# Spatio-Temporal Fusion of Multiple View Video Rate 3D Surfaces

Gordon Collins and Adrian Hilton
*Centre for Speech Vision and Signal Processing*
*University of Surrey*
A.Hilton@surrey.ac.uk

## Abstract

*We consider the problem of geometric integration and representation of multiple views of non-rigidly deforming 3D surface geometry captured at video rate. Instead of treating each frame as a separate mesh we present a representation which takes into consideration temporal and spatial coherence in the data where possible. We first segment gross base transformations using correspondence based on a closest point metric and represent these motions as piecewise rigid transformations. The remaining residual is encoded as displacement maps at each frame giving a displacement video. At both these stages occlusions and missing data are interpolated to give a representation which is continuous in space and time.*

*We demonstrate the integration of multiple views for four different non-rigidly deforming scenes: hand, face, cloth and a composite scene. The approach achieves the integration of multiple-view data at different times into one representation which can processed and edited.*

## 1. Introduction

Recent advances in sensor technology have led to a number of systems for video-rate 3D shape capture. This paper addresses the problem of integration and representation of captured sequences of 3D surfaces measurements of unknown non-rigid objects from multiple overlapping views. Such a stream of 3D surface measurements is inherently unstructured, uncompressed and incomplete since for each spatial view at each time frame the captured range image will give a surface mesh with different topology as well as geometry. The objective is to integrate such multiple view mesh sequences into a single deforming mesh to give an efficient, integrated representation. The integrated representation should support further processing such as dynamic object modelling, analysis of surface deformation, hole filling, noise reduction or editing of surface geometry.

In general, a vertex of a mesh at a certain view and a certain frame will not have corresponding vertices on other views or frames. If we can establish correspondence between vertices on meshes then our representation will be more compact (since only one connectivity will be needed). Furthermore, missing data (from occlusions or measurement errors) may be reconstructed by interpolating between known correspondences in both space and time rather than just in space.

In this paper we seek to register surface measurements using a variation of iterative closest point (ICP) on segments of the data. However where the correspondence measure fails (on smooth regions or missing data for example) many vertices will not have a corresponding point. For this reason we adopt a paradigm from computer graphics whereby the representation is split into a low resolution deforming base mesh and a high resolution residual component - a displacement map. The low resolution deforming base mesh consists of those parts of the surface which can be matched and the remainder is encoded as a residual.

We encode the low resolution deformations as piecewise rigid transformations of a coarse base mesh. We can represent non-rigid deformations arbitrarily accurately by choosing small pieces of the base mesh to transform rigidly (in the limit a single vertex can be translated). The base mesh is segmented by performing an ICP of piecewise rigid segments at each frame and clustering those vertices which lie within a given tolerance. Temporal coherence is exploited to deal with missing data so that if a surface region is occluded for a short period it is not discarded from the registration. At the end of the segmentation, the positions of unregistered base vertices are interpolated in space and time.

Residuals are then encoded using displacement mapping and stored as a displacement video. Again missing data is easily interpolated in space and time. The representation has the same connectivity at each frame since we ensure that the same pixels are present in all frames of the video.Furthermore we show how the displacement video can be edited to give novel animations of the data's detail.

We apply the method to four examples with very different deformations to demonstrate the generality of the representation. First a composite rigid;y transforming scene of an arm and a moving block shows that the method suc-

cessfully tracks and segments rigid movements. The cloth example is highly non-rigid. The hand example, although rigid, has multiple occlusions and finally the face example is a mixture of rigid and non-rigid deformations.

## 2. Previous Work

### 2.1. Video-Rate Shape Capture

Recent research has resulted in systems for video-rate dense 3D shape acquisition using both active [11, 13, 14, 9, 18] and passive [5, 12, 17] techniques. Active sensors produce relatively accurate high-resolution surfaces measurements but require the projection of a visible pattern onto the object surface prohibiting simultaneous capture of surface colour appearance. A number of active sensors for dynamic 3D shape capture are now commercially available for capture of the human face and body (3DMD, C3D, EyeMatic, Z-cam). Infra-red (IR) structured light projection has been used to allow simultaneous acquisition of shape and colour at video frame rates (25Hz) [18] which we have found to be sufficient to capture the complicated surface dynamics of a variety of scenes. Simultaneous acquisition of shape and colour ensures accurate registration without spatial or temporal misalignment. Multiple view data from a video-rate IR 3D shape and colour capture system is used throughout this paper.

### 2.2. Shape Integration and Fusion

Previous research in object and environment modelling from 3D surface measurements has focused on the reconstruction of static scenes [8]. Methods have been introduced for the registration and integration of multiple view range images which assume that the surface is rigid. Rusinkiewicz et al. [14] extended these techniques to efficient reconstruction of rigid models from video-rate sequences of range images for real-time object modelling. Recent research on video-rate sequences of 3D face shape introduced methods for parametric model fitting [10, 16]. Li et al. [10] use an extension of optic flow [1] to fit parametric models to 3D shape captured from multiple views. Wang et al. [16] used a deformable surface framework to fit a high resolution (8000 vertex) parametric face model to sequences of face shape from a single view. Both approaches achieve reconstruction of detailed parametric models of non-rigid face shape deformation which are used for animation of facial expressions. An alternative 3D video face representation using ellipsoidal displacement maps together with non-rigid alignment of face shape was introduced in [19]. The resulting representation was used for concatenative synthesis of facial dynamics during speech with both shape and colour. These approaches for analysis of video-rate 3D surface measurements for non-rigid shape has been limited to faces requiring either a prior model [10, 16] or injective mapping to a simple shape [19]. Representation of complex non-rigid structures from captured measurements such as cloth or hair requires more general techniques for processing 3D surface measurements. In this paper we introduce a general method for spatio-temporal integration of video-rate 3D surface measurements from multiple views into a single surface model and representation of non-rigid surface dynamics.

## 3. Representation Overview and Notation

We consider a video stream of meshes from $K$ multiple views such that at frame $i$ and view $k$ the coordinates of a mesh with $N_k^i$ vertices are given by the homogeneous coordinates:

$$H_k^i = \{\mathbf{h}_j\}$$
$$\text{for } j = 1 \ldots N_k^i \text{ , } i = 1 \ldots M. \text{ and for } k = 1 \ldots K.$$

where $M$ is the number of frames in the sequence. Spatial dependence is given by the subscript $j$ and time dependence is denoted by the superscript $i$.

Our problem is to represent the sequences by a single base mesh, a series of rigid transformations and a displacement video.

We define the texture mapped base mesh $B$ with vertices $\{\mathbf{b}_j\}$ for $j = 1 \ldots N$. The base mesh can either be obtained automatically from the first frame of the sequence or from a user defined model. If the first frames ($H_k^0$ for k = 1 ... K) is used we fuse the multiple views using standard fusion techniques and then decimate this to simplify the model. In this case the base mesh is already closely fitted to the first frame. A user defined control model may be preferable for an animator who may choose one with vertex positions which approximate the degrees-of-freedom of the captured data. In both cases texture mapping is determined for the base mesh either manually or using standard techniques.

Given an initial base mesh $B$ we introduce a method to automatically segment the base mesh and learn a set of transformations for each segment which approximate the non-rigid surface dynamics for the captured multiple view 3D sequences.

The $M \times N$ rigid transformations (RTs) are given by $4 \times 4$ matrices $T_j^i$ so that the coarse deformations are given by the piecewise rigidly transforming base mesh consisting of vertices given by

$$T_j^i \mathbf{b}_j \tag{1}$$

The segmentation process means that many of the transformations $T_j^i$ will be the same for different vertices $j$.

Although any number of deformation bases are possible, we have chosen to represent the data as sequences of piecewise rigid transformations. Such a basis spans the set of all

possible transformations and provides a suitable approximation of the captured non-rigid deformations. Rigid transformation bases are also easily incorporated into the conventional computer graphics pipeline where they are widely used for character animation.

## 4. Displacement Mapping

Displacement mapping is used both in the closest point search for registering the coarse piecewise RTs and in encoding the detailed deformations. We define a displacement mapping from mesh $L$ to mesh $H$ as,

$$\mathbf{x} = D(\mathbf{X}) = \mathbf{X} + d\mathbf{N}(\mathbf{X}) \tag{2}$$

where $\mathbf{X} \in L$ and $\mathbf{x} \in H$ are points in 3D and $\mathbf{N}(\mathbf{X})$ is a unit normal to the surface $L$ at $\mathbf{X}$. The displacement mapping equation (2) can be solved for arbitrary points $\mathbf{X} \in L$ to give a distance $d$ along the normal $\mathbf{N}$. This value is then encoded via texture coordinates as a grey scale value $I^i(l, m)$ at pixel $(l, m)$ at frame $i$ in the displacement video. If this is done for all frames the resulting displacement video encodes the detailed deformations.

Displacement mapping for meshes has previously been used to represent surface detail for static objects represented for B-spline[6], subdivision [7] and mesh [15] surfaces. Here we give an overview of the scheme used in [15] for displacement mapping from a low resolution mesh $L$ to a high resolution mesh $H$. We define a continuous normal on the low resolution mesh by a bilinear interpolation of the normals of a triangle $T_j$ so that we compute a local mapping $D_j : T_j \in L \rightarrow H$ of a point $\mathbf{X} \in L$ as:

$$\begin{aligned} \mathbf{x} = D_j(\mathbf{X}) &= \mathbf{x}(\alpha, \beta, d, j) \\ &= \alpha\mathbf{V}_1 + \beta\mathbf{V}_2 + (1 - \alpha - \beta)\mathbf{V}_3 \\ &+ d(\alpha\mathbf{N}_1 + \beta\mathbf{N}_2 + (1 - \alpha - \beta)\mathbf{N}_3) \end{aligned} \tag{3}$$

where $\alpha$ and $\beta$ are the barycentric coordinates of a point on a triangle $j$ of $L$ with vertices $\mathbf{V}_i$ and normals $\mathbf{N}_i$.

The mapping can be inverted so that given a point $\mathbf{x}$ in $H$ we can solve equation (3) for $\alpha$, $\beta$ and $d$ and hence compute $D_j^{-1}$ for any triangle $j$. The global mapping $D^{-1}$ is computed by finding a triangle $j$ to which $\mathbf{x}$ maps inside. In the case of $k$ multiple overlapping meshes then such a point might not be unique and so we take the average of the displacements $d$ so that the inverse mapping is

$$\begin{aligned} &\mathbf{X} = D^{-1}(\mathbf{x}) = D_j^{-1}(\mathbf{x}) \text{ for a triangle } j \in L \text{ such that} \\ &\quad |d_j| = \min_j \{|d_j| : 0 \leq \alpha_j, \beta_j, 1 - \alpha_j - \beta_j \leq 1\} \end{aligned} \tag{4}$$

In this way the displacement mapping fuses together overlapping surface measurements in space.

### 4.1. Limitations of Displacement Mapping

Displacement mapping is only valid when the detail mesh is near to the base mesh. If this is not the case then it is likely that the mapping will fail to be a bijection. Several points on the data $H_k^i$ will map to the same point on the base mesh or not map at all to the base mesh.

We assume that the initial base mesh is close enough to the first frame of the data for a bijective mapping to exist. In the case of a user defined base mesh this requires that the base mesh is fitted to the first frame of the data. If $B$ is automatically defined through decimation and fusion then an initially bijective mapping can be enforced using the mesh simplification algorithm introduced in [4]. For subsequent frames we require that the base mesh move in such a way that a bijective mapping is maintained between the base mesh and the captured data. In the following section we suggest that this can be achieved using a basis of piecewise rigid transformations given by 1.

## 5. Representation of Coarse Dynamics

In this section we present the algorithm developed to estimate a segmentation of the base mesh together with a set of rigid transformations for each segment at each frame for a given multiple view sequence of 3D surface measurements and initial base mesh $B$. Since we are considering 3D surface deformations we might expect that locally these deformations are close to rigid (this may not be true for scenes containing hair or liquid spray for example). We automatically segment these approximately rigidly transforming pieces and compute their rigid transform at each frame. Such a basis of piecewise rigid transformations spans the domain of all possible deformations since in the least compact case each vertex $\mathbf{b}_j$ could have its own set of translations for each frame. The problem is coupled since we must decide on both the segmentation of the base mesh and the transformations on each segment. Essentially we compute a rigid transformations on the base mesh $B$ using an Iterative Closest Point (ICP) algorithm between each frame. Any vertices of $B$ which transform further than a given user specified error tolerance are rejected from the segment. Once a segment has been calculated we iterate the process until all vertices have been segmented.

### 5.1. Closest Point Computation

Since a large number of ICPs are performed we require a computationally efficient closest point algorithm. In a manner similar to the "normal shooting" method of Chen et al. [2] we define the closest point by the displacement mapping from a decimated version $\hat{H}_k^i$ of each view at each frame of the data $H_k^i$. We ensure that there is always such a mapping between $H_k^i$ and $\hat{H}_k^i$ by mesh decimation of $H_k^i$ in a

manner that ensures there is always a bijective displacement mapping between the two surfaces as described in [4].

For a point $\mathbf{x}$ then, we consider the closest point on $\hat{H}_k^i$ to be $D^{-1}(\mathbf{x})$ given by equation (4) with $L = \hat{H}_k^i$. This can be found efficiently since it involves solving (4) for each of the triangles in $\hat{H}_k^i$ and typically $\hat{H}_k^i$ has only $5\%$ of the triangles in $H_k^i$.

### 5.2. Multi-frame Rigid Transform Estimation

In order to segment and calculate a sequence of rigid transformations for a segment of $B$ we minimise the error between corresponding points over a sequence of transformations $T_j^i$ for $i = 1 \dots M$. The approach is similar to ICP although here the whole sequence of transformations is minimised in order not to bias the segmentation towards earlier frames. To segment the base vertices $\mathbf{b}_j \in B$ we solve for $T_j^i$ by minimising

$$\min_{T_j^i} \sum_{i=0}^{M} |T_j^i \mathbf{b}_j - D^{-1}(T_j^i \mathbf{b}_j)|^2 \qquad (5)$$

Segmentation is performed by only considering those vertices $\mathbf{b}_j$ that transform to points within a given tolerance of their corresponding points. So the above minimisation is performed on the subset $B_l$ of $B$ which contains those $\mathbf{b_j}$ such that

$$B_l = \{\mathbf{b}_j : D^{-1}\mathbf{b}_j \in \hat{H}_i^k \text{ with } |d| < tol\} \qquad (6)$$

Where $tol$ is a user defined tolerance on the maximum geometric error. After an iteration of the minimisation (5) the subset $B_l$ is recalculated for all unregistered points in $B$. The process is iterated until all the vertices in $B$ have been registered.

### 5.3. Segmentation Algorithm

The above minimisation (5) and (6) can be solved to give the sequence transforms $T_j^i$ and to compute the segment $B_l$. We now describe in detail how this is employed to segment the whole base mesh $B$.

It is well known that, in order to converge quickly and accurately ICP requires an accurate initial rigid transformation and so to initialise the transformations we make two adjustments to the above minimisation. First instead of solving (5) for all frames we solve up to frame $M' = 0 \dots M$. We assume that the base mesh $B$ is close to frame $0$ and so $T_j^0$ is close to the identity for all $j$. Then on each increment of $M'$ we initialise the new transform $T_j^k$ to the previous $T_j^{k-1}$. When $M' = M$ then all the transforms have been initialised and the minimisation (5) is performed over the whole sequence and so the final segmentation is still not order dependent.

Secondly, in order to include as many points as possible in each segment and to initialise the transformation, we require a large value of $tol$ initially. Once the minimisation (5) has converged for this large value we decrease $tol$ and perform the minimisation again until either $tol$ is as small as a user tolerance $UserTol$ or the number of vertices in the segment is less than 3 which is when the minimisation (5) becomes underdetermined. We have found that an initial value of 3 times $UserTol$ and a decrease by a factor of $0.6$ to be practical

The final segmentation algorithm is then,

Fit baseModel to frame $0$
While ( $B$ has $> 3$ vertices ) {
  $tol$=3*$UserTol$
  For( Frames $M' = 0 \dots M$ ) {
    While( $tol > UserTol$ ) {
      Solve minimisation problem (5) on verts $B_l$
      given by (6) for frames $0 \dots M'$
      $tol = tol \times 0.6$ }
  }
  Remove $B_l$ from $B$
  Increment $l$
}

### 5.4. Exploiting Spatio-Temporal Coherence

By assuming continuous deformations in both space and time we may enhance the above algorithm to improve the tracking and to interpolate missing data.

The algorithm as described above is concerned with spatially unconnected trajectories of points which are clustered into segments. Spatial coherence can be exploited by stipulating that vertices of a given segment are connected to each other. We have found that a more physically realistic representation is achieved if we stipulate that segments are not disconnected and so once the closest point correspondence has been calculated using (6) we reject those vertices which do not belong to the largest connected piece.

Secondly temporal correspondence is exploited to deal with occlusions. In the above algorithm if a vertex is occluded then it will not be included in the segment. This condition can be relaxed by allowing vertices to "disappear" from the segment for a user defined amount of time $t_c$ which is the maximum amount of occlusion time. Thus if, during the optimisation a vertex is included in a segment $B_l$ for $t_c \leq M$ frames it is included in the segment even if (6) is violated at other frames.

Finally spatio-temporal coherence is exploited in a post-processing step whereby the transformations are interpolated. This is done so that any unregistered vertices (those that were left in the segment when the minimisation (5 became undetermined) are transformed and to smooth the transformations to maintain continuity. In both cases the

interpolation is an average of neighbouring vertices over space and time so that the transformation of an unregistered vertex $\mathbf{b}_j^i$ is interpolated in space and time as

$$T_j^i \mathbf{b}_j = \frac{1}{3} \sum_{k=-1}^{1} \frac{1}{valence} \sum_l T_l^k \mathbf{b}_l$$

where $l$ indexes the vertices surrounding $\mathbf{b}_j$ in the mesh and only registered vertices are included in the summation.

**Figure 1. Segmentation of composite scene, cloth, hand and face models**

## 6. Representation of Surface Detail

Having estimated the coarse transformations for the base mesh we can then represent the detailed deformations on top of this by creating displacement maps at each frame.

### 6.1. Displacement Video

A displacement image is created by texture mapping the distances $d$ given by equation (3) onto an image. In general the base meshes are small enough for a texture map to be specified manually although standard techniques could also be used. For each pixel $(l, m)$ of the image we determine texture coordinates and hence barycentric coordinates for a triangle in $L$. At each frame $i$ and each view $k$ equation (3) is solved by computing the intersection of the interpolated normal with each detailed triangle in $H_k^i$ to give a displacement $d$. In the case of multiple views this normal may intersect multiple detailed triangles where the views overlap and so we take an average of these displacement thus spatially fusing the detail. The displacement is then quantised and stored as a grey scale pixel value $I^i(l, m)$. We have found that in general, quantisation into $2^{10}$ levels gives acceptable detail encoding.

Displacement images are calculated from each mesh of the piecewise rigidly transformed base mesh $B^i$ to the original data $H_k^i$ to give a displacement video.

### 6.2. Reconstruction

The representation comprises the base mesh $B$ with texture coordinates, a set of transformations $T_j^i$ for $i = 0 \dots M$ and $j < N$ and the displacement video. First the transformations are applied to obtain the coarsely transforming base mesh (1). The detail is then reconstructed by converting each pixel of the displacement image into a vertex of the reconstructed mesh. Each pixel is mapped via its texture coordinates to a barycentric coordinate and then displaced along the interpolated normal by the unquantised grey scale distance via the mapping equation (3). Thus the user defined resolution of the image gives the resolution of the reconstructed mesh. The connectivity of the reconstructed mesh is given by connecting neighbouring pixels.

### 6.3. Interpolation

A fused and integrated reconstruction with the same connectivity everywhere is possible only if the same pixels are reconstructed for each frame of the displacement video. However in general, due to missing detail, each frame of the displacement video will have different missing pixels. In order to ensure an integrated reconstruction (with the same connectivity at each frame) we spatio-temporally interpolate any missing pixels by taking an average of the nearest surrounding pixels in the two time directions $\pm i$ and the four pixel directions $\pm l$ and $\pm m$. A missing pixel is filled if that pixel exists at any other frame so that $I^i(l, m)$ is always filled if $I^s(l, m)$ is filled for any $s \neq i$.

It might be thought that there is a danger of erroneously filling holes that are really in the data (such as the mouth in the face example). In general we have found that the base transformations track the deformations closely enough so that large holes are successfully resolved by the base mesh and so will not be filled by interpolation of the displacement image.

## 7. Results

We tested the algorithm on four data sets - a face, a piece of cloth, a hand and a composite scene of an arm moving a block. Each example was captured from three views and at 25 fps. We used a manually fitted model in the face and cloth sequences and a decimated fused first frame for the other sequences.

Figure 1 shows the segmentation of the base meshes. The composite example shows the correct segmentation of rigid elements of a scene. The face model clearly segments regions of different deformations. Large segments are visible around the skull which deforms more or less rigidly whereas small segments appear around regions of non-rigid deformation such as the mouth and cheeks. Furthermore the individual fingers in the hand example have been segmented.

The cloth example in figure 4 shows that the method will also work on sequences which are noisy, incomplete and non-rigid. Figure 3 shows that the representation is easily

controlled. The initial frame of the displacement video is edited and the edit is copied onto the subsequent frames. The resulting edit is transmitted through the sequence.

The hand example in figure 2 shows a limitation of the method when applied to very complicated occluded data. Here the control model is insufficiently complex to track the movements of the data which has many occlusions. For this reason the segmentation is too coarse to track the deformations with enough resolution. Although individual fingers are segmented, they are segmented as only one rigidly transforming piece. The reconstructed mesh shows gaps in the displacement mapping where the base mesh is too far from the detail. Video clips can be seen at www.ee.surrey.ac.uk/CVSSP/VMRG/VCPanimation.html.

## 8. Conclusion

We have presented a fused representation of 3D data captured at video rate and multiple views. The coarse deformations are approximated as piecewise rigid transformations of a base mesh and the detailed deformations are fused as a displacement video. The method is applied to a diverse range of captured data and is shown to work well for reasonably complete data.

## Acknowledgements

The authors would like to acknowledge the support from EPSRC Visual Media Research Platform Grant GR/13576.

## References

[1] M. J. Black and P. Anandan. The robust estimation of multiple motions:parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75—104, January 1996.

[2] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10(3), 1992.

[3] G. Collins and Hilton A. A rigid transform basis for animation. Submitted to Vision, Video and Graphics, March 2005.

[4] Gordon Collins and Adrian Hilton. Mesh decimation for displacement mapping. In *Eurographics Short Papers 02*, Sept 2002.

[5] T. Kanade. Virtualized reality: putting reality into virtual reality. In *2nd International Workshop on Object Representation for Computer Vision ECCV*, 1996.

[6] Venkat Krishnamurthy and Marc Levoy. Fitting smooth surfaces to dense polygon meshes. *Proceedings of SIGGRAPH 96*, pages 313–324, August 1996. ISBN 0-201-94800-1. Held in New Orleans, Louisiana.

[7] Aaron Lee, Henry Moreton, and Hugues Hoppe. Displaced subdivision surfaces. *Proceedings of SIGGRAPH 2000*, pages 85–94, July 2000. ISBN 1-58113-208-5.

[8] 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.

[9] Z. Li, B. Curless, and S.M. Seitz. Spacetime stereo: Shape recovery of dynamic scenes. 2003.

[10] Z. Li, N. Snavely, B. Curless, and S.M. Seitz. Spacetime Faces: High-resolution capture for modelling and animation. 2004.

[11] P. Lindsey and A. Blake. Real-time tracking of surfaces with structured light. pages 619—628, 1994.

[12] S.K. Nayar, M. Watanabe, and M. Noguchi. Real-time focus range sensor. pages 995—1001, 1995.

[13] M. Proesmans and L. VanGool. Active acquisition of 3d shape for moving objects. pages 647—650, 1996.

[14] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. 2002.

[15] W. Sun, A. Hilton, R. Smith, and J.Illingworth. Layered animation of captured data. *Visual Computer: International Journal of Computer Graphics*, 17(8):457—474, 2001.

[16] Y. Wang, X. Huang, C-S. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. High Resolution Acquisition, Learning and Transfer of Dynamic 3-D Facial Expressions. 2004.

[17] M. Watanabe and S.K. Nayar. Telecentric optics for computer vision. pages 439—451, 1995.

[18] I.A. Ypsilos, A. Hilton, and S. Rowe. Video-rate Capture of Dynamic Face Shape and Appearance. In *IEEE Face and Gesture Recognition*, 2004.

[19] I.A. Ypsilos, A. Hilton, A. Turkmani, and P. Jackson. Speech Driven Face Synthesis from 3D Video. In *IEEE Symposium on 3D Data Processing, Visualisation and Transmission*, 2004.

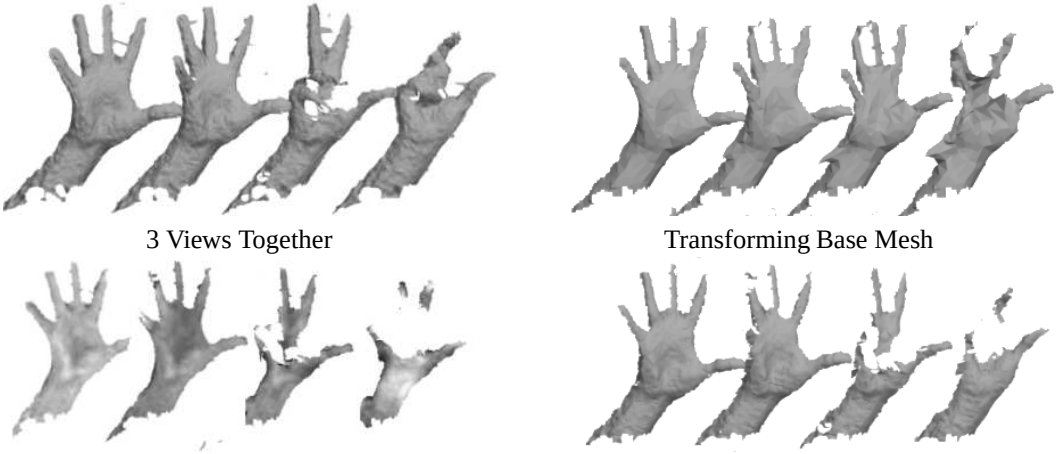


**Figure 2. Hand Sequence (frames 0,5,10,15**

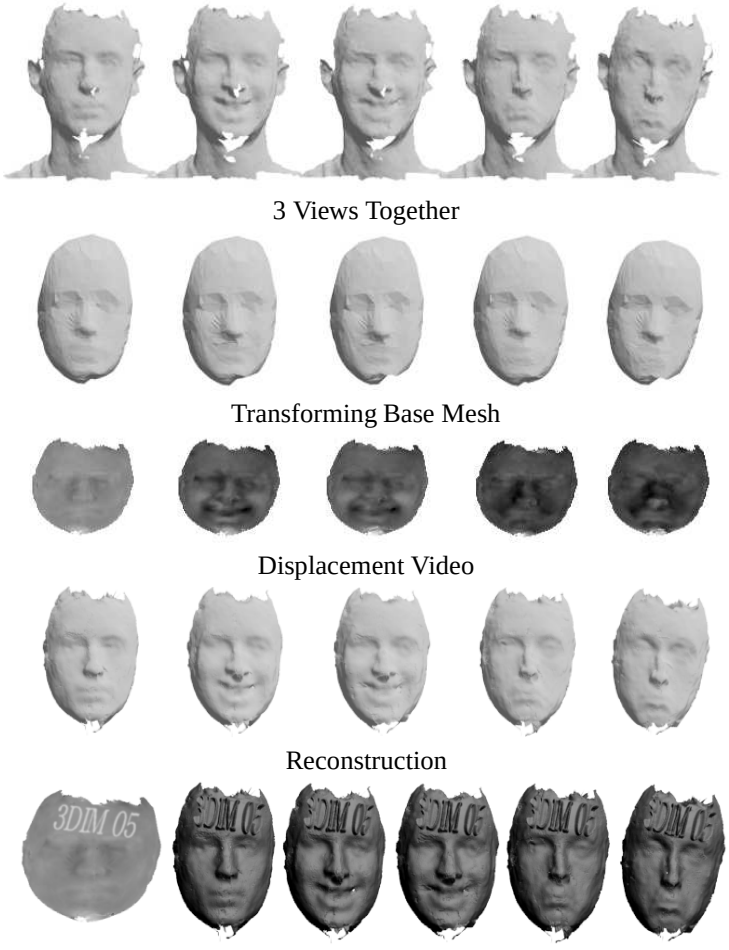


**Figure 3. Face Sequence (frames 0,10,20,30,40)**

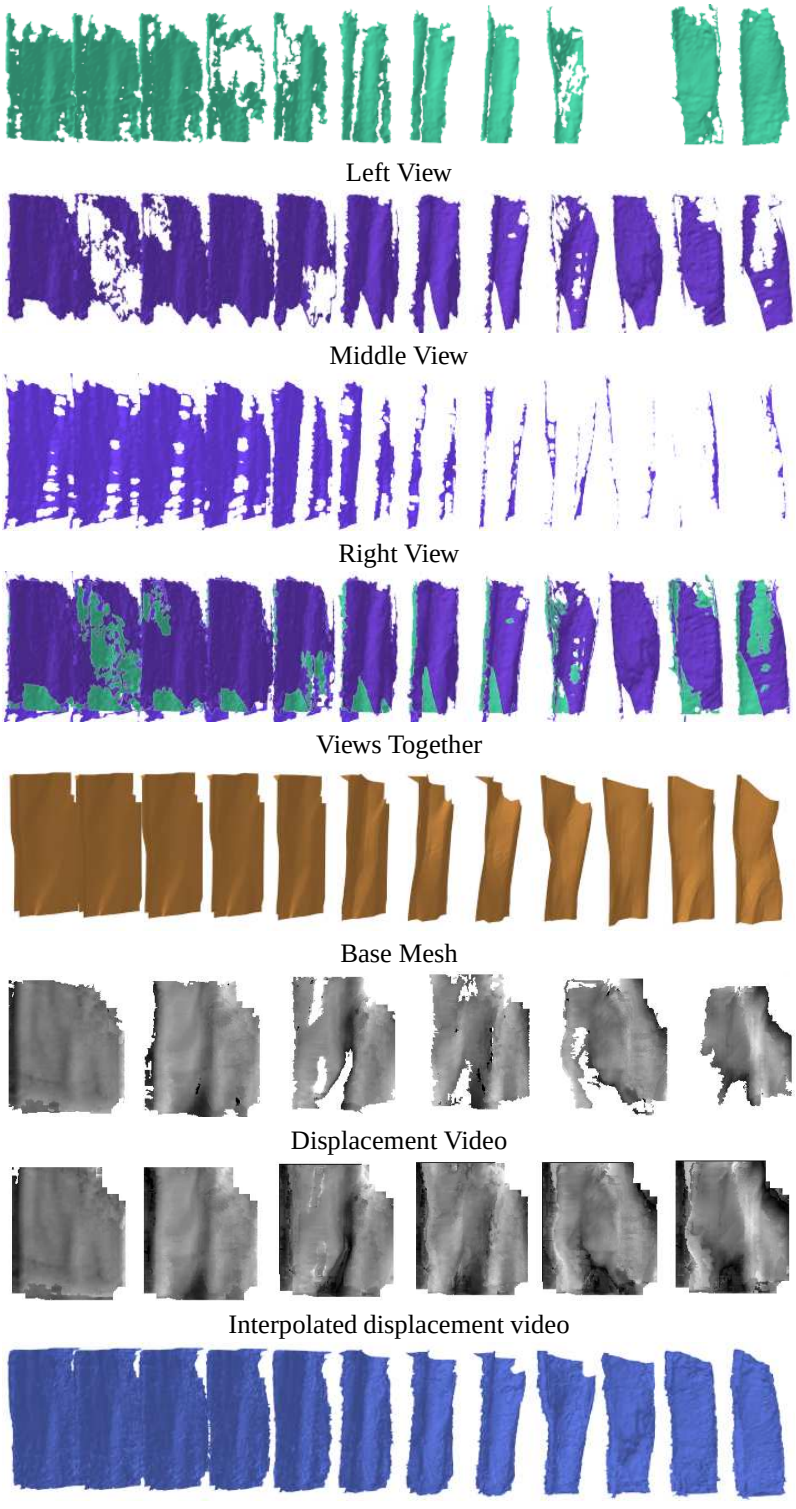


**Figure 4. Cloth Sequence (frames 0 - 12)**