Proc. of ICCV 2005, pp. 877-882, Beijing, China, October 2005.

Efficient model-based 3D tracking of deformable objects

Enrique Muñoz†, José M. Buenaposada‡, Luis Baumela†

† Facultad de Informática, Univ. Politécnica Madrid ‡ ESCET, Univ. Rey Juan Carlos
Campus Montegancedo s/n, 28660 Madrid, Spain c/Tulipán, 28933 Móstoles, Spain
{kike,jmbuena}@dia.fi.upm.es, lbaumela@fi.upm.es

Abstract

Efficient incremental image alignment is a topic of renewed interest in the computer vision community because of its applications in model fitting and model-based object tracking. Successful compositional procedures for aligning 2D and 3D models under weak-perspective imaging conditions have already been proposed. Here we present a mixed compositional and additive algorithm which is applicable to the full projective camera case.

1. Introduction

Tracking non-rigid objects, and in particular human heads, is a topic of intense research within the computer vision community for its application to the construction of advanced computer interfaces and to achieving graphical models with realistic animation. Early approaches modelled the face as a rigid 3D textured object and tracked it by using corner features [6] or by using a model of face texture mapped onto planar [7], ellipsoidal [2] or cylindrical [9] 3D models. More recently, *generative linear models* of face appearance such as 2D *Active Appearance Models* (AAMs) [11] or 3D *Morphable Models* (MMs) [4, 13] have been successfully used respectively for real-time tracking and accurate modelling of human faces across changes in facial expressions and scene illumination.

Fitting a generative linear model to an image is a non-linear optimisation problem successfully solved by incrementally aligning the model with the target image. Two efficient minimisation procedures have been proposed in the literature which can be used for real-time tracking: the factorisation-based *additive* approach of Hager and Belhumeur [7] and the *Inverse Compositional Image Alignment Algorithm* (ICIA) of Baker and Matthews [1]. Both approaches have its drawbacks. Hager and Belhumeur's requires the Jacobian matrix to be factored. This is possible for appearance-based affine and projective planar models [5], but still has to be investigated whether it is applicable to the more sophisticated generative linear models. Baker and Matthews' approach requires the warping function to be closed under inverse composition, something which does not hold for AAMs or MMs.

By using an approximation to the composition of AAMs, Matthews and Baker [11] have recently used ICIA in a real-time algorithm for tracking faces using AAMs. One limitation of this approach is that AAMs are intrinsically 2D models and, although they can be used to track a 3D object, this is achieved at the expense of requiring more shape parameters. In consequence, the minimisation must be properly constrained in order to achieve a robust tracker [16]. Romdhani and Vetter [13] also used ICIA for efficiently adjusting a 3D MM to the image of a static face (a problem similar to tracking). An important drawback of both approaches is that they work under weak-perspective imaging conditions. This is a limitation if, for example, we would like to track a face imaged by a camera with short focal length and strong perspective distortion (e.g a low-cost web-cam).

In this paper we present an efficient incremental image alignment procedure for non-rigid 3D object tracking, based on a generative linear model of object appearance. By separating image projection from target motion we introduce a simple non-rigid motion model in which rigid and non-rigid motion parameters are easily decoupled, independently of which camera projection model is used. This enables us to write an exact inverse composition function. We demonstrate our technique by tracking synthetic and real image sequences using a human head as target.

The main contributions of this paper are:

a) Our tracker is independent of the camera model (in our experiments we use a full projective camera).

b) We use an exact inverse composition function, contrary to some previous approximations [11].

c) Rigid and non-rigid motion parameters are easily decoupled (this is an important issue in terms of computational efficiency).

2. The model

Our goal is to use a simple target model which can be easily acquired and which is suitable for tracking an arbitrary non-rigid object, in our experiments a human head. In order to achieve this goal we will use as model a set of images of the target and a sparse representation composed of a set of small planar textured patches, a set of shape bases which encode the modes of deformation and a set of texture bases which represent variations in the brightness caused by changes in the illumination of the scene (see Fig. 1).

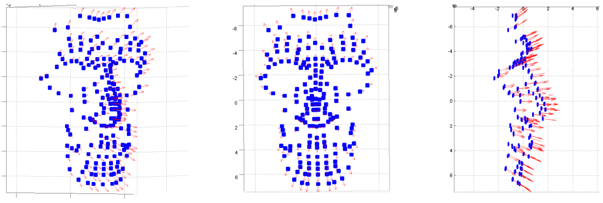


Figure 1. Our model for a human face.

2.1. The patches

Each patch of our model is tangent to the 3D volume of the object at the patch centre. The texture of the patch is the result of orthogonally projecting the underlying object texture onto a small plane. Our patches are similar to the “hyperpatches” of Wiles et al. [15]. The main difference being that hyperpatches are related to corner-like regions on the face, since they are individually searched for and registered between frames. Our patches are not necessarily attached to corner-like features, since we track them globally and the aperture problem applies to the set of all patches. In the case of a human face, texture patches are distributed over the face (see Fig. 1).

2.2. Motion model

The 3D motion of a point is the composition of a rigid motion caused by the translation and rotation of the object in space and a non-rigid motion caused by the deformation of the object. Let $\mathbf{X}_i = (x_i, y_i, z_i)^\top$ denote the co-ordinates of a point in 3D space and let $\mathbf{S} = (\mathbf{X}_1^\top, \mathbf{X}_2^\top, \ldots, \mathbf{X}_N^\top)^\top$ be the 3D structure represented by a set of N points in space.

Non-rigid motion.

The non-rigid motion of point $\mathbf{X}_i$ can be described as a linear combination of k_s basis points, b^s_{ij}, plus a mean component: $\mathbf{X}'_i = \mathbf{X}_{0_i} + \sum_{j=1}^{k_s} c^s_j b^s_{ij}$, $\mathbf{X}'_i, \mathbf{X}_{0_i} \in \Re^{3\times1}$, $c^s_j, b^s_{ij} \in \Re$, being c^s_j the weight of the linear combination.

Then, the shape of any configuration of the non-rigid object is expressed as a linear combination of a set of k_s basis shapes stored in matrix $\mathtt{B}^s$ plus a mean vector $\mathbf{S}_0$: $\mathbf{S} = \mathbf{S}_0 + \mathtt{B}^s\mathbf{c}^s$, $\mathbf{S}, \mathbf{S}_0 \in \Re^{3N\times1}$, $\mathtt{B}^s \in \Re^{3N\times k_s}$, $\mathbf{c}^s \in \Re^{k_s\times1}$, where $\mathbf{c}^s = (c^s_1, c^s_2, \ldots, c^s_{k_s})^\top$ is the vector of shape configuration weights. The mean vector $\mathbf{S}_0$, also called *rigid component*, represents the rigid configuration of the object, and the basis $\mathtt{B}^s$ represents the allowed *modes of deformation*.

Rigid motion.

The 3D shape can rotate and translate rigidly in space. Let $\mathtt{R}(\alpha, \beta, \gamma) \in \Re^{3\times3}$ and $\mathbf{t} \in \Re^{3\times1}$ be the rotation matrix and translation vector representing such motion. Then, the rigid motion of point $\mathbf{X}_i$ would be given by $\mathbf{X}'_i = \mathtt{R}\mathbf{X}_i + \mathbf{t}$. We will denote $\mathbf{S}' = \mathtt{R}\mathbf{S} + \mathbf{t}$ the result of applying rotation $\mathtt{R}$ and translation $\mathbf{t}$ to each point of the 3D shape $\mathbf{S}$, producing a new shape $\mathbf{S}'$.

Motion model

Any configuration of the object in 3D space, $\mathbf{S}$, can be generated with a motion model, f, which moves and deforms the average shape

$$\mathbf{S} = f(\mathbf{S}_0, \boldsymbol{\mu}) = \mathtt{R}(\mathbf{S}_0 + \mathtt{B}^s\mathbf{c}^s) + \mathbf{t}, \tag{1}$$

where $\boldsymbol{\mu} = (\alpha, \beta, \gamma, t_x, t_y, t_z, (\mathbf{c}^s)^\top)^\top$ is the vector of motion parameters. Note that f is such that $f(\mathbf{S}, \mathbf{0}) = \mathbf{S}$. Conversely, the average shape can be reached from any object configuration via $\mathbf{S}_0 = f^{-1}(\mathbf{S}, \boldsymbol{\mu}) = \mathtt{R}^\top(\mathbf{S} - \mathbf{t}) - \mathtt{B}^s\mathbf{c}^s$.

2.3. Shape projection

The projection of point $\mathbf{X}_i$ onto an image is represented by $\mathbf{x}_i = p(\mathbf{X}_i, \mathbf{q}) \in \Re^{2\times1}$, where $\mathbf{q}$ is the vector of projection parameters. Similarly, the 3D object shape $\mathbf{S}$ projected onto a 2D image is denoted $\mathbf{s} = p(\mathbf{S}, \mathbf{q}) \in \Re^{2N\times1}$. Here we make no assumption as to which projection model is used, although in our experiments we will assume a projective camera.

In previous approaches the motion model also included implicitly [7, 11] or explicitly [13] the projection of the point onto the image plane. In general, this is not a good choice since it complicates unnecessarily the computation of the inverse shape $f^{-1}(\mathbf{S}, \boldsymbol{\mu})$ (e.g. see Sec. 4 in [11]) and prevents $f \circ f^{-1}$ from being closed. This is why an approximated inverse composition has to be used in [11]. Another collateral advantage of having a simpler motion model is that rigid and non-rigid motion parameters are decoupled and can be easily identified.

2.4. Texture model

Let us denote $I[p(\mathbf{X}_i, \mathbf{q})]$ the brightness value (or RGB values) assigned to the projection of point $\mathbf{X}_i$ onto image

$I(\mathbf{x})$. It depends on the object colour, the colour and intensity of the illumination source and the relative orientation between source and object surface at $\mathbf{X}_i$ [3]. These factors can be modelled by

$$I[p(\mathbf{X}_i, \mathbf{q})] = T[p(\mathbf{X}_i, \mathbf{q})] + \sum_{j=1}^{k_t} b_{ij}^t c_j^t, \quad b_{ij}^t, c_j^t \in \Re,$$

where $\mathbf{c}^t = (c_1^t, c_2^t, \ldots, c_{k_t}^t)^\top$ is the vector of *texture configuration weights*, b_{ij}^t is the j-th component of the texture base associated with 3D point $\mathbf{X}_i$ and $T[p(\mathbf{X}_i)]$ is the *average texture* for that point. The texture base models changes in the brightness of a pixel caused by the illumination of the scene.

The texture model for a deformable object represented by structure vector $\mathbf{S}$ is $\mathbf{I}[p(\mathbf{S}, \mathbf{q})] = \mathbf{T}[p(\mathbf{S}, \mathbf{q})] + \mathtt{B}^t \mathbf{c}^t$, $\mathbf{I}, \mathbf{T} \in \Re^{N \times 1}$, $\mathbf{c}^t \in \Re^{k_t \times 1}$, where $\mathtt{B}^t \in \Re^{N \times k_t}$ is the matrix storing the texture basis shapes and $\mathbf{c}^t = (c_1^t, c_2^t, \ldots, c_{k_s}^t)^\top$ is the vector of texture configuration weights. Here we assume a gray level image, a similar model could be built for RGB colour values [13].

In general the projected point $p(\mathbf{X}_i, \mathbf{q})$ may not coincide with an integer position in $I(\mathbf{x})$. In this case the brightness value $I[p(\mathbf{X}_i, \mathbf{q})]$ is computed through interpolation from neighbouring pixels.

The tracking procedure described in the following section is based on a constancy constraint on the brightness values normalised with respect to the illumination. We define the average texture of a point to be its *normalised brightness*, $N(I[p(\mathbf{X}_i, \mathbf{q})], \mathbf{c}^t) = T[p(\mathbf{X}_i, \mathbf{q})]$, and the *normalised texture* for an object configuration

$$\mathbf{N}(\mathbf{I}[p(\mathbf{S}, \mathbf{q})], \mathbf{c}^t) = \mathbf{T}[p(\mathbf{S}, \mathbf{q})] = \mathbf{I}[p(\mathbf{S}, \mathbf{q})] - \mathtt{B}^t \mathbf{c}^t. \quad (2)$$

3. Tracking

In this section we describe an efficient procedure for tracking a non-rigid object through an image sequence using the object model presented in section 2. First we introduce the brightness constancy constraint and pose the tracking problem as a parametric minimisation based on such constraint. We then show how a mixed compositional and additive algorithm can be used for efficiently computing the best set of parameters.

3.1. Problem statement

Let $\mathbf{S}_0$ be the rigid component of a deformable object, $\boldsymbol{\mu}_t$ be the set of parameters that aligns $\mathbf{S}_0$ with the image acquired at time t, $I[\mathbf{x}, t]$, and $\mathbf{c}_t^t$ be the texture configuration weights which normalise the brightness values of $I[\mathbf{x}, t]$. Then, for any time instants t_0 and t, the following *brightness constancy* equation holds

$$\mathbf{N}(\mathbf{I}[p(f(\mathbf{S}_0, \boldsymbol{\mu}_t), \mathbf{q}), t], \mathbf{c}_t^t) = \\ \mathbf{N}(\mathbf{I}[p(f(\mathbf{S}_0, \boldsymbol{\mu}_{t_0}), \mathbf{q}_0), t_0], \mathbf{c}_{t_0}^t), \quad (3)$$

which is a generalisation of the so-called image constancy assumption [7, 8].

Let us assume that $I[\mathbf{x}, t_0]$ is a fixed *reference image* that we will denote $I_r(\mathbf{x})$, and that $I[\mathbf{x}, t]$ is the *target image* which varies over time as the object moves and deforms. We will also assume that the motion model parameters are related to our target object in such a way that $\boldsymbol{\mu}_{t_0} = \mathbf{0}$.

Tracking amounts to finding, for each time instant t, the set of parameters $\boldsymbol{\mu}_t$ and $\mathbf{c}_t^t$ for which equation (3) holds. This can be achieved by solving the following least squares problem[1]

$$\min_{\boldsymbol{\mu}_t, \mathbf{c}_t^t} ||\mathbf{N}(\mathbf{I}[p(f(\mathbf{S}_0, \boldsymbol{\mu}_t), \mathbf{q}), t], \mathbf{c}_t^t) - \\ \mathbf{N}(\mathbf{I}_r[p(\mathbf{S}_0, \mathbf{q}_r)], \mathbf{c}_r^t)||^2. \quad (4)$$

This a complex minimisation problem since the cost function is non-convex. Similar problems have been traditionally solved linearly by estimating the model parameters incrementally. We can achieve this by making a Taylor series expansion of (4) and computing the increment in the motion parameters by Gauss-Newton iterations. Different solutions have been proposed in the literature depending on which term of (4) the Taylor expansion is made on and how the motion parameters are updated [10, 7, 14, 1].

3.2. Efficient tracking

The computational cost of tracking with this approach is due mainly to the cost of estimating the Jacobian of the image brightness values w.r.t. the motion model's parameters and its pseudo-inverse, which are needed to make the Gauss-Newton iterations. The factorisation-based additive approach of Hager and Belhumeur [7] and the compositional approach of Baker and Matthews [1] are two efficient solutions for similar problems. Here we introduce an efficient minimisation procedure which uses a compositional approach for estimating the motion parameters and an additive one for the texture configuration weights.

The minimisation solved for tracking is the following

$$\min_{\delta\boldsymbol{\mu}, \delta\mathbf{c}^t} ||\mathbf{N}(\mathbf{I}[p(f(\mathbf{S}_0, \boldsymbol{\mu}_t), \mathbf{q}), t + \delta t], \mathbf{c}_t^t) - \\ \mathbf{N}(\mathbf{I}_r[p(f(\mathbf{S}_0, \delta\boldsymbol{\mu}), \mathbf{q}_r)], \mathbf{c}_r^t + \delta\mathbf{c}^t)||^2, \quad (5)$$

where the first term represents the normalised brightness values obtained when projecting the configuration of the

[1] In general, several reference images may be used, the only requirement being that all of them represent the same non-rigid deformation.

object at time t onto the image acquired at time $t + \delta t$. The second term is the incremental non-rigid motion and the changes in texture that must take place so that the same set of normalised brightness values in the first term are obtained from the reference image. Parameters $\delta\boldsymbol{\mu}$ and $\delta\mathbf{c}^t$ represent respectively the motion and deformation of the target object between time instants t and $t + \delta t$, and the changes in texture caused by the illumination.

Estimating $\delta\boldsymbol{\mu}$ and $\delta\mathbf{c}^t$

The increment in motion and texture parameters can be linearly estimated by making a Taylor series expansion of the second term in (5)

$$\mathbf{N}(\mathbf{I}_r[p(f(\mathbf{S}_0, \delta\boldsymbol{\mu}), \mathbf{q}_r)], \mathbf{c}_r^t + \delta\mathbf{c}) = \mathbf{I}_r[p(f(\mathbf{S}_0, \mathbf{0}), \mathbf{q}_r)] - \mathtt{B}^t\mathbf{c}_r^t + \mathtt{M}_{\boldsymbol{\mu}}\delta\boldsymbol{\mu} - \mathtt{M}_{\mathbf{c}^t}\delta\mathbf{c}^t, \quad (6)$$

where

$$\mathtt{M}_{\boldsymbol{\mu}} = \left.\frac{\partial\mathbf{I}_r[p(f(\mathbf{S}_0, \boldsymbol{\mu}), \mathbf{q}_r)]}{\partial\boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\mathbf{0}}, \mathtt{M}_{\mathbf{c}^t} = \left.\frac{\partial\mathtt{B}^t\mathbf{c}}{\partial\mathbf{c}}\right|_{\mathbf{c}=\mathbf{c}_r^t} = \mathtt{B}^t.$$

Then, from (6), minimisation (5) can be rewritten as

$$\min_{\delta\boldsymbol{\mu}, \delta\mathbf{c}^t} ||\mathcal{E}(t + \delta t) - \mathtt{M}_{\boldsymbol{\mu}}\delta\boldsymbol{\mu} + \mathtt{B}^t\delta\mathbf{c}^t||^2,$$

which can be solved by least squares

$$\begin{bmatrix} \delta\boldsymbol{\mu} \\ \delta\mathbf{c}^t \end{bmatrix} = (\mathtt{M}_0^\top\mathtt{M}_0)^{-1}\mathtt{M}_0^\top\mathcal{E}(t + \delta t)$$

where $\mathcal{E}(t + \delta t) = \mathbf{I}[p(f(\mathbf{S}_0, \boldsymbol{\mu}_t), \mathbf{q}), t + \delta t] - \mathbf{I}_r[p(\mathbf{S}_0, \mathbf{q}_r)] - \mathtt{B}^t(\mathbf{c}_t^t - \mathbf{c}_r^t)$ is the error made when projecting the configuration at time t onto the image acquired at $t + \delta t$ and $\mathtt{M}_0 = [\,\mathtt{M}_{\boldsymbol{\mu}}\,|-\mathtt{B}^t\,]$ is the Jacobian of the reference image with respect to the motion and texture parameters. Note that $\mathtt{M}_0$ is constant and its inverse can be precomputed off-line. This is the key for the efficiency of this algorithm.

In [1] this minimisation is performed by making the columns of $\mathtt{B}^s$ orthogonal to those of $\mathtt{B}^t$. This has been reported in [13] to introduce perturbations in $\mathtt{B}^s$ which decrease the accuracy of the shape recovery. Instead, here we explicitly solve for both sets of parameters.

The Jacobian matrix $\mathtt{M}_0$ models how the brightness of each $\mathbf{X}_i$ changes as the target moves infinitesimally. It represents the information provided by each point to the tracking process. When $\mathtt{M}_0^\top\mathtt{M}_0$ is singular, the motion cannot be recovered. This would be a generalisation of the so called *aperture problem* in the estimation of optical flow. This is also the reason why we can track an object with low-textured patches (non corner-like), because each patch contributes to the minimisation and the aperture problem applies to the set all of them.

Estimating $\boldsymbol{\mu}_{t+\delta t}$ and $\mathbf{c}_{t+\delta t}^t$

From (2) and introducing the change of variable $\mathbf{S}_0' = f(\mathbf{S}_0, \delta\boldsymbol{\mu})$, (5) can be rewritten

$$\min_{\delta\boldsymbol{\mu}, \delta\mathbf{c}^t} ||\mathbf{I}[p(f(f^{-1}(\mathbf{S}_0', \delta\boldsymbol{\mu}), \boldsymbol{\mu}_t), \mathbf{q}), t + \delta t] - \mathtt{B}^t(\mathbf{c}_t^t - \delta\mathbf{c}^t) - (\mathbf{I}_r[p(\mathbf{S}_0', \mathbf{q}_r)] - \mathtt{B}^t\mathbf{c}_r^t)||^2. \quad (7)$$

Following ICIA convention [1] and comparing (4) and (7) we can conclude that $\mathbf{c}_{t+\delta t}^t = \mathbf{c}_t^t - \delta\mathbf{c}^t$ and $f(\mathbf{S}_0', \boldsymbol{\mu}_{t+\delta t}) = f(f^{-1}(\mathbf{S}_0', \delta\boldsymbol{\mu}), \boldsymbol{\mu}_t)$. For our 3D model $f(f^{-1}(\mathbf{S}_0, \delta\boldsymbol{\mu}), \boldsymbol{\mu}_t)$ is an approximation to $f(\mathbf{S}_0, \boldsymbol{\mu}_{t+\delta t})$, but a strict equality for a 2D model.

In order to obtain $\boldsymbol{\mu}_{t+\delta t}$ we expand

$$f(f^{-1}(\mathbf{S}_0', \delta\boldsymbol{\mu}), \boldsymbol{\mu}_t) = \mathtt{R}_t\delta\mathtt{R}^\top(\mathbf{S}_0' + \delta\mathtt{R}\mathtt{B}^s(\mathbf{c}_t^s - \delta\mathbf{c}^s)) + \mathbf{t}_t - \mathtt{R}_t\delta\mathtt{R}^\top\delta\mathbf{t}, \quad (8)$$

and again comparing (1) with (8) we can conclude that $\mathtt{R}_{t+\delta t} = \mathtt{R}_t\delta\mathtt{R}^\top$, $\mathbf{t}_{t+\delta t} = \mathbf{t}_t - \mathtt{R}_t\delta\mathtt{R}^\top\delta\mathbf{t}$ and $\mathbf{c}_{t+\delta t}^s = \mathbf{c}_t^s - \delta\mathbf{c}^s$. Note that as $\mathbf{S}_0'$ is rotated by $\delta\mathtt{R}$ from $\mathbf{S}_0$, then $\mathtt{B}^s$ must also be corrected to $\delta\mathtt{R}\mathtt{B}^s$.

Previously, decoupling rigid and non-rigid motion parameters in the motion model was only possible for a weak-perspective camera model and required a complex procedure [11, 13].

The final algorithm is as follows:

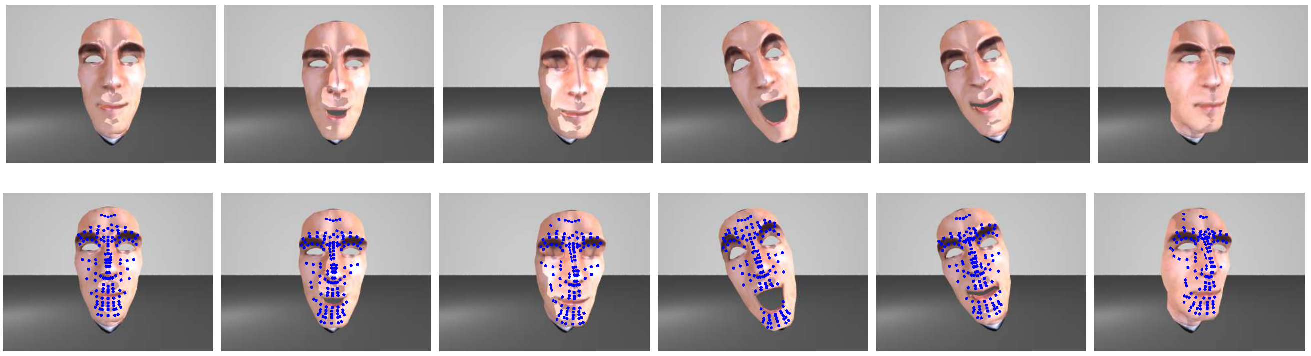
- **Off-line:**
  1. Compute $\mathtt{M}_0$.
  2. Compute and store $\mathtt{M}^+ = (\mathtt{M}_0^\top\mathtt{M}_0)^{-1}\mathtt{M}_0^\top$.
  3. Compute and store $\mathbf{i}_r = \mathbf{I}_r[p(\mathbf{S}_0, \mathbf{q}_r)]$
- **Online:**
  1. $\mathcal{E} = \mathbf{I}[p(f(\mathbf{S}_0, \boldsymbol{\mu}_t), \mathbf{q}), t + \delta t] - \mathbf{i}_r - \mathtt{B}^t(\mathbf{c}_t^t - \mathbf{c}_r^t)$.
  2. Compute $[\delta\boldsymbol{\mu}, \delta\mathbf{c}]^\top = \mathtt{M}^+\mathcal{E}$.
  3. Update $\mathbf{c}_{t+\delta t}^t = \mathbf{c}_t^t - \delta\mathbf{c}^t$.
  4. Update $\mathtt{R}_{t+\delta t} = \mathtt{R}_t\delta\mathtt{R}^\top$, $\mathbf{t}_{t+\delta t} = \mathbf{t}_t - \mathtt{R}_t\delta\mathtt{R}^\top\delta\mathbf{t}$.
  5. Update $\mathbf{c}_{t+\delta t}^s = \mathbf{c}_t^s - \delta\mathbf{c}^s$.

4. Experiments

In order to evaluate our algorithm empirically, we have set up experiments with synthetic and real image sequences. Synthetic experiments aim to validate the theoretical basis of the algorithm and real ones intend to demonstrate the suitability of our approximation for tracking live sequences.

4.1. Synthetic experiments

We have developed a framework for creating synthetic sequences of a deforming head model. The head model is based on a previous work by Parke et. al. [12] which includes 512 vertices and encodes 18 different muscles of the face. We generate facial expressions by actuating on the



Frame 1 Frame 19 Frame 79 Frame 160 Frame 167 Frame 259

Figure 2. Synthetic sequence key-frames and tracking results.

different facial muscles. A rigid body transformation (orientation change plus translation) to the computed model determines head pose and orientation. Then we map a photo-realistic texture of a face onto the model and project both onto the image using a free ray-tracing tool[2]. The ray-tracer simulates a projective camera located at 20 units of distance from the head model, which is has a depth of 5 units. Fig. 2 shows several key frames of a 300 frames synthetic sequence. From its starting position, the head translates along the horizontal image axis while rotating around its three main axis. The sequence comprises a total of ten facial expressions which includes mouth opening, eyebrows raising, frowning, etc. To the left of the scene we have placed a light source, pointing directly towards the head, and we have assumed the head surface to be Lambertian.

We obtained our basis shapes from a 750 frames sequence which comprised all the possible facial expressions for our model. We place our patches on 194 polygon vertices distributed over the face. By performing PCA on the matrix which stores the tracks of all patches across the sequence we we obtain the modes of deformation. We used five modes of deformation which encoded 98% of the variance in the data. By orbiting the light source around the head model in neutral position we obtained a 200 image sequence representing different lighting conditions. We obtained the texture basis $\mathtt{B}^t$ by performing PCA on the matrix storing the brightness values of the projections of our head model onto each image.

In Fig. 2 we show some results from the 300 frames synthetic sequence. Figure 3 shows some of the computed parameters plotted against their ground truth values. These ground truth values are the ones used to create the synthetic sequence. Estimated values from the tracking algorithm for the rotation around the horizontal axis, α, translation along the horizontal axis, t_x , and the first linear coefficients for both the shape and texture deformations, c_1^s and c_1^t, are plotted along with their ground truth. Results show that both motion and texture parameters are accurately estimated even when there are quite noticeable changes in illumination and facial expressions.

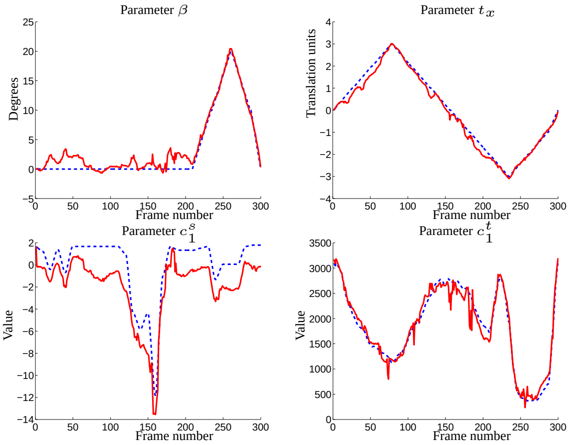


Figure 3. Estimated vs. ground truth values. First row, rotation around head vertical axis (left) and horizontal translation (right). Second row, fist shape configuration weight (left) and first texture configuration weight (right). Red continuous line stands for estimated values for each frame whereas blue dashed line stands for ground truth data.

4.2. Real experiments

We have also some preliminary results for a 20 seconds real video sequence. We imaged an actor performing several expressions (anger, sadness and surprise) with a calibrated Basler A102fc colour camera located roughly 1 meter away from the actor.

[2] See http://www.povray.org

In a video sequence different from the previous one and using a VICON motion capture system we tracked 39 markers on the actor's face. The motion of a total of 121 patches was interpolated from the 3D tracks of the markers and stored in a motion matrix. We obtained the shape basis for the actor's head by performing PCA on the motion matrix.

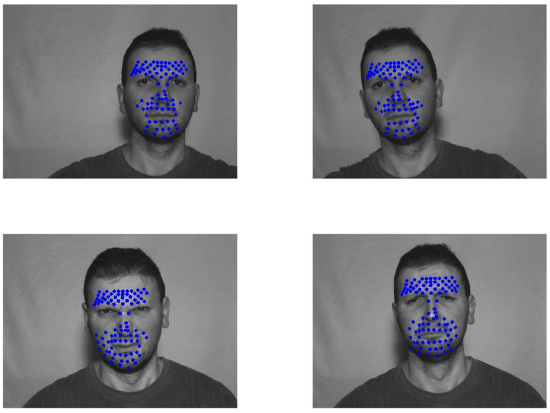


Figure 4. Real sequence key frames and tracking results.

We show in Fig 4 some key frames of the real video sequence with the estimated location of the patches overlayed on it. In spite of the sparseness and low quality of the model the tracker performs well.

5. Conclusions

We have presented a new formulation of an efficient image alignment algorithm for non-rigid 3D generative linear models of object appearance. Separating projection and motion models enables us to:

a) Build a tracker independent of the image projection model. We have shown that it performs correctly for sequences captured under projective imaging conditions.

b) Introduce a simple deformable motion model in which the inverse shape composition can be exactly computed.

c) Directly identify rigid and non-rigid motion parameters.

Also, the whole tracker itself is interesting in its own right given its theoretical simplicity and ease of programming.

Although in our experiments we have used a sparse patch-based model of target appearance, the algorithm is applicable to any generative linear models such as AAMs or MMs.

Acknowledgements

The authors gratefully acknowledge funding from the Spanish Ministry of Science and Technology under grant TIC2002-00591. Enrique Muñoz was funded by a FPU grant from the Ministry of Education. They are also grateful to Lourdes Agapito and Alessio del Bue for helpful discussions and for providing the 3D model used in the real video sequences.

References

[1] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *Proc. of CVPR*, volume 1, pages 1090–1097. IEEE, 2001.

[2] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. In *Proc. of ICPR*, 1996.

[3] P. Belhumeur and D. J. Kriegman. What is the set of images of an object under all possible illumination conditions? *IJCV*, 28(3):245–260, 1998.

[4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proc. of SIGGRAPH*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[5] J. M. Buenaposada, E. Muñoz, and L. Baumela. Efficient appearance-based tracking. In *Proc. CVPR-Workshop on Nonrigid and Articulated Motion*, volume 1. IEEE, June 2004.

[6] A. Gee and R. Cipolla. Fast visual tracking by temporal consensus. *Image and Vision Computing*, 14(2):105–114, 1996.

[7] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, 1998.

[8] B. Horn. *Computer Vision*. MIT-Press, Cambridge, Mass, 1986.

[9] M. La Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on robust registration of texture-mapped 3d models. *PAMI*, 22(4), April 2000.

[10] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of Image Understanding Workshop*, pages 121–130, 1981.

[11] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.

[12] F. I. Parke and K. Waters. *Computer Facial Animation*. AK Peters Ltd, 1996.

[13] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *Proc. of ICCV*, volume 1, pages 59–66. IEEE, 2003.

[14] H.-Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *IJCV*, 36(2):101–130, 2000.

[15] C. S. Wiles, A. Maki, and N. Matsuda. Hyperpatches for 3d model acquisition and tracking. *PAMI*, 23(12):1391–1403, 2001.

[16] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *Proc. of CVPR*, Washington, D.C., June 2004. IEEE.