathbf{x}), \cdots, a_m(\mathbf{x})] \tag{24}$$

We now optimize eq. 20, using the partial derivative with respect to the joint parameter vector $\Delta\mathbf{q} = \left[\Delta\mathbf{p}^T, \Delta\lambda^T\right]^T$. Using the vectorial notation $\mathbf{e} = [e(\mathbf{x}_1), e(\mathbf{x}_2), \cdots, e(\mathbf{x}_d)]^T$, $\mathbf{J}_\lambda = \left[\mathbf{j}_\lambda^T(\mathbf{x}_1), \mathbf{j}_\lambda^T(\mathbf{x}_2), \cdots, \mathbf{j}_\lambda^T(\mathbf{x}_d)\right]^T$ and $\mathbf{J_p} = \left[\mathbf{j}_\mathbf{p}^T(\mathbf{x}_1), \mathbf{j}_\mathbf{p}^T(\mathbf{x}_2), \cdots, \mathbf{j}_\mathbf{p}^T(\mathbf{x}_d)\right]^T$, the following expression is obtained:

$$\begin{bmatrix} \mathbf{C}^*(\lambda)^T \mathbf{J}_\mathbf{p}^T \\ \mathbf{J}_\lambda^T \end{bmatrix} [\mathbf{J_p}\mathbf{C}^*(\lambda), \mathbf{J}_\lambda] \Delta\mathbf{q} = - \begin{bmatrix} \mathbf{C}^*(\lambda)^T \mathbf{J}_\mathbf{p}^T \\ \mathbf{J}_\lambda^T \end{bmatrix} \mathbf{e} \tag{25}$$

The resulting composed parameters increment is then given by:

$$\Delta\mathbf{q} = - \begin{bmatrix} \mathbf{C}^*(\lambda)^T \mathbf{J}_\mathbf{p}^T \mathbf{J}_p \mathbf{C}^*(\lambda) & \mathbf{C}^*(\lambda)^T \mathbf{J}_\mathbf{p}^T \mathbf{J}_\lambda \\ \mathbf{J}_\lambda^T \mathbf{J_p} \mathbf{C}^*(\lambda) & \mathbf{J}_\lambda^T \mathbf{J}_\lambda \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{C}^*(\lambda)^T \mathbf{J}_\mathbf{p}^T \\ \mathbf{J}_\lambda^T \end{bmatrix} \mathbf{e} \tag{26}$$

Applying the matrix inversion lemma:

$$\Delta\mathbf{p} = - \left((\mathbf{J_p}\mathbf{C}^*(\lambda))^T \mathbf{N}_\lambda (\mathbf{J_p}\mathbf{C}^*(\lambda)) \right)^{-1} (\mathbf{J_p}\mathbf{C}(\lambda))^T \mathbf{N}_\lambda \mathbf{e} \tag{27}$$

$$\Delta\lambda = \left(\mathbf{J}_\lambda^T \mathbf{N}_\mathbf{p}^* \mathbf{J}_\lambda\right)^{-1} \mathbf{J}_\lambda^T \mathbf{N_p} \mathbf{e} \tag{28}$$

with $\mathbf{N}_\mathbf{p}^* = \mathbf{I} - (\mathbf{J_p}\mathbf{C}^*(\lambda))(\mathbf{J_p}\mathbf{C}^*(\lambda))^+$ and $\mathbf{N}_\lambda = \mathbf{I} - \mathbf{J}_\lambda \mathbf{J}_\lambda^+$.

Considering that $\mathbf{N}_\mathbf{p}^* = \mathbf{N}_\mathbf{p}^{*T} \mathbf{N}_\mathbf{p}^*$ and $\mathbf{N}_\lambda = \mathbf{N}_\lambda^T \mathbf{N}_\lambda$, it results:

$$\Delta\mathbf{p} = - \left(\mathbf{C}^T(\lambda)(\mathbf{N}_\lambda \mathbf{J_p})^T (\mathbf{N}_\lambda \mathbf{J_p}) \mathbf{C}(\lambda)\right)^{-1} \mathbf{C}^T(\lambda)(\mathbf{N}_\lambda \mathbf{J_p})^T \mathbf{e} \tag{29}$$

$$\Delta\lambda = ((\mathbf{N}_\mathbf{p}^* \mathbf{J}_\lambda)^T (\mathbf{N}_\mathbf{p}^* \mathbf{J}_\lambda))^{-1} (\mathbf{N}_\mathbf{p}^* \mathbf{J}_\lambda)^T \mathbf{e} \tag{30}$$

Here $\mathbf{N}_\lambda \mathbf{J_p}$ represents the component of $\mathbf{J_p}$ orthogonal to $\mathbf{J}_\lambda$. Similarly, $\mathbf{N}_\mathbf{p}^* \mathbf{J}_\lambda$ represents the component of $\mathbf{J}_\lambda$ orthogonal to different linear combinations of $\mathbf{J_p}$ (whose coefficients are $\mathbf{C}^*(\lambda)$). It is not necessary to compute the orthogonal subspace $\mathbf{N}_\mathbf{p}^* \mathbf{J}_\lambda$ for each value of λ as if we find the orthogonal complement of $\mathbf{J}_\lambda$ with respect to $\mathbf{J_p}$ it will be orthogonal to $\mathbf{J_p}\mathbf{C}^*(\lambda)$ as well.

Finally:

$$\Delta\mathbf{p} = - \left(\mathbf{C}^*(\lambda)^T \mathbf{H}_{\mathbf{p}\perp} \mathbf{C}^*(\lambda)\right)^{-1} \mathbf{C}^*(\lambda)^T \mathbf{J}_{\mathbf{p}\perp}^T \mathbf{e} \tag{31}$$

$$\Delta\lambda = \mathbf{H}_{\lambda\perp}^{-1} \mathbf{J}_{\lambda\perp}^T \mathbf{e} \tag{32}$$

with:

$$\mathbf{J}_{\mathbf{p}\perp} = \mathbf{N}_\lambda \mathbf{J_p} \tag{33}$$

$$\mathbf{J}_{\lambda\perp} = \mathbf{N_p} \mathbf{J}_\lambda \tag{34}$$

$$\mathbf{H}_{\mathbf{p}\perp} = \mathbf{J}_{\mathbf{p}\perp}^T \cdot \mathbf{J}_{\mathbf{p}\perp} \tag{35}$$

$$\mathbf{H}_{\lambda\perp} = \mathbf{J}_{\lambda\perp}^T \cdot \mathbf{J}_{\lambda\perp} \tag{36}$$

$$\mathbf{N_p} = \mathbf{I} - \mathbf{J_p}\mathbf{J_p^+} \tag{37}$$

In this formulation, all the components $\mathbf{J_{p\perp}}$, $\mathbf{J_{\lambda\perp}}$, $\mathbf{H_{p\perp}}$, $\mathbf{H_{\lambda\perp}}$ can be precomputed, reducing the required computations on each iteration compared to the standard simultaneous inverse compositional approach presented in [1, 6]. The resulting steps in the algorithm and the associated computation times are then summarized in tables 1 and 2 respectively. We can see how the iteration time is reduced, obtaining a linear increment of the computational cost in respect to the number of appearance vectors.

Template alignment algorithm
(Efficient formulation for the SIC algorithm)

Precompute:

(3) Compute the gradients ∇t and ∇a_k for $k = 1, \cdots, m$
(4) Compute the Jacobian $\frac{\partial \mathbf{w}}{\partial \mathbf{p}}$ evaluated at $(\mathbf{x}, \mathbf{0})$
(5a) Compute the steepest descent matrix $\mathbf{J_p}$ and find the orthogonal complement to $a_k(\mathbf{x})$.
(5b) Compute the steepest descent matrix $\mathbf{J}_\lambda$ and find the orthogonal complement to $\mathbf{J_p}$.
(6a) Compute the hessian matrix $\mathbf{H_{p\perp}}$ and invert it.
(6b) Compute the hessian matrix $\mathbf{H_{\lambda\perp}}$ and invert it.

Iterate:

(1) Obtain $i(\mathbf{w}(\mathbf{x}, \mathbf{p}))$, warping the input image $i(\mathbf{x})$
(2) Compute the error image $e_{sic}(\mathbf{x})$ using eq. 21
(7) Use equations 31 and 32 to compute $\Delta\mathbf{p}$ and $\Delta\lambda$
(8) Update the parameters using $\mathbf{w}(\mathbf{x}, \mathbf{p}) \leftarrow \mathbf{w}(\mathbf{x}, \mathbf{p}) \circ \mathbf{w}(\mathbf{x}, \Delta\mathbf{p})^{-1}$ and $\lambda \leftarrow \lambda + \Delta\lambda$

Table 1. Efficient formulation of the simultaneous inverse compositional linear appearance model fitting algorithm.

4. Experimental results

In this section, we provide an empirical comparison of the convergence and computational costs for the different image alignment algorithms proposed in previous section. We make use of the "Database of Moving Faces and People" [11] kindly provided by Perception Lab. at the University of Texas at Dallas. This database contains images and video sequences of different subjects captured in several sessions.

First, we analyze the performance of the algorithms in an experiment involving static images. A linear appearance model is built from different face images using PCA analysis (the selected template dimension is 100×100 pixels and the experiments were repeated for 4 and 6 training images). Then, we use one of the training images as input to the fitting algorithms. Affine transformations (six parameters warps) are generated by randomly perturbing the four corners of the model previously aligned with its instance in the considered image. We then try to recover the initial pose running the different fitting algorithms. Appearance parameters are initialized to $\mathbf{0}$. Two magnitudes were measured: the *average frequency of convergence* and the *average rate of convergence*. We perform as many experiments as needed to let all the algorithms converge at least 25 times, and 15 iterations per experiment were used. A maximum spatial error of one pixel is given as criterion for convergence. Figure 1 represents the obtained results for a basis containing 5 appearance vectors. We can see how "simultaneous" approaches achieve better performance than the "projected-out" algorithms.

On the other hand, traditional "simultaneous" formulations require a higher number of operations per iteration than the "projected out", resulting in increased execution times as presented in figure 2. However, it can be seen as the efficient implementation of the simultaneous algorithm allows for competitive times while keeping the same performance. The offline computation of different numerical structures drastically reduces the iteration time. All the measured times correspond to unoptimized matlab implementations.

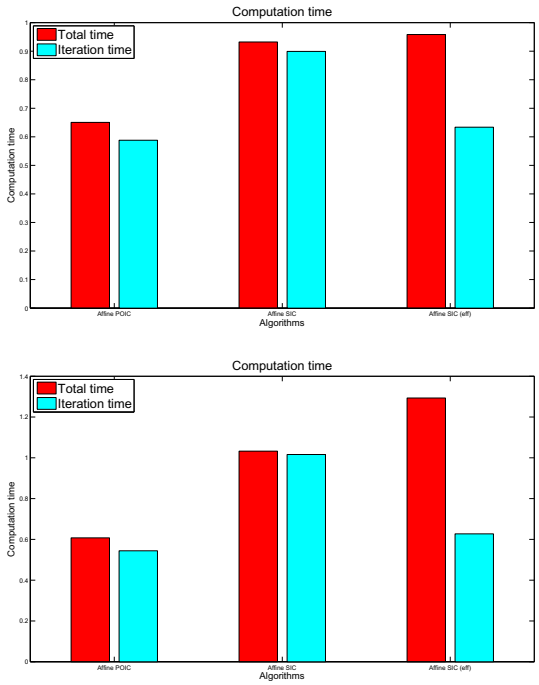

Figure 2. Execution time for different fitting algorithms. Displayed times correspond to the total number of iterations (15). Top image corresponds to the linear model containing 3 appearance vectors. The bottom image corresponds to the model with 5 appearance vectors

Precomputation

Step 3	Step 4	Step 5	Step 6	Total
$O(md)$	$O(nd)$	$O(nmd)$	$O(n^2m^2d + n^3m^3)$	$O(n^2m^2d + n^3m^3)$

Per Iteration

Step 1	Step 2	Step 7	Step 8	Total
$O(nd)$	$O(md)$	$O(mnd + n^3m^3)$	$O(n^2 + m)$	$O(mnd + n^3m^3)$

Table 2. Computation cost of one iteration of the efficient inverse compositional linear appearance models fitting algorithm. d is the number of pixels in the template image $t(\mathbf{x})$, n is the number of warp parameters, m is the number of illumination vectors.

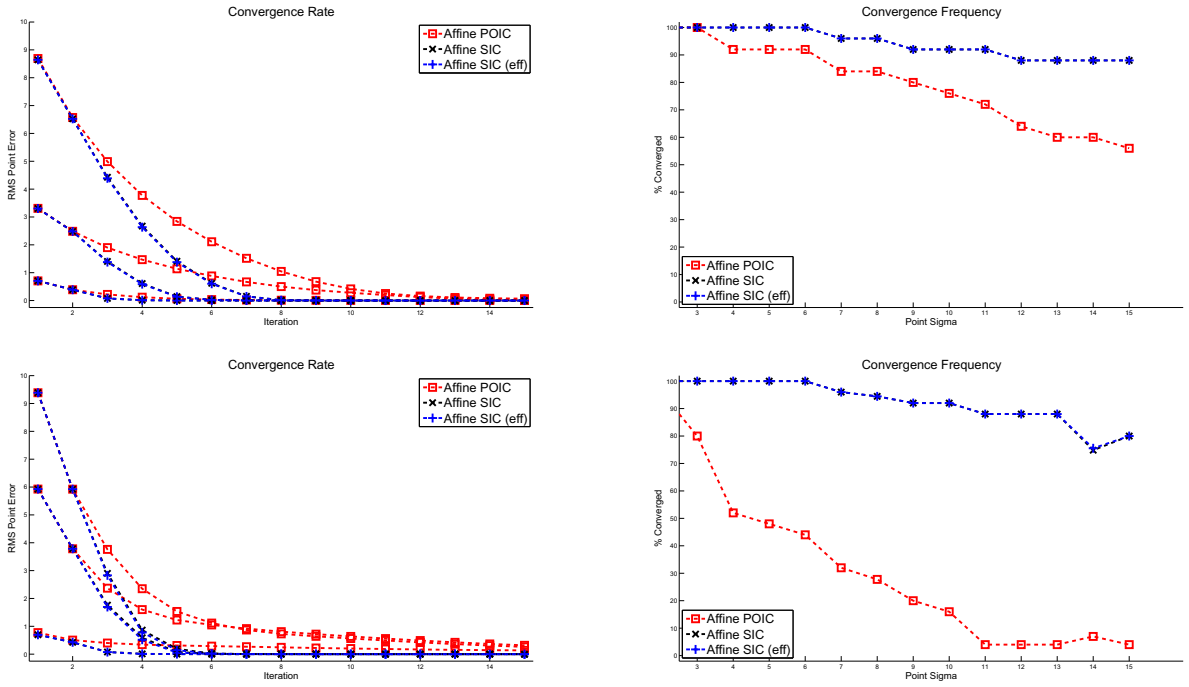

Figure 1. Comparison between different inverse compositional algorithms: "Projected-out", "Simultaneous" and the efficient "Simultaneous" formulation. The top representation correspond to the linear model containing 3 appearance vectors. The bottom graphs correspond to the model with 5 appearance vectors

In a second experiment we apply the efficient simultaneous inverse composition algorithm to the tracking of face video sequences to test the generalization properties of the algorithm. In this case, the appearance basis are used to model expression changes in a considered subject. Figure 3 represents the results for an example sequence. First we learn a linear model from a training sequence. We choose a modular eigenspace composed of three areas, both eyes and the mouth, resulting in basis with dimensions 5 for mouth (23×23 pixels) and 4 for each eye region (15×18 pixels). The model is used to track another sequence of the same subject captured in a different session using a rotation/translation/scale (RTS) parameterization for the motion function. While the iteration time required for the proposed efficient formulation is greatly reduced, the visual inspection of the tracking results confirms the accuracy of the technique when fitting new sequences not included in the training set.

5. Conclusion and future work

In this paper, we have introduced an efficient formulation of the simultaneous inverse compositional algorithm for the problem of image alignment using linear appearance models. This approach reduces the computational cost of the algorithm as the computational time increases linearly with respect to the number of appearance vectors (versus the quadratic dependency of the standard formulation). Additionally, it achieves higher convergence rates and frequencies of convergence that alternative efficient minimization schemes such as the "projected out" algorithm. The performance of the proposed algorithm is empirically analyzed by

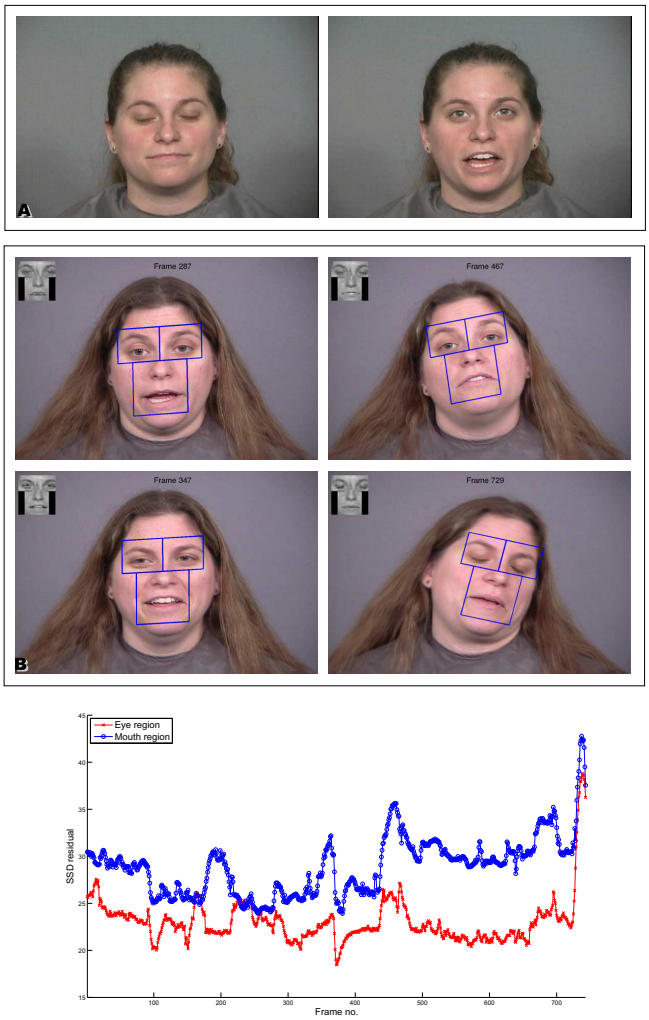

Figure 3. Using the simultaneous inverse compositional algorithm to track a face sequence. (a) Example images from the training sequence. (b) Some of the obtained frames in the tracking process. In the graph, the sum of squares differences errors for the mouth and eye regions is represented

comparing the results achieved by these model fitting techniques on a dataset of facial images. Future work will include the study of efficient fitting formulations for non-rigid motion.

A. Motion parameterizations

A.1. Rotation, translation and scale model

The rotation, translation and scale model is described by four parameters $\mathbf{p} = [s, \theta, t_x, t_y]^T$:

$$\mathbf{w}(\mathbf{x}, \mathbf{p}) = (s+1)\mathbf{R}(\theta) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{38}$$

with $\mathbf{R}(\theta)$ a 2D rotation matrix.

The Jacobian matrix is given by:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial w_x}{\partial s}, \frac{\partial w_x}{\partial \theta}, \frac{\partial w_x}{\partial t_x}, \frac{\partial w_x}{\partial t_y} \\ \frac{\partial w_y}{\partial s}, \frac{\partial w_y}{\partial \theta}, \frac{\partial w_y}{\partial t_x}, \frac{\partial w_y}{\partial t_y} \end{bmatrix} = \begin{bmatrix} cos\theta x - sin\theta y & -(s+1)(sin\theta x + cos\theta y) & 1 & 0 \\ sin\theta x + cos\theta y & (s+1)(cos\theta x - sin\theta y) & 0 & 1 \end{bmatrix} \tag{39}$$

For the identity transformation ($\mathbf{p} = \mathbf{0}$) its value is:

$$\left.\frac{\partial \mathbf{w}}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}} = \begin{bmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \end{bmatrix} \tag{40}$$

The resulting matrices $\frac{\partial \mathbf{w}}{\partial \mathbf{p}}^*$ and $\mathbf{C}^*(\lambda)$ are:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{p}}^* = \left[\begin{bmatrix} xI_{m' \times m'} & -yI_{m' \times m'} \\ yI_{m' \times m'} & xI_{m' \times m'} \end{bmatrix}, I_{2m' \times 2m'} \right] \tag{41}$$

$$\mathbf{C}^*(\lambda) = \mathbf{I}_{2\times 2} \otimes \mathbf{C}(\lambda) \tag{42}$$

with $m' = m + 1$.

A.2. Affine model

In this case the warp function is represented by

$$\mathbf{w}(\mathbf{x}, \mathbf{p}) = \begin{bmatrix} (1+p_1) & p_3 \\ p_2 & (1+p_4) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} p_5 \\ p_6 \end{bmatrix} \tag{43}$$

The Jacobian matrix does not depend on the evaluation point $\mathbf{p}$ and is given by the following expression:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial \mathbf{w}_x}{\partial p_1}, \frac{\partial \mathbf{w}_x}{\partial p_2}, \frac{\partial \mathbf{w}_x}{\partial p_3}, \frac{\partial \mathbf{w}_x}{\partial p_4}, \frac{\partial \mathbf{w}_x}{\partial p_5}, \frac{\partial \mathbf{w}_x}{\partial p_6} \\ \frac{\partial \mathbf{w}_y}{\partial p_1}, \frac{\partial \mathbf{w}_y}{\partial p_2}, \frac{\partial \mathbf{w}_y}{\partial p_3}, \frac{\partial \mathbf{w}_y}{\partial p_4}, \frac{\partial \mathbf{w}_y}{\partial p_5}, \frac{\partial \mathbf{w}_y}{\partial p_6} \end{bmatrix} = \begin{bmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{bmatrix} = \left.\frac{\partial \mathbf{w}}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}} \tag{44}$$

The corresponding $\frac{\partial \mathbf{w}}{\partial \mathbf{p}}^*$ and $\mathbf{C}^*(\lambda)$ matrices for the affine parameterization are:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{p}}^* = [x\mathbf{I}_{2m' \times 2m'}, y\mathbf{I}_{2m' \times 2m'}, \mathbf{I}_{2m' \times 2m'}] \tag{45}$$

$$\mathbf{C}^*(\lambda) = \mathbf{I}_{3\times 3} \otimes \mathbf{C}(\lambda) \tag{46}$$

with $m' = m + 1$

A.3. Projective model

The projective warp function can be expressed as:

$$\mathbf{w}(\mathbf{x},\mathbf{p}) = \begin{bmatrix} r(\mathbf{x},\mathbf{p}) \\ s(\mathbf{x},\mathbf{p}) \\ t(\mathbf{x},\mathbf{p}) \end{bmatrix} = \begin{bmatrix} (1+p_1) & p_4 & p_7 \\ p_2 & (1+p_5) & p_8 \\ p_3 & p_6 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{47}$$

where the transformed point coordinates are obtained dividing by the homogeneous coordinate:

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} r(\mathbf{x},\mathbf{p})/t(\mathbf{x},\mathbf{p}) \\ s(\mathbf{x},\mathbf{p})/t(\mathbf{x},\mathbf{p}) \end{bmatrix} \tag{48}$$

In this case, the Jacobian matrix is:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial w_x}{\partial p_1}, \frac{\partial w_x}{\partial p_2}, \frac{\partial w_x}{\partial p_3}, \frac{\partial w_x}{\partial p_4}, \frac{\partial w_x}{\partial p_5}, \frac{\partial w_x}{\partial p_6}, \frac{\partial w_x}{\partial p_7}, \frac{\partial w_x}{\partial p_8} \\ \frac{\partial w_y}{\partial p_1}, \frac{\partial w_y}{\partial p_2}, \frac{\partial w_y}{\partial p_3}, \frac{\partial w_y}{\partial p_4}, \frac{\partial w_y}{\partial p_5}, \frac{\partial w_y}{\partial p_6}, \frac{\partial w_y}{\partial p_7}, \frac{\partial w_y}{\partial p_8} \end{bmatrix} = \begin{bmatrix} x/t & 0 & -rx/t^2 & y/t & 0 & -ry/t^2 & 1/t & 0 \\ 0 & x/t & -sx/t^2 & 0 & y/t & -sy/t^2 & 0 & 1/t \end{bmatrix} \tag{49}$$

Considering that $r(\mathbf{x},\mathbf{0}) = x$, $s(\mathbf{x},\mathbf{0}) = y$ and $t(\mathbf{x},\mathbf{0}) = 1$, the value of the Jacobian at $\mathbf{p} = \mathbf{0}$ is:

$$\left.\frac{\partial \mathbf{w}}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}} = \begin{bmatrix} x & 0 & -x^2 & y & 0 & -xy & 1 & 0 \\ 0 & x & -xy & 0 & y & -y^2 & 0 & 1 \end{bmatrix} \tag{50}$$

Thus, $\frac{\partial \mathbf{w}}{\partial \mathbf{p}}^*$ and $\mathbf{C}^*(\lambda)$ matrices are:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{p}}^* = \left[x\mathbf{I}_{2m'\times 2m'}, \begin{bmatrix} -x^2\mathbf{I}_{m'\times m'} \\ -xy\mathbf{I}_{m'\times m'} \end{bmatrix}, y\mathbf{I}_{2m'\times 2m'}, \begin{bmatrix} -xy\mathbf{I}_{m'\times m'} \\ -y^2\mathbf{I}_{m'\times m'} \end{bmatrix}, \mathbf{I}_{2m'\times 2m'} \right] \tag{51}$$

$$\mathbf{C}^*(\lambda) = \mathbf{I}_{4x4} \otimes \mathbf{C}(\lambda) \tag{52}$$

with $m' = m + 1$.

Acknowledgments

This work was partially supported by the Ministry of Education and Science (CICYT) of Spain under contract TIN2006-01078 and Junta de Andalucia under contract TIC-02800. Thanks to the Perception Lab at the University of Texas at Dallas and its supervisors A.J. O'Toole and H. Abdi for kindly providing their "Database of Moving Faces and People".

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, 2004. 2, 5

[2] M. Black and A. Jepson. Eigen-tracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):6384, 1998. 1

[3] J. Buenaposada, E. Muoz, and L. Baumela. Efficient illumination independent appearance-based face tracking. *Image and Vision Computing, in press*, 2008. 1, 2

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1

[5] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(12):1080–1093, 2005. 1, 2

[6] R. Gross, I. Matthews, and S. Baker. Active appearance models with occlusion. *Image and Vision Computing*, 24(6):593–604, 2006. 5

[7] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 10251039, 1998. 1, 2

[8] B. Lucas and T. Kanade. An iterative image registration technique with application to stereo vision. In *IJCAI81*, page 674679, 1981. 2

[9] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004. 1, 2

[10] H. Mercier, J. Peyras, and P. Dalle. Toward an efficient and accurate AAM fitting on appearance varying faces. In *International Conference on Automatic Face and Gesture Recognition*, pages 363– 368, 2006. 2

[11] A. O'Toole, J. Harms, S. Snow, D. Hurst, M. Pappas, and H. Abdi. A video database of moving faces and people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):812–816. 5

[12] M. Turk. *Eigenfaces and Beyond*, chapter in Face Processing: Advanced Modeling and Methods. Academic Press, 2005. 1

[13] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition*, pages 586–591, 1991. 1