Image and Vision Computing xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Efficient illumination independent appearance-based face tracking

José M. Buenaposada [a], Enrique Muñoz [b,1], Luis Baumela [b,*]

[a] Departamento de Ciencias de la Computación, ETSI Informática, Universidad Rey Juan Carlos, Spain
[b] Departamento de Inteligencia Artificial, Facultad Informática, Universidad Politécnica de Madrid, Spain

ARTICLE INFO

Article history:
Received 26 January 2007
Received in revised form 11 September 2007
Accepted 24 April 2008
Available online xxxx

Keywords:
Linear models of appearance
Illumination invariance
Efficient linear subspace model fitting
Facial expression analysis

ABSTRACT

One of the major challenges that visual tracking algorithms face nowadays is being able to cope with changes in the appearance of the target during tracking. Linear subspace models have been extensively studied and are possibly the most popular way of modelling target appearance. We introduce a linear subspace representation in which the appearance of a face is represented by the addition of two approximately independent linear subspaces modelling facial expressions and illumination, respectively. This model is more compact than previous bilinear or multilinear approaches. The independence assumption notably simplifies system training. We only require two image sequences. One facial expression is subject to all possible illuminations in one sequence and the face adopts all facial expressions under one particular illumination in the other. This simple model enables us to train the system with no manual intervention. We also revisit the problem of efficiently fitting a linear subspace-based model to a target image and introduce an additive procedure for solving this problem. We prove that Matthews and Baker's inverse compositional approach makes a smoothness assumption on the subspace basis that is equivalent to Hager and Belhumeur's, which worsens convergence. Our approach differs from Hager and Belhumeur's additive and Matthews and Baker's compositional approaches in that we make no smoothness assumptions on the subspace basis. In the experiments conducted we show that the model introduced accurately represents the appearance variations caused by illumination changes and facial expressions. We also verify experimentally that our fitting procedure is more accurate and has better convergence rate than the other related approaches, albeit at the expense of a slight increase in computational cost. Our approach can be used for tracking a human face at standard video frame rates on an average personal computer.

1. Introduction

Much effort is being devoted nowadays to developing machines that can recognise people and interpret their actions, gestures and facial expressions. Determining a person's facial expression or in which direction a person is looking is extremely important for developing advanced human computer interfaces that are aware of people and their emotions. It also plays an important role in other computer vision applications, such as lip reading, graphical animation, surveillance or video-based face recognition.

Tracking a human face is a challenging problem because the face is a deformable low-textured object and because its visual appearance changes dramatically with identity and in the presence of occlusions, changes in the illumination or pose. One way to cope with this problem is by adopting a model-based image analysis approach. This approach uses a face model describing all possible facial configurations and the way the face interacts with scene illumination. The parameters that best fit the model to the target image describe scene illumination, the pose and facial expression.

Model-based computer vision brings about two other problems, namely, *model construction* and *model fitting*. We need a model that is accurate enough to precisely describe and separate all sources of variation in a face's appearance. To do this we can build off-line, and possibly manually, a model of the face that is so general as to represent any possible face under any imaging condition. The major drawback of this approach is that this model will require many degrees of freedom in order to be able to represent all possible variations in the appearance of any human face. Fitting this high dimensional model to a target image is a hard problem [1]. Complex minimisation algorithms have to be used in order to avoid local minima. Alternatively, we can build a simpler model valid only for the face identity and imaging conditions of a given task. In this case, it is easier to devise an efficient and robust model fitting procedure [1]. However, if the tracker is to be at all useful, a simple, and possibly automated, procedure for model construction is needed. In this paper, we introduce a real-time face tracking system based on the second approach.

* Corresponding author. Tel.: +34 913367440; fax: +34 913524819.
E-mail address: lbaumela@fi.upm.es (L. Baumela).
URL: http://www.dia.fi.upm.es/~pcr (L. Baumela).
[1] Present address: Facultad de Informática, Universidad Complutense de Madrid, Ciudad Universitaria s/n, 28040 Madrid, Spain.

0262-8856/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2008.04.015

Being able to identify and model some of the various sources of facial appearance variation is also a key issue for many applications. In an automated graphical animation system, for example, the tracker must estimate and separate appearance changes due to facial expressions from variations caused by illumination, so that these changes can be re-targeted in a graphical model. In this paper, we introduce a subspace representation of the appearance of a face, that models facial expressions and illumination variations. In this approach a face is represented by adding two linear subspaces. The first subspace models the deformations of the face caused facial by expressions. The second subspace represents variations in facial appearance caused by changes in the illumination. Both subspaces are independent of each other in the sense that they have different basis, which can be trained almost independently. With this model we will be able to train the tracker without any manual intervention.

Most applications not only require visual tracking algorithms to be robust to changes in the target appearance, but also to work in real-time. We introduce a minimisation procedure which can efficiently fit the previous appearance model to a target image. In the experiments conducted we show that our procedure is able to track a face at frequencies higher than video frame rates, running on an average personal computer.

In summary, one of the present challenges of face tracking is to develop easy-to-train, efficient and robust tracking algorithms that can identify the various sources of facial appearance variation. In this paper, we introduce a face tracking system that: (a) can be trained without any manual intervention; (b) identifies changes in facial appearance caused by facial expressions and by illumination changes; and (c) runs at frequencies higher than standard video frame rates.

1.1. Related work

Model-based face tracking is generally stated as a minimisation problem. The tracker tries to minimise the discrepancies between the model and the actual configuration of the face in each image of a sequence. Early approaches modelled the face as a rigid 3D-textured object and tracked it using corner features [2] or a global model of face texture mapped onto planar [3], ellipsoidal [4] or cylindrical [5] 3D models.

Rigid and non-rigid facial deformation was initially represented using physics-based anatomically motivated models [6] or local parametric motion models [7]. Later, changes in facial expressions were also modelled using image-based approaches and generative linear models of shape and texture. Image-based approaches model the visual appearance of a deforming face using linear subspace representations of facial appearance [8]. More recently, non-linear subspaces have also been used to model the appearance of a face across changes in pose, facial expression and illumination [9,10]. Generative linear models of shape and texture are based on the fact that the visual appearance of a deforming face can be modelled by combining two linear subspaces, one representing changes in shape (e.g. facial expressions, identity) and the other representing changes in texture caused, for example, by variations in identity or illumination [11–14]. They use a polygon mesh that can be deformed according to a statistically learned set of modes of deformation. Most prominent examples of this approach are Cootes, Edwards and Taylor's 2D *Active Appearance Models* [15,12] and Blanz and Vetter's 3D *Morphable Models* [11]. Anthropometric face models use a 3D mesh whose structure and modes of deformation have been constructed by hand from tables of anthropometric measurements of the face [16].

The main drawback of the approaches based on generative linear models of shape and texture is that they have complex training procedures that often require manual intervention [12,17,18]. This problem is even more critical in hand-made physics and anthropometric-based face models [6,16]. On the other hand, image-based approaches are gaining popularity, since there are various procedures for automatically learning linear [8,19,20] or non-linear [9] subspace models and for probabilistically representing the dynamics of appearance variation [21–24]. Unfortunately, automated procedures for learning image-based models [8,9,19] cannot automatically factor the various sources of appearance variation. In this paper, we will introduce a subspace representation of facial appearance that can be automatically trained and separates facial expressions from illumination variations.

Subspace [25] and geometrical [26,27] approaches have been used traditionally to separate illumination changes from other sources of variations in facial appearance to build face recognition systems. Subspace approaches have also been used to separate multiple orthogonal factors using bilinear [28,29] or multilinear [30] models. Tenenbaum and Freeman [28] used SVD to decompose face appearance into two factors: pose and identity, termed *content and style*. Grimes et al. [29] used expectation maximisation and particle filters to introduce a probabilistic approach in the bilinear model. More recently, procedures for separating *style and content* on a non-linear manifold have also been proposed [31]. Vasilescu and Terzopoulos used multilinear tensor analysis to decompose face images into orthogonal [30] and independent [32] factors representing identity, expression, pose and illumination. These approaches cannot be used in a real-time tracker, either because they were conceived to analyse a single image [26], for use in batch processing [28,30,32], or because of the computational requirements of the minimisation procedure [29,31].

In this paper, we show that the appearance of a deforming face perceived under varying illumination can be represented up to a first order approximation by adding two linear subspaces. The first subspace describes the deformations of the face caused by facial expressions and the second one models changes in the illumination. Both subspaces are independent of each other in the sense that they can be trained almost independently (see Section 5). This linear model will enable us to train the system without any manual intervention and to build a tracker that can factor changes in facial expressions from variations in illumination. Another remarkable property of this model is that it will require much fewer training images than earlier bilinear and multilinear approaches. Since deformation and illumination subspaces are assumed to be independent, we will not need training samples with all facial expressions under all possible illumination conditions, as was the case previously. We will only require two sets of sample images. In one set one facial expression is subject to all possible illuminations and in the other the face performs all facial expressions under one illumination. Consequently, the number of parameters and training images with this model will grow linearly with image size, whereas they would grow non-linearly in previous bilinear or multilinear approaches.

There are two basic approaches to linear subspace model registration. Approaches based on traditional optimisation techniques [3,33–35] and approaches based on particle filters [29,20,36]. Very efficient registration procedures have been introduced using Gauss–Newton iterations [3,34,35]. Unfortunately, depending on the structure of the target and the starting point, they may converge to a local minimum. To avoid this problem less efficient multi-scale minimisation must be used [33]. Particle filters, however, have recently emerged as a simple and robust method for model adjustment in the presence of non-normal measurement and/or dynamics. They do not get trapped in local minima, but their computational cost grows exponentially as the number of model parameters increase. Rao-Blackwellisation [37] is used in [36] to improve tracking efficiency by integrating out the coefficients of the linear subspace model.

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015

In Section 3 we introduce a minimisation procedure, based on Gauss–Newton iterations, which can efficiently fit a linear subspace model to a target image. It is directly related to Hager and Belhumeur's factorisation-based approach [3], whose tracking procedure is robust to changes in illumination, but assumes a rigid face. We have extended their approach to the case in which the target face can also deform. It is also related to the series of papers by Baker, Matthews et al. [38,35,34,39]. In Section 3, we revisit their inverse compositional image alignment (ICIA) algorithm introduced in [38,35] and prove that its subspace extension introduced in [34] is based on a smoothness assumption equivalent to the one used by Hager and Belhumeur in [3], which is not valid for a deforming face. In the experiments described in Section 6, we compare the fitting procedure introduced in this paper with those from [3,34] and show that the procedure introduced here has better convergence properties for difficult images.

Some preliminary results of this work appeared in [40,41].

2. The model

In this section, we introduce a new appearance-based model representing the variations in the appearance of a face caused by changes in facial expressions and in the illumination of the scene. It is based on a first order approximation to the face's reflectance function.

Let $\mathsf{I}(\mathbf{x},t)$ be the image acquired at time t, where $\mathbf{x}$ is a vector representing the coordinates of a point in the image, and let $\mathbf{I}(\mathbf{x},t)$ be a vector storing the brightness values of $\mathsf{I}(\mathbf{x},t)$. Let us assume that the target moves rigidly (with no deformation) between time instants $t_0=0$ and t, and that this motion can be described by the motion model $f(\mathbf{x},\boldsymbol{\mu})$, $\boldsymbol{\mu}$ being the vector of rigid motion parameters. If there are no changes in the target appearance caused by the scene illumination, the brightness constancy equation $\mathbf{I}(f(\mathbf{x},\boldsymbol{\mu}_t),t)=\mathbf{I}(\mathbf{x},0)$ holds. If the face is now allowed to deform non-rigidly, then we may write a new brightness constancy equation $\mathbf{I}(f(\mathbf{x},\boldsymbol{\mu}_t),t)=\mathbf{I}(\mathbf{x},0)+[\mathsf{B}_d\mathbf{c}_{d,t}](\mathbf{x})$, where the non-rigid deformations have been modelled by a linear subspace with basis B_d, mean value $\mathbf{I}(\mathbf{x},0)$ and linear deformation parameters $\mathbf{c}_{d,t}$. We denote the value of $\mathsf{B}_d\mathbf{c}_{d,t}$ for the pixel with position $\mathbf{x}$ by $[\mathsf{B}_d\mathbf{c}_{d,t}](\mathbf{x})$. Finally, for a given rigid motion $\boldsymbol{\mu}_t$ and deformation $\mathbf{c}_{d,t}$, we could also model the illumination of the face by including a new subspace with basis B_i and linear illumination parameters $\mathbf{c}_i$, which represents all the possible illuminations of the mean face $\mathbf{I}(\mathbf{x},0)$. So, the final brightness constancy equation is

$$\begin{aligned}\mathbf{I}(f(\mathbf{x},\boldsymbol{\mu}_t),t) &= \mathbf{I}(\mathbf{x},0)+[\mathsf{B}_i\mathbf{c}_{i,t}](\mathbf{x})+[\mathsf{B}_d\mathbf{c}_{d,t}](\mathbf{x})\\ &= \mathbf{I}(\mathbf{x},0)+[\mathsf{B}\mathbf{c}_t](\mathbf{x})\ \forall\mathbf{x}\in\mathscr{F},\end{aligned}\tag{1}$$

where $\mathsf{B}=[\mathsf{B}_i|\mathsf{B}_d]$, $\mathbf{c}_t^\top=(\mathbf{c}_{i,t}^\top,\mathbf{c}_{d,t}^\top)^\top$, $k=\dim(\mathbf{c}_t)$, and $\mathscr{F}$ represents the set of pixels of the face used for tracking. Vectors $\mathbf{c}_i$ and $\mathbf{c}_d$ are, respectively, the illumination and the deformation appearance parameters.

Models similar to (1) have been used previously in the context of illumination change invariant 3D and 2D rigid face tracking. LaCascia et al. [5] introduced a 3D rigid face model, whose constancy equation also had two independent linear subspaces. Each subspace represented illumination and rigid motion warping templates, respectively. Also, Hager and Belhumeur [3] used a 2D rigid face model with a similar constancy equation in which a single linear subspace was used to model changes in illumination. Our model differs from these in that the target face can deform and the appearance model is computed in the image plane, rather than the texture map plane, as is the case of [5]. Like our model, Tenenbaum and Freeman's [28] and Grimes et al.'s [29] bilinear models and Vasilescu and Terzopoulos' multilinear model [30] also used a linear approach to represent variations in the illumination and facial expression, but illumination and appearance are not independent in their models. Here we assume that illumination and facial deformation subspaces are independent. This assumption gives our model several remarkable properties:

- It reduces the number of model parameters, which grow linearly with image size.
- It reduces the number of training images. We will not need training samples with all facial expressions under all possible illumination conditions, as was the case in all previous bilinear or multilinear approaches. We will only require two sets of sample images: one in which one facial expression is subject to all possible illuminations and another set in which the face adopts all facial expressions under one particular illumination (see Section 5).
- Finally, because of the previous property, this model can be trained automatically without any manual intervention (see Section 5).

The experiments described in Sections 5 and 6 show that the model introduced in this section accurately models facial expressions and illumination.

3. Efficient linear subspace model fitting

If the brightness constancy Eq. (1) holds, then tracking consists of estimating, for each image in the sequence, the values of the motion, $\boldsymbol{\mu}$, and appearance, $\mathbf{c}$, parameters that minimise the error function

$$\begin{aligned}E(\boldsymbol{\mu},\mathbf{c}) &= \sum_{\forall\mathbf{x}\in\mathscr{F}}[I(f(\mathbf{x},\boldsymbol{\mu}),t)-I(\mathbf{x},0)-[\mathsf{B}\mathbf{c}](\mathbf{x})]^2\\ &= \|\mathbf{I}(f(\cdot,\boldsymbol{\mu}),t)-\mathbf{I}(0)-\mathsf{B}\mathbf{c}\|^2,\end{aligned}\tag{2}$$

where $\mathbf{I}(f(\cdot,\boldsymbol{\mu}),t)$ denotes the vector of values of $I(f(\mathbf{x},\boldsymbol{\mu}),t)$ for all $\mathbf{x}\in\mathscr{F}$. In order to robustly estimate the minimum value of (2), the quadratic error norm can be replaced by a robust one (see e.g. [33,3,39]). Generally, it is hard to minimise (2) since pixel brightness values are non-linearly related to $\boldsymbol{\mu}$ and $\mathbf{c}$.

Different subspace-based tracking approaches use various minimisation procedures. In [33], Black and Jepson use an analogy with parametrised optical flow estimation to introduce an iterative algorithm based on a gradient-descent procedure with multi-resolution and a robust error norm. Their approach is computationally very demanding, since, for example, the image Jacobian has to be computed for each incoming image and for each level in the multi-resolution pyramid. This makes their approach unsuitable for a real-time implementation. Particle filter-based approaches [29,20,36] are inadequate for an efficient implementation too since our parameter space will be of the order of tens of parameters.

Other approaches based on traditional optimisation techniques [38,3,42,5] use the continuity of motion to find an initial starting point in the minimisation, based on which they introduce efficient minimisation procedures that can be used to track at video frame rates. Suppose that $\boldsymbol{\mu}_t$ and $\mathbf{c}_t$ are known and describe the position and deformation of the target at time t. Then, at time $t+\tau$, the minimisation can be expressed in terms of the parameter offsets $\delta\boldsymbol{\mu}$ and $\delta\mathbf{c}$

$$E(\delta\boldsymbol{\mu},\delta\mathbf{c})=\|\mathbf{I}(f(\cdot,\boldsymbol{\mu}_t,\delta\boldsymbol{\mu}),t+\tau)-\mathbf{I}(0)-\mathsf{B}(\mathbf{c}_t,\delta\mathbf{c})\|^2,\tag{3}$$

where we use $\mathsf{B}(\mathbf{c}_t,\delta\mathbf{c})$ to denote that the value of that vector depends on both $\mathbf{c}_t$ and $\delta\mathbf{c}$. The minimum of $E(\delta\boldsymbol{\mu},\delta\mathbf{c})$ can be iteratively reached by expanding the Taylor series of E and solving linearly for $\delta\boldsymbol{\mu}$ and $\delta\mathbf{c}$. Depending on how the parameter offsets are combined

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015

with $\boldsymbol{\mu}_t$ and $\mathbf{c}_t$ these efficient minimisation procedures can be grouped into compositional and additive approaches.

In the *compositional* approaches [38,42,34,43]

$$f(\mathbf{x}, \boldsymbol{\mu}_t, \delta\boldsymbol{\mu}) = f(\mathbf{x}, \boldsymbol{\mu}_{t+\tau}) = f(g(\mathbf{x}, \delta\boldsymbol{\mu}), \boldsymbol{\mu}_t), \tag{4}$$

where f and g are warping functions such that $f \circ g \mapsto f$ and $g(\mathbf{x}, \mathbf{0}) = \mathbf{x}$. Baker and Matthews' *inverse compositional image alignment* (ICIA) [38,35,34] and Jurie's *hiperplane approximation template matching* [42] are two prominent examples. The main advantage of the compositional approach is that the Jacobian of the image brightness values w.r.t. the motion parameters that emerges in the Taylor series expansion of E is constant, and hence can be precomputed off-line in order to efficiently solve the minimisation.

In the *additive* algorithms

$$f(\mathbf{x}, \boldsymbol{\mu}_t, \delta\boldsymbol{\mu}) = f(\mathbf{x}, \boldsymbol{\mu}_{t+\tau}) = f(\mathbf{x}, \boldsymbol{\mu}_t + \delta\boldsymbol{\mu}). \tag{5}$$

Lucas and Kanade's [44] seminal work and Hager and Belhumeur's [3] and La Cascia et al.'s [5] subspace-based results are well known examples of this approach. Its main advantage is that it can be applied to any warping function f, as long as it is differentiable.

Both additive and compositional approaches have their drawbacks. For the compositional approach, f and g have to be closed under composition, something which does not hold for subspace-based models and additional approximations are necessary [34]. On the other hand, the additive approach is computationally more demanding, since the image Jacobian must be recomputed for each frame in the sequence, although, for certain fs, the Jacobian can be factored and can also be efficiently estimated [3].

In this section, we revisit both approaches and introduce an efficient minimisation scheme for (3) based on an additive algorithm. The new minimisation is more general than the subspace-based ICIA algorithm described in [34] or Hager and Belhumeur's [3] previous factorisation-based additive approach, since both approaches are based on similar assumptions on the smoothness of the subspace basis, whereas the new procedure makes no such assumption.

In the following sections, we will briefly review the compositional and additive image alignment procedures and introduce the new algorithm in the context of the additive approach.

3.1. Basic compositional image alignment algorithm

In the basic approaches we will assume that the target's appearance is constant in the sequence. Later we lift this restriction. Let us assume for now that target's appearance does not change, i.e. $\mathbf{c} = \mathbf{0}$. In the compositional framework, the parameter offsets, $\delta\boldsymbol{\mu}$, are compositionally combined with the known position of the target at time t, $\boldsymbol{\mu}_t$, to obtain $\boldsymbol{\mu}_{t+\tau}$, as shown in (4) (see Fig. 1). Therefore, Eq. (3) is transformed into

$$E(\delta\boldsymbol{\mu}) = \|\mathbf{I}(f(g(\cdot, \delta\boldsymbol{\mu}), \boldsymbol{\mu}_t), t+\tau) - \mathbf{I}(0)\|^2.$$

The ICIA algorithm assumes $g \equiv f^{-1}$ and introduces a modification in the previous scheme by exchanging the rôles of $\mathbf{I}(\mathbf{x}, t+\tau)$ and $\mathbf{I}(\mathbf{x}, 0)$ [35,34]. Rather than computing the incremental warp w.r.t. $\mathbf{I}(f(\mathbf{x}, \boldsymbol{\mu}_t), t+\tau)$ it is computed w.r.t. $\mathbf{I}(\mathbf{x}, 0)$

$$E(\delta\boldsymbol{\mu}) = \|\mathbf{I}(f(\cdot, \boldsymbol{\mu}_t), t+\tau) - \mathbf{I}(f(\cdot, \delta\boldsymbol{\mu}), 0)\|^2. \tag{6}$$

Expanding a Taylor series of (6) we get

$$E(\delta\boldsymbol{\mu}) = \|\mathscr{E} - \mathsf{M}\delta\boldsymbol{\mu}\|^2, \tag{7}$$

where $\mathsf{M} = \left[\frac{\partial \mathbf{I}(f(\cdot,\boldsymbol{\mu}),0)}{\partial \boldsymbol{\mu}}\right]_{\boldsymbol{\mu}=\mathbf{0}}$ is the Jacobian of $\mathbf{I}(0)$, w.r.t. $\boldsymbol{\mu}$, and $\mathscr{E} = \mathbf{I}(f(\cdot, \boldsymbol{\mu}_t), t+\tau) - \mathbf{I}(0)$ is the error made when warping the image acquired at time $t+\tau$ with parameters $\boldsymbol{\mu}_t$.

Now, the motion offset can be estimated by minimising (7). This is achieved by solving

$$\mathsf{M}\,\delta\boldsymbol{\mu} = \mathscr{E}. \tag{8}$$

The least-squares solution of (8) is given by

$$\delta\boldsymbol{\mu} = \mathsf{M}^{+}\mathscr{E}, \tag{9}$$

where M^{+} is the generalised matrix inverse of M. Note that M^{+} is a constant matrix that can be precomputed off-line. This is the key to this algorithm's efficiency.

Once we know the motion offset the motion parameters at time $t+\tau$ may be obtained from (4)

$$f(\mathbf{x}, \boldsymbol{\mu}_{t+\tau}) = f(f^{-1}(\mathbf{x}, \delta\boldsymbol{\mu}), \boldsymbol{\mu}_t).$$

The major advantage of the compositional approach is that the Jacobian M is constant, and can therefore be precomputed off-line in order to efficiently minimise Eq. (6). Unfortunately, subspace-based models of appearance are not closed under composition, and approximations to the basic compositional approach must be introduced (see e.g. [34]) in order to minimise (3).

Although it was introduced in the context of the additive image alignment approach, the dynamic extension of Jurie's *hiperplane approximation template matching* algorithm [42] is actually a compositional procedure based on a constant image Jacobian, like ICIA. The major difference is that, in this case, M^{+} is directly estimated from (9) by least-squares approximation from a set of training samples synthetically generated by warping $\mathbf{I}(f(\mathbf{x}, \delta\boldsymbol{\mu}), 0)$ with different values of $\delta\boldsymbol{\mu}$. Estimating M^{+} in this way improves the convergence of the minimisation [42].

3.2. Subspace extension to ICIA

In [34] Matthews and Baker introduced an extension to the basic ICIA algorithm in which the target's appearance is allowed to vary. Here we will revisit their approach and prove that it is based on an smoothness assumption on the subspace basis.

Following the conventions of ICIA, (3) is now expressed as

$$E(\delta\boldsymbol{\mu}, \mathbf{c}) = \|\mathbf{I}(f(\cdot, \boldsymbol{\mu}_t), t+\tau) - \mathbf{I}(f(\cdot, \delta\boldsymbol{\mu}), 0) - [\mathsf{B}\mathbf{c}](f(\cdot, \delta\boldsymbol{\mu}))\|^2. \tag{10}$$

This expression must be simultaneously minimised with respect to $\delta\boldsymbol{\mu}$ and $\mathbf{c}$. If we denote the linear subspace spanned by B as $\mathrm{span}(\mathsf{B})$ and its orthogonal complement by $\mathrm{span}(\mathsf{B})^{\perp}$, then (10) can be rewritten as

$$\begin{aligned} E(\boldsymbol{\mu}, \mathbf{c}) = {} & \|\mathbf{I}(f(\cdot, \boldsymbol{\mu}_t), t+\tau) - \mathbf{I}(f(\cdot, \delta\boldsymbol{\mu}), 0) - [\mathsf{B}\mathbf{c}](f(\cdot, \delta\boldsymbol{\mu}))\|^2_{\mathrm{span}(\mathsf{B})^{\perp}} \\ & + \|\mathbf{I}(f(\cdot, \boldsymbol{\mu}_t), t+\tau) - \mathbf{I}(f(\cdot, \delta\boldsymbol{\mu}), 0) - [\mathsf{B}\mathbf{c}](f(\cdot, \delta\boldsymbol{\mu}))\|^2_{\mathrm{span}(\mathsf{B})}, \end{aligned} \tag{11}$$

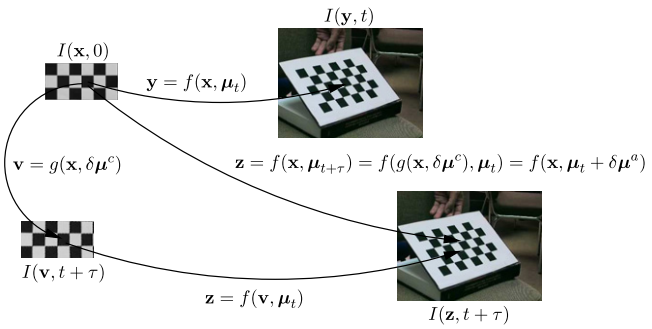


Fig. 1. Relation among images in the basic compositional and additive image alignment algorithms. Vectors $\delta\boldsymbol{\mu}^a$ and $\delta\boldsymbol{\mu}^c$ are the additive and compositional parameter increments, respectively.

where $\|\cdot\|_L^2$ denotes the square of the L2 norm of the vector projected onto the linear subspace L. The subspace extension introduced in Section 4 of [34] implicitly assumes.

$$[\mathsf{B}\mathbf{c}](f(\cdot,\delta\boldsymbol{\mu})) = \mathsf{B}\mathbf{c}, \tag{12}$$

where $[\mathsf{B}\mathbf{c}](f(\cdot,\delta\boldsymbol{\mu}))$ denotes the result of warping each column of B with $f(\mathbf{x},\delta\boldsymbol{\mu})$ and multiplying the result with $\mathbf{c}$. Given that $\|\cdot\|^2_{\mathrm{span}(\mathsf{B})^\perp}$ only considers the component of vectors in the orthogonal complement of span(B), any component in span(B) can be dropped. Then, using assumption (12), Eq. (11) simplifies to

$$\begin{aligned} E(\boldsymbol{\mu},\mathbf{c}) = {} & \|\mathbf{I}(f(\cdot,\boldsymbol{\mu}),t+\tau) - \mathbf{I}(f(\cdot,\delta\boldsymbol{\mu}),0)\|^2_{\mathrm{span}(\mathsf{B})^\perp} \\ & + \|\mathbf{I}(f(\cdot,\boldsymbol{\mu}),t+\tau) - \mathbf{I}(f(\cdot,\delta\boldsymbol{\mu}),0) - \mathsf{B}\mathbf{c}\|^2_{\mathrm{span}(\mathsf{B})}. \end{aligned} \tag{13}$$

Since the first term of (13) only depends on $\delta\boldsymbol{\mu}$, minimising it is equivalent to solving the basic ICIA minimisation (7) in the subspace $\mathrm{span}(\mathsf{B})^\perp$. This is achieved by just premultiplying either M or the error term $\mathscr{E}$ by the projection matrix $\mathsf{N}_{\mathsf{B}^\perp} = \mathsf{I} - \mathsf{B}\mathsf{B}^+$ (see Fig. 2). Now, the equation equivalent to (8) is

$$\mathsf{N}_{\mathsf{B}^\perp}\mathsf{M}\delta\boldsymbol{\mu} = \mathscr{E},$$

and the solution is

$$\delta\boldsymbol{\mu} = (\mathsf{M}^\top\mathsf{N}_{\mathsf{B}^\perp}\mathsf{M}^\top)^{-1}\mathsf{M}^\top\mathsf{N}_{\mathsf{B}^\perp}\mathscr{E}.$$

Here we have used the following properties $\mathsf{N}^\top_{\mathsf{B}^\perp} = \mathsf{N}_{\mathsf{B}^\perp}$ and $\mathsf{N}^\top_{\mathsf{B}^\perp}\mathsf{N}_{\mathsf{B}^\perp} = \mathsf{N}_{\mathsf{B}^\perp}$.

Once $\delta\boldsymbol{\mu}$ is known, $\mathbf{c}$ can be obtained by minimising the second term of (13),

$$\mathbf{c} = \mathsf{B}^+[\mathbf{I}(f(\cdot,\boldsymbol{\mu}_t),t+\tau) - \mathbf{I}(f(\cdot,\delta\boldsymbol{\mu}),0)]. \tag{14}$$

In section 6, we will denote this algorithm as MBC (Mathews and Baker's Compositional approach) and we will show that the approximation introduced in (12) worsens the convergence of the minimisation and biases the solution.

3.3. Correct compositional approach

Here we describe a correct compositional algorithm which is less efficient, but minimises (10) properly. It is similar to the *simultaneous compositional algorithm* described in [39].

Following the approach in [13], we will incrementally compute both motion, $\boldsymbol{\mu}$, and appearance, $\mathbf{c}$, parameters. Motion parameters will be estimated compositionally, while appearance parameters will be estimated additively. In this way we will be able to simultaneously solve (3) for both $\delta\boldsymbol{\mu}$ and $\delta\mathbf{c}$.

Eq. (3) would now be expressed as

$$E(\delta\boldsymbol{\mu},\delta\mathbf{c}) = \|\mathbf{I}(f(\cdot,\boldsymbol{\mu}_t),t+\tau) - \mathbf{I}(f(\cdot,\delta\boldsymbol{\mu}),0) - [\mathsf{B}(\mathbf{c}_t+\delta\mathbf{c})](f(\cdot,\delta\boldsymbol{\mu}))\|^2. \tag{15}$$

Expanding a Taylor series of (15) at $\delta\boldsymbol{\mu} = \delta\mathbf{c} = \mathbf{0}$ we get

$$E(\delta\boldsymbol{\mu},\delta\mathbf{c}) = \|\mathbf{I}(f(\cdot,\boldsymbol{\mu}_t),t+\tau) - \mathbf{I}(0) - \mathsf{B}\mathbf{c}_t - [\mathsf{M}+\dot{\mathsf{B}}\mathbf{c}_t]\delta\boldsymbol{\mu} - \mathsf{B}\delta\mathbf{c}\|^2, \tag{16}$$

where, again, $\mathsf{M} = \left[\frac{\partial\mathbf{I}(f(\cdot,\boldsymbol{\mu}),0)}{\partial\boldsymbol{\mu}}\right]_{\boldsymbol{\mu}=\mathbf{0}}$ is the Jacobian of $\mathbf{I}(0)$ with respect to $\boldsymbol{\mu}$, and

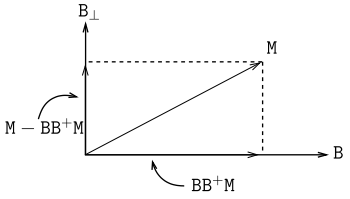


Fig. 2. The components of M orthogonal to B.

$$\dot{\mathsf{B}}\mathbf{c}_t = \left.\frac{\partial[\mathsf{B}\mathbf{c}](f(\cdot,\boldsymbol{\mu}))}{\partial\boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\mathbf{0}} = \sum_{i=1}^{k} c_{t_i}\left[\left.\frac{\partial\mathbf{b}_i(f(\cdot,\boldsymbol{\mu}))}{\partial\boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\mathbf{0}}\right]$$

is the Jacobian of the subspace basis w.r.t. the motion parameters, $\mathbf{b}_i$ being the ith column of B, c_{t_i} the ith component of $\mathbf{c}_t$ and k the dimension of the subspace.

Now, the minimum of (16) can be computed by solving

$$\mathsf{A}(\mathbf{c}_t)\begin{bmatrix}\delta\boldsymbol{\mu}\\ \delta\mathbf{c}\end{bmatrix} = \mathscr{E}, \tag{17}$$

where $\mathsf{A}(\mathbf{c}_t) = [\mathsf{M}+\dot{\mathsf{B}}\mathbf{c}_t|\mathsf{B}]$ and, in this case, $\mathscr{E} = \mathbf{I}(f(\cdot,\boldsymbol{\mu}_t),t+\tau) - \mathbf{I}(\cdot,0) - \mathsf{B}\mathbf{c}_t$. The solution to (17) is

$$\begin{bmatrix}\delta\boldsymbol{\mu}\\ \delta\mathbf{c}\end{bmatrix} = \mathsf{A}(\mathbf{c}_t)^+\mathscr{E}.$$

Now $f(\mathbf{x},\boldsymbol{\mu}_{t+\tau}) = f(f^{-1}(\mathbf{x},\delta\boldsymbol{\mu}),\boldsymbol{\mu}_t)$ and $\mathbf{c}_{t+\tau} = \mathbf{c}_t + \delta\mathbf{c}$. In the experiments conducted in Section 6 this algorithm is termed correct compositional approach (COC).

Unfortunately $\mathsf{A}(\mathbf{c}_t)^+$ depends on $\mathbf{c}_t$ and has to be recomputed for each image in the sequence. Although we now have a genuine minimiser of (3), each iteration of the algorithm is computationally more expensive than Matthews and Baker's approximated ICIA [34] and the efficient additive algorithm that will be introduced in the following section.

Here we can introduce a new approximation:

$$\dot{\mathsf{B}}\mathbf{c}_t = \left.\frac{\partial[\mathsf{B}\mathbf{c}](f(\cdot,\boldsymbol{\mu}))}{\partial\boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\mathbf{0}} = 0. \tag{18}$$

In this case, we have a new algorithm whose efficiency is similar to ICIA's, since A^+ is constant. This is actually the approximation used in [13]. In the experiments conducted in Section 6 this algorithm is denoted approximate compositional approach (APC), and we will see that, as with ICIA, assumption (18) worsens convergence.

3.4. Basic additive image alignment algorithm

Here we describe the image alignment algorithm introduced in Lucas and Kanade's seminal work [44]. Let us assume again that the target appearance does not change. In this framework, the parameter offsets, $\delta\boldsymbol{\mu}$, are additively combined with the known position of the target at time t, $\boldsymbol{\mu}_t$, to obtain $\boldsymbol{\mu}_{t+\tau}$, as shown in (5) (see Fig. 1). Therefore, Eq. (3) is transformed into

$$E(\delta\boldsymbol{\mu}) = \|\mathbf{I}(f(\cdot,\boldsymbol{\mu}_t+\delta\boldsymbol{\mu}),t+\tau) - \mathbf{I}(0)\|^2.$$

To do Gauss–Newton iterations, a Taylor series of E is expanded at $(\delta\boldsymbol{\mu}=\mathbf{0}, t=t)$, producing a new error function

$$E(\delta\boldsymbol{\mu}) = \|\mathscr{E} - \mathsf{M}(\boldsymbol{\mu}_t)\delta\boldsymbol{\mu}\|^2, \tag{19}$$

where $\mathsf{M}(\boldsymbol{\mu}_t) = \left[\frac{\partial\mathbf{I}(f(\cdot,\boldsymbol{\mu}),t)}{\partial\boldsymbol{\mu}}\right]_{\boldsymbol{\mu}=\boldsymbol{\mu}_t}$ is the Jacobian of the image acquired at time t, w.r.t. to the motion parameters $\boldsymbol{\mu}$, and $\mathscr{E} = \mathbf{I}(f(\cdot,\boldsymbol{\mu}_t),t+\tau) - \mathbf{I}(0)$ is the error made when warping the image acquired at time $t+\tau$ with parameters $\boldsymbol{\mu}_t$. Here it is assumed that

$$\left[\frac{\partial\mathbf{I}(f(\cdot,\boldsymbol{\mu}_t),r)}{\partial r}\right]_{r=t}\tau \approx \mathbf{I}(f(\cdot,\delta\boldsymbol{\mu}_t),t+\tau) - \mathbf{I}(f(\cdot,\delta\boldsymbol{\mu}_t),t).$$

Now, the motion offset can be estimated by minimising (19). This is done by solving

$$\mathsf{M}(\boldsymbol{\mu}_t)\,\delta\boldsymbol{\mu} = \mathscr{E}. \tag{20}$$

The least-squares solution of (20) is given by

$$\delta\boldsymbol{\mu} = \mathsf{M}^+(\boldsymbol{\mu}_t)\mathscr{E}. \tag{21}$$

The motion parameters at time $t+\tau$ are obtained from (5).

The main advantage of the additive approach is that it can be applied to any warping function $f(\mathbf{x},\boldsymbol{\mu})$, provided it is differentiable w.r.t. $\boldsymbol{\mu}$. Another advantage is the simple procedure used to update

the motion parameters, $\boldsymbol{\mu}_{t+\tau} = \boldsymbol{\mu}_t + \delta\boldsymbol{\mu}$. The main drawback of this approach is its computational cost, since $\mathsf{M}(\boldsymbol{\mu}_t)$ has to be recomputed and inverted for each frame in the sequence. This prevents this algorithm from being used in real-time applications.

Hager and Belhumeur [3] introduced an efficient procedure for minimising (19). By using a restricted version of the brightness constancy Eq. (1) in which $\mathbf{c} = \mathbf{0}$, the Jacobian of $\mathbf{I}(t), \mathsf{M}(\boldsymbol{\mu}_t)$, can be expressed in terms of the gradients of $\mathbf{I}(0)$ (a particularisation of (25) for $\mathbf{c} = \mathbf{0}$).

3.5. Efficient subspace extension for the additive approach

Following the convention of the additive approach (5), Eq. (3) is now expressed as

$$E(\delta\boldsymbol{\mu}, \delta\mathbf{c}) = \|\mathbf{I}(f(\cdot, \boldsymbol{\mu}_t + \delta\boldsymbol{\mu}), t+\tau) - \mathbf{I}(0) - \mathsf{B}(\mathbf{c}_t + \delta\mathbf{c})\|^2.$$

Expanding a Taylor series of this equation at $\delta\boldsymbol{\mu} = \delta\mathbf{c} = 0$, and $t = t$ we obtain a new linearised error function

$$E(\delta\boldsymbol{\mu}, \delta\mathbf{c}) = \|\mathbf{I}(f(\cdot, \boldsymbol{\mu}_t), t+\tau) - \mathbf{I}(0) - \mathsf{B}\mathbf{c}_t + \mathsf{M}\delta\boldsymbol{\mu} - \mathsf{B}\delta\mathbf{c}\|^2, \tag{22}$$

where $\mathsf{M} = \left[\left.\frac{\partial \mathbf{I}(f(\cdot,\boldsymbol{\mu}),t)}{\partial \boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\boldsymbol{\mu}_t}\right]$ is the $N \times n$ ($n = \dim(\boldsymbol{\mu})$) Jacobian matrix of $\mathbf{I}(f(\cdot, \boldsymbol{\mu}_t), t)$. Hager and Belhumeur [3] also introduced an efficient procedure for minimising (22) in the context of invariance to illumination changes, by assuming $\nabla_{\mathbf{x}}[\mathsf{B}\mathbf{c}](\mathbf{x}) = 0$, and estimating the minimum of (22) in the subspace orthogonal to B. In this case M can be expressed in terms of the gradient of $\mathbf{I}(0)$ and can be partially precomputed off-line. The result of this off-line computation is a Jacobian that only depends on $\boldsymbol{\mu}_t$. In the experiments section we denote this algorithm as Hager and Belhumeur's additive approach (HBA).

In this section, we introduce an efficient procedure for minimising (22) based on a factorisation of the Jacobian without Hager and Belhumeur's restriction. Following an approach similar to [3], we will use the Gauss–Newton procedure and we will show that the Jacobian can be efficiently computed by expressing it in terms of the gradient of the subspace basis vectors. Then $\delta\boldsymbol{\mu}$ and $\delta\mathbf{c}$ will be estimated in closed-form. In the experiments section we will denote this new algorithm with the letters OUr Additive approach (OUA).

3.5.1. Jacobian matrix factorisation

One of the obstacles for minimising (22) on-line while tracking is the computational cost of estimating M for each frame. In this subsection we will show that M can be factored into the product of two matrices, $\mathsf{M}_0\Sigma(\boldsymbol{\mu}, \mathbf{c})$, where M_0 is a constant matrix, which can be computed off-line.

Each row m_i of M can be written as the product

$$m_i = \nabla_{\mathbf{f}}\mathbf{I}(f(\mathbf{x}_i, \boldsymbol{\mu}_t), t)^\top f_{\boldsymbol{\mu}}(\mathbf{x}_i, \boldsymbol{\mu}_t). \tag{23}$$

where $\nabla_{\mathbf{f}}\mathbf{I}(f(\mathbf{x}_i, \boldsymbol{\mu}_t), t)^\top = \left[\left.\frac{\partial \mathbf{I}(\mathbf{y},t)}{\partial \mathbf{y}}\right|_{\mathbf{y}=f(\mathbf{x}_i,\boldsymbol{\mu}_t)}\right]$ and $f_{\boldsymbol{\mu}}(\mathbf{x}_i, \boldsymbol{\mu}_t) = \left[\left.\frac{\partial f(\mathbf{x}_i,\boldsymbol{\mu})}{\partial \boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\boldsymbol{\mu}_t}\right]$.

Taking derivatives w.r.t. $\mathbf{x}_i$ on both sides of (1) we get

$$\nabla_{\mathbf{f}}\mathbf{I}(f(\mathbf{x}_i, \boldsymbol{\mu}_t), t)^\top f_{\mathbf{x}_i}(\mathbf{x}_i, \boldsymbol{\mu}_t) = \nabla_{\mathbf{x}_i}\mathbf{I}(\mathbf{x}_i, 0) + \nabla_{\mathbf{x}_i}[\mathsf{B}\mathbf{c}_t](\mathbf{x}_i), \tag{24}$$

where $f_{\mathbf{x}_i}(\mathbf{x}_i, \boldsymbol{\mu}_t) = \left[\left.\frac{\partial f(\mathbf{x},\boldsymbol{\mu}_t)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_i}\right]$ and $\nabla_{\mathbf{x}}$ denotes the image gradient. Finally, from (23) and (24) we get a new expression for M, in which the Jacobian is now expressed in terms of the gradients of $\mathbf{I}(0)$ and B and the Jacobians of f, which depends on $\boldsymbol{\mu}$ and $\mathbf{c}$. This is denoted by representing M as

$$\mathsf{M}(\boldsymbol{\mu}, \mathbf{c}) = \begin{bmatrix} (\nabla_{\mathbf{x}_1}\mathbf{I}(\mathbf{x}_1, 0) + \sum_j \nabla_{\mathbf{x}_1}[\mathbf{b}_j c_j](\mathbf{x}_1))^\top f_{\mathbf{x}_1}(\mathbf{x}_1, \boldsymbol{\mu})^{-1} f_{\boldsymbol{\mu}}(\mathbf{x}_1, \boldsymbol{\mu}) \\ \vdots \\ (\nabla_{\mathbf{x}_N}\mathbf{I}(\mathbf{x}_N, 0) + \sum_j \nabla_{\mathbf{x}_N}[\mathbf{b}_j c_j](\mathbf{x}_N))^\top f_{\mathbf{x}_N}(\mathbf{x}_N, \boldsymbol{\mu})^{-1} f_{\boldsymbol{\mu}}(\mathbf{x}_N, \boldsymbol{\mu}) \end{bmatrix}, \tag{25}$$

where $\mathbf{b}_j$ is the jth column of B, c_j is the jth element of the appearance vector $\mathbf{c}$ and N is the number of pixels in $\mathscr{F}$.

Let

$$\mathsf{B}_\nabla(\mathbf{x}_i) = \left[\begin{bmatrix} \nabla_u\mathbf{I}(\mathbf{x}_i,0) \\ \nabla_u[\mathbf{b}_1](\mathbf{x}_i) \\ \vdots \\ \nabla_u[\mathbf{b}_k](\mathbf{x}_i) \end{bmatrix}^\top, \begin{bmatrix} \nabla_v\mathbf{I}(\mathbf{x}_i,0) \\ \nabla_v[\mathbf{b}_1](\mathbf{x}_i) \\ \vdots \\ \nabla_v[\mathbf{b}_k](\mathbf{x}_i) \end{bmatrix}^\top\right] \text{ and } \mathsf{C} = \begin{bmatrix} [1\mathbf{c}^\top] & \mathbf{0}^\top \\ \mathbf{0}^\top & [1\mathbf{c}^\top] \end{bmatrix}^\top,$$

where $\mathbf{x}^\top = (u, v)$. Then (25) can then be rewritten as

$$\mathsf{M}(\boldsymbol{\mu}, \mathbf{c}) = \begin{bmatrix} \mathsf{B}_\nabla(\mathbf{x}_1)\mathsf{C}f_{\mathbf{x}}(\mathbf{x}_1, \boldsymbol{\mu})^{-1} f_{\mu}(\mathbf{x}_1, \boldsymbol{\mu}) \\ \vdots \\ \mathsf{B}_\nabla(\mathbf{x}_N)\mathsf{C}f_{\mathbf{x}}(\mathbf{x}_N, \boldsymbol{\mu})^{-1} f_{\mu}(\mathbf{x}_N, \boldsymbol{\mu}) \end{bmatrix}. \tag{26}$$

Therefore, M can be expressed in terms of the gradient of the subspace basis vectors, B_∇, which are constant, and the motion and appearance parameters $(\boldsymbol{\mu}, \mathbf{c})$, which vary over time. If we choose a motion model f such that $\mathsf{C}f_{\mathbf{x}}(\mathbf{x}_i, \boldsymbol{\mu})^{-1} f_{\mu}(\mathbf{x}_i, \boldsymbol{\mu}) = \Gamma(\mathbf{x}_i)\Sigma(\boldsymbol{\mu}, \mathbf{c})$, then $\mathsf{M}(\boldsymbol{\mu}, \mathbf{c})$ may be factored into

$$\mathsf{M}(\boldsymbol{\mu}, \mathbf{c}) = \begin{bmatrix} \mathsf{B}_\nabla(\mathbf{x}_1)\Gamma(\mathbf{x}_1) \\ \vdots \\ \mathsf{B}_\nabla(\mathbf{x}_N)\Gamma(\mathbf{x}_N) \end{bmatrix} \Sigma(\boldsymbol{\mu}, \mathbf{c}) = \mathsf{M}_0\Sigma(\boldsymbol{\mu}, \mathbf{c}), \tag{27}$$

where M_0 is a constant matrix and Σ depends on $\mathbf{c}$ and $\boldsymbol{\mu}$. The columns of M_0 are the motion templates of our tracking algorithm. If we use a motion model with n parameters and an appearance subspace basis with l vectors we will have

$$\mathsf{M}_0 = (\mathbf{m}_1^1, \ldots, \mathbf{m}_1^l, \mathbf{m}_2^1, \ldots, \mathbf{m}_2^l, \ldots, \mathbf{m}_n^1, \ldots, \mathbf{m}_n^l),$$

where $\mathbf{m}_j^i$ is the motion template corresponding to the jth motion parameter and the ith subspace basis vector. Motion templates can be represented as images (see Fig. 3).

In Appendix A, we show how the factorisation introduced above applies to the motion models most commonly used in computer vision.

3.5.2. Minimising $E(\delta\boldsymbol{\mu}, \delta\mathbf{c})$

The minimum of (22) can be estimated by least-squares

$$\begin{bmatrix} \delta\boldsymbol{\mu} \\ \delta\mathbf{c} \end{bmatrix} = -(\mathsf{M}_J^\top \mathsf{M}_J)^{-1}\mathsf{M}_J\mathscr{E}, \tag{28}$$

where $\mathsf{M}_J = [\mathsf{M}| - \mathsf{B}]$ and $\mathscr{E} = \mathbf{I}(f(\cdot, \boldsymbol{\mu}_t), t+\tau) - \mathbf{I}(0) - \mathsf{B}\mathbf{c}_t$.

Using the matrix inversion lemma $(\mathsf{M}_J^\top \mathsf{M}_J)^{-1}$ can be expanded into

$$\begin{bmatrix} \mathsf{M}^\top\mathsf{M} & -\mathsf{M}^\top\mathsf{B} \\ -\mathsf{B}^\top\mathsf{M} & \mathsf{B}^\top\mathsf{B} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathsf{M}^\top\mathsf{N}_B\mathsf{M})^{-1} & (\mathsf{M}^\top\mathsf{N}_B\mathsf{M})^{-1}\mathsf{M}^\top\mathsf{B}(\mathsf{B}^\top\mathsf{B})^{-1} \\ (\mathsf{B}^\top\mathsf{N}_M\mathsf{B})^{-1}\mathsf{B}^\top\mathsf{M}(\mathsf{M}^\top\mathsf{M})^{-1} & (\mathsf{B}^\top\mathsf{N}_M\mathsf{B})^{-1} \end{bmatrix}, \tag{29}$$

where $\mathsf{N}_B = \mathsf{I} - \mathsf{B}\mathsf{B}^+$ and $\mathsf{N}_M = \mathsf{I} - \mathsf{M}\mathsf{M}^+$. Introducing Eq. (29) into (28) we get the solution for $\delta\boldsymbol{\mu}$ and $\delta\mathbf{c}$

$$\delta\boldsymbol{\mu} = -(\mathsf{M}^\top\mathsf{N}_B\mathsf{M})^{-1}\mathsf{M}^\top\mathsf{N}_B\mathscr{E}, \tag{30}$$

$$\delta\mathbf{c} = (\mathsf{B}^\top\mathsf{N}_M\mathsf{B})^{-1}\mathsf{B}^\top\mathsf{N}_M\mathscr{E}. \tag{31}$$

Fortunately, N_B is a constant matrix, then from (30) by factoring M according to (27) we get an efficient solution for $\delta\boldsymbol{\mu}$

$$\delta\boldsymbol{\mu} = -(\Sigma^{\mathrm{T}}\Lambda_{M1}\Sigma)^{-1}\Lambda_{M2}\mathscr{E}, \tag{32}$$

where $\Lambda_{M1} = \mathsf{M}_0^\top\mathsf{N}_B\mathsf{M}_0$ and $\Lambda_{M2} = \mathsf{M}_0^\top\mathsf{N}_B$ are constant and can be precomputed off-line.

On the other hand, the solution obtained in (31) for $\delta\mathbf{c}$ is not efficient, since N_M depends on $(\boldsymbol{\mu}, \mathbf{c})$ and would have to be recomputed for each frame in the sequence. Nevertheless, an effi-

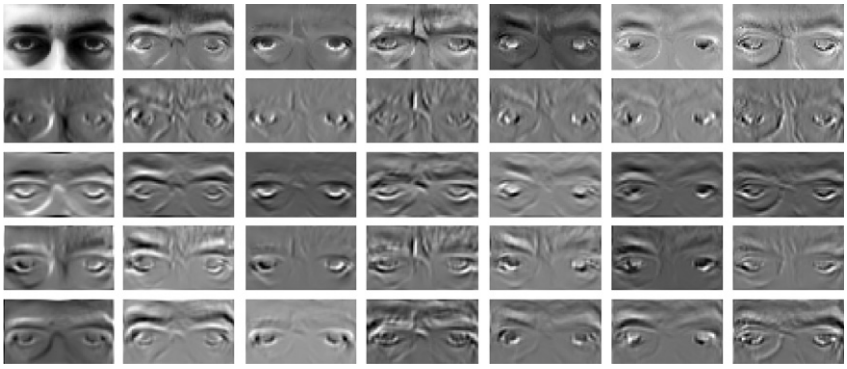


Fig. 3. Motion templates for the eye region with $N = 2100$ pixels, $l = 35$ basis vectors, and $n = 4$ (rotation, translation and scale). The first row shows from left to right the mean image and the first six basis vectors. Rows two to four show, from top to bottom, the motion templates corresponding to the horizontal and vertical translation, rotation and scale, respectively.

cient solution can be obtained from (22) by least-squares, considering that $\delta\boldsymbol{\mu}$ is known

$$\delta\mathbf{c} = \Lambda_B[\mathsf{M}\delta\boldsymbol{\mu} + \mathscr{E}], \tag{33}$$

where $\Lambda_B = \mathsf{B}^+$ is also constant and can be precomputed off-line.

The term $\mathsf{M}\delta\boldsymbol{\mu}$ represents the brightness variation in $\mathbf{I}(t)$ due to a motion of magnitude $\delta\boldsymbol{\mu}$. Intuitively Eq. (33) states that the appearance parameters are computed by projecting onto the subspace B the rectified image corrected to take into account the incremental motion $\delta\boldsymbol{\mu}$ and the already known appearance $\mathsf{B}\mathbf{c}_t$.

This result differs from those presented in [34] and in [3] in that: (a) here model parameters are additively updated, whereas the update procedure in [34] is compositional; (b) here subspace appearance parameters are incrementally estimated and additively updated ($\mathbf{c}_{t+1} = \delta\mathbf{c} + \mathbf{c}_t$) and, consequently, $\mathscr{E}$ includes term $-\mathsf{B}\mathbf{c}_t$, whereas, in [34] and in [3], there is no such term and appearance parameters are not estimated incrementally; (c) here the gradient of the subspace basis is part of the Jacobian, whereas, in [34] and in [3], it is not. In the experiments conducted in Section 6 we show that for our problem the procedure introduced in this section performs better than those in [3] and [34]. Recently, in a simultaneous work to this [48], Bartoli also addresses the problem of efficient registration. He introduces a compositional solution that is slightly more efficient than ours, since his Jacobian matrix is constant. It nevertheless does not apply to the linear appearance models used in this paper but to the restricted case of global photometric transformations.

3.6. On the equivalence of the subspace basis smoothness constraints

Here we will prove that the three subspace basis smoothness constraints introduced in this section are equivalent.

The proof is immediate if we expand a Taylor series of $[\mathsf{B}\mathbf{c}](f(\cdot,\boldsymbol{\mu}))$ at $\boldsymbol{\mu} = \mathbf{0}$

$$[\mathsf{B}\mathbf{c}](f(\cdot,\delta\boldsymbol{\mu})) = \mathsf{B}\mathbf{c} + \left.\frac{\partial[\mathsf{B}\mathbf{c}](f(\cdot,\boldsymbol{\mu}))}{\partial\boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\mathbf{0}}\delta\boldsymbol{\mu} + o(\delta\boldsymbol{\mu}^2). \tag{34}$$

From (34) we can immediately infer that (12), Matthews and Baker's smoothness assumption, holds iif $\left.\frac{\partial[\mathsf{B}\mathbf{c}](f(\cdot,\boldsymbol{\mu}))}{\partial\boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\mathbf{0}} = 0$, which is actually (18), the smoothness assumption used in [13] and also in the approximate compositional algorithm.

Finally, in order to prove that the above two smoothness assumptions are equivalent to Hager and Belhumeur's, it suffices to see that

$$\left.\frac{\partial[\mathsf{B}\mathbf{c}](f(\cdot,\boldsymbol{\mu}))}{\partial\boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\mathbf{0}} = \left[\left.\frac{\partial[\mathsf{B}\mathbf{c}](\mathbf{x})}{\partial\mathbf{x}}\right|_{\mathbf{x}=f(\cdot,\mathbf{0})}\right]^\top\left[\left.\frac{\partial f(\cdot,\boldsymbol{\mu})}{\partial\boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\mathbf{0}}\right].$$

Since the Jacobian of the image warping function w.r.t. the motion parameters does not vanish (otherwise it would not be a warping function), $\left.\frac{\partial f(\mathbf{x},\boldsymbol{\mu})}{\partial\boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\mathbf{0}} \neq 0$, then $\left.\frac{\partial[\mathsf{B}\mathbf{c}](f(\cdot,\boldsymbol{\mu}))}{\partial\boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\mathbf{0}} = 0$ iif $\left.\frac{\partial[\mathsf{B}\mathbf{c}](\mathbf{x})}{\partial\mathbf{x}}\right|_{\mathbf{x}=f(\cdot,\mathbf{0})} = 0$, which is actually Hager and Belhumeur's smoothness assumption [3].

To evaluate the correctness of these smoothness assumptions, we have computed the histogram of the gradient of the illumination and deformation components of the appearance subspace, $\nabla_{\mathbf{x}}\mathsf{B}_i(\mathbf{x})$ and $\nabla_{\mathbf{x}}\mathsf{B}_d(\mathbf{x})$. In Fig. 4 we show three sample basis vectors of each subspace and their corresponding gradient histograms. The previous assumptions are true if the columns in B vary smoothly when we move in the image space, and their corresponding gradient vanishes. This is approximately the case when the subspace model only represents changes in the illumination of a rigid head, since most of the gradient values cluster tightly around zero in their histograms (see Fig. 4(b)). This was in fact the problem that Hager and Belhumeur addressed in [3]. On the other hand, the smoothness assumption is less correct when tracking faces whose appearance changes due to facial deformations, since the corresponding histograms are more spread (see Fig. 4(a)).

4. Modular appearance tracking

A modular eigenspace is a partition of the original data vector into subsets (modules) in order to compute an independent subspace model for each one [45]. This allows a more flexible, compact, accurate and better conditioned model of the regions of interest. We will consider that all the regions are part of the same object and hence that they share the same $\delta\boldsymbol{\mu}$ but could have different appearance variations.

Let $\{\mathsf{B}_1,\ldots,\mathsf{B}_r\}$ be the set of subspace basis for all modules. Then matrix B_{me} for modular appearance-based tracking can be written as

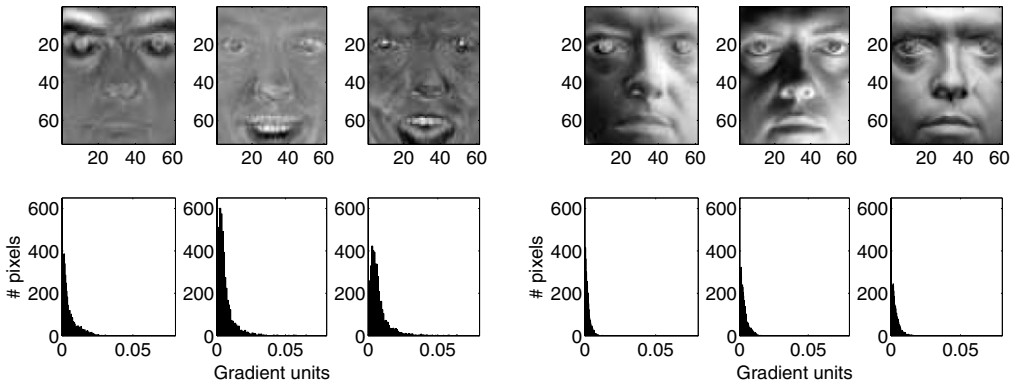
$$\mathsf{B}_{me} = \begin{bmatrix} \mathsf{B}_1 & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathsf{B}_r \end{bmatrix},$$

which is a block diagonal matrix representing the disjoint sets of image regions. The appearance of each region is modelled by subspace base B_j. Therefore, the appearance parameter vector will be $\mathbf{c} = (\mathbf{c}_1^\top,\cdots,\mathbf{c}_r^\top)^\top$, where $\mathbf{c}_j$ is the parameter vector of module j. The Jacobian matrix of the modular appearance tracker can be written as:

$$\mathsf{M} = \begin{bmatrix} \mathsf{M}_{0,1}\Sigma_1(\boldsymbol{\mu}_t,\mathbf{c}_1) \\ \vdots \\ \mathsf{M}_{0,r}\Sigma_r(\boldsymbol{\mu}_t,\mathbf{c}_r) \end{bmatrix},$$

where $\mathsf{M}_{0,j}$ and Σ_j are the result of factorising the Jacobian matrix corresponding to region j. The final factored modular tracking algorithm is shown in Algorithm 1.

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015



(a) Deformation basis vectors. (b) Illumination basis vectors.

Fig. 4. Smoothness of illumination and deformation basis vectors. The top row shows the basis vector as a 61×72 pixels image (scaled from 0 to 255) and the bottom row depicts the corresponding histogram of the gradient vectors modules. (a) Deformation basis vectors. (b) Illumination basis vectors.

5. Model training

One of the advantages of the appearance model introduced in Section 2 is that deformation and illumination subspaces are decoupled. This way, they can be trained independently. This simplifies the training process. We do not need image sequences with all facial expressions under all possible illumination conditions. Now, each subspace is trained with one video sequence. For the illumination subspace we use a sequence in which a face is subject to the illumination conditions in which the system is to operate. In our experiments these conditions were generated by orbiting a light in front of the target face. For the deformation subspace we use a sequence captured with a non-saturating frontal illumination in which the target face adopts different facial expressions. The face is located and aligned in the first frame of both sequences, then, with a procedure similar to the one described in [19], both sequences are independently tracked to extract the sample images of both subspaces (see Fig. 5).

Once we have aligned the sample images in both sequences, the basis of each subspace, B_i and B_d, are independently built using PCA. Here we assume $\mathbf{I}(\cdot, 0)$ to be the mean of the illumination and deformation samples.

Algorithm 1. Factored modular tracking

off-line
  for all region j **do**
    Compute and store M_{0j} using B_j.
    Compute and store $\Lambda_{M2j} = \mathsf{M}_{0j}^{\dagger}\mathsf{N}_{\mathsf{B}_j}$.
    Compute and store $\Lambda_{M1j} = \Lambda_{M2j}\mathsf{M}_{0j}$.
    Compute and store $\Lambda_{Bj} = (\mathsf{B}_j^{\top}\mathsf{B}_j)^{-1}\mathsf{B}_j^{\top}$.
  end for
On-line (**one iteration**):
  for all regions j **do**
    Warp $\mathbf{I}_j(\cdot, t+\delta t)$ to $\mathbf{I}_j(f(\cdot, \boldsymbol{\mu}_t), t+\delta t)$.
    Compute $\mathscr{E}_j = \mathbf{I}_j(f(\mathbf{x}, \boldsymbol{\mu}_t), t+\delta t) - \mathbf{I}_j(0) - \mathsf{B}_j\mathbf{c}_{j,t}$.
    Compute $\Sigma_j(\boldsymbol{\mu}_t, \mathbf{c}_{j,t})$.
    Compute $\mathsf{H}_j = \Sigma(\boldsymbol{\mu}_t, \mathbf{c}_{j,t})^{\top}\Lambda_{M1j}\Sigma(\boldsymbol{\mu}_t, \mathbf{c}_{j,t})$.
    Compute $\mathsf{A}_j = \Sigma(\boldsymbol{\mu}_t, \mathbf{c}_{j,t})^{\top}\Lambda_{M2j}\mathscr{E}_j$.
  end for
  Compute $\mathsf{H} = \sum_{j=1}^{r}\mathsf{H}_j$.
  Compute $\mathsf{A} = \sum_{j=1}^{r}\mathsf{A}_j$.
  Compute $\delta\boldsymbol{\mu} = -\mathsf{H}^{-1}\mathsf{A}$.
  Update $\boldsymbol{\mu}_{t+\delta t} = \boldsymbol{\mu}_t + \delta\boldsymbol{\mu}$.
  for all region j **do**
    Compute $\delta\mathbf{c}_{j,t+\delta t} = \Lambda_{Bj}[\mathsf{M}_{0j}\Sigma(\boldsymbol{\mu}_t, \mathbf{c}_{j,t})\delta\boldsymbol{\mu} + \mathscr{E}_j]$.
    Update $\mathbf{c}_{j,t+\delta t} = \mathbf{c}_{j,t} + \delta\mathbf{c}_{j,t+\delta t}$.
  end for

This initial model estimation would be correct if the illumination in the sequence used for training the deformation basis was such that $\mathbf{c}_i = \mathbf{0}$ and the facial expression in the sequence used for training the illumination was such that $\mathbf{c}_d = \mathbf{0}$. But this is not the case, since the facial expression in the illumination sequence and the illumination in the deformation sequence are arbitrary. Therefore, this initial estimate must be refined.

We iteratively improve this initial estimate by computing the illumination parameters in each sample image of the deformation sequence and subtracting the illumination component. Following a similar procedure we subtract the facial expression component from the samples of the illumination sequence. We then re-estimate the basis of the subspaces using the corrected samples. This process may be repeated again and converges in a few iterations to a definitive appearance model. This training procedure is shown in Algorithm 2, where E_i and E_d are matrices storing the aligned samples from the illumination and deformation training sequences, respectively.

Algorithm 2. Appearance model training

Input: $\mathsf{E}_i, \mathsf{E}_d$
Result: $\mathbf{I}(\cdot, 0), \mathsf{B}_i, \mathsf{B}_d$
  $\mathbf{I}(\cdot, 0) = \text{mean}(\mathsf{E}_i, \mathsf{E}_d)$;
  $\mathsf{B}_i = \text{PCA}(\mathsf{E}_i, \mathbf{I}(\cdot, 0))$;
  while Appearance basis changed **do**
    {Illumination parameters for deformation samples}
    $\mathsf{C}_i^d = (\mathsf{B}_i^{\top}\mathsf{B}_i)^{-1}\mathsf{B}_i^{\top} * (\mathsf{E}_d - \mathbf{I}(\cdot, 0))$;
    {Remove illumination from deformation samples}
    $\hat{\mathsf{E}}_d = \mathsf{E}_d - \mathsf{B}_i * \mathsf{C}_i^d$;
    {PCA from the corrected deformation samples};
    $\mathsf{B}_d = \text{PCA}(\hat{\mathsf{E}}_d, \mathbf{I}(\cdot, 0))$;
    {Deformation parameters for illumination samples}
    $\mathsf{C}_d^i = (\mathsf{B}_d^{\top}\mathsf{B}_d)^{-1}\mathsf{B}_d^{\top} * (\mathsf{E}_i - \mathbf{I}(\cdot, 0))$;
    {Remove deformation from illumination samples}
    $\hat{\mathsf{E}}_i = \mathsf{E}_i - \mathsf{B}_d * \mathsf{C}_d^i$;
    {PCA from the corrected illumination samples}
    $\mathsf{B}_i = \text{PCA}(\hat{\mathsf{E}}_i, \mathbf{I}(\cdot, 0))$;
  end while

To validate our model we did the following experiment. First, we trained the model according to the procedure described above. Then we manually selected the parameters of two facial expressions and two illuminations, and generated a set of intermediate illuminations and expressions by uniformly sampling the parameter space between those locations. We have repeated this process three times. The results are shown in Fig. 6. In spite of the model's linearity, it correctly generates the appearance of the faces.

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015

Fig. 5. Some images used to build the deformation (top row) and illumination (bottom row) subspaces.

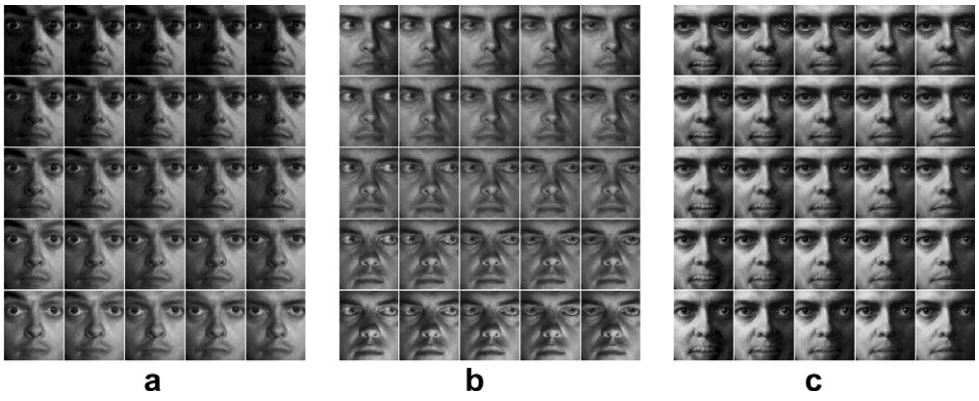


Fig. 6. Images generated using our appearance model; (a) from left to right images generated by falling eyebrows, and from top to down images generated by varying illumination; (b) idem, rolling eyes with a different illumination; (c) idem, closing the mouth using a different illumination from the previous ones.

6. Experiments

Here we evaluate the model and the minimisation procedure introduced in this paper. We have divided our experiments into three groups which are described in different sections. Section 6.1 evaluates the appearance-based model introduced in Section 2. Section 6.2 compares the minimisation algorithm introduced in Section 3.5 with the other minimisation procedures described

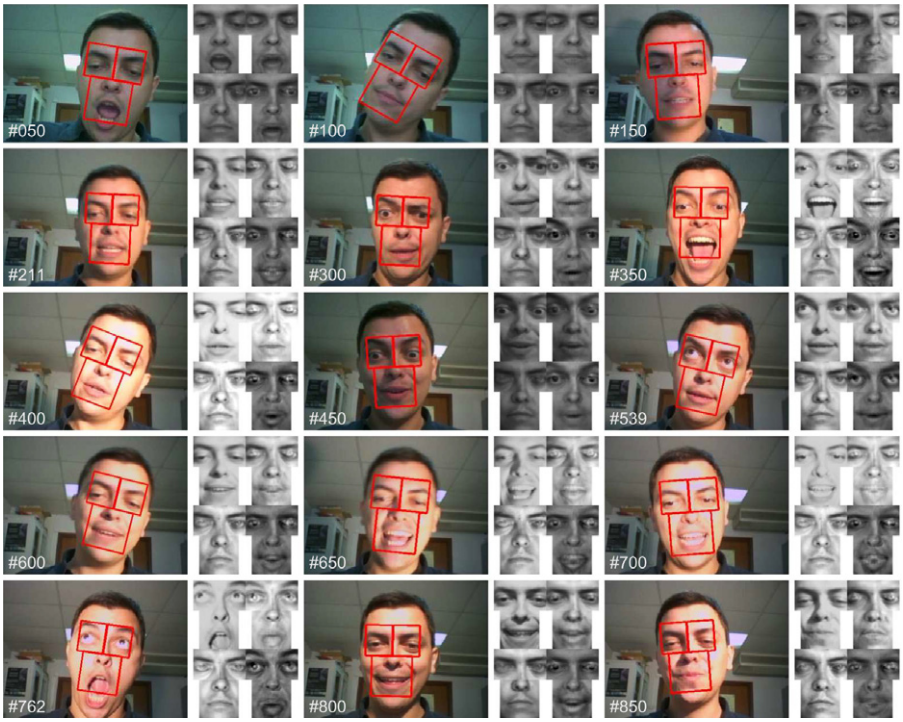


Fig. 7. Real tracking experiment.

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015

in Section 3. Section 6.3 evaluates the performance of the whole system in terms of speed, relating model size to the number of frames per second (FPS).

We will use a RTS rigid motion model in all the experiments. The modular eigenspace used for all experiments is composed of three modules chosen to include only pixels in the target face (no background pixels) (see e.g. Fig. 11(b)). The first subspace is attached to the mouth and each of the other two includes one eye and its eyebrow. The behaviour of both eyes in the experiments conducted in this section is essentially the same. Therefore, we will only display plots of the mouth and one of the eyes in order to avoid redundancy.

We use the same appearance model in all experiments described in this section. We have acquired two image sequences to train the deformation and illumination subspaces according to the procedure described in Section 5. They were captured with a Basler A312fc camera with no automatic white balance. Fig. 5 shows some frames from these sequences. The illumination subspaces resulting from the training process have dimension 5 for both the mouth (23×23 pixels) and eye regions (15×18 pixels),

Fig. 8. Contribution of each appearance subspace to the tracking process. (a) Tracking with a deformation appearance model (without illumination). (b) Tracking with illumination appearance model (without deformations).

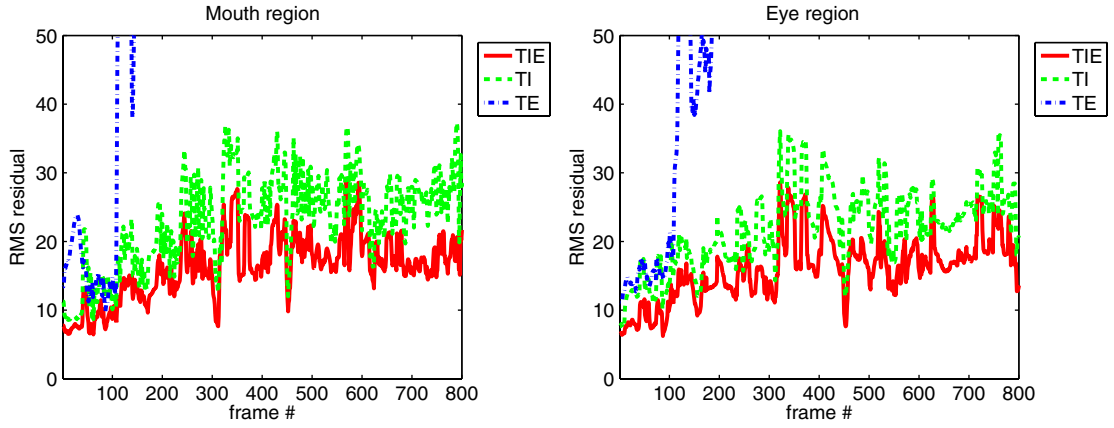


Fig. 9. Error obtained when tracking a real sequence with and without each model subspace.

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015

whereas the deformation subspaces have dimension 18 for the mouth and 9 for each eye region. With this model our tracker runs at 40 fps (including the time it takes to read images from disk and display the results on screen) with an unoptimised C++ implementation on a Pentium-M Sonoma 1.83 GHz with 2 MBytes cache memory. All images are full colour and have been conveniently converted to grey values.

6.1. Appearance model evaluation

Here we test the appearance-based model introduced in Section 2. Our first goal is to assess the global behaviour of the system when tracking a real sequence in a challenging situation. To do this we acquire a test image sequence in a different location and with different illumination conditions from the training sequence. This time we use a low-cost Apple iSight camera with automatic white balance. We modify the illumination conditions by moving a flashlight in front of the face. In this sequence fluorescent ceiling lights were on (note that during model training there were no ceiling lights). In Fig. 7 we show the tracking results for a sequence in which the face is in rigid and non-rigid motion with drastic illumination changes that produce various cast shadows. The estimated position of the face is overlayed in red (a rectangle for each module tracked). To the right of each resulting image we show four smaller images: the rectified images of the three regions used in tracking ($\mathbf{I}(f(\cdot,\boldsymbol{\mu}_t+\delta\boldsymbol{\mu}),t+\tau)$) (top left), the reconstructed images ($\mathbf{I}_j(\cdot,0)+\mathsf{B}_{i,j}\mathbf{c}_{i,j,t}+\mathsf{B}_{d,j}\mathbf{c}_{d,t,j}$) (top right), the illumination reconstructed images ($\mathbf{I}_j(\cdot,0)+\mathsf{B}_{i,j}\mathbf{c}_{i,j,t}$) (bottom left) and the deformation reconstructed images ($\mathbf{I}_j(\cdot,0)+\mathsf{B}_{d,j}\mathbf{c}_{d,j,t}$) (bottom right). The tracker is locked on the face through all the 966 frames of the experiment.

The rectified images give us an idea of how robust the tracker is to the changes in the appearance throughout the sequence. Performance is almost perfect in terms of robustness. The images reconstructed with the illumination and deformation models inform us about how well each source of appearance is separated during tracking. Here again performance is remarkable, given that the illumination subspace accurately estimates the changes in the illumination of the scene and the deformation subspace represents the facial expressions. Occasionally, the images reconstructed with the deformation model show "ghost" expressions. These are caused by facial expressions not present in the training sequence. Consequently, the generalisation properties of the model are limited since we are using a linear subspace to approximate the manifold of facial expressions, which is actually non-linear. Finally, the image reconstructed with both models gives us information about how well our model reconstructs the target image. Here, again, the reconstruction is good, except for those expressions not present in the training sequence.

The next experiment examines the contribution of each subspace in the model to the tracking process. To do this we have used two new trackers to process the image sequence in Fig. 7 using an appearance model including only the illumination (TI tracker) and only the deformation subspace (TD tracker) in each one. The full tracker is denoted by TIE (tracker with illumination and expression subspaces). Fig. 8 shows some sample images from the tracking process and Fig. 9 illustrates the RMS error. The tracker using deformation and illumination subspaces (TIE) performs consistently better than the trackers including only illumination (TI) or deformations (TE). Since the appearance changes caused by facial expressions are less significant than those due to illumination, the performance of the TE tracker is much worse than the others, and eventually it loses track. Although the illumination subspace alone can successfully track the sequence, it cannot explain all the appearance variations. That is why its residual is higher than the one obtained with the whole appearance model (TIE). Most of the appearance variation in a sequence is caused by changes in illumination. This is why the tracker using only the illumination subspace does not lose track. Nevertheless, the deformation subspace is necessary, since it makes the RMS residual smaller, and, most importantly, it is the subspace that captures the facial expression information.

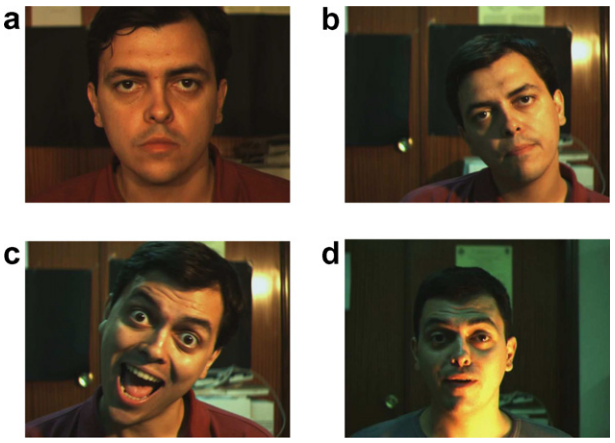


Fig. 10. Images used in static convergence experiments. (a) Image CTRI1. (b) Image CTRI2. (c) Image CTRI3. (d) Image CTRI4.

6.2. Model fitting algorithm evaluation

Here we compare the performance of the minimisation algorithm OUA introduced in Section 3.5 of this paper with the other minimisation procedures described in Section 3. We have performed static and dynamic tests. In the static tests, we used a small set of carefully chosen real images and evaluated algorithm convergence to these target images. In the dynamic tests we used test image sequences with challenging imaging situations on which we evaluated algorithm performance.

For the static tests, we used the four images shown in Fig. 10. They depict different imaging situations of increasing complexity: neutral illumination with no facial expression (CTRI1), lateral illumination with neutral facial expression (CTRI2), frontal illumination with strong facial expression (CTRI3) and strong lateral illumination with facial expression (CTRI4). We wanted to compare the convergence of the five subspace-based minimisation algorithms described in Section 3 (COC, MBC, APC, HBA and OUA) to each of the test images. We randomly selected a starting point in the space of rigid transformations and rectified the resulting piece of image texture. Then, we set the initial appearance parameters to those obtained by projecting this texture onto the appearance subspace. Finally, we used all five subspace-based minimisation algorithms to fit the model to the four static test images. Fig. 11 shows a sample starting point and the result of the fitting process for the CTRI1 test image. We repeated this experiment for different initial starting points contaminated with increasing noise levels (the process is repeated 1000 times per noise level). Figs. 12–15 show the results of the convergence rate (top left), final RMS residual (top right), number of iterations per frame (bottom left) and number of milliseconds per frame (bottom right) for each target image and noise level. The convergence rate is the percentage of tests in which a minimisation algorithm has converged to the solution. We say that a test has converged if the RMS error of the corner locations of the tracker modules is below 7 pixels from a manually selected ground truth solution.[2] As the complexity of the imaging scenario and as the noise contaminating the initial starting point increase, the convergence rate of all five algorithms decreases. Imaging

[2] We are assuming that the error made when manually aligning the model to the image is smaller than 7 pixels.

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015

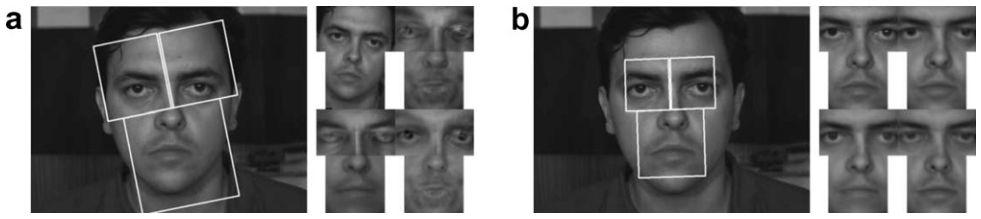


Fig. 11. An example of static convergence test. (a) Initial parameters. (b) Final parameters.

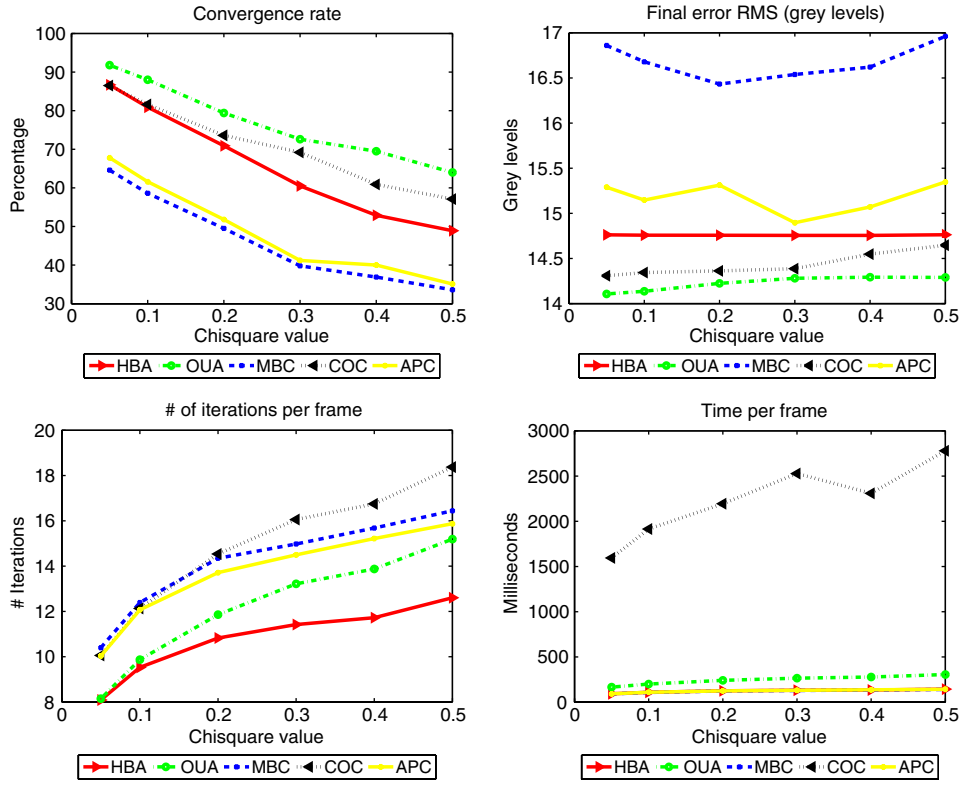


Fig. 12. Static convergence results for the CTRI1 image.

situations in which the appearance is least like the one used during training (e.g. strong lateral illuminations, like those in images CTRI2 and CTRI4) are the most challenging. In these situations the convergence rate of the "correct" algorithms (OUA and COC) is notably higher than those involving approximations (MBC, APC and HBA). Of these, the additive algorithm HBA behaves slightly better than the compositional algorithms, MBC and APC. When the illumination is neutral (CTRI1) or frontal (CTRI3) all algorithms have closer results, which get worse when the facial expression is strong (CTRI3). Nevertheless, correct algorithms (OUA and COC) are consistently better in these cases than algorithms involving approximations, of the latter, the additive HBA still has an advantage over the compositional algorithms, MBC and APC.

If we average $\mathscr{E}$, the RMS residual, of those tests in which all five algorithms converge, we obtain the final RMS residual (Figs. 12–15, top right). As the appearance of the target image becomes less like the neutral situation (image CTRI1), the final RMS residual increases. That is why images CTRI3 and CTRI4 have the highest residuals. Nevertheless, for all test images, correct algorithms (OUA and COC) have a smaller residual than algorithms involving approximations. The residual difference between correct and approximate algorithms is the highest for image CTRI4. This is the image in which the change in illumination is the strongest. From this experiment we can conclude that the smoothness assumptions introduced in the approximate algorithms bias the solution of the minimisation.

The number of iterations per processed frame (Figs. 12–15, bottom left) is obtained by averaging the number of iterations needed to converge to a solution for each noise level, only among those tests for which all five algorithms converge. Here HBA consistently takes the fewest and COC the highest iterations to converge. But, since the computational cost of each iteration is different for all five algorithms, the actual plot to be used to compare performance is the plot displaying the average time used to process one frame (Figs. 12–15, bottom right), computed from the timings of all tests in which all algorithms converge.[3] In this case the algorithms involving approximations (MBC, APC and HBA) perform best, and there is no significant difference among them. COC is by far the slowest, since it has to recompute the Jacobian for each frame. Finally, the OUA algorithm introduced in this paper is slower than the algorithms involving approximations, but notably faster than COC.

Although at first glance the additive algorithm HBA would appear to be slower than the compositional algorithms (MBC and APC), because part of the Jacobian must be recomputed in each frame, the final outcome is that it is as fast as the algorithms with

[3] These timings were obtained using a non-optimised Matlab implementation.

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015

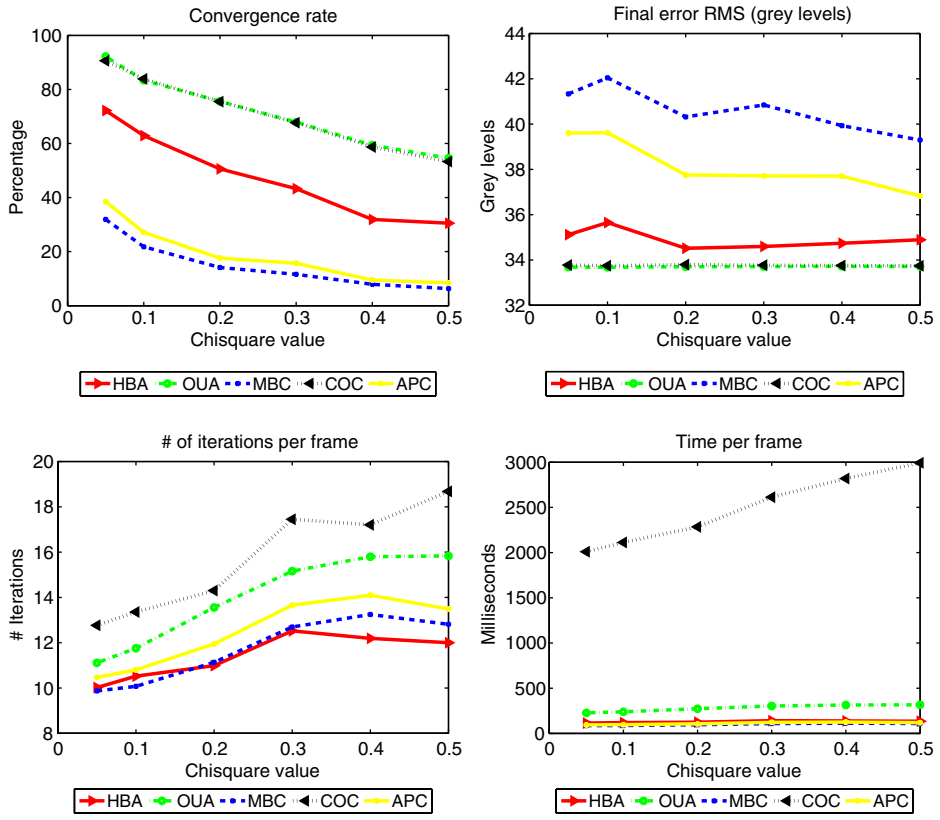


Fig. 13. Static convergence results for the CTRI2 image.

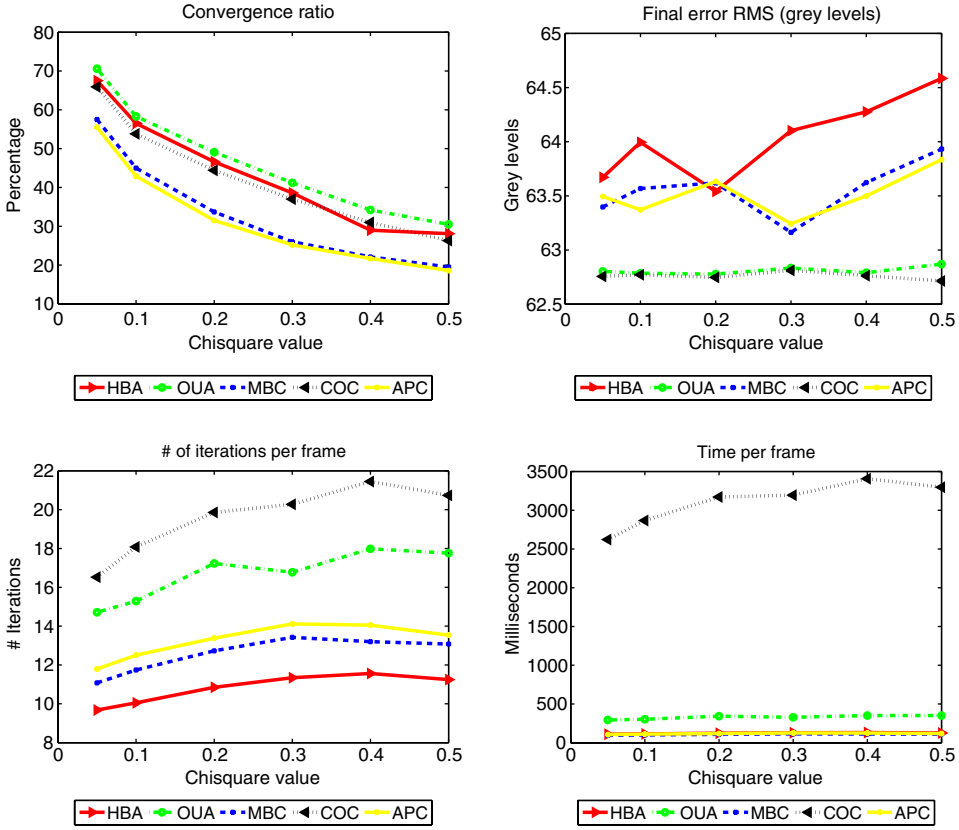


Fig. 14. Static convergence results for the CTRI3 image.

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015

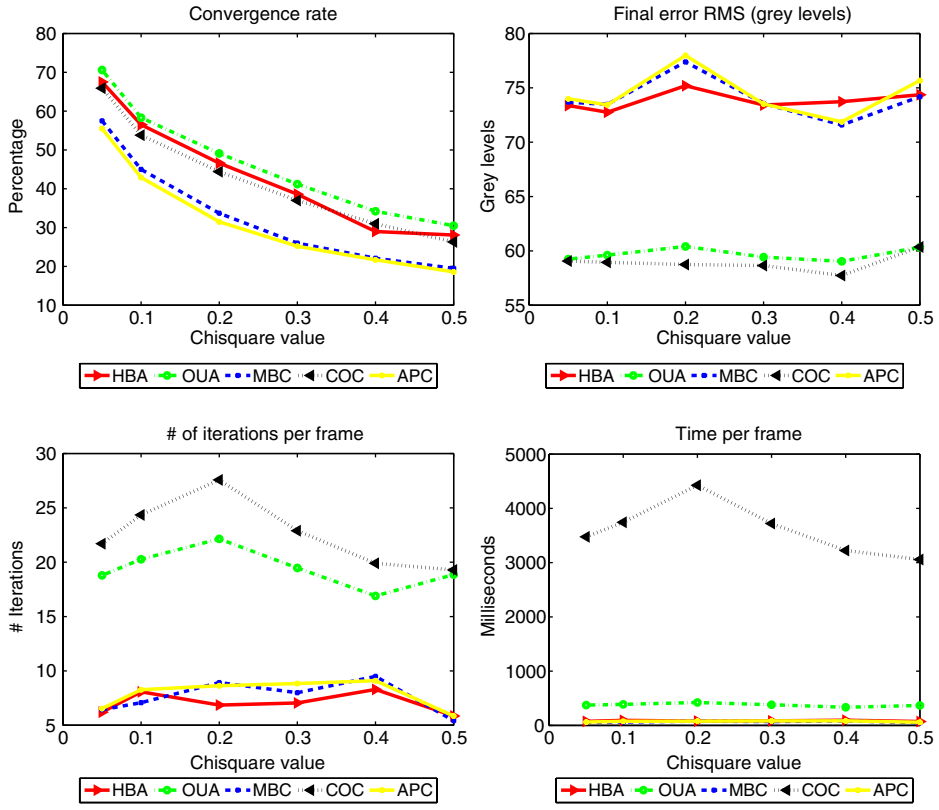


Fig. 15. Static convergence results for the CTRI4 image.

a full precomputed Jacobian because it converges faster (with fewer iterations) to a solution.

Finally, we use two real image sequences in the dynamic convergence tests. In these sequences the target face is subject to strong illumination variations, while it adopts different facial expressions. The first test sequence was captured with a Basler A312fc camera with no automatic white balance. This sequence is different from the one used for training, but was captured with the same camera and in similar illumination conditions. The second sequence was captured with an Apple iSight camera with illumination conditions different from those of the training sequence. Figs. 16 and 17 show sample images and RMS residual plots of the first sequence, respectively. This sequence shows a talking head in a dark environment. During the sequence, the illumination is varied by moving a tungsten flashlight in front of the head while it moves, talks and performs several facial expressions. Approximate algorithms (MBC and HBA) have a higher residual than correct algorithms and eventually lose track at frame 125, because of a strong lateral illumination of the face. Figs. 18 and 19 show sample images and RMS residual plots of the second sequence. In this sequence we again have a talking head in an environment illuminated by fluorescent ceiling lights. During the sequence, the head performs various rotations in the camera plane, movements and small out of camera plane rotations. At frame 36 the MBC algorithm loses track, possibly because of motion blur caused by rigid head motion. At frame 107 a tungsten flashlight starts orbiting in front of the head. From this point onwards the RMS of HBA notably increases, and it loses track at frame 320 in which an upward motion of the head coincides with a facial expression and a strong illumination. In both tests, correct algorithms (COC and OUA) correctly track the face during all the sequence, in spite of the strong illumination changes and the induced cast shadows. In terms of RMS, both achieve the lowest values and their performance is identical. In this experiment we chose not to show the results for the APC algorithm because, as happened in the static tests, the results are exactly the same as for MBC.

6.3. Performance evaluation

In this section, we evaluate the performance of our tracker in a real sequence with models of different size and on various personal computers. Best performances are obtained when all data structures used in the algorithm fit the processor's cache memory. As shown in Fig. 20, performance quickly degrades as the model size increases. Model size is the number of pixels per region times the number of basis per region times the number of regions. We have tested our algorithm on an Athlon XP 2500+ with 512 Kbytes Cache (AXP_2005), Pentium 4 2.4 GHz with 512 Kbytes Cache (P4_2.4), Pentium 4 3.2 GHz with 1 MByte Cache (P4_3.2) and on a Pentium-M Sonoma 1.86 GHz with 2 MByte (PM_1.85).

7. Conclusions

In this paper, we have experimentally shown that a linear model, based on a first order approximation to the appearance of a deforming face under varying illumination, provides enough information to efficiently track a human face when there are strong facial expression and illumination changes. In such a model both sources of variation are approximately independent and one can be trained almost independently of the other. The most remarkable consequence of this property is the considerable reduction in the number and complexity of the set of images used for estimating the parameters of the model. Contrary to previous bilinear and multilinear approaches we now do not need training samples with all facial expressions under all possible illumination conditions. Instead we only require two sets of sample images, one in which one facial expression

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015

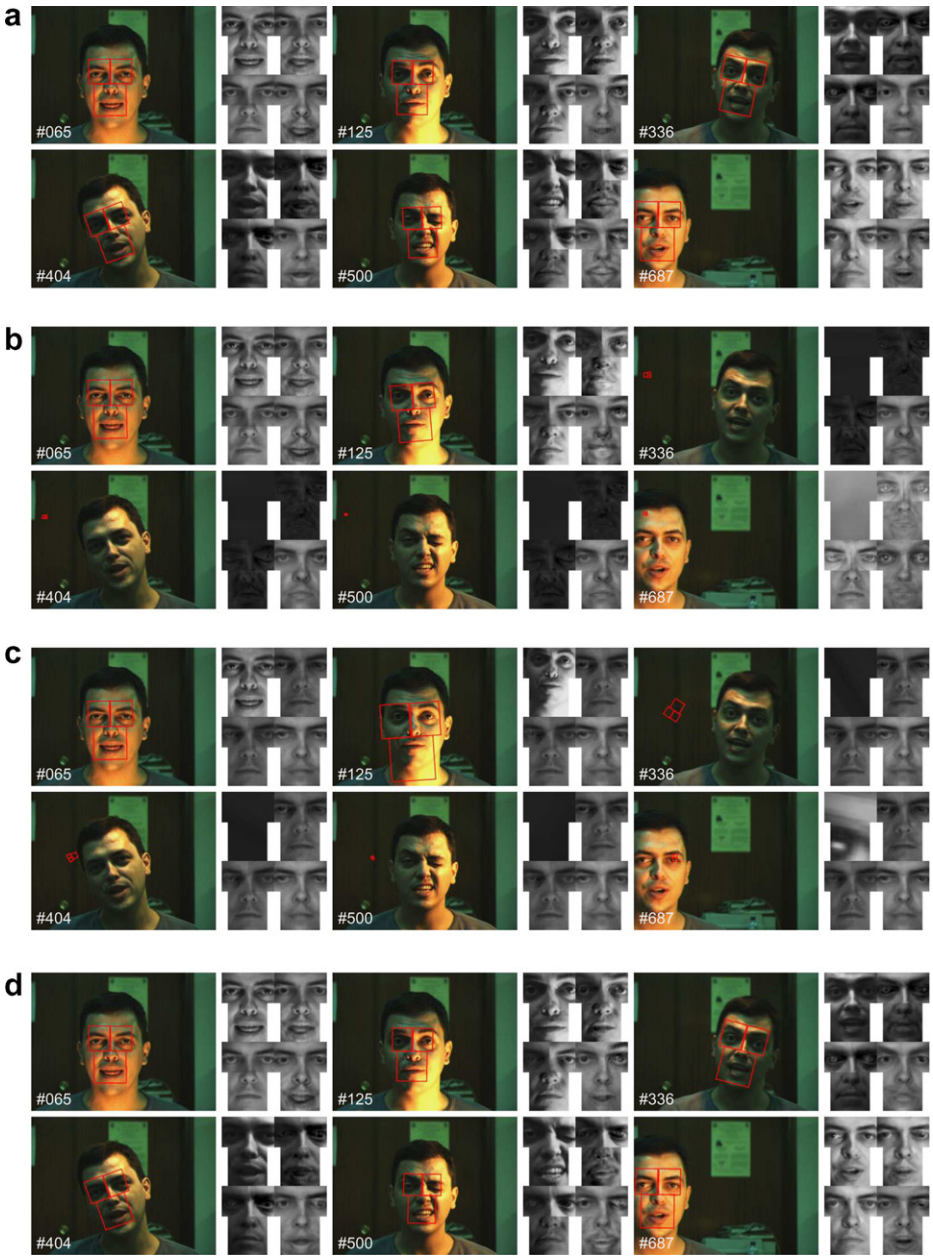


Fig. 16. Sample images comparing minimisation algorithms with the first test image sequence. (a) Tracking with OUA. (b) Tracking with MBC. (c) Tracking with HBA. (d) Tracking with COC.

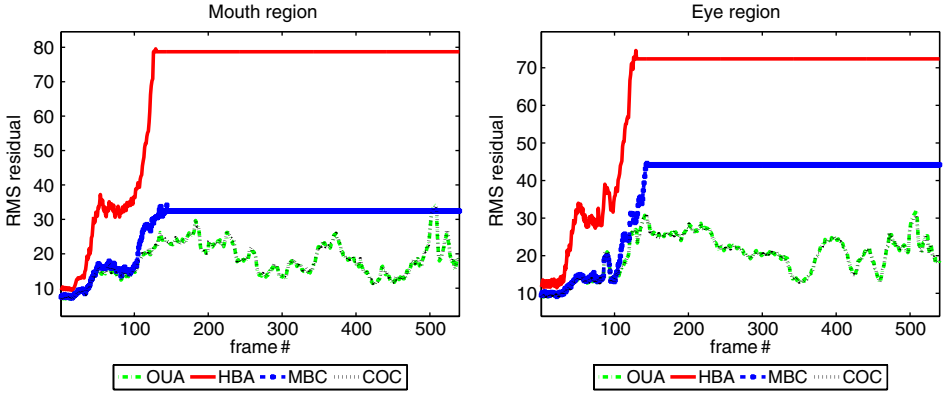


Fig. 17. RMS residual comparison for the first test image sequence.

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015

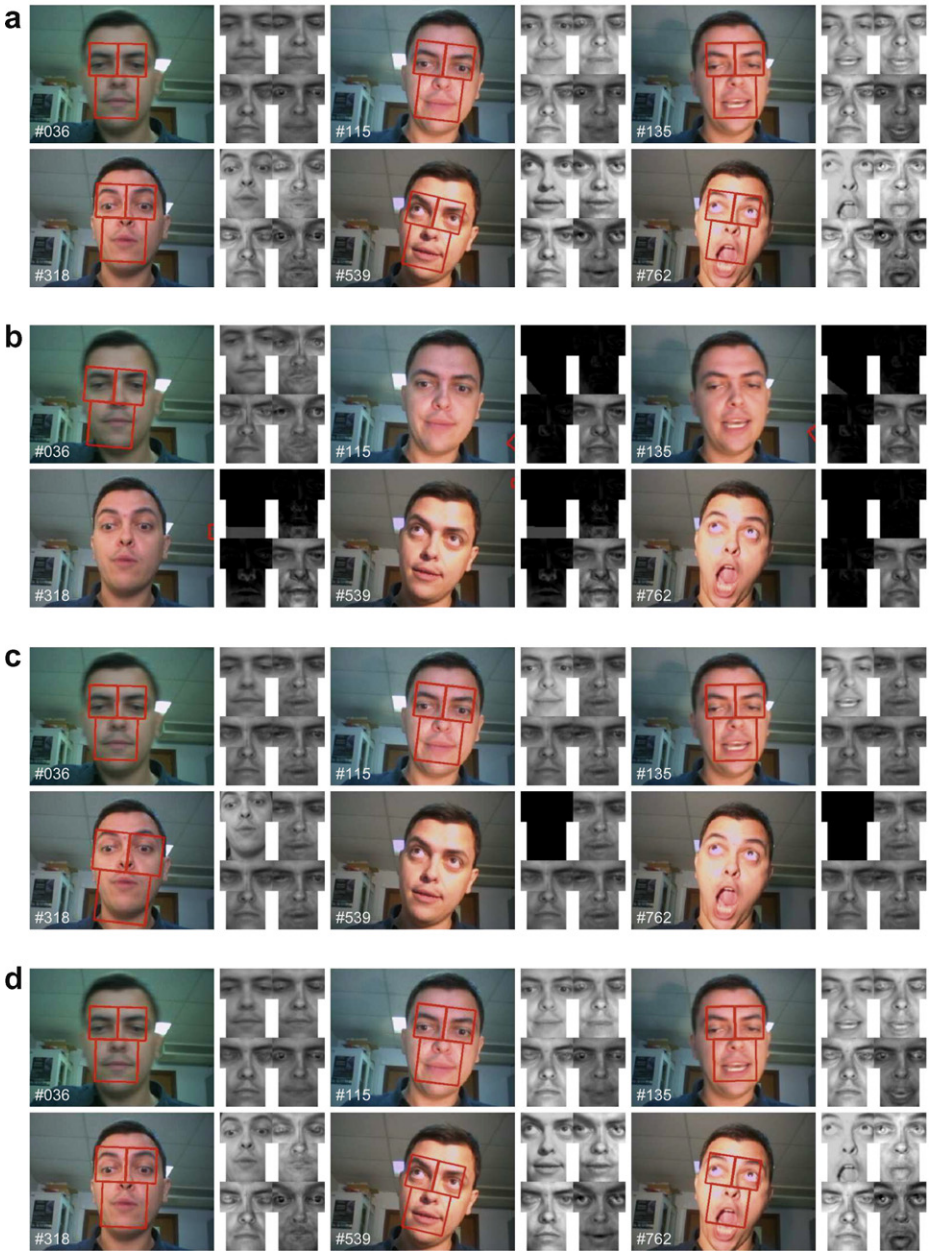


Fig. 18. Sample images comparing minimisation algorithms with the second test image sequence. (a) Tracking with OUA. (b) Tracking with MBC. (c) Tracking with HBA. (d) Tracking with COC.

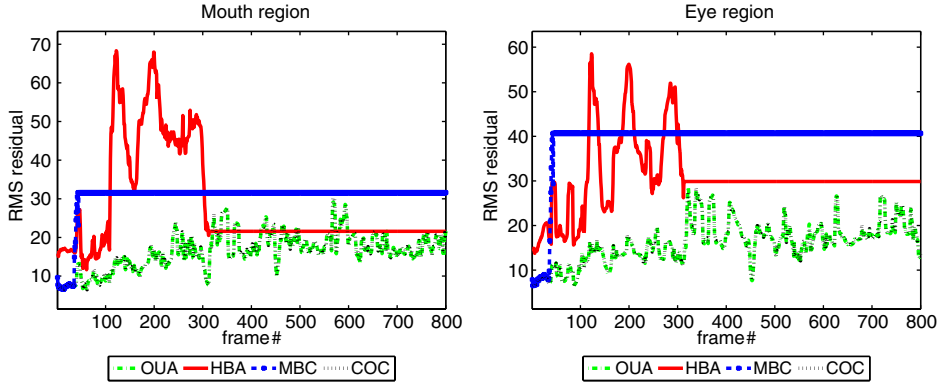


Fig. 19. RMS residual comparison for the second test image sequence.

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015

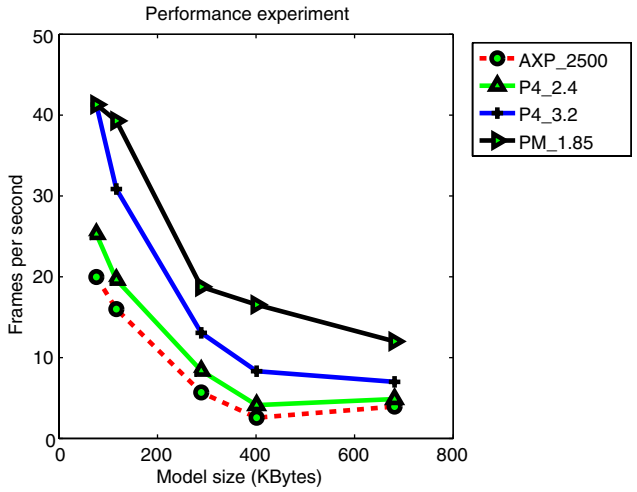


Fig. 20. Tracker performance for different processors and for different model sizes.

is subject to all possible illuminations and another set in which the face, under one illumination, adopts all facial expressions. This result is important for the development of simple and efficient face trackers, which have an immediate application in facial expression recognition, performance-based graphical animation or video-based face recognition systems.

We have also introduced an efficient procedure for fitting linear subspace-based appearance models. It is an extension of Hager and Belhumeur's factorisation-based additive approach [3] for a deforming face subject to strong illumination variations. In our procedure, we make no assumptions about the smoothness of the linear subspace basis. Although it was introduced in the context of appearance-based models, it could be used as well in other linear approaches, such as Active Appearance Models or Morphable Models.

We have also reviewed all previous efficient linear subspace-based fitting procedures and we have proved that Matthews and Baker's compositional approach [34] is based on a smoothness assumption on the linear subspace which is equivalent to that used in Muñoz at al.'s compositional approach [13], and to the one used in Hager and Belhumeur's additive approach [3]. These assumptions were conceived to alleviate the computational cost involved in the minimisation. That is why approximate methods (HBA, MBC and APC) have the lowest computational cost in the experiments conducted. This gain, nevertheless, comes at the price of an important drop in the convergence rate which is approximately one fifth of that associated with correct algorithms, like OUA and COC, for difficult images. The other major drawback of approximate methods is that the smoothness assumption biases the minimisation solution. This bias had been previously reported by Romdhani and Vetter [46], although it has been properly explained here.

Correct algorithms (COC and OUA) are the best choice if accuracy and convergence rate are the key issues in an image alignment application. If computational cost is also important, then the OUA algorithm introduced in this paper is the best choice. If, on the other hand, computational cost is the main concern and we are ready to sacrifice accuracy and convergence rate, then HBA, Hager and Belhumeur's algorithm, is the best choice.

There seems to be some consensus across part of the computer vision community concerning the superiority of the ICIA algorithm [38,35,34] to Hager and Belhumeur's factorisation-based additive approach [3]. Here we have proved that a correct factorisation-based additive approach, like the one introduced in this paper, has better convergence properties than ICIA. Moreover, even if it also involves a smoothness assumption, as is the case of Hager and Belhumeur's approach [3], the convergence properties (convergence rate, final RMS error and number of iterations per frame) are still better than those of ICIA. One additional argument against factorisation-based approaches was that they could not be used with typical motion models, such as homographies. In Section A.3, by deriving the factorisation of a homography-based motion model for our fitting algorithm, we prove that factorisation-based algorithms can be used with homographies. A similar result for Hager and Belhumeur's basic planar tracking algorithm had been presented in [47].

Acknowledgements

This work was funded by Grants TRA2005-08529-C02-02 from the Spanish *Ministerio de Educación y Ciencia* and R05/10454 from the *Universidad Politécnica de Madrid*.

Appendix A. Some Jacobian factorisation examples

Here we will show how the Jacobian matrix factorisation introduced in Section 3.5.1 applies to some motion models commonly used in computer vision.

A.1. Rotation, translation and scale model (RTS)

This motion model can be described by four parameters, $\boldsymbol{\mu} = (t_u, t_v, \theta, s)$, corresponding to rotation, translation and scale, $f(\mathbf{x}, \boldsymbol{\mu}) = s\mathsf{R}(\theta)\mathbf{x} + \mathbf{t}$, where $\mathbf{x} = (u, v)^\top, \mathbf{t} = (t_u, t_v)^\top$ and $\mathsf{R}(\theta)$ is a 2D rotation matrix.

Taking derivatives of f with respect to $\mathbf{x}$ and $\boldsymbol{\mu}$,

$$f_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\mu}) = s\mathsf{R}(\theta), \quad f_{\boldsymbol{\mu}}(\mathbf{x}, \boldsymbol{\mu}) = \left[\mathsf{I}_{2\times 2} \middle| -s\mathsf{R}(\theta)\begin{bmatrix} -v \\ u \end{bmatrix} \middle| \mathsf{R}(\theta)\begin{bmatrix} u \\ v \end{bmatrix}\right], \tag{A.1}$$

where the $\mathsf{I}_{d\times d}$ is the $d \times d$ identity matrix. Introducing the derivatives in (A.1) into (26), we get the factorisation:

$$\Gamma(\mathbf{x}_i) = \left[\mathsf{I}_{2l\times 2l}, \begin{bmatrix} -v_i\mathsf{I}_{l\times l} & u_i\mathsf{I}_{l\times l} \\ u_i\mathsf{I}_{l\times l} & v_i\mathsf{I}_{l\times l} \end{bmatrix}\right], \quad \Sigma(\mathbf{c}, \boldsymbol{\mu}) = \begin{bmatrix} \mathsf{C}\frac{1}{s}\mathsf{R}(-\theta) & 0 \\ 0 & \mathsf{C}\begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{s} \end{bmatrix} \end{bmatrix},$$

where $l = k + 1$ and k is the dimension of the subspace. For this model M_0 has dimensions $N \times 4l$ and $\Sigma, 4l \times 4$.

A.2. Affine model

The 2D affine motion model can be written as $f(\mathbf{x}, \boldsymbol{\mu}) = \underbrace{\begin{bmatrix} a & c \\ b & d \end{bmatrix}}_{\mathsf{A}}\mathbf{x} + \begin{bmatrix} e \\ f \end{bmatrix}$, where A is a non-singular matrix and $\boldsymbol{\mu} = (e, f, a, b, c, d)^\top$ are the six model parameters. Taking derivatives of f with respect to $\mathbf{x}$ and $\boldsymbol{\mu}$,

$$f_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\mu}) = \mathsf{A}, \quad f_{\boldsymbol{\mu}}(\mathbf{x}, \boldsymbol{\mu}) = [\mathbf{I}_{2\times 2}|u\mathbf{I}_{2\times 2}|v\mathbf{I}_{2\times 2}]. \tag{A.2}$$

where the $\mathbf{I}_{d\times d}$ is the $d \times d$ identity matrix. From (A.2) and (26), we get the factorisation we are looking for

$$\Gamma(\mathbf{x}_i) = [\mathbf{I}_{2k\times 2k}|u_i\mathbf{I}_{2k\times 2k}|v_i\mathbf{I}_{2k\times 2k}], \quad \Sigma = \begin{bmatrix} \mathsf{CA}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathsf{CA}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathsf{CA}^{-1} \end{bmatrix},$$

where M_0 has dimensions $N \times 6k$ and Σ $6k \times 6$.

A.3. Projective model

A sufficient condition for factoring the Jacobian $\mathsf{M}(\boldsymbol{\mu}, \mathbf{c})$ in (27) is that f be linear. Here we choose to work with homogeneous coordinates to assure a linear warping function. Let $\mathbf{x} = (u, v)^\top$ and $\tilde{\mathbf{x}} = (r, s, \lambda)^\top$ be the cartesian and projective coordinates of an image pixel, respectively. They are related by a projection function $\mathbf{x} = p(\tilde{\mathbf{x}}) : \mathscr{P}^2 \to \Re^2$ such that:

$$\tilde{\mathbf{x}} = \begin{bmatrix} r \\ s \\ \lambda \end{bmatrix} \to \mathbf{x} = p(\tilde{\mathbf{x}}) = \begin{bmatrix} r/\lambda \\ s/\lambda \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}; \lambda \neq 0.$$

In this case, the brightness constancy constraint (1) takes the form

$$I(p(f(\tilde{\mathbf{x}}, \boldsymbol{\mu}_t)), t) = I(p(\tilde{\mathbf{x}}), 0) + [\mathsf{B}\mathbf{c}_t](p(\tilde{\mathbf{x}})) \quad \forall \tilde{\mathbf{x}} \in \mathscr{F}. \tag{A.3}$$

The warping function that describes the motion of a planar region is a 2D homography

$$f(\tilde{\mathbf{x}}, \boldsymbol{\mu}) = \mathsf{H}\tilde{\mathbf{x}} = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & 1 \end{bmatrix} \begin{bmatrix} r \\ s \\ \lambda \end{bmatrix}, \tag{A.4}$$

where $\boldsymbol{\mu} = (a, b, c, d, e, f, g, h)^\top$. Now, $\mathsf{B}_\nabla(\tilde{\mathbf{x}}_i)$ has an extra set of columns associated with the gradient of the homogeneous coordinate,

$$\mathsf{B}^p_\nabla(\tilde{\mathbf{x}}_i) = \left[\begin{bmatrix} \nabla_r \mathbf{I}(p(\tilde{\mathbf{x}}_i), 0) \\ \nabla_r[\mathbf{b}_1](p(\tilde{\mathbf{x}}_i)) \\ \vdots \\ \nabla_r[\mathbf{b}_k](p(\tilde{\mathbf{x}}_i)) \end{bmatrix}^\top, \begin{bmatrix} \nabla_s \mathbf{I}(p(\tilde{\mathbf{x}}_i), 0) \\ \nabla_s[\mathbf{b}_1](p(\tilde{\mathbf{x}}_i)) \\ \vdots \\ \nabla_s[\mathbf{b}_k](p(\tilde{\mathbf{x}}_i)) \end{bmatrix}^\top, \begin{bmatrix} \nabla_\lambda \mathbf{I}(p(\tilde{\mathbf{x}}_i), 0) \\ \nabla_\lambda[\mathbf{b}_1](p(\tilde{\mathbf{x}}_i)) \\ \vdots \\ \nabla_\lambda[\mathbf{b}_k](p(\tilde{\mathbf{x}}_i)) \end{bmatrix}^\top \right]^\top \tag{A.5}$$

and matrix C is

$$\mathsf{C}^P = \begin{bmatrix} [1\mathbf{c}^\top] & \mathbf{0}^\top & \mathbf{0}^\top \\ \mathbf{0}^\top & [1\mathbf{c}^\top] & \mathbf{0}^\top \\ \mathbf{0}^\top & \mathbf{0}^\top & [1\mathbf{c}^\top] \end{bmatrix}^\top.$$

The derivatives of f w.r.t $\tilde{\mathbf{x}}$ and $\boldsymbol{\mu}$ are

$$\frac{\partial f(\tilde{\mathbf{x}}, \boldsymbol{\mu})}{\partial \tilde{\mathbf{x}}} = H \tag{A.6}$$

$$\left.\frac{\partial f(\tilde{\mathbf{x}}, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\boldsymbol{\mu}_t} = \begin{bmatrix} r\mathsf{I}_{3\times 3} & |s\mathsf{I}_{3\times 3}| & \begin{matrix} \lambda \mathsf{I}_{2\times 2} \\ 0_{1\times 2} \end{matrix} \end{bmatrix}, \tag{A.7}$$

where $0_{p\times q}$ is a $p \times q$ matrix with all elements equal to zero. The derivative of the image grey values w.r.t. the homogeneous coordinates are

$$\nabla_{\tilde{\mathbf{x}}} I(p(\tilde{\mathbf{x}})) = \frac{\partial I(p(\tilde{\mathbf{x}}))}{\partial \tilde{\mathbf{x}}} = \left[\left.\frac{\partial I(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=p(\tilde{\mathbf{x}})} \right]^\top \left[\frac{\partial p(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}} \right].$$

Since

$$\frac{\partial p(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}} = \begin{bmatrix} \frac{1}{\lambda} & 0 & -\frac{r}{\lambda^2} \\ 0 & \frac{1}{\lambda} & -\frac{s}{\lambda^2} \end{bmatrix},$$

and given that the coordinates $\tilde{\mathbf{x}}$ in the brightness constancy Eq. (A.3) are defined on a finite image patch, we can safely assume that $\lambda = 1$. Then

$$\nabla_{\tilde{\mathbf{x}}} I(p(\tilde{\mathbf{x}})) = \left[\frac{\partial I(\mathbf{x})}{\partial u}, \frac{\partial I(\mathbf{x})}{\partial v}, -r\frac{\partial I(\mathbf{x})}{\partial u} - s\frac{\partial I(\mathbf{x})}{\partial v} \right]^\top.$$

Then, M is factored from (A.5), (A.6), (A.7) and (26) as follows:

$$\Gamma(\mathbf{x}_i) = [r_i \mathsf{I}_{3l\times 3l} | s_i \mathsf{I}_{3l\times 3l} | \lambda_i \mathsf{I}_{3l\times 3l}], \Sigma = \begin{bmatrix} \mathsf{C}^P \mathsf{H}^{-1} & 0 & 0 \\ 0 & \mathsf{C}^P \mathsf{H}^{-1} & 0 \\ 0 & 0 & \mathsf{C}^P \mathsf{H}^{-1}_{1-2} \end{bmatrix},$$

where H^{-1}_{1-2} is the matriz composed of the first two columns of H^{-1} and again $l = k + 1$. Now the dimensions of M_0 and Σ are $N \times 9l$ and $9l \times 8$, respectively.

References

[1] R. Gross, I. Matthews, S. Baker, Generic vs. person specific active appearance models, in: Proc. British Machine Vision Conference, 2004, pp. 457–466.
[2] A. Gee, R. Cipolla, Fast visual tracking by temporal consensus, Image and Vision Computing 14 (2) (1996) 105–114.
[3] G. Hager, P. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (10) (1998) 1025–1039.
[4] S. Basu, I. Essa, A. Pentland, Motion regularization for model-based head tracking, in: Proc. International Conference on Pattern Recognition, vol. III, 1996, pp. 611– 16.
[5] M. La Cascia, S. Sclaroff, V. Athitsos, Fast, reliable head tracking under varying illumination: an approach based on robust registration of texture-mapped 3d models, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (4) (2000) 322–336.
[6] D. Terzopoulos, K. Waters, Analysis and synthesis of facial image sequences using physical and anatomical models, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (6) (1993) 569–579.
[7] M.J. Black, Y. Yacoob, Recognizing facial expressions in image sequences using local parameterized models of image motion, International Journal of Computer Vision 25 (1) (1997) 23–48.
[8] F. de la Torre, M.J. Black, Robust parameterized component analysis: applications to 2d facial modeling, in: Proc. ECCV (4), vol. 2353 of LNCS, Springer, 2002, pp. 653–669.
[9] K.-C. Lee, D. Kriegman, Online learning of probabilistic appearance manifolds for video-based recognition and tracking, in: Proc. CVPR, vol. I, 2005, pp. 852–859.
[10] A. Rahimi, B. Recht, T. Darrel, Learning appearance manifolds from video, in: Proc. CVPR, vol. I, 2005, pp. 868–875.
[11] V. Blanz, T. Vetter, A morphable model for the synthesis of 3d faces, in: Proc. SIGGRAPH, ACM Press, 1999, pp. 187–194.
[12] T. Cootes, G. Edwards, C. Taylor, Active appearance models, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 681–685.
[13] E. Muñoz, J.M. Buenaposada, L. Baumela, Efficient model-based 3d tracking of deformable objects, in: Proc. International Conference on Computer Vision, vol. I, Beijing, China, 2005, pp. 877–882.
[14] F. Dornaika, J. Ahlberg, Fast and reliable active appearance model search for 3d face tracking, Transactions on SMC-B 34 (4) (2004) 1838–1853.
[15] T. Cootes, G. Edwards, C. Taylor, Active appearance models, in: Proc. European Conference on Computer Vision, vol. LNCS 1047, Springer-Verlag, 1998, pp. 484–498.
[16] D. DeCarlo, D. Metaxas, Optical flow constraints on deformable models with applications to face tracking, International Journal of Computer Vision 38 (2) (2000) 99–127.
[17] V. Blanz, T. Vetter, Face recognition based on fitting a 3d morphable model, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (9) (2003) 1–12.
[18] S. Baker, I. Matthews, J. Schneider, Automatic construction of active appearance models as an image coding problem, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (10) (2004) 1380–1384.
[19] J. Lim, D.A. Ross, R.-S. Lin, M.-H. Yang, Incremental learning for visual tracking, in: L.K. Saul, Y. Weiss, L. Bottou (Eds.), Advances in Neural Information Processing Systems, vol. 17, MIT Press, Cambridge, MA, 2005, pp. 793–800.
[20] D. Ross, J. Lim, M.-H. Yang, Adaptive probabilistic visual tracking with incremental subspace update, in: Proc. European Conference on Computer Vision, vol. LNCS 3022, Springer-Verlag, 2004, pp. 470–482.
[21] K. Toyama, A. Blake, Probabilistic tracking with exemplars on a metric space, International Journal of Computer Vision 48 (1) (2002) 9–19.
[22] H. Fei, I. Reid, Joint bayes filter: a hybrid tracker for non-rigid hand motion recognition, in: Proc. European Conference on Computer Vision, vol. 3023 of LNCS, 2004, pp. 497–508.
[23] A. Elgammal, R. Duraiswami, L.S. Davis, Probabilistic tracking in joint feature-spacial spaces, in: Proc. CVPR, vol. I, 2003, pp. 781–788.
[24] O. Williams, A. Blake, R. Cipolla, A sparse probabilistic learning algorithm for real-time tracking, in: Proc. International Conference on Computer Vision, vol. I, 2003, pp. 353–360.
[25] R. Basri, D.W. Jacobs, Lambertian reflectance and linear subspaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2) (2003) 218–233.
[26] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 643–660.
[27] A. Shashua, T. Riklin-Raviv, The quotient image: class-based re-rendering and recognition with varying illuminations, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2) (2001) 129–139.
[28] J.B. Tenenbaum, W.T. Freeman, Separating style and content with bilinear models, Neural Computation 12 (2000) 1247–1283.

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015

[29] D. Grimes, A. Shon, R. Rao, Probabilistic bilinear models for appearance-based vision, in: Proc. International Conference on Computer Vision, vol. II, 2003, pp. 1478–1485.
[30] M. Vasilescu, D. Terzopoulos, Multilinear analysis of image ensembles: tensorfaces, in: Proc. European Conference on Computer Vision, vol. LNCS 2350, Springer-Verlag, 2002, pp. 447–460.
[31] A. Elgammal, C.-S. Lee, Separating style and content on a non-linear manifold, in: Proc. CVPR, vol. I, 2004, pp. 478–485.
[32] M.A.O. Valilescu, D. Terzopoulos, Multilinear independent component analysis, in: Proc. CVPR, vol. I, 2005, pp. 547–553.
[33] M.J. Black, A.D. Jepson, Eigentracking: robust matching and tracking of articulated objects using a view-based representation, International Journal of Computer Vision 26 (1) (1998) 63–84.
[34] I. Matthews, S. Baker, Active appearance models revisited, International Journal of Computer Vision 60 (2) (2004) 135–164.
[35] S. Baker, I. Matthews, Lucas-Kanade 20 years on: a unifying framework, International Journal of Computer Vision 56 (3) (2004) 221–255.
[36] Z. Khan, T. Balch, F. Dellaert, A Rao-Blackwellized particle filter for eigentracking, in: Proc. CVPR, vol. II, 2004, pp. 980–986.
[37] K. Murphy, S. Russel, Rao-Blackwellised particle filtering for dynamic bayesian networks, in: A. Doucet, N. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer-Verlag, 2001, pp. 499–515.
[38] S. Baker, I. Matthews, Equivalence and efficiency of image alignment algorithms, in: Proc. CVPR, vol. 1, IEEE, 2001, pp. 1090–1097.
[39] R. Gross, I. Matthews, S. Baker, Active appearance models with occlusion, Image and Vision Computing 24 (6) (2006) 593–604.
[40] J.M. Buenaposada, E. Muñoz, L. Baumela, Efficient appearance-based tracking, in: Proc. CVPR-Workshop on Nonrigid and Articulated Motion, IEEE, 2004.
[41] J.M. Buenaposada, E. Muñoz, L. Baumela, Efficiently estimating facial expression and illumination in appearance-based tracking, in: Proc. British Machine Vision Conference, vol. I, 2006, pp. 57–66.
[42] F. Jurie, M. Dhome, Hyperplane approximation for template matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 996–1000.
[43] H.-Y. Shum, R. Szeliski, Construction of panoramic image mosaics with global and local alignment, International Journal of Computer Vision 36 (2) (2000) 101–130.
[44] B.D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: Proc. Image Understanding Workshop, 1981, pp. 121–130.
[45] A. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspaces, in: Proc. CVPR, 1994, pp. 84–91.
[46] S. Romdhani, T. Vetter, Efficient, robust and accurate fitting of a 3d morphable model, in: Proc. International Conference on Computer Vision, vol. 1, 2003, pp. 59–66.
[47] J.M. Buenaposada, L. Baumela, Real-time tracking and estimation of plane pose, in: Proc. International Conference on Pattern Recognition, vol. II, IEEE, Quebec, Canada, 2002, pp. 697–700.
[48] A. Bartoli, Groupwise geometric and photometric direct image registration. IEEE Transactions on Pattern Analysis and Machine Intelligence, in press, doi:10.1109/TPAMI.2008.22.

Please cite this article in press as: J.M. Buenaposada et al., Efficient illumination independent appearance-based face tracking, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.015