Realistic Performance-driven Facial Animation using Hardware Acceleration

MANUEL A. SÁNCHEZ, JAMES D. EDGE and STEVE MADDOCK
Dept. of Computer Science
University of Sheffield
Technical Report CS-04-10

Whereas the use of Motion Capture (MoCap) revolutionised the workflow of full-body character animation, it has yet to perform an equivalent change for its facial counterpart. Recent contributions have applied MoCap to pose-based face models, bringing forth results of a quality unrivalled by purely marker-driven approaches. This paper describes an animation framework that attains similar effects with arbitrary face meshes, by means of a surface-driven warping technique that suits the deformation of facial skin. The small-scale effects of tissue mechanics, overlooked by the geometric warp, are derived from deformation analysis conducted on the control surface, in conjunction with a wrinkling model extracted from photographs of an actual performer. This method means that realistic dynamic features like creases and furrows can be produced in real time, using dedicated graphics hardware.

1. INTRODUCTION

Over the last decade, marker-based Motion Capture (MoCap) systems have significantly changed the way character animation is produced, introducing a high degree of automation and realism into the animation process. This has been achieved by allowing retargeted MoCap data to directly drive the deformation of the body, using skeletal warping techniques specially suited for this particular context.

However, the use of Motion Capture hasn't had such an appreciable impact on the workflow of Facial Animation, which still remains a rather laborious process. This is principally due to facial expression being such a sensitive process for human perception. For this reason the bulk of the Computer Graphics industry is reluctant to adopt fully automated methods, as they could result in implausible renditions of highly detailed face models, and prefer instead artist-driven procedures. In this context, pose-based techniques have become the method of choice, by providing a rather intuitive control paradigm for the artist. At the same time they guarantee high quality for the most easily identifiable facial expressions, representing them with individually modelled poses.

Following this trend, recent research has applied morphable models of faces [Lanitis et al. 1994; Blanz and Vetter 1999] to the extraction of pose space parameters from video of actual performances: examples include the original work by Pighin [Pighin et al. 1999], later extended along with Joshi et al. in [Joshi et al. 2003]. Both these approaches succesfully use region segmentation to enrich the dimensionality of their pose space, extrapolating small local deformations from a reduced set of blend shapes. This approach may reduce by an order of magnitude the amount of poses required for a plausible face model (usually counted in the hundreds), but it still relies heavily on the quality of blend shapes and correct specification of regions.

This work was supported by the ESPRC and Fundación Pedro Barrié de la Maza.
Authors' addresses: M. Sánchez, J. Edge and S. Maddock. Dept. of Computer Science, University of Sheffieldm Regent Court, 211 Portobello St., Sheffield S1 4DP, U.K. e-mails: {m.sanchez, j.edge, s.maddock}@dcs.shef.ac.uk

Whereas Joshi's work presents a promising automatic solution for the segmentation problem, based on observation of mechanic properties of the tissue, pose-based frameworks benefiting from these improvements don't enjoy the self-containment of the previously described full-body systems.

Fully MoCap-driven approaches, pioneered in [Williams 1990], do not attain comparable degrees of quality to those of their pose-based counterparts. Neither techniques using spatial interpolation of marker displacements (e.g. [Noh et al. 2000] and [Kshirsagar et al. 2001]) nor volume deformation methods [Escher et al. 1998; Tao and Huang 1998] manage to convey the complex behaviour of the facial tissue, up to the level of realism that accurate blend shapes do. This is especially evident in the presence of dynamic skin features such as bulges and furrows, of key relevance for gesture perception. Blend shape techniques effectively portray them using either morphable texture models or extremely fine pose meshes, while wrinkling is generally ignored by marker-based facial MoCap, due to the small scale of these features compared to the density of tracked markers.

The compromise between quality and labour requirements essentially balances out these two animation paradigms; however interest in the academic community seems biased towards pose-based models, either because of the impressive results, or for its adoption by the CGI industry. Recently, the work in [Liu et al. 2001] has proposed using ratio images to improve the realism of image-warped photographs. We attempt to vindicate the use of marker-driven methods by extending this two-tiered concept to animated 3D faces, in order to achieve more realistic results than the approaches commented in the previous paragraph. Firstly, we describe a warping method consistent with the large-scale geometric aspects of skin motion, and later we complement it with a hybrid model of dynamic wrinkling, combining image-based procedures with deformation analysis.

2. GEOMETRIC WARPING

In the vast field of deformation techniques for polygonal meshes, different paradigms have related the warping of the initial figure to simpler control elements, ranging from points [Arad et al. 1994] to curves [Lazarus et al. 1994; Singh and Fiume 1998], volumes [Sederberg and Parry 1986; Moccozet and Thalmann 1997] and even other polygonal meshes [Singh and Kokkevis 2000; Bloomenthal 2002] (for a more detailed classification please refer to [Milliron et al. 2002])]. However these generic methods may not accommodate the particular nature of the deformation exhibited by facial skin: motion is driven by muscular traction, and transmited along the elastic continuum of the tissue, while dampened by its connection to underlying layers of fat and bone. These tissue properties are reflected in physically-based techniques, prominently in [Lee et al. 1995] and [Koch et al. 1998], but this comes at the cost of large computation times that do not scale linearly with respect to the complexity (number of elastic elements) of the face model.

The large-scale aspects of skin deformation, however, can be replicated by purely geometric approaches. Many of the methods described before operate by interpolating the displacement of controlling features across the Euclidean space the model is immersed in, or along a parameterisation of volumetric partitions, and this does not correspond with the way motion propagates across the connectivity of the facial tissue. Using the Riemannian metrics of the model, or interpolating displacement along an approximating surface would more accurately reflect the physical fact. Also, apart from the areas in which elasticity is compromised (e.g. crease lines), skin preserves its overall smoothness, requiring the warp formulation to enforce a certain degree of *continuity*. Finally, the dampening in skin traction has to reflect on the *locality* of the deformation, restricting it to a neighbourhood of the actuating control elements.

In [Krishnamurthy and Levoy 1996] the authors introduce the use of scalar fields mapped over a piecewise B-Spline surface to approximate range scan data. The projective paradigm intrinsic to this modelling approach can be otherwise used to drive the deformation of a polygonal model with an approximating parametric surface, as proposed before. Such a control surface is defined over a piecewise set of domains spanning

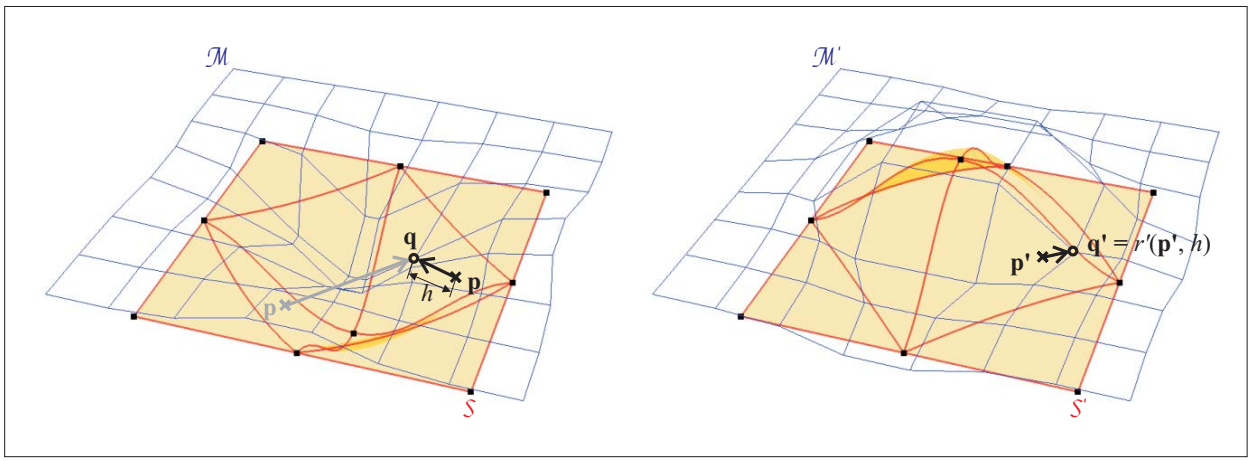


Fig. 1. Projection and reconstruction of a given vertex in a polygonal mesh (M) deformed in application of BIDS. The control surface S is constructed as a piecewise composition of Bézier Triangles spanning the markers (black dots).

the markers used by the MoCap, and its construction is conditioned by the continuity and locality properties we want to enforce in the deformation algorithm, as we shall describe next.

2.1 Bézier-Induced Deformation of Surfaces (BIDS)

The algorithm outlined before operates by exploiting the relation between the polygonal mesh M we want to deform and a control surface S that approximates it, composed of Bézier Triangles. Once the position of the markers change, spanning a new control surface S', the deformed mesh M' is computed by reversing the mapping that relates M to the initial configuration of S, and then evaluating it over S'. In order to further describe this mapping, let's first consider the application r, relating tuples of points of S and elements of the real line to a subset of $\mathbb{E}^3$ we will refer to as the *normal coverage* of the control surface:

$$\begin{aligned} r : S \times \mathbb{R} &\rightarrow \mathscr{C}(S) \subseteq \mathbb{E}^3 \\ r(\mathbf{p}, h) &= \mathbf{p} + h\, U(\mathbf{p}) \end{aligned} \tag{1}$$

considering U as the *unitary normal field* of S.

Depending on the shape of the control surface, there could be points of $\mathbb{E}^3$ onto which no scaled normal vector of S is incident, causing its normal coverage $\mathscr{C}(S)$ not to encompass the whole Euclidean space. Also, it must be noted that r is not necessarily injective, since, as Fig. 2.1 illustrates (left), there can be different points of the control surface mapped onto the same element of $\mathscr{C}(S)$.

Bearing this in mind, we can define the inverse mapping of r, denoted by $p : \mathscr{C}(S) \rightarrow S \times \mathbb{R}$, as the result of projecting the points of $\mathscr{C}(S)$ over the control surface along its normal field. Given the non-injective nature of r, the projection will be ill-defined in those points where its inverse mapping is not one-to-one.

Constraining $p(\mathbf{q})$ with $\mathbf{q} \in \mathscr{C}(S)$ to be the tuple with minimum magnitude for h partially solves this problem, disambiguating the projection for all points but those equidistant to different locii on S. Yet we are really only concerned about the restriction of p to $M \cap \mathscr{C}(S)$, so this limitation will be relevant only if the equidistance condition is met on the polygonal mesh itself. Testing this by brute force would result in a rather intensive computation, but fortunately it can be done much more efficiently by just checking

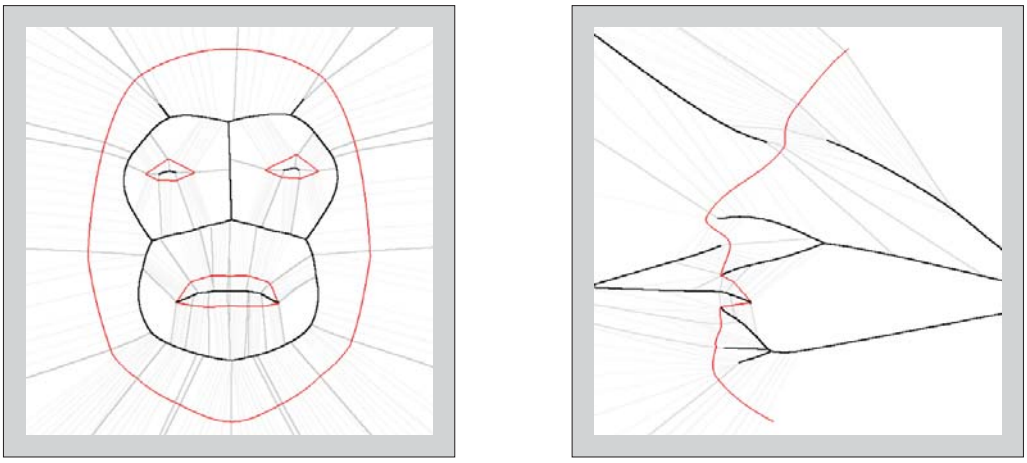


Fig. 2. Approximation of medial curves (in black) by concatenation of contiguous Voronoi cells. Given the solution for the Voronoi problem in $\mathbb{E}^3$, the construction of medial surfaces is analogous.

for intersections between M and approximations of the *medial surface* of S (see [Bloomenthal 2002] and [Yoshizawa et al. 2003]).

Provided that the polygonal mesh is exempt of these singularities with respect to the control surface, the projection p thus defined describes a non-degenerate mapping of $M \cap \mathscr{C}(S)$ onto S. Also, its inverse r (henceforth referred to as r' when evaluated on S') allows reconstruction of the deformed image of M over different configurations of the control mesh it was originally projected onto. This allows expressing the deformation of a vertex $\mathbf{q}$ of M into its image $\mathbf{q}'$ as the following procedure:

$$\begin{aligned} \{\mathbf{p}, h\} &= p\,(\mathbf{q}) \\ \mathbf{p}' &= \mathbf{s}'_i\left(\mathbf{s}_i^{-1}\,(\mathbf{p})\right) \\ \mathbf{q}' &= r'\,(\mathbf{p}', h) = \mathbf{p}' + h\,U'(\mathbf{p}') \end{aligned} \tag{2}$$

where $\mathbf{s}_i$ and $\mathbf{s}'_i$ are the parametric expressions for the neutral and deformed configurations, respectively, of the Bézier Triangle in S where the initial projection of $\mathbf{q}$ lies on.

2.2 Deformation continuity

By construction, the parametric expression $\mathbf{s}_i$ for any Bézier Triangle in S is differentiable over the interior of its (barycentric) domain of definition D_i, provided that its control net is non-degenerate. Therefore r is also differentiable over $D_i^o \times \mathbb{R}$. Under these conditions, it can also be proved that r is regular for all those points in its domain whose image lies out of the medial surface of $\mathbf{s}_i$ (henceforth $\mathscr{E}(S_i)$, with $S_i = \mathbf{s}_i(D_i^o)$). Consequently, by application of the inverse mapping theorem, p is a C^∞ diffeomorphism over $\mathscr{C}(S_i) \setminus \mathscr{E}(S_i)$. Hence we can conclude that the composition implicit in Eq. 2 is ultimately differentiable over this same domain.

Whenever several triangular patches exert a combined deformation, p restricts the region actually affected by each of them by ruling out intersections amongst different $\mathscr{C}(S_j)$. With the polygonal mesh guaranteed not to intersect $\mathscr{E}(S)$, we still have to verify whether the deformation is continuous across the boundaries shared between regions corresponding to adjacent Bézier Triangles. Assuming a uniformly varying normal over S,

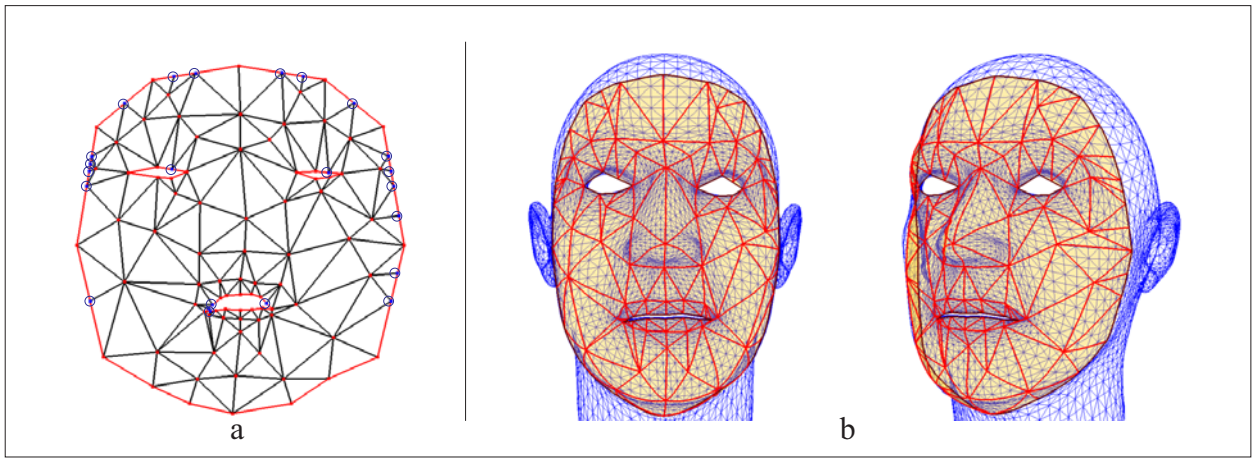


Fig. 3. a) Delaunay triangulation of the projected markers, highlighting the vertices inserted while enforcing countours (mouth and eye openings, along with the control mask boundary). b) Resulting control surface with the same topology, overlayed onto the model whose deformation it drives.

this verification becomes equivalent to assessing continuity for those points being projected on boundaries between adjacent patches.

Since these boundaries are themselves images of the boundary of their respective parametric domains, we need a common parameterisation of the incident patches to study the degree of continuity for the components of Eq. 2. Enforcing geometric continuity of degree k means that adjacent patches can be reparameterised so that the resulting expression is C^k continuous (see [Boehm 1988] and [Peters 2002]). This would also lead to a C^{k-1} definition of the normal vector field, implying that p is diffeomorphic of class C^{k-1} over the combined domain corresponding to the incident patches. Following identical reasoning to that used for a single patch, we can finally ascertain that the whole deformation attains this same degree of continuity.

Summarizing, for a control surface where adjacent Bézier Triangles meet with G^k continuity, the deformation induced over $M \cap \mathscr{C}(S)$ will be at least C^{k-1} continuous, provided that $M \cap \mathscr{E}(S) = \emptyset$. This will determine the way Bézier control nets will be constructed over an arrangement of triangular patches whose topology is yet to be defined. The procedure followed for this definition is described in the next section.

2.3 Constructing the topology of triangular domains

For a given set of markers on M, a triangulation has to be found so that the resulting polyhedron best approximates this polygonal mesh, while preserving the connectivity implicit in it. In this particular case where M represents a face, we can extrapolate a suitable solution from the Delaunay triangulation on $\mathbb{R}^2$, by using cyclindrical or ellipsoidal projections that do not carry a significant metric distortion. If such projections are not suited (e.g. when dealing with non-human faces) a relaxation approach like the one used in [Viaud and Yahia 1992] could be used. Additionally, to enforce symmetry in the deformation, we may choose to mirror the triangulation produced for the markers on one half of the face over the other.

In order to mimic the connectivity of M, we should be able to specify boundaries so that openings in the mesh can be represented on the control surface (and consequently regarded as discontinuities by the deformation). This is done by defining closed polylines connecting the markers that lie on the corresponding boundary; the resulting segments are preserved in the triangulation by means of forced insertion of midpoints in the Delaunay mesh, as described in [Lawson 1977]. Then the orientation of boundaries is used to remove

the triangles that lie inside the openings, through a simple propagation scheme on the graph defined by face adjacency.

However, the addition of new points to the set of markers, as part of the segment-preserving triangulation process, brings up the issue of how to provide control input for them throughout the deformation process, consistently with the rest of points whose displacements will be known. Our solution is to parameterise them over the Bézier curve corresponding to the domain edge that is split by their insertion, and derive their location by a top-down evaluation of the hierarchy of edge partitions, computing the midpoint of the curve corresponding to each segment. This is possible since edge curves are constructed prior to the patches themselves, as discussed in 2.4.

2.4 Conditioning of the control surface

Constructing a piecewise G^n continuous surface over a topology of domains like the one described in the previous subsection is a recurrent topic in Surface Fairing, and, in general, in Computer-Aided Design. The work in [Peters 1990] provides a classification of methods solving this problem for surfaces composed of Bézier patches.

From this point we will concentrate on the case of cubic Bézier Triangles connecting with G^1 continuity that, according to the previous analysis in 2.2, allows us derive a differentiable deformation for all vertices of M, except for those being projected on patch boundaries. In this last case the deformation is C^0 continuous.

Following the comments in 2.3, we aim to allow the (cubic) edge curves of these patches to be specified prior to the construction of the patches themselves. This is so not only for the sake of enabling the boundary-preserving triangulation approach, but also to introduce more flexibility in outlining the shape of these boundaries, especially when using sparse sets of markers. There are, however, specific constraints that these curves must comply with, because of being embedded in a G^1 continuous surface. Particularly, the internal points of their Bézier net must lie on the tangent plane of their respective vertices, as illustrated in Fig. 4.

As for the construction of the rest of the patch, many of the approaches compiled in [Peters 1990] provide G^1 conditioning solutions for piecewise surfaces composed of triangular Bézier patches whose cubic boundary curves are known. To achieve compatibility amongst the constraints imposed by G^1 continuity accross each edge, these methods either relax the problem by partitioning the patches adding free (partially unconstrained) vertices, or blend the control nets prescribed by the continuity conditions on each edge while raising the degree of the expression of the whole patch. Our method of choice, Clough-Tocher partitioning [Clough and Tocher 1965], belongs to the group adopting the first of these two options: it splits every triangular domain into three by adding a new vertex at the barycenter. The placement of this domain vertex, together with that of the other undefined Bézier control points, is then determined by G^1 conditions across the boundaries shared by the three *micropatches*; for a more detailed discussion consult [Clough and Tocher 1965] and [Farin 1986].

While the use of a partitioning scheme carries a drawback in terms of the number of patches to proccess, it also allows us to derive simpler expressions for the computations over S that our implementation of BIDS require, such as the projection of vertices of M. As result, their corresponding numerical approximations are faster and generally better behaved that those constructed in the blending case.

2.5 Implications on locality

As illustrated by Fig. 4, the conditioning scheme we use relies on knowing the tangent plane of the surface at every marker. As usual, the latter is computed by averaging the normals of incident faces from the polyhedron constructed in 2.3. Implicitly, this expands the effects of displacing one marker from the set of incident *macropatches* to all those adjacent to any of them. Hence the degree of locality attained with the present scheme.

Should we provide a higher degree of continuity for the algorithm, by raising that in the control surface to G^2, curvature information would be needed at every marker. The approximated normal at connected

Fig. 4. Conditioning for G^1 continuity in vertex boundaries

For a set of Bézier Triangles to meet with G^1 continuity at a given vertex $\mathbf{p}_i$, their respective control nets have to be conditioned so all the Bézier points directly connected to that vertex lie on the same plane.

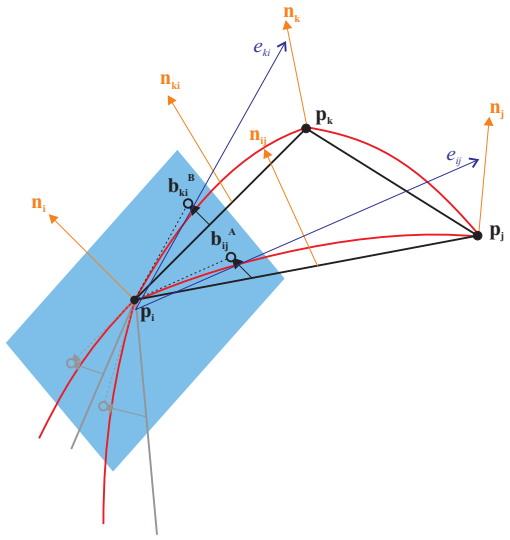
For each triangle with corner points $\{\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k\}$ this condition can be met by deriving the internal points ${\mathbf{b}_{ij}}^A$ and ${\mathbf{b}_{ki}}^B$ from the following expression, after substituting the corresponding indices:

$${\mathbf{b}_{ml}}^B = {\mathbf{b}_{lm}}^A = \frac{1}{3}\left(2\,\mathbf{p}_l + \mathbf{p}_m + \frac{(\mathbf{p}_m - \mathbf{p}_l)\cdot\mathbf{n}_l}{{\mathbf{n}_{lm}}^A \cdot \mathbf{n}_l}\,{\mathbf{n}_{lm}}^A\right)$$

where the interpolated normal ${\mathbf{n}_{ml}}^A$ is given as:

$${\mathbf{n}_{ml}}^B = {\mathbf{n}_{lm}}^A = B_0^2\left(\frac{1}{3}\right)\mathbf{n}_l + B_1^2\left(\frac{1}{3}\right)\mathbf{n}_{lm} + B_2^2\left(\frac{1}{3}\right)\mathbf{n}_m$$

with B_i^j standing for the i^{th} Berstein polynomial of degree j. Vertex and edge normals ($\mathbf{n}_m$ and $\mathbf{n}_{ml}$, respectively) are computed by averaging face normals on the polyhedron constructed in 2.3.



In the case of cubic Bézier Triangles, the previous procedure fully determines the boundary curves of the patches. Should we require more flexibility in specifying them (e.g. to fit a particular contour in the mesh being approximated), internal control points ${\mathbf{b}_{ij}}^A$ and ${\mathbf{b}_{ki}}^B$ can be given small offsets in the subspace spanned by e_{ij} and e_{ki}, mantaining the coplanarity condition along deformation.

markers could be then used to compute this data, and as a result locality would expand one more level across the topology of adjacent patches. This illustrates the balance existing between these two aspects of the deformation, acting in detriment of each other.

3. WRINKLING MODEL

Wrinkling is a very representative factor in facial expression, and as such it has been extensively treated by numerous researchers. Early contributions, like [Viaud and Yahia 1992], provided purely synthetic methods that relied on particular modelling requirements. Subsequent development of physically-based face models supported the simulation of wrinkles on arbitrary polygonal meshes [Wu et al. 1994; Wu et al. 1999], while pose-based systems included them as part of the facial texture model [Pighin et al. 1999]. Each of these methods uses different ways of representing wrinkles: either with actual geometry, bump maps, or photographic textures. Portraying furrows and creases with polygonal detail can be quite inefficient, because of their small scale in relation to movement of the whole face. Alternatively normal maps, or the equivalent bump maps [Blinn 1978], produce a comparable visual effect while reducing the impact on rendering times. Yet photographic pose-based texture models can achieve more realistic results, but they are bound to particular lighting conditions, dramatically restraining their scope of application.

While normal maps stand clearly as a suitable method for statically representing the wrinkling of facial skin, the model we aim to construct has to also accommodate its dynamic nature. For this purpose the actual mechanisms driving the wrinkling process have to be taken into consideration. As elastic material, skin presents a significant resistance to compression, while sustaining bend and shear with much more ease. Also, the elastic properties of the tissue are not homogeneous: successive plastic deformation causes

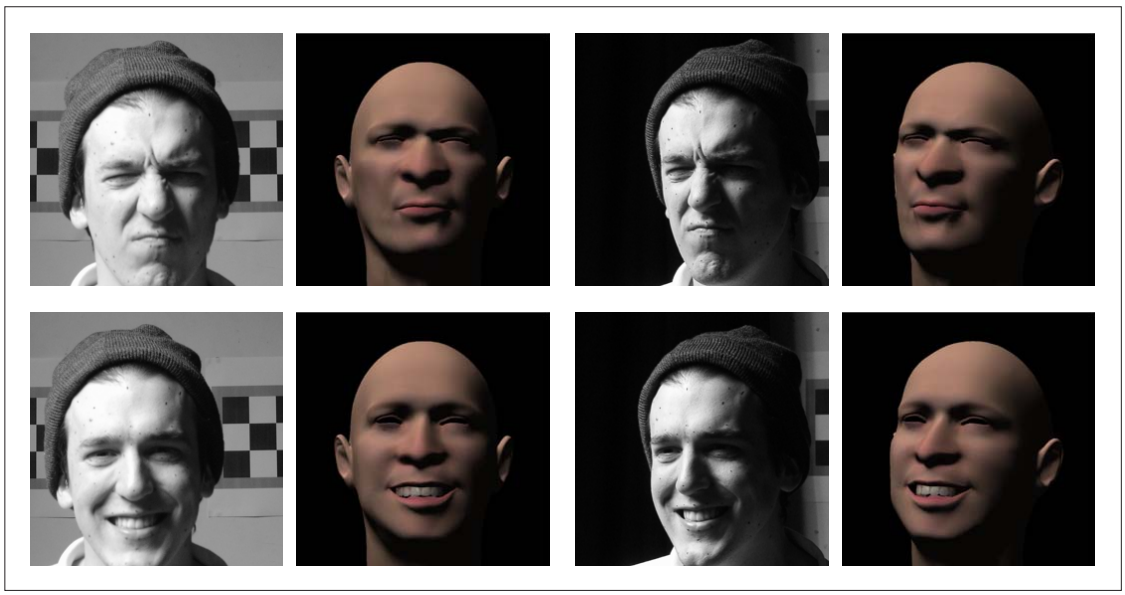


Fig. 5. A face model deformed using BIDS to represent expressions extracted from photographs. Markers are retargetted through the RBF approach from [Sanchez et al. 2003]. The lack of skin wrinkling on the synthetic model makes it look far less realistic than the photographs.

reorientation of the collagen fibres along crease lines, so that the elastic response to compression in a particular direction becomes much stiffer. Ultimately, this causes the surrounding tissue to buckle leading to the familiar furrowing effect that we visually perceive.

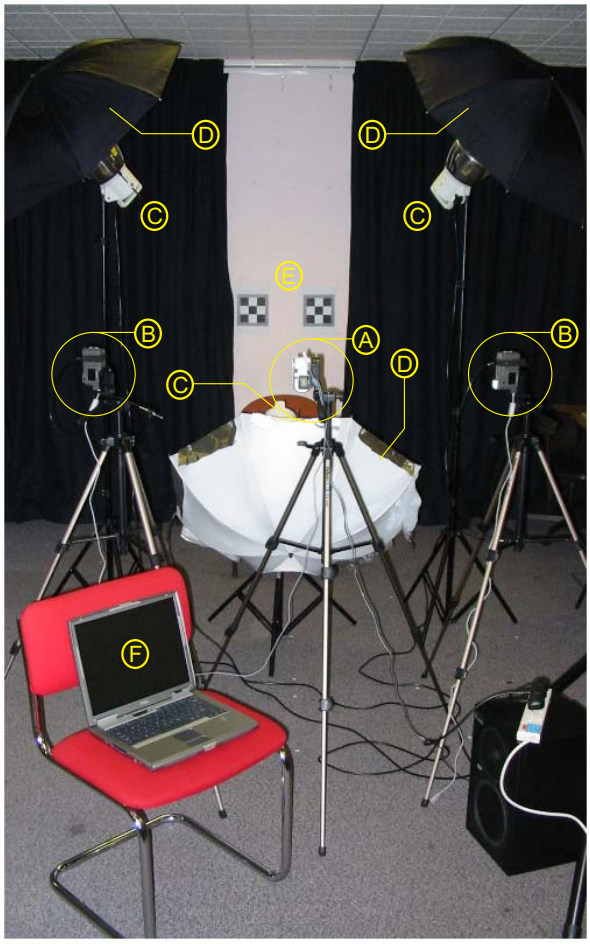
We can conclude that wrinkling is a localized phenomenon that appears in particular patterns, determined by the position of permanent crease lines (macrofurrows), and by the magnitude of compression (negative strain) in specific directions. Building a realistic model of this process, suitable for a physically-based approach, can prove to be quite a complicated task, since there is not an equivalent to typical strain-stress analysis procedures for living tissue.

However, should we use normal maps to represent the wrinkling, image-based techniques can be put into practice to capture their dynamic behaviour from an actual performer. Also, using this same input, a deformable model is fitted to the performer's face so that skin strain can be approximated and related to the results of the previous process. This finally leads to the definition of a functional model of the variations of the normal map, parameterised on the magnitude and direction of compression. Such a model can be applied on a geometrically warped face model to provide an accurate representation of the skin behaviour, retargeted from the observations on the performer.

3.1 Normal map extraction

Following the introduction of photometric stereo in [Woodham 1980], successive research has extensively used the Lambertian lighting model for the extraction of shape information from photographs. Rushmeier et al. were the first to introduce its application to the reconstruction of normal maps from an actual object [Rushmeier et al. 1998], using a fixed lighting set. More recent work [Debevec et al. 2000] has extracted full reflectance data from a human face, whose lighting properties differ greatly from the Lambertian model. These results, combined with pointwise separation of diffuse and specular components, provide a better

Fig. 6. Basic setup used for the captures



A) Sony DCSP10.5 Megapixels (MP) camera

B) Olympus Camedia D510 2.1 MP camera, controlled remotely

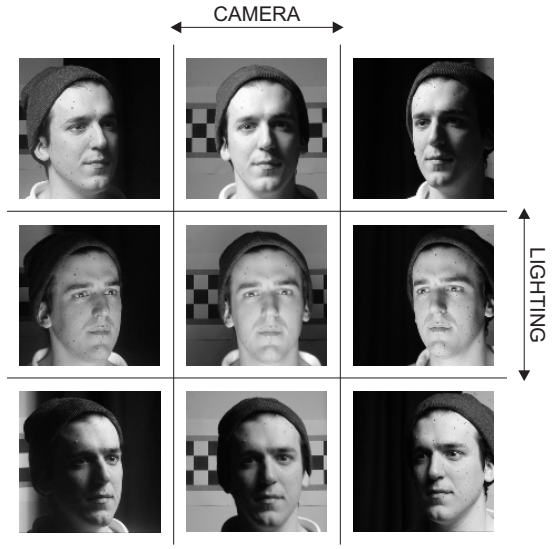
C) Interfit 3200K kit, fitted with 275W tungsten bulbs

D) 50cm. ∅ silver reflectors

E) Calibration patterns

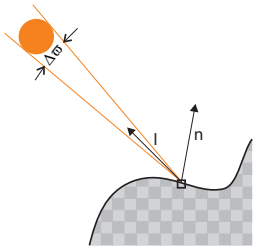
F) Laptop used to collect the data

Combining these elements we produce 9 different takes for every single pose, fed as input to the algorithm computing the normal map:



estimate of the normal field on the skin surface. However, the articulated lighting system used in this approach is an overly-strong requirement for our purposes, leading us to construct our capture set around the simpler paradigm described by Rushmeier.

In the case of a Lambertian surface illuminated by a source with radiance L_o, the reflected radiance L_r at a given point is defined by the following equation:



$$L_r = \rho \, (L_o \, \Delta\omega / \pi) \, n \cdot l \tag{3}$$

where ρ is the Lambertian reflectance (albedo) at that point, l a unitary vector in the direction of incidence of the light, and n the corresponding surface normal.

We can assume the solid angle $\Delta\omega$ to be approximately constant for all points in the surface, provided that the light source is sufficiently distant in comparison with the scale of the object being illuminated. Taking the constant $L'_o = L_o \, \Delta\omega / \pi$ as the effective radiance emitted by any light source in the system, we can then use three of them to derive a straightforward approximation of the surface normal as in Eq. 3.

$$\begin{bmatrix} n^x \\ n^y \\ n^z \end{bmatrix} = \frac{1}{\rho L'_o} \begin{bmatrix} l_1^x & l_1^y & l_1^z \\ l_2^x & l_2^y & l_2^z \\ l_3^x & l_3^y & l_3^z \end{bmatrix}^{-1} \begin{bmatrix} L_{r,1} \\ L_{r,2} \\ L_{r,3} \end{bmatrix} \tag{4}$$

In Eq. 4, $L_{r,i}$ stands for the reflected radiance observed when only the light source with incidence vector l_i is active. This is recovered from three photographs of a given facial pose with different illumination.

In our particular capture set, we arrange the three lights along the edges of a squashed tetrahedron whose apex lies on the face, enforcing an optimal spacing of 90° amongst each of them [McGunnigle 1998]. To compensate for the non-Lambertian properties of human skin we apply a thin layer of matte foundation on the face of the performer, which also makes the albedo much more even. This, in addition to the use of scattering reflectors on the light sources, helps to attenuate significantly the specular component of skin reflectance (alternatively, light polarizers could be used both on lights and cameras to perform this separation).

Additionally, in order to evaluate the normal field retrieved as a normal map over the skin surface, we have to provide an approximate model for the latter. In the absence of a range scanner or equivalent equipment, we proceed by fitting a generic model to a set of markers placed on the performer, by means of the BIDS algorithm. The three-dimensional location of these markers is determined by reverse stereoscopy from the input of a rig of three concurrent cameras. While this approximation of the skin geometry is not accurate enough to extract static normal maps, it allows us to measure the variations of the normal field between different poses, and expressing it on the tangent space of the corresponding point on the model. Since these are ultimately the results we need to build our functional model, no further refinement of the fitting is required in practice.

3.2 Analysis of the deformation

The purpose of marking the performer's face is not only to provide a fitted model to help compute the normal map, as discussed in the previous section, but also to analyse the deformation sustained by the skin in each of the poses captured. Given the relatively small scale of this deformation, an adequate measure can be found in the *infinitesimal strain tensor*, introduced by Cauchy in the classical theory of elasticity. It represents not only the magnitude of strain, but also its directionality, that is specially relevant for the wrinkling process.

Since we are only interested in the deformation along the surface of the skin, it is sensible to evaluate this tensor just on the planar subspace tangent to every given point. Then it takes the form:

$$\begin{bmatrix} \epsilon_{uu} & \gamma_{uv} \\ \gamma_{uv} & \epsilon_{vv} \end{bmatrix} \text{ with } \begin{cases} \epsilon_{uu} = \mathbf{u}^u_{,u} \\ \epsilon_{vv} = \mathbf{u}^v_{,v} \\ \gamma_{uv} = 1/2\,(\mathbf{u}^u_{,v} + \mathbf{u}^v_{,u}) \end{cases} \tag{5}$$

where $\mathbf{u}$ is the displacement function with respect to the reference configuration (given by a neutral pose), expressed in the tangent subspace of the point where it is evaluated. The operator $\bullet_{,i}$ stands for derivation in the coordinate system $\{e_u, e_v\}$ of that subspace.

Giving a physical interpretation to the components of this tensor, ϵ_{uu} and ϵ_{vv} can be considered as the linear strains along the direction of e_u and e_v, respectively. γ_{uv} is instead an angular measure, representing the shearing between these two axes. However, this is not yet a suitable representation of the deformation for our particular purposes, since we want to isolate compression from elongating, positive strain that does not reflect on the local wrinkling process. Conveniently, the symmetric bilinear form from Eq. 5 admits eigen-decomposition into *characteristic directions of strain*, without shearing. The magnitude of the strain in these directions can be then clamped to $(-\infty, 0]$, so that when the eigen-decomposition is reverted only elements of compression will appear in the tensor.

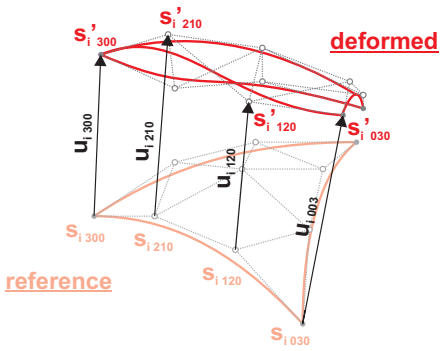


Fig. 7. Bézier net of the displacement interpolant for a given patch $\mathbf{s}_i$.

Were we to evaluate compression on the fitted model as described before, we would lack of an analytic definition for the displacement function $\mathbf{u}$ that could be differentiated at a given point to compute the strain tensor. However, such definition can be derived for points on the control surface used by BIDS, just by evaluating the corresponding patch expression with the difference vectors relating the neutral and deformed configurations of the Bézier net (see Fig. 7). Therefore we proceed by approximating the strain on the fitted model with that sustained by the BIDS control surface, something that is logical given the nature of the warping algorithm that ultimately drives the deformation.

As a basis of the planar tangent subspace where strain is computed, we use the curvilinear coordinate system given by $\{e_u, e_v\} = \{\mathbf{s}^i_{,u}, \mathbf{s}^i_{,v}\}$ with $\mathbf{s}^i_{,l} = \frac{\partial \mathbf{s}^i}{\partial l} - \frac{\partial \mathbf{s}^i}{\partial w}$ for every patch $\mathbf{s}^i$ of S. With this particular choice, together with the definition of the displacement function outlined before, the computation of the strain tensor (Eq. 5) becomes straightforward using the terms of the Jacobian of $\mathbf{u}$ in that basis, as given by the following expression:

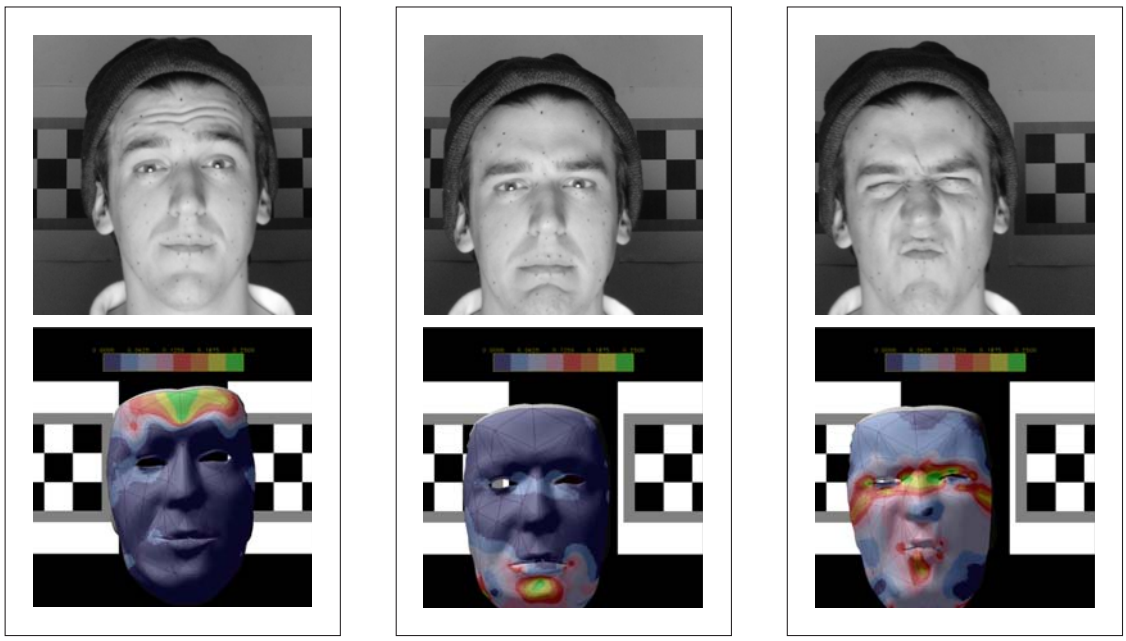


Fig. 8. Strain diagrams showing the modulus of compression, computed as the determinant of the strain tensor over the face model fitted to photographs of the subject used to build our wrinkling model.

$$J\mathbf{u} = \begin{bmatrix} \mathbf{u}^u_{,u} & \mathbf{u}^u_{,v} \\ \mathbf{u}^v_{,u} & \mathbf{u}^v_{,v} \end{bmatrix} = \begin{bmatrix} e^u \\ e^v \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{u}^i}{\partial u} - \frac{\partial \mathbf{u}^i}{\partial w} \\ \frac{\partial \mathbf{u}^i}{\partial v} - \frac{\partial \mathbf{u}^i}{\partial w} \end{bmatrix}^T \tag{6}$$

where e^u and e^v are the dual vectors in $\mathbb{R}^3$ of e_u and e_v, respectively, and $\mathbf{u}^i$ is the parametric expression of the displacement on the i^{th} patch.

Finally, in order to assemble the mosaic of strain information collected from each individual patch, we use the projection introduced in 2.3 to construct a single atlas for S, and obtain a continuous diagram of strain under this new parameterisation, after applying the appropriate coordinate changes to the mixed tensor in Eq. 6. This also allows us to express the resulting compression in the texture space induced by the projection, something that will prove very helpful in the next section.

3.3 Functional model

Provided that normal maps are expressed in the same texture space as the compression data collected from the control surface, we can build a polynomial approximation to the normal maps based on the components of the strain tensor (after clamping off positive values in the characteristic directions). At a given point in the texture, the approximated k^{th} component of the normal map will be expressed as:

$$\hat{n}^k = \sum_l \alpha_{k,l} \left(\epsilon_{ss}\right)^l + \sum_m \beta_{k,m} \left(\epsilon_{tt}\right)^m + \sum_n \lambda_{k,n} \left(\gamma_{st}\right)^n \tag{7}$$

with the strain tensor now referred to texture coordinates (s, t), as discussed before. Mixed terms are

discarded since we model the components of strain as non-intercorrelated variables.

Coefficients $\alpha_{k,l}$, $\beta_{k,m}$ and $\lambda_{k,n}$ can be estimated from the collection of normal maps generated for every individual pose, in conjunction with their respective compression diagrams, using least squares. This requires fixed ranges for indices l, m, n. Also, since normal maps are computed as the difference of the normal field in local tangent space with respect to the neutral configuration, there is no constant term, meaning that $\alpha_{k,0} = \beta_{k,0} = \lambda_{k,0} = 0$.

Therefore, for $C = \left[\ \alpha_{k,1} \ \cdots \ \alpha_{k,L},\ \beta_{k,1} \ \cdots \ \beta_{k,M},\ \lambda_{k,1} \ \cdots \ \lambda_{k,N}\ \right]$ representing the coefficients of Eq. 7 at a given texture sample, we can solve the least squares estimate by:

$$C^T = \left(R + \sum_i S_i^T S_i \right)^{-1} \sum_i n_i^k \, S_i^T \tag{8}$$

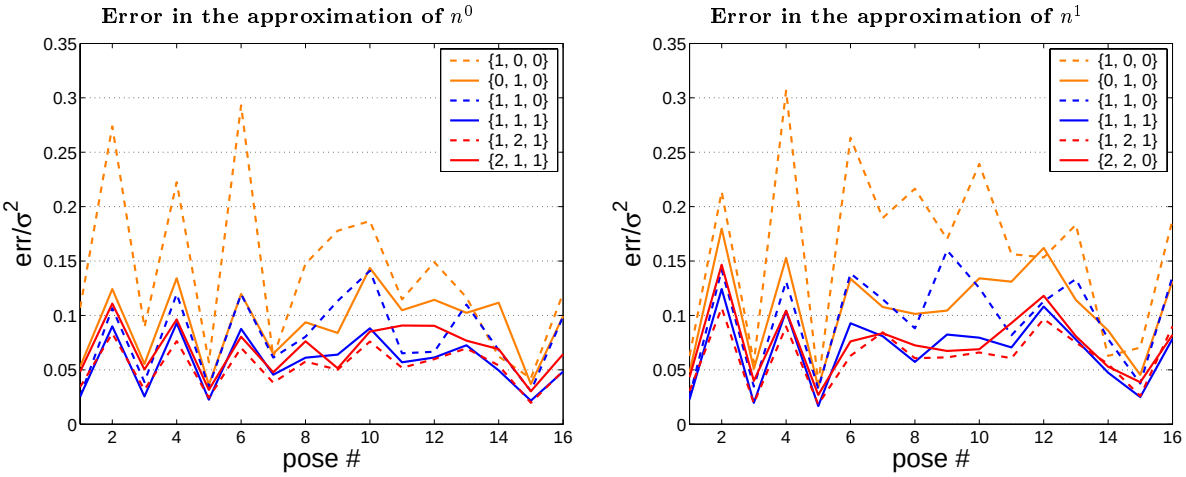
taking S_i as the vector of strain powers $\left[\ \epsilon_{ss} \ \cdots \ \epsilon_{ss}^L,\ \epsilon_{tt} \ \cdots \ \epsilon_{tt}^M,\ \gamma_{st} \ \cdots \ \gamma_{st}^N\ \right]$ measured in the i^{th} pose, for which the corresponding sample of the normal map is given by n_i. R stands for the usual regularization term in the form $diag(\epsilon)$ with $\epsilon \to 0$.

An appropiate metric for the accuraccy of a given approximation $\hat{n}^k(s,t)$ of the k^{th} component of normal map can be found in the expression of the *mean squared error*, whose terms are minimized by Eq. 8:

$$err = \frac{1}{WH} \sum_{s=1}^{W} \sum_{t=1}^{H} (\hat{n}^k(s,t) - n^k(s,t))^2 \tag{9}$$

where W and H are the dimensions of the normal map n^k. For estimating the scale of the error, this metric can be compared with the variance of the reference map itself (σ^2), as we subsequently do.

Provided a sampling space consisting of the normal maps for 16 different poses, we can determine the suitable degree of the approximating polynomial in Eq. 7 by considering the error/variance ratio for different ranges $(0..L), (0..M), (0..N)$ of the indices l, m, n, respectively. In the following plots these ranges are characterised by the tuple $\{L, M, N\}$:



These plots illustrate how rough linear approximations can be built on e_{ss} and e_{tt} alone ($\{1,0,0\}$ and $\{0,1,0\}$, repectively), but only the second shows reasonable convergence with more accurate models. This

Fig. 9. Partitioning by edge-boundary intersection

The intersecting point $\mathbf{q}$ of the edge with origin $\mathbf{o}$ and direction v verifies the following identity with respect to $\mathbf{p}$, its projection on the boundary curve along the *surface* normal n_s:

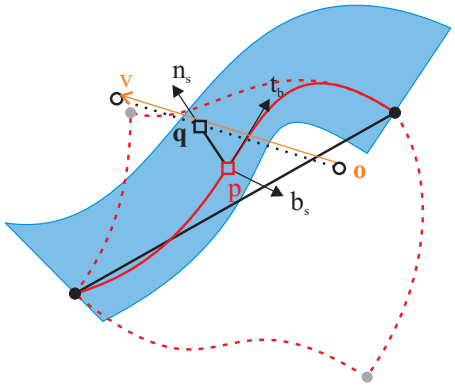
$$\mathbf{q} = \begin{cases} \frac{(\mathbf{p}-\mathbf{o})\cdot n_s}{v\cdot n_s} v + \mathbf{o} & \text{if } v \not\parallel n_s \\ \frac{(\mathbf{p}-\mathbf{o})\cdot b_s}{v\cdot b_s} v + \mathbf{o} & \text{if } v \perp n_s \end{cases}$$

Taking $p(t)$ as the parametric expression of this curve, and vectors t_b and b_s as $p_{,t}$ and $t_b \wedge n_s$, respectively, we can find t_0 so that $\mathbf{p} = p(t_0)$ as root of the equation:

$$(q_{,t} - t_b) \cdot (q - p) = 0$$

where

$$q_{,t} = \begin{cases} \frac{(n_{s,t}\cdot(p-\mathbf{o}))(v\cdot n_s) - (v\cdot n_{s,t})((p-\mathbf{o})\cdot n_s)}{(v\cdot n_s)^2} & \text{if } v \not\parallel n_s \\ \frac{(b_{s,t}\cdot(p-\mathbf{o}))(v\cdot b_s) - (v\cdot b_{s,t})((p-\mathbf{o})\cdot b_s)}{(v\cdot b_s)^2} & \text{if } v \perp n_s \end{cases}$$



The derivatives of the surface normal and binormal fields (n_s and b_s) involve multiple differential terms of the incident patches, omitted here for the sake of brevity.

An analytic solution for t_0 is not feasible due to the high degree of the resulting equations. However, even a simple numeric approximation like Newton-Raphson will converge for the case of cubic patches, provided that is fed with a reasonably good starting point (e.g. derived from a coarse sampling of the boundary curve).

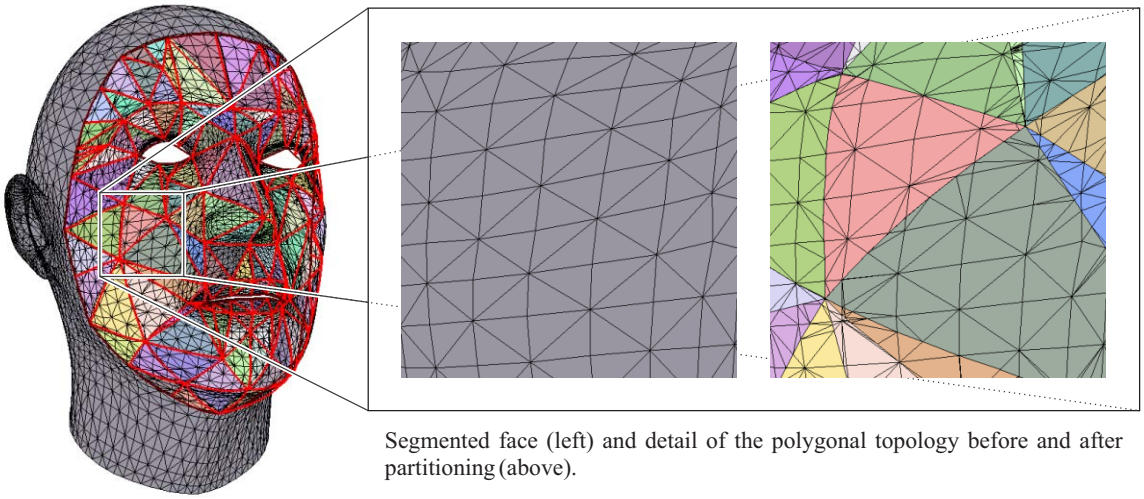
indicates that compression in the vertical direction of the texture (corresponding itself to verticality on the face) is the more predominant source of wrinkling. A linear expression of both terms of strain provides a lesser error, while including shearing (γ_{st}) in the model doesn't significantly improve accuracy, proving that this factor is not so relevant to the wrinkling process (as it could be expected from the elastic properties of the tissue).

Higher degree approximations, of which only $\{2, 2, 0\}$ and $\{1, 2, 1\}$ are shown here, hardly outperform the trilinear case. This can be explained as the process itself being quasi-linear on the components of the stain tensor. The residual error that bounds the performance of the models constructed responds to the portion of the normal map that is not related to the measured local compression; however, being such a small fraction of the variance ($1/20^{th}$), it is clearly negligible. Therefore we can conclude that the evolution of the normal map along deformation can be accurately represented by low-degree polynomial approximations, something that facilitates its implementation with the limited resources of dedicated hardware, as we will further discuss in the next section.

4. REAL-TIME CONSIDERATIONS

With the coefficients of the polynomial wrinkling model expressed in texture space, the normal map can be evaluated at any instant of the deformation with little computational cost. In fact, this can be done as part of the rasterization process, using the programmable features of the current generation of graphics accelerators. We choose to implement this evaluation as part of a fragment program in Cg, a propietary shading language for NVIDIA cards, whose sibling HLSL is supported by a wide range of manufacturers as part of the DirectX standard.

Two separate textures are used to store coefficients for the normal map components in the first two axes of the tangent space (usually named *tangent* and *binormal* in shading languages' literature, albeit referred to a surface). The need for a third component can be circumvented by renormalizing the resulting displaced normal, sparing space in redundant data. Using the four channels of the texture we can produce a quadratic



Segmented face (left) and detail of the polygonal topology before and after partitioning (above).

model in at least one of the components, while requiring only twice as much space as a usual static texture map. However, coefficients will have to be re-scaled to fit in the colour interval $[0, 1]$, and consequently lose precision. Using floating point textures can alleviate such problem, although this also carries heavier memory requirements.

The simple expressions for evaluation of Bézier Triangles allows embedding both BIDS deformation and strain analysis in a vertex shader, that will be executed at the transformation and lighting stage of the programmable rendering pipeline. This allows for geometry to be stored in the dedicated memory of the graphics card, without having to upload the results of the deformation for every single frame.

However, the previous technique also implies that geometry will have to be partitioned into batches with the same warping control, in this case comprising all vertices projected over the same *macropatch*. Since with current graphics architectures both topology and geometry have to be fed at the same time to the rendering pipeline before any stage is completed, we also have to split the faces spanning across different batches. This is performed by intersecting edges with the boundaries of contiguous $C(\mathbf{s}_i)$, spanned by the surface normal and tangent vector to any point on the corresponding boundary curve (see Fig. 4).

The subsequent increase in number of vertices (+11%) displayed in the diagram above may pose a disadvantage, but the benefits are having all the processing, apart from patch conditioning, done in dedicated hardware. On a AMD AtlhonXP 2600+ equipped with an entry-level NVIDIA GeForceFX 5200, the model of 12k vertices pictured above runs at ˜80 fps in a window of 640x640 pixels. Also, the next generation of graphic accelerators will include detached stages for topology processing in the pipeline, possibly removing this partitioning hinderance.

5. DISCUSSION OF RESULTS AND FUTURE WORK

The final results obtained from combining BIDS warping with the present wrinkling model are displayed in Fig. 10, contrasted with the original poses they reproduce (first column). In order to translate the position of markers from the face of the subject to the facial model employed, we used our RBF retargeting framework previously described in [Sánchez et al. 2003]. The second column of images depicts the analysis of strain conducted over the BIDS-deformed model in order to synthesise the normal map rendered in columns 3 and

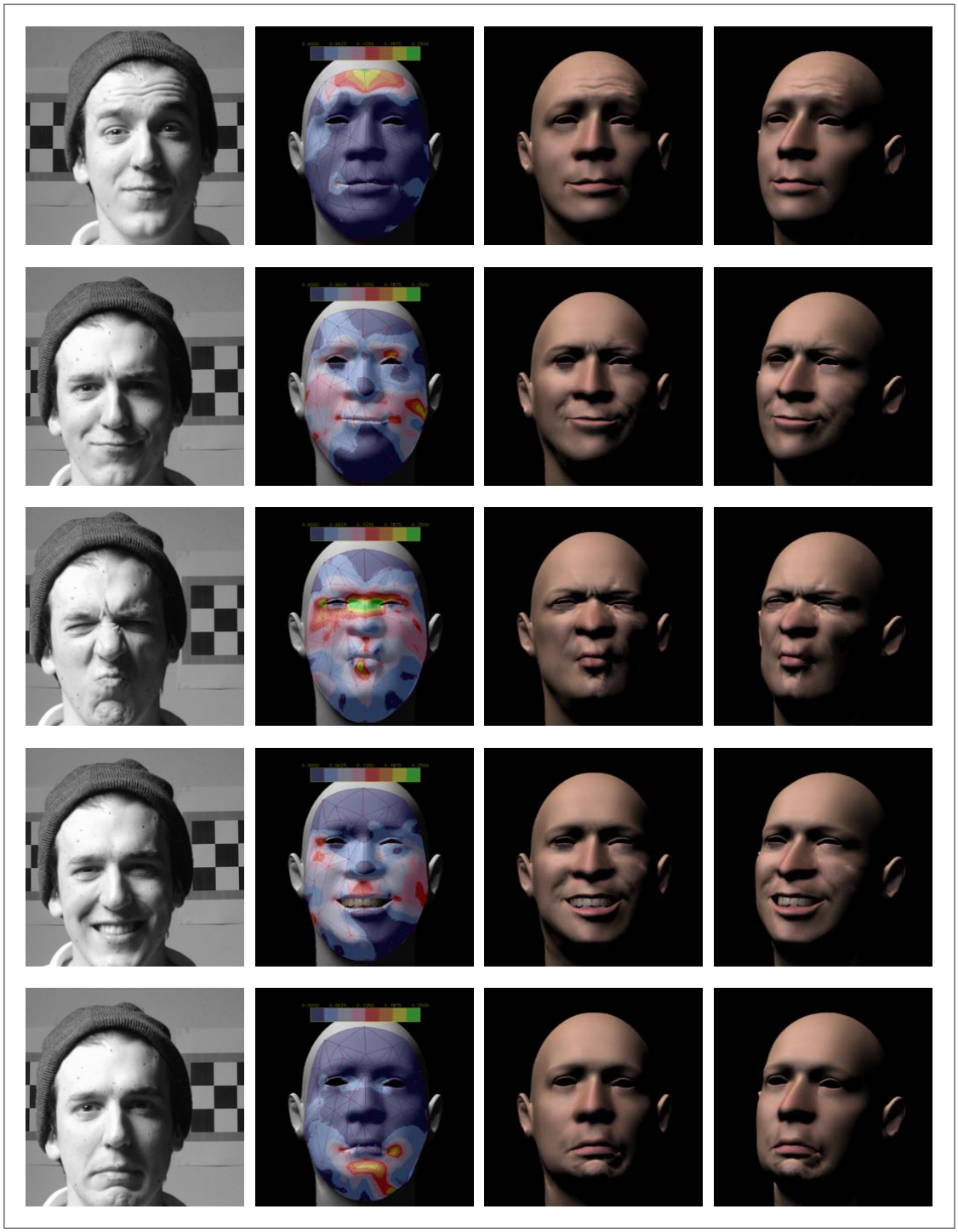


Fig. 10. Results obtained by combining BIDS deformation with the wrinkling model proposed in 3.3.
TR CS-04-10 Dept. of Comp. Science, Univ. of Sheffield, September 2004.

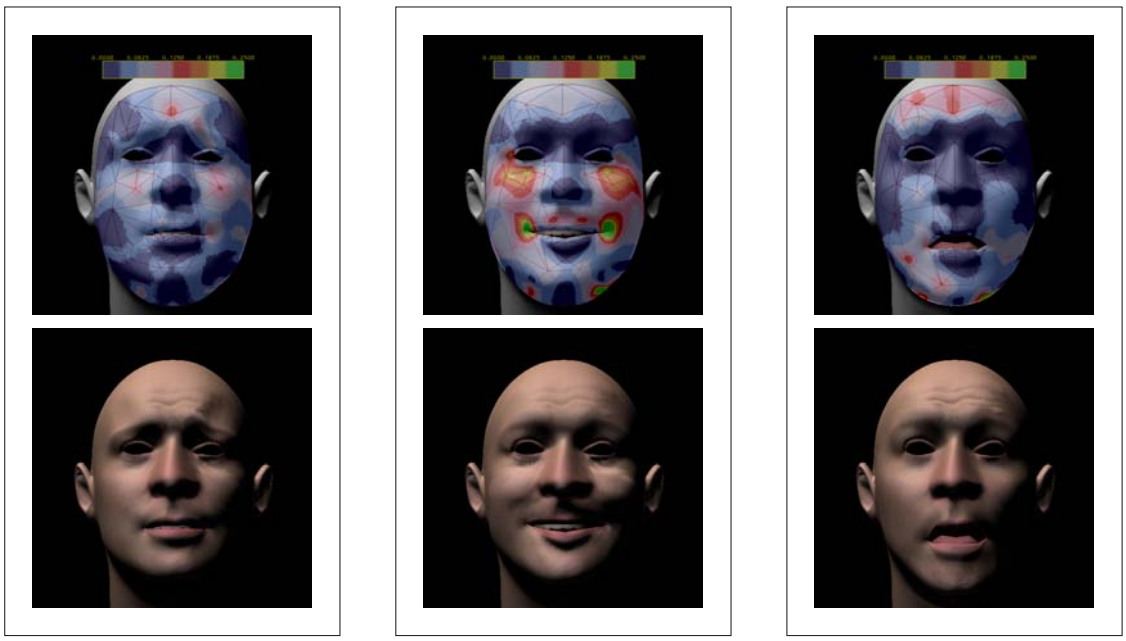


Fig. 11. Our method applied to Motion Capture data retargetted from a different subject from that used to construct the wrinkling model (MoCap courtesy of Scott King).

4. This process is fully animatable, since the deformation induced by any time-changing set of markers (e.g. MoCap) can be analysed in an identical way. Fig. 11 illustrates this.

In our opinion, these results are good for the standards of real-time content; even the models produced with just geometric warping are a similar standard to what an artist could produce, since the deformation paradigm itself mimics the usual B-Spline modelling techniques. However, in order to attain quality comparable to CGI production levels, more precise cameras would be needed to capture sharper and more detailed normal maps. This would ideally have to be combined with a finer distribution of markers, to better capture the localized strain that causes small-scale wrinkling. Also, an automated light setup, as found in most production facilities, would aid the capture of normal maps by preventing slight face shifts between the different takes of the same pose, something that we currently experience with our ad-hoc manual approach.

Another aspect to bear in mind is producing a better approximation of the performer's face geometry. This would not only improve the quality of representation of dynamic wrinkles, impoverished in areas where the fitting fails (like the upper nose in our examples), but also allow us to derive good static normal maps with which to represent static creases like those due to aging.

Whenever normal maps don't suffice, and geometric detail is needed for represention medium-scale furrowing (e.g. forehead wrinkles) we can proceed by integrating the low frequencies of the normal map, and represent them through subdivison and displacement mapping as in [Krishnamurthy and Levoy 1996].

Finally, it must be noted that a particular wrinkling model will always be bound to the subject captured, since there is no apparent way of directly combining several results, given the significant anatomical differences between individuals. The suitability of this for a given face model is really a matter of choosing the right performer. However, once a significant library of wrinkling models is built, our approach would allow the behaviour of the skin to be changed as easily as switching its texture.

REFERENCES

Arad, N., Dyn, N., Reisfeld, D., and Yeshurun, Y. 1994. Image warping by radial basis functions: Application to facial expressions. *Computer Vision, Graphics, and Image Processing. Graphical Models and Image Processing 56,* 2, 161–172.

Blanz, V. and Vetter, T. 1999. A morphable model for the synthesis of 3d faces. In *Siggraph 1999, Computer Graphics Proceedings*, A. Rockwood, Ed. Addison Wesley Longman, Los Angeles, 187–194.

Blinn, J. 1978. Simulation of wrinkled surfaces. In *Proceedings of SIGGRAPH 1978*. 286–293.

Bloomenthal, J. 2002. Medial-based vertex deformation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM Press, 147–151.

Boehm, W. 1988. Local smooth surface interpolation: a classification. *Computer-Aided Design 20,* 10, 370–372.

Clough, R. and Tocher, J. 1965. Finite element stiffness matrices for analysis of plates in blending. In *Proceedings of Conference on Matrix Methods in Structural Analysis*.

Debevec, P., Hawkins, T., Tchou, C., Duiker, H. P., Sarokin, W., and Sagar, M. 2000. Acquiring the reflectance field of a human face. In *Proceedings of SIGGRAPH 2000*.

Escher, M., Pandzic, I. S., and Magnenat-Thalmann, N. 1998. Facial deformations for mpeg-4. In *Proceedings of Computer Animation 1998, (CA'98)*. IEEE Computer Society, 56–.

Farin, G. 1986. Triangular bernstein-bézier patches. *Computer Aided Geometric Design 3*, 83–127.

Joshi, P., Tien, W. C., Desbrun, M., and Pighin, F. 2003. Learning controls for blend shape based realistic facial animation. In *Proceedings of the ACM SIGGRAPH Symposium on Computer Animation*.

Koch, R. M., Gross, M. H., and Bosshard, A. A. 1998. Emotion editing using finite elements. In *Proceedings of the Eurographics '98*. Lisbon, Portugal, 295–302.

Krishnamurthy, V. and Levoy, M. 1996. Fitting smooth surfaces to dense polygon meshes. In *Proceedings of SIGGRAPH 1996*. 313–324.

Kshirsagar, S., Garchery, S., and Magnenat-Thalmann, N. 2001. Feature point based mesh deformation applied to mpeg-4 facial animation. In *DEFORM/AVATARS, IFIP Conference Proceedings*. Vol. 196. Kluwer, 24–34.

Lanitis, A., Taylor, C. J., and Cootes, T. F. 1994. Automatic tracking, coding and reconstruction of human faces using flexible appearance models. In *IEE Electronic Letters*. IEE, 1578–1579.

Lawson, C. L. 1977. Software for c1 surface interpolation. *Mathematical Software III (John R. Rice, editor)*, 161–194.

Lazarus, F., Coquillart, S., and Jancène, P. 1994. Axial deformations: an intuitive deformation technique. *Computer-Aided Design 26,* 8 (Aug.), 607–613.

Lee, Y., Terzopoulos, D., and Waters, K. 1995. Realistic modeling for facial animation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM Press, 55–62.

Liu, Z., Shan, Y., and Zhang, Z. 2001. Expressive expression mapping with ratio images. In *Proceedings of SIGGRAPH 2001*. Los Angeles, US, 271–276.

McGunnigle, G. 1998. The classification of texture surfaces under varying illumination direction, phd thesis.

Milliron, T., Jensen, R. J., Barzel, R., and Finkelstein, A. 2002. A framework for geometric warps and deformations. *ACM Trans. Graph. 21,* 1, 20–51.

Moccozet, L. and Thalmann, N. M. 1997. Dirichlet free-form deformation and their application to hand simulation. In *Proceedings of Computer Animation 1997 (CA'97)*. IEEE Computer Society.

Noh, J. Y., Fidaleo, D., and Neumann, U. 2000. Animated deformations with radial basis functions. In *Proceedings of VRST 2000*. Seoul, Korea, 166–174.

Peters, J. 1990. Local smooth surface interpolation: a classification. *Computer Aided Geometric Design 7,* 1-4, 191–195.

Peters, J. 2002. Geometric continuity. In *Handbook of Computer Aided Geometric Design, Chapter 8*. Elsevier.

Pighin, F., Szeliski, R., and Salesin, D. H. 1999. Resynthesizing facial animation through 3d model-based tracking. In *ICCV (1)*. 143–150.

Rushmeier, H., Taubin, G., and Gueziec, A. 1998. Applying shape from lighting variation to bump capture. In *Proceedings of Eurographics Rendering Workshop 1998*. 35–44.

Sánchez, M. A., Edge, J. D., King, S. A., and Maddock, S. 2003. Use and re-use of facial motion capture data. In *Proceedings of Vision, Video and Graphics*. Bath, UK.

Sederberg, T. W. and Parry, S. R. 1986. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. ACM Press, 151–160.

Singh, K. and Fiume, E. 1998. Wires: a geometric deformation technique. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM Press, 405–414.

Singh, K. and Kokkevis, E. 2000. Skinning characters using surface oriented free-form deformations. In *Proceedings of Graphics Interface 2000*. 27–34.

Tao, H. and Huang, T. 1998. Bezier volume deformation model for facial animation and video tracking. In *Proceedings of IFIP Workshop on Modeling and Motion Capture Techniques for Virtual Environments (CAPTECH'98)*.

Viaud, M. L. and Yahia, H. 1992. Facial animation with wrinkles. In *3nd Workshop on Animation, Eurographics '92*. Cambridge, UK.

Williams, L. 1990. Performance-driven facial animation. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*. ACM Press, 235–242.

Woodham, R. J. 1980. Photometric method to determining surface orientation from multiple images. *Optical Engineering 19*, 139–144.

Wu, Y., Kalra, P., Moccozet, L., and Magnenat-Thalmann, N. 1999. Simulating wrinkles and skin aging. *The Visual Computer 15*, 183–198.

Wu, Y., Magnenat-Thalmann, N., and Thalmann, D. 1994. A plastic-visco-elastic model for wrinkles in facial animation and skin aging. In *Proceedings of the second Pacific conference on Fundamentals of computer graphics*. World Scientific Publishing Co., Inc., 201–213.

Yoshizawa, S., Belyaev, A. G., and Seidel, H. P. 2003. Free-form skeleton-driven mesh deformations. In *Proceedings of the eighth ACM symposium on Solid modeling and applications*. ACM Press, 247–253.