From 3D Shape Capture to Animated Models

Adrian Hilton, Jonathan Starck and Gordon Collins

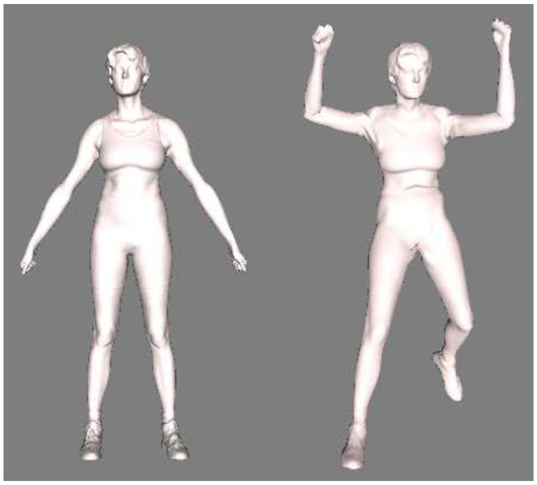
Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford GU27XH, UK
a.hilton@surrey.ac.uk
http://www.ee.surrey.ac.uk/Research/VSSP/3DVision

Abstract

This paper presents a framework for construction of animated models from captured surface shape of real objects. Algorithms are introduced to transform the captured surface shape into a layered model. The layered model comprises an articulation structure, generic control model and a displacement map to represent the high-resolution surface detail. Novel methods are presented for automatic control model generation, shape constrained fitting and displacement mapping of the captured data. Results are demonstrated for surface shape captured using both multiple view images and active surface measurement. The framework enables rapid transformation of captured data into a structured representation suitable for realistic animation.

1 Introduction

Realistic object representation remains a primary goal of computer graphics research [13, 11]. Model construction is a major bottleneck in the wide-spread use of shape capture for computer animation. Currently manual techniques are widely used in film and broadcast production to build models of real objects suitable for realistic animation. Advances in active sensor technology together with research in computer vision and graphics has resulted in systems for capturing surface models of complex 3D objects, people and internal environments [2, 13, 10, 11]. These approches produce accurate and realistic 3D models of complete objects with a level-of-detail not possible with previous manual techniques. However, such techniques result in object models which are represented as unstructured polygonal meshes consisting of millions of polygons. Conversion of such models to a structured form suitable for animation require labour intensive manual remeshing. In this paper we present a general framework for reconstructing structured representation suitable for realistic animation from captured 3D data. Figure 1 illustrates the use of this framework to animate a Cyberware whole-body 3D scan. Captured 3D data can either be from active sensors which project a structured light onto the object surface [2, 4] or passive reconstruction from multiple view images such as shape-from-silhouette or structure-from-motion.



(a) Captured Data (b) Animation

Figure 1. Example animated model from captured 3D surface measurements

Terzopoulos [19] introduced the reconstruction of ‘functional models’ instrumented for facial animation from captured 3D face shape of real people. However, previous work based on this approach was limited to generic models with simple shapes [14] or limited classes of shape [7]. In this paper we present a general framework for fitting generic functional models to captured measurements of 3D objects with complex shape. This framework bridges-the-gap between layered models widely used for character animation [6] and captured 3D surface shape.

2 Reconstruction of Functional Models

In this section the algorithms developed to reconstruct animated models from captured data are presented. A general approach is presented for reconstruction of functional animated models from a set of 3D surface measurements. Captured data of a real object may have a wide variation in shape and pose and does not contain any information on the non-rigid surface deformation required during animation. Therefore, prior knowledge is required either in the form of a generic model for a particular class of objects or manual intervention to explicitly identify the object pose and constraints for non-rigid animation. Section 2.1 first introduces the layered animation framework used for representation of the captured data in a form suitable for animation. The remainder of this section then presents the algorithms developed for reconstruction of layered representations for known objects where a generic animation model exists and for previously unmodelled objects where no generic model is available.

2.1 Overview of Layered Animation Framework

The pipeline for reconstructing animated models from captured 3D surface data is illustrated in Figure 2. The output from the system is a layered representation of the captured 3D data which is structured in a form suitable for realistic and efficient animation. The representation consists of three layers: articulated skeleton, control model and displacement map. The skeleton and control model provide a generic structure for animation with the control model surface non-rigidly deformed by animation of the skeleton. A displacement map is then used to represent the captured surface detail by mapping the captured data onto the control model surface. This enables animation of the high-resolution captured surface detail based on deformation of the underlying control model. The challenge is to establish a mapping between the generic model and captured data with the correct correspondence for non-rigid deformation. Reconstruction of a layered representation from 3D surface data comprises three stages:

1. **Manual Registration:** Initially the generic control model is manually posed for approximate alignment with the capture data.

2. **Shape Fitting:** A shape constrained fitting algorithm is introduced to deform the control model to approximate the data while preserving the control model parameterisation required for animation.

3. **Displacement Mapping:** A continuous mapping is then established between the generic model surface and the unstructured captured data. This mapping is then used to parameterise the high-resolution captured data in the form of a displacement map from the control model surface.

Layered representations are widely used in character animation to achieve realistic surface deformation based on manipulation of an underlying skeleton model [6]. Skeleton structures are widely used as the basis for manipulating animated models either interactively or from motion-capture data. The control model is a low-resolution polygon mesh, M^L, representing a coarse approximation of the character shape and topology. A control model can be derived either by simplification of the captured data model (section 2.5) or from a library of generic objects models which have been structured for efficient animation such as those available from ViewPoint DataLabs [3]. A primary motivation for using existing databases of generic animated models is that the representations have been optimised by experienced animators for realistic and efficient surface deformation. Displacement maps have previously been used to represent high-resolution surface shape [17, 10, 11]. This paper presents a complete framework for reconstruction of animated models using displacement maps. The control model is animated based on the underlying skeleton structure and enables real-time visualisation. As in previous work [18] the vertices of the low-resolution model are mapped to the skeleton structure. Real-time seamless animation is achieved using a geometric transform [18]. More realistic deformation corresponding to a particular surface type could also be implemented using a parametric or physics-based approach to deform the control model.

2.2 Shape Constrained Fitting

Once the generic model is posed to match the 3D data set the next stage is to deform the shape of the generic control model so that it closely conforms to the 3D surface. A requirement for natural animation of the conformed control model is that the mesh topology and vertex parameterisation does not change during conformance. A shape constrained deformable model is used to preserve the prior parameterisation of the control model while fitting to the 3D data set. The novelty of this approach lies in the formulation of a unique parameterisation for arbitrary triangular meshes which is used as the internal energy in mesh deformation.

The deformable surface model $\vec{x}$ minimises the energy function $E(\vec{x})$ incorporating the potential energy from data fitting $P(\vec{x})$, and the internal energy from the shape of the model $S(\vec{x})$.

$$E(\vec{x}) = P(\vec{x}) + S(\vec{x}) \qquad (1)$$

In previous work internal energy terms have been derived based on treating the surface as a membrane or thin-plate

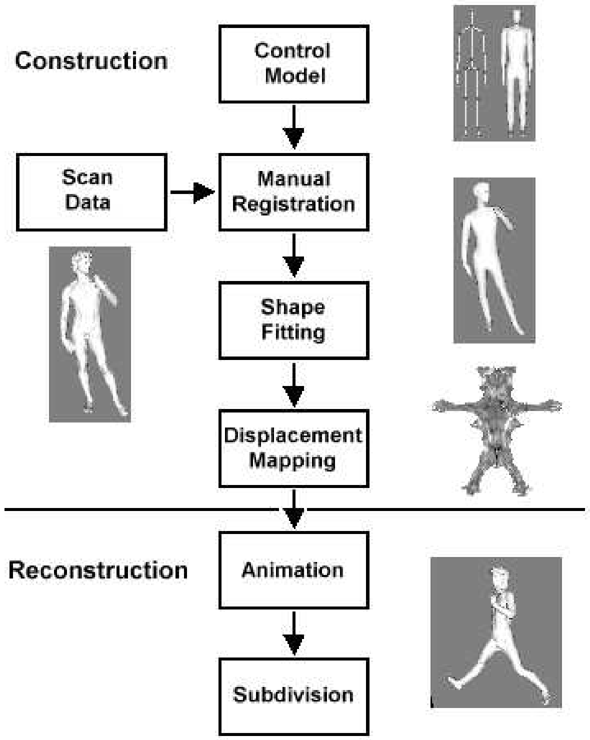
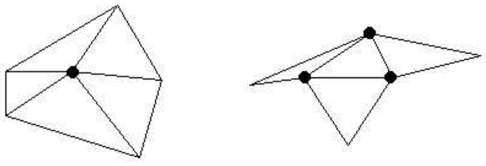


Figure 2. Functional Models Pipeline for Animating Michelangelo's David from Captured Data

material under tension[19]. This yields the surface with minimum area or distortion that fits to the data. However, the thin-plate and membrane energy do not preserve the surface parameterisation which is essential for animation.

To maintain a control model suitable for animation we must constrain both the local shape and parameterisation. Montagnat and Delingette [15] developed an internal energy constraint for simplex meshes with 3 connected vertices. Local frames were used to define the location of each vertex in terms of the 3 neighbour vertices using the barycentric coordinates in the plane and the simplex angle defining the offset from the plane. This gives a local shape parameterisation for simplex meshes which is unchanged under rigid-body transformations and uniform scaling. An internal energy constraint for mesh deformation is defined to preserve the local shape and parameterisation using a spring force between the current vertex position and the reconstructed default position during deformation. In this work we generalise the approach to arbitrary triangular meshes commonly used in computer animation.

For an arbitrary triangular mesh a vertex position is not well defined in relation to the vertex neighbourhood, Figure 3(a). With an irregular number of vertices in the 1-neighbourhood it is not possible to obtain a consistent definition of a local frame to describe the position of the central vertex. We therefore consider a triangle face-based scheme as used by Kobbelt et. al. [9]. The vertex positions of a triangle face can be defined by the barycentric coordinates and height offset in the local frame of the vertices on the faces edge-connected to the central face, the vertices surrounding the central triangle as shown in Figure 3(b). The position of a mesh vertex is therefore constrained by the local position in the triangle face frames in the 1-neighbourhood of the vertex, leading to a 2-neighbourhood support structure for a vertex position.



(a) Triangular mesh (b) Triangle-face frame

Figure 3. Triangle face frame defining local vertex locations

We define the internal energy of a shape constrained model as the integral across the surface of the deviation in the local shape from the generic shape defined in each face based frame. This is given by the summation of the error at the mesh vertices, x_i, preserving the local parameterisation and shape in the vertex positions. Equation 2 defines the internal energy where $(\alpha^0_{if}, \beta^0_{if}, h^0_{if})$ are the default barycentric coordinates (α, β) and height offset h in the f^{th} face based frame for the i^{th} vertex with valence N_i.

$$S(\vec{x}) = \sum_i \sum_f \frac{\|\vec{x}_i - \vec{x}(\alpha^0_{if}, \beta^0_{if}, h^0_{if})\|^2}{N_i} \qquad (2)$$

Figure 4 illustrates the internal energy using the local shape constraint for an arbitrary triangulated mesh to restore the original shape Figure 4(a) of a severly distorted mesh Figure 4(b). Iterations of the mesh deformation to minimise the internal energy, equation 2, are shown in Figures 4(c—e) with the final model converging to the original shape.

For fitting to dense real 3D data sets the external energy of equation 1 $P(\vec{x})$ is derived by summing a data fit error $e(\vec{x})$ across the model surface $\vec{x}(u,v)$. We define the error metric in fitting the data as the least-squared error between the model and the 3D data set. The potential energy function is given by Equation 3 where $\vec{x}_i$ spans the set of I model vertices and $\vec{y}_j$ spans the set of J 3D data points.

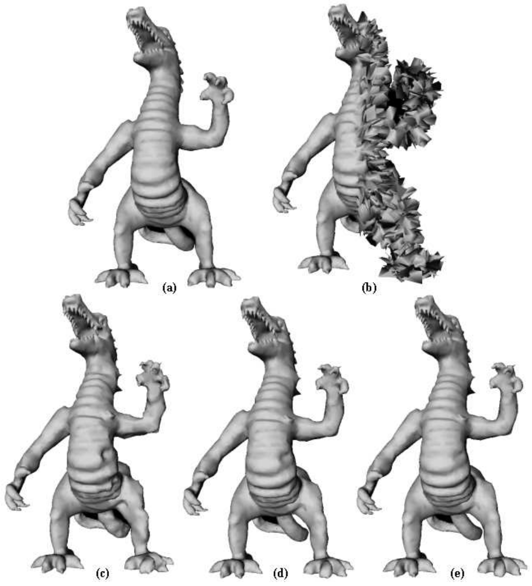


Figure 4. Shape constrained fitting

$$P(\vec{x}) = \sum_i \sum_j m_{ij} ||\vec{y}_j - \vec{x}_i||^2 \tag{3}$$

To achieve robust matching we use the all-neighbours assignment [8] with the constraints: $0 \leq m_{ij} \leq 1$, $\sum_{i=1}^{i=I+1} m_{ij} = 1$ and $\sum_{j=1}^{j=J} m_{ij} = 1$. The approach uses the robust point matching technique introduced by Rangarajan et al. [16] to determine the assignment weights through a coarse-to-fine refinement, further details are omitted due to limitations on space. The advantage of this approach over the standard nearest point matching is that it allows for greater initial misalignment between the control model and the data. Fitting is performed by an iterative gradient descent solution to minimise the energy function equation 1:

$$\frac{d\vec{x}_i}{dt} = \sum_j m_{ij} \left(\vec{y}_j - \vec{x}_i\right) - \sum_f \frac{\left(\vec{x}_i - \vec{x}(\alpha^0_{if}, \beta^0_{if}, h^0_{if})\right)}{N_i} \tag{4}$$

We fit the deformable model to the 3D data-set by alternately updating the assignment to the 3D points m_{ij}, then updating the vertex positions of the model $\vec{x}_i$. Figure 12(c) illustrates the control model obtained after shape constrained fitting to the Stanford Michelangelo data set [13].

2.3 Mapping Captured Data

In previous work [18] we introduced the normal-volume mapping to automatically parameterise an arbitrary high-resolution mesh model, M^H, with respect to a low-resolution control model, M^L. It was shown that this parameterisation could be used to seamlessly animate the high-resolution model based on deformation of the underlying control model. In this section we summarise the key details of the normal-volume mapping which are used for displacement map generation.

For each triangle t_r in an arbitrary mesh M we define a *normal-volume*, $V^N(t_r)$, by displacing the triangle vertices, $\vec{v}_j$, along the vertex normals, $\vec{n}_j$, as illustrated in Figure 5. The union of the normal-volumes for all triangles in mesh M encloses a continuous volumetric envelope which can be used to define a continuous mapping between points in 3-space and the mesh surface.

To obtain a continuous mapping we project points in 3-space along the corresponding interpolated triangle normal. A point $\vec{p}_j$ on the surface of triangle $t_j = (\vec{v}_r, \vec{v}_s, \vec{v}_t)$ and its unit normal $\vec{n}_j$ can be defined by bilinear interpolation using barycentric coordinates as:

$$\begin{aligned} \vec{p}_j &= \alpha\vec{v}_r + \beta\vec{v}_s + (1-\alpha-\beta)\vec{v}_t \\ \vec{n}_j &= \alpha\vec{n}_r + \beta\vec{n}_s + (1-\alpha-\beta)\vec{n}_t \end{aligned} \tag{5}$$

where for a point inside the triangle the barycentric coordinates α, β and $(1-\alpha-\beta)$ are all positive scalar variables. Bilinear interpolation of the normal gives a continuous variation in the triangle normal, $\vec{n}_j$, across the planar surface and between adjacent triangles. The resulting normal field is continuous such that for every point $\vec{x}$ inside the normal-volume there is a corresponding normal $\vec{n}_j$ which passes through that points. Figure 5 illustrates the normal passing through a point $\vec{x}$. Thus any point $\vec{x}$ inside the normal volume $V^N(t_r)$ can be expressed as:

$$\vec{x} = \alpha(\vec{v}_r + d\vec{n}_r) + \beta(\vec{v}_s + d\vec{n}_s) + (1-\alpha-\beta)(\vec{v}_t + d\vec{n}_t) \tag{6}$$

where d is the Euclidean distance of the points $\vec{x}$ along the normal $\vec{n}_j$ from $\vec{p}$. The above equation can be viewed geometrically as an offset surface at a distance d which defines a plane passing through point $\vec{x}$ as illustrated in Figure 5. As shown in previous work [18] for an arbitrary point $\vec{x}$ we can obtain the parameters (α, β, d) for triangle t_r by solving for the plane in the normal-volume which passes through the point. If α, β and $(1-\alpha-\beta)$, are in the range $[0, 1]$ then the point, x, maps to a point on the triangle surface.

Parameterisation of a high-resolution model, M^H, with respect to a low-resolution model, M^L, is achieved by evaluation of the normal-volume mapping to the high-resolution model vertices. Each model vertex $\vec{v}_i^H$ has an associated set of four mapping parameters $(r_i, \alpha_i^H, \beta_i^H, d_i^H)$ where r_i is the low-resolution triangle index, (α_i^H, β_i^H) are the barycentric coordinates of the corresponding point on the low-resolution model triangle and d_i^H is the distance along the interpolated triangle normal. Having computed this

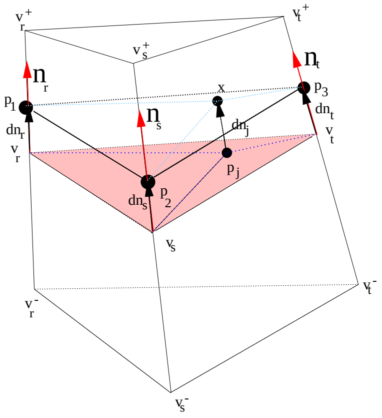


Figure 5. Normal volume mapping.

mapping the model can be exactly reproduced and animated from the low-resolution model, M^L, together with the high-resolution vertex parameters and mesh topology.

2.4 Displacement Mapping of Surface Detail

In this section we present a process for computing a displacement map representation of the high-resolution surface model. The displacement map is an efficient image based representation of the high-resolution surface detail[17]. An approximation of the high-resolution model can be rapidly reconstructed from the displacement map image together with the low-resolution control model. Animation of the low-resolution control model can be used to achieve efficient animation of the high-resolution surface detail.

Figure 6 illustrates the process of displacement map generation for a single triangle in the low-resolution control model, M^L. Initially a mapping is established between the high-resolution mesh, M^H, and the control model, M^L, in 3D space using the triangle normal-volume as shown in Figure 6(a). The high-resolution model is then mapped to a 2D image plane using a set of texture coordinates for each triangle in the low-resolution model, Figure 6(b). The distance between the high and low resolution model surfaces along the interpolated surface normal is then sampled on a regular grid to obtain the displacement map image, Figure 6(c).

The normal-volume mapping defines a parameterisation for all triangle vertices $\vec{v}_i^H$ in a high-resolution model, M^H, in terms of the nearest triangle t_r^L on a low-resolution control model M^L. The mapping of a vertex, $\vec{v}_i^H$, to a point on triangle t_r^L is defined by equation 6 as a point on the triangle surface in barycentric coordinates $p_i^H = (\alpha_i^H, \beta_i^H)$ and the distance d_i^H along the interpolated triangle normal. Thus, for every vertex $\vec{v}_i^H$ in the high-resolution model M^H we obtain a mapping specified by four parameters $(r_i, \alpha_i^H, \beta_i^H, d_i^H)$ where r is the low-resolution triangle index.

Let us define a set of texture coordinates, $\{\vec{u}_i^L = (u_i^L, v_i^L)\}_{i=0}^{N_{tc}}$, for the low-resolution model M^L which uniquely map each triangle to a two-dimensional texture map plane, T. For each triangle, $t_r^L = (\vec{v}_o^L, \vec{v}_p^L, \vec{v}_q^L)$, there is a continuous mapping to a corresponding triangle in the texture image domain $t_r^{LT} = (\vec{u}_o^L, \vec{u}_p^L, \vec{u}_q^L)$. We can obtain a mapping of the high-resolution model to the texture map plane, T, by combining the low-resolution model texture mapping with the normal-volume mapping of the high-resolution model M^H onto the low-resolution model M^L. Given the normal-volume mapping $(r_i, \alpha_i^H, \beta_i^H, d_i^H)$ for vertex $\vec{v}_i^H$ in the high-resolution model onto triangle t_r^L in the low-resolution model the mapping to texture coordinates is:

$$\vec{u}_i = \alpha_i^H \vec{u}_o^L + \beta_i^H \vec{u}_p^L + (1 - \alpha_i^H - \beta_i^H)\vec{u}_q^L \tag{7}$$

This enables us to map any triangle $t_n^H = (\vec{v}_i^H, \vec{v}_j^H, \vec{v}_k^H)$ in the high-resolution model into a triangle the $t_n^{HT} = (\vec{u}_i^H, \vec{u}_j^H, \vec{u}_k^H)$ in the texture image plane, T. Thus any point on the high-resolution model surface $\vec{x}$ can be mapped to a point in the texture image plane $\vec{u}(\vec{x})$. It should be noted that this mapping is not injective, multiple points on the high-resolution model may map to the same point on the low-resolution model due to over-folding of the surface. This many-to-one projection is a known limitation of displacement maps. In practice this problem can be avoided by either adding additional triangles to the low-resolution model to ensure a one-to-one mapping for the entire surface or by approximating the surface geometry as discussed in section 2.5.

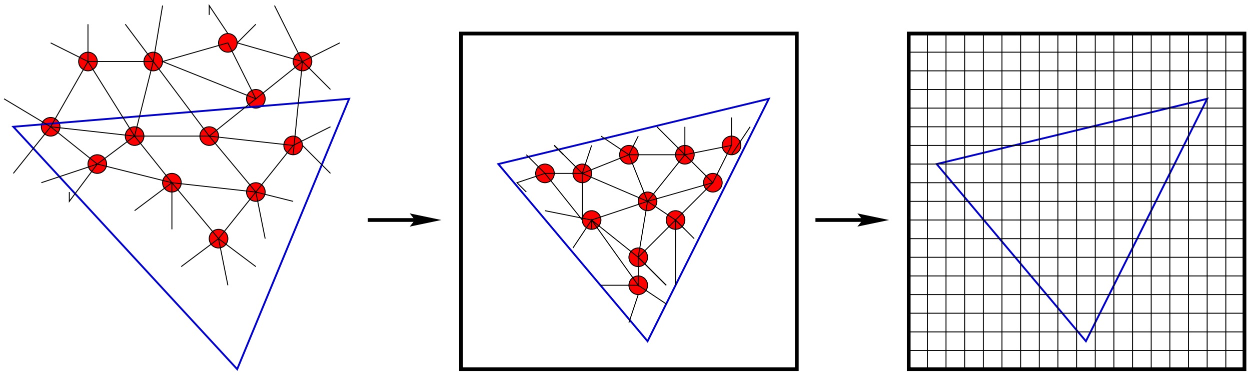
Given the mapping of the high resolution model, M^H, to the texture plane, T, we can obtain a sample of the high-resolution model surface, $\vec{x}(\vec{u})$, for any point $\vec{u}$ in the texture plane which is inside the region to which the high-resolution model maps. For a point $\vec{u}$ we can find the triangle $t_n^{HT} = (\vec{u}_i^H, \vec{u}_j^H, \vec{u}_k^H)$ from the high-resolution model which it is inside such that:

$$\vec{u} = \gamma \vec{u}_i^{HT} + \nu \vec{u}_j^{HT} + (1 - \gamma - \nu)\vec{u}_k^{HT} \tag{8}$$

where $\gamma, \nu, (1 - \gamma - \nu) \in [0, 1]$ are barycentric coordinates. From the normal-volume mapping we know the distance d_i^H of each vertex $\vec{v}_i^H$ along the interpolated low-resolution triangle normal. The distance for any point inside the triangle is:

$$d^H(\vec{u}) = \gamma d_i^H + \nu d_j^H + (1 - \gamma - \nu) d_k^H \tag{9}$$

The mapping defined above enables the distance between the low and high resolution models to be sampled at any point in the image plane. This mapping is used to generate a displacement map image D^H where the distance



(a) Mapping of M^H to M^L (b) Mapping $M^H$to 2D plane (c) Sampling distance M^L to M^H

Figure 6. Displacement map generation for high-resolution model, M^H.

is sampled for a set of discrete points $\vec{u}(n,m)$ in the image plane. Further details of the implementation of the displacement map generation are presented in [17].

2.5 Control Model Generation

The layered model reconstruction from 3D scan data presented thus far assumes that a generic shape model is available a priori. Instrumented generic models are available from commercial databases [3] for a wide variety of objects. In this section we consider the generation of a suitable control model for the case where no generic model is available a priori. A requirement of the layered representation using displacement maps is that there is a one-to-one injective mapping between the captured data and control model surface. This requirement must be satisfied to ensure that the captured surface is represented without loss of detail. In this section we introduce a constrained mesh simplification algorithm that guarantees that the resulting simplified mesh is injective with the control model.

Mesh simplification algorithms for efficient representation have been widely developed in computer graphics using both geometric and appearances based error metrics [5]. Techniques for ensuring an injective mapping between the simplified mesh and the original surface have been developed [12] which use heuristic constraints on the normals. This method does not guarantee that the resulting model is injective. In this work we have developed a mesh simplification algorithm that ensures that the mapping is injective using the normal volume mapping.

Simplification is performed by edge-collapse with edges prioritised by a measure of the resulting geometric error [12]. Injectivity between the high-resolution captured data model and control model is preserved by testing the normal-volume mapping for each vertex. Given a high-resolution triangle $t_H(\vec{x}_i, \vec{x}_j, \vec{x}_k)$ and a low resolution triangle on the control model $t_L(\vec{x}_m, \vec{x}_n, \vec{x}_o)$ the normal-volume defines a mapping such that $x_i = (\alpha_i, \beta_i, d_i)$, $x_j = (\alpha_j, \beta_j, d_j)$ and $x_k = (\alpha_k, \beta_k, d_k)$. Now we can express the condition which must be satisfied for injective mapping as:

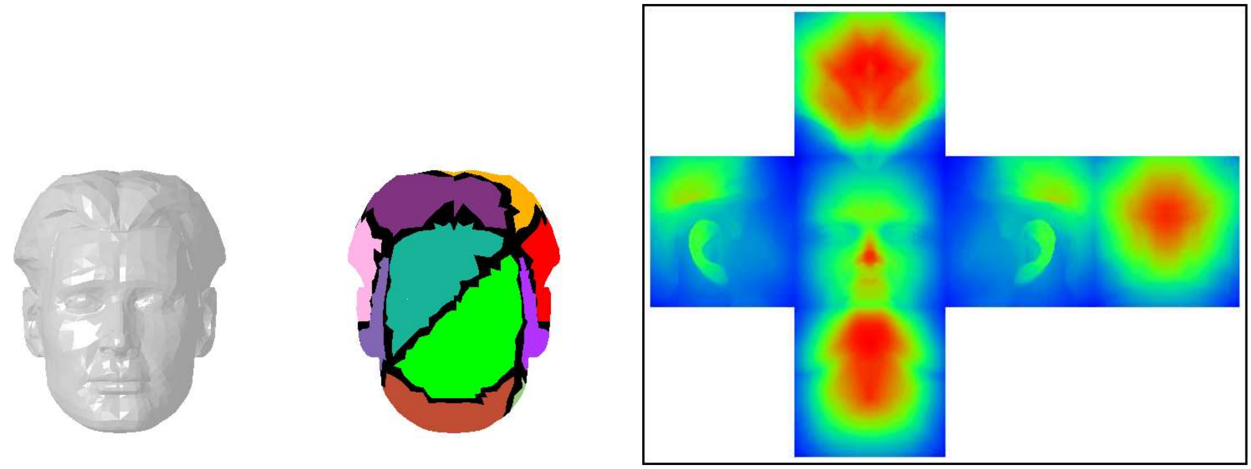
$$[(\vec{p}_j(\alpha_j,\beta_j) - \vec{p}_i(\alpha_i,\beta_i)) \times (\vec{p}_k(\alpha_j,\beta_j) - \vec{p}_i(\alpha_i,\beta_i))] \bullet [(\vec{x}_j - \vec{x}_i) \times (\vec{x}_k - \vec{x}_i)] > 0 \qquad (10)$$

where the mapped triangle points $(\vec{p}_i, \vec{p}_j, \vec{p}_k)$ are defined according to equation 5. This condition states that the normal of the captured triangle t_H and the triangle mapped on to the low-resolution triangle t_L must lie in the same half-space. Enforcing this condition during simplification ensures that there is a one-to-one injective mapping between the control model and captured data. The simplification proceeds by testing the mapping according to equation 10 for each triangle in the 2-neighbourhood of an edge prior to collapse. During simplification additional constraints can be placed on the mesh to obtain a suitable structure for animation by interactively labelling edges to not be deleted.

Figures 8 illustrates the constrained mesh simplification applied without any fixed edges according to captured models of a horse and Venus statue. The resulting simplified for the horse from 96K to 504 triangles and for the Venus statue from 100K to 84 triangles. The resulting simplified meshes are injective with the original mesh and enable parameterisation of the captured data using the normal-volume mapping without loss of accuracy. The constrained mesh simplification enables automatic generation of control models of previously unmodelled objects suitable for a layered representation.

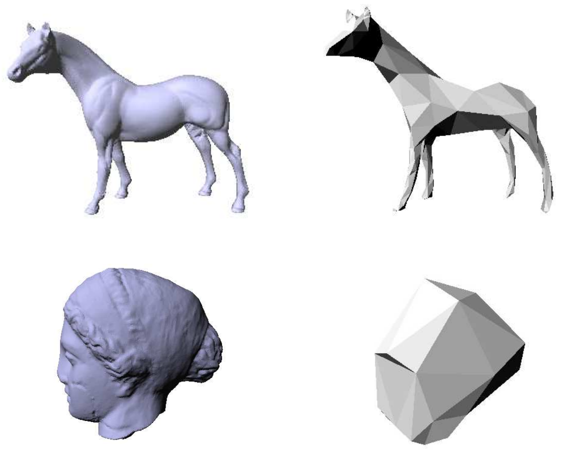
3 Results

In this section we present results for application of the framework to 3D surface measurements from



(a) Captured Model (b) Mapping to Cube (c) Displacement Map

Figure 7. Displacement Map of Head-to-Cube using the Normal Volume Mapping



(a) Captured Data (b) Simplification

Figure 8. Injective Control Model Generation

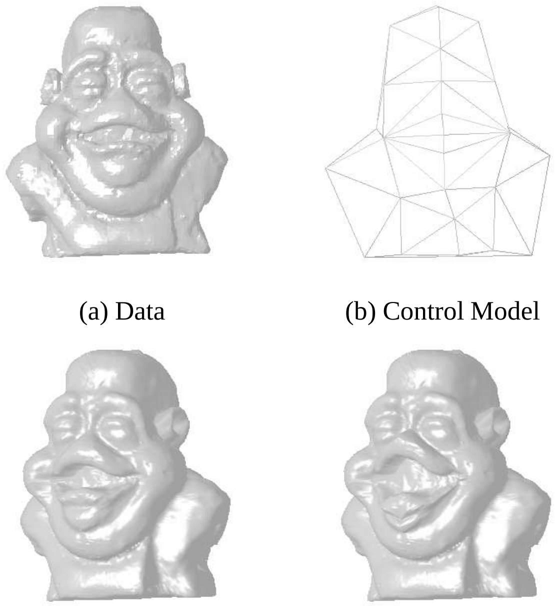
both active and passive sensors. Further examples of the animations produced can be viewed at www.ee.surrey.ac.uk/CVSSP/3DVision /AnimatedStatues and /PrometheusResults.

3.1 Active 3D Surface Measurement Data

Figure 9 presents the results for animation from a laser scan of a monster character head produced by an animation studio[4]. The control model generated by simplification of the captured data is shown in Figure 9(b) and resulting mouth animation produced by manipulating the control model in Figure 9(c). The original captured data size is 935Kb and the layered model size including the control model and skeleton is 34Kb.

Figure 10 shows the reconstructed model for a Cyberware (www.cyberware.com) 3D scan of a person. The model was reconstructed using a generic humanoid model of approximately 2500 polygons. Initially the model was manually aligned with the data using the articulated skeleton. Shape constrained fitting was then performed followed by normal-volume mapping to generated the displacement map. Figure 11 shows an analysis of the errors between the captured data and the reconstructed model with uniform quaternary subdivision of the control model at levels of 0,1,2 and 3. Results indicate a mean error of 0.5mm with an rms error of less-than 2mm for level 3 subdivision. A quantisation error of 0.48mm was used for the displacement map. Mean and rms errors were computed using the Metro tool [1].

Figure 12 presents the reconstruction of a layered animation model for the 3D Scan of Michelangelo's David (data courtesy of Stanford [13]). The original data and layered representation are presented for comparison in Figure 12(a) and (b). Accurate reconstruction is achieved even though the captured statue is in a complex pose with the right hand connected to the hip and left hand connected to the shoulder. The shape constrained fit of the 2500 triangle generic model to the data together with the resulting displacement map image are shown in Figure 12(c). Finally sample poses from the animation of the statue are shown in Figure 12(d). This example is presented to illustrate the capacity of the framework to reconstruct accurate layered representation for highly detailed objects.



(a) Data

(b) Control Model

(c) Animation

Figure 9. Layered animation for monster head

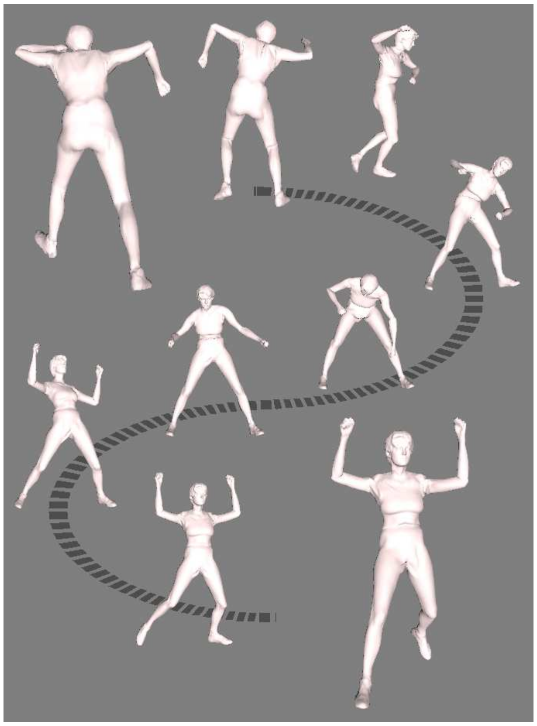


Figure 10. Animation of a Cyberware Whole-body Scan

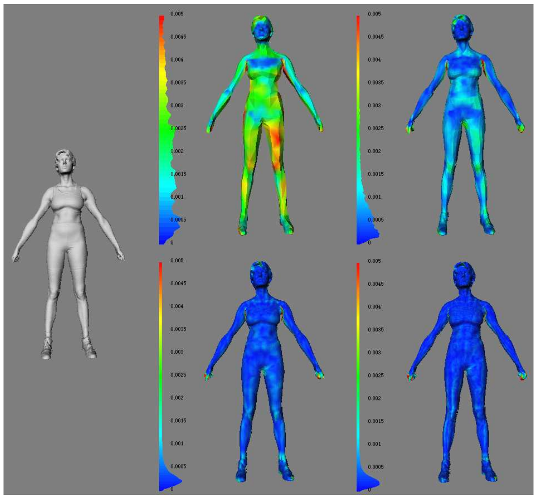


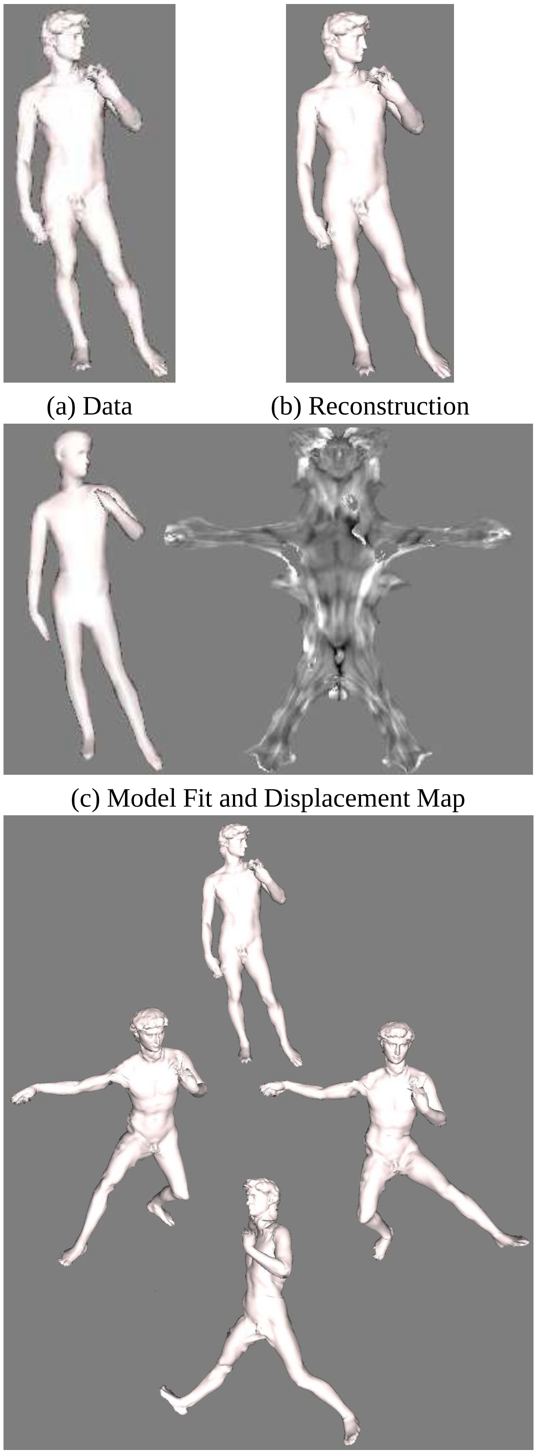
Figure 11. Errors in reconstruction from Cyberware model with uniform subdivision at 0,1,2 and 3 levels

3.2 Multiple View Images

The framework developed can also be applied for the reconstruction of animated models from shape captured from multiple view images or image sequences. Application of the framework is the same as for reconstruction from 3D scan data. Figure 13 shows the reconstruction of a ballet dancer in a six camera blue-screen studio. Five of the images are captured from the front and one image from the back looking down, three images are shown. The visual hull of the dancer reconstructed from the image silhouettes is shown in Figure 13(b) together with the initial skeleton pose. Figure 13(c) is the shape constrained model reconstruction using the 2500 polygon generic model. It should be noted that in the raw voxel data there is a large ambiguity at the front and back of the chest due to occlusion. The shape constraints ensure that a reasonable model is reconstructed in ambiguous regions. In the case of image silhouettes the reconstructed surface is too noisy to warrant displacement mapping of the surface detail the control model surface is therefore used for animation. An example animation of the texture mapped dancer model standing in a Venice street scene is shown in Figure 13(d). The full animation is available at:
www.ee.surrey.ac.uk/CVSSP/3DVision/PrometheusResults/.

4 Conclusions

In this paper we have presented a general framework for animation of 3D surface date captured from either active sensors or multiple view images. A layered representa-

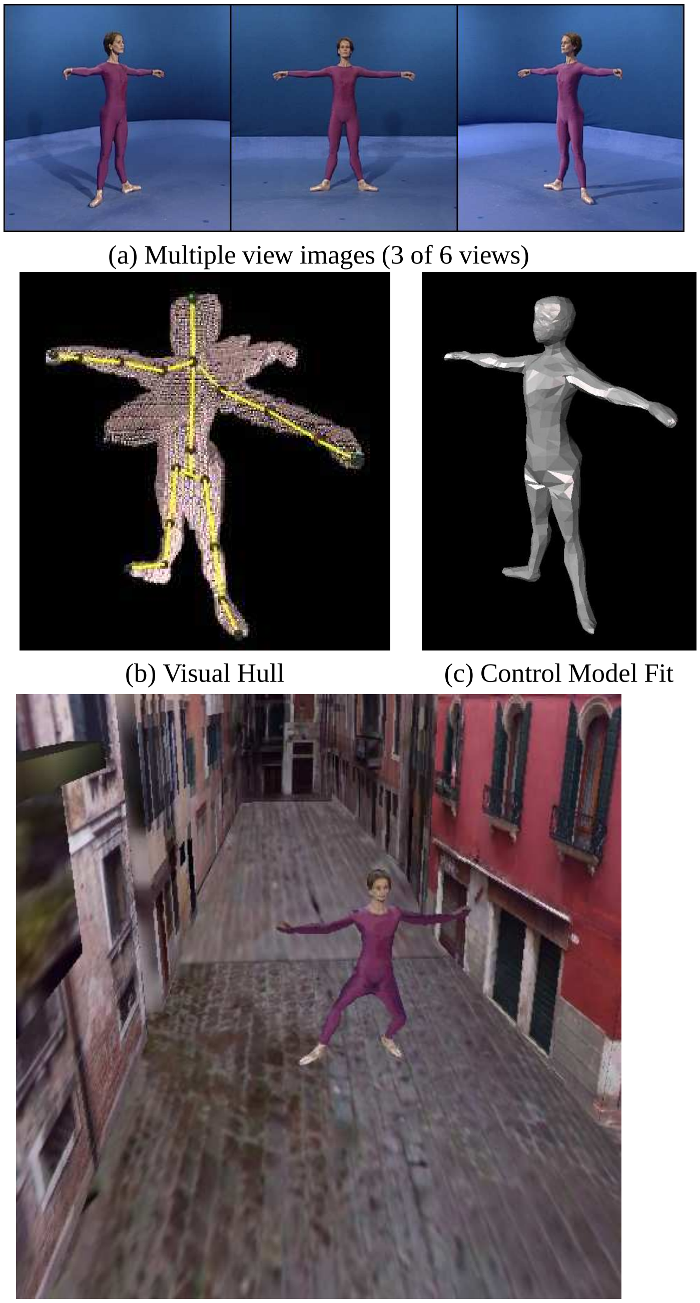


(a) Data (b) Reconstruction

(c) Model Fit and Displacement Map

(d) Animation

Figure 12. Layered Model Reconstruction for Michelangelo's David (courtesy Stanford [13])



(a) Multiple view images (3 of 6 views)

(b) Visual Hull (c) Control Model Fit

(d) Animation in Street Scene

Figure 13. Dancer Model Reconstruction from Multiple View Image Silhouettes

tion is reconstructed composed of a skeleton, control model and displacement map. The control model is manipulated via the skeleton to produce non-rigid mesh deformation using techniques widely used in animation. High-resolution captured surface detail is represented using a displacement map from the control model surface. This structure enables seamless and efficient animation of highly detailed captured object surfaces. Novel algorithms have been introduced to perform the following tasks:

- Shape constrained fitting of a generic control model to approximate the captured data. The shape constrained fitting ensures that for an arbitrary triangulated control model the original parameterisation is preserved which is essential for realistic animation.
- Automatic mapping of the high-resolution data to the control model surface based on the normal-volume is used to parameterise the captured data. This parameterisation is then used to generate a displacement map representation. The displacement map provides an efficient representation of the captured surface detail which can be adaptively resampled to generate animated models at multiple levels-of-detail.
- Automatic control model generation for previously unmodelled objects. A mesh simplification algorithm has been developed to produce control models from the captured 3D surface. The control models produced are guaranteed to be injective with the captured data enabling displacement mapping without loss of accuracy using the normal-volume.

This framework enables rapid transformation of 3D surface measurement data of real objects into a structured representation for realistic animation. Limited manual interaction is required to initially align the generic control model or define constraints for remeshing of previously unmodelled objects. The algorithms developed then enable automatic construction of a layered shape representation. Results have been presented for reconstruction of a layered representation from 3D scan data of computer graphics character models and people. The shape constrained fitting of the generic control model has been used for reconstruction of animated models from the visual hull produced by multiple view image silhouettes. Shape constrained fitting to relatively inaccurate and ambiguous data produces a reasonable shape approximation. In both cases the approach produces models suitable for realistic animation.

Acknowledgements

This work was supported by UK EPSRC funding agency on Broadcast LINK project PROMETHEUS GR/M88075 and EPSRC Advanced Fellowship AF/95/2531 in collaboration with the BBC and BT.

References

[1] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167—174, 1998.

[2] B. Curless and M. Levoy. A Volumetric Method for Building Complex Models from Range Images. In *ACM Computer Graphics Proceedings, SIGGRAPH, NewOrleans,USA*, pages 303–312, 1996.

[3] Viewpoint Datalabs. *Viewpoint Catalog*. (http://www.viewpoint.com), 1996.

[4] A. Hilton and J. Illingworth. Geometric fusion for a hand-held 3d sensor. *Machine Vision Applications*, 12(1):44–51, 2000.

[5] M. Isenburg and J. Snoeyink. Face fixer: Compressing polygonal meshes with properties. In *Proc. ACM SIGGRAPH*, pages 263—270, 2000.

[6] John E.Chadwick, David R.Haumann and Richard E.Parent. Layered Construction for Deformable Animated Characters. In *Proceedings of SIGGRAPH '89, Boston, 31 July-4 August, 1989*, pages 243–252, 1989.

[7] X. Ju and J.P. Siebert. Conformation from generic animatable models to 3d scanned data. In *International conference of 3D Scanning, Paris*, page 1.

[8] I.A. Kakadiaris and D. Metaxas. Three-dimensional human body model acquisition from multiple views. *International Journal of Computer Vision*, 30(3):191—218, 1998.

[9] L. Kobbelt, S. Campagna, J. Vorsatz, and H.P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Proc. ACM SIGGRAPH*, pages 105—114, 1998.

[10] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. In *In ACM Computer Graphics Proc. SIGGRAPH, NewOrleans, USA*, 1996.

[11] A. Lee, H. Moreton, and H. Hoppe. Displaced subdivision surfaces. In *Proc. ACM SIGGRAPH*, pages 85—94, 2000.

[12] A. Lee, W. Sweldens, P. Schroder, L. Cowsar, and D. Dobkin. Maps: Multiresolution adaptive parameterisation of surfaces. In *Proc. ACM SIGGRAPH*, pages 95—104, 1998.

[13] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Periera, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project. In *ACM Computer Graphics Proceedings, SIGGRAPH*, pages 131—144, 2000.

[14] D. Metaxas and I.A. Kakadiaris. Elastically adaptive deformable models. In *European Conference on Computer Vision*, pages 550—559, 1996.

[15] J. Nibtaganat and H. Delingette. A hybrid framework for surface registration and deformable models. In *Conference on Computer Vision and Pattern Recognition*, page 1041, 1997.

[16] A. Rangarajan, H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P. Goldman-Rakik, and J. Duncan. A robust point matching algorithm for autoradiograph allignment. *Medical Image Analysis*, 4(1):371—398, 1997.

[17] R. Smith, W. Sun, A. Hilton, and J. Illingworth. Layered animation using displacement maps. In *IEEE International Conference on Computer Animation*, pages 146—154, May 2000.

[18] W. Sun, A. Hilton, R. Smith, and J.Illingworth. Layered animation of captured data. *Accepted for publication in Visual Computer: International Journal of Computer Graphics*, 17(4):457—474, 2001.

[19] D. Terzopoulos. From physics-based representation to functional modeling of highly complex objects. In *NSF-ARPA Workshop on Object Representation in Computer Vision*, pages 347—359. Springer Verlag, 1994.